

AR11

DIAGNOSTIC TEST II
MD-11-DZARB-B

EP-DZARB-B-DL

MAY 1978

COPYRIGHT © 74-76

digital

FICHE 1 OF 1

MADE IN USA

Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6
Frame 7	Frame 8	Frame 9	Frame 10	Frame 11	Frame 12
Frame 13	Frame 14	Frame 15	Frame 16	Frame 17	Frame 18
Frame 19	Frame 20	Frame 21	Frame 22	Frame 23	Frame 24
Frame 25	Frame 26	Frame 27	Frame 28	Frame 29	Frame 30
Frame 31	Frame 32	Frame 33	Frame 34	Frame 35	Frame 36
Frame 37	Frame 38	Frame 39	Frame 40	Frame 41	Frame 42
Frame 43	Frame 44	Frame 45	Frame 46	Frame 47	Frame 48

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-07ARB-R-D
PRODUCT NAME: AR-11 DIAGNOSTIC TEST II
DATE: MAY 21, 1976
MAINTAINERS: DIAGNOSTIC GROUP

FIRST PRINTING, 1976

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1976, 1976 BY DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

THIS DIAGNOSTIC EXERCISES THE "AR-11" ANALOG CIRCUITRY. THE PROGRAM WHEN STARTED WILL TYPE OUT THE PROGRAM TITLE. A MESSAGE IS THEN PRINTED GIVING THE LETTER DESIGNATORS TO BE TYPED TO RUN ANY ONE OF THE FOUR (4) SEPERATE TESTS OF WHICH THIS PROGRAM IS COMPRISED. THE PROGRAM THEN TYPES A 'CR .' AND THEN WAITS IN A KEYBOARD MONITOR MODE FOR A LETTER TO BE TYPED. ALTHOUGH THESE TESTS MAY BE RUN IN ANY ORDER IT IS IMPERATIVE THAT TEST I IS RUN FIRST AND PROVED FULLY OPERATIONAL.

THE PROGRAM IS SET UP TO GIVE THE OPERATOR AS MUCH CONTROL OVER THE PROGRAM AS POSSIBLE VIA THE TELETYPE. TYPING A '"C' (OBTAINED VIA TYPING THE 'CNTR' AND 'C' KEYS SIMULTANEOUSLY) WHILE RUNNING ANY TEST WILL ENABLE THE PROGRAM TO RETURN TO THE KEYBOARD MONITOR AND AWAIT A NEW LETTER DESIGNATOR TO BE TYPED. TYPING A '"A' WHILE IN MONITOR MODE WILL ENABLE THE LETTER DESIGNATORS TO BE RETYPED.

UNLIKE AR-11 TEST I, THIS PROGRAM DOES NOT DETERMINE IF ADDITIONAL AR-11'S ARE CONNCTED. TO RUN ANOTHER AR-11, THE OPERATOR MUST SUPPLY THE BUS ADDRESS INDIVIDUALLY.

2. REQUIREMENTS (EQUIPMENT)

- A. PDP-11 COMPUTER WITH 8K OF MEMORY
- B. TELETYPE
- C. AR11 HEX OPTION MODULE INSTALLED
- D. VOLTAGE STANDARD (E.D.C.)
- E. VR14 OR STORAGE SCOPE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY TAPES.

4. STARTING PROCEDURE

THE PROGRAM STARTING ADDRESS IS '200'.
THE RESTART ADDRESS IS '204'.

5. CONSOLE SWITCH SETTINGS

THIS PROGRAM HAS BEEN MODIFIED TO RUN WITH OR WITHOUT A HARDWARE SWITCH REGISTER.

- A. ALL SWITCHES SHOULD BE DOWN (0) WHEN THE PROGRAM IS STARTED.
- B. REFER TO THE INDIVIDUAL TEST DESCRIPTIONS FOR APPLICABLE CONSOLE SWITCH SETTINGS

WHILE IN KEYBOARD MONITOR MODE, TYPING 'S WILL ENABLE THE SOFTWARE SWITCH REGISTER TO BE LOADED FROM THE TELETYPE WITHOUT HALTING THE PROCESSOR.

* TYPE 'CARRIAGE RETURN' (CR) TO TERMINATE ALL INPUT DATA.

6.

QUICK STATIC REGISTER TEST

A. THIS TEST IS DESIGNED TO PROVIDE A REGISTER VERIFICATION TEST IN THIS PROGRAM.
IF THIS SUB-TEST FAILS, AR-11 TEST 1 SHOULD BE LOADED.

B. STARTING SEQUENCE

1. TYPE 'A' TO RUN THE QUICK REGISTER TEST.
2. THE PROGRAM WILL THEN EXECUTE THE QUICK REGISTER TEST.

C. CONTROL SWITCHES

1. TYPING 'C' AT ANY TIME WILL ENABLE THE PROGRAM TO EXIT
AND RETURN TO THE MONITOR.

CONSOLE SWITCHES -----	FUNCTION -----
CONSOLE SW15=1	HALT ON ERROR
CONSOLE SW13=1	INHIBIT ERROR TYPEOUTS

D. ERRORS

THIS PROGRAM USES THE DIAGNOSTIC 'SYSMAC' PACKAGE FOR
ERROR REPORTING AND TYPEOUT. REFER TO THE "ERROR POINTER TABLE"
FOR TYPE OF LOGIC ERROR AND DESCRIPTION.

E. RESTRICTIONS

NONE

F. EXECUTION TIME

IT TAKES APPROXIMATELY 30 SECONDS TO THIS TEST.

7.

POINT PLOT VISUAL DISPLAY TEST

A. THIS TEST IS DESIGNED TO AID IN THE ADJUSTING AND ALIGNMENT OF THE VR14 OR STORAGE SCOPE SCOPE ON THE AR-11 DISPLAY CONTROL.

B. STARTING SEQUENCE

1. TYPE 'B' TO RUN THE VISUAL DISPLAY TEST.
2. THE PROGRAM WILL THEN EXECUTE THE VISUAL DISPLAY TEST.

C. CONTROL SWITCHES

1. TYPING 'C' AT ANY TIME WILL ENABLE THE PROGRAM TO EXIT AND RETURN TO THE MONITOR.

CONSOLE SWITCHES	FUNCTION
CONSOLE SW00=0	LOOP THRU DISPLAY TEST
CONSOLE SW00=1	SELECT TEST IN SW 00-02
CONSOLE SW07=0	HORIZONTAL SETTling TEST <SETTLING TEST>
CONSOLE SW07=1	VERTICAL SETTling TEST <SETTLING TEST>
CONSOLE SW05=0	STORAGE SCOPE NOT CONNECTED <PHOSPHOR TEST>
CONSOLE SW05=1	STORAGE SCOPE CONNECTED <PHOSPHOR TEST>
CONSOLE SW00-02=0	DISPLAY A HORIZONTAL LINE
CONSOLE SW00-02=1	DISPLAY A VERTICAL LINE
CONSOLE SW00-02=2	DISPLAY A SQUARE
CONSOLE SW00-02=3	DISPLAY AN "X"
CONSOLE SW00-02=4	DISPLAY SETTling TEST
CONSOLE SW00-02=5	DISPLAY CHARACTER TEST
CONSOLE SW00-02=6	DISPLAY CHANNEL TEST <VR14>
CONSOLE SW00-02=7	DISPLAY ERASE AND PHOSPHOR <STORAGE SCOPE>

D. ERRORS

NO PROVISIONS ARE MADE FOR LOGIC ERRORS. THE ONLY ERRORS IN THIS TEST ARE CHECKED VISUALLY.

E. RESTRICTIONS

IF VR14, CHANNEL SWITCH MUST BE SET TO "1 & 2" POSITION.
IF STORAGE SCOPE, POWER MUST BE APPLIED.

F. EXECUTION TIME

IT TAKES APPROXIMATELY 90 SECONDS TO THIS TEST.

B. VISUAL DISPLAY TEST DESCRIPTIONS

DISPLAY HORIZONTAL LINE

A HORIZONTAL LINE IS DISPLAYED ON THE SCOPE BY INITIALLY SETTING THE X AND Y DAC'S TO ZERO AND THEN INCREMENTING THE X VALUE WHILE HOLDING THE Y VALUE AT 1000. THE POINTS ARE DISPLAYED USING THE DISPLAY INTERRUPT ENABLED.

DISPLAY VERTICAL LINE

A VERTICAL LINE IS DISPLAYED ON THE SCOPE IN THE SAME MANNER AS FOR A HORIZONTAL LINE EXCEPT NOW THE Y VALUE IS INCREMENTED WHILE HOLDING THE X VALUE AT 1000.

DISPLAY SQUARE

A SQUARE IS DISPLAYED BY INITIALLY SETTING THE X AND Y VALUES TO NEGATIVE FULL SCALE, THEN X IS INCREMENTED TO POSITIVE FULL SCALE (BOTTOM LINE) THEN Y IS INCREMENTED TO POSITIVE FULL SCALE (RIGHT LINE) THEN X IS DECREMENTED TO NEGATIVE FULL SCALE (TOP LINE) AND FINALLY Y IS DECREMENTED TO NEGATIVE FULL SCALE (LEFT LINE). MODE B1 (INTENSIFY ON LOADING X) AND MODE 10 (INTENSIFY ON LOADING Y) ARE USED.

DISPLAY X

AN X IS DISPLAYED BY INITIALLY SETTING THE X AND Y VALUES TO NEGATIVE FULL SCALE AND THEN INCREMENTING BOTH TO POSITIVE FULL SCALE (LOWER LEFT TO UPPER RIGHT DIAGONAL) THEN X IS RESET TO NEGATIVE FULL SCALE, Y REMAINS AT POSITIVE FULL SCALE AND THEN X IS INCREMENTED WHILE Y IS DECREMENTED UNTIL BOTH REACH FULL SCALE AGAIN (UPPER LEFT TO LOWER RIGHT DIAGONAL). MODE 01 (INTENSIFY ON LOADING X) IS USED.

DISPLAY SETTLING TEST

A TWO CYCLE SQUARE WAVE WILL BE DISPLAYED TO TEST THE SETTLING DELAY. IF A SETTLING PROBLEM EXISTS, THE LEADING EDGES WILL APPEAR TO BE ROUNDED. THE PROGRAM PLOTS A LINE AT THE MINIMUM AXIS VALUE. UPON COMPLETION, ANOTHER LINE IS PLOTTED AT THE MAXIMUM VALUE. THIS IS REPEATED WITH THE RESULT BEING A SQUARE WAVE. SWITCH BIT 7 DETERMINES HORIZ., OR VERT. SETTLING PATTERN.

DISPLAY ALPHA-NUMERIC CHARACTER SET

THE ALPHABET AND NUMBERS 0 THRU 9 ARE DISPLAYED.
THE FIRST ROW CONSISTS OF THE LETTERS 'A' THRU 'M'. THE SECOND CONTAINS
THE LETTERS 'N' THRU 'Z'. THE LAST LINE CONTAINS THE NUMBERS
'0' THRU '9'.

DISPLAY CHANNEL 1 AND CHANNEL 2 <VR14>

THE TEXT "CHANNEL 1" IS DISPLAYED ON CHANNEL 1 SWITCH POSITION.
THE TEXT "CHANNEL 2" IS DISPLAYED ON CHANNEL 2 SWITCH POSITION.
THE COMBINED MESSAGE WILL APPEAR IF THE CHANNEL SELECTOR SWITCH IS
IN THE 1 & 2 POSITION.

PHOSPHOR AND ERASE TEST

THIS TEST PROVIDES A METHOD OF CHECKING FOR PHOSPHOR BURNS ON THE
SCREEN. THIS ROUTINE WILL FIRST ERASE THE STORAGE SCOPE SCREEN.
A DESCENDING HORIZONTAL LINE IS DISPLAYED IN "STORE" MODE. THE RESULT
IS AN INTENSIFICATION OF THE ENTIRE SCREEN. IF EXECUTED USING A VR14,
OR DISPLAY SCOPE THE RESULT WILL BE ONLY A DESCENDING LINE.

9. A TO D CALIBRATION TEST

A. THE 'A/D CALIBRATION' TEST IS DESIGNED TO ACCEPT AN INPUT FROM THE TELETYPE TO INDICATE THE TYPE OF SYNC (EXTERNAL, INTERNAL OR AR-11 CLOCK TO BE USED AND THEN TAKES CONTINUOUS CONVERSIONS USING THE 'CH.' SELECTED VIA THE CONSOLE SWITCHES. THESE SETTINGS MAY BE CHANGED AT ANY TIME. THIS CAN ALSO BE USED FOR FINDING 50-50 POINT AND FOR FINDING MIDDLE OF A STATE FOR SUBSEQUENT REPEATABILITY TEST.

B. STARTING SEQUENCE

1. TYPE 'C' TO RUN THE A/D CALIBRATION TEST.
2. THE TEST HEADER PLUS A REQUEST FOR A SYNC TYPE WILL THEN BE TYPED.
3. TYPE IN THE DESIRED SYNC, 'I' FOR INTERNAL, 'E' FOR EXTERNAL 'C' FOR AR-11 CLOCK FOLLOWED BY 'CR'.
4. THE TEST WILL START.

C. CONTROL SWITCHES

1. 'A (CONTROL A)

TYPING 'A' WILL ENABLE A NEW SYNC TYPE TO BE ENTERED.

2. 'C (CONTROL C)

TYPING 'C' WILL CAUSE THE PROGRAM TO EXIT THE CALIBRATION TEST AND RETURN TO THE MONITOR.

3. CONSOLE SWITCH

FUNCTION

CONSOLE SW '0-3'	CHANNEL SELECT
CONSOLE SW 05 = 0	SELECT BIPOLAR CHANNEL
CONSOLE SW 05 = 1	SELECT UNIPOLAR CHANNEL
CONSOLE SW 06 = 0	CONTINUOUS SAMPLES
CONSOLE SW 06 = 1	FREEZE ON CURRENT DATA
CONSOLE SW 07 = 0	DISPLAY DATA ON X AXIS TO CHECK FOR AD INTERACTION WITH X DA
CONSOLE SW 07 = 1	DISPLAY DATA ON Y AXIS TO CHECK FOR AD INTERACTION WITH Y DA
CONSOLE SW 10 = 0	DISPLAY VALUE ON THE SCREEN
CONSOLE SW 10 = 1	PRINT CONVERSION VALUE

D. CALIBRATION ERRORS

NONE

10. A TO D REPEATABILITY TEST

A. THIS TEST REQUESTS A CH.(S) AND A COUNT SPREAD OF '1-8'
AND MODE OF OPERATION TO BE TYPED IN BY THE OPERATOR.
A SERIES OF '512' CONVERSIONS ARE THEN TAKEN ON THE INPUT CH.(S)
CONVERSIONS ARE THEN AVERAGED OUT AND IF THE COUNT SPREAD IS
FOUND TO BE GREATER THAN REQUEST, THE RESULTS OF THE CONVER-
SIONS ARE TYPED OUT. A SINGLE CHANNEL OR A SERIES OF CHANNELS
MAY BE TESTED VIA TYPING EITHER 'N(CR)' TO SELECT A SINGLE
CHANNEL OR 'N,N(CR)' TO TEST A SERIES OF CHANNELS.

B. STARTING SEQUENCE

1. TYPE 'D' TO RUN THE 'REPEATABILITY' TEST.
2. A REQUEST IS THEN MADE FOR CH.(S) TO BE TESTED AND
COUNT SPREAD (RANGE IN WHICH ALL 512 COUNTS MUST FALL
FOR THE CH. TO BE CONSIDERED ACCEPTABLE).
3. IF THE CHANNEL IS FOUND TO BE WITHIN THE SELECTED COUNT
SPREAD, THE PROGRAM WILL EITHER CONTINUE TO THE NEXT
CHANNEL IF SELECTED OR RETEST THE CURRENT CHANNEL.

C. CONTROL SWITCHES

1. 'A (CONTROL A)

TYPING A 'A' WHILE THE PROGRAM IS RUNNING WILL ENABLE
A NEW CH.(S) AND COUNT SPREAD TO BE SELECTED.

2. 'C (CONTROL)

TYPING 'C' WILL CAUSE THE PROGRAM TO EXIT THE 'REPEATABILITY'
TEST AND RETURN TO THE MONITOR.

3. CONSOLE SWITCHES

FUNCTION

CONSOLE SW 10=0	PRINT ERRORS ONLY
CONSOLE SW 10=1	PRINT OUT ALL CONVERSIONS
CONSOLE SW 13=0	PRINT ERRORS
CONSOLE SW 13=1	INHIBIT ERROR PRINTOUTS
CONSOLE SW 15=0	DO NOT HALT UPON REPEATABILITY REPORT
CONSOLE SW 15=1	HALT UPON REPEATABILITY REPORT

D. REPEATABILITY ERRORS

ON ENCOUNTERING AN ERROR (CONSOLE SWITCHES DOWN) THE ERROR DATA IS TYPED OUT.

1. ERROR FORMAT

CH.		MI		AV		LO								
A		B		C		D								
LO	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	MI		
E	F	G	H	I	J	K	L	M	N	O	P	Q		

WHERE:

- A=CHANNEL BEING TESTED
- B=THE HIGHEST READING OF THE '512' CONVERSIONS
- C=THE AVERAGE READING OF THE '512' CONVERSIONS
- D=THE LOWEST READING OF THE '512' CONVERSIONS
- E=NUMBER OF COUNTS 'OUT OF RANGE' LOWER THAN 5 COUNTS
- F-J=NUMBER OF COUNTS IN EACH PART LOWER THAN AVERAGE.
- K=NUMBER OF COUNTS AT AVERAGE OF THE '512'
- L-P=NUMBER OF COUNTS IN EACH PART HIGHER THAN AVERAGE.
- Q=NUMBER OF COUNTS 'OUT OF RANGE' HIGHER THAN 5 COUNTS

E. RESTRICTIONS

1. IF A SECOND CH. IS ENTERED, IT MUST BE LARGER THAN THE FIRST CH. OR THE INPUT WILL NOT BE ACCEPTED.

11. A TO D RECOVERY TEST

A. THE "RECOVERY TEST" IS DESIGNED TO DETERMINE THE INTER-CHANNEL SETTLING CAPABILITY OF THE "AR-11 A TO D". THE TEST REQUESTS FOR TWO (2) CH. INPUTS TO BE TYPED IN. THE TEST THEN TAKES A SERIES OF SIXTEEN (16) CONVERSIONS (8 ON EACH CH.) AND THEN TYPES OUT THE "8" AVERAGE VALUES IN THE ORDER THEY WERE TAKEN ON THE SECOND CH.

B. STARTING SEQUENCE

1. TYPE 'E' TO RUN THE RECOVERY TEST.
2. A REQUEST IS THEN MADE FOR THE CH.S TO BE TESTED.
3. TYPE 'N,N (CR)' WHERE 'N' IS ANY CH.
4. THE PROGRAM WILL THEN TAKE CONTINUOUS CONVERSIONS TYPING OUT THE CONVERSION VALUES FOR THE SECOND CH.

EXAMPLE:

CH. A XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

WHERE:

A = THE SECOND CH.
X = THE "8" CONVERSIONS TAKEN ON THAT CH.

C. CONTROL SWITCHES

1. "A (CONTROL A)

TYPING A "A" WILL ENABLE A NEW SET OF CH.S TO BE ENTERED.

2. "C (CONTROL C)

TYPING A "C" WILL ENABLE THE PROGRAM TO RETURN TO THE MONITOR.

3. CONSOLE SWITCHES

FUNCTION

CONSOLE SW 10 = 0
CONSOLE SW 10 = 1

PRINT RECOVERY REPORT
INHIBIT PRINTING OF RECOVERY REPORT

D. RESTRICTIONS

NONE

12. MISC. INFORMATION

ACT-11 OR XXDP

IF THE PROGRAM WAS LOADED FROM ACT-11 OR CHAINED FROM XXDP, THE VISUAL DISPLAY TEST WILL BE RUN.

APT

THE APT HOOKS HAVE BEEN INSTALLED BUT NOT TESTED.

13. PROGRAM VARIABLE LOCATIONS

LOCATION SBASE CONTAINS THE AR-11 STARTING DEVICE ADDRESS <170400>
LOCATION SVECT1 CONTAINS THE AR-11 STARTING VECTOR <300>
LOCATION SNULL CONTAINS THE TTY FILLER CHARACTER
LOCATION SFILLS CONTAINS THE TTY FILLER CHARACTER COUNT

NOTE: IF LOCATIONS SBASE OR SVECT1 ARE CHANGED, THE TEST MUST BE RE-INITIALIZED AT 200.

14. TABLE OF CONTENTS

ATTACHED

13	BASIC DEFINITIONS
129	OPERATIONAL SWITCH SETTINGS
142	TRAP CATCHER
152	STARTING ADDRESS(ES)
157	ACT11 HOOKS
169	APT PARAMETER BLOCK
192	COMMON TAGS
239	APT MAILBOX-EABLE
311	ERROR POINTER TABLE
493	T1 DISPLAY HORIZONTAL LINE
507	T2 DISPLAY A VERTICAL LINE
537	T3 PINCUSHION TEST (DISPLAY SQUARE)
590	T4 PLOT AN X
632	T5 SCOPE SETTLE TIME TEST
677	T6 PLOT CHARACTER SET
831	T7 CHANNEL 1 CHANNEL 2
884	T10 PHOSPHOR TEST
902	END OF PASS ROUTINE
1028	T11 CALIBRATION ROUTINE
1225	T12 REPEATABILITY TEST
1348	T13 RECOVERY TEST
1432	T14 LOAD DIFFERENT NUMBERS INTO DIFFERENT REG.
1523	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
2059	SCOPE HANDLER ROUTINE
2125	ERROR HANDLER ROUTINE
2174	ERROR MESSAGE TIMEOUT ROUTINE
2222	POWER DOWN AND UP ROUTINES
2274	BINARY TO OCTAL (ASCII) AND TYPE
2352	TYPE ROUTINE
2432	TTY INPUT ROUTINE
2548	READ AN OCTAL NUMBER FROM THE TTY
2587	APT COMMUNICATIONS ROUTINE
2645	TRAP DECODER
2662	TRAP TABLE



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

.TITLE MAINDEC-11-DZARR-B
.COPYRIGHT (C) 1976
.DIGITAL EQUIPMENT CORP.
.MAYNARD, MASS. 01754
.
.PROGRAM BY RAYMOND SHORP
.
.THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
.PACKAGE (MAINDEC-11-DZARR-B2), NOV 21, 1975.
.

.SBTTL BASIC DEFINITIONS

.INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK# 1100

.EQUIV EMT,ERROR ;BASIC DEFINITION OF ERROR CALL
.EQUIV IOY,SCOPE ;BASIC DEFINITION OF SCOPE CALL

.MISCELLANEOUS DEFINITIONS

HT# 11 ;CODE FOR HORIZONTAL TAB
LF# 12 ;CODE FOR LINE FEED
CR# 15 ;CODE FOR CARRIAGE RETURN
CRLF# 200 ;CODE FOR CARRIAGE RETURN-LINE FEED
PS# 177776 ;PROCESSOR STATUS WORD
.EQUIV PS,PSW
STKLMY# 177774 ;STACK LIMIT REGISTER
PIRQ# 177772 ;PROGRAM INTERRUPT REQUEST REGISTER
DSWR# 177570 ;HARDWARE SWITCH REGISTER
DISP# 177570 ;HARDWARE DISPLAY REGISTER

.GENERAL PURPOSE REGISTER DEFINITIONS

R0# X0 ;GENERAL REGISTER
R1# X1 ;GENERAL REGISTER
R2# X2 ;GENERAL REGISTER
R3# X3 ;GENERAL REGISTER
R4# X4 ;GENERAL REGISTER
R5# X5 ;GENERAL REGISTER
R6# X6 ;GENERAL REGISTER
R7# X7 ;GENERAL REGISTER
.EQUIV R6,SP ;STACK POINTER
.EQUIV R7,PC ;PROGRAM COUNTER

.PRIORITY LEVEL DEFINITIONS

PR0# 0 ;PRIORITY LEVEL 0
PR1# 40 ;PRIORITY LEVEL 1
PR2# 100 ;PRIORITY LEVEL 2
PR3# 140 ;PRIORITY LEVEL 3
PR4# 200 ;PRIORITY LEVEL 4
PR5# 240 ;PRIORITY LEVEL 5
PR6# 300 ;PRIORITY LEVEL 6
PR7# 340 ;PRIORITY LEVEL 7

.SWITCH REGISTER SWITCH DEFINITIONS

SW15# 100000
SW14# 40000
SW13# 20000

001100

000011

000012

000015

000200

177776

177774

177772

177570

177570

000000

000001

000002

000003

000004

000005

000006

000007

000000

000040

000100

000140

000200

000240

000300

000340

100000

040000

020000

57	010000	SW12=	10000
58	004000	SW11=	40000
59	002000	SW10=	20000
60	001000	SW09=	10000
61	000400	SW08=	4000
62	000200	SW07=	2000
63	000100	SW06=	1000
64	000040	SW05=	400
65	000020	SW04=	200
66	000010	SW03=	100
67	000004	SW02=	40
68	000002	SW01=	20
69	000001	SW00=	10
70		.EQUIV	SW00,SW0
71		.EQUIV	SW08,SW8
72		.EQUIV	SW07,SW7
73		.EQUIV	SW06,SW6
74		.EQUIV	SW05,SW5
75		.EQUIV	SW04,SW4
76		.EQUIV	SW03,SW3
77		.EQUIV	SW02,SW2
78		.EQUIV	SW01,SW1
79		.EQUIV	SW00,SW0

1=DATA BIT DEFINITIONS (BIT00 TO BIT15)

81		BIT15=	100000
82	100000	BIT14=	400000
83	040000	BIT13=	200000
84	020000	BIT12=	100000
85	010000	BIT11=	40000
86	004000	BIT10=	20000
87	002000	BIT09=	10000
88	001000	BIT08=	4000
89	000400	BIT07=	2000
90	000200	BIT06=	1000
91	000100	BIT05=	400
92	000040	BIT04=	200
93	000020	BIT03=	100
94	000010	BIT02=	40
95	000004	BIT01=	20
96	000002	BIT00=	10
97	000001	.EQUIV	BIT09,BIT9
98		.EQUIV	BIT08,BIT8
99		.EQUIV	BIT07,BIT7
100		.EQUIV	BIT06,BIT6
101		.EQUIV	BIT05,BIT5
102		.EQUIV	BIT04,BIT4
103		.EQUIV	BIT03,BIT3
104		.EQUIV	BIT02,BIT2
105		.EQUIV	BIT01,BIT1
106		.EQUIV	BIT00,BIT0

1=BASIC "CPU" TRAP VECTOR ADDRESSES

109	000004	ERRVEC=	4	11TIME OUT AND OTHER ERRORS
110	000010	RESVEC=	10	11RESERVED AND ILLEGAL INSTRUCTIONS
111	000014	TBITVEC=	14	11"T" BIT
112				

113	000014	TRTVEC= 14	ITRACE TRAP
114	000014	BPTVEC= 14	IBREAKPOINT TRAP (BPT)
115	000020	IOTVEC= 20	IINPUT/OUTPUT TRAP (IOT) **SCOPE**
116	000024	PWRVEC= 24	IPOWER FAIL
117	000030	EMTVEC= 30	IEMULATOR TRAP (EMT) **ERROR**
118	000034	TRAPVEC=34	I"TRAP" TRAP
119	000060	TKVEC= 60	ITTY KEYBOARD VECTOR
120	000064	TPVEC= 64	ITTY PRINTER VECTOR
121	000240	PIRQVEC=240	IPROGRAM INTERRUPT REQUEST VECTOR
122			
123	170400	ABASE=170400	
124	000340	AVECT1=340	
125	000200	APRIOR=200	


```

126
127
128          .SBTTL  OPERATIONAL SWITCH SETTINGS
129          ;*
130          ;*      SWITCH              USE
131          ;*      -----              -----
132          ;*      15              HALT ON ERROR
133          ;*      14              LOOP ON TEST
134          ;*      13              INHIBIT ERROR TYPEOUTS
135          ;*      12              STORAGE SCOPE CONNECTED
136          ;*      11              INHIBIT ITERATIONS
137          ;*      10              BELL ON ERROR
138          ;*      9              LOOP ON ERROR
139          ;*      8              LOOP ON TEST IN SWR<710>
140
141          .SBTTL  TRAP CATCHER
142
143          .OR
144          ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A "+2,MALT"
145          ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
146          ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
147          .OR
148          ;*174
149          ;*174
150          ;*176
151          .SBTTL  STARTING ADDRESS(ES)
152          JMP      @@REGIN  ;JUMP TO STARTING ADDRESS OF PROGRAM
153          JMP      @@BEGIN  ;JUMP TO RESTART ADDRESS
  
```

```

154
155
156          .SBTTL  ACT11 HOOKS
157
158          ;;*****
159          IHOOKS REQUIRED BY ACT11
160          SSVPC=          ISAVE PC
161          .#46
162          SENDAD          I1)SET LOC.46 TO ADDRESS OF SENDAD IN .VEOP
163          .#52
164          .WORD 0          I2)SET LOC.52 TO ZERO
165          .SSVPC          I RESTORE PC
166          .#1000
167
168          .SBTTL  APT PARAMETER BLOCK
169
170          ;;*****
171          ISET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
172          ;;*****
173          .SYN.          I)SAVE CURRENT LOCATION
174          .#24          I)SET POWER FAIL TO POINT TO START OF PROGRAM
175          200          I)FOR APT START UP
176          .#44          I)POINT TO APT INDIRECT ADDRESS PTR.
177          SAPTHDR I)POINT TO APT HEADER BLOCK
178          .#.SX          I)RESET LOCATION COUNTER
179
180          ;;*****
181          ISETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-POP11 DIAGNOSTIC
182          IINTERFACE SPEC.
183
184          SAPTHD:
185          SHIBTS: .WORD 0          I)TWO HIGH BITS OF 16 BIT MAILBOX ADDR.
186          SHBADR: .WORD SHAIL          I)ADDRESS OF APT MAILBOX (BITS 0-15)
187          STSYM: .WORD 30          I)RUN TIM OF LONGEST TEST
188          SPASTM: .WORD 60          I)RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
189          SUNITH: .WORD 120          I)ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
190          .WORD SETEND-SHAIL/2 I)LENGTH MAILBOX-ETABLE(WORDS)

```

190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235

001100
001100 000000
001102 000
001103 000
001104 000000
001106 000000
001110 000000
001112 000000
001114 000
001115 001
001116 000000
001120 000000
001122 000000
001124 000000
001126 000000
001130 000000
001132 000000
001134 000000
001136 177570
001140 177570
001142 177560
001144 177562
001146 177564
001150 177566
001152 000
001153 002
001154 012
001155 000
001156 000000
001160 000000
001162 000000
001164 000000
001166 000000
001170 177637
001174 077
001175 015
001176 000P12

000377

.SRTTL COMMON TAGS

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

.=1100

SCMTAG: .WORD 0
 STSTNMI .BYTE 0
 SERPLGI .BYTE 0
 SICNTI .WORD 0
 SLPADRI .WORD 0
 SLPERRI .WORD 0
 SERTYLI .WORD 0
 SITEMRI .BYTE 0
 SERMAXI .BYTE 1
 SERRPCI .WORD 0
 SGADARI .WORD 0
 SBDARI .WORD 0
 SGODATI .WORD 0
 SBDDATI .WORD 0
 .WORD 0
 .WORD 0
 .WORD 0
 SWRI .WORD DSWR
 DISPLAYI .WORD DDISP
 STKSI 177560
 STKBI 177562
 STPSI 177564
 STPBI 177566
 SNULLI .BYTE 0
 SFILLSI .BYTE 2
 SFILLCI .BYTE 12
 STPPLGI .BYTE 0
 SREGADI .WORD 0
 SREGOI .WORD 0
 SREGII .WORD 0
 STIMESI 0
 SESCAPE:0
 SBELLI .ASCIZ <207><377><377>
 SQUESI .ASCII /?/
 SCRLEI .ASCII <15>
 SLFI .ASCIZ <12>

START OF COMMON TAGS

CONTAINS THE TEST NUMBER
 CONTAINS ERROR FLAG
 CONTAINS SUBTEST ITERATION COUNT
 CONTAINS SCOPE LOOP ADDRESS
 CONTAINS SCOPE RETURN FOR ERRORS
 CONTAINS TOTAL ERRORS DETECTED
 CONTAINS ITEM CONTROL BYTE
 CONTAINS MAX. ERRORS PER TEST
 CONTAINS PC OF LAST ERROR INSTRUCTION
 CONTAINS ADDRESS OF 'GOOD' DATA
 CONTAINS ADDRESS OF 'BAD' DATA
 CONTAINS 'GOOD' DATA
 CONTAINS 'BAD' DATA
 RESERVED--NOT TO BE USED
 ADDRESS OF SWITCH REGISTER
 ADDRESS OF DISPLAY REGISTER
 TTY KBD STATUS
 TTY KBD BUFFER
 TTY PRINTER STATUS REG. ADDRESS
 TTY PRINTER BUFFER REG. ADDRESS
 CONTAINS NULL CHARACTER FOR FILLS
 CONTAINS # OF FILLER CHARACTERS REQUIRED
 INSERT FILL CHARS. AFTER A "LINE FEED"
 "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
 CONTAINS THE ADDRESS FROM
 WHICH (SREGP) WAS OBTAINED
 CONTAINS ((SREGAD)+0)
 CONTAINS ((SREGAD)+2)
 MAX. NUMBER OF ITERATIONS
 ESCAPE ON ERROR ADDRESS
 CODE FOR BELL
 QUESTION MARK
 CARRIAGE RETURN
 LINE FEED

236			;;						
237									
238			.SRTTL	APT MAILBOX=FTARLE					
239			;;						
240			.EVEN						
241			SMAIL:		;;APT MAILBOX				
242	001200	000000	SMGTY:	.WORD	AMSGTY	;;MESSAGE TYPE CODE			
243	001200	000000	SPATAL:	.WORD	APATAL	;;FATAL ERROR NUMBER			
244	001202	000000	STESTN:	.WORD	ATESTN	;;TEST NUMBER			
245	001204	000000	SPASS:	.WORD	APASS	;;PASS COUNT			
246	001206	000000	SDEVCT:	.WORD	ADEVCT	;;DEVICE COUNT			
247	001210	000000	SUNIT:	.WORD	AUNIT	;;I/O UNIT NUMBER			
248	001212	000000	MSGAD:	.WORD	AMSGAD	;;MESSAGE ADDRESS			
249	001214	000000	MSGLG:	.WORD	AMSGLG	;;MESSAGE LENGTH			
250	001216	000000	SETABLE:			;;APT ENVIRONMENT TABLE			
251	001220		SENV:	.BYTE	AENV	;;ENVIRONMENT BYTE			
252	001220	000	SENVH:	.BYTE	AENVH	;;ENVIRONMENT MODE BITS			
253	001221	000	SSWREG:	.WORD	ASWREG	;;APT SWITCH REGISTER			
254	001222	000000	SUSWR:	.WORD	AUSWR	;;USER SWITCHES			
255	001224	000000	SCPUOP:	.WORD	ACPUOP	;;CPU TYPE,OPTIONS			
256	001226	000000				RITS 15-11=CPU TYPE			
257						11/00=01,11/05=02,11/20=03,11/40=04,11/45=05			
258						11/70=06,PD0=07,0=1E			
259						RIT 10=REAL TIME CLOCK			
260						RIT 9=FLOATING POINT PROCESSOR			
261						RIT 8=MEMORY MANAGEMENT			
262									
263	001230	000	SHAMS1:	.BYTE	AHAMS1	;;HIGH ADDRESS,M.S. BYTE			
264	001231	000	SHTYP1:	.BYTE	AHTYP1	;;MEM. TYPE,BLK#1			
265						MEM. TYPE BYTE -- (HIGH BYTE)			
266						900 NSEC CORE=001			
267						300 NSEC BIPOLAR=002			
268						500 NSEC MOS=003			
269	001232	000000	SHADR1:	.WORD	AHADR1	;;HIGH ADDRESS,BLK#1			
270						MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE			
271	001234	000	SHAMS2:	.BYTE	AHAMS2	;;HIGH ADDRESS,M.S. BYTE			
272	001235	000	SHTYP2:	.BYTE	AHTYP2	;;MEM. TYPE,BLK#2			
273	001236	000000	SHADR2:	.WORD	AHADR2	;;MEM.LAST ADDRESS,BLK#2			
274	001240	000	SHAMS3:	.BYTE	AHAMS3	;;HIGH ADDRESS,M.S. BYTE			
275	001241	000	SHTYP3:	.BYTE	AHTYP3	;;MEM. TYPE,BLK#3			
276	001242	000000	SHADR3:	.WORD	AHADR3	;;MEM.LAST ADDRESS,BLK#3			
277	001244	000	SHAMS4:	.BYTE	AHAMS4	;;HIGH ADDRESS,M.S. BYTE			
278	001245	000	SHTYP4:	.BYTE	AHTYP4	;;MEM. TYPE,BLK#4			
279	001246	000000	SHADR4:	.WORD	AHADR4	;;MEM.LAST ADDRESS,BLK#4			
280	001250	340	SVECT1:	.BYTE	AVECT1	;;INTERRUPT VECTOR#1			
281	001251	000	SVECT2:	.BYTE	AVECT2	;;INTERRUPT VECTOR#2			
282	001252	200	SPRIOR:	.BYTE	APRIOR	;;BUS PRIORITY #1,#2			
283	001253	000		.BYTE	0	;;SPARE,NOT USED			
284				.EVEN					
285	001254	170400	SBASE:	.WORD	ABASE	;;BASE ADDRESS OF EQUIPMENT UNDER TEST			
286	001256	000000	SDEVH:	.WORD	ADEVH	;;DEVICE MAP			
287	001260	000000	SCDW1:	.WORD	ACDW1	;;CONTROLLER DESCRIPTION WORD#1			
288	001262	000000	SCDW2:	.WORD	ACDW2	;;CONTROLLER DESCRIPTION WORD#2			
289	001264	000000	SDDW1:	.WORD	ADWW1	;;DEVICE DESCRIPTOR WORD#1			
290	001266	000000	SDDW2:	.WORD	ADWW2	;;DEVICE DESCRIPTOR WORD#2			
291	001270	000000							

292	001272	000000	SDDW3:	.WORD	ADDW3	DEVICE	DESCRIPTOR	WORD03
293	001274	000000	SDDW4:	.WORD	ADDW4	DEVICE	DESCRIPTOR	WORD04
294	001276	000000	SDDW5:	.WORD	ADDW5	DEVICE	DESCRIPTOR	WORD05
295	001300	000000	SDDW6:	.WORD	ADDW6	DEVICE	DESCRIPTOR	WORD06
296	001302	000000	SDDW7:	.WORD	ADDW7	DEVICE	DESCRIPTOR	WORD07
297	001304	000000	SDDW8:	.WORD	ADDW8	DEVICE	DESCRIPTOR	WORD08
298	001306	000000	SDDW9:	.WORD	ADDW9	DEVICE	DESCRIPTOR	WORD09
299	001310	000000	SDDW10:	.WORD	ADDW10	DEVICE	DESCRIPTOR	WORD10
300	001312	000000	SDDW11:	.WORD	ADDW11	DEVICE	DESCRIPTOR	WORD11
301	001314	000000	SDDW12:	.WORD	ADDW12	DEVICE	DESCRIPTOR	WORD12
302	001316	000000	SDDW13:	.WORD	ADDW13	DEVICE	DESCRIPTOR	WORD13
303	001320	000000	SDDW14:	.WORD	ADDW14	DEVICE	DESCRIPTOR	WORD14
304	001322	000000	SDDW15:	.WORD	ADDW15	DEVICE	DESCRIPTOR	WORD15

305
306
307 001324

SETEND:

.SRTTL ERROR POINTER TABLE

! THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
 ! THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
 ! LOCATION SITE#B. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
 ! NOTE1: IF SITE#B IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
 ! NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

!	FM	POINTS TO THE ERROR MESSAGE
!	DM	POINTS TO THE DATA HEADER
!	DT	POINTS TO THE DATA
!	DF	POINTS TO THE DATA FORMAT

324 001324

SERRTA:

!ITEM 1

328	001324	013032	FM1	!DUAL REGISTER ADDRESSING DETECTED
329	001326	013075	DM1	!SERRPC RUFADR EXPECT READ
330	001330	013134	DT1	!SERRPC RUFADR SGO DAT SRO DAT
331	001332	000000	0	

333 001334 000000

NREXT: 0

334

335									
336	001336	170400			ADCS:	170400			IA TO D STATUS/CONTROL REGISTER
337	001340	170402			ADDR:	170402			IA TO D CONVERTED VALUE «READ»
338									
339	001342	170404			CSR:	170404			ICLOCK STATUS REGISTER
340	001344	170406			CSB:	170406			ICLOCK PRESET BUFFER
341									
342	001346	170410			VCSTAT:	170410			
343	001350	170412			VCYREG:	170412			
344	001352	170414			VCYREG:	170414			
345									
346	001354	170416			CSC:	170416			
347									
348	001356	005037	013226		REGINI:	CLR	TEMP		
349	001362	000403				BR	REG		
350	001364	012737	000001	013226	BEGIN1:	MOV	01,TEMP		
351	001372	000005			REG:	RESET			
352									;;CLEAR THE COMMON TAGS (SCHTAG) AREA
353	001374	012706	001100			MOV	0SCHTAG,R6		;;FIRST LOCATION TO BE CLEARED
354	001400	005026				CLR	(R6)+		;;CLEAR MEMORY LOCATION
355	001402	022706	001126			CMP	0SRDAT,R6		;;DONE?
356	001406	001374				BNE	.-6		;;LOOP BACK IF NO
357	001410	012706	001100			MOV	0STACK,SP		;;SETUP THE STACK POINTER
358									;;INITIALIZE A FEW VECTORS
359	001414	012737	013344	000020		MOV	0SSCOPE,00IOTVEC		;;IOT VECTOR FOR SCOPE ROUTINE
360	001422	012737	000340	000022		MOV	0340,00IOTVEC+2		;;LEVEL 7
361	001430	012737	013624	000030		MOV	0BERROR,00EMTVEC		;;EMT VECTOR FOR ERROR ROUTINE
362	001436	012737	000340	000032		MOV	0340,00EMTVEC+2		;;LEVEL 7
363	001444	012737	016134	000034		MOV	0STRAP,00TRAPVEC		;;TRAP VECTOR FOR TRAP CALLS
364	001452	012737	000340	000036		MOV	0340,00TRAPVEC+2		;;LEVEL 7
365	001460	012737	014142	000024		MOV	0SPWRDN,00PWRVEC		;;POWER FAILURE VECTOR
366	001466	012737	000340	000026		MOV	0340,00PWRVEC+2		;;LEVEL 7
367	001474	005037	001164			CLR	0TIMES		;;INITIALIZE NUMBER OF ITERATIONS
368	001500	005037	001166			CLR	0ESCAPE		;;CLEAR THE ESCAPE ON ERROR ADDRESS
369	001504	112737	000001	001115		MOVR	01,0SERMAX		;;ALLOW ONE ERROR PER TEST
370	001512	012737	001512	001106		MOV	0,,0LADR		;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
371	001520	012737	001520	001110		MOV	0,,0LPERR		;;SETUP THE ERROR LOOP ADDRESS
372									;;SIZE FOR A HARDWARE SWITCH REGISTER, IF NOT FOUND OR IT IS
373									;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
374	001526	013746	000004			MOV	0ERRVEC,-(SP)		;;SAVE ERROR VECTOR
375	001532	012737	001570	000004		MOV	0648,0ERRVEC		;;SET UP ERROR VECTOR
376	001540	012737	177570	001136		MOV	00SWR,0SWR		;;SETUP FOR A HARDWARE SWICH REGISTER
377	001546	012737	177570	001140		MOV	00DISP,0DISPLAY		;;AND A HARDWARE DISPLAY REGISTER
378	001554	022777	177777	177354		CMP	0-1,0SWR		;;TRY TO REFERENCE HARDWARE SWR
379	001562	001013				BNE	658		;;BRANCH IF NO TIMEOUT TRAP OCCURRED
380									;;AND THE HARDWARE SWR IS NOT = -1
381	001564	005737	000001			TST	001		;;FORCE A TRAP THROUGH ERRVEC
382	001570	012737	000176	001136	648:	MOV	0SWREG,0SWR		;;POINT TO SOFTWARE SWR
383	001576	012737	000174	001140		MOV	0DISPREG,0DISPLAY		;;POINT TO SOFTWARE DISPLAY REG
384	001604	012716	001612			MOV	0648,(SP)		;;REPLACE OLD PC WITH NEW
385	001610	000002				RTT			;;RESTORE PC AND PSW
386	001612	012637	000004		658:	MOV	(SP)+,0ERRVEC		;;RESTORE ERROR VECTOR
387									
388	001616				SARG1:				
389	001616	005037	001206			CLR	0PASS		;;CLEAR PASS COUNT
390	001622	132737	000200	001221		BITR	0APTSIZE,0ENVH		;;TEST USER SIZE UNDER APT

391	001630	001403			REQ	648	YES,USE NON-APT SWITCH
392	001632	012737	001222	001136	MOV	009wREG,SWR	NO,USE APT SWITCH REGISTER
393	001640						
394	001640	005037	177776	648:	CLR	00PS	

```
395
396
397
398
399
400
401          177772          PIRQ=177772
402          177776          PS=177776
403
404 001644 012737 000002 000006      MOV      RTI,006
405 001652 012737 000006 000004      MOV      06,004
406 001660 012700 000003              MOV      03,R0          ILOAD R0
407 001664 000261              SEC              ISET C BIT
408 001666 005737 177772          TST      00PIRQ          ITEST PIRQ
409 001672 005600              SBC      R0
410 001674 000261              SEC
411 001676 105737 177777          TSTR     0003+1
412 001702 005600              SBC      R0
413 001704 005037 177700          CLR      001770R
414 001710 005037 000006          CLR      006
415 001714 010037 010750          MOV      R0,CPTYPE
416 001720 005237 010750          INC      CPTYPE
417 001724 006300              ASI      R0
418 001726 016037 010736 010746      MOV      CPDLAY(R0),CPTIME      IGET CP DELAY TIME
419 001734 000137 002042          JMP      INIT1
420
421
422
423 001740 012702 000242          ILOAD TRAP CATCHER
424 001744 012701 000240          LDTRAP: MOV      0242,R2          ILOAD R2
425 001750 010221              MOV      0240,R1          ILOAD R1
426 001752 005021              SBI      MOV      R2,(R1)+      ILOAD .+2
427 001754 010102              CLR      (R1)+          ILOAD HALT
428 001756 005722              MOV      R1,R2          ILOAD R2
429 001760 020227 001402          TST      (R2)+          IBUMP R2
430 001764 001371              CMP      R2,01002          ITEST FOR LAST
431
432
433
434 001766 012700 001336              BNE      00          IBR UNTIL DONE
435 001772 013720 001254          MOV      0ADCS,R0          ILOAD POINTER
436 001776 022700 001356          1081: MOV      0BASE,(R0)+      ILOAD BASE ADDRESS
437 002002 001373              CMP      0BEGIN,R0          ITEST FOR DONE
438 002004 013737 001254 010752          BNE      00          IBR
439 002012 005237 010752          MOV      0BASE,ADCS1      ILOAD HIGH BYTE POINTER
440 002016 012700 001340          INC      ADCS1
441 002022 012701 000002          MOV      0ADDBR,R0          ILOAD 2ND ADDRESS
442 002026 060120              MOV      02,R1          ILOAD R1
443 002030 005721              1281: ADD      R1,(R0)+      IUPDATE REAL DEVICE WORD
444 002032 022701 000020          TST      (R1)+          IBUMP R1
445 002036 001373              CMP      020,R1          ITEST FOR DONE
446 002040 000207              BNE      00          IBR
446 002040 000207              RTS      PC          IEXIT
```


447											
448	002042	004737	001740		INIT11	JSR	PC,LOTRAP				
449	002046	005737	013226			TST	TEMP				ITEST IF START OR RESTART
450	002052	001011				BNF	INIT2				IRESTART
451	002054	005737	000042			TST	0002				ITEST IF MONITOR
452	002060	001402				BEG	18				IRR IF NOT
453	002062	000137	002254			JMP	VSUAL0				IRUN SCOPE IF UNDER MONITOR
454	002066	104400			181	TYPE					ICALL MESSAGE PRINTER VIA 'EMT'
455	002070	011706				TITLE					ITYPE PROGRAM HEADFR.
456	002072	104400			INITA1	TYPE					
457	002074	012070				MESS					IPRINT THE TEST CALL LETTERS.
458	002076	012737	002072	013152	INIT21	MOV	0INITA,AVECTR				ISET UP 'A' VECTOR ADDRESS.
459	002100	004737	001740			JSR	PC,LOTRAP				ILOAD TRAP CATCHER AND ADDRESSES
460	002110	104400				TYPE					
461	002112	012343				CNTRLC					IPRINT '.' TO INDICATE MONITOR READY
462	002114	005077	177216			CLR	0ANCS				
463	002120	005077	177216			CLR	0CSR				
464	002124	005077	177216			CLR	0VCSTAT				
465	002130	004737	010206			JSR	PC,XTTYIN				IWAIT FOR TTY ENTRY
466	002134	122737	000101	010436		CMPS	0'A,INBUF				ITEST FOR 'A'
467	002142	001002				BNE	18				INOT 'A'
468	002144	000137	007444			JMP	REPTST				IYES, RUN QUICK REGISTER TEST
469	002150	122737	000102	010436	181	CMPS	0'B,INBUF				ITEST FOR 'B'
470	002156	001002				BNE	28				INOT 'B'
471	002160	000137	002236			JMP	VISUAL				IYES, RUN 'SCOPE ADJUSTMENT TEST'
472	002164	122737	000103	010436	281	CMPS	0'C,INBUF				ITEST FOR 'C'
473	002172	001002				BNE	38				INOT 'C'
474	002174	000137	005100			JMP	CALBRT				IYES, RUN 'A TO D CALIBRATION' TEST
475	002200	122737	000104	010436	381	CMPS	0'D,INBUF				ITEST FOR 'D'
476	002206	001002				BNE	48				INOT 'D'
477	002210	000137	006244			JMP	REPTST				IYES, RUN 'A TO D REPEATABILITY' TEST
478	002214	122737	000105	010436	481	CMPS	0'E,INBUF				ITEST FOR 'E'
479	002222	001002				BNE	58				INOT 'E'
480	002224	000137	007024			JMP	RFCVRV				IYES RUN 'A TO D RECOVERY TEST
481	002230	104400			581	TYPE					ILLEGAL ENTRY
482	002232	012365				ONARK					ITYPE '9'
483	002234	000720				BR	INIT2				IWAIT AGAIN

```

484
485 002236 012737 002250 013152 VISUAL1 MOV 0VISAU1,AVECTR
486 002244 104400 TYPE
487 002246 012370 MEB6 IHEADER ABOUT SCOPE ADJ.
488 002250 104400 VISAU11 TYPE
489 002252 012633 MEB15 ITEXT ABOUT SWR
490 002254 000240 VSUAL01 NOP
491
492
493
494
495 002256 000004
496 002260 012737 000001 001164
497 002266 013700 001350
498 002272 013701 001352
499 002276 012737 000070 013210
500 002304 004737 005050
501 002310 004737 002370
502 002314 004737 004726
503 002320 000773
504
505
506
507
508 002322 000004
509 002324 012737 000001 001164
510 002332 013700 001352
511 002336 013701 001350
512 002342 012737 000070 013210
513 002350 004737 005050
514 002354 004737 002370
515 002360 004737 004726
516 002364 000773
517 002366 000427
518
519 002370 005077 176752
520 002374 013704 001346
521 002400 012703 001777
522 002404 012702 000001
523 002410 012711 001000
524 002414 012710 000000
525 002420 060210
526 002422 005214
527 002424 105714
528 002426 100376
529 002430 021003
530 002432 001372
531 002434 000207
532
  
```

```
533
534
535
536
537 002436 000004
538 002440 012737 000001 001164
539
540
541
542 002446 005037 013200
543 002452 005077 176670
544 002456 012737 000100 013210
545 002464 004737 005050
546 002470 013701 001350
547 002474 013702 001352
548 002500 013703 001346
549 002504 012704 000004
550 002510 013777 013200 176672 P31
551 002516 013777 013200 176626
552
553 002524 012700 000377
554 002530 012713 000004
555 002534 060411
556 002536 105713
557 002540 100376
558 002542 005300
559 002544 001373
560
561 002546 012713 000010
562 002552 012700 000377
563 002556 060412
564 002560 105713
565 002562 100376
566 002564 005300
567 002566 001373
568
569 002570 012713 000004
570 002574 012700 000377
571 002600 160411
572 002602 105713
573 002604 100376
574 002606 005300
575 002610 001373
576
577 002612 012713 000010
578 002616 012700 000377
579 002622 160412
580 002624 105713
581 002626 100376
582 002630 005300
583 002632 001373
584 002634 004737 004726
585 002640 000723
586
587
588
```

```

//*****
//TEST 3          PINCUSHION TEST (DISPLAY SQUARE)
//*****
TST3:  SCOPE
      MOV      01,0TIMES          //DO 1 ITERATION
//PLOT A SQUARE FROM LOWER LEFT TO LOWER RIGHT TO
//UPPER RIGHT TO UPPER LEFT TO LOWER LEFT.
//NON STORE DISPLAY
PIC3:  CLR      LOW
      CLR      OVCSTAT
      MOV      0100,TICKS
      JSR      PC,CMTIME
      MOV      VCYREG,R1
      MOV      VCYREG,R2
      MOV      VCSTAT,R3
      MOV      04,R4
P31:   MOV      LOW,OVCYREG
      MOV      LOW,OVCYREG
//DRAW BOTTOM LINE
      MOV      0377,R0
      MOV      04,(3)
P3A:   ADD      R4,(1)
      TSTB     (3)
      BPL      ,-2
      DEC      R0
      BNE      P3A
      JNO
//DRAW RIGHT LINE
      MOV      010,(3)
      MOV      0377,R0
P3B:   ADD      R4,(2)
      TSTB     (3)
      RPL      ,-2
      DEC      R0
      BNE      P3B
      JNO
//DRAW TOP LINE
      MOV      04,(3)
      MOV      0377,R0
P3C:   SUB      R4,(1)
      TSTB     (3)
      BPL      ,-2
      DEC      R0
      BNE      P3C
      JNO
//DRAW LEFT LINE
      MOV      010,(3)
      MOV      0377,R0
P3D:   SUB      R4,(2)
      TSTB     (3)
      RPL      ,-2
      DEC      R0
      BNE      P3D
      JNO
      JSR      PC,TIMER
      BR       PS
//*****
//TEST 4          PLOT AN X
//*****
```

580	002642	000000		TST41	SCOPE		
590	002644	012737	000001		MOV	01,STIMFS	1100 1 ITERATION
591	002652	012737	000000	PIC41	MOV	02,LOW	
592	002660	012737	001770		MOV	01770,HIGH	
593	002666	004077	176450		CLR	0VCSTAT	
594	002672	012737	000200		MOV	0200,TICKS	
595	002700	004737	005050		JSR	PC,CHTIME	1CHANGE TIMER FOR CP TYPE
596	002704	013701	001350	PIC401	MOV	VCXREG,R1	
597	002710	013702	001352		MOV	VCYREG,R2	
598	002714	013703	001346		MOV	VCSTAT,R3	
599	002720	012704	000000		MOV	00,R4	
600	002724	013712	013200	PAR	MOV	LOW,(2)	
601	002730	011211			MOV	(2),(1)	
602							
603							
604	002732	012713	000204		MOV	04,(3)	1PLOT LINE BEGINNING IN LOWER LEFT CORNER
605	002736	012700	000377		MOV	0377,R0	1ENABLE INTENSIFY ON LOADING X
606	002742	105713		PAR:	TSTR	(3)	
607	002744	100376			RPL	.-?	
608	002746	060412			ADD	R0,(2)	1+4 TO Y
609	002750	060411			ADD	R0,(1)	1+4 TO X
610	002752	005300			DEC	R0	
611	002754	001372			RNF	PAR	1NO
612	002756	105713			TSTR	(3)	
613	002760	100376			RPL	.-?	
614							
615	002762	013712	013202		MOV	HIGH,(2)	1PLOT LINE BEGINNING IN UPPER LEFT CORNER
616	002766	013711	013200		MOV	LOW,(1)	
617	002772	012700	000377		MOV	0377,R0	
618	002776	105713		PAR:	TSTR	(3)	
619	003000	100376			RPI	.-?	
620	003002	160412			SUB	R0,(2)	1-4 TO Y
621	003004	060411			ADD	R0,(1)	1+4 TO X
622	003006	005300			DEC	R0	
623	003010	001372			RNF	PAR	1NO
624	003012	004737	004726		JSR	PC,TIMER	
625	003016	000742			RR	R0	

```

626
627
628
629
630 003020 000000
631 003022 012737 000001 001160
632 003030 012737 000200 013210
633 003036 004737 005050
634
635 003042 005077 176300 151
636 003046 005077 176276
637 003052 005077 176274
638 003056 013703 001306
639 003062 032777 000200 176046
640 003070 001407
641 003072 013701 001352
642 003076 013700 001350
643 003102 012713 000010
644 003106 000406
645
646 003110 013700 001352 351
647 003114 013701 001350
648 003120 012713 000004
649 003124 004537 003154 451
650 003130 000000
651 003132 004537 003154
652 003136 001777
653
654 003140 005710
655 003142 001370
656 003144 004737 004726
657 003150 000734
658 003152 000413
659
660 003154 012702 000200
661 003160 011511
662 003162 105713
663 003164 100376
664 003166 062710 000002
665 003172 005302
666 003174 001371
667 003176 005725
668 003200 000205

//*****
//TEST 5 SCOPE SETTLING TIME TEST
//*****
TST5: SCOPE
MOV 01,STIMES //NO 1 ITERATION
PIC9: MOV 0200,TICKS //LOAD TIMER
JSR PC,CHTIME //CHANGE TIMER FOR CP TYPE

151 CLR 0VCSTAT //CLEAR STATUS
CLR 0VCXREG //CLEAR X AXIS
CLR 0VCYREG //CLEAR Y AXIS
MOV VCSTAT,R3 //LOAD STATUS ADDRESS
RTT 00177,0SWR //TEST 0SWR
REQ 38 //RQ IF CLEARED
MOV VCYREG,R1 //LOAD R1
MOV VCXREG,R0 //LOAD R0
MOV 010,(R3) //LOAD INTENS ON Y
RR 48

351 MOV VCYREG,R0 //LOAD R0
MOV VCXREG,R1 //LOAD R1
MOV 04,(R3) //LOAD INTENS ON X
451 JSR R5,LODPNT //LOAD A LINE
R //
JSR R5,LODPNT //LOAD A LINE
1777

TST (R0) //END
RNE 48 //RQ IF NOT
JSR PC,TIMER //TEST TIME
RR 19
RR TST6 //

LODPNT: MOV 0200,R2
151 MOV (R5),(R1) //LOAD AXIS
281 TST0 (R5) //DONE
BPL 28 //WAIT
ADD 02,(R0) //UPDATE AXIS
DEC R2 //DONE
RNE 19 //RQ IF NOT
TST (R5)+ //UPDATE
RTS R5 //EXIT

```

```
669
670
671
672
673
674 003202 000004
675 003204 012737 000001 001164
676 003212 012737 000100 013210
677 003220 004737 005050
678 003224 012737 000340 003546
679 003232 012737 001400 003544
680 003240 012702 003554
681 003244 005077 176076
682 003250 004737 003346
683 003254 012737 000340 003546
684 003262 012737 001000 003544
685 003270 012702 003655
686 003274 005077 176046
687 003300 004737 003346
688 003304 012737 000340 003546
689 003312 012737 000400 003544
690 003320 012702 003756
691 003324 005077 176016
692 003330 004737 003346
693 003334 004737 004726
694 003340 000731
695 003342 000137 004070
696 003346 012737 177763 003550
697 003354 013705 001346
698 003360 000240
699 003362 000240
700 003364 000240
701 003366 004737 003402
702 003372 005237 003550
703 003376 001373
704 003400 000207
705

;*****
;TEST 6          PLOT CHARACTER SET
;*****
TST6:  SCOPE
      MOV      01,STIMES          ;100 1 ITERATION
      MOV      0100,TICKS
      JSR      PC,CHTIME         ;CHANGE TIMER FOR CP TYPE
      MOV      0340,XPOS
      MOV      01400,YPOS
      MOV      0A,R2             ;LOAD STARTING CHARACTER
      CLR      0VCSTAT
      JSR      PC,PIC6A
      MOV      0340,XPOS         ;LOAD X POS
      MOV      01000,YPOS       ;LOAD Y POS
      MOV      0A,R2             ;LOAD STARTING CHARACTER
      CLR      0VCSTAT
      JSR      PC,PIC6B         ;DISPLAY CHARACTERS
      MOV      0340,XPOS         ;LOAD X POS
      MOV      0000,YPOS        ;LOAD Y POS
      MOV      0A,R2             ;LOAD STARTING CHARACTER
      CLR      0VCSTAT
      JSR      PC,PIC6B         ;DISPLAY CHARACTERS
      JSR      PC,TIMER
      RR      PIC6A
      JMP      PIC7
      MOV      0-13.,CHRCOL     ;CHARACTERS PER ROW
      MOV      0VCSTAT,R5
      NOP
      NOP
      NOP
      JSR      PC,CHAR
      INC     CHRCOL
      BNE    GEN1
      RTS     PC
      GEN1:

```

```
706
707
708 003402 013737 003500 003552 1PLOT CHARACTER
709 003410 042715 000010 CHAR1: MOV YPOS,YPT
710 003414 013777 003500 175726 MOV #16,(5)
711 003422 013777 003500 175722 MOV XPOS,OVXREG
712 003430 105777 175712 CHAR0: TSTR YPOS,OVYREG
713 003430 100375 RPL OVSTAT
714 003436 053715 003502 RPL CHAR0
715 003442 012700 000010 RPL MODE,(5)
716 003446 012700 177773 MOV #10,R0
717 003452 012701 177771 CHAR1: MOV #5,R0
718 003456 112203 MOV #7,R1
719 003460 106103 CHAR2: MOVR (2)+,R3
720 003462 100006 RPL R3
721 003464 013777 003500 175656 MOV CHAR3
722 003472 013777 003500 175652 MOV XPOS,OVXREG
723 003500 105777 175602 CHAR3: TSTR YPOS,OVYREG
724 003500 100375 RPL OVSTAT
725 003506 060437 003500 RPL CHAR3
726 003512 005201 ADD R0,YPOS
727 003514 001361 INC R1
728 003516 013737 003552 003500 RNF CHAR2
729 003524 060437 003506 MOV #1 TO ROW
730 003530 005200 INC R2
731 003532 001347 RNF CHAR2
732 003534 060437 003506 MOV #FINISH ROW
733 003540 000207 RTS PC
734
735 003542 000010 MODE: 10
736 003544 000000 YPOS: 0
737 003546 000000 XPOS: 0
738 003550 000000 CHRCOL: 0
739 003552 000000 YPT: 0
740
```

```

IENABLE INTENSIFY ON LOADING Y
IINITIALIZE COLUMN COUNT
IINITIALIZE ROW COUNT
IPUT CHARACTER POINTS IN R3
INO
I+1 TO ROW
IFINISH ROW
IREINITIALIZE ROW FOR NEXT COLUMN
I+1 TO COLUMN COUNT
IEXIT
ICONTAINS Y POSITION AT ANY GIVEN TIME
ICONTAINS X POSITION AT ANY GIVEN TIME
```

DATA BYTES FOR VISUAL CHARACTERS

741						
742						
743						
744	003554	176	021	021	A	.BYTE 176,21,21,21,176
745	003557	021	176			
746	003561	177	111	111	B	.BYTE 177,111,111,111,66
747	003564	111	066			
748	003566	076	101	101	C	.BYTE 76,101,101,101,42
749	003571	101	042			
750	003573	177	101	101	D	.BYTE 177,101,101,101,76
751	003576	101	076			
752	003600	177	111	111	E	.BYTE 177,111,111,111,101
753	003603	111	101			
754	003605	177	011	011	F	.BYTE 177,11,11,11,1
755	003610	011	001			
756	003612	076	101	121	G	.BYTE 76,101,121,121,62
757	003615	121	062			
758	003617	177	010	010	H	.BYTE 177,10,10,10,177
759	003622	010	177			
760	003624	000	101	177	I	.BYTE 0,101,177,101,0
761	003627	101	000			
762	003631	060	100	100	J	.BYTE 60,100,100,100,77
763	003634	100	077			
764	003636	177	010	024	K	.BYTE 177,10,24,42,101
765	003641	042	101			
766	003643	177	100	100	L	.BYTE 177,100,100,100,100
767	003646	100	100			
768	003650	177	004	010	M	.BYTE 177,4,10,4,177
769	003653	004	177			
770	003655	177	004	010	N	.BYTE 177,4,10,20,177
771	003660	020	177			
772	003662	076	101	101	O	.BYTE 76,101,101,101,76
773	003665	101	076			
774	003667	177	011	011	P	.BYTE 177,11,11,11,6
775	003672	011	006			
776	003674	076	101	121	Q	.BYTE 76,101,121,141,176
777	003677	141	176			
778	003701	177	011	031	R	.BYTE 177,11,31,51,106
779	003704	051	106			
780	003706	046	111	111	S	.BYTE 46,111,111,111,62
781	003711	111	062			
782	003713	001	001	177	T	.BYTE 1,1,177,1,1
783	003716	001	001			
784	003720	077	100	100	U	.BYTE 77,100,100,100,77
785	003723	100	077			
786	003725	037	040	100	V	.BYTE 37,40,100,40,37
787	003730	040	037			
788	003732	177	020	010	W	.BYTE 177,20,10,20,177
789	003735	020	177			
790	003737	143	024	010	X	.BYTE 143,24,10,24,143
791	003742	024	143			
792	003744	003	004	170	Y	.BYTE 3,4,170,4,3
793	003747	004	003			
794	003751	141	121	111	Z	.BYTE 141,121,111,105,103
795	003754	105	103			
796	003756	076	121	111	NA	.BYTE 76,121,111,105,76

797	003761	105	076			
798	003763	000	102	177	N11	.BYTE 0,102,177,100,0
799	003766	100	000			
800	003770	102	121	111	N21	.BYTE 102,121,111,105,102
801	003773	105	102			
802	003775	042	101	111	N31	.BYTE 42,101,111,111,06
803	004000	111	066			
804	004002	030	020	022	N41	.BYTE 30,20,22,177,20
805	004005	177	020			
806	004007	047	105	105	N51	.BYTE 47,105,105,105,71
807	004012	105	071			
808	004014	076	111	111	N61	.BYTE 76,111,111,111,02
809	004017	111	062			
810	004021	101	041	021	N71	.BYTE 101,01,21,11,7
811	004024	011	007			
812	004026	066	111	111	N81	.BYTE 66,111,111,111,06
813	004031	111	066			
814	004033	046	111	111	N91	.BYTE 46,111,111,111,76
815	004036	111	076			
816	004040	000	000	000	SPACEA1	.BYTE 0,0,0,0,0
817	004043	000	000			
818	004045	000	000	000		.BYTE 0,0,0,0,0
819	004050	000	000			
820	004052	000	000	000		.BYTE 0,0,0,0,0
821	004055	000	000			
822						
823	004060					.EVEN

```

824
825
826
827 004260 000000
828 004262 012737 000001 001164
829 004270 012737 000200 013210
830 004276 004737 005050
831 004102 013705 001306
832 004106 012777 000000 175232
833 004114 012737 000400 003546
834 004122 012737 001200 003544
835 004130 012737 000011 004262
836 004136 012737 004266 004264
837 004144 017702 000114
838 004150 004737 003402
839 004154 062737 000002 004264
840 004162 005337 004262
841 004166 001366
842 004170 012777 001000 175150
843 004176 012737 000400 003546
844 004204 012737 000600 003544
845 004212 012737 000011 004262
846 004220 012737 004310 004264
847 004226 017702 000032
848 004232 004737 003402
849 004236 062737 000002 004264
850 004244 005337 004262
851 004250 001366
852 004252 004737 004726
853 004256 000713
854 004260 000424
855 004262 000000
856 004264 000000
857 004266 003566
858 004270 003617
859 004272 003554
860 004274 003655
861 004276 003655
862 004300 003600
863 004302 003643
864 004304 004040
865 004306 003763
866 004310 003566
867 004312 003617
868 004314 003554
869 004316 003655
870 004320 003655
871 004322 003600
872 004324 003643
873 004326 004040
874 004330 003770
875

```

```

//*****
//TEST 7 CHANNEL 1 CHANNEL 2
//*****
TST7: SCOPE
      MOV 01,STIMES //DO 1 ITERATION
      MOV 0200,TICKS //SET UP A TIMER
      JSR PC,CHTIME //CHANGE TIMER FOR CP TYPE
      MOV VCSTAT,R5 //SET UP R5 FOR STATUS REGISTER POINTER
      MOV 00,OVSTAT //SET UP SCOPE CONTROL
      MOV 0400,XPOS //LOAD X POSITION
      MOV 01200,YPOS //LOAD Y POSITION
      MOV 00,,P7CNT //SAVE THE NUMBER OF CHARACTERS
      MOV 0CH01,P7PNT //SAVE CHANNEL 1 POINTER
      MOV 0P7PNT,R2 //MOVE MESSAGE POINTER INTO R2 FOR DISPLAY ROUTINE
      JSR PC,CHAR //DISPLAY A CHARACTER
      ADD 02,P7PNT //ADD 2 TO THE MESSAGE POINTER
      DEC P7CNT //DECREMENT CHARACTER COUNT
      RNE PIC7A //NOT FINISHED WITH ALL CHARACTERS
      MOV 01000,OVSTAT
      MOV 0400,XPOS //SET UP X POS FOR CHANNEL 2
      MOV 0600,YPOS //SET UP Y
      MOV 00,,P7CNT //SET UP CHARACTER COUNT
      MOV 0CH02,P7PNT //SET UP CHANNEL 2 POINTER
      MOV 0P7PNT,R2 //SET UP
      JSR PC,CHAR //DISPLAY A CHARACTER
      ADD 02,P7PNT //ADD 2 TO THE POINTER
      DEC P7CNT //DECREMENT COUNT
      RNE PIC7B //NOT FINISHED
      JSR PC,TIMER //CHECK THE RUNTIME OF THIS ROUTINE
      RR PIC7AA //NOT FINISHED
      RR TST10 //JOB TO NEXT TEST
      P7CNT: 0
      P7PNT: 0
      CH01: C
           M
           A
           N
           F
           L
           SPACEA
      CH02: C
           M
           A
           N
           F
           L
           SPACEA
           N2

```

876
877
878
879 004332 000000
880 004334 012737 000001 001160
881 004342 012737 000002 013210
882 004350 032777 000000 174560
883 004356 001002
884 004360 004737 004536
885 004364 004737 004626
886 004370 004737 004726
887 004374 000765
888 004376 032777 000000 174532
889 004404 001002
890 004406 004737 004536
891 004412 000005
892
893
894
895
896
897
898
899
900
901
902 004414
903 004414 000000
904 004416 005037 001102
905 004422 005037 001160
906 004426 005237 001206
907 004432 002737 100000 001206
908 004440 005327
909 004442 000001
910 004444 003022
911 004446 012737
912 004450 000001
913 004452 004442
914 004454 104400 004521
915 004460 013746 001206
916 004464 104400
917 004466 104400 004516
918 004472 013700 000002
919 004476 001005
920 004500 000005
921 004502 004710
922 004504 000240
923 004506 000240
924 004510 000240
925 004512
926 004512 000137
927 004514 002254
928 004516 377 377 000
929 004521 015 002412 002116
930 004526 050000 051501 020123
931 004534 000005

```

;*****
;TEST IF PHOSPHOR TEST
;*****
TST10: SCOPF
        MOV     01,STIMES          ;100 I ITERATION
PIC12:  MOV     02,TICKS
PIC12A: BIT     0015,0SWR          ;TEST IF STORAGE SCOPE
        BEQ     10                ;BR IF NOT
        JSR     PC,CLRVC          ;ERASE THE SCREEN
        JSR     PC,LOADVF         ;LOAD THE SCREEN
        JSR     PC,TIMER         ;CHECK THE TIME
        RR
        BIT     0015,0SWR          ;TEST IF STORAGE SCOPE
        BEQ     20                ;BR IF NOT
        JSR     PC,CLRVC
20:
20:     REPT

.SBTL  END OF PASS ROUTINE

;*****
;INCREMENT THE PASS NUMBER (SPASS)
;INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
;TYPE "END PASS 0XXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
;IF THERES A MONITOR GO TO IT
;IF THERE ISN'T JUMP TO VSUAL0

SEOP:
        CLR     STBYM             ;ZERO THE TEST NUMBER
        CLR     STIMES           ;ZERO THE NUMBER OF ITERATIONS
        INC     SPASS            ;INCREMENT THE PASS NUMBER
        BIC     010000,SPASS     ;DON'T ALLOW A NEG. NUMBER
        DEC     (PC)+           ;LOOP?
SEOPCT: .WORD 1
        BGT     S00AGN          ;YES
        MOV     (PC)+,0(PC)+    ;RESTORE COUNTER
SENDCT: .WORD 1
        SEOPCT
        TYPE    ,SENDMG         ;TYPE "END PASS 0"
        MOV     SPASS,-(SP)     ;SAVE SPASS FOR TYPEOUT
        TYPE    ,SENULL        ;GO TYPE--DECIMAL ASCII WITH SIGN
        TYPE    ,SENULL        ;TYPE A NULL CHARACTER
        MOV     0002,00        ;GET MONITOR ADDRESS
        BEQ     S00AGN         ;BRANCH IF NO MONITOR
        RESET
        JSR     PC,(00)        ;CLEAR THE WORLD
        NOP
        NOP
        NOP
        JMP     0(PC)+         ;GO TO MONITOR
        .WORD  VSUAL0         ;SAVE ROOM
        .BYTE  -1,-1,0        ;FOR
        .ASCII <15><12>/END PASS 0/ ;ACT11
        .WORD  0(PC)+         ;RETURN
SETNAD: .WORD  VSUAL0
SENULL: .BYTE  -1,-1,0
SENDMG: .ASCII <15><12>/END PASS 0/

```

932										
933										
934	004536	012777	002000	174602	CLRVCI	MOV	00110,0VCSTAT			IFRASE THE SCREEN
935	004544	052777	010000	174574		BIS	00112,0VCSTAT			
936	004552	000240				NOP				
937	004554	012700	000020			MOV	020,R0			IFSET UP DELAY
938	004560	005001				CLR	R1			
939	004562	105777	174560		CLRVCAI	TSTB	0VCSTAT			IFTEST FOR READY
940	004566	100416				AND	CLRVCB			IFRANCH IF SET
941	004570	005301				DEF	R1			IFPLAV
942	004572	001373				AND	CLRVCA			
943	004574	005300				DEF	R0			IFDELAY
944	004576	001371				AND	CLRVCA			
945	004600	037727	174332	010000		BIT	0SWR,0SW12			IFTEST INHIBIT PRINTOUT
946	004606	001002				AND	10			
947	004610	104400				TYPE				
948	004612	012023				HEB3				
949	004614	005777	174316		101	TST	0SWR			IFTEST 0SWR
950	004620	100001				RPL	CLRVCR			
951	004622	000000				HALT				IFERASE RETURN FAILED TO SET READY
952										
953	004624	000207			CLRVCI	RTS	PC			
954										
955	004626	005077	174514		LOADVCI	CLR	0VCSTAT			IFCLEAR STATUS
956	004632	012737	001777	013230		MOV	01777,TEMP1			
957	004640	013700	001346			MOV	VCSTAT,R0			
958	004644	013701	001350			MOV	VCRREG,R1			
959	004650	013702	001352			MOV	VCRREG,R2			
960	004654	012710	002000			MOV	00110,(R)			IFSET STORE MODE
961	004660	013712	013230			MOV	TEMP1,(2)			
962	004664	012711	001777		LOADVCAI	MOV	01777,(1)			
963	004670	000402				RR	LOADVCC			
964	004672	162711	000004		LOADVCI	SUB	00,(1)			
965	004676	005210			LOADVCCI	INC	(R)			
966	004700	000240				NOP				
967	004702	105710				TSTB	(R)			
968	004704	100376				BPL	.-2			
969	004706	022711	000003			CMR	03,(1)			
970	004712	001367				AND	LOADVCR			
971	004714	104405				CKSWR				IFTEST FOR "CTRL G"
972	004716	162712	000001			SUB	01,(2)			
973	004722	001360				AND	LOADVCA			
974	004724	000207				RTS	PC			
975										

```

976
977
978          TIMER ROUTINE
979          I ENTER VIA JSR PC,TIMER
980
981 004726 104405          TIMER: CKSWR
982 004730 017737 174202 013206          MOV      @SWR,TIMSV
983 004736 004737 011672          JSR      PC,TSTFLG
984 004742 032737 000400 013206  TIMER:1 BIT      @RTA,TIMSV
985 004750 001006          BNF      TIMER2          IRT 0 SET 7
986 004752 005337 013210          DEC      TICKS          INO, DECREMENT TICKS
987 004756 001002          ANP      TIMER1
988 004760 062716 000002          ADD      @2,(6)          IADD 2 TO STACK POINTER
989 004764 000207          TIMER:1 RTS      PC          IRETURN
990
991          I SWR 0=1 SELECT TEST TO LOCK ON
992          I SWR 2=00 TEST NUMBER
993
994 004766 042737 177770 013206  TIMER:2 RLC      @177770,TIMSV
995 004774 006337 013206          ASL      TIMSV
996 005000 062737 005030 013206          ADD      @ROUTPT,TIMSV
997 005006 017737 006174 013206          MOV      @TIMSV,TIMSV
998 005014 022600          CMP      (SP)+,R0
999 005016 000240          NJP
1000 005020 000240          NOP
1001 005022 000240          NOP
1002 005024 000177 006156          TIMER:4 JMP      @TIMSV
1003
1004 005030 002266          ROUTPT: PIC0          IDISPLAY A HORIZONTAL LINE
1005 005032 002332          PIC1          IDISPLAY A VERTICAL LINE
1006 005034 002446          PIC3          IDISPLAY A SQUARE
1007 005036 002652          PIC4          IDISPALY A "X"
1008 005040 003030          PIC5          IDISPLAY SETTLING TIME
1009 005042 003212          PIC6          IDISPLAY CHARACTER SET
1010 005044 004070          PIC7          IDISPLAY CHANNEL TPST
1011 005046 004342          PIC:12          IDISPLAY ERASE AND PHOSPHOR TEST
1012
1013 005050 013737 010750 013236  CHTIME:1 MOV      CPTYPE,@RLEVI
1014 005056 005337 013236          CHTMA:1 DEC      @RLEVI
1015 005062 001403          BEQ      CHTM0
1016 005064 006337 013210          ASL      TICKS
1017 005070 000772          RR      CHTMA
1018 005072 000207          CHTM:1 RTS      PC

```

```
1019                                     ;|*****|
1020                                     ;|TEST 11      CALIBRATION ROUTINE|
1021                                     ;|*****|
1022 005274 000000 TST111 SCOPE
1023 005076 012737 000001 001160      MOV      01,STIMES      ;|DO 1 ITERATION|
1024                                     ;|*****|
1025
1026 ;ROUTINE REQUESTS THE TYPE OF 'SYNC' TO BE USED ('I' INTERNAL OR 'E' EXTERNAL
1027 ;OR 'C' OR CLOCK)
1028 ;THE PROGRAM THEN TAKES CONTINUOUS CONVERSIONS USING DATA SW'S 3-8
1029 ;TO SELECT THE CH., AND SW5 TO SELECT UNIPOLAR/BIPOLAR
1030 ;USE SW '6' TO FREEZE ON SINGLE 512 POINT BURST.
1031 ;USE SW '7' TO DISPLAY DATA ON Y AXIS TO CHECK FOR A/D INTERACTION
1032 ;USE SW '10' TO PRINT THE CONVERSION VALUE.
1033
1034 005104 012737 005116 013152 CALBT1: MOV      0CALBT1,AVECTR  ;SET UP 'A' RESTART ADDRESS
1035 005112 104400                                     TYPE
1036 005114 012420                                     NEXT
1037 005116 104400 CALBT1: TYPE
1038 005120 012513                                     MES10
1039 005122 004737 010206 JSR      PC,XTTYIN      ;WAIT FOR INPUT.
1040 005126 013737 010436 013154      MOV      INRUP,PROC    ;SAVE IT IN TEMP STORAGE
1041 005134 104400                                     TYPE
1042 005136 012356 ACRLF
1043 005140 005037 013222      CLR      USECLK      ;CLEAR CLOCK FLAG
1044 005144 012737 000001 013160      MOV      01,COUNT    ;SET UP FOR '1' CONVERSION
1045 005152 117737 173760 013147 CALBT2: MOVR   08WR,ADWRD2+1 ;GET CH. FROM THE SW REG.
1046 005160 042737 150377 013146      RIC      0150377,ADWRD2 ;CLR UNWANTED BITS, A/D WORD COMPLETE
1047 005166 005037 006224      CLR      ADWRD1
1048 005172 017737 173740 013164      MOV      08WR,KSTOR2 ;SAVE ORIGINAL SWITCH SETTING.
1049 004200 022737 000105 013154      CMP      0105,PROC    ;TEST SYNC SELECT
1050 005206 001004      BNF      10
1051 005210 052737 000120 006224      RIS      0120,ADWRD1 ;BRANCH IF NOT 'E'
1052 005216 000416      RR      CALB2A      ;OTHERWISE ADD 'PXT' SYNC BIT TO A/D.
1053 005220 122737 000103 013154 101  CMPR    0103,PROC
1054 005226 001007      RNE      20
1055 005230 052737 000140 006224      BIS      0140,ADWRD1 ;BRANCH IF NOT 'C'
1056 005236 012737 177777 013222      MOV      0-1,USECLK  ;OTHERWISE ADD 'CLOCK' SYNC TO A/D
1057 005244 000403      RR      CALB2A
1058 005246 052737 000101 006224 201  BIS    0101,ADWRD1
1059 005254 012777 006154 173766 CALB2A: MOV    0CNST51,0SVECT1
1060 005262 000240      NOP
```

1061	005264	012705	016200		MOV	000RUPP,R5	IRFSET POINTER	
1062	005270	000240		118:	NOP			
1063	005272	000240			NOP			
1064	005274	000240			NOP			
1065	005276	013777	013100	174032	MOV	00WRD2,0A0CS	ILOAD A/D STATUS	
1066	005304	000240			NOP			
1067	005306	005077	174034		CLR	0VCSTAT	ICLEAR STAT	
1068	005312	005077	174032		CLR	0VCYREG		
1069	005316	005077	174030		CLR	0VCYREG		
1070	005322	032777	000200	173606	BIT	0RIT7,0SWR	ITEST BIT	
1071	005330	001410			REQ	20	JRR IF DOWN	
1072	005332	013703	001350		MOV	VCYREG,R3	ILOAD X ADDRESS	
1073	005336	013704	001352		MOV	VCYREG,R4	ILOAD Y ADDRESS	
1074	005342	012777	000004	173776	MOV	00,0VCSTAT	IINTEN. ON X LOAD	
1075	005350	000407			BR	30	I	
1076	005352	013703	001352	28:	MOV	VCYREG,R3	ILOAD Y ADDRESS	
1077	005356	013704	001350		MOV	VCYREG,R4	ILOAD X ADDRESS	
1078	005362	012777	000010	173756	MOV	010,0VCSTAT	IINTEN. ON Y LOAD	
1079	005370	017737	173752	006226	38:	MOV	0VCSTAT,AXIS1	ISAVE STATUS
1080	005376	010337	006230		MOV	R3,AXIS2	ISAVE R3	
1081	005402	010437	006232		MOV	R4,AXIS3	ISAVE R4	
1082	005406	012737	000004	006054	MOV	04,1010	ILOAD COUNT	
1083	005414	012700	000160	108:	MOV	0160,R0	ILOAD A CONVERSION COUNT	
1084	005420	012701	000000		MOV	00,R1	ICLEAR OFFSET	
1085	005424	005037	006222		CLR	MARKER	ISET MARKER SAMPLE	
1086	005430	004737	006112		JSR	PC,CNVTSV	ICONVERT AND SAVE	
1087								
1088	005434	012700	000020		MOV	020,R0	ILOAD MARKER COUNT	
1089	005440	012701	000011		MOV	011,R1	ILOAD OFFSET	
1090	005444	005237	006222		INC	MARKER	ISET MARKER SAMPLE	
1091	005450	004737	006112		JSR	PC,CNVTSV	ICONVERT AND SAVE	
1092								
1093	005454	005337	006054		DEF	1010	I00NE ?	
1094	005460	001355			BNE	100	JRR IF NOT	
1095								
1096								
1097	005462	012737	000077	013230	158:	MOV	077,TEMP1	ILOAD CONVERSION COUNT
1098	005470	012737	000006	011330	MOV	06,CMPENT	ILOAD SHIFT COUNT	
1099	005476	004737	011166		JSR	PC,CMPTR	I AVERAGE	
1100	005502	013700	013266		MOV	AVERAGE,R0	I GET A VALUE	
1101	005506	006200			ASR	R0		
1102	005510	006200			ASR	R0		
1103	005512	005200			INC	R0		
1104	005514	006200			ASR	R0		
1105	005516	010037	006056		MOV	R0,1020	ISAVE FOR TYPEOUT	
1106	005522	012737	000004	006052	MOV	04,1000	ILOAD A COUNT	
1107	005530	012702	006070		MOV	00GT4,R2	ILOAD A POINTER	
1108	005534	000403			BR	160		
1109	005536	006200		178:	ASR	R0	IMOVE RIGHT	
1110	005540	006200			ASR	R0		
1111	005542	006200			ASR	R0		
1112	005544	010001		168:	MOV	R0,R1	I GET VALUE	
1113	005546	042701	177770		RIC	0177770,R1	IMASK	
1114	005552	006301			ASL	R1	IX2	
1115	005554	016142	006072		MOV	AND(R1),-(R2)	ILOAD ADDRESS	
1116	005560	005337	006052		DEC	1000	IFINISHED ?	

Line	Address	Label	Value	Code	Comment
1117	005564	001364		RNF	178
1118					
1119	005566	005077	173554		
1120	005572	013703	001346		
1121	005576	012737	000000	003546	
1122	005604	012737	000200	003544	
1123	005612	004737	006010		
1124	005616	012737	001500	003546	
1125	005624	012737	001500	003544	
1126	005632	004737	006010		
1127	005636	032777	000100	173272	
1128	005644	001432			
1129					
1130	005646	012703	016200		
1131	005652	012700	001000		
1132	005656	005077	173464		
1133	005662	013703	006230		
1134	005666	013704	006232		
1135	005672	005013			
1136	005674	005014			
1137	005676	013777	006226	173442	
1138	005704	012513		138:	
1139	005706	105777	173434	128:	
1140	005712	100373			
1141	005714	062714	000002		
1142	005720	005300			
1143	005722	001370			
1144	005724	004737	011672		
1145	005730	000716			
1146	005732	012737	000001	013234	228:
1147	005740	032777	002000	173170	
1148	005746	001406			
1149	005750	013746	006056		
1150	005754	104402			
1151	005756	004	001		
1152	005760	104400			
1153	005762	012356			
1154	005764	004737	011672	218:	
1155	005770	023777	013164	173140	
1156	005776	001402			
1157	006000	000137	005152		
1158	006004	000137	005254	238:	

Code	Instruction	Comment
148:	CLR	OVSTAT
	MOV	VCSTAT,R5
	MOV	00,XPOS
	MOV	0200,YPOS
	JSR	PC,303
	MOV	01500,XPOS
	MOV	01500,YPOS
	JSR	PC,303
	BIT	00176,0SWR
	REQ	228
	MOV	000000,R5
	MOV	01000,R0
	CLR	OVSTAT
	MOV	AX132,R3
	MOV	AX133,R4
	CLR	(R3)
	CLR	(R4)
	MOV	AX131,OVSTAT
	MOV	(R5)+,(R3)
	TSTB	OVSTAT
	RPL	128
	ADD	00,(R4)
	DEC	R0
	BNE	138
	JSR	PC,TSTFLG
	RR	148
	MOV	01,TEMP3
	BIT	00W10,0SWR
	BEG	218
	MOV	1020,-(SP)
	TYPOS	
	.BYTE	4,1
	TYPE	
	ACRLF	
	JSR	PC,TSTFLG
	CMP	KSTOR2,0SWR
	REQ	238
	JMP	CAL072
	JMP	CAL02A

Code	Instruction	Comment
		IRR IF NOT
		ILOAD R5
		ILOAD X POSITION
		ILOAD Y POSITION
		IDISPLAY DIGITS
		ILOAD X POS
		ILOAD Y POS
		ITEST 0SWR
		IRR IF CONTINUOUS SAMPLES
		ILOAD BUFFER POINTER
		ILOAD COUNT
		ICLEAR STATUS
		IRESTORE R3
		IRESTORE R4
		IRESTORE STATUS
		ILOAD POINT
		WAIT FOR SCOPE
		IUPDATE AXIS
		IDONE ?
		IRR IF NOT
		ITEST KEYBOARD FLAG
		ISFTUP TO PRINT '1' VALUE
		ITEST FOR KEYBOARD INTERRUPT
		ITEST IF SWITCH REGISTER HAS CHANGED
		IBRANCH AND TAKE NEXT BURST OF 512 CONVERSIONS
		IFYES, COMPUTE NEW INPUT

1159	006010	012737	000004	006054	3001	MOV	00,1013	ILOAD COUNT
1160	006016	012737	006060	006052		MOV	00GTP,1005	ILOAD POINTER
1161	006024	017702	000022		1001	MOV	01005,R2	IGET ADDRESS
1162	006030	004737	003402			JSR	PC,CHAR	IDISPLAY 0
1163	006034	062737	000002	006052		ADD	02,1005	IUPDATE POINTER
1164	006242	005337	006054			DEC	1019	IDEC COUNTER
1165	006046	001366				RAF	100	
1166	006050	000207				RTS	PC	IEXIT
1167	006052	000000			10051	0		
1168	006054	000000			10151	0		
1169	006056	000000			10251	0		
1170	006060	000000			00GTP1	0		
1171	006062	000000			00GT11	0		
1172	006064	000000			00GT21	0		
1173	006066	000000			00GT31	0		
1174	006070	000000			00GT41	0		
1175	006072	003756			AND1	00		
1176	006074	003763				01		
1177	006076	003770				02		
1178	006100	003775				03		
1179	006102	004002				04		
1180	006104	004007				05		
1181	006106	004014				06		
1182	006110	004021				07		
1183								
1184	006112	005737	006222		CNVTSV1	TST	MARKER	IIFST IF MARKER SAMPLE
1185	006116	001401				REQ	00	I0R IF NOT
1186	006120	005301				DEC	01	IDFC OFFSET
1187	006122	005737	013222		401	TST	IUSECLK	ICLOCK ENABE
1188	006126	001406				REQ	50	I0R IF NOT
1189	006130	012777	177750	173206		MOV	0-TR,0CSR	ILOAD PRESPT
1190	006136	012777	000003	173176		MOV	03,0CSR	ILOAD RATE
1191	006144	153777	006224	173164	501	RIQR	ADWRD1,0ADCS	IFNABLE A/D
1192	006152	000001				WAIT		
1193	006154	105077	173156		CNSTS11	CLR	0ADCS	
1194	006160	022626				CMR	(SP)+,(SP)+	ICLEAN STACK
1195	006162	017702	173152			MOV	0ADWR,R2	ILOAD CONVERTED VALUE
1196	006166	060102				ADD	R1,R2	IADD OFFSET IF ANY
1197	006170	006302				ASL	R2	I0R
1198	006172	006302				ASL	R2	
1199	006174	006302				ASL	R2	
1200	006176	010213				MOV	R2,(R3)	ILOAD AN AXIS
1201	006200	010225				MOV	R2,(R5)+	ISAVE IN BUFFER
1202	006202	105777	173140		501	TSTR	0VCSTAT	IWAIT FOR SCOPE
1203	006206	100375				RPI	50	
1204	006210	062714	000002			ADD	02,(R4)	IUPDATE O/A AXIS
1205	006214	005300				DEC	00	IDONE ALL CONVERSIONS ?
1206	006216	001335				BNP	CNVTSV	I0R IF NOT
1207	006220	000207				RTS	PC	IEXIT
1208	006222	000000			MARKER1	0		
1209	006224	000101			ADWRD11	101		
1210	006226	000000			AXIS11	0		
1211	006230	000000			AXIS21	0		
1212	006232	000000			AXIS31	0		

```
1213
1214
1215
1216
1217
1218 006234 000000
1219 006236 012737 000001 001160
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231 006244 012737 006256 013152 REPTST: MOV 0REPT1,AVECTR ISET UP CNTR 'A' VECTOR ADDRESS
1232 006252 104400 TYPE
1233 006254 012561 MES13 ITEXT 'REPEATABILITY TEST'
1234 006256 005037 013244 REPT1: CLR MESPRT
1235 006262 005037 013224 CLR OPSI
1236 006266 104400 TYPE
1237 006270 012617 MES14 IREQUEST CHANNEL (S)
1238 006272 004737 010206 JSR PC,XTTYIN IWAIT FOR INPUT
1239 006276 004737 010454 JSR PC,RCDBIN ICONVERT TO OCTAL
1240 006302 013737 010610 013162 MOV BCNTAB,KSTOR1 ISAVE AS INITIAL CH.
1241 006310 013737 013162 013164 MOV KSTOR1,KSTOR2 IALSO SAVE AS 2ND CH. ENTRY
1242 006316 005737 010612 TST RCNTAB+2 ITEST FOR SECOND ENTRY
1243 006322 001407 REQ REPT2 IBRANCH IF NO SECOND ENTRY
1244 006324 023737 010612 013162 CMP BCNTAB+2,KSTOR1 ICOMPARE ENTRY 1 TO ENTRY 2
1245 006332 100751 BMT REPT1 IBRANCH AND RESTART IF ILLEGAL
1246 006334 013737 010612 013164 MOV RCNTAB+2,KSTOR2 IOTHERWISE SAVE AS SECOND CH.
1247 006342 104400 REPT2: TYPE
1248 006344 012701 MES16 ITEXT 'COUNT SPREAD ?'
1249 006346 004737 010206 JSR PC,XTTYIN IWAIT FOR ENTRY
1250 006352 004737 010454 JSR PC,RCDBIN IDECODE TO OCTAL
1251 006356 013737 010610 013166 MOV RCNTAB,KSTOR3 ISAVE IT
1252 006364 013737 013162 013170 REPT2A: MOV KSTOR1,KSTOR4 ISAVE STARTING CH.
1253 006372 004737 011672 REPT3: JSR PC,TSTPLG ITEST FOR KEYBOARD FLAG
1254 006376 012737 001000 013160 MOV 01000,COUNT ISET FOR '512' CONVERSIONS
1255 006404 113737 013170 013147 MOV8 KSTOR4,ADWRD2+1 IMOV SELECTED CH. TO HIGH BYTE OF ADWRD
1256 006412 042737 100377 013146 RIC 0100377,ADWRD2 IMASK
1257 006420 052737 000001 013146 RIS 01,ADWRD2
1258 006426 004737 010620 JSR PC,ADCNTV ITAKE THE CONVERSIONS
1259 006432 004737 011152 JSR PC,CMPTE I AVERAGE & COMPUTE DISTRIBUTION
1260 006436 004737 011332 JSR PC,CATORZ
1261 006442 032777 002000 172066 RIT 0SW10,0SWR ITEST DATA SW10
1262 006450 001047 BNF REPT4 IIF SET, FORCE TYPE OUT
1263 006452 032777 020000 172056 TSTCT4: RIT 0SW13,0SWR ITEST FOR INHIBIT TYPEOUT
1264 006460 001142 BNE REPT7 IBRANCH IF SW SET
1265 006462 022737 000004 013166 CMP 04,KSTOR3 I WAS 4 TYPED
1266 006470 001005 BNE TSTCT5 INO, TEST FOR '3'
1267 006472 022737 001000 013362 CMP 01000,XSPRD4 ITOTAL COUNTS WITHIN 4 COUNTS
1268 006500 001033 BNE REPT4 IBRANCH IF NO.
```

1269	006502	000531			BR	REPT7				IYES, TEST NEXT CH.
1270	006504	022737	000003	013166	TSTCT3:	CMP	03,KSTOR3			ICOUNT = TO 3
1271	006512	001005			BNE	TSTCT2				I NO TEST COUNT 2
1272	006514	022737	001000	013340	CMP	01000,XSPRD3				
1273	006522	001022			BNE	REPT4				I BRANCH IF COUNT NOT WITHIN 3
1274	006524	000520			RR	REPT7				IYES, TEST NEXT CH.
1275										
1276	006526	022737	000002	013166	TSTCT2:	CMP	02,KSTOR3			ICOUNT = TO 2
1277	006534	001005			BNE	TSTCT1				I NO, TEST COUNT 1
1278	006536	022737	001000	013336	CMP	01000,XSPRD2				
1279	006544	001011			BNE	REPT4				I BRANCH IF NOT WITHIN 2
1280	006546	000507			RR	REPT7				IYES, TEST NEXT CH.
1281	006550	022737	000001	013166	TSTCT1:	CMP	01,KSTOR3			ICOUNT = TO 1
1282	006556	001004			BNE	REPT4				I NO, REPORT EVEN IF NOT '0'
1283	006560	022737	001000	013334	CMP	01000,XSPRD1				
1284	006566	001477			REQ	REPT7				I BRANCH IF TOTAL WITHIN 1 COUNT
1285	006570	104400			REPT4:	TYPE				
1286	006572	012356			ACRLF					
1287	006574	005737	013244		TST	MESPRY				I TEST IF HEADER HAS BEEN TYPED
1288	006600	001002			BNE	REPT5				I BRANCH IF YES
1289	006602	104400			TYPE					
1290	006604	012720			MES19					I TEXT 'CH, LOW AVG, HIGH'
1291	006606	104400			REPT5:	TYPE				
1292	006610	012356			ACRLF					I CARRIAGE RETURN, LINE FEED
1293	006612	013737	013170	007020	MOV	KSTOR4,REPT8A				I MOV. CH.
1294	006620	042737	177700	007020	RIC	0177700,REPT8A				
1295	006626	013746	007020		REPT8:	MOV	REPT8A,-(SP)			
1296	006632	104402			TYPOS					
1297	006634	002	001		.BYTE	2,1				
1298	006636	004737	007662		JSR	PC,XSPACE				
1299	006642	013746	013252		MOV	ADLOW,-(SP)				I SAVE LOW VALUE
1300	006646	104402			TYPOS					
1301	006650	004	001		.BYTE	4,1				
1302	006652	004737	007662		JSR	PC,XSPACE				
1303	006656	013746	013266		MOV	AVRAGE,-(SP)				I SAVE AVERAGE
1304	006662	104402			TYPOS					
1305	006664	004	001		.BYTE	4,1				
1306	006666	004737	007662		JSR	PC,XSPACE				
1307	006672	013746	013250		MOV	ADMIGH,-(SP)				
1308	006676	104402			TYPOS					
1309	006700	004	001		.BYTE	4,1				
1310	006702	005737	013244		TST	MESPRY				
1311	006706	001002			BNE	REPT6				
1312	006710	104400			TYPE					
1313	006712	012743			MES20					I PRINT 'COUNT SPREAD' HEADER
1314	006714	052737	000007	013244	REPT6:	RIS	07,MESPRY			I INHIBIT OTHER HEADERS
1315	006722	022737	001000	013316	CMP	01000,AVGCNT				I TEST IF ALL COUNTS WERE AT AVG.
1316	006730	000240			NOP					I <REQ REPT7> BRANCH TO NEXT CH. IF YES.
1317	006732	104400			TYPE					
1318	006734	012356			ACRLF					
1319	006736	012704	013302		MOV	00RLOW,R4				
1320	006742	012402			REPT6A:	MOV	(R4)+,R2			
1321	006744	004737	010754		JSR	PC,DECPY				I TYPE OUT COUNT SPREAD
1322	006750	022704	013334		CMP	0XSPRD1,R4				I TEST FOR NONE
1323	006754	001372			BNE	REPT6A				I BRANCH IF NO AND TYPE NEXT COUNT
1324	006756	005777	172154		TST	0SWR				

1325	006762	100001		RPL	REPT7	
1326	006764	000000		HALT		IREPEATABILITY ERROR
1327	006766	013700	013170	REPT71	MOV	KSTOR4,R4
1328	006772	042700	177700		RIC	0177700,R4
1329	006776	023700	013164		CMP	KSTOR2,R4
1330	007002	001400			REQ	REPT7A
1331	007004	005237	013170		INC	KSTOR4
1332	007010	000137	006372		JMP	REPT3
1333	007014	000137	006364	REPT7A1	JMP	REPT2A
1334						
1335	007220	000000		REPT8A1	R	

```
1336
1337
1338
1339
1340 007022 000000
1341
1342
1343
1344
1345
1346 007024 012737 007036 013152 RECVY1: MOV 00RECVY1,AVECTR ISET UP THE 'A' RETURN ADDRESS
1347 007032 104400 TYPE
1348 007034 012454 MESA ITEXT 'RECOVERY TEST'
1349 007036 104400 RECVY1: TYPE IREQUEST CHANNELS
1350 007040 012617 MESA
1351 007042 004737 010206 JSR PC,ITTYVIN IWAIT FOR INPUT
1352 007046 004737 010454 JSR PC,ACDBIN ICONVERT TO OCTAL
1353 007052 013737 010610 013162 MOV BCNTAB,KSTOR1 ISAVE 1ST CH.
1354 007060 013737 010612 013164 MOV BCNTAB+2,KSTOR2 ISAVE 2ND CH.
1355 007066 004737 011672 RECVY2: JSR PC,TSTFLG ICHECK FOR KEYBOARD FLAG
1356 007072 012737 000020 007406 MOV 016,,108 ILOAD COUNT
1357 007100 005037 007410 CLR 118
1358 007104 012737 000010 013234 MOV 010,TEMP3 ISET UP TO PRINT EIGHT
1359 007112 012737 000010 013160 MOV 010,COUNT ISETUP TO TAKE '8' CONVERSIONS
1360 007120 113737 013162 013147 18: MOVB KSTOR1,ADWRD2+1 ILOAD 1ST CH.
1361 007126 042737 140377 013146 BIC 0140377,ADWRD2
1362 007134 052737 000001 013146 BIS 01,ADWRD2
1363 007142 005237 013216 INC DELAY INO DELAY BETWEEN MUX CHANGE AND CONVERT
1364 007146 004737 010620 JSR PC,ADCNVT ITAKE THE CONVERSIONS
1365 007152 113737 013164 013147 MOVB KSTOR2,ADWRD2+1 ISET UP 2ND CH.
1366 007160 042737 140377 013146 BIC 0140377,ADWRD2
1367 007166 052737 000001 013146 BIS 01,ADWRD2
1368 007174 005237 013216 INC DELAY INO DELAY BETWEEN MUX CHANGE AND CONVERT
1369 007200 004737 010620 JSR PC,ADCNVT ITAKE 2ND SERIES OF CONVERSIONS
1370 007204 013701 007410 MOV 118,R1 ILOAD A POINTER
1371 007210 012700 016200 MOV 00ANBUFF,R0 ILOAD POINTER
1372 007214 012061 016242 MOV (R0)+,ADTB0(R1)
1373 007220 012061 016304 MOV (R0)+,ADTB1(R1)
1374 007224 012061 016346 MOV (R0)+,ADTB2(R1)
1375 007230 012061 016410 MOV (R0)+,ADTB3(R1)
1376 007234 012061 016452 MOV (R0)+,ADTB4(R1)
1377 007240 012061 016514 MOV (R0)+,ADTB5(R1)
1378 007244 012061 016556 MOV (R0)+,ADTB6(R1)
1379 007250 012061 016620 MOV (R0)+,ADTB7(R1)
1380 007254 062737 000002 007410 ADD 02,118 IUPDATE POINTER
1381 007262 005337 007406 DEC 108
1382 007266 001314 BNE 18
1383
1384 007270 012700 000010 MOV 00,,R0 ILOAD COUNT
1385 007274 012702 016200 MOV 00ANBUFF,R2 ILOAD POINTER
1386 007300 012701 000000 MOV 00,R1
1387 007304 012737 000017 013230 28: MOV 015,,TEMP1 ILOAD TEMP
1388 007312 012737 000004 011330 MOV 04,CMPCNT ILOAD AVRG. COUNT
1389 007320 016104 007410 MOV LANTB(R1),R4 IGET POINTER
1390 007324 004737 011172 JSR PC,CMPTEA I AVERAGE
1391 007330 013722 013266 MOV AVRG,(R2)+ ISAVE AVERAGE
```

1392	007334	005721			TSY	(R1)*		IUPDATE POINTER
1393	007336	005300			DEC	RR		
1394	007340	001361			RNE	29		IIR IF NOT DONE
1395								
1396								
1397	007342	032777	002000	171566	RIT	0SW10,0SWR		I TEST BIT 10
1398	007350	001246			RNF	RECV2		
1399	007352	104400			TYPE			
1400	007354	012505			MES0			I TEXT 'CH.'
1401	007356	013746	013164		MOV	KSTOR2,-(SP)		
1402	007362	104402			TYPOS			
1403	007364	002	001		.BYTE	2,1		
1404	007366	012737	000002	007712	MOV	02,SPACEY		
1405	007374	004737	007662		JSR	PC,XSPACE		
1406	007400	004737	007714		JSR	PC,XPRYAV		I PRINT VALUES OF 2ND CH.
1407	007404	000630			RR	RECV2		I DO IT AGAIN
1408	007406	000000						
1409	007410	000000						
1410	007412	000000						
1411								
1412	007414	016242						
1413	007416	016304						
1414	007420	016346						
1415	007422	016410						
1416	007424	016452						
1417	007426	016514						
1418	007430	016556						
1419	007432	016620						

108:	0
118:	0
128:	0

LADTR:	ADTR0
	ADTR1
	ADTR2
	ADTR3
	ADTR4
	ADTR5
	ADTR6
	ADTR7

```
1420
1421
1422
1423 007434 000000
1424 007436 012737 000000 001164
1425 007444 012737 007452 013152
1426 007452 104400
1427 007454 011772
1428 007456 005037 007636
1429 007462 000005
1430 007464 012737 000000 007636 118:
1431 007472 000200 108:
1432 007474 013700 001354 20:
1433 007500 062700 000002
1434 007504 012701 007660
1435 007510 012702 000010
1436 007514 010100 18:
1437 007516 005302
1438 007520 001375
1439
1440 007522 013700 001336
1441 007526 012701 007640
1442 007532 012702 000010
1443
1444 007536 011137 001124 38:
1445 007542 011037 001126
1446 007546 023737 001124 001126
1447 007554 001403
1448 007556 010037 007660
1449 007562 104001
1450 007564 022021 48:
1451 007566 005302
1452 007570 001362
1453 007572 005077 171550
1454 007576 004737 011672
1455 007602 005337 007634
1456 007606 001331
1457 007610 104400
1458 007612 004521
1459 007614 005237 007636
1460 007620 013746 007636
1461 007624 104400
1462 007626 104400 004516
1463 007632 000714
1464 007634 000000 708:
1465 007636 000000 718:
1466
1467
1468
1469 007640 027560
1470 007642 000000
1471 007644 140736
1472 007646 000377
1473 007650 007214
1474 007652 001292
1475 007654 000525

;*****
;TEST 10 LOAD DIFFERENT NUMBERS INTO DIFFERENT REG.
;*****
TST14: SCOPF
MOV 000,STIMES ;1000 40 ITERATIONS
MOV 0200,AVECTR
208: TYPE
MES2 ;TYPE HEADER ABOUT TEST
CLR 718
RESET
MOV 00,708 ;LOAD COUNTER
108: NOP
20: MOV CSC,RR ;LOAD STARTING ADDRESS
ADD 02,00
MOV 001FNUM+20,R1 ;LOAD STARTING TABLE ADDRESS
MOV 0A,,R2 ;LOAD COUNT
MOV -(R1),-(R0) ;LOAD THE REG
DEC R2 ;DONE ALL ?
BNF 19 ;BR IF NOT

MOV ANCS,RR ;LOAD STARTING POINTER
MOV 00RPNUM,R1 ;LOAD STARTING POINTER <EXPECTED>
MOV 0A,,R2 ;LOAD # OF REG

38: MOV (R1),SGDDAT ;READ REG
MOV (RR),SDDAT ;READ REG
CMP SGDDAT,SDDAT ;COMPARE
;BR IF EQUAL
MOV 00,BUFADR ;SAVE BUS ADDRESS
ERROR 1 ;INCORRECT DATA, REG. WAS CHANGED IN ERROR
48: CMP (RR)+,(R1)+ ;UPDATE POINTERS
DEC R2 ;DONE ALL REG ?
BNF 38 ;BR IF NOT

CLR 0VCESTAT
JSR PC,TSTFLG
DEC 708 ;DONE
BNF 108 ;NO
TYPE
SENDMG
INC 718
MOV 719,-(SP)
TYPDS
TYPE ,SPNULL
RR 118 ;LOOP BACK
708: R
718: 0

;0 TO BE LOADED INTO DIFFERENT REG
BUFNUM: 27560
0
140736
377
7214
1292
525
;A TO D STATUS
;A TO D BUFFER
;LOCK STATUS
;CLOCK PRESET
;VC STATUS
;VC X POS
;VC Y POS
```

```

1476 007656 000377          377          ICLOCK COUNTER
1477
1478 007660 170400          BUFADR: 170400          IBUS ADDRESS OF REG IN ERROR
1479
1480
1481
1482
1483
1484
1485
1486 007662 105777 171260          XSPACE: TSTR          0STPS          IWAIT FOR TTY READY
1487 007666 100375          RPL          -4
1488 007670 012777 000240 171252          MOV          0240,0STPR          IOUTPUT A SPACE
1489 007676 005337 007712          DEC          SPACEX          IDECREMENT COUNT
1490 007702 003367          RGT          XSPACE          ILOOP IF NOT DONE
1491 007704 005037 007712          CLR          SPACEX          IRESET COUNT TO ZERO
1492 007710 000207          RTS          PC          IRETURN
1493 007712 000000          SPACEX: 0
1494
1495
1496
1497 007714 012737 016200 007724          XPRTAB: MOV          0ADRUFF,AVGTAB
1498 007722 013746          XPTAB: MOV          0(PC)+,-(SP)
1499 007724 016200          AVGTAB: ADRUFF
1500 007726 104402          TYPOS
1501 007730          004          001          .BYTE          0,1
1502 007732 062737 000002 007724          ADD          02,AVGTAB
1503 007740 012737 000002 007712          MOV          02,SPACEX
1504 007746 004737 007662          JSR          PC,XSPACE
1505 007752 005337 013234          DEC          TFMPS
1506 007756 001361          RNE          XPTAB
1507 007760 000207          RTS          PC
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523 007762          010046
1524 007764          010146
1525 007766          010246
1526 007770          010346
1527 007772          010546
1528 007774          012746 020200
1529 010000          016605 000020
1530 010004          100004
1531 010006          005405

```

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```

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

```

```

STYPDS:
MOV          R0,-(SP)          I PUSH R0 ON STACK
MOV          R1,-(SP)          I PUSH R1 ON STACK
MOV          R2,-(SP)          I PUSH R2 ON STACK
MOV          R3,-(SP)          I PUSH R3 ON STACK
MOV          R4,-(SP)          I PUSH R4 ON STACK
MOV          020200,-(SP)          I SET BLANK SWITCH AND SIGN
MOV          20(SP),R5          I GET THE INPUT NUMBER
BPL          IS          I BR IF INPUT IS POS.
NEG          R5          I MAKE THE BINARY NUMBER POS.

```


1532	010010	112766	000055	000001	MOVR	0',1(SP)	;;MAKE THE ASCII NUMBER NEG.
1533	010016	005000			CLR	R0	;;ZERO THE CONSTANTS INDEX
1534	010020	012703	010176		MOV	00BLK,R3	;;SETUP THE OUTPUT POINTER
1535	010024	112723	000000		MOVR	0',(R3)+	;;SET THE FIRST CHARACTER TO A BLANK
1536	010030	005002			CLR	R2	;;CLEAR THE BCD NUMBER
1537	010032	016001	010166		MOV	00TAL(R0),R1	;;GET THE CONSTANT
1538	010036	160105			SUB	R1,R5	;;FORM THIS BCD DIGIT
1539	010040	002402			RLT	R9	;;RR IF DONE
1540	010042	005202			INC	R2	;;INCREASE THE BCD DIGIT BY 1
1541	010044	000774			RR	38	
1542	010046	060105			ADD	R1,R5	;;ADD BACK THE CONSTANT
1543	010050	005702			TST	R2	;;CHECK IF BCD DIGIT=0
1544	010052	001002			RNF	R8	;;FALL THROUGH IF 0
1545	010054	105716			TSTB	(SP)	;;STILL DOING LEADING 0'S?
1546	010056	100007			BMJ	78	;;RR IF YES
1547	010060	106316			ASLR	(SP)	;;MSD?
1548	010062	103003			RCC	68	;;RR IF NO
1549	010064	116663	000001	177777	MOVR	1(SP),-1(R3)	;;YES--SET THE SIGN
1550	010072	052702	000060		RIS	0',R2	;;MAKE THE BCD DIGIT ASCII
1551	010076	052702	000000		BIS	0',R2	;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
1552	010102	110223			MOV8	R2,(R3)+	;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
1553	010104	005720			TST	(R0)+	;;JUST INCREMENTING
1554	010106	020027	000010		CMR	R0,#10	;;CHECK THE TABLE INDEX
1555	010112	002746			BLT	28	;;GO DO THE NEXT DIGIT
1556	010114	003002			RGT	08	;;GO TO EXIT
1557	010116	010502			MOV	R5,R2	;;GET THE LSD
1558	010120	000764			RR	68	;;GO CHANGE TO ASCII
1559	010122	105726			TSTB	(SP)+	;;WAS THE LSD THE FIRST NON-ZERO?
1560	010124	100003			RPI	08	;;RR IF NO
1561	010126	116663	177777	177776	MOVR	-1(SP),-2(R3)	;;YES--SET THE SIGN FOR TYPING
1562	010134	105013			CLRB	(R3)	;;SET THE TERMINATOR
1563	010136	012605			MOV	(SP)+,R5	;;POP STACK INTO R5
1564	010140	012603			MOV	(SP)+,R3	;;POP STACK INTO R3
1565	010142	012602			MOV	(SP)+,R2	;;POP STACK INTO R2
1566	010144	012601			MOV	(SP)+,R1	;;POP STACK INTO R1
1567	010146	012600			MOV	(SP)+,R0	;;POP STACK INTO R0
1568	010150	100000	010176		TYPE	,00BLK	;;NOW TYPE THE NUMBER
1569	010154	016666	000002	000000	MOV	2(SP),4(SP)	;;ADJUST THE STACK
1570	010162	012616			MOV	(SP)+,(SP)	
1571	010164	000002			RTI		;;RETURN TO USER
1572	010166	023020					
1573	010170	001750					
1574	010172	000144					
1575	010174	000012					
1576	010176	000004					

00TAL: 10000.
00BLK: .0LKW 4

```

1577                                IKEYBOARD SERVICE ROUTINE
1578
1579 010206 012704 010436          XTTYINI: MOV      01NBUF,R4          ISETUP CHARACTER BUFFER
1580 010212 042777 000100 170722  BIC      00176,08TKS
1581 010220 005037 013156          CLR      CHRCNT          ICLEAR CHARACTER COUNTER
1582 010224 005037 010436          CLR      INBUF
1583 010230 005037 010440          CLR      INBUF+2
1584 010234 105777 170702          118:   TSTR     08TKS          ICHARACTER READY?
1585 010240 100375                    BPL      118              INO, WAIT IT OUT
1586 010242 017701 170676          MOV      08TKR,R1        ISAVE CHARACTER
1587 010246 042701 177640          BIC      0177640,R1     ISTRIP PARITY BIT
1588 010252 120127 000060          CMPR    R1,060          IIS IT A SPECIAL CHARACTER ?
1589 010256 100420          RMT      148              IYES, TEST IT
1590 010260 122701 000137          CMPR    0137,R1
1591 010264 100015          RMT      148
1592 010266 010124          128:   MOV      R1,(R4)+    ISAVE CHARACTER
1593 010270 005237 013156          INC     CHRCNT          IINCREMENT THE CHARACTER COUNT.
1594 010274 022737 000006 013156  CMP      06,CHRCNT
1595 010302 100451          BMT      48
1596 010304 105777 170636          132:   TSTR     08TSP          IECHO CHARACTER
1597 010310 100375                    BPL      138
1598 010312 110177 170632          MOV     R1,08TSP
1599 010316 000746          BR       118              IWAIT FOR NEXT CHARACTER
1600                                ISUBROUTINE TO TEST FOR SPECIAL CHARACTERS 'A','C','G','CR',',',' OR 'RUBOUT'
1601
1602 010320 122701 000001          148:   CMPE    01,R1          ICHAR,0'A'?
1603 010324 001005          BNE     18              INO, NOT 'A'
1604 010326 104400          TYPE          IECHO 'A'
1605 010330 012351          CNTRLA
1606 010332 005726          TST     (SP)+          IRESTORE SP
1607 010334 000177 002612          JMP     0AVECTR        IYES, EXIT VIA 'A' VECTOR ADDRESS.
1608 010340 122701 000003          18:   CMPE    03,R1          ICHAR,0'C'?
1609 010344 001003          BNE     28              INO, NOT 'C'
1610 010346 005726          TST     (SP)+
1611 010350 000137 002076          JMP     INIT2          IYES, EXIT TO MONITOR
1612 010354 122701 000137          28:   CMPE    0137,R1        ICHAR,0'RUBOUT'?
1613 010360 001011          BNE     38              IIGNORE CHAR. & EXIT
1614 010362 005737 013156          TST     CHRCNT          IIS RUBOUT LEGAL?
1615 010366 001722          BEQ     118             INO, IGNORE IT
1616 010370 005337 013156          DEC     CHRCNT
1617 010374 012701 000134          MOV     0130,R1          ITYPE '\ ' TO INDICATE RUBOUT
1618 010400 005744          TST     -(R4)           IPOP OFF LAST CHARACTER
1619 010402 000740          RR      138              IWAIT FOR NEXT CHARACTER
1620 010404 122701 000054          38:   CMPE    050,R1          ITEST FOR ','
1621 010410 001726          BEQ     178              ILEGAL CHAR., SAVE IT
1622 010412 122701 000015          CMPE    015,R1         I'YO 'CARRIAGE RETURN' TO TERMINATE?
1623 010416 001003          BNE     48              INO, CONTINUE
1624 010420 104400          TYPE          IYES, TYPE 'CR-LF'
1625 010422 012356          ACRLF
1626 010424 000207          RTS      PC              IEXIT

```

1627 P10426 104405
 1628 P10430 104400
 1629 P10432 P12365
 1630 P10434 000664
 1631 P10436 000000
 1632
 1633
 1634 P10450
 1635
 1636
 1637
 1638 P10454 P12704 P10436
 1639 P10460 P12703 P10610
 1640 P10464 P05037 P10612
 1641 P10470 P05001
 1642 P10472 P05002
 1643 P10474 P05737 P13156
 1644 P10500 P03424
 1645 P10502 P05337 P13156
 1646 P10506 122714 P00054
 1647 P10512 P01417
 1648 P10514 121427 P00060
 1649 P10520 P02425
 1650 P10522 P21427 P00067
 1651 P10526 P03022
 1652 P10530 P42714 177770
 1653 P10534 P12400
 1654 P10536 P10102
 1655 P10540 P06301
 1656 P10542 P06301
 1657 P10544 P06301
 1658 P10546 P60001
 1659 P10550 P00751
 1660 P10552 P20127 P00077
 1661 P10556 P03006
 1662 P10560 P05724
 1663 P10562 P10123
 1664 P10564 P05737 P13156
 1665 P10570 P01337
 1666 P10572 P00207
 1667 P10574 104400
 1668 P10576 P12365
 1669 P10600 P04737 P10206
 1670 P10604 P00137 P10454
 1671 P10610 P00000
 1672 P10612 P00000
 1673 P10614 P00000
 1674 P10616 P00000

481 CKSWR
 TYPE
 QMARK
 RR XTTYIN
 INBUF: 0
 .0.+14
 BCDINI: MOV @INBUF,R4
 MOV @BCDTAB,R3
 CLR BCDTAB+2
 BCDNB1: CLR R1
 CLR R2
 BCDNB2: TST CHRCNT
 BLE BCDEND
 DEC CHRCNT
 CMPS B54,(R4)
 BEQ BCDP'D
 CMPS (R4),B60
 BLT BCDERR
 CMP (R4),B67
 BGT BCDERR
 BIC @177770,(R4)
 MOV (R4)+,R2
 MOV R1,R2
 ASL R1
 ASL R1
 ASL R1
 ADD R0,R1
 RR BCDNB2
 BCDEND: CMP R1,B77
 RGT BCDERR
 TST (R4)+
 MOV R1,(R3)+
 TST CHRCNT
 RNF BCDNB1
 RTS PC
 BCDERR: TYPE
 QMARK
 JSR PC,XTTYIN
 JMP BCDNB1
 BCDTAB: 0
 0
 0
 0

I TEST FOR CTRL G
 I OTHERWISE TYPE '7'
 I WAIT FOR NEW ENTRY
 I CHARACTER STORAGE BUFFER
 I SETUP ASCII STORAGE TABLE
 I TABLE FOR STORAGE OF CONVERTED WORDS
 I REG. TO STORE RUNNING TOTAL
 I TEMP. STORAGE FOR 'R1'
 I END OF DATA?
 I YES, EXIT
 I DECREMENT CHARACTER COUNTER
 I IS CHARACTER = TO ' '?
 I YES, DECODE NEW WORD
 I TEST FOR LEGAL NO.
 I STRIP NO.
 I SAVE NO. IN RR.
 I SAVE CURRENT TOTAL
 INX2
 INX4
 INX8
 IN+NEW NO.
 I UPDATE BUFFER
 I SAVE CONVERTED VALUE & SETUP TO SAVE NEXT
 I FINISHED?
 I NO, CONVERT NEXT WORD
 I YES, EXIT
 I TYPE '7'.
 I TO BE TYPED ON QUESTIONABLE ENTRIES.
 I OCTAL STORAGE TABLE

```
1675          ISURROUTINE TO TAKE 'N' CONVERSIONS AND STORE THEM IN AN A/D BUFFER. ROUTINE
1676          IS ENTERED WITH 'N' IN COUNT AND THE CH TO BE CONVERTED IN 'ADWORD'.
1677          ENTERING WITH DELAY = P CAUSES DELAY BETWEEN MUX LOAD AND START OF
1678          IFIRST CONVERSION
1679
1680 P10620 005077 167152          ADCNVT: CLR          0PSW
1681 010624 013737 013160 013290          MOV          COUNT,TEMP1      ISET UP NO. OF CONVERSIONS TO BE TAKEN
1682 010632 012704 016200          MOV          @ANDBUFF,R4      ISET UP BUFFER ADDRESS.
1683 010636 005737 013216          TST          DELAY          ICHECK IF DELAY SET
1684 010642 001017          BNE          30              IBR IF INHIBIT DELAY
1685 010644 113777 013147 000100          MOVR         @WORD2+1,@ADCS1   ILOAD MUX
1686 010652 013737 010746 013220          MOV          CPTIME,DFLAY1     ISET UP TIMER
1687 010660 005337 013220          40:         DEC          DELAY1   ILOOP
1688 010664 001375          RNE          40
1689 010666 113777 013146 170442          MOVB        @WORD2,@ADCS      ILOAD LOW BYTE
1690 010674 000405          RR          10
1691 010676 000240          20:         NOP
1692 010700 000240          NOP
1693 010702 013777 013146 170426 30:         MOV          @WORD2,@ADCS      ILOAD CONTROL
1694          ILOAD CH. & START CONVERT
1695 010710 105777 170422          10:         TSTB          @ADCS
1696 010714 100375          BPL          10              IWAIT FOR DONE
1697 010716 017724 170416          MOV          @ANDBR,(R4)+     ISAVE DATA
1698 010722 005337 013230          DEC          TEMP1           IDECREMENT COUNTER
1699 010726 003363          BGT          20              IIF NOT '0' TAKE NEXT CONVERSION
1700 010730 005037 013216          CLR          DELAY
1701 010734 000207          RTB          PC
1702
1703          CPDLAY: 2000          IPDP-11/05
1704          2400          IPDP-11/20
1705          3500          IPDP-11/40
1706          6000          IPDP-11/65
1707
1708          CPTIME: 2000
1709          CPTYPE: 1
1710          ADCS1: 170401
1711
```

```
1712
1713          JPRINT DECIMAL VALUE IN R2
1714
1715 010754 005077 167010      DECPRT: CLR      0PSW
1716 010760 012737 177774 011134      MOV      0-4,DIGCNT
1717 010766 012737 011102 011140      MOV      00FCPNT+2,DECPNT
1718 010774 012737 000200 011136      MOV      0200,ZERO
1719 011002 012737 177777 011132  TVPT1:  MOV      0-1,DIGIT
1720 011010 005237 011132      TVPT2:  INC      DIGIT
1721 011014 167702 000120      SUB      0DECPNT,X2
1722 011020 100373      BPL      TVPT2
1723 011022 067702 000112      ADD      0DECPNT,X2
1724 011026 004737 011052      JSR      PC,DECOUT
1725 011032 005237 011134      TNP      DIGCNT
1726 011036 001001      RNE      TVPT3
1727 011040 000207      RTS      PC
1728 011042 062737 000002 011140  TVPT3:  ADD      02,DECPNT
1729 011050 000754      BR      TVPT1
1730 011052 005737 011132      DECOUT:  TST      DIGIT
1731 011056 001010      BNE      DEC1
1732 011060 022737 177777 011134      CMP      0-1,DIGCNT
1733 011066 001404      BEQ      DEC1
1734 011070 013737 011136 011132      MOV      ZERO,DIGIT
1735 011076 000406      RR      DEC2
1736 011100 012737 000200 011136  DEC1:  MOV      0200,ZERO
1737 011106 052737 000200 011132      RIS      0200,DIGIT
1738 011114 105777 170020      DEC2:  TSTR      0STPB
1739 011120 100375      BPL      0-4
1740 011122 013777 011132 170020      MOV      DIGIT,0STPB
1741 011130 000207      RTS      7
```

1742 011132 000000
1743 011134 000000
1744 011136 000200
1745 011140 011100
1746 011142 001750
1747 011144 000100
1748 011146 000012
1749 011150 000001

DIGITS: 0
DIGCNT: 0
ZEROS: 240
DECPNT: .02
1000.
100.
10.
1.

1750
1751

COMPUTE THE RESULTS OF 512 CONVERSIONS AS HIGH, LOW AND AVERAGE

1752
1753 011152 012737 000777 013230
1754 011160 012737 000011 011330
1755 011166 012700 016200
1756 011172 005037 013246
1757 011176 012437 013266
1758 011202 013737 013266 013250
1759 011210 013737 013266 013252
1760 011216 012437 013232
1761 011222 023737 013232 013250
1762 011230 003003
1763 011232 013737 013232 013250
1764 011240 023737 013232 013252
1765 011246 003003
1766 011250 013737 013232 013252
1767 011256 003737 013232 013266
1768 011264 005537 013246
1769 011270 005537 013230
1770 011274 001350
1771 011276 013737 011330 013230
1772 011304 006237 013246
1773 011310 006037 013266
1774 011314 005537 013230
1775 011320 001371
1776 011322 005537 013266
1777 011326 000207
1778
1779 011330 000011

CMPTES: MOV 0777,TEMP1
MOV 011,CMPCNT
CMPTES: MOV 0000FF,00
CMPTES: CLR HIORDV
MOV (00)+,AVRAGE
MOV AVRAGE,ADHIGH
MOV AVRAGE,ADLOW
GETDAT: MOV (00)+,TEMP2
CMP TEMP2,ADHIGH
BLE TSLO
MOV TEMP2,ADHIGH
TSLO: CMP TEMP2,ADLOW
AGT TAGA
MOV TEMP2,ADLOW
TAGA: ADD TEMP2,AVRAGE
ADC HIORDV
DEC TEMP1
RNF GETDAT
MOV CMPCNT,TEMP1
AVGDAT: ABR HIORDV
ROR AVRAGE
DEC TEMP1
RNF AVGDAT
ADC AVRAGE
RTS PC

1SET UP TO COMPARE '511' NUMBERS
1LOAD COUNTER
1LOAD STARTING ADDRESS
1CLR HI ORDER DIVIDEND
1STORE 1ST VALUE AS AVERAGE
1HIGH
1B LOW

1IS NEW NO. GREATER THAN OLD NO. ?
1BRANCH IF NOT GREATER
1OTHERWISE SAVE AS NEW HIGH

1OTHERWISE SAVE AS NEW LOW
1ADD LOW ORDER
1ADD CARRY TO HI ORDER

1512 ADDITIONS?
1YES, DIVIDE/512

1SHIFT CARRY BIT INTO LO ORDER

1DONE?
1YES, ADD REMAINDER TO LO ORDER

CMPCNT: 11

```
1780                                     ISUBROUTINE TO CALCULATE THE PLUS & MINUS 5 COUNT LIMITS FROM AN AVERAGE
1781
1782 011332 012737 000005 013230 CATR21 MOV 05,TEMP1
1783 011340 013737 013266 013232          MOV AVERAGE,TEMP2      IMOV AVER. TO WORK AREA
1784 011346 012703 013270          MOV SAVERP1,R3        ISETUP DISTRIBUTION TABLE (POS.)
1785 011352 005237 013232          FILE11 INC TEMP2      IABA+1
1786 011356 013723 013232          MOV TEMP2,(R3)+      ISAVE A+1
1787 011362 005337 013230          DEC TEMP1            ISAVED '5' COUNTS?
1788 011366 001371          BNE FILE1            IBRANCH IF NO
1789
1790 011370 012737 000005 013230          MOV 05,TEMP1
1791 011376 013737 013266 013232          MOV AVERAGE,TEMP2   IMOV AVG. TO WORK AREA.
1792 011400 012703 013266          MOV SAVERAGE,R3     ISET UP DISTRIBUTION TABLE NEG.
1793 011410 005337 013232          FILE21 DEC TEMP2     IAB1=1
1794 011414 013743 013232          MOV TEMP2,-(R3)     ISAVE 'A-1'
1795 011420 005337 013230          DEC TEMP1            ISAVED '5' COUNTS?
1796 011424 001371          BNE FILE2            IBRANCH IF NO
1797
1798                                     ICATEGORIZE THE COUNT SPREAD AS '+6 & -6' COUNTS FROM THE AVERAGE
1799
1800 011426 012703 013302          MOV ORLOW,R3         ICLEAR COUNTS
1801 011432 005023          CATR11 CLR (R3)+
1802 011434 022703 013334          CMP ORHIGH+2,R3     IFINISHED?
1803 011440 001374          ANF CATR1           INO, CLEAR NEXT COUNTER
1804 011442 012737 001001 013230          MOV 01001,TEMP1     ICOMPARE '512' COUNTS
1805 011450 012700 016200          MOV MADRUFF,R0      ISET UP A/D BUFFER
1806 011454 005337 013230          CATR21 DEC TEMP1
1807 011460 001437          BEQ CATR5           IEXIT IF '0'
1808 011462 012037 013232          MOV (R0)+,TEMP2
1809 011466 023737 013300 013232          CMP AVERP5,TEMP2
1810 011474 100423          BMT OVRHI
1811 011476 023737 013232 013254          CMP TEMP2,AVERM5
1812 011504 100422          BMT OVRLO
1813 011506 005001          CLR R1
1814 011510 012702 013254          MOV SAVERM5,R2
1815 011514 022237 013232          CATR31 CMP (R2)+,TEMP2
1816 011520 001405          REQ CATR4
1817 011522 005201          INC R1
1818 011524 022701 000013          CMP 013,R1
1819 011530 001371          BNE CATR3
1820 011532 000000          HALT
1821 011534 006301          CATR41 ASL R1          IFATAL ERROR
1822 011536 005261 013304          INC MINUS5(R1)      IMULTIPLY 'OFFSET' X2
1823 011542 000744          BR CATR2
1824 011544 005237 013332          OVRHI: INC ORHIGH
1825 011550 000741          BR CATR2
1826 011552 005237 013302          OVRLO: INC ORLOW
1827 011556 000736          BR CATR2
```

```
1828
1829
1830
1831 011560 013737 013316 013334 CATRS1 MOV AVGCNT,XSPRD1
1832 011566 063737 013320 013334 ADD PLUS1,XSPRD1
1833 011574 063737 013314 013334 ADD MINUS1,XSPRD1
1834 011602 013737 013334 013336 MOV XSPRD1,XSPRD2
1835 011610 063737 013322 013336 ADD PLUS2,XSPRD2
1836 011616 063737 013312 013336 ADD MINUS2,XSPRD2
1837 011624 013737 013336 013340 MOV XSPRD2,XSPRD3
1838 011632 063737 013324 013340 ADD PLUS3,XSPRD3
1839 011640 063737 013310 013340 ADD MINUS3,XSPRD3
1840 011646 013737 013340 013342 MOV XSPRD3,XSPRD4
1841 011654 063737 013326 013342 ADD PLUS4,XSPRD4
1842 011662 063737 013306 013342 ADD MINUS4,XSPRD4
1843 011670 000207 RTS PC
1844
1845
1846
1847 011672 105777 167244 TSTFLG1 TSTB 08YK9
1848 011676 100002 BPL 18
1849 011700 004737 010206 JSR PC,XTTYIN
1850 011704 000207 ISI RTS PC
1851
```

1ADD THE COUNTS AND SAVE TOTAL IN SPREADS OF '1-4'

1=TO NO. COUNTS AT SPREAD OF '1'

1=TO NO. COUNTS AT SPREAD OF '2'

1=TO NO. COUNTS AT SPREAD OF '3'

1=TO NO. COUNTS AT SPREAD OF '4'

1SUBROUTINE TO TEST FOR THE KEYBOARD FLAG BEING SET

1FLAG SET?
1NO, EXIT
1YES, INQUIRE

1852	011706	005015	040412	026522	TITLE1	.ASCIZ	<15><12><12>'AR-11 DIAGNOSTIC TEST II, (MAINDEC-11-DZARR-8)'<15><12>
1853	011714	030461	042040	040511			
1854	011722	047107	051517	044524			
1855	011730	020103	042524	052123			
1856	011736	044440	026111	024040			
1857	011744	040515	047111	042504			
1858	011752	026503	030461	042055			
1859	011760	040532	041122	041055			
1860	011766	006451	000012				
1861	011772	005015	052123	052101	MESS21	.ASCIZ	<15><12>'STATIC REGISTER TEST'<15><12>
1862	012000	041511	051040	043505			
1863	012006	051511	042524	020122			
1864	012014	042524	052123	005015			
1865	012022	000					
1866	012023	015	042412	040522	MESS31	.ASCIZ	<15><12>'ERASE RETURN FAILED TO SET READY'<15><12>
1867	012030	042523	051040	052105			
1868	012036	051125	020116	040506			
1869	012044	046111	042105	052040			
1870	012052	020117	042523	020124			
1871	012060	042522	042101	006531			
1872	012066	000012					
1873	012070	005015	054524	042520	MESS41	.ASCIZ	<15><12>'TYPE LETTER ' ' TO RUN DESIRED TEST'<15><12>
1874	012076	046040	052105	042524			
1875	012104	020122	020047	020047			
1876	012112	047524	051040	047125			
1877	012120	042040	051505	051111			
1878	012126	042105	052040	051505			
1879	012134	035124	005015				
1880	012140	040447	036447	052123		.ASCIZ	'A'<15><12>'STATIC REGISTER TEST'<15><12>
1881	012146	052101	041511	051040			
1882	012154	043505	051511	042524			
1883	012162	020122	042524	052123			
1884	012170	005015					
1885	012172	041047	036447	041523		.ASCIZ	'R'<15><12>'SCOPE ADJUSTMENT TESTS'<15><12>
1886	012200	050117	020105	042101			
1887	012206	052512	052123	042515			
1888	012214	052116	052040	051505			
1889	012222	051524	005015				
1890	012226	041447	036447	020101		.ASCIZ	'C'<15><12>'A TO D CALIBRATION TEST'<15><12>
1891	012234	047524	042040	041440			
1892	012242	046101	041111	040522			
1893	012250	044524	047117	052040			
1894	012256	051505	006524	012			
1895	012263	047	023504	040475		.ASCIZ	'D'<15><12>'A TO D REPEATABILITY'<15><12>
1896	012270	052040	020117	020104			
1897	012276	042522	042520	052101			
1898	012304	041101	046111	052111			
1899	012312	006531	012				
1900	012315	047	023505	040475		.ASCIZ	'E'<15><12>'A TO D RECOVERY'<15><12>
1901	012322	052040	020117	020104			
1902	012330	042522	047503	042526			
1903	012336	054522	005015	000			
1904	012343	136	006503	027012	CNTRL01	.ASCIZ	'C'<15><12><56>
1905	012350	000					
1906	012351	136	006501	000012	CNTRLA1	.ASCIZ	'A'<15><12>
1907	012356	015	012	000	ACRLF1	.BYTF	15,12,0

1908	012361	015	012	056	DOT:	.BVTF	15,12,56,0
1909	012364	000					
1910	012365	077	000040		MARK:	.ASCIZ	'?'
1911	012370	005015	041523	050117	MSG1:	.ASCIZ	<15><12>'SCOPE ADJUSTMENT TEST'
1912	012376	020105	042101	052512			
1913	012404	052123	042515	052116			
1914	012412	052040	051505	000120			
1915	012420	005015	020101	047524	MSG7:	.ASCIZ	<15><12>'A TO D CALIBRATION TEST'<15><12>
1916	012426	042040	041440	046101			
1917	012434	041111	040522	044524			
1918	012442	047117	052040	051505			
1919	012450	006524	000012				
1920	012454	005015	020101	047524	MSG8:	.ASCIZ	<15><12>'A TO D RECOVERY TEST'<15><12>
1921	012462	042040	051040	041505			
1922	012470	053117	051105	020131			
1923	012476	042524	052123	005015			
1924	012504	000					
1925	012505	015	041412	020110	MSG9:	.ASCIZ	<15><12>'CH'
1926	012512	000					
1927	012513	015	023412	023505	MSG10:	.ASCIZ	<15><12>'E'XT. OR 'I'NT. OR 'C'LOCK. SYNC?'
1928	012520	052130	020056	051117			
1929	012526	023440	023511	052116			
1930	012534	020056	051117	023440			
1931	012542	023503	047514	045503			
1932	012550	020056	054523	041516			
1933	012556	020077	000				
1934	012561	015	040412	052040	MSG13:	.ASCIZ	<15><12>'A TO D REPEATABILITY TEST'<15><12>
1935	012566	020117	020104	042522			
1936	012574	042520	052101	041101			
1937	012602	046111	052111	020131			
1938	012610	042524	052123	005015			
1939	012616	000					
1940	012617	015	005012	044103	MSG14:	.ASCIZ	<15><12><12>'CH.(8)?'
1941	012624	024056	024523	020077			
1942	012632	000					
1943	012633	015	051412	051127	MSG15:	.ASCIZ	<15><12>'SWRB AND D THRU 2 CONTROL PATTERN'<15><12>
1944	012640	020070	047101	020104			
1945	012646	020060	044124	052522			
1946	012654	031040	041440	047117			
1947	012662	051124	046117	050040			
1948	012670	052101	042524	047122			
1949	012676	005015	000				
1950	012701	103	052517	052116	MSG16:	.ASCIZ	'COUNT SPREAD?'
1951	012706	051440	051120	040505			
1952	012714	037504	000040				
1953	012720	005015	044103	020056	MSG19:	.ASCIZ	<15><12>'CH. LO AV HI'
1954	012726	047514	020040	040440			
1955	012734	020126	020040	044510			
1956	012742	000					
1957	012743	015	020012	046040	MSG20:	.ASCIZ	<15><12>' LO -5 -4 -3 -2 -1 AV +1 +2 +3 +4 +5 HI'
1958	012750	020117	026440	020065			
1959	012756	026440	020064	026440			
1960	012764	020063	026440	020062			
1961	012772	026440	020061	040440			
1962	013000	020126	025440	020061			
1963	013006	025440	020062	025440			

1964	013014	020063	025440	020064		
1965	013022	025440	020065	040000		
1966	013030	000111				
1967	013032	042522	044507	052123	EM11	.ASCIZ /REGISTER CONTENTS CHANGED IN ERROR/
1968	013040	051105	041400	047117		
1969	013046	042524	052116	020123		
1970	013054	044103	047101	042507		
1971	013062	020104	047111	042440		
1972	013070	051122	051117	000		
1973	013075	105	051122	041520	DM11	.ASCIZ /ERRPC BUFADR EXPECT RAD/
1974	013102	020040	041040	043125		
1975	013110	042101	020122	020040		
1976	013116	054105	042520	052103		
1977	013124	020040	020040	040502		
1978	013132	000104				
1979						
1980	013134	001116	027660	001124	DT11	.EVEN SERRPC, BUFADR, SGGDAT, SRDDAT, 0
1981	013142	001126	000000			
1982	013146	000000			ANWRD2:	0 FLOW BYTE OF 'ANWORD'
1983	013150	000000			PRINT1:	0
1984	013152	002072			AVECTR:	INITA JMA VECTOR ADDRESS
1985	013154	000000			PROCI:	0 ITEMP STORAGE FOR 'PSW'
1986	013156	000000			CHRCNT:	0 ITEMP STORAGE
1987	013160	000000			COUNT:	0 ITEMP STORAGE
1988	013162	000000			KSTOR1:	0 PERMANENT STORAGE
1989	013164	000000			KSTOR2:	0 PERMANENT STORAGE
1990	013166	000000			KSTOR3:	0 PERMANENT STORAGE
1991	013170	000000			KSTOR4:	0 PERMANENT STORAGE
1992	013172	000000			KSTOR5:	0
1993	013174	000000			KSTR11:	0
1994	013176	000000			KSTR12:	0
1995	013200	000000			LOW:	0
1996	013202	000000			HIGH:	0
1997	013204	000010			INCR:	10
1998	013206	000000			TIMSV:	0
1999	013210	000000			TICKS:	0
2000	013212	000000			SYCHAN:	0
2001	013214	000000			FIRST:	0
2002	013216	000000			DELAY:	0
2003	013220	000000			DELAY1:	0
2004	013222	000000			USECLK:	0
2005	013224	000000			OPS:	0
2006	013226	000000			TEMP:	0
2007	013230	000000			TEMP1:	0 TEMPORARY STORAGE
2008	013232	000000			TEMP2:	0 TEMPORARY STORAGE
2009	013234	000000			TEMP3:	0 TEMPORARY STORAGE
2010	013236	000000			BRLEV1:	0
2011	013240	000000			BRLEV2:	0
2012	013242	000000			BRLEV3:	0
2013	013244	000000			MESPR:	0
2014	013246	000000			MIORDV:	0
2015	013250	000000			ADHIGH:	0
2016	013252	000000			ADLOW:	0
2017	013254	000000			AVRM5:	0
2018	013256	000000			AVRM4:	0
2019	013260	000000			AVRM3:	0

2023 013262 000000 AVERM2: 0
2021 013264 000000 AVERM1: 0
2022 013266 000000 AVERAGE: 0
2023 013270 000000 AVERP1: 0
2024 013272 000000 AVERP2: 0
2025 013274 000000 AVERP3: 0
2026 013276 000000 AVERP4: 0
2027 013300 000000 AVERP5: 0
2028 013302 000000 ORLOW: 0
2029 013304 000000 MINUS5: 0
2030 013306 000000 MINUS4: 0
2031 013310 000000 MINUS3: 0
2032 013312 000000 MINUS2: 0
2033 013314 000000 MINUS1: 0
2034 013316 000000 AVGCNT: 0
2035 013320 000000 PLUS1: 0
2036 013322 000000 PLUS2: 0
2037 013324 000000 PLUS3: 0
2038 013326 000000 PLUS4: 0
2039 013330 000000 PLUS5: 0
2040 013332 000000 ORHIGH: 0
2041 013334 000000 XSPRD1: 0
2042 013336 000000 XSPRD2: 0
2043 013340 000000 XSPRD3: 0
2044 013342 000000 XSPRD4: 0

.SBTTL SCOPE HANDLER ROUTINE

2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075

11*****
11THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS, IT WILL INCREMENT
11AND LOAD THE TEST NUMBER(STSTNM) INTO THE DISPLAY REG.(DISPLAY<7:R>)
11AND LOAD THE ERROR FLAG (SERFLG) INTO DISPLAY<15:00>
11THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
11SW14=1 LOOP ON TEST
11SW11=1 INHIBIT ITERATIONS
11SW09=1 LOOP ON ERROR
11SW08=1 LOOP ON TEST IN SWR<7:0>
11CALL
11 SCOPE 11SCOPE=TOT

SSCOPE:
CKSWR
18: BIT 0BIT14,0SWR 11LOOP ON PRESENT TEST?
RNE 0SWR 11YES IF SW14=1
110000START OF CODE FOR THE XOR TESTER0000
SXTSTRI RR 68 11IF RUNNING ON THE "XOR" TESTER CHANGE
MOV 00ERRVEC,-(SP) 11THIS INSTRUCTION TO A "NOP" (NOP=260)
MOV 050,00ERRVEC 11SAVE THE CONTENTS OF THE ERROR VECTOR
TST 00177060 11SET FOR TIMEOUT
MOV (SP)+,00ERRVEC 11TIME OUT ON XOR?
RR 00VLAD 11RESTORE THE ERROR VECTOR
58: CMP (SP)+,(SP)+ 11GO TO THE NEXT TEST
MOV (SP)+,00ERRVEC 11CLEAR THE STACK AFTER A TIME OUT
RR 78 11RESTORE THE ERROR VECTOR
68110000END OF CODE FOR THE XOR TESTER0000 11LOOP ON THE PRESENT TEST

```

2076 013414 032777 000400 165514      RIT      00IT00,0SWR      ;;LOOP ON SPFC. TEST?
2077 013422 001404                      REG      20              ;;RR IF NO
2078 013424 127737 165506 001102      CMPR     0SWR,STSTNM     ;;ON THE RIGHT TEST? SWR<7IP>
2079 013432 001465                      REG      SOVER          ;;RR IF YES
2080 013434 105737 001103          201      TSTR      SPFLG              ;;HAS AN ERROR OCCURRED?
2081 013440 001421                      REG      30              ;;RR IF NO
2082 013442 123737 001115 001103      CMPR     SERMAX,SERFLG  ;;MAX. ERRORS FOR THIS TEST OCCURRED?
2083 013450 101015                      RMT      30              ;;RR IF NO
2084 013452 032777 001000 165456      BIT      00IT00,0SWR     ;;LOOP ON ERROR?
2085 013460 001404                      REG      40              ;;RR IF NO
2086 013462 013737 001110 001106  701      MOV      SLPERR,SLPADR  ;;SET LOOP ADDRESS TO LAST SCOPE
2087 013470 000446                      RR
2088 013472 105037 001103          401      CLRR     SPFLG              ;;ZERO THE ERROR FLAG
2089 013476 005037 001164                      CLR      STIMES         ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
2090 013502 000415                      RR      10              ;;ESCAPE TO THE NEXT TEST
2091 013504 032777 000400 165424  301      RIT      00IT11,0SWR     ;;INHIBIT ITERATIONS?
2092 013512 001011                      BNE     10              ;;RR IF YES
2093 013514 005737 001206                      TST     SPASS          ;;IF FIRST PASS OF PROGRAM
2094 013520 001406                      REG      10              ;;      INHIBIT ITERATIONS
2095 013522 005237 001104                      INC     SICNT          ;;INCREMENT ITERATION COUNT
2096 013526 023737 001164 001104      CMP     STIMES,SICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
2097 013534 002024                      RGE     SOVER          ;;RR IF MORE ITERATION REQUIRED
2098 013536 012737 000001 001104  101      MOV     01,SICNT       ;;REINITIALIZE THE ITERATION COUNTER
2099 013544 013737 013622 001164      MOV     SMXCNT,STIMES  ;;SET NUMBER OF ITERATIONS TO DO
2100 013552 105237 001102          00VLAD: INCR     STSTNM     ;;COUNT TEST NUMBERS
2101 013556 113737 001102 001204      MOVBR  STSTNM,STSTNM   ;;SET TEST NUMBER IN APT MAILBOX
2102 013564 011637 001106                      MOV     (SP),SLPADR    ;;SAVE SCOPE LOOP ADDRESS
2103 013570 011637 001110                      MOV     (SP),SLPERR    ;;SAVE ERROR LOOP ADDRESS
2104 013574 005037 001166                      CLR     ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
2105 013600 112737 000001 001115      MOVBR  01,SERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
2106 013606 013777 001102 165324  SOVER:  MOV     STSTNM,0DISPLAY  ;;DISPLAY TEST NUMBER
2107 013614 013716 001106                      MOV     SLPADR,(SP)    ;;FUDGE RETURN ADDRESS
2108 013620 000002                      RTT
2109 013622 003720          SMXCNT: 2000.        ;;FIXES PS
                        ;;MAX. NUMBER OF ITERATIONS
    
```

.SBTTL ERROR HANDLER ROUTINE

```

2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
    
```

```

;;*****
;;THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
;;SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
;;AND GO TO SERRTYP ON ERROR
;;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;;SW15=1      HALT ON ERROR
;;SW13=1      INHIBIT ERROR TYPEOUTS
;;SW10=1      BELL ON ERROR
;;SW09=1      LOOP ON ERROR
;;CALL
;;      ERROR  N          ;;ERROR=ENT AND N=ERROR ITEM NUMBER
SERROR:
701      INCR     SERFLG          ;;SET THE ERROR FLAG
        REG      70              ;;DON'T LET THE FLAG GO TO ZERO
        MOV     STSTNM,0DISPLAY  ;;DISPLAY TEST NUMBER AND ERROR FLAG
        RIT     00IT10,0SWR     ;;BELL ON ERROR?
        REG     10              ;;NO = SKIP
        TYPE    ,SBELL          ;;RING BELL
    
```

```

2132 013654 005237 001112      181  INC  SERRTL      ;COUNT THE NUMBER OF ERRORS
2133 013660 011637 001116      MOV  (SP),SERRPC ;GET ADDRESS OF ERROR INSTRUCTION
2134 013664 162737 000002 001116  SUM  02,SERRPC
2135 013672 117737 165220 001114  MOVA 0SFERRPC,SITEMA ;STRIP AND SAVE THE ERROR ITEM CODE
2136 013700 032777 020000 165230  RIT  00IT13,0SWR  ;SKIP TYPEOUT IF SET
2137 013706 001004  ANP  20S          ;SKIP TYPEOUTS
2138 013710 004737 014006  JSR  PC,SERRTYP  ;GO TO USER ERROR ROUTINE
2139 013714 104400 001175  TYPE  ,SCRLF
2140 013720
2141 013720 122737 000001 001270 208:  CMVA 0APTENV,SENV  ;RUNNING IN APT MODE
2142 013726 001007  ANP  2S          ;NO,SKIP APT ERROR REPORT
2143 013730 113737 001114 013742  MOVA  SITEMB,21S  ;SET ITEM NUMBER AS ERROR NUMBER
2144 013736 004737 015704  JSR  PC,0ATY4    ;REPORT FATAL ERROR TO APT
2145 013742 000
2146 013743 000
2147 013744 000777      228:  RR  22S          ;APT ERROR LOOP
2148 013746 005777 165164 281  TST  0SWR        ;HALT ON ERROR
2149 013752 100001  BPL  3S          ;SKIP IF CONTINUE
2150 013754 000000  HALT
2151 013756 032777 001000 165152 381  RIT  00IT09,0SWR ;HALT ON ERROR!
2152 013760 001402  BEQ  4S          ;LOOP ON ERROR SWITCH SET?
2153 013766 013716 001110  MOV  0LPERR,(SP) ;IFUDGE RETURN FOR LOOPING
2154 013772 005737 001166 481  TST  0ESCAPE    ;CHECK FOR AN ESCAPE ADDRESS
2155 013776 001402  BEQ  5S          ;IF NO
2156 014000 013716 001166  MOV  0ESCAPE,(SP) ;IFUDGE RETURN ADDRESS FOR ESCAPE
2157 014004 581
2158 014004 000002  RTI  ;RETURN
2159
2160 .SBTTL ERROR MESSAGE TYPEOUT ROUTINE
2161
2162 ;*****
2163 ;THIS ROUTINE USES THE "ITEM CONTROL BYTE" (SITEMB) TO DETERMINE WHICH
2164 ;ERROR IS TO BE REPORTED, IT THEN OBTAINS, FROM THE "ERROR TABLE" (SERRTB),
2165 ;AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
2166
2167 SERRTYP:
2168 014006 104400 001175  TYPE  ,SCRLF      ;"CARRIAGE RETURN" & "LINE FEED"
2169 014012 010046  MOV  0R,-(SP)    ;SAVE 0R
2170 014014 005000  CLR  0R          ;PICKUP THE ITEM INDEX
2171 014016 153700 001114  RISA 00SITEMB,0R
2172 014022 001004  ANE  1S          ;IF ITEM NUMBER IS ZERO, JUST
2173 ;TYPE THE PC OF THE ERROR
2174 014024 013746 001116  MOV  SERRPC,-(SP) ;SAVE SERRPC FOR TYPEOUT
2175 ;ERROR ADDRESS
2176 014030 104401  TPOC
2177 014032 000426  RR  6S          ;GO TYPE--OCTAL ASCII(ALL DIGITS)
2178 014034 005300 181  DEC  0R          ;GET OUT
2179 014036 006300  ASI  0R          ;ADJUST THE INDEX SO THAT IT WILL
2180 014040 006300  ASI  0R          ; WORK FOR THE ERROR TABLE
2181 014042 006300  ASI  0R
2182 014044 062700 001324  ADD  0SERRTB,0R  ;FORM TABLE POINTER
2183 014050 012037 014060  MOV  (0R)+,2S   ;PICKUP "ERROR MESSAGE" POINTER
2184 014054 001404  BEQ  3S          ;SKIP TYPEOUT IF NO POINTER
2185 014056 104400  TYPE
2186 014060 000000 281  .WORD 0        ;TYPE THE "ERROR MESSAGE"
2187 014062 104400 001175  TYPE  ,SCRLF    ;"ERROR MESSAGE" POINTER GOES HERE
; "CARRIAGE RETURN" & "LINE FEED"

```

```

2188 014066 012037 014076 351 MOV (R0)+,48 ;;PICKUP "DATA HEADER" POINTER
2189 014072 001400 REQ 55 ;;SKIP TYPEOUT IF 0
2190 014074 104400 TYPE ;;TYPE THE "DATA HEADER"
2191 014076 000000 451 .WORD 0 ;;"DATA HEADER" POINTER GOES HERE
2192 014100 104400 001175 TYPE ,SCRLF ;;"CARRIAGE RETURN" & "LINE FEED"
2193 014104 011000 551 MOV (R0),R0 ;;PICKUP "DATA TABLE" POINTER
2194 014106 001000 RNE 75 ;;GO TYPE THE DATA
2195 014110 012600 651 MOV (SP)+,R0 ;;RESTORE R0
2196 014112 104400 001175 TYPE ,SCRLF ;;"CARRIAGE RETURN" & "LINE FEED"
2197 014116 000207 RTS PC ;;RETURN
2198 014120 751
2199 014120 013046 MOV 0(R0)+,-(SP) ;;SAVE 0(R0)+ FOR TYPEOUT
2200 014122 104401 TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2201 014124 005710 TST (R0) ;;IS THERE ANOTHER NUMBER?
2202 014126 001770 BEQ 65 ;;BR IF NO
2203 014130 104400 014136 TYPE ,R5 ;;TYPE TWO(2) SPACES
2204 014134 000771 BR 75 ;;LOOP
2205 014136 020040 000 851 .ASCIZ / / ;;TWO(2) SPACES
2206 .EVEN
2207
2208 .SBTTL POWER DOWN AND UP ROUTINES
2209
2210 ;;*****
2211 ;;POWER DOWN ROUTINE
2212 014142 012737 014306 000025 SPWRDN: MOV 0STILLUP,0SPWRVEC ;;SET FOR FAST UP
2213 014150 012737 000340 000026 MOV 0340,0SPWRVEC+2 ;;PRIORITY
2214 014156 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
2215 014160 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
2216 014162 010246 MOV R2,-(SP) ;;PUSH R2 ON STACK
2217 014164 010346 MOV R3,-(SP) ;;PUSH R3 ON STACK
2218 014166 010446 MOV R4,-(SP) ;;PUSH R4 ON STACK
2219 014170 010546 MOV R5,-(SP) ;;PUSH R5 ON STACK
2220 014172 017746 164740 MOV 0SWR,-(SP) ;;PUSH 0SWR ON STACK
2221 014176 010637 014312 MOV SP,0SAVR6 ;;SAVE SP
2222 014202 012737 014214 000024 MOV 0SPWRUP,0SPWRVEC ;;SET UP VECTOR
2223 014210 000000 HALT
2224 014212 000776 BR -2 ;;HANG UP
2225
2226 ;;*****
2227 ;;POWER UP ROUTINE
2228 014214 012737 014306 000024 SPWRUP: MOV 0STILLUP,0SPWRVEC ;;SET FOR FAST DOWN
2229 014222 013706 014312 MOV 0SAVR6,SP ;;GET SP
2230 014226 005037 014312 CLR 0SAVR6 ;;WAIT LOOP FOR THE TTY
2231 014232 005237 014312 151 INC 0SAVR6 ;;WAIT FOR THE INC
2232 014236 001375 BNE 15 ;;IF WORD
2233 014240 012677 164672 MOV (SP)+,0SWR ;;POP STACK INTO 0SWR
2234 014244 012605 MOV (SP)+,R5 ;;POP STACK INTO R5
2235 014246 012604 MOV (SP)+,R4 ;;POP STACK INTO R4
2236 014250 012603 MOV (SP)+,R3 ;;POP STACK INTO R3
2237 014252 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
2238 014254 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
2239 014256 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
2240 014260 012737 014142 000024 MOV 0SPWRDN,0SPWRVEC ;;SET UP THE POWER DOWN VECTOR
2241 014266 012737 000340 000026 MOV 0340,0SPWRVEC+2 ;;PRIORITY
2242 014274 104400 TYPE ;;REPORT THE POWER FAILURE
2243 014276 014314 SPWRMSG: .WORD 0PWRMSG ;;POWER FAIL MESSAGE POINTER

```

```

2244 014300 012716          MOV      (PC)+,(SP)      ;;RESTART AT BEGIN
2245 014302 001356          SPWRAD: .WORD    R6IN      ;;RESTART ADDRESS
2246 014304 000002          RTI
2247 014306 000000          SILLUP: HALT          ;;THE POWER UP SEQUENCE WAS STARTED
2248 014310 000776          BR      .-2           ;; BEFORE THE POWER DOWN WAS COMPLETE
2249 014312 000000          SSAVE6: 0            ;;PUT THE SP HERE
2250 014314 005015 042522 052123  PWRMSG: .ASCII <15><12>/RESTARTING AFTER A POWER FAILURE/<15><12><12>
2251 014322 051101 044524 043516
2252 014330 040040 052106 051105
2253 014336 040040 050040 053517
2254 014344 051105 043040 044501
2255 014352 052514 042522 005015
2256 014360 000012
2257
2258
2259
2260          .SBYTL  BINARY TO OCTAL (ASCII) AND TYPE
2261
2262          ;;*****
2263          ;;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
2264          ;;OCTAL (ASCII) NUMBER AND TYPE IT.
2265          ;;STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
2266          ;;CALL:
2267          ;;      MOV      NUM,=(SP)          ;;NUMBER TO BE TYPED
2268          ;;      TYPOS          ;;CALL FOR TYPEOUT
2269          ;;      .BYTE  N            ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
2270          ;;      .BYTE  M            ;;M=1 OR 0
2271          ;;
2272          ;;          ;;1=TYPE LEADING ZEROS
2273          ;;          ;;0=SUPPRESS LEADING ZEROS
2274
2275          ;;STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
2276          ;;STYPOS OR STYPOC
2277          ;;CALL:
2278          ;;      MOV      NUM,=(SP)          ;;NUMBER TO BE TYPED
2279          ;;      TYPON          ;;CALL FOR TYPEOUT
2280
2281          ;;STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
2282          ;;CALL:
2283          ;;      MOV      NUM,=(SP)          ;;NUMBER TO BE TYPED
2284          ;;      TYPOC          ;;CALL FOR TYPEOUT
2285 014362 017646 000000          STYPOS: MOV      0(SP),-(SP)      ;;PICKUP THE MODE
2286 014366 116637 000001 014605          MOVR    1(SP),S0FILL      ;;LOAD ZERO FILL SWITCH
2287 014374 112637 014607          MOVR    (SP)+,S0MODE+1    ;;NUMBER OF DIGITS TO TYPE
2288 014400 062716 000002          ADD     02,(SP)          ;;ADJUST RETURN ADDRESS
2289 014404 000406          RR      STYPON
2290 014406 112737 000001 014605          STYPOC: MOVR    01,S0FILL      ;;SET THE ZERO FILL SWITCH
2291 014414 112737 000006 014607          MOVR    06,S0MODE+1      ;;SET FOR SIX(6) DIGITS
2292 014422 112737 000005 014604          STYPON: MOVR    05,S0CNT      ;;SET THE ITERATION COUNT
2293 014430 010346          MOV     R3,=(SP)          ;;SAVE R3
2294 014432 010446          MOV     R4,=(SP)          ;;SAVE R4
2295 014434 010546          MOV     R5,=(SP)          ;;SAVE R5
2296 014436 113704 014607          MOVR    S0MODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
2297 014442 005404          NEG     R4
2298 014444 062704 000006          ADD     06,R4            ;;SUBTRACT IT FOR MAX. ALLOWED
2299 014450 110437 014606          MOVR    R4,S0MODEF      ;;SAVE IT FOR USE

```


2300	014454	113704	014605	MOV8	80FILL,R4	;;GET THE ZERO FILL SWITCH
2301	014460	016605	000012	MOV	12(SP),R5	;;PICKUP THE INPUT NUMBER
2302	014464	005003		CLR	R3	;;CLEAR THE OUTPUT WORD
2303	014466	006105		181 ROL	R5	;;ROTATE MSB INTO "C"
2304	014470	000404		RR	30	;;GO DO MSR
2305	014472	006105		281 ROL	R5	;;FORN THIS DIGIT
2306	014474	006105		ROL	R5	
2307	014476	006105		ROL	R5	
2308	014500	010503		MOV	R5,R3	
2309	014502	006103		381 ROL	R3	;;GET LSR OF THIS DIGIT
2310	014504	105337	014606	DECB	SOMODE	;;TYPE THIS DIGIT?
2311	014510	100016		RPL	70	;;RR IF NO
2312	014512	042703	177770	BIC	0177770,R3	;;GET RID OF JUNK
2313	014516	001002		BNE	00	;;TEST FOR 0
2314	014520	005704		TST	R4	;;SUPPRESS THIS R?
2315	014522	001403		REQ	50	;;RR IF YES
2316	014524	005204		481 INC	R4	;;DON'T SUPPRESS ANYMORE 0'S
2317	014526	052703	000060	BIS	0'0,R3	;;MAKE THIS DIGIT ASCII
2318	014532	052703	000040	581 BIS	0' ,R3	;;MAKE ASCII IF NOT ALREADY
2319	014536	110337	014602	MOV8	R3,00	;;SAVE FOR TYPING
2320	014542	104400	014602	TYPE	,R4	;;GO TYPE THIS DIGIT
2321	014546	105337	014604	781 DECB	SOCNT	;;COUNT BY 1
2322	014552	003347		BGT	70	;;RR IF MORE TO DO
2323	014554	002402		BLT	60	;;RR IF DONE
2324	014556	005204		INC	R4	;;INSURE LAST DIGIT ISN'T A BLANK
2325	014560	000744		RR	20	;;GO DO THE LAST DIGIT
2326	014562	012605		681 MOV	(SP)+,R5	;;RESTORE R5
2327	014564	012604		MOV	(SP)+,R4	;;RESTORE R4
2328	014566	012603		MOV	(SP)+,R3	;;RESTORE R3
2329	014570	016666	000002 000004	MOV	2(SP),4(SP)	;;SET THE STACK FOR RETURNING
2330	014576	012616		MOV	(SP)+,(SP)	
2331	014600	000002		RTT		;;RETURN
2332	014602	000		881 .BYTE	0	;;STORAGE FOR ASCII DIGIT
2333	014603	000		.BYTE	0	;;TERMINATOR FOR TYPE ROUTINE
2334	014604	000		SOCNT: .BYTE	0	;;OCTAL DIGIT COUNTER
2335	014605	000		S0FILL: .BYTE	0	;;ZERO FILL SWITCH
2336	014606	000000		SOMODE: .WORD	0	;;NUMBER OF DIGITS TO TYPE
2337						
2338				.SBTTL	TYPE ROUTINE	
2339						
2340						
2341				;;*****		
2342				;;ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.		
2343				;;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.		
2344				;;NOTE1: SNUL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.		
2345				;;NOTE2: SFILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.		
2346				;;NOTE3: SFILLC CONTAINS THE CHARACTER TO FILL AFTER.		
2347				;;		
2348				;;CALL:		
2349				;;1) USING A TRAP INSTRUCTION		
2350				;; TYPE ,MESADR		;;MESADR IS FIRST ADDRESS OF AN ASCII STRING
2351				;;OR		
2352				;; TYPE		
2353				;; MESADR		
2354				;;		
2355	014610	105737	001155	STYPE: TSTB	STPFLG	;;IS THERE A TERMINAL?

2356	R14614	1000R2			RPL	118		11BR IF YES
2357	R14616	R00000			HALT			11HALT HERE IF NO TERMINAL
2358	R14620	R00030			RR	38		11LEAVE
2359	R14622	R10046		181	MOV	RR,-(SP)		11SAVE RR
2360	R14624	R17600	R00002		MOV	R2(SP),RR		11GET ADDRESS OF ASCII STRING
2361	R14630	122737	R00001	R01220	CMPR	RPTENV,SENV		11RUNNING IN APT MODE
2362	R14636	R01011			BNE	628		11NO,GO CHECK FOR APT CONSOLE
2363	R14640	132737	R00100	R01221	RITR	RPTSP00L,SENVH		11SPOOL MESSAGE TO APT
2364	R14646	R01405			REQ	628		11NO,GO CHECK FOR CONSOLE
2365	R14650	R10037	R14660		MOV	RR,618		11SETUP MESSAGE ADDRESS FOR APT
2366	R14654	R004737	R15674		JBR	PC,SATV3		11SPOOL MESSAGE TO APT
2367	R14660	R00000		6181	.WORD	R		11MESSAGE ADDRESS
2368	R14662	132737	R00040	R01221	6281	RITR	RPTCSUP,SENVH	11APT CONSOLE SUPPRESSED
2369	R14670	R01003			BNE	608		11YES,SKIP TYPE OUT
2370	R14672	112046		281	MOVR	(R0)+,-(SP)		11PUSH CHARACTER TO BE TYPED ONTO STACK
2371	R14674	R01005			RNE	AS		11BR IF IT ISN'T THE TERMINATOR
2372	R14676	R05726			TST	(SP)+		11IF TERMINATOR POP IT OFF THE STACK
2373	R14700	R12600		6081	MOV	(SP)+,R0		11RESTORE R0
2374	R14702	R02716	R00002		381	ADD	R2,(SP)	11ADJUST RETURN PC
2375	R14706	R00002			RTR			11RETURN
2376	R14710	122716	R00011		481	CMPS	RMT,(SP)	11BRANCH IF <MT>
2377	R14714	R01430			REQ	88		
2378	R14716	122716	R00200		CMPS	RCLRF,(SP)		11BRANCH IF NOT <CLRF>
2379	R14722	R01006			BNE	58		
2380	R14724	R05726			TST	(SP)+		11POP <CR><LF> EQUIV
2381	R14726	104400			TYPE			11TYPE A CR AND LF
2382	R14730	R01175			RCLRF			
2383	R14732	105037	R15066		CLRB	SCHARCNT		11CLEAR CHARACTER COUNT
2384	R14736	R00755			BR	28		11GET NEXT CHARACTER
2385	R14740	R004737	R15022		581	JBR	PC,STYPEC	11GO TYPE THIS CHARACTER
2386	R14744	123726	R01154		681	CMPS	RFILLC,(SP)+	11IS IT TIME FOR FILLER CHARS.?
2387	R14750	R01350			BNE	28		11IF NO GO GET NEXT CHAR.
2388	R14752	R013746	R01152		MOV	RNULL,-(SP)		11GET # OF FILLER CHARS. NEEDED
2389								11AND THE NULL CHAR.
2390	R14756	105366	R00001		781	DECR	1(SP)	11DOES A NULL NEED TO BE TYPED?
2391	R14762	R02770			BLT	68		11BR IF NO--GO POP THE NULL OFF OF STACK
2392	R14764	R004737	R15022		JBR	PC,STYPEC		11GO TYPE A NULL
2393	R14770	105337	R15066		DECB	SCHARCNT		11DO NOT COUNT AS A COUNT
2394	R14774	R000770			BR	78		11LOOP
2395								
2396								
2397								
2398	R14776	112716	R00040		881	MOVB	R' ,(SP)	11REPLACE TAB WITH SPACE
2399	R15002	R004737	R15022		981	JBR	PC,STYPEC	11TYPE A SPACE
2400	R15006	132737	R00007	R15066,		BITB	R7,SCHARCNT	11BRANCH IF NOT AT
2401	R15014	R01372			RNE	98		11TAB STOP
2402	R15016	R05726			TST	(SP)+		11POP SPACE OFF STACK
2403	R15020	R00724			RR	28		11GET NEXT CHARACTER
2404	R15022	105777	164120		STYPEC:	RSTPB		11WAIT UNTIL PRINTER IS READY
2405	R15026	100375			BPL	STYPEC		
2406	R15030	116677	R00002	164112	MOVR	R2(SP),RSTPB		11LOAD CHAR TO BE TYPED INTO DATA REG.
2407	R15036	122766	R00015	R00002	CMPR	RCR,2(SP)		11IS CHARACTER A CARRIAGE RETURN?
2408	R15044	R01003			BNE	18		11BRANCH IF NO
2409	R15046	105037	R15066		CLRB	SCHARCNT		11YES--CLEAR CHARACTER COUNT
2410	R15052	R000406			BR	STYPEC		11EXIT
2411	R15054	122766	R00012	R00002	181	CMPS	RLF,2(SP)	11IS CHARACTER A LINE FEED?

11HORIZONTAL TAB PROCESSOR

DZARRB.P11 TYPE ROUTINE

2412	015262	001402			REF	STYPEX	;;BRANCH IF YES
2413	015264	105227			INCR	(PC)+	;;COUNT THE CHARACTER
2414	015266	000000			SEARCHNT:WORD	R	;;CHARACTER COUNT STORAGE
2415	015270	000207			STYPEX:RTS	PC	
2416							
2417							
2418							
2419							
2420							
2421							
2422							
2423							
2424							
2425	015072	022737	000176	001136	SCKSWR: CMP	0SWREG,SWR	;;IS THE SOFT-SWR SELECTED?
2426	015100	001073			RNF	100	;;BRANCH IF NO
2427	015102	105777	164034		TSTR	0STKS	;;CHAR THERE?
2428	015106	100070			BPI	100	;;IF NO, DON'T WAIT AROUND
2429	015110	117746	164030		201: MOVR	0STKR,-(SP)	;;SAVE THE CHAR
2430	015114	042716	177600		RIC	0177,(SP)	;;STRIP-OFF THE ASCII
2431	015120	022726	000007		CMR	07,(SP)+	;;IS IT A CONTROL GT
2432	015124	001061			RNF	100	;;NO, RETURN TO USER
2433	015126	104400	015535		TYPE	,0CNTLG	;;YES, ECHO CONTROL G
2434							
2435	015132	104400	015542		601: TYPE	,0MSWR	;;TYPE CURRENT CONTENTS
2436	015136	013746	000176		MOV	SWREG,-(SP)	;;SAVE SWREG FOR TYPEOUT
2437	015142	104401			TYPOC		;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2438	015144	104400	015553		TYPE	,0MNEA	;;PROMPT FOR NEW SWR
2439	015150	005046			CLR	-(SP)	;;CLEAR COUNTER
2440	015152	005046			CLR	-(SP)	;;THE NEW SWR
2441	015154	104406			701: RDCMR		;;GET NEXT CHAR
2442							
2443	015156	022716	000025		801: CMP	025,(SP)	;;IS IT A CONTROL UP
2444	015162	001005			BNE	00	;;BRANCH IF NO
2445	015164	104400	015530		TYPE	,0CNTLU	;;YES, ECHO IT
2446	015170	062706	000006		ADD	06,SP	;;IGNORE PREVIOUS INPUT
2447	015174	000756			RR	05	;;LET'S TRY IT AGAIN
2448							
2449	015176	022716	000015		901: CMP	015,(SP)	;;IS IT A <CR>?
2450	015202	001011			RNE	110	;;BRANCH IF NO
2451	015204	005766	000004		TST	4(SP)	;;YES, IS IT THE FIRST CHAR?
2452	015210	001403			BEQ	100	;;BRANCH IF YES
2453	015217	016677	000002	163716	MOV	2(SP),0SWR	;;SAVE NEW SWR
2454	015220	062706	000006		1001: ADD	06,SP	;;CLEAR UP STACK
2455	015224	000417			RR	170	;;RETURN TO USER
2456	015226	022716	000012		1101: CMP	012,(SP)	;;IS IT A <LF>
2457	015232	001017			RNE	150	;;BRANCH IF NO
2458	015234	005766	000004		TST	4(SP)	;;YES, IS IT THE FIRST CHAR?
2459	015240	001403			BEQ	120	;;YES
2460	015242	016677	000002	163666	MOV	2(SP),0SWR	;;SAVE NEW SWR
2461	015250	062706	000006		1201: ADD	06,SP	;;CLEAR UP STACK
2462	015254	013716	000046		MOV	0046,(SP)	;;GET RESTART
2463	015260	062716	000010		ADD	010,(SP)	;;ADDRESS
2464	015264	104400	001175		1301: TYPE	,0CRLF	;;ECHO <CR> AND <LF>
2465	015270	000002			1401: RTI		;;RETURN
2466	015272	004737	015022		1501: JSR	PC,STPEC	;;ECHO CHAR
2467	015276	042726	177770		RIC	0177770,(SP)+	;;RESTRICT TO R-7

```

2468 015302 005766 000002          TST      2(SP)          ;;IS THIS THE FIRST CHAR
2469 015306 001403          REQ      163          ;;BRANCH IF YES
2470 015310 006316          ASL      (SP)          ;;NO, SHIFT PRESENT
2471 015312 006316          ASL      (SP)          ;; CHAR OVER TO MAKE
2472 015314 006316          ASL      (SP)          ;; ROOM FOR NEW ONE.
2473 015316 005266 000002          160: INC      2(SP)          ;;KEEP COUNT OF CHAR
2474 015322 056616 177776          BIS      -2(SP),(SP)  ;;SET IN NEW CHAR
2475 015326 000712          RR       78           ;;GET THE NEXT ONE
2476                                     ;;*****
2477                                     ;;THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
2478                                     ;;CALL:
2479                                     ;; RDCHR          ;;INPUT A SINGLE CHARACTER FROM THE TTY
2480                                     ;; RETURN HERE    ;;CHARACTER IS ON THE STACK
2481                                     ;;               ;;WITH PARITY BIT STRIPPED OFF
2482                                     ;;
2483
2484 015330 011646          SRDCHR: MOV      (SP),-(SP)  ;;PUSH DOWN THE PC
2485 015332 016666 000004 000002          18:  MOV      4(SP),2(SP)  ;;SAVE THE PS
2486 015340 105777 163576          TSTB    0BTKS          ;;WAIT FOR
2487 015344 100375          BPL     15           ;;A CHARACTER
2488 015346 117766 163572 000004          MOVB    0BTKR,4(SP)    ;;READ THE TTY
2489 015354 042766 177600 000004          BIC     0^C<177>,4(SP) ;;GET RID OF JUNK IF ANY
2490 015362 026627 000004 000140          CMP     4(SP),0140    ;;IS IT UPPER CASE?
2491 015370 002407          BLY     25           ;;BRANCH IF YES
2492 015372 026627 000004 000175          CMP     4(SP),0175    ;;IS IT A SPECIAL CHAR?
2493 015400 003003          BGT     29           ;;BRANCH IF YES
2494 015402 042766 000040 000004          BIC     040,4(SP)     ;;MAKE IT UPPER CASE
2495 015410 000002          28:  RTI          ;;GO BACK TO USER
2496                                     ;;*****
2497                                     ;;THIS ROUTINE WILL INPUT A STRING FROM THE TTY
2498                                     ;;CALL:
2499                                     ;; RDLIN          ;;INPUT A STRING FROM THE TTY
2500                                     ;; RETURN HERE    ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
2501                                     ;;               ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
2502
2503 015412 010346          SRDLIN: MOV      R3,-(SP)  ;;SAVE R3
2504 015414 012703 015520          18:  MOV      0STTYIN,R3  ;;GET ADDRESS
2505 015420 022703 015530          28:  CMP      0STTYIN+0,,R3 ;;BUFFER FULL?
2506 015424 101405          RLOS    45           ;;BR IF YES
2507 015426 104406          RDCHR          ;;GO READ ONE CHARACTER FROM THE TTY
2508 015430 112613          MOVB    (SP)+,(R3)    ;;GET CHARACTER
2509 015432 122713 000177          108:  CMPB    0177,(R3)    ;;IS IT A RUBOUT
2510 015436 001003          RNE     39           ;;SKIP IF NOT
2511 015440 104400 001174          48:  TYPE    ,QUES      ;;TYPE A '9'
2512 015444 000763          BR      15           ;;CLEAR THE BUFFER AND LOOP
2513 015446 111337 015516          38:  MOVB    (R3),98     ;;ECHO THE CHARACTER
2514 015452 104400 015516          TYPE    ,98
2515 015456 122723 000015          CMPB    015,(R3)+    ;;CHECK FOR RETURN
2516 015462 001356          RNE     28           ;;LOOP IF NOT RETURN
2517 015464 105063 177777          CLRB   -1(R3)        ;;CLEAR RETURN (THE 15)
2518 015470 104400 001176          TYPE    ,SLF        ;;TYPE A LINE FEED
2519 015474 012603          MOV     (SP)+,R3     ;;RESTORE R3
2520 015476 011646          MOV     (SP),-(SP)   ;;ADJUST THE STACK AND PUT ADDRESS OF THE
2521 015500 016666 000004 000002          MOV     4(SP),2(SP)  ;; FIRST ASCII CHARACTER ON IT
2522 015506 012766 015520 000004          MOV     0STTYIN,4(SP)
2523 015514 000002          RTI          ;;RETURN

```

```

2524 015516 000 981 .BYTE 0 ; STORAGE FOR ASCII CHAR. TO TYPE
2525 015517 000 .BYTE 0 ; TERMINATOR
2526 015520 000010 STTVINI .BLKB 0 ; RESERVE 8 BYTES FOR TTY INPUT
2527 015530 052536 005015 000 SCNTLUI .ASCIZ /U/<15><12> ; CONTROL "U"
2528 015535 136 006507 000012 SCNTLGI .ASCIZ /G/<15><12> ; CONTROL "G"
2529 015542 005015 053523 020122 SMSWR1 .ASCIZ <15><12>/SWR = /
2530 015550 020075 000
2531 015553 040 047040 053505 SMNEW1 .ASCIZ / NEW = /
2532 015560 036440 000040
2533
2534 .SBTTL READ AN OCTAL NUMBER FROM THE TTY
2535
2536 ;*****
2537 ; THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
2538 ; CHANGE IT TO BINARY.
2539 ; CALL:
2540 ; RDOCT ; READ AN OCTAL NUMBER
2541 ; RETURN HERE ; LOW ORDER BITS ARE ON TOP OF THE STACK
2542 ; ; HIGH ORDER BITS ARE IN SHIOCT
2543
2544 015564 011646 000004 000002 SRDOCT: MOV (SP),-(SP) ; PROVIDE SPACE FOR THE
2545 015566 016666 000004 000002 MOV 4(SP),2(SP) ; INPUT NUMBER
2546 015574 010046 MOV R0,-(SP) ; PUSH R0 ON STACK
2547 015576 010146 MOV R1,-(SP) ; PUSH R1 ON STACK
2548 015600 010246 MOV R2,-(SP) ; PUSH R2 ON STACK
2549 015602 104407 181 RDLIN ; READ AN ASCII LINE
2550 015604 012600 MOV (SP)+,R0 ; GET ADDRESS OF 1ST CHARACTER
2551 015606 005001 CLR R1 ; CLEAR DATA WORD
2552 015610 005002 CLR R2
2553 015612 112046 281 MOVB (R0)+,-(SP) ; PICKUP THIS CHARACTER
2554 015614 001012 BEQ 35 ; IF ZERO GET OUT
2555 015616 006301 ASL R1 ; *2
2556 015620 006102 ROL R2
2557 015622 006301 ASL R1 ; *4
2558 015624 006102 ROL R2
2559 015626 006301 ASL R1 ; *8
2560 015630 006102 ROL R2
2561 015632 042716 177770 BIC 0^C7,(SP) ; STRIP THE ASCII JUNK
2562 015636 062601 ADD (SP)+,R1 ; ADD IN THIS DIGIT
2563 015640 000764 RR 25 ; LOOP
2564 015642 005726 381 TST (SP)+ ; CLEAN TERMINATOR FROM STACK
2565 015644 010166 000012 MOV R1,12(SP) ; SAVE THE RESULT
2566 015650 010237 015664 MOV R2,SHIOCT
2567 015654 012602 MOV (SP)+,R2 ; POP STACK INTO R2
2568 015656 012601 MOV (SP)+,R1 ; POP STACK INTO R1
2569 015660 012600 MOV (SP)+,R0 ; POP STACK INTO R0
2570 015662 000002 RTI ; RETURN
2571 015664 000000 SHIOCT: .WORD 0 ; HIGH ORDER BITS GO HERE
2572
2573 .SBTTL APT COMMUNICATIONS ROUTINE
2574
2575 ;*****
2576 015666 112737 000001 016132 SATY1: MOVB 01,SFFLG ; TO REPORT FATAL ERROR
2577 015674 112737 000001 016130 SATY3: MOVB 01,SMPLG ; TO TYPE A MESSAGE
2578 015702 000403 RR SATVC
2579 015704 112737 000001 016132 SATY4: MOVB 01,SFFLG ; TO ONLY REPORT FATAL ERROR

```

```

2500 015712          SATVCI
2501 015712 010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
2502 015714 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
2503 015716 105737 016130  TSTB    SMFLG        ;;SHOULD TYPE A MESSAGE?
2504 015722 001450      REG      55          ;;IF NOT: BR
2505 015724 122737 000001 001220  CMPR    0APTENV,SENV  ;;OPERATING UNDER APT?
2506 015732 001031      BNE     34          ;;IF NOT: BR
2507 015734 132737 000100 001221  RTR     0APTSPool,SENVM ;;SHOULD SPOOL MESSAGE?
2508 015742 001425      REG      35          ;;IF NOT: BR
2509 015744 017600 000004      MOV      04(SP),R0    ;;GET MESSAGE ADDR.
2590 015750 062766 000002 000004  ADD     02,4(SP)      ;;RUMP RETURN ADDR.
2591 015756 005737 001200 131  TST     MSGTYPE      ;;SEE IF DONE W/ LAST XMISSION?
2592 015762 001375      BNE     15          ;;IF NOT: WAIT
2593 015764 010037 001214      MOV      R0,MSGADR    ;;PUT ADDR IN MAILBOX
2594 015770 105720 281  TSTB   (R0)+         ;;FIND END OF MESSAGE
2595 015772 001376      RNE     25          ;;SUB START OF MESSAGE
2596 015774 163700 001214      SUR     MSGADR,R0     ;;GET MESSAGE LGTH IN WORDS
2597 016000 006200      ASR    R0            ;;PUT LENGTH IN MAILBOX
2598 016002 010037 001216      MOV      R0,MSGGLGT   ;;TELL APT TO TAKE MSG.
2599 016006 012737 000004 001200  MOV     04,MSGTYPE
2600 016014 000413      BR     55
2601 016016 017637 000004 016042 331  MOV     04(SP),05     ;;PUT MSG ADDR IN JSR LINKAGE
2602 016024 062766 000002 000004  ADD     02,4(SP)      ;;RUMP RETURN ADDRESS
2603 016032 013746 177776      MOV     177776,-(SP)  ;;PUSH 177776 ON STACK
2604 016036 004737 014610      JSR    PC,STYPE      ;;CALL TYPE MACRO
2605 016042 000000 431  .WORD  0
2606 016044 531
2607 016044 105737 016132 1001  TSTB   SFPLG        ;;SHOULD REPORT FATAL ERROR?
2608 016050 001414      BEQ    120         ;;IF NOT: BR
2609 016052 005737 001220      TST    SENV        ;;RUNNING UNDER APT?
2610 016056 001413      REG    120        ;;IF NOT: BR
2611 016060 005737 001200 1131  TST    MSGTYPE      ;;FINISHED LAST MESSAGE?
2612 016064 001375      RNE    118        ;;IF NOT: WAIT
2613 016066 017637 000004 001202  MOV     04(SP),SFATAL ;;GET ERROR #
2614 016074 062766 000002 000004  ADD     02,4(SP)      ;;RUMP RETURN ADDR.
2615 016102 005237 001200      INC    MSGTYPE      ;;TELL APT TO TAKE ERROR
2616 016106 105037 016132 1201  CLRR   SFPLG        ;;CLEAR FATAL FLAG
2617 016112 105037 016131      CLRR   SLPLG        ;;CLEAR LOG FLAG
2618 016116 105037 016130      CLRR   SMPLG        ;;CLEAR MESSAGE FLAG
2619 016122 012601      MOV    (SP)+,R1     ;;POP STACK INTO R1
2620 016124 012600      MOV    (SP)+,R0     ;;POP STACK INTO R0
2621 016126 000207      RTS    PC           ;;RETURN
2622 016130 000      SMPLG: .BYTE 0      ;;MSG. FLAG
2623 016131 000      SLPLG: .BYTE 0      ;;LOG FLAG
2624 016132 000      SFPLG: .BYTE 0      ;;FATAL FLAG
2625 016134 .EVEN
2626 000200  APTSIZE=200
2627 000001  APTENV=001
2628 000100  APTSPool=100
2629 000040  APTCSUP=040
2630
2631 .SBTTL  TRAP DECODER
2632
2633 ;;*****
2634 ;;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
2635 ;;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS

```

2636 ; OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
 2637 ; GO TO THAT ROUTINE.

```

2638 STRAP: MOV RR,=(SP) ;SAVE RR
2639 MOV 2(SP),RR ;GET TRAP ADDRESS
2640 TST -(RR) ;BACKUP RV 2
2641 MOVR (RR),RR ;GET RIGHT BYTE OF TRAP
2642 ASL RR ;POSITION FOR INDEXING
2643 MOV STRPAD(RR),RR ;INDEX TO TABLE
2644 RTS RR ;GO TO ROUTINE
  
```

.SBTTL TRAP TABLE

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

;	ROUTINE	;
2653	STRPAD:	;
2654	STVPE ;CALL=TYPE	TRAP+0(104400) TTY TYPEOUT ROUTINE
2655	STVPOC ;CALL=TYPOC	TRAP+1(104401) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
2656	STVPOS ;CALL=TYPOS	TRAP+2(104402) TYPE OCTAL NUMBER (NO LEADING ZEROS)
2657	STVPON ;CALL=TYPON	TRAP+3(104403) TYPE OCTAL NUMBER (AS PER LAST CALL)
2658	STVPOS ;CALL=TYPOS	TRAP+4(104404) TYPE DECIMAL NUMBER (WITH SIGN)
2659	CKSWR ;CALL=CKSWR	TRAP+5(104405) TEST FOR CHANGE IN SOFT-SWR
2660	BRDCHR ;CALL=BRDCHR	TRAP+6(104406) TTY TYPEIN CHARACTER ROUTINE
2661	BRDLIN ;CALL=BRDLIN	TRAP+7(104407) TTY TYPEIN STRING ROUTINE
2662	BRDOCT ;CALL=BRDOCT	TRAP+1P(104410) READ AN OCTAL NUMBER FROM TTY

```

2665 ADDBUFF: 0
2666 ;.+.40
2667 ADTB0: 0
2668 ;.+.40
2669 ADTB1: 0
2670 ;.+.40
2671 ADTB2: 0
2672 ;.+.40
2673 ADTB3: 0
2674 ;.+.40
2675 ADTB4: 0
2676 ;.+.40
2677 ADTB5: 0
2678 ;.+.40
2679 ADTB6: 0
2680 ;.+.40
2681 ADTB7: 0
2682 ;.+.40
2683 ;.+.40
2684 ;.END
  
```

A	003550	600	7000	050	060															
ABASE	0170400	1230	201	205																
ACDW1	0000000	201	207																	
ACDW2	0000000	201	208																	
ACPUOP	0000000	201	256																	
ACRLF	012356	1042	1157	1206	1292	1310	1625	10070												
ADBUFF	016200	1061	1130	1371	1385	1407	1400	1682	1755	1000	26660									
ADCNVY	010620	1250	1364	1369	16000															
ADCS	001336	3360	034	0620	10650	11910	11030	1440	16890	16930	1695									
ADCS1	010752	0300	0390	16850	17100															
ADDBR	001340	3370	040	1195	1697															
ADDW0	0000000	201	209																	
ADDW1	0000000	201	290																	
ADDW10	0000000	201	299																	
ADDW11	0000000	201	300																	
ADDW12	0000000	201	301																	
ADDW13	0000000	201	302																	
ADDW14	0000000	201	303																	
ADDW15	0000000	201	304																	
ADDW2	0000000	201	291																	
ADDW3	0000000	201	292																	
ADDW4	0000000	201	293																	
ADDW5	0000000	201	294																	
ADDW6	0000000	201	295																	
ADDW7	0000000	201	296																	
ADDW8	0000000	201	297																	
ADDW9	0000000	201	298																	
ADEVCT	0000000	201	247																	
ADEVH	0000000	201	206																	
ADMIGH	013250	1307	17500	1761	17630	20150														
ADLOW	013252	1290	17500	1764	17660	20160														
ADTB0	016202	13720	1012	26680																
ADTB1	016304	13730	1013	26700																
ADTB2	016306	13740	1014	26720																
ADTB3	016410	13750	1015	26740																
ADTB4	016452	13760	1016	26760																
ADTB5	016514	13770	1017	26780																
ADTB6	016556	13780	1018	26800																
ADTB7	016620	13790	1019	26820																
ADWRD1	006224	10070	10510	10550	10580	1191	12090													
ADWRD2	013146	10050	10460	1065	12550	12560	12570	13600	13610	13620	13650	13660	13670	1605						
		1609	1693	19820																
AENV	0000000	201	252																	
AENVH	0000000	201	253																	
AFATAL	0000000	201	240																	
AMADR1	0000000	201	260																	
AMADR2	0000000	201	273																	
AMADR3	0000000	201	276																	
AMADR4	0000000	201	279																	
AMANS1	0000000	201	263																	
AMANS2	0000000	201	271																	
AMANS3	0000000	201	278																	
AMANS4	0000000	201	277																	
AMSGAD	0000000	201	200																	
AMSGLG	0000000	201	250																	
AMSGTY	0000000	201	203																	

COUNT	013160	1044	1254	1359	1601	1947						
CPDLAY	010736	410	1703									
CPTIME	010746	410	1686	1700								
CPTYPE	010750	415	416	1013	1709							
CR	000015	220	2407	2417								
CRLF	000200	230	2378	2417								
C80	001344	3400	1189									
C8C	001354	3460	1432									
C8R	001342	3390	463	1190								
D	003573	7500										
DDISP	0177570	290	217	377								
DECOUT	011052	1724	1730									
DECPNT	011140	1717	1721	1723	1728	1745						
DECPRT	010754	1321	1715									
DEC1	011100	1731	1733	1736								
DEC2	011114	1735	1738									
DELAY	013216	1363	1368	1603	1700	2022						
DELAY1	013220	1686	1687	2003								
DGT0	006060	1160	1170									
DGT1	006062	1171										
DGT2	006064	1172										
DGT3	006066	1173										
DGT4	006070	1107	1174									
DH1	013075	329	1073									
DIGCNT	011134	1716	1725	1732	1743							
DIGIT	011132	1719	1720	1730	1734	1737	1740	1742				
DISPLA	001140	2170	377	383	2106	2128						
DISPRE	000174	1400	303									
DOT	012361	1900										
OSHR	0177570	200	216	376								
OT1	013134	330	1900									
E	003600	7520	862	871								
EMTVEC	000030	1170	361	362								
EM1	013032	320	1967									
ERRVEC	000004	1100	374	375	386	2067	2068	2070	2073			
F	003605	7540										
FILE1	011352	1705	1708									
FILE2	011410	1703	1796									
FIRST	013214	2001										
G	003612	7560										
GEN1	003366	7010	703									
GETDAT	011216	1760	1770									
GNS	000000 U	147	2656	2657	2658	2659	2660	2661	2662	2663	2664	
H	003617	7500	858	867								
HIGH	013202	592	615	1996								
HIORDV	013246	1756	1760	1772	2014							
HT	000011	200	2376	2417								
I	003624	7600										
INBUF	010436	466	469	472	475	478	1040	1579	1582	1583	1631	143A
INCR	013204	1907										
INITA	002072	456	458	1984								
INIT1	002042	419	440									
INIT2	002076	450	450	483	1611							
IOTVEC	000020	1150	359	360								
J	003631	7620										
K	003636	7640										

KSTOR1	013162	12400	1241	1244	1252	12530	1360	19800		
KSTOR2	013164	10480	1155	12410	12460	1229	13540	1365	1421	19890
KSTOR3	013166	12510	1265	1270	1276	1281	19900			
KSTOR4	013170	12520	1255	1293	1227	12310	19910			
KSTOR5	013172	19920								
KSTR11	013174	19930								
KSTR12	013176	19940								
L	003643	7660	863	872						
LADTB	007414	1389	10120							
LDTRAP	001740	4230	448	459						
LF	000012	210	2411	2417						
LOADVC	004626	845	9590							
LODPNT	003154	649	651	6600						
LODVCA	004664	9620	973							
LODVCB	004672	9640	970							
LODVCC	004676	963	9650							
LOW	013200	5420	550	551	9910	400	616	19950		
M	003650	7680								
MARKER	006222	10850	10900	1184	12080					
MESPRY	013244	12340	1287	1310	12140	20130				
MES10	012513	1030	19270							
MES13	012561	1233	19340							
MES14	012617	1237	1350	19400						
MES15	012633	489	19430							
MES16	012701	1240	19500							
MES19	012720	1290	19530							
MES2	011772	1427	18610							
MES20	012743	1313	19570							
MES3	012023	940	18660							
MES4	012070	457	18730							
MES6	012370	487	19110							
MES7	012420	1036	19150							
MES8	012454	1340	19200							
MES9	012505	1400	19250							
MINUS1	013314	1833	20330							
MINUS2	013312	1836	20320							
MINUS3	013310	1839	20310							
MINUS4	013306	1842	20300							
MINUS5	013304	18220	20290							
MODE	003542	714	7350							
N	003655	685	7700	860	861	869	870			
NBEXT	001334	3330								
N0	003756	690	7960	1175						
N1	003763	7980	865	1176						
N2	003770	8000	874	1177						
N3	003775	8020	1178							
N4	004002	8040	1179							
N5	004007	8060	1180							
N6	004014	8080	1181							
N7	004021	8100	1182							
N8	004026	8120								
N9	004033	8140								
O	003662	7720								
OPS1	013224	12350	20050							
ORHIGH	013332	1802	18240	20400						
ORLOW	013302	1319	1800	18260	20200					

OVRMI	011544	1010	10240																
OVRLO	011552	1012	10260																
P	003667	7740																	
PBB	002370	501	514	5190															
PC	0000007	410	4460	4480	4500	4650	5000	5010	5020	5130	5140	5150	5310	5450					
		5040	5950	6240	6330	6560	6770	6820	6870	6920	6930	7010	7040	7330					
		8300	8380	8480	8420	8040	8850	8860	8980	9080	9110	9210	9260	9530					
		9740	9830	9890	10180	10390	10860	10910	10990	11230	11240	11440	11540	11620					
		11660	12070	12380	12390	12490	12580	12530	12580	12590	12600	12980	13020	13060					
		13210	13510	13520	13550	13640	13690	13980	14050	14060	14540	14920	14980	15040					
		15070	16240	16660	16690	17010	17240	17270	17770	18430	18490	18500	21380	21440					
		21970	22440	23660	24850	25920	25990	26130	26150	26660	26840	26210							
PIC0	002266	4970	1004																
PIC1	002332	5100	1005																
PIC12	004342	8810	1011																
PIC12A	004350	8820	887																
PIC3	002446	517	5420	1006															
PIC4	002652	5910	1007																
PIC4B	002704	5960																	
PIC5	003030	6320	1008																
PIC6	003212	6760	1009																
PIC6A	003224	6780	694																
PIC6B	003346	682	687	692	6960														
PIC7	004070	695	8290	1010															
PIC7A	004144	8370	841																
PIC7AA	004106	8320	853																
PIC7B	004226	8470	851																
PIRQ	177772	270	4010	400															
PIRQVE	000240	1210																	
PLUS1	013320	1032	20350																
PLUS2	013322	1035	20360																
PLUS3	013324	1038	20370																
PLUS4	013326	1041	20380																
PLUS5	013330	20390																	
PRINT1	013150	19830																	
PROC	013154	10400	1049	1053	10850														
PR0	000000	440																	
PR1	000040	450																	
PR2	000100	460																	
PR3	000140	470																	
PR4	000200	480																	
PR5	000240	490																	
PR6	000300	500																	
PR7	000340	510																	
PS	177776	240	25	3940	4020	411													
PSW	177776	250	16000	17150															
PWRMSG	014314	2243	22500																
PWRVEC	000024	1160	3650	3660	22120	22130	22220	22280	22400	22410									
P3	002510	5500	585																
P3A	002534	5550	559																
P3B	002556	5630	567																
P3C	002600	5710	575																
P3D	002622	5790	583																
P4	002724	6000	625																
P4A	002742	6060	611																
P4B	002776	6100	623																

PYCNT	004262	835	840	845	850	855								
PYPNT	004264	836	837	839	846	847	849	856						
Q	003674	776												
QMARK	012365	482	1629	1668	1910									
R	003701	778												
RDCNR	104406	2441	2507	2662										
RDLIN	104407	2549	2663											
RDOCT	104410	2664												
RECVY	007024	488	1346											
RECVY1	007036	1346	1349											
RECVY2	007066	1355	1398	1407										
REPT8T	006244	477	1231											
REPT1	006256	1231	1234	1245										
REPT2	006342	1243	1247											
REPT2A	006364	1252	1333											
REPT3	006372	1253	1332											
REPT4	006570	1262	1268	1273	1279	1282	1245							
REPT5	006606	1288	1291											
REPT6	006714	1311	1314											
REPT6A	006742	1320	1323											
REPT7	006766	1264	1269	1274	1280	1284	1325	1327						
REPT7A	007014	1330	1333											
REPT8	006626	1295												
REPT8A	007020	1293	1294	1295	1335									
REPT8T	007444	468	1425											
RESVEC	000010	1110												
ROUTPT	005030	996	1004											
R0	000000	320	406	409	412	415	417	418	434	435	436	440	442	497
		510	553	558	562	566	570	574	578	582	605	610	617	622
		642	646	654	664	716	730	918	921	937	943	957	998	1003
		1000	1100	1101	1102	1103	1104	1105	1109	1110	1111	1112	1131	1142
		1205	1371	1372	1373	1374	1375	1376	1377	1378	1379	1384	1393	1432
		1433	1436	1440	1445	1448	1450	1523	1533	1537	1553	1554	1567	1653
		1658	1805	1808	2169	2170	2171	2178	2179	2180	2181	2182	2183	2188
		2193	2195	2199	2201	2214	2239	2359	2360	2365	2370	2373	2386	2550
		2553	2569	2581	2589	2593	2594	2596	2597	2598	2620	2639	2640	2641
		2642	2643	2644	2645									
R1	000001	338	424	425	426	427	441	442	443	444	498	511	546	596
		641	647	661	717	726	938	941	958	1084	1089	1112	1113	1114
		1115	1186	1196	1370	1372	1373	1374	1375	1376	1377	1378	1379	1386
		1389	1392	1434	1436	1441	1444	1450	1524	1537	1538	1542	1566	1586
		1587	1588	1590	1492	1498	1602	1608	1612	1617	1620	1622	1641	1654
		1655	1656	1657	1658	1660	1663	1813	1817	1818	1821	1822	2215	2238
		2547	2551	2555	2557	2559	2562	2565	2568	2582	2619			
R2	000002	340	423	425	427	428	429	522	525	547	597	660	665	680
		685	698	837	847	959	1107	1115	1161	1195	1196	1197	1198	1199
		1200	1201	1320	1385	1391	1435	1437	1442	1451	1525	1536	1540	1543
		1550	1551	1552	1557	1565	1642	1654	1814	1815	2216	2237	2548	2552
		2556	2558	2560	2566	2567								
R3	000003	350	521	529	548	598	638	643	648	662	718	719	1072	1076
		1080	1133	1135	1138	1200	1526	1534	1535	1549	1552	1561	1562	1564
		1639	1663	1784	1786	1792	1794	1800	1801	1802	2217	2236	2293	2302
		2308	2309	2312	2317	2318	2319	2320	2503	2504	2505	2508	2509	2513
		2515	2517	2519										
R4	000004	360	528	549	555	563	571	579	599	608	609	620	621	715
		725	729	732	1073	1077	1081	1134	1136	1141	1204	1319	1320	1322

		1327	1328	1329	1380	1479	1502	1618	1638	1646	1648	1659	1642	1653
		1662	1682	1697	1755	1757	1760	2218	2234	2244	2296	2297	2298	2299
R5	01000005	2300	2314	2316	2424	2527								
		370	649	651	661	667	668	697	831	1061	1120	1130	1138	1201
		1527	1529	1531	1938	1942	1957	1963	2219	2234	2295	2301	2303	2305
		2306	2307	2308	2324									
R6	01000006	380	40	353	454	455								
R7	01000007	390	41											
S	003706	7000												
SP	01000006	400	357	374	404	386	915	998	1149	1104	1295	1299	1303	1307
		1401	1460	1498	1523	1524	1525	1526	1527	1528	1529	1532	1545	1547
		1549	1559	1561	1963	1964	1565	1566	1567	1568	1570	1406	1610	2067
		2070	2072	2073	2102	2103	2107	2133	2153	2156	2169	2174	2195	2199
		2214	2215	2216	2217	2218	2219	2220	2221	2229	2233	2234	2235	2236
		2237	2238	2239	2244	2245	2286	2287	2288	2293	2294	2295	2301	2326
		2327	2328	2329	2330	2359	2368	2370	2372	2373	2374	2376	2378	2380
		2386	2388	2390	2398	2402	2406	2407	2411	2429	2430	2431	2436	2439
		2440	2443	2446	2449	2451	2453	2454	2456	2458	2460	2461	2462	2463
		2467	2468	2470	2471	2472	2473	2474	2484	2485	2488	2489	2490	2492
		2494	2503	2508	2519	2520	2521	2522	2544	2545	2546	2547	2548	2550
		2553	2561	2562	2564	2565	2567	2568	2569	2581	2582	2589	2590	2601
		2602	2603	2613	2614	2619	2620	2630	2640					
SPACEA	004040	8160	864	873										
SPACEX	007712	1404	1409	1491	1493	1493								
STACK	001100	150	357											
STCHAN	013212	20000												
STKLMY	177774	260												
SWR	001136	2160	376	378	402	392	639	882	888	945	949	982	1045	1048
		1070	1127	1147	1155	1261	1267	1324	1397	2062	2076	2078	2084	2091
		2129	2136	2148	2151	2220	2233	2425	2453	2460				
SWREG	000176	1490	382	2425	2436									
SW0	000001	790												
SW00	000001	690	79											
SW01	000002	680	78											
SW02	000004	670	77											
SW03	000010	660	76											
SW04	000020	650	75											
SW05	000040	640	74											
SW06	000100	630	73											
SW07	000200	620	72											
SW08	000400	610	71											
SW09	001000	600	70											
SW1	000002	780												
SW10	002000	590	1147	1261	1397									
SW11	004000	580												
SW12	010000	570	945											
SW13	020000	560	1263											
SW14	040000	550												
SW15	100000	540												
SW2	000004	770												
SW3	000010	760												
SW4	000020	750												
SW5	000040	740												
SW6	000100	730												
SW7	000200	720												
SW8	000400	710												

SW9	001020	700												
T	003713	7020												
TAGA	011256	1765	17670											
TBITVE	000014	1120												
TEMP	013226	3000	3500	400	20060									
TEMP1	013230	0560	061	10070	10070	10000	10000	17530	17600	17710	17740	17820	17870	17900
		17950	18040	18060	20070									
TEMP2	013232	17600	1761	1763	1764	1766	1767	17830	17850	1786	17910	17930	1794	18000
		1809	1811	1815	20000									
TEMP3	013234	11460	13500	15050	20000									
TICKS	013210	0000	0120	0000	0000	0320	0760	0200	0010	0000	10160	10000		
TIMER	004726	002	015	000	020	056	003	052	006	0010				
TIMERA	004742	0000												
TIMER1	004764	007	0000											
TIMER2	004766	005	0000											
TIMERA	005024	10020												
TIMSV	013206	0020	000	0000	0050	0060	0070	1022	10000					
TITLE	011706	055	10520											
TKVEC	000060	1100												
TPVEC	000064	1200												
TRAPVE	000034	1100	3630	3640										
TRTVEC	000014	1130												
TSLO	011240	1762	17640											
TSYCT1	006550	1277	12810											
TSYCT2	006526	1271	12760											
TSYCT3	006504	1266	12700											
TSYCT4	006452	12630												
TSYFLG	011672	003	1100	1150	1253	1355	1454	10070						
TSY1	002256	0050												
TSY10	004332	054	0700											
TSY11	005074	10220												
TSY12	006234	12100												
TSY13	007022	13000												
TSY14	007434	14230												
TSY2	002322	0000												
TSY3	002436	0370												
TSY4	002642	0000												
TSY5	003020	0300												
TSY6	003202	050	6740											
TSY7	004060	0270												
TYPOS	104404	016	1461	26600										
TYPE	104400	050	056	060	001	006	000	010	017	007	1035	1037	1041	1152
		1232	1236	1247	1285	1200	1201	1312	1317	1307	1300	1300	1426	1457
		1462	1560	1600	1620	1620	1667	2131	2130	2160	2185	2187	2190	2192
		2196	2203	2242	2320	2401	2433	2435	2430	2445	2460	2511	2510	2510
		26560												
TYPOC	104401	2176	2200	2437	26570									
TYPON	104403	26590												
TYPOS	104402	1150	1296	1300	1300	1300	1402	1500	26500					
TYPT1	011002	17100	1720											
TYPT2	011010	17200	1722											
TYPT3	011042	1726	17200											
U	003720	7040												
USECLK	013222	10030	10560	1107	20000									
V	003725	7060												
VCSTAT	001346	3020	0600	0100	020	0430	040	0030	000	0350	030	0010	0000	0010

		697	712	723	831	832*	842*	934*	935*	939	955*	957	1067*	1074*
VXREG	001350	1078*	1079	1119*	1120	1132*	1137*	1139	1202	1453*				
		3430	497	511	546	550*	596	636*	642	647	710*	721*	950	1068*
VYREG	001352	1072	1077											
		3440	498	510	547	551*	597	637*	641	646	711*	722*	959	1069*
		1073	1076											
VISAU1	002250	405	408*											
VISUAL	002236	471	485*											
VSUAL0	002254	453	490*	927										
W	003732	788*												
X	003737	790*												
XPOS	003546	678*	683*	688*	710	721	729*	732*	737*	833*	843*	1121*	1124*	
XPRYAV	007714	1406	1497*											
XPTA1	007722	1408*	1506											
XSPACE	007662	1298	1302	1306	1405	1406*	1490	1504						
XSPRD1	013334	1283	1322	1831*	1832*	1833*	1834	2041*						
XSPRD2	013336	1278	1834*	1835*	1836*	1837	2042*							
XSPRD3	013340	1272	1837*	1838*	1839*	1840	2043*							
XSPRD4	013342	1267	1840*	1841*	1842*	2044*								
XTTYIN	010206	465	1039	1238	1249	1351	1579*	1630	1669	1849				
Y	003744	792*												
YPOS	003544	679*	684*	689*	708	711	722	725*	728*	736*	834*	844*	1122*	1125*
YPT	003552	700*	728	739*										
Z	003751	794*												
ZERO	011136	1718*	1734	1736*	1740*									
SAPTHD	001000	177	183*											
SARG1	001616	388*												
SASTAT*	***** U	2607	2622											
SATYC	015712	2578	2580*											
SATY1	015666	2576*												
SATY3	015674	2366	2577*											
SATY4	015704	2144	2579*											
SBASE	001254	2850	435	438										
SDDADR	001122	2100												
SDDDAY	001126	2120	355	1445*	1446	1449								
SBELL	001170	2320	2131	2159										
SCDW1	001260	2870												
SCDW2	001262	288*												
SCHARC	015066	2383*	2393*	2400	2409*	2414*								
SCKSWR	015072	2425*	2661											
SCHTAG	001100	198*	352	353	361	367	368							
SCH1	000002	2280	229*	230*										
SCH2	000004	2280	229*	230*										
SCH3	000002	226*	228											
SCNTLG	015535	2433	2520*											
SCNTLU	015530	2445	2527*											
SCPUOP	001226	256*												
SCRLF	001175	2340	2139	2159	2168	2187	2192	2196	2382	2417	2464	2527		
SDBLK	010176	1534	1568	1576*										
SDDW0	001264	289*												
SDDW1	001266	290*												
SDDW10	001310	299*												
SDDW11	001312	300*												
SDDW12	001314	301*												
SDDW13	001316	302*												
SDDW14	001320	303*												

SDDW15	001322	3040							
SDDW2	001270	2910							
SDDW3	001272	2920							
SDDW4	001274	2930							
SDDW5	001276	2940							
SDDW6	001300	2950							
SDDW7	001302	2960							
SDDW8	001304	2970							
SDDW9	001306	2980							
SDEVCT	001210	2470							
SDEVH	001256	2860							
SDOAGN	004512	910	919	9250					
SOTBL	010166	1537	15720						
SENDAD	004502	162	9210						
SENDCT	004450	9120							
SENDMG	004521	914	9290	1450					
SENULL	004516	917	9280	1462					
SENV	001220	2520	2141	2361	2505	2609			
SENVH	001221	2530	390	2363	2360	2507			
SEOP	004414	9020							
SEOPCT	004442	9090	913						
SERFLG	001103	2010	2051	2080	2082	2080	2110	2126	2159
SERMAX	001115	2070	360	2082	2105	2110			
SERROR	013624	361	21250						
SERRPC	001116	2000	1980	2133	2134	2135	2159	2174	
SERRTB	001324	3240	2182						
SERRTY	014006	2130	21670						
SERTYL	001112	2050	2132	2159					
SESCAP	001166	2310	360	2104	2154	2156	2159		
SETABL	001220	2510							
SETEND	001324	109	3070						
SFATAL	001202	2440	2613						
SFFLG	016132	2576	2579	2607	2616	26240			
SFILLC	001154	2240	2306	2417					
SFILLS	001153	2230	2417						
SGDADR	001120	2090							
SGDDAT	001124	2110	1444	1446	1480				
SGEY02	004472	9100							
SHD	000000	11							
SHIBTS	001000	1040							
SHIOCT	015664	2566	25710						
SICNT	001104	2020	2095	2096	2090	2109			
SILLUP	014306	2212	2228	22470					
SITEMB	001114	2060	2135	2143	2159	2171			
SLF	001176	2350	2159	2417	2518	2527			
SLFLG	016131	2617	26230						
SLPADR	001106	2030	370	2086	2102	2107	2109	2109	
SLPERR	001110	2040	371	2086	2103	2109	2153		
SHADR1	001232	2690							
SHADR2	001236	2730							
SHADR3	001242	2760							
SHADR4	001246	2790							
SHAIL	001200	185	189	2420	300	2101	2141	2361	
SHAMS1	001230	2630							
SHAMS2	001234	2710							
SHAMS3	001240	2740							

SMAMS0	001244	2770												
SMADR	001002	1850												
SMFLG	016130	2577*	25A3	2610*	26220									
SMNEW	015553	2430	25310											
SMGAD	001214	2490	2593*	2596										
SMGLG	001216	2500	259A*											
SMGTY	001200	2430	2591	2599*	2611	26150								
SMHR	015542	2435	25290											
SMTYP1	001231	2640												
SMTYP2	001235	2720												
SMTYP3	001241	2750												
SMTYP4	001245	2780												
SMXCNT	013622	2099	21090											
SNUL	001152	2220	230A	2417										
SNWTST	000001	4020	5050	5340	5860	6270	6710	0240	0760	10190	12150	13370	14200	
SOCNT	014604	2292*	2321*	23340										
SOMODE	014606	2287*	2291*	2296	2299*	2310*	23360							
SOVER	013606	2063	2070	2087	2097	21060								
SPASS	001206	2460	309*	906*	907*	915	920	2093	2110					
SPASTM	001006	1870												
SPRIOR	001252	2820												
SPHRAD	014302	22450												
SPHRDN	014142	365	22120	2240										
SPHRMG	014276	22430												
SPHRUP	014214	2222	22200											
SQUES	001174	2330	2150	2417	2511	2527								
SROCHR	015330	24840	2662											
SRODEC	***** U	2665												
SROLIN	015412	25030	2663											
SROCT	015564	25440	2664											
SROSZ	000010	24960												
SREGAD	001156	2260												
SREG0	001160	2280												
SREG1	001162	2290												
SRTNAD	004514	9270												
SR2A	***** U	2665												
SSAVRE	***** U	2665												
SSAVR6	014312	2221*	2229	2230*	2231*	22490								
SSCOPE	013344	359	20600											
SSETUP	000017	3260	3520	358	359	361	363	365	367	368	370	387	904	2061
		2126	2151	2150										
SSTUP	177777	3260	3520											
SSVLAD	013552	2071	21000											
SSVPC	000210	1600	165											
SSWR	167400	10	11	132	133	134	135	136	137	138	139	230	231	232
		367	368	370	371	406	509	530	590	631	675	020	800	899
		905	920	926	928	1023	1219	1341	1424	2052	2053	2054	2055	2056
		2062	2074	2076	2077	2080	2081	2082	2089	2090	2091	2103	2106	2109
		2117	2118	2119	2120	2121	2129	2136	2140	2151	2159	2246		
SSWREG	001222	2500	392											
SSWRMK	000000	139	140	2056	2057	2070								
STESTN	001204	2450	2101*											
STIMES	001164	2300	367*	496*	509*	530*	590*	631*	675*	820*	880*	905*	1023*	1219*
		1424*	2089*	2096	2099*	2109								
STKB	001144	2190	1586	2421	2429	2480								
STKS	001142	2100	1580*	1584	1047	2421	2427	2486						

STN	000015	10	11	492	4960	505	5090	534	53A0	586	5900	627	6310	650
		671	6750	820	8280	854	876	8800	1019	10230	1215	12190	1337	13410
		1420	14240											
STPB	001150	2210	14800	15900	17400	24260	2417							
STPFLG	001155	2250	2355	2417										
STPS	001146	2200	1486	1596	1730	2404	2417							
STRAP	016134	363	26390											
STRP	000011	26470	26570	26580	26590	26600	26610	26620	26630	26640	26650			
STRPAD	016156	2644	26550											
STSYM	001004	1860												
STSYNM	001102	2000	9040	2051	2070	21000	2101	2106	2110	2120	2150			
STTYIN	015520	2504	2505	2522	25260									
STYPOB	***** U	2661												
STYPOS	007762	15220	2660											
STYPE	014610	23550	2604	2647	2656									
STYPEC	015022	2305	2392	2390	24040	2405	2466							
STYPEX	015070	2410	2417	24150										
STYPOC	014406	22900	2657											
STYPOB	014422	2289	22920	2650										
STYPOS	014362	22850	2650											
SUNIT	001212	2400												
SUNITM	001010	1800												
SUBWR	001224	2550												
SVECT1	001250	2000	10500											
SVECT2	001251	2010												
SXTSTR	013356	20650												
SSGET4	000000	9200												
SOFILL	014605	22860	22900	2300	23350									
S00CAT	***** U	2062	2130											
.	016662	1430	1470	160	1610	1630	1650	1660	173	1740	1760	1780	1970	236
		356	370	371	520	597	565	573	501	607	613	610	8230	920
		932	960	1407	14760	16340	1739	1745	2100	2110	2150	22060	2224	2248
		2417	2421	25260	2427	2433	26250	26670	26690	26710	26730	26750	26770	26790
		26010	26030											
.SASTA	***** U	2577	2580											
.SX	001000	1730	170											

COMMEN	1220														
ENDCOM	1220														
ERROR	160	1409													
ESCAPE	1220														
GETPRI	1220														
MULT	1220														
NEWTST	1220	492	585	534	586	627	671	824	876	1019	1215	1337	1420		
POP	1220	1563	2233	2234	2567	2619	2620								
PUSH	1220	1522	2214	2220	2546	2580	2582	2603							
REPORT	1220														
SCOPE	170	495	588	537	589	630	674	827	879	983	1022	1218	1340	1423	
SETPRI	1220														
SETTRA	26470	2657	2658	2659	2660	2661	2662	2663	2664						
SETUP	1220	352													
SKIP	1220	658	854	1447											
SLASH	1220														
SPACE	1220														
STARS	1220	158	170	172	179	193	236	240	492	494	495	507	534	576	586
	588	627	629	671	673	824	826	876	878	895	1019	1021	1215	1217	1337
	1339	1420	1422	1512	2040	2113	2162	2210	2226	2262	2540	2420	2476	2496	2536
	2575	2633													
SWRSU	1220	3720													
TRMTRP	26470														
TYPBIN	1220														
TYPDEC	1220	915													
TYPNAM	1220														
TYPNUM	1220														
TYPOCS	1220														
TYPOCT	1220	2174	2198	2436											
TYPTXT	1220														
SSCHRE	1980	228	229												
SSCHTM	1980														
SSESCA	1220														
SSNEWT	1220	492	585	534	586	627	671	824	876	1019	1215	1337	1420		
SSSET	26470	2657	2658	2659	2660	2661	2662	2663	2664						
SSSETH	3880	389													
SSSKIP	1220	658	854												
.EQUAT	10	11													
.HEADE	10														
.SETUP	10	326	352												
.SWRHI	10	127													
.SWRLO	1400														
.SACT1	10	155													
.SAPT0	10	2370													
.SAPTH	10	167													
.SAPTY	10	2572													
.SCATC	10	140													
.SCMTA	10	100													
.SEOP	10	892													
.SERRO	10	2110													
.SERRY	10	2150													
.SPARM	10														
.SPOWE	10	2207													
.SRDOC	10	2533													
.SREAD	10	2417													
.SSAVE	10														

.SSCOP	1#	2045
.SSPAC	1#	
.SSWDO	1#	
.STRAP	1#	2630
.STVPO	1#	1509
.STVPE	1#	2337
.STVPO	1#	2250

ADC	1768	1776													
ADD	442	525	555	563	608	609	621	664	725	729	732	A30	A49	90A	996
	1141	1163	1196	1204	1380	1433	1502	1542	1658	1723	1728	1767	1A32	1A33	1835
	1836	1838	1839	1841	1842	21A2	2288	2298	2374	2446	2454	2461	2463	2562	2590
	2602	2614													
ASL	417	994	1016	1114	1197	119A	1199	1654	1656	1657	1A21	2179	21A0	21A1	2470
	2471	2472	2555	2557	2559	2643									
ASLB	1547														
ASR	1101	1102	1104	1109	1110	1111	1772	2597							
BCC	1548														
BEQ	391	452	640	883	889	919	1015	1071	1128	114A	1156	1185	1188	1243	1284
	1330	1447	1615	1621	1647	1733	1807	1816	2077	2079	20A1	2085	2094	2127	2130
	2152	2155	2184	2189	2202	2315	2364	2377	2412	2452	2459	2469	2554	2584	2588
	2600	2610													
BGE	2097														
BGT	910	1400	1556	1651	1661	1699	1765	2322	2493						
BHI	2083														
BIC	709	907	994	1046	1113	1256	1294	1328	1361	1366	1500	1507	1652	2312	2430
	2467	2489	2494	2561											
BIS	714	935	1051	1055	105A	1257	1314	1362	1367	1550	1551	1737	2317	2318	2474
BISB	1191	2171													
BIT	639	802	808	945	984	1070	1127	1147	1261	1263	1397	2062	2076	2084	2091
	2129	2136	2151												
BITB	390	2363	2368	2400	2507										
BLE	1644	1762													
BLOS	2506														
BLT	1539	1555	1649	2323	2391	2491									
BMI	940	1245	1546	1589	1591	1595	1810	1812							
BNE	356	379	430	437	445	450	467	470	473	476	479	530	559	567	575
	583	611	623	655	666	703	727	731	841	851	942	944	946	970	973
	985	987	1050	1054	1094	1117	1143	1165	1206	1262	1264	1266	1268	1271	1273
	1277	1279	1282	1288	1311	1323	1302	1394	1398	1438	1452	1456	1506	1544	1603
	1609	1613	1623	1665	1684	1688	1726	1731	1770	1775	1788	1796	1803	1819	2063
	2092	2137	2142	2172	2194	2232	2313	2362	2369	2371	2379	2387	2401	2408	2426
	2432	2444	2450	2457	2510	2516	2586	2592	2595	2612					
BPL	528	557	565	573	581	607	613	619	663	713	720	724	950	968	1140
	1203	1325	1407	1530	1560	1585	1597	1696	1722	1739	1848	2149	2311	2356	2485
	2428	2487													
BR	349	483	503	516	517	585	625	644	657	658	694	853	854	847	963
	1017	1052	1057	1075	1108	1145	1269	1274	1280	1407	1463	1541	155A	1599	1619
	1630	1659	1690	1729	1735	1823	1825	1827	2065	2071	2074	2087	2090	2147	2177
	2204	2224	2240	2289	2304	2325	2358	2384	2394	2403	2410	2447	2455	2475	2512
	2563	2578	2600												
CLR	348	354	367	368	389	394	413	414	426	462	463	464	519	542	543
	593	635	636	637	681	686	691	904	905	938	955	1043	1047	1067	1068
	1069	1085	1119	1132	1135	1136	1234	1235	1357	1428	1453	1491	1533	1536	1581
	1582	1583	1640	1641	1642	1680	1700	1715	1756	1801	1813	2089	2104	2170	2230
	2302	2439	2440	2551	2552										
CLRB	1193	1562	2088	2303	2409	2517	2616	2617	2618						
CMP	355	378	429	436	444	529	969	998	1049	1155	1194	1244	1265	1267	1270
	1272	1276	1278	1281	1283	1315	1322	1329	1446	1450	1554	1594	1658	1660	1732
	1761	1764	1802	1809	1811	1815	1818	2072	2096	2425	2431	2443	2449	2456	2490
	2492	2505													
CMPB	466	469	472	475	478	1093	1588	1590	1602	1608	1612	1620	1622	1646	1648
	2078	2082	2141	2361	2376	2378	2386	2407	2411	2509	2515	2585			
DEC	558	566	574	582	610	622	665	840	850	900	941	943	986	1014	1093

	1116	1142	1164	1186	1205	1301	1393	1437	1451	1459	1489	1505	1416	1645	1687
	1698	1769	1774	1787	1794	1795	1806	2178							
DECB	2310	2321	2390	2393											
EMT	16														
HALT	147	951	1326	1820	2150	2223	2247	2357							
INC	416	439	526	702	726	740	906	945	1090	1103	1331	1363	1364	1459	1540
	1593	1720	1725	1785	1817	1822	1824	1826	2095	2132	2231	2316	2324	2473	2615
INCB	2100	2126	2413												
IOT	17														
JMP	152	153	419	453	468	471	474	477	480	695	926	1002	1157	1158	1332
	1333	1607	1611	1670											
JBR	440	459	465	500	501	502	513	514	515	545	584	595	624	633	649
	651	656	677	682	687	692	693	701	701	730	730	730	730	730	730
	890	921	983	1039	1086	1091	1099	1123	1126	1144	1154	1162	1230	1239	1249
	1250	1253	1250	1259	1260	1298	1302	1306	1321	1351	1352	1355	1364	1369	1390
MOV	1405	1406	1454	1504	1669	1724	1849	2138	2144	2364	2385	2392	2399	2466	2674
	350	353	357	359	360	361	362	363	364	365	366	370	371	374	375
	376	377	382	383	384	386	392	404	405	406	415	418	423	424	425
	427	434	435	438	440	441	458	485	496	497	498	499	509	510	511
	512	520	521	522	523	524	538	544	546	547	548	549	550	551	553
	554	561	562	569	570	577	578	590	591	592	598	596	597	598	599
	600	601	604	605	615	616	617	631	632	638	641	642	643	646	647
	648	660	661	675	676	678	679	680	683	684	685	688	689	690	696
	697	708	710	711	715	716	717	721	722	728	728	729	731	732	733
	834	835	836	837	842	843	844	845	846	847	880	881	911	915	918
	934	937	956	957	958	959	960	961	962	982	997	1013	1023	1034	1040
	1044	1048	1056	1059	1061	1065	1072	1073	1074	1076	1077	1078	1079	1080	1081
	1082	1083	1084	1088	1089	1097	1098	1100	1105	1106	1107	1112	1115	1120	1121
	1122	1124	1125	1130	1131	1133	1134	1137	1138	1146	1149	1159	1160	1161	1189
	1190	1195	1200	1201	1219	1231	1240	1241	1246	1251	1252	1254	1293	1295	1299
	1303	1307	1319	1320	1327	1346	1353	1354	1356	1358	1359	1370	1371	1372	1373
	1374	1375	1376	1377	1378	1379	1384	1385	1386	1387	1388	1389	1391	1401	1404
	1424	1425	1430	1432	1434	1435	1436	1440	1441	1442	1444	1445	1448	1460	1488
	1497	1498	1503	1523	1524	1525	1526	1527	1528	1529	1534	1537	1557	1563	1564
	1565	1566	1567	1569	1570	1579	1586	1592	1617	1638	1639	1653	1654	1663	1681
	1682	1686	1693	1697	1716	1717	1718	1719	1734	1736	1740	1753	1754	1755	1757
	1758	1759	1760	1763	1766	1771	1782	1783	1784	1786	1790	1791	1792	1794	1800
	1804	1805	1808	1814	1831	1834	1837	1840	2067	2068	2070	2073	2086	2098	2099
	2102	2103	2106	2107	2120	2133	2153	2156	2169	2174	2183	2188	2193	2195	2199
	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2228	2229	2233	2234
	2235	2236	2237	2238	2239	2240	2241	2244	2285	2293	2294	2295	2301	2308	2326
	2327	2328	2329	2330	2359	2360	2365	2373	2388	2436	2453	2460	2462	2484	2485
	2503	2504	2519	2520	2521	2522	2544	2545	2546	2547	2548	2550	2565	2566	2567
	2568	2569	2581	2582	2589	2593	2598	2599	2601	2603	2613	2619	2620	2639	2640
	2644														
MOVB	369	718	1045	1255	1360	1365	1532	1535	1549	1552	1561	1590	1605	1609	2101
	2105	2135	2143	2286	2287	2290	2291	2292	2296	2299	2300	2319	2370	2388	2406
	2429	2488	2500	2513	2553	2576	2577	2579	2642						
NEG	1531	2297													
NOP	490	698	699	700	922	923	924	936	966	999	1000	1001	1060	1062	1063
	1064	1066	1316	1431	1691	1692									
RESET	351	891	920	1429											
ROL	2303	2305	2306	2307	2309	2556	2558	2560							
ROLB	719														
ROR	1773														
RTI	385	404	1571	2100	2158	2246	2331	2375	2465	2495	2523	2570			

RTS	446	531	668	708	733	943	974	9A9	1P18	1166	1207	1492	1507	142A	1666
	1701	1727	1741	1777	1843	1A50	2197	2415	2621	2645					
SBC	429	412													
SEC	407	410													
SUB	571	579	620	964	972	153A	1721	2174	2596						
TRAP	2647	2657	2658	2659	2660	2A61	2642	2667	2664						
YST	381	408	428	443	449	451	454	667	949	1184	11A7	1242	1287	1310	1324
	1392	1543	1553	1686	1610	1614	1618	1643	1662	1664	1683	1730	2060	2093	2148
	2154	2201	2314	2372	2380	2402	2451	2458	2468	2564	2491	2689	2411	2641	
YSTB	411	527	556	564	572	580	406	612	618	662	712	723	930	967	1139
	1202	1486	1545	1550	1584	1596	1A95	1738	1847	2080	2455	2488	2427	2486	2583
	2594	2607													
WAIT	1192														
.ASCII	233	234	1873	1880	1885	1890	1A95								
.ASCIZ	232	235	929	1852	1861	1A66	1908	1904	1906	1910	1911	1915	1920	1925	1927
	1934	1940	1943	1950	1953	1957	1967	1973	2205	2250	2527	252A	2529	2531	
.BLKB	2526														
.BLKW	1576														
.BYTE	200	201	206	207	222	223	224	225	252	253	263	264	271	272	274
	275	277	278	280	281	282	283	744	746	748	750	752	754	756	758
	760	762	764	766	768	770	772	774	776	778	780	782	784	786	788
	790	792	794	796	798	800	802	804	806	808	810	812	814	816	818
	820	928	1151	1297	1301	1305	1309	1403	1501	1907	1908	2145	2146	2332	2333
	2334	2335	2524	2525	2622	2623	2624								
.ENABL	1														
.END	2688														
.ENDC	6	16	108	122	136	138	139	140	153	159	163	165	171	173	180
	194	198	200	226	230	231	232	233	237	241	263	271	274	277	280
	281	282	283	286	287	288	289	290	291	292	293	294	295	296	297
	298	299	300	301	302	303	304	305	309	326	352	357	358	361	363
	365	367	368	370	372	394	493	494	495	496	497	506	507	508	509
	510	535	536	537	538	539	587	588	589	590	591	628	629	630	631
	632	650	672	673	674	675	676	825	826	827	828	829	855	877	878
	879	880	881	896	898	899	901	904	910	913	914	91A	920	926	928
	929	932	1020	1021	1022	1023	1024	1216	1217	1218	1219	1220	1338	1339	1340
	1341	1421	1422	1423	1424	1425	1448	1513	2049	2052	2057	2062	2064	2075	2078
	2079	2080	2082	2084	2091	2095	2100	2102	2106	2109	2110	2114	2117	2126	2133
	2138	2139	2140	2148	2150	2159	2163	2178	2207	2211	2220	2221	2227	2233	2234
	2244	2246	2250	2263	2341	2370	2421	2477	2496	2497	2504	2506	2509	2511	2527
	2537	2539	2572	2576	2577	2580	2607	2622	2634	2640	2643	2656	2657	2658	2659
	2660	2661	2662	2663	2664	2665									
.EQUIV	16	17	25	40	41	70	71	72	73	74	75	76	77	78	79
	98	99	100	101	102	103	104	105	106	107					
.EVEN	241	284	823	1979	2206	2257	2625								
.IF	2	14	80	108	135	137	138	139	140	150	15A	161	163	170	172
	179	193	197	199	226	230	231	232	236	237	240	263	271	274	277
	280	281	282	283	286	287	288	289	290	291	292	293	294	295	296
	297	298	299	300	301	302	303	304	305	309	326	352	357	359	361
	363	365	367	368	370	388	492	494	496	497	505	507	508	510	534
	536	538	539	586	588	590	591	627	629	631	632	650	671	673	675
	676	824	826	828	829	854	876	878	880	881	895	896	897	898	899
	900	901	903	909	912	914	918	920	926	928	929	1019	1021	1023	1024
	1215	1217	1219	1220	1337	1339	1341	1420	1422	1424	1425	1447	1512	2040	2051
	2056	2061	2062	2074	2076	2077	2078	2080	2081	2082	2091	2093	2101	2103	2108
	2109	2110	2113	2116	2126	2129	2136	2138	2139	2141	2148	2151	2158	2159	2162
	2177	2193	2210	2220	2221	2226	2233	2234	2242	2244	2246	2250	2262	2340	2361

	2420	2421	2476	2496	2504	2505	2509	2510	2526	2527	2536	2539	2551	2574	2577
	2500	2607	2622	2633	2639	2643	2647	2657	2658	2659	2660	2661	2662	2663	2664
	2665														
.IFF	14	135	138	139	140	159	163	165	171	173	180	194	197	200	226
	237	241	357	493	494	495	496	497	506	507	508	509	510	535	536
	537	538	539	587	588	589	590	591	628	629	630	631	632	659	672
	673	674	675	676	825	826	827	828	829	855	877	878	879	880	881
	896	900	904	909	912	928	1020	1021	1022	1023	1024	1216	1217	1218	1219
	1220	1330	1339	1340	1341	1421	1422	1423	1424	1447	1413	2049	2075	2078	2079
	2082	2109	2110	2114	2116	2129	2158	2159	2163	2178	2207	2211	2227	2242	2263
	2341	2421	2443	2477	2479	2484	2496	2497	2506	2510	2527	2537	2576	2634	2640
.IFT	2090	2139	2421	2443	2479	2484	2495	2571	2572						
.IFTF	2080	2130	2435	2449	2477	2480	2551	2555	2571						
.IIF	1	6	11	132	133	134	136	139	140	147	236	241	358	361	367
	368	370	371	387	898	904	905	916	920	932	2052	2053	2054	2055	2056
	2057	2061	2089	2090	2106	2109	2110	2117	2118	2119	2120	2121	2126	2151	2158
	2159	2175	2200	2417	2421	2437	2519	2527	2533	2656	2657	2658	2659	2660	2661
	2662	2663	2664												
.IRP	326	352	492	505	534	586	627	671	824	876	1019	1215	1337	1420	1523
	1563	2061	2214	2220	2233	2234	2546	2567	2581	2582	2603	2619	2620		
.LIST	1	122	139	147	226	228	229	230	237	241	326	352	372	492	496
	505	509	534	538	586	590	627	631	671	675	824	828	876	880	904
	928	1019	1023	1215	1219	1337	1341	1420	1424	2056	2158	2496	2647	2656	2657
	2492	2459	2460	2461	2462	2463	2464	2465							
.MACRC	100	140	308	2647											
.MCALL	1	122	237	372											
.NLIST	1	122	139	147	226	228	229	230	237	241	326	352	372	492	496
	496	509	534	538	586	590	627	631	671	675	824	828	876	880	904
	928	1019	1023	1215	1219	1337	1341	1420	1424	2256	2158	2496	2647	2656	2657
	2492	2459	2460	2461	2462	2463	2464	2465							
.PAGE	100	246													
.REPT	147	228													
.SBYTL	12	128	141	151	156	164	191	248	310	492	525	534	586	627	671
	628	876	893	1019	1215	1337	1420	1510	2046	2111	2160	2208	2260	2338	2418
	2534	2573	2631	2648											
.TITLE	1														
.WORD	147	148	149	164	184	185	186	187	188	189	199	202	203	204	205
	208	209	210	211	212	213	214	215	216	217	226	228	229	243	244
	245	246	247	248	249	250	254	255	256	269	273	276	279	285	286
	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301
	302	303	304	909	912	927	2106	2191	2243	2245	2336	2367	2414	2571	2605

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

*DZARRB/SOL/CRF<DZARRB.P11
 RUN-TIME: 55 24 6 SECONDS
 RUN-TIME RATIO: 390/86=4.4
 CORE USED: 26K (51 PAGES)