

11/03

LSI-11 EIS INST TEST
CVKABBO

AH-S366B-MC
FICHE 1 OF 1

MAR 1982
COPYRIGHT © 75-81
MADE IN USA



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

000000

.REPT 0

IDENTIFICATION

PRODUCT CODE: AC-8190B-MC
PRODUCT NAME: CVKABBO LSI-11 EIS INST TEST
DATE CREATED: JULY, 1981
MAINTAINER: DIAGNOSTIC GROUP

COPYRIGHT (C) 1975,1981
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

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

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

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

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

CONTENTS

1.	ABSTRACT
2.	REQUIREMENTS
2.1	EQUIPMENT
2.2	STORAGE
2.3	PRELIMINARY PROGRAMS
3.	LOADING PROCEDURE
4.	STARTING PROCEDURE
4.1	CONTROL SWITCH SETTINGS
4.2	STARTING ADDRESS
4.3	PROGRAM AND/OR OPERATOR ACTION
5.	OPERATING PROCEDURE
5.1	SWITCH SETTINGS
5.2	SUBROUTINE ABSTRACTS
6.	ERRORS
6.1	ERROR PRINTOUT
6.2	ERROR RECOVERY
7.	RESTRICTIONS
8.	MISCELLANEOUS
8.1	EXECUTION TIME
8.2	STACK POINTER
8.3	PASS COUNTER
8.4	TEST NUMBER
8.5	POWER FAIL
9.	PROGRAM DESCRIPTION

103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158

1. ABSTRACT

THIS PROGRAM TESTS THE LSI-11 EXTENDED INSTRUCTION SET <ASH, ASHC, MUL, AND DIV> OPTION USING REGISTERS 0-5 AT-LEAST ONCE WITH EACH INSTRUCTION. IT IS ALSO CHECKED THAT EXTENDED INSTRUCTIONS CAN BE INTERRUPTED (BY THE CONSOLE TELETYPE) [HOWEVER THIS TEST WILL NOT BE EXECUTED WHEN BIT 5 OF \$ENVM BYTE IS HIGH]. THE PROGRAM SHOULD BE RUN FOR AT LEAST 2 PASSES WITH ALL SWITCHES LOW. THE PROGRAM IS DESIGNED TO RUN UNDER APT. AND ACT. SYSTEMS. WHEN RUNNING UNDER APT WITH BIT 5 OF \$ENVM LOW IT WILL BE REQUIRED TO HAVE A SLU WITH TTY REGISTERS HAVING ADDRESSES OF 176560-66 AND INTERRUPT VECTORS OF 70 FOR RECEIVER AND 74 FOR TRANSMITTER

2. REQUIREMENTS

2.1 EQUIPMENT

LSI-11 STANDARD COMPUTER WITH EIS OPTION AND 4K OF MEMORY

2.2 STORAGE

PROGRAM STORAGE - THE ROUTINES USE MEMORY 0 - 17500

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1 (ALL LOW FOR WORST CASE TESTING)

159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211

4.2 STARTING ADDRESS

AFTER LOADING THE PROGRAM IT SHOULD ALWAYS BE STARTED AT 200. IF IT IS DESIRED TO SAVE THE PASS COUNTER THEN THE PROGRAM SHOULD BE RESTARTED AT LOCATION RESTRT [I.E. 222] OTHERWISE THE PROGRAM CAN BE RESTARTED AT 200

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 STAND ALONE

- 1) PLACE LTC SWITCH IN OFF POSITION (IF APPLICABLE).
- 2) LOAD PROGRAM INTO MEMORY USING ABS LOADER OR XXDP+ (.R VKAB??).
- 3) SET SWITCHES (SEE SEC 5.1) ALL LOW FOR WORST CASE.
- 4) TYPE 200G IF USING ABS LOADER.
- 5) THE PROGRAM WILL LOOP AND 'END PASS' WILL BE TYPED AFTER COMPLETION OF FIRST PASS AND EVERY 4TH PASS. HOWEVER TYPE OUT WILL BE SUPPRESSED IF BIT 5 OF LOCATION \$ENVM IS HIGH
- 6) A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN.

4.3.2 UNDER APT

LOAD THE PROGRAM AND START AFTER SETTING THE DESIRED SWITCHES (SEE SEC. 5.1). HOWEVER IF THE DIAGNOSTIC IS RUN UNDER APT. WITH BIT 5 OF \$ENVM LOW THEN IT WILL BE REQUIRED THAT A SLU WITH TTY REGISTERS HAVING ADDRESSES OF 176560-66, AND INTERRUPT VECTORS OF 70 FOR RECEIVER AND 74 FOR TRANSMITTER BE PRESENT, IT WILL ALSO BE REQUIRED TO CHANGE THE PASS TIME FROM 5 SEC. TO 15 SECONDS AND THE TEST TIME FROM 3 TO 10 SECONDS

5. OPERATING PROCEDURE

5.1 SWITCH SETTINGS

A 16 BIT LOCATION CALLED \$SWREG (I.E. LOCATION 422) HAS BEEN USED TO GIVE THE FOLLOWING OPTIONS BY INSERTING A 1 IN THEIR RESPECTIVE POSITIONS

BIT #	OCTAL VALUE	FUNCTION
15	100000.....	HALT ON ERROR
13	020000.....	INHIBIT PRINTOUT

AN 8 BIT BYTE \$ENVM [I.E. LOCATION 421] HAS BEEN USED TO DEFINE THE OPERATING MODE. ALL TYPEOUTS CAN BE SUPPRESSED BY MAKING BIT 5 OF BYTE \$ENVM HIGH, IN OTHER WORDS BY PLACING A 20000 IN LOCATION 420

213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268

5.2 SUBROUTINE ABSTRACTS

5.2.1 HALT ROUTINE

THIS ROUTINE CALLED VIA JSR INSTRUCTION IS USED EACH TIME AN ERROR IS SEEN AND AN ERROR MESSAGE OF THE FORMAT GIVEN IN SEC. 6.1 IS TYPED OUT UNLESS SUPRESSED BY THE SWITCHES DEFINED IN SEC. 5.1

5.2.2 TRAP CATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0-776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR +2.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ADR ERRNM

WHERE:

ADR = ADDRESS OF ERROR

ERRNM = ERROR NUMBER

IN MOST CASES THE COMMENT BESIDE THE CALL FOR HALT SUBROUTINE TELLS WHAT WAS BEING CHECKED AND WHAT WAS EXPECTED. ALL PRINTOUTS WILL BE SUPPRESSED WHEN BIT 5 OF LOCATION \$ENVM IS HIGH. WHILE RUNNING UNDER APT THE DIAGNOSTIC WILL NOT SUPPORT SPOOLING OF CONSOLE OUTPUTS.

6.2 ERROR RECOVERY

RESTART AT 200 OR 222 (SEE SEC 4.2)

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317

8.1 EXECUTION TIME

DUE TO THE RANDOM CHARACTERISTIC OF THE INTERRUPT TESTS, THE EXECUTION TIME CAN BE 5 SECONDS OR MORE PER PASS. HOWEVER, NORMALLY FIRST "END PASS" WILL BE TYPED WITHIN 5 SECONDS AND WITHIN 50 SECONDS FOR EVERY CONSECUTIVE 10 PASSES

8.2 STACK POINTER

STACK IS INITIALLY SET TO 600

8.3 PASS COUNT

A 16 BIT LOCATION "\$PASS" (I.E. LOCATION 406) IS USED TO KEEP PASS COUNT. IT CAN BE CLEARED BY RESTARTING THE PROGRAM AT 200

8.4 TEST NUMBER

A 16 BIT LOCATION "\$TESTN" (I.E. LOCATION 404) IS USED TO KEEP TRACK OF THE TEST NUMBER, UPPER BYTE OF THIS LOCATION GIVES THE ITERATION NUMBER AND THE LOWER BYTE THE TEST THAT WAS BEING EXECUTED

8.5 POWER FAIL

THE DIAGNOSTIC CAN BE POWER FAILED WITH NO ERRORS. TO USE, START THE TEST AS USUAL AND POWER DOWN THEN UP AT ANY TIME, THE PROGRAM SHOULD RESTART FROM TEST 0 AFTER TYPING "POWER" WITH NO ERRORS. HOWEVER IF THE PROGRAM IS STORED IN A MOS MEMORY THAT CAN NOT HOLD DATA WITH POWER DOWN THEN THE PROGRAM WILL NOT RECOVER FROM A POWER FAIL

9. PROGRAM DESCRIPTION

THIS PROGRAM TESTS ALL THE EIS INSTRUCTIONS OF THE LSI-11 FOR ASH AND ASHC INSTRUCTIONS EVERY EVEN PASS IS EXECUTED WITH DESTINATION MODE 0 FOR ALL REGISTERS AND EVERY ODD PASS WITH DESTINATION MODE OF 67. THE DIAGNOSTIC DOES NOT MAKE A PASS WITH T BIT SET.

.ENDR

319
320
321
322
325
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
326
327

000001
160000

```
.ABS  
.NLIST MD,MC,CND  
.LIST ME  
.TITLE DVKABA  
:*COPYRIGHT (C) AUGUST 1975  
:*DIGITAL EQUIPMENT CORP.  
:*MAYNARD, MASS. 01754  
:*  
:*PROGRAM BY PERVEZ ZAKI  
:*  
:*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC  
:*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.  
:*  
$TN=1  
$SWR=160000      ;:HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOU
```



```
329  
330 ;:*****  
331  
332      000000      .=0      ;TRAP CATCHER 0 - 776  
333  
334 ;:*****  
341  
342      .SBTTL ACT11 HOOKS  
(1)  
(2) ;:*****  
(1) ;HOOKS REQUIRED BY ACT11  
(1)      001000      $SVPC=.      ;SAVE PC  
(1)      000046      .=46  
(1) 000046 017032      $ENDAD      ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP  
(1)      000052      .=52  
(1) 000052 000000      .WORD 0      ;;2)SET LOC.52 TO ZERO  
(1)      001000      .= $SVPC      ;; RESTORE PC  
343  
344      000000      DUMMY= 0  
345      000001      ERRNM= 1  
346      000051      F= 51  
347      000176      N= 176  
348      000007      PC= %7  
349      000006      SP= %6  
350      010701      SCOPE= 10701  
351      010701      SCOPE1= 10701  
352      010703      SCOPE3= 10703  
353      001000      SW09= 1000  
354      002000      SW10= 2000  
355      004000      SW11= 4000  
356      010000      SW12= 10000  
357      000004      TYPE= 10T  
358  
359  
360      000020  
361 000020 017300      .-20  
      $TYPE
```



```

716          000200          . = 200
717 000200 012737 017064 000024      MOV    #SPWRDN,@#24      ;PREPARE TO SERVICE POWER DOWN ROUTINE
718 000206 012700 000410          MOV    #SDEVCT,R0      ;PREPARE TO INITIALIZE THE STACK
719 000212 005040          2$:    CLR    -(R0)
720 000214 022700 000400          CMP    #SMAIL,R0
721 000220 001374          BNE   2$
722 000222 000167 000352      RESTRT: JMP   BEGIN
723
724          000600          . = 600
725
726 000600 012705 000404      BEGIN: MOV    #STESTN,R5      ;MAKE R5 POINT TO THE LOCATION $TESTN
727 000604 005037 000430          CLR    @#COUNT      ;CLEAR THE COUNTER
728 000610 012715 000001          MOV    #1,(R5)      ;INITIALIZE TEST NUMBER
729 000614 012706 000600          MOV    #BEGIN,SP     ;** STACK AT BEGIN **
733 000620          MTPS   #0          ;PLACE #0 IN PSW
(1) 000620 106427          .WORD 106400!...C
737 000624 132737 000001 000420      BITB  #1,@#SENV      ;ARE WE UNDER APT ?
738 000632 001410          BEQ   2$            ;IF NOT THEN GO TO 2$
739 000634 012700 000510          MOV    #STPS+2,R0    ;OTHERWISE SET FOR OTHER SLU
740 000640 012740 176564          MOV    #176564,-(R0)
741 000644 012740 176566          MOV    #176566,-(R0)
742 000650 012740 000074          MOV    #74,-(R0)
743 000654 012737 000001 000434 2$:    MOV    #1,@#TEMP1     ;TEMP1=1
744 000662 005037 000436          CLR    @#TEMP2      ;TEMP2=0
745 000666 012737 000001 000440          MOV    #1,@#TEMP3     ;TEMP3=1
746 000674 005037 000442          CLR    @#TEMP4      ;TEMP4 0
747
748
752
753          ;*****
754          ;      ASH INSTRUCTION TESTS
755          ;*****
756
757
758
759
760
761          ;*****
762          ;      TESTS 1-36
763          ;*****
764
765 000700 010701          START: SCOPE1
766 000702 013700 000434          MOV    @#TEMP1,%0    ;LOAD R0 WITH THE CONTENTS OF TEMP1
767 000706 032737 000001 000406      BIT    #1,@#SPASS    ;IS IT AN EVEN PASS ?
768 000714 001004          BNE   2$            ;IF NOT THEN GO TO 2$
769 000716 013701 000436          MOV    @#TEMP2,R1    ;OTHERWISE EXECUTE THE INSTRUCTION
770          ;IN MODE 0 USING R1
771 000722 072001          ASH   R1,R0
772 000724 000402          BR   4$
773 000726 072067 177504          2$:    ASH   TEMP2,%0      ;SHIFT R0 BY THE NUMBER SPECIFIED BY TEMP2
777 000732          4$:    MFPS   @#PSWORD     ;SAVE PS
(1) 000732 106737          .WORD 106700!...C
781 000736 123737 000442 000432      CMPB  @#TEMP4,@#PSWORD;IS THE PS - TEMP4 ?
782 000744 001403          BEQ   .+10
783 000746 004767 016142          JSR   PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(2)          ;THE PS IS NOT EQUAL TO 0
  
```

(2)	000752	000001			1				
784	000754	005237	000430		INC	@#COUNT		:INCREMENT THE COUNTER	
785	000760	023700	000440		CMP	@#TEMP3,%0		:IS THE RESULT IN R0 EQUAL TO TEMP3?	
786	000764	001403			BEQ	+.10			
787	000766			6\$:					
(2)	000766	004767	016122		JSR	PC,\$HLT		:SEEN AN ERROR, GO TO TH HALT ROUTINE	
(2)								:EITHER INCORRECT R0 OR INCORRECT SEQUENCE	
(2)	000772	000002			2				
788	000774	021537	000430		CMP	(R5),@#COUNT		:IS THE TEST NUMBER EQUAL TO THE	
789								:COUNTER?	
790	001000	001372			BNE	6\$:IF NOT GO TO THE HLT ABOVE	
791	001002	005215			INC	(R5)			
792	001004	010701			SCOPE1				
793	001006	021527	000037		CMP	(R5),#37		:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT	
794								:BY 14. AND RIGHT BY 14.?	
795	001012	002011			BGE	8\$			
796	001014	005237	000436		INC	@#TEMP2			
797	001020	006367	1,7414		ASL	TEMP3		:SHIFT TEMP3 LEFT.	
798	001024	021527	000020		CMP	(R5),#20		:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?	
799	001030	001004			BNE	REG1			
800	001032	000167	000764		JMP	NEGAT		:IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT	
801	001036	004767	001006		JSR	PC,TST37		:IF SO GO AND CONTINUE THE REST OF THE PROGRAM	
802	001042	010703		8\$:					
803	001044	013701	000434	REG1:	SCOPE3				
804	001050	032737	000001	000406	MOV	@#TEMP1,%1		:LOAD R1 WITH THE CONTENTS OF TEMP1	
805	001056	001004			BIT	#1,@#SPASS		:IS IT AN EVEN PASS ?	
806	001060	013702	000436		BNE	2\$:IF NOT THEN GO TO 2\$	
807	001064	072102			MOV	@#TEMP2,R2		:OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0	
808	001066	000402			ASH	R2,R1		:USING R1	
809	001070	072167	177342		BR	4\$			
813	001074			2\$:	ASH	TEMP2,%1		:SHIFT R1 BY THE NUMBER SPECIFIED BY TEMP2	
(1)	001074	106737		4\$:	MFPS	@#PSWORD		:SAVE PS	
817	001100	123737	000442	000432	.WORD	106700!..C			
818	001106	001403			CMPB	@#TEMP4,@#PSWORD		:IS THE PS = TEMP4 ?	
819	001110	004767	016000		BEQ	+.10			
(2)					JSR	PC,\$HLT		:SEEN AN ERROR, GO TO TH HALT ROUTINE	
(2)								:THE PS IS NOT EQUAL TO 0	
(2)	001114	000003			3				
820	001116	005237	000430		INC	@#COUNT		:INCREMENT THE COUNTER	
821	001122	023701	000440		CMP	@#TEMP3,%1		:IS THE RESULT IN R1 EQUAL TO TEMP3?	
822	001126	001403			BEQ	+.10			
823	001130			6\$:					
(2)	001130	004767	015760		JSR	PC,\$HLT		:SEEN AN ERROR, GO TO TH HALT ROUTINE	
(2)								:EITHER INCORRECT R1 OR INCORRECT SEQUENCE	
(2)	001134	000004			4				
824	001136	021537	000430		CMP	(R5),@#COUNT		:IS THE TEST NUMBER EQUAL TO THE COUNTER?	
825	001142	001372			BNE	6\$:IF NOT GO TO THE HLT ABOVE	
826	001144	005215			INC	(R5)			
827	001146	010703			SCOPE3				
828	001150	021527	000037		CMP	(R5),#37		:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT	
829								:BY 14. AND RIGHT BY 14.?	
830	001154	002011			BGE	8\$			
831	001156	005237	000436		INC	@#TEMP2			
832	001162	006367	177252		ASL	TEMP3		:SHIFT TEMP3 LEFT	
833	001166	021527	000020		CMP	(R5),#20		:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?	
834	001172	001004			BNE	REG2			
835	001174	000167	000622		JMP	NEGAT		:IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT	

DVKABA MACY 1 30(1046) 14-SEP-81 16:32 PAGE 4-8
 CVKABB.P11 14-SEP-81 16:31 ASH INSTRUCTION TESTS

SEQ 0013

836	001200	004767	000644	8\$:	JSR	PC,TST37	:IF SO GO AND CONTINUE THE REST OF THE PROGRAM
837	001204	010701		REG2:	SCOPE1		
838	001206	013702	000434		MOV	@#TEMP1,%2	:LOAD R2 WITH THE CONTENTS OF TEMP1
839	001212	032737	000001 000406		BIT	#1,@#SPASS	:IS IT AN EVEN PASS ?
840	001220	001004			BNE	2\$:IF NOT THEN GO TO 2\$
841	001222	013703	000436		MOV	@#TEMP2,R3	:OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
842	001226	072203			ASH	R3,R2	:USING R2
843	001230	000402			BR	4\$	
844	001232	072267	177200	2\$:	ASH	TEMP2,%2	:SHIFT R2 BY THE NUMBER SPECIFIED BY TEMP2
848	001236			4\$:	MFPS	@#PSWORD	:SAVE PS
(1)	001236	106737			.WORD	106700!..C	
852	001242	123737	000442 000432		CMPB	@#TEMP4,@#PSWORD	:IS THE PS = TEMP4 ?
853	001250	001403			BEQ	..+10	
854	001252	004767	015636		JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(2)							:THE PS IS NOT EQUAL TO 0
(2)	001256	000005			5		
855	001260	005237	000430		INC	@#COUNT	
856	001264	023702	000440		CMP	@#TEMP3,%2	:IS THE RESULT IN R2 EQUAL TO TEMP3?
857	001270	001403			BEQ	..+10	
858	001272			6\$:			
(2)	001272	004767	015616		JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(2)							:EITHER INCORRECT R2 OR INCORRECT SEQUENCE
(2)	001276	000006			6		
859	001300	021537	000430		CMP	(R5),@#COUNT	:IS THE TEST NUMBER EQUAL TO THE COUNTER?
860	001304	001372			BNE	6\$:IF NOT GO TO THE HLT ABOVE
861	001306	005215			INC	(R5)	
862	001310	010701			SCOPE1		
863	001312	021527	000037		CMP	(R5),#37	:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
864							:LEFT BY 14, AND RIGHT BY 14.?
865	001316	002011			BGE	8\$	
866	001320	005237	000436		INC	@#TEMP2	
867	001324	006367	177110		ASL	TEMP3	:SHIFTED TEMP3 LEFT
868	001330	021527	000020		CMP	(R5),#20	:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
869	001334	001004			BNE	REG3	
870	001336	000167	000460		JMP	NEGAT	:IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
871	001342	004767	000502	8\$:	JSR	PC,TST37	:IF SO GO AND CONTINUE THE REST OF THE PROGRAM
872	001346	010701		REG3:	SCOPE1		
873	001350	013703	000434		MOV	@#TEMP1,%3	:LOAD R3 WITH THE CONTENTS OF TEMP1
874	001354	032737	000001 000406		BIT	#1,@#SPASS	:IS IT AN EVEN PASS ?
875	001362	001004			BNE	2\$:IF NOT THEN GO TO 2\$
876	001364	013704	000436		MOV	@#TEMP2,R4	:OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
877	001370	072304			ASH	R4,R3	:USING R3
878	001372	000402			BR	4\$	
879	001374	072367	177036	2\$:	ASH	TEMP2,%3	:SHIFT R3 BY THE NUMBER SPECIFIED BY TEMP2
883	001400			4\$:	MFPS	@#PSWORD	:SAVE PS
(1)	001400	106737			.WORD	106700!..C	
887	001404	123737	000442 000432		CMPB	@#TEMP4,@#PSWORD	:IS THE PS = TEMP4 ?
888	001412	001403			BEQ	..+10	
889	001414	004767	015474		JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(2)							:THE PS IS NOT EQUAL TO 0.
(2)	001420	000007			7		
890	001422	005237	000430		INC	@#COUNT	
891	001426	023703	000440		CMP	@#TEMP3,%3	:IS THE RESULT IN R3 EQUAL TO TEMP3?
892	001432	001403			BEQ	..+10	
893	001434			6\$:			
(2)	001434	004767	015454		JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE


```

948 001670 001004      BNE      2$      ;IF NOT THEN GO TO 2$
949 001672 013700 000436  MOV      @#TEMP2,R0 ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
950 001676 072500      ASH      R0,R5    ;USING R5
951 001700 000402      BR       4$
952 001702 072567 176530  2$:     ASH      TEMP2,%5 ;SHIFT R5 BY THE NUMBER SPECIFIED BY TEMP2
956 001706      4$:     MFPS     @#PSWORD ;SAVE PS
(1) 001706 106737      .WORD   106700...C
960 001712 123737 000442 000432  CMPB    @#TEMP4,@#PSWORD;IS PS = TEMP4 ?
961 001720 001403      BEQ
962 001722 004767 015166  JSR      PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(2)                                     ;THE PS IS NOT EQUAL TO 0.
(2) 001726 000013      13
963 001730 005237 000430  INC      @#COUNT
964 001734 023705 000440  CMP      @#TEMP3,%5 ;IS THE RESULT IN R5 EQUAL TO TEMP3?
965 001740 001403      BEQ
966 001742      6$:     JSR      PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(2) 001742 004767 015146  ;EITHER INCORRECT R5 OR INCORRECT SEQUENCE
(2) 001746 000014      14
967 001750 021137 000430  CMP      (R1),@#COUNT ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
968 001754 001372      BNE      6$      ;IF NOT GO TO THE HLT ABOVE
969 001756 010105      MOV      R1,R5    ;RESTORE R5
970 001760 005215      INC      (R5)
971 001762 010701      SCOPE1
972 001764 021527 000037  CMP      (R5),#37 ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
973                                     ;LEFT BY 14. AND RIGHT BY 14.?
974 001770 002010      BGE      8$      ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
975 001772 005237 000436  INC      @#TEMP2
976 001776 006367 176436  ASL      TEMP3    ;SHIFT TEMP3 LEFT
977 002002 021527 000020  CMP      (R5),#20 ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
978 002006 001405      BEQ      NEGAT   ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
979 002010 000402      BR       10$
980 002012 004767 000032  8$:     JSR      PC,TST37
981 002016 000167 176656  10$:    JMP      START   ;GO BACK TO START
982 002022 012737 040000 000434  NEGAT:  MOV      #40000,@#TEMP1 ;TEMP1=40000
983 002030 012737 177762 000436  MOV      #177762,@#TEMP2 ;TEMP2=177762
984 002036 012737 000001 000440  MOV      #1,@#TEMP3   ;TEMP3=1
985 002044 000167 176630      JMP      START
986 002050 021527 000037  TST37:  CMP      (R5),#37 ;IS IT TEST 37?
987 002054 001013      BNE      TST40   ;IF NOT THEN TRY TEST 40
988 002056 005037 000434      CLR      @#TEMP1 ;0
989 002062 012737 000020 000436  MOV      #16,@#TEMP2 ;SHIFTED BY 16
990 002070 005037 000440      CLR      @#TEMP3 ;IS=0
991 002074 012737 000004 000442  MOV      #4,@#TEMP4  ;AND PS=4
992 002102 000207      RTS      PC
993 002104 021527 000040  TST40:  CMP      (R5),#40 ;IS IT TEST 40?
994 002110 001003      BNE      TST41   ;IF NOT THEN TRY TEST 41
995 002112 005037 000436      CLR      @#TEMP2 ;0 SHIFTED BY 0=0 AND PS=4
996 002116 000207      RTS      PC
997 002120 021527 000041  TST41:  CMP      (R5),#41 ;IS IT TEST 41?
998 002124 001004      BNE      TST42   ;IF NOT THEN TRY TEST 42
999 002126 012737 177760 000436  MOV      #-16,@#TEMP2 ;0 SHIFTED BY -16.=0 AND PS=4
1000 002134 000207      RTS      PC
1001 002136 021527 000042  TST42:  CMP      (R5),#42 ;IS IT TEST 42?
1002 002142 001013      BNE      TST43   ;IF NOT THEN TRY TEST 43
1003 002144 012737 100000 000434  MOV      #100000,@#TEMP1 ;100000

```

DVKABA MACY 1 30(1046) 14-SEP-81 16:32 PAGE 4-11
 CVKABB.P11 14-SEP-81 16:31 ASH INSTRUCTION TESTS

1004	002152	005237	000436		INC	@#TEMP2	:SHIFTED BY -15
1005	002156	005337	000440		DEC	@#TEMP3	:IS=-1
1006	002162	012737	000010	000442	MOV	#10,@#TEMP4	:AND PS=10
1007	002170	000207			RTS	PC	
1008	002172	021527	000043		TST43: CMP	(R5),#43	:IS IT TEST 43?
1009	002176	001012			BNE	TST44	:IF NOT THEN IF NOT THEN TRY TEST 44
1010	002200	012737	125252	000434	MOV	#125252,@#TEMP1	:125252
1011	002206	012737	177777	000436	MOV	#-1,@#TEMP2	:SHIFTED BY -1
1012	002214	012737	152525	000440	MOV	#152525,@#TEMP3	:IS=152525 AND PS=10
1013	002222	000207			RTS	PC	
1014	002224	021527	000044		TST44: CMP	(R5),#44	:IS IT TEST 44?
1015	002230	001012			BNE	TST45	:IF NOT THEN TRY TEST 45
1016	002232	012737	000001	000436	MOV	#1,@#TEMP2	:125252 SHIFTED BY 1
1017	002240	012737	052524	000440	MOV	#52524,@#TEMP3	:IS=52524
1018	002246	012737	000003	000442	MOV	#3,@#TEMP4	:AND PS=3
1019	002254	000207			RTS	PC	
1020	002256	021527	000045		TST45: CMP	(R5),#45	:IS IT TEST 45?
1021	002262	001012			BNE	TST46	:IF NOT THEN TRY TEST 46
1022	002264	012737	177776	000436	MOV	#-2,@#TEMP2	:125252 SHIFTED BY -2
1023	002272	012737	165252	000440	MOV	#165252,@#TEMP3	:IS=165252
1024	002300	012737	000011	000442	MOV	#11,@#TEMP4	:AND PS=11
1025	002306	000207			RTS	PC	
1026	002310	021527	000046		TST46: CMP	(R5),#46	:IS IT TEST 46?
1027	002314	001014			BNE	TST47	:IF NOT THEN TRY TEST 47
1028	002316	012737	177777	000434	MOV	#-1,@#TEMP1	: -1
1029	002324	012737	000020	000436	MOV	#16,@#TEMP2	:SHIFTED BY 15.
1030	002332	005037	000440		CLR	@#TEMP3	:IS=0
1031	002336	012737	000007	000442	MOV	#7,@#TEMP4	:AND PS=7
1032	002344	000207			RTS	PC	
1033	002346	021527	000047		TST47: CMP	(R5),#47	:IS IT TEST 47?
1034	002352	001011			BNE	TST50	:IF NOT THEN TRY TEST 50
1035	002354	005337	000436		DEC	@#TEMP2	: -1 SHIFTED BY 15
1036	002360	012737	100000	000440	MOV	#100000,@#TEMP3	:IS=100000
1037	002366	012737	000011	000442	MOV	#11,@#TEMP4	:AND PS=11
1038	002374	000207			RTS	PC	
1039	002376	021527	000050		TST50: CMP	(R5),#50	:IS IT TEST 50
1040	002402	001007			BNE	ENT51	:IF NOT THEN TRY TEST 51
1041	002404	012737	137777	000434	MOV	#137777,@#TEMP1	:137777 SHIFTED BY 15. IS=100000
1042	002412	012737	000013	000442	MOV	#13,@#TEMP4	:AND PS=13
1043	002420	000207			RTS	PC	
1044	002422	021527	000051		ENT51: CMP	(R5),#51	:IS IT ENTERING TEST 51?
1045	002426	001405			BEQ	+10	
1046	002430	004767	014460		JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE :TEST NUMBER GOOFED
(2)							
(2)	002434	000015				15	
1047							
1048	002436	005726			TST	(SP)+	:RESTORE STACK POINTER
1049	002440	012704	177771		MOV	#-7,%4	
1050	002444	012702	000454		MOV	#S1,%2	
1051	002450	012703	000456		MOV	#S2,%3	

1054
 (1)
 (1)
 (1)
 (1) 002614 010701
 (1) 002616 012700 125252
 (1) 002622 072037 000454
 (1) 002626
 (2) 002626 106737
 (1) 002632 122737 000010 000432
 (1) 002640 001403
 (3) 002642 004767 014246
 (3)
 (3) 002646 000022
 (1) 002650 022700 177525
 (1) 002654 001403
 (1) 002656
 (3) 002656 004767 014232
 (3)
 (3) 002662 000023
 (1) 002664 021527 000053
 (1) 002670 001372
 (1) 002672 005215
 (1)
 (1)
 (1)

 :TEST:53 LSI-11 ASH 125252 SHIFTED BY @#S1 = 177525 PS = 10
 :*****

TST53: SCOPE1
 MOV #125252,%0 ;LOAD R0 WITH 125252
 ASH @#S1,%0 ;SHIFT R0 BY @#S1
 MFPS @#PSWORD ;SAVE PS
 .WORD 106700!..C
 CMPB #10,@#PSWORD ;IS THE PS 10?
 BEQ .+10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;THE PS IS NOT EQUAL TO 10
 22
 CMP #177525,%0 ;IS THE RESULT 177525?
 BEQ .+10
 1\$: JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
 23
 CMP (R5),#53 ;IS \$TESTN = #53
 BNE 1\$;IF NOT THEN GO TO HLT ABOVE
 INC (R5)

1055
 (1)
 (1)
 (1)
 (1) 002674 010701
 (1) 002676 012700 125252
 (1) 002702 072012
 (1) 002704
 (2) 002704 106737
 (1) 002710 122737 000010 000432
 (1) 002716 001403
 (3) 002720 004767 014170
 (3)
 (3) 002724 000024
 (1) 002726 022700 177525
 (1) 002732 001403
 (1) 002734
 (3) 002734 004767 014154
 (3)
 (3) 002740 000025
 (1) 002742 021527 000054
 (1) 002746 001372
 (1) 002750 005215
 (1)
 (1)
 (1)

 :TEST:54 LSI-11 ASH 125252 SHIFTED BY (2) = 177525 PS = 10
 :*****

TST54: SCOPE1
 MOV #125252,%0 ;LOAD R0 WITH 125252
 ASH (2),%0 ;SHIFT R0 BY (2)
 MFPS @#PSWORD ;SAVE PS
 .WORD 106700!..C
 CMPB #10,@#PSWORD ;IS THE PS 10?
 BEQ .+10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;THE PS IS NOT EQUAL TO 10
 24
 CMP #177525,%0 ;IS THE RESULT 177525?
 BEQ .+10
 1\$: JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
 25
 CMP (R5),#54 ;IS \$TESTN = #54
 BNE 1\$;IF NOT THEN GO TO HLT ABOVE
 INC (R5)

1056

:TEST:55 LSI-11 ASH 125252 SHIFTED BY (2)+ 177525 PS 10

(1) 002752 010701
(1) 002754 012700 125252
(1) 002760 072022
(1) 002762
(2) 002762 106737
(1) 002766 122737 000010 000432
(1) 002774 001403
(3) 002776 004767 014112
(3)
(3) 003002 000026
(1) 003004 022700 177525
(1) 003010 001403
(1) 003012
(3) 003012 004767 014076
(3)
(3) 003016 000027
(1) 003020 021527 000055
(1) 003024 001372
(1) 003026 005215
(1)
(1)
(1)

TST55: SCOPE1
MOV #125252,%0 ;LOAD R0 WITH 125252
ASH (2)+,%0 ;SHIFT R0 BY (2)+
MFPS @#PSWORD ;SAVE PS
.WORD 106700!..C
CMPB #10,@#PSWORD ;IS THE PS 10?
BEQ .+10
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 10
26
CMP #177525,%0 ;IS THE RESULT 177525?
BEQ .+10
1\$:
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
27
CMP (R5),#55 ;IS \$TESTN = #55
BNE 1\$;IF NOT THEN GO TO HLT ABOVE
INC (R5)

1057

:TEST:56 LSI-11 ASH 125252 SHIFTED BY -(2) 177525 PS 10

(1) 003030 010701
(1) 003032 012700 125252
(1) 003036 072042
(1) 003040
(2) 003040 106737
(1) 003044 122737 000010 000432
(1) 003052 001403
(3) 003054 004767 014034
(3)
(3) 003060 000030
(1) 003062 022700 177525
(1) 003066 001403
(1) 003070
(3) 003070 004767 014020
(3)
(3) 003074 000031
(1) 003076 021527 000056
(1) 003102 001372
(1) 003104 005215
(1)
(1)
(1)

TST56: SCOPE1
MOV #125252,%0 ;LOAD R0 WITH 125252
ASH -(2),%0 ;SHIFT R0 BY -(2)
MFPS @#PSWORD ;SAVE PS
.WORD 106700!..C
CMPB #10,@#PSWORD ;IS THE PS 10?
BEQ .+10
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 10
30
CMP #177525,%0 ;IS THE RESULT 177525?
BEQ .+10
1\$:
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
31
CMP (R5),#56 ;IS \$TESTN = #56
BNE 1\$;IF NOT THEN GO TO HLT ABOVE
INC (R5)

1060
(1)
(1)
(1)
(1)
(2)
(1)
(1)
(3)
(3)
(3)
(1)
(1)
(1)
(1)
(3)
(3)
(1)
(1)
(1)
(1)
(1)
(1)
1061
(1)
(1)
(1)
(1)
(1)
(1)
(2)
(1)
(1)
(3)
(3)
(1)
(1)
(1)
(1)
(3)
(3)
(1)
(1)
(1)
(1)
(1)
(1)
1065
1066
1067
1068

003246 010701
003250 012700 125252
003254 072033
003256 106737
003262 122737 000010 000432
003270 001403
003272 004767 013616
003276 000036
003300 022700 177525
003304 001403
003306 004767 013602
003312 000037
003314 021527 000061
003320 001372
003322 005215

003324 010701
003326 012700 125252
003332 072053
003334 106737
003340 122737 000010 000432
003346 001403
003350 004767 013540
003354 000040
003356 022700 177525
003362 001403
003364 004767 013524
003370 000041
003372 021527 000062
003376 001372
003400 005215

:TEST:61 LSI-11 ASH 125252 SHIFTED BY @ (3)+ = 177525 PS = 10

TST61: SCOPE1
MOV #125252,%0 ;LOAD R0 WITH 125252
ASH @ (3),%0 ;SHIFT R0 BY @ (3)+
MFPS @#PSWORD ;SAVE PS
.WORD 106700!...C
CMPB #10,@#PSWORD ;IS THE PS 10?
BEQ .+10
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 10

36
CMP #177525,%0 ;IS THE RESULT 177525?
BEQ .+10

1\$: JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE

37
CMP (R5),#61 ;IS \$TESTN = #61
BNE 1\$;IF NOT THEN GO TO HLT ABOVE
INC (R5)

:TEST:62 LSI-11 ASH 125252 SHIFTED BY @-(3) = 177525 PS = 10

TST62: SCOPE1
MOV #125252,%0 ;LOAD R0 WITH 125252
ASH @-(3),%0 ;SHIFT R0 BY @-(3)
MFPS @#PSWORD ;SAVE PS
.WORD 106700!...C
CMPB #10,@#PSWORD ;IS THE PS 10?
BEQ .+10
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 10

40
CMP #177525,%0 ;IS THE RESULT 177525?
BEQ .+10

1\$: JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE

41
CMP (R5),#62 ;IS \$TESTN = #62
BNE 1\$;IF NOT THEN GO TO HLT ABOVE
INC (R5)

: ASHC INSTRUCTION TESTS

1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1099
(1)
1103
1104
1105
(2)
(2)
1106
1107
1108
1109
(2)
(2)
1110
1111
1112
1113
(2)
(2)
1114
1115
1116
1117
(2)
(2)
1118
1119
1120
1121

003402 012737 000062 000430
003410 005037 000434
003414 012737 000001 000436
003422 005037 000440
003426 005037 000442
003432 012737 000001 000444
003440 005037 000446

003444 010703
003446 010502
003450 013700 000434
003454 013701 000436
003460 000241
003462 032737 000001 000406
003470 001004
003472 013705 000440
003476 073005
003500 000402
003502 073067 174732
003506
(1) 003506 106737
003512 123737 000446 000432
003520 001403
003522 004767 013366

(2) 003526 000042
003530 005237 000430
003534 023700 000442
003540 001403
003542 004767 013346

(2) 003546 000043
003550 023701 000444
003554 001403

003556 004767 013332

(2) 003562 000044
003564 010205
003566 021537 000430
003572 001403
003574 004767 013314

(2) 003600 000045
003602 005215
003604 021527 000160
003610 002014
003612 005237 000440

```

:*****
:TESTS 63-157
:*****

MOV #62,@#COUNT
CLR @#TEMP1 ;TEMP1=0
MOV #1,@#TEMP2 ;TEMP2=1
CLR @#TEMP3 ;TEMP3=0
CLR @#TEMP4 ;TEMP4=0
MOV #1,@#TEMP5 ;TEMP5=1
CLR @#TEMP6 ;0 1 SHIFTED BY 0=0 1, PS=0

REG01: SCOPE3
MOV R5,R2 ;SAVE R5
MOV @#TEMP1,%0 ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 0
MOV @#TEMP2,%0.1 ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 1
CLC
BIT #1,@#SPASS ;IS IT AN EVEN PASS ?
BNE 2$ ;IF NOT THEN GO TO 2$
MOV @#TEMP3,R5 ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
ASHC R5,R0 ;USING R0
BR 4$
2$: ASHC TEMP3,%0 ;ASHC REGISTER 0 BY THE CONTENTS OF TEMP3
4$: MFPS @#PSWORD ;SAVE PS
.WORD 106700!..C
CMPB @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;WRONG PS

42
INC @#COUNT
CMP @#TEMP4,%0 ;IS THE RESULT IN R0 SAME AS TEMP4?
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;WRONG RESULT IN R0

43
CMP @#TEMP5,%1 ;IS THE RESULT IN R1 SAME AS TEMP5?
BEQ .+10 ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
;AND PS=TEMP6
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;WRONG RESULT IN R1

44
MOV R2,R5 ;RESTORE R5
CMP (R5),@#COUNT ;IS TEST NUMBER=COUNTER?
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;NO

45
INC (R5)
CMP (R5),#160 ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
BGE 6$ ;YES
INC @#TEMP3
```

1122	003616	000241			CLC		
1123	003620	006137	000444		ROL	@#TEMP5	:ROTATE TEMPS LEFT BY 1 PLACE
1124	003624	006137	000442		ROL	@#TEMP4	:INTRODUCE CARRY FROM TEMP4 IN TEMP5
1125	003630	021527	000121		CMP	(R5),#121	:IS IT TEST 121?
1126	003634	001004			BNE	REG23	
1127	003636	004467	000410		JSR	R4,RITSH	:IF SO THEN GO AND INITIATE RIGHT SHIFT
1128	003642	004777	000440		JSR	%7,TST160	
1129	003646	010700			6\$: REG23:	SCOPE1	
1130	003650	013702	000434		MOV	@#TEMP1,%2	:PLACE THE CONTENTS OF TEMP1 IN REGISTER 2
1131	003654	013703	00436		MOV	@#TEMP2,%2.1	:PLACE THE CONTENTS OF TEMP2 IN REGISTER 3
1132	003660	000241			CLC		
1133	003662	032737	000001	000406	BIT	#1,@#SPASS	:IS IT AN EVEN PASS ?
1134	003670	001004			BNE	2\$:IF NOT THEN GO TO 2\$
1135	003672	013704	000440		MOV	@#TEMP3,R4	:OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
1136	003676	073204			ASHC	R4,R2	:USING R2
1137	003700	000402			BR	4\$	
1138	003702	073267	174532		2\$: 4\$:	ASHC MFPS	TEMP3,%2 @#PSWORD
1142	003706						:ASHC REGISTER 2 BY THE CONTENTS OF TEMP3 :SAVE PS
(1)	003706	106737				.WORD	106700!..C
146	003712	123737	000446	000432	CMPB	@#TEMP6,@#PSWORD	:COMPARE PS WITH THE CONTENTS OF TEMP6
1147	003720	001403			BEQ	+.10	
1148	003722	004767	013166		JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE :WRONG PS
(2)							
(2)	003726	000046			46		
1149	003730	005237	000430		INC	@#COUNT	
1150	003734	023702	000442		CMP	@#TEMP4,%2	:IS THE RESULT IN R2 SAME AS TEMP4?
1151	003740	001403			BEQ	+.10	
1152	003742	004767	013146		JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE :WRONG RESULT IN R2
(2)							
(2)	003746	000047			47		
1153	003750	023703	000444		CMP	@#TEMP5,%3	:IS THE RESULT IN R3 SAME AS TEMP5?
1154	003754	001403			BEQ	+.10	:TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5 :AND PS=TEMP6
1155							
1156	003756	004767	013132		JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE :WRONG RESULT IN R1
(2)							
(2)	003762	000050			50		
1157	003764	021537	000430		CMP	(R5),@#COUNT	:IS TEST NUMBER=COUNTER?
1158	003770	001403			BEQ	+.10	
1159	003772	004767	013116		JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE :NO
(2)							
(2)	003776	000051			51		
1160	004000	005215			INC	(R5)	
1161	004002	021527	000160		CMP	(R5),#160	:HAVE THE FIRST 159 TEST BEEN EXECUTED?
1162	004006	002014			BGE	6\$:YES
1163	004010	005237	000440		INC	@#TEMP3	
1164	004014	000241			CLC		
1165	004016	006137	000444		ROL	@#TEMP5	:ROTATE TEMPS LEFT BY 1 PLACE
1166	004022	006137	000442		ROL	@#TEMP4	:INTRODUCE CARRY FROM TEMP5 IN TEMP4
1167	004026	021527	000121		CMP	(R5),#121	:IS IT TEST 121?
1168	004032	001004			BNE	REG45	
1169	004034	004467	000212		JSR	R4,RITSH	:IF SO THEN GO AND INITIATE RIGHT SHIFT
1170	004040	004767	000242		6\$: REG45:	JSR SCOPE1	%7,TST160
1171	004044	010701					
1172	004046	010501			MOV	R5,R1	:SAVE R5
1173	004050	013704	000434		MOV	@#TEMP1,%4	:PLACE THE CONTENTS OF TEMP1 IN REGISTER 4
1174	004054	013705	000436		MOV	@#TEMP2,%4.1	:PLACE THE CONTENTS OF TEMP2 IN REGISTER 5

1175	004060	000241			CLC		
1176	004062	032737	000001	000406	BIT	#1,@#SPASS	:IS IT AN EVEN PASS ?
1177	004070	001004			BNE	2\$:IF NOT THEN GO TO 2\$
1178	004072	013700	000440		MOV	@#TEMP3,R0	:OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
1179	004076	073400			ASHC	R0,R4	:USING R4
1180	004100	000402			BR	4\$	
1181	004102	073467	174332		2\$: ASHC	TEMP3,%4	:ASHC REGISTER 4 BY THE CONTENTS OF TEMP3
1185	004106				4\$: MFPS	@#PSWORD	:SAVE PS
(1)	004106	106737			.WORD	106700!..C	
1189	004112	123737	000446	000432	CMPB	@#TEMP6,@#PSWORD	:COMPARE PS WITH THE CONTENTS OF TEMP6
1190	004120	001403			BEQ	.+10	
1191	004122	004767	012766		JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(2)							:WRONG PS
(2)	004126	000052			52		
1192	004130	005237	000430		INC	@#COUNT	
1193	004134	023704	000442		CMP	@#TEMP4,%4	:IS THE RESULT IN R4 SAME AS TEMP4?
1194	004140	001403			BEQ	.+10	
1195	004142	004767	012746		JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(2)							:WRONG RESULT IN R4
(2)	004146	000053			53		
1196	004150	023705	000444		CMP	@#TEMP5,%5	:IS THE RESULT IN R5 SAME AS TEMP5?
1197	004154	001403			BEQ	.+10	:TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
1198							:AND PS=TEMP6
1199	004156	004767	012732		JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(2)							:WRONG RESULT IN R5
(2)	004162	000054			54		
1200	004164	021137	000430		CMP	(R1),@#COUNT	:IS TEST NUMBER=COUNTER?
1201	004170	001403			BEQ	.+10	
1202	004172	004767	012716		JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(2)							:NO
(2)	004176	000055			55		
1203	004200	010105			MOV	R1,R5	:RESTORE R5
1204	004202	005215			INC	(R5)	
1205	004204	021527	000160		CMP	(R5),#160	:HAVE THE FIRST 159 TEST BEEN EXECUTED?
1206	004210	002014			BGE	6\$:YES
1207	004212	005237	000440		INC	@#TEMP3	
1208	004216	000241			CLC		
1209	004220	006137	000444		ROL	@#TEMP5	:ROTATE TEMP5 LEFT BY 1 PLACE
1210	004224	006137	000442		ROL	@#TEMP4	:INTRODUCE CARRY FROM TEMP5 IN TEMP4
1211	004230	021527	000121		CMP	(R5),#121	:IS IT TEST 121?
1212	004234	001004			BNE	8\$	
1213	004236	004467	000010		JSR	R4,RITSH	:IF SO THEN GO AND INITIATE RIGHT SHIFT
1214	004242	004767	000040		6\$: JSR	%7,TST160	
1215	004246	000167	177172		8\$: JMP	REG01	
1216	004252	022424			RITSH: CMP	(R4)+,(R4)+	:MAKE R4 POINT TO THE NEXT REG TAG
1217	004254	012737	040000	000434	MOV	#40000,@#TEMP1	:TEMP1=4000
1218	004262	005037	000436		CLR	@#TEMP2	:TEMP2=0
1219	004266	012737	177742	000440	MOV	#-30,@#TEMP3	:TEMP3=-30
1220	004274	005037	000442		CLR	@#TEMP4	:TEMP4=0
1221	004300	005237	000444		INC	@#TEMP5	:TEMP5=1
1222	004304	000204			RTS	R4	
1223	004306	021527	000160		TST160: CMP	(R5),#160	:IS IT TEST 160
1224	004312	001010			BNE	TST161	:If NOT THEN TRY TEST 161
1225	004314	005037	000434		CLR	@#TEMP1	:0 0 SHIFTED BY 0
1226	004320	005037	000442		CLR	@#TEMP4	:IS EQUAL TO 0 0
1227	004324	012737	000004	000446	MOV	#4,@#TEMP6	:AND PS=4

1228	004332	000207			RTS	%7	
1229	004334	021527	000161		TST161: CMP	(R5),#161	:IS IT TEST 161
1230	004340	001004			BNE	TST162	
1231	004342	012737	177746	000440	MOV	#-32,@#TEMP3	:0 0 SHIFTED BY -32=0 0, PS=4
1232	004350	000207			RTS	%7	
1233	004352	021527	000162		TST162: CMP	(R5),#162	:IS IT TEST 162
1234	004356	001004			BNE	TST163	:IF NOT THEN TRY TEST 163
1235	004360	012737	000032	000440	MOV	#32,@#TEMP3	:0 0 SHIFTED BY 32=0 0, PS=4
1236	004366	000207			RTS	%7	
1237	004370	021527	000163		TST163: CMP	(R5),#163	:IS IT TEST 163?
1238	004374	001016			BNE	TST164	:IF NOT THEN TRY TEST 164
1239	004376	012737	052525	000434	MOV	#52525,@#TEMP1	:52525 0
1240	004404	012737	177760	000440	MOV	#-16,@#TEMP3	:SHIFTED BY -16.
1241	004412	005037	000442		CLR	@#TEMP4	
1242	004416	012737	052525	000444	MOV	#52525,@#TEMP5	:IS EQUAL TO 0 52525
1243	004424	005037	000446		CLR	@#TEMP6	:AND PS = 0
1244	004430	000207			RTS	%7	
1245	004432	021527	000164		TST164: CMP	(R5),#164	:IS IT TEST 164?
1246	004436	001014			BNE	TST165	:IF NOT THEN TRY TEST 165
1247	004440	012737	125252	000434	MOV	#125252,@#TEMP1	:125252 0 SHIFTED BY -16.
1248	004446	005337	000442		DEC	@#TEMP4	
1249	004452	012737	125252	000444	MOV	#125252,@#TEMP5	:IS EQUAL TO -1 125252
1250	004460	012737	000010	000446	MOV	#10,@#TEMP6	:AND PS=10
1251	004466	000207			RTS	%7	
1252	004470	021527	000165		TST165: CMP	(R5),#165	:IS IT TEST 165?
1253	004474	001007			BNE	TST166	:IF NOT THEN TRY TEST 166
1254	004476	012737	177777	000434	MOV	#-1,@#TEMP1	:-1 0 SHIFTED BY -16
1255	004504	012737	177777	000444	MOV	#-1,@#TEMP5	:IS EQUAL TO -1 -1, AND PS=10
1256	004512	000207			RTS	%7	
1257	004514	021527	000166		TST166: CMP	(R5),#166	:IS IT TEST 166?
1258	004520	001011			BNE	TST167	:IF NOT THEN TRY TEST 167
1259	004522	012737	100000	000434	MOV	#100000,@#TEMP1	:100000 0
1260	004530	012737	177740	000440	MOV	#-32,@#TEMP3	:SHIFTED BY -32 IS EQUAL TO -1 -1
1261	004536	005237	000446		INC	@#TEMP6	:AND PS=11
1262	004542	000207			RTS	%7	
1263	004544	021527	000167		TST167: CMP	(R5),#167	:IS IT TEST 167?
1264	004550	001014			BNE	TST170	:IF NOT THEN TRY TEST 170
1265	004552	005037	000434		CLR	@#TEMP1	
1266	004556	005337	000436		DEC	@#TEMP2	:0 -1
1267	004562	012737	000020	000440	MOV	#16,@#TEMP3	:SHIFTED BY 16.
1268	004570	005037	000444		CLR	@#TEMP5	:IS EQUAL TO -1 0
1269	004574	005237	000446		INC	@#TEMP6	:AND PS=12
1270	004600	000207			RTS	%7	
1271	004602	021527	000170		TST170: CMP	(R5),#170	:IS IT TEST 170?
1272	004606	001007			BNE	TST171	:IF NOT THEN TRY TEST 171
1273	004610	012737	125252	000436	MOV	#125252,@#TEMP2	:0 125252 SHIFTED BY 16
1274	004616	012737	125252	000442	MOV	#125252,@#TEMP4	:IS EQUAL TO 125252 0, AND PS=12
1275	004624	000207			RTS	%7	
1276	004626	021527	000171		TST171: CMP	(R5),#171	:IS IT TEST 171?
1277	004632	001010			BNE	TST172	:IF NOT THEN TRY TEST 172
1278	004634	005337	000440		DEC	@#TEMP3	:0 125252 SHIFTED BY 15
1279	004640	012737	052525	000442	MOV	#52525,@#TEMP4	:IS EQUAL TO 52525 0
1280	004646	005037	000446		CLR	@#TEMP6	:AND PS=0
1281	004652	000207			RTS	%7	
1282	004654	021527	000172		TST172: CMP	(R5),#172	:IS IT TEST 172?
1283	004660	001006			BNE	TST173	:IF NOT THEN TRY TEST 173


```

1284 004662 012737 052525 000436      MOV    #52525,@#TEMP2    ;0 52525
1285 004670 005237 000440      INC    @#TEMP3          ;SHIFTED BY 16. IS EQUAL TO 52525 0, AND PS=0
1286 004674 000207                RTS                    %7
1287 004676 021527 000173      TST173: CMP    (R5),#173    ;IS IT TEST 173?
1288 004702 001014                BNE    TST174          ;IF NOT THEN TRY TEST 174
1289 004704 012737 177777 000436      MOV    #-1,@#TEMP2     ;0 -1
1290 004712 005337 000440      DEC    @#TEMP3          ;SHIFTED BY 15.
1291 004716 012737 077777 000442      MOV    #77777,@#TEMP4
1292 004724 012737 100000 000444      MOV    #100000,@#TEMP5 ;IS EQUAL TO 77777 100000, AND PS=0
1293 004732 000207                RTS                    %7
1294 004734 021527 000174      TST174: CMP    (R5),#174    ;IS IT TEST 174?
1295 004740 001013                BNE    TST175          ;IF NOT THEN TRY TEST 175
1296 004742 012737 100000 000434      MOV    #100000,@#TEMP1
1297 004750 005337 000436      DEC    @#TEMP2          ;100000 -2 SHIFTED BY 15.
1298 004754 005037 000444      CLR    @#TEMP5          ;IS EQUAL TO 77777 0
1299 004760 012737 000002 000446      MOV    #2,@#TEMP6      ;AND PS=2
1300 004766 000207                RTS                    %7
1301 004770 021527 000175      TST175: CMP    (R5),#175    ;IS IT TEST 175?
1302 004774 001015                BNE    ENT176          ;IF NOT THEN TRY TEST 176
1303 004776 012737 177777 000434      MOV    #-1,@#TEMP1
1304 005004 005037 000436      CLR    @#TEMP2          ;-1 0
1305 005010 005237 000440      INC    @#TEMP3          ;SHIFTED BY 16.
1306 005014 005037 000442      CLR    @#TEMP4          ;IS EQUAL TO 0 0
1307 005020 012737 000007 000446      MOV    #7,@#TEMP6      ;AND PS=7
1308 005026 000207                RTS                    %7
1309 005030 021527 000176      ENT176: CMP    (R5),#176    ;IS THE PROGRM ENTERING TEST 176?
1310 005034 001403                BEQ    .+10
1311 005036 004767 012052      JSR    PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(2)                                ;TEST NUMBER GOOFED
(2) 005042 000056                56
1312                                ;
1313 005044 005726                TST    (SP)+            ;RESTORE STACK POINTER
1314
    
```


(1)
(1)

1317

```

(1)
(1)
(1)
(1) 005236 010701
(1) 005240 010501
(1) 005242 012705 000000
(1) 005246 012705 052525
(1) 005252 000241
(1) 005254 073527 000000
(1) 005260
(2) 005260 106737
(1) 005264 122737 000000 000432
(1) 005272 001403
(3) 005274 004767 011614
(3)
(3) 005300 000065
(1) 005302 022705 052525
(1) 005306 001403
(3) 005310 004767 011600
(3)
(3) 005314 000066
(1) 005316 010105
(1) 005320 021527 000200
(1) 005324 001403
(3) 005326 004767 011562
(3)
(3) 005332 000067
(1) 005334 005215

```

```

:*****
:TEST:200      52525 SHIFTED BY 0 = 52525 PS 0
:*****
TST200: SCOPE1
MOV      R5,R1      ;SAVE R5
MOV      #DUMMY,%5  ;LOAD R5 WITH DUMMY
MOV      #52525,%5!1 ;LOAD R5!1 WITH 52525
CLC
ASHC     #0,%5      ;SHIFT R5,R5!1 BY 0
MFPS     @#PSWORD   ;SAVE PS
        .WORD      106700!..C
CMPB     #0,@#PSWORD ;IS THE PS 0?
BEQ      .+10
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
        ;THE PS IS NOT EQUAL TO 0
        65
CMP      #52525,%5  ;IS THE RESULT 52525?
BEQ      .+10
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
        ;R5 IS NOT EQUAL TO 52525
        66
MOV      R1,R5      ;RESTORE R5
CMP      (R5),#200  ;IS $TESTN = #200?
BEQ      .+10      ;IF NOT THEN GO TO HLT
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
        ;TEST IS IN WRONG SEQUENCE
        67
INC      (R5)

```

1318

```

(1)
(1)
(1)
(1) 005336 010701
( ) 005340 012701 000000
(1) 005344 012701 020010
(1) 005350 000241
(1) 005352 073127 177763
(1) 005356
(2) 005356 106737
(1) 005362 122737 000000 000432
(1) 005370 001403
(3) 005372 004767 011516
(3)
(3) 005376 000070
(1) 005400 022701 000101
(1) 005404 001403
(3) 005406 004767 011502
(3)
(3) 005412 000071
(1) 005414 021527 000201
(1) 005420 001403
(3) 005422 004767 011466

```

```

:*****
:TEST:201      20010 SHIFTED BY -13. - 101 PS - 0
:*****
TST201: SCOPE1
MOV      #DUMMY,%1  ;LOAD R1 WITH DUMMY
MOV      #20010,%1.1 ;LOAD R1!1 WITH 20010
CLC
ASHC     #-13.,%1   ;SHIFT R1,R1.1 BY -13.
MFPS     @#PSWORD   ;SAVE PS
        .WORD      106700!..C
CMPB     #0,@#PSWORD ;IS THE PS 0?
BEQ      .+10
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
        ;THE PS IS NOT EQUAL TO 0
        70
CMP      #101,%1    ;IS THE RESULT 101?
BEQ      .+10
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
        ;R1 IS NOT EQUAL TO 101
        71
CMP      (R5),#201  ;IS $TESTN = #201?
BEQ      .+10      ;IF NOT THEN GO TO HLT
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
        ;TEST IS IN WRONG SEQUENCE

```

DVKABA MACY:1 30(1046) 14-SEP-81 16:32 PAGE 4-25
CVKABB.P11 14-SEP-81 16:31 ASHC INSTRUCTION TESTS

EQ 0030

(3)	005426	000072	72	
(1)	005430	005215	INC	(R5)
(1)				
(1)				

1319
 (1)
 (1)
 (1)
 (1) 005432 010701
 (1) 005434 012703 000000
 (1) 005440 012703 177777
 (1) 005444 000241
 (1) 005446 073327 000020
 (1) 005452
 (2) 005452 106737
 (1) 005456 122737 000011 000432
 (1) 005464 001403
 (3) 005466 004767 011422
 (3)
 (3) 005472 000073
 (1) 005474 022703 000000
 (1) 005500 001403
 (3) 005502 004767 011406
 (3)
 (3) 005506 000074
 (1) 005510 021527 000202
 (1) 005514 001403
 (3) 005516 004767 011372
 (3)
 (3) 005522 000075
 (1) 005524 005215

```

:*****
:TEST:202      -1 SHIFTED BY 16. = 0  PS = 11
:*****
TST202: SCOPE1
MOV      #DUMMY,%3      ;LOAD R3 WITH DUMMY
MOV      #-1,%3.1      ;LOAD R3!1 WITH -1
CLC
ASHC     #16,%3         ;SHIFT R3,R3.1 BY 16.
MFPS     @#PSWORD      ;SAVE PS
.WORD    106700!..C
CMPB     #11,@#PSWORD   ;IS THE PS 11?
BEQ      .+10
JSR      PC,$HLT       ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 11
73
CMP      #0,%3         ;IS THE RESULT 0?
BEQ      .+10
JSR      PC,$HLT       ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R3 IS NOT EQUAL TO 0
74
CMP      (R5),#202     ;IS $TESTN - #202?
BEQ      .+10         ;IF NOT THEN GO TO HLT
JSR      PC,$HLT       ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;TEST IS IN WRONG SEQUENCE
75
INC      (R5)

```

1320
 (1)
 (1)
 (1)
 (1) 005526 010701
 (1) 005530 010501
 (1) 005532 012705 000000
 (1) 005536 012705 000001
 (1) 005542 000241
 (1) 005544 073527 177777
 (1) 005550
 (2) 005550 106737
 (1) 005554 122737 000001 000432
 (1) 005562 001403
 (3) 005564 004767 011324
 (3)
 (3) 005570 000076
 (1) 005572 022705 100000
 (1) 005576 001403
 (3) 005600 004767 011310
 (3)
 (3) 005604 000077
 (1) 005606 010105
 (1) 005610 021527 000203
 (1) 005614 001403
 (3) 005616 004767 011272
 (3)

```

:*****
:TEST:203      1 SHIFTED BY -1 = 100000  PS - 1
:*****
TST203: SCOPE1
MOV      R5,R1         ;SAVE R5
MOV      #DUMMY,%5     ;LOAD R5 WITH DUMMY
MOV      #1,%5!1      ;LOAD R5!1 WITH 1
CLC
ASHC     #-1,%5        ;SHIFT R5,R5!1 BY -1
MFPS     @#PSWORD      ;SAVE PS
.WORD    106700!..C
CMPB     #1,@#PSWORD   ;IS THE PS 1?
BEQ      .+10
JSR      PC,$HLT       ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 1
76
CMP      #100000,%5    ;IS THE RESULT 100000?
BEQ      .+10
JSR      PC,$HLT       ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R5 IS NOT EQUAL TO 100000
77
MOV      R1,R5         ;RESTORE R5
CMP      (R5),#203     ;IS $TESTN = #203?
BEQ      .+10         ;IF NOT THEN GO TO HLT
JSR      PC,$HLT       ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;TEST IS IN WRONG SEQUENCE

```


(3)	005622	000100	100	
(1)	005624	005215	INC	(R5)
(1)				
(1)				

(1)	006020	001403		BEO	.+10	
(3)	006022	004767	011066	JSR	PC,\$HLT	
(3)						
(3)	006026	000107		107		
(1)	006030	005215		INC	(R5)	
(1)						
(1)						
1323						
1324	006032	012702	177771	MOV	#-7,%2	
1325	006036	012703	000454	MOV	#51,%3	
1326	006042	012704	000456	MOV	#52,%4	
1327						

:IF NOT THEN GO TO HLT
 :SEEN AN ERROR, GO TO TH HALT ROUTINE
 :TEST IS IN WRONG SEQUENCE

:

(3)	006234	004767	010654	JSR	PC,\$HLT
(3)					
(3)	006240	000115		115	
(1)	006242	021527	000207	CMP	(R5),#207
(1)	006246	001372		BNE	1\$
(1)	006250	005215		INC	(R5)
(1)					
(1)					

:SEEN AN ERROR, GO TO TH HALT ROUTINE
:R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
:IS THE \$TESTN = #207?
:IF NOT THEN GO TO HLT ABOVE

1330

```
(1)
(1)
(1)
(1) 006252 010701
(1) 006254 012700 125252
(1) 006260 012701 125252
(1) 006264 000241
(1) 006266 073037 000454
(1) 006272
(2) 006272 106737
(1) 006276 122737 000010 000432
(1) 006304 001403
(3) 006306 004767 010602
(3)
(3) 006312 000116
(1) 006314 022700 177525
(1) 006320 001403
(3) 006322 004767 010566
(3)
(3) 006326 000117
(1) 006330 022701 052525
(1) 006334 001403
(1) 006336
(3) 006336 004767 010552
(3)
(3) 006342 000120
(1) 006344 021527 000210
(1) 006350 001372
(1) 006352 005215
( )
(1)
```

```
*****
:TEST:210 125252 125252 SHIFTED BY @#S1 = 177525 52525 PS = 10
*****
TST210: SCOPE1
MOV #125252,%0 ;LOAD R0 WITH 125252
MOV #125252,%0!1 ;LOAD R0!1 WITH 125252
CLC
ASHC @#S1,%0 ;SHIFT R0,R0!1 BY @#S1
MFPS @#PSWORD ;SAVE PS
.WORD 106700!..C
CMPB #10,@#PSWORD ;IS THE PS 10?
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 10
116
CMP #177525,%0 ;IS THE RESULT 177525?
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0 IS NOT EQUAL TO 177525
117
CMP #52525,%0!1 ;IS THE RESULT 52525?
BEQ .+10
1$: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
120
CMP (R5),#210 ;IS THE $TESTN = #210?
BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
INC (R5)
```

1331

```
(1)
(1)
(1)
(1) 006354 010701
(1) 006356 012700 125252
(1) 006362 012701 125252
(1) 006366 000241
(1) 006370 073013
(1) 006372
(2) 006372 106737
(1) 006376 122737 000010 000432
(1) 006404 001403
(3) 00 06 004767 010502
(3)
(3) 006412 000121
(1) 006414 022700 177525
(1) 006420 001403
(3) 006422 004767 010466
(3)
(3) 006426 000122
(1) 006430 022701 052525
(1) 006434 001403
(1) 006436
```

```
*****
:TEST:211 125252 125252 SHIFTED BY (3) = 177525 52525 PS = 10
*****
TST211: SCOPE1
MOV #125252,%0 ;LOAD R0 WITH 125252
MOV #125252,%0!1 ;LOAD R0!1 WITH 125252
CLC
ASHC (3),%0 ;SHIFT R0,R0!1 BY (3)
MFPS @#PSWORD ;SAVE PS
.WORD 106700!..C
CMPB #10,@#PSWORD ;IS THE PS 10?
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 10
121
CMP #177525,%0 ;IS THE RESULT 177525?
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0 IS NOT EQUAL TO 177525
122
CMP #52525,%0!1 ;IS THE RESULT 52525?
BEQ .+10
1$: JSR PC,$HLT
```

(3)	006436	004767	010452	JSR	PC,\$HLT
(3)					
(3)	006442	000123		123	
(1)	006444	021527	000211	CMP	(R5),#211
(1)	006450	001372		BNE	1\$
(1)	006452	005215		INC	(R5)
(1)					
(1)					

```

:SEEN AN ERROR, GO TO TH HALT ROUTINE
:R0:1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE

:IS THE $TESTN = #211?
:IF NOT THEN GO TO HLT ABOVE

```

```

1332
(1)
(1)
(1)
(1) 006454 010701
(1) 006456 012700 125252
(1) 006462 012701 125252
(1) 006466 000241
(1) 006470 073023
(1) 006472
(2) 006472 106737
(1) 006476 122737 000010 000432
(1) 006504 001403
(3) 006506 004767 010402
(3)
(3) 006512 000124
(1) 006514 022700 177525
(1) 006520 001403
(3) 006522 004767 010366
(3)
(3) 006526 000125
(1) 006530 022701 052525
(1) 006534 001403
(1) 006536
(3) 006536 004767 010352
(3)
(3) 006542 000126
(1) 006544 021527 000212
(1) 006550 001372
(1) 006552 005215
(1)
(1)

```

```

:*****
:TEST:212      125252 125252 SHIFTED BY (3)+ = 177525 52525 PS = 10
:*****
TST212: SCOPE1
MOV      #125252,%0          ;LOAD R0 WITH 125252
MOV      #125252,%0!1      ;LOAD R0!1 WITH 125252
CLC
ASHC     (3)+,%0           ;SHIFT R0,R0!1 BY (3)+
MFPS     @#PSWORD         ;SAVE PS
.WORD    106700!..C
CMPB     #10,@#PSWORD      ;IS THE PS 10?
BEQ      .+10
JSR      PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 10
124
CMP      #177525,%0        ;IS THE RESULT 177525?
BEQ      .+10
JSR      PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0 IS NOT EQUAL TO 177525
125
CMP      #52525,%0.1      ;IS THE RESULT 52525?
BEQ      .+10
1$:
JSR      PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
126
CMP      (R5),#212        ;IS THE $TESTN = #212?
BNE      1$              ;IF NOT THEN GO TO HLT ABOVE
INC      (R5)

```

```

1333
(1)
(1)
(1)
(1) 006554 010701
(1) 006556 012700 125252
(1) 006562 012701 125252
(1) 006566 000241
(1) 006570 073043
(1) 006572
(2) 006572 106737
(1) 006576 122737 000010 000432
(1) 006604 001403
(3) 006606 004767 010302
(3)
(3) 006612 000127
(1) 006614 022700 177525
(1) 006620 001403
(3) 006622 004767 010266
(3)
(3) 006626 000130
(1) 006630 022701 052525
(1) 006634 001403
(1) 006636

```

```

:*****
:TEST:213      125252 125252 SHIFTED BY -(3) = 177525 52525 PS = 10
:*****
TST213: SCOPE1
MOV      #125252,%0          ;LOAD R0 WITH 125252
MOV      #125252,%0!1      ;LOAD R0!1 WITH 125252
CLC
ASHC     -(3),%0          ;SHIFT R0,R0!1 BY -(3)
MFPS     @#PSWORD         ;SAVE PS
.WORD    106700!..C
CMPB     #10,@#PSWORD      ;IS THE PS 10?
BEQ      .+10
JSR      PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 10
127
CMP      #177525,%0        ;IS THE RESULT 177525?
BEQ      .+10
JSR      PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0 IS NOT EQUAL TO 177525
130
CMP      #52525,%0!1      ;IS THE RESULT 52525?
BEQ      .+10
1$:

```


(3) 006636 004767 010252
 (3)
 (3) 006642 000131
 (1) 006644 021527 000213
 (1) 006650 001372
 (1) 006652 005215
 (1)
 (1)

JSR PC,\$HLT
 131
 CMP (R5),#213
 BNE 1\$
 INC (R5)

:SEEN AN ERROR, GO TO TH HALT ROUTINE
 :R0.1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
 :IS THE \$TESTN = #213?
 :IF NOT THEN GO TO HLT ABOVE

1334

```
(1)
(1)
(1)
(1) 006654 010701
(1) 006656 012700 125252
(1) 006662 012701 125252
(1) 006666 000241
(1) 006670 073064 000002
(1) 006674
(2) 006674 106737
(1) 006700 122737 000011 000432
(1) 006706 001403
(3) 006710 004767 010200
(3)
(3) 006714 000132
(1) 006716 022700 177252
(1) 006722 001403
(3) 006724 004767 010164
(3)
(3) 006730 000133
(1) 006732 022701 125252
(1) 006736 001403
(1) 006740
(3) 006740 004767 010150
(3)
(3) 006744 000134
(1) 006746 021527 000214
(1) 006752 001372
(1) 006754 005215
(1)
(1)
```

```
*****
:TEST:214 125252 125252 SHIFTED BY 2(4) = 177252 125252 PS = 11
*****
TST214: SCOPE1
MOV #125252,%0 ;LOAD R0 WITH 125252
MOV #125252,%0!1 ;LOAD R0!1 WITH 125252
CLC
ASHC 2(4),%0 ;SHIFT R0,R0!1 BY 2(4)
MFPS @#PSWORD ;SAVE PS
.WORD 106700!..C
CMPB #11,@#PSWORD ;IS THE PS 11?
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 11
132
CMP #177252,%0 ;IS THE RESULT 177252?
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0 IS NOT EQUAL TO 177252
133
CMP #125252,%0!1 ;IS THE RESULT 125252?
BEQ .+10
1$: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0!1 IS NOT EQUAL TO 125252 OR INCORRECT SEQUENCE
134
CMP (R5),#214 ;IS THE $TESTN = #214?
BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
INC (R5)
```

1335

```
(1)
(1)
(1)
(1) 006756 010701
(1) 006760 012700 125252
(1) 006764 012701 125252
(1) 006770 000241
(1) 006772 073074 000000
(1) 006776
(2) 006776 106737
(1) 007002 122737 000010 000432
(1) 007010 001403
(3) 007012 004767 010076
(3)
(3) 007016 000135
(1) 007020 022700 177525
(1) 007024 001403
(3) 007026 004767 010062
(3)
(3) 007032 000136
(1) 007034 022701 052525
(1) 007040 001403
(1) 007042
```

```
*****
:TEST:215 125252 125252 SHIFTED BY @ (4) = 177525 52525 PS = 10
*****
TST215: SCOPE1
MOV #125252,%0 ;LOAD R0 WITH 125252
MOV #125252,%0!1 ;LOAD R0!1 WITH 125252
CLC
ASHC @(4),%0 ;SHIFT R0,R0!1 BY @(4)
MFPS @#PSWORD ;SAVE PS
.WORD 106700!..C
CMPB #10,@#PSWORD ;IS THE PS 10?
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 10
135
CMP #177525,%0 ;IS THE RESULT 177525?
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0 IS NOT EQUAL TO 177525
136
CMP #52525,%0.1 ;IS THE RESULT 52525?
BEQ .+10
1$: JSR PC,$HLT
```

(3)	007042	004767	010046	JSR	PC,\$HLT
(3)					
(3)	007046	000137		137	
(1)	007050	021527	000215	CMP	(R5),#215
(1)	007054	001372		BNE	1\$
(1)	007056	005215		INC	(R5)
(1)					
(1)					

```

;SEEN AN ERROR, GO TO TH HALT ROUTINE
;R0:1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE

;IS THE $TESTN = #215?
;IF NOT THEN GO TO HLT ABOVE

```

1336

(1)
 (1)
 (1)
 (1) 007060 010701
 (1) 007062 012700 125252
 (1) 007066 012701 125252
 (1) 007072 000241
 (1) 007074 073034
 (1) 007076
 (2) 00707E 106737
 (1) 007102 122737 000010 000432
 (1) 007110 001403
 (3) 007112 004767 007776
 (3)
 (3) 007116 000140
 (1) 007120 022700 177525
 (1) 007124 001403
 (3) 007126 004767 007762
 (3)
 (3) 007132 000141
 (1) 007134 022701 052525
 (1) 007140 001403
 (1) 007142
 (3) 007142 004767 007746
 (3)
 (3) 007146 000142
 (1) 007150 021527 000216
 (1) 007154 001372
 (1) 007156 005215

```

:*****
:TEST:216      125252 125252 SHIFTED BY @-(4)+ = 177525 52525 PS = 10
:*****
TST216: SCOPE1
MOV      #125252,%0          ;LOAD R0 WITH 125252
MOV      #125252,%0!1       ;LOAD R0!1 WITH 125252
CLC
ASHC    @-(4),%0           ;SHIFT R0,R0!1 BY @-(4)
MFPS    @#PSWORD          ;SAVE PS
        .WORD 106700!..C
CMPB    #10,@#PSWORD       ;IS THE PS 10?
BEQ     .+10
JSR     PC,$HLT            ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;THE PS IS NOT EQUAL TO 10
        140
CMP     #177525,%0         ;IS THE RESULT 177525?
BEQ     .+10
JSR     PC,$HLT            ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;R0 IS NOT EQUAL TO 177525
        141
CMP     #52525,%0.1       ;IS THE RESULT 52525?
BEQ     .+10
1$:     JSR     PC,$HLT            ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
        142
CMP     (R5),#216          ;IS THE $TESTN = #216?
BNE     1$                ;IF NOT THEN GO TO HLT ABOVE
INC     (R5)
    
```

1337

(1)
 (1)
 (1)
 (1) 007160 010701
 (1) 007162 012700 125252
 (1) 007166 012701 125252
 (1) 007172 000241
 (1) 007174 073054
 (1) 007176
 (2) 007176 106737
 (1) 007202 122737 000010 000432
 (1) 007210 001403
 (3) 007212 004767 007676
 (3)
 (3) 007216 000143
 (1) 007220 022700 177525
 (1) 007224 001403
 (3) 007226 004767 007662
 (3)
 (3) 007232 000144
 (1) 007234 022701 052525
 (1) 007240 001403
 (1) 007242

```

:*****
:TEST:217      125252 125252 SHIFTED BY @-(4) = 177525 52525 PS = 10
:*****
TST217: SCOPE1
MOV      #125252,%0          ;LOAD R0 WITH 125252
MOV      #125252,%0!1       ;LOAD R0!1 WITH 125252
CLC
ASHC    @-(4),%0           ;SHIFT R0,R0!1 BY @-(4)
MFPS    @#PSWORD          ;SAVE PS
        .WORD 106700!..C
CMPB    #10,@#PSWORD       ;IS THE PS 10?
BEQ     .+10
JSR     PC,$HLT            ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;THE PS IS NOT EQUAL TO 10
        143
CMP     #177525,%0         ;IS THE RESULT 177525?
BEQ     .+10
JSR     PC,$HLT            ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;R0 IS NOT EQUAL TO 177525
        144
CMP     #52525,%0!1       ;IS THE RESULT 52525?
BEQ     .+10
1$:
    
```

```
(3) 007242 004767 007646 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
(3) ;RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE  
(3) 007246 000145 145  
(1) 007250 021527 000217 CMP (R5),#217 ;IS THE $TESTN = #217?  
(1) 007254 001372 BNE 1$ ;IF NOT THEN GO TO HLT ABOVE  
(1) 007256 005215 INC (R5)
```

1338
1339
1340
1341
1342
1343
1344
1438
1439
1440
1441
1442
1443
1444
1445

: MUL INSTRUCTION TESTS

:TEST:220 MUL 1 * #0 = 0 0 PS = 4

```
(1) 007260 010701 TST220: SCOPE  
(1) 007262 012700 000001 MOV #1,%0 ;LOAD MULTIPLICAND WITH 1  
(1) 007266 070027 000000 MUL #0,%0 ;MULTIPLY 1 * #0  
(1) 007272 MFPS @#PSWORD ;SAVE PS  
(2) 007272 106737 .WORD 106700!..C  
(1) 007276 122737 000004 000432 CMPB #4,@#PSWORD ;IS PS = 4  
(1) 007304 001403 BEQ .+10  
(3) 007306 004767 007602 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
(3) ;PS IS WRONG  
(3) 007312 000146 146  
(1) 007314 022700 000000 CMP #0,%0 ;IS HIGH ORDER - 0  
(1) 007320 001403 BEQ .+10  
(3) 007322 004767 007566 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
(3) ;HIGH ORDER IS WRONG  
(3) 007326 000147 147  
(1) 007330 022701 000000 CMP #0,%0!1 ;IS LOW ORDER - 0  
(1) 007334 001403 BEQ .+10  
(1) 007336 1$:  
(3) 007336 004767 007552 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
(3) ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
(3) 007342 000150 150  
(1) 007344 021527 000220 CMP (R5),#220  
(1) 007350 001372 BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
(1) 007352 005215 INC (R5)
```

:TEST:221 MUL -1 * #1 = -1 -1 PS = 10

```
(1) 007354 010701 TST221: SCOPE  
(1) 007356 012700 177777 MOV #-1,%0 ;LOAD MULTIPLICAND WITH -1
```

```

(1) 007362 070027 000001      MUL      #1,%0      ;MULTIPLY -1 * #1
(1) 007366                    MFPS     @#PSWORD   ;SAVE PS
(2) 007366 106737                    .WORD   106700!..C
(1) 007372 122737 000010 000432  CMPB    #10,@#PSWORD ;IS PS = 10
(1) 007400 001403                    BEQ     .+10
(3) 007402 004767 007506      JSR     PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;PS IS WRONG
(3) 007406 000151                    151
(1) 007410 022700 177777      CMP     #-1,%0     ;IS HIGH ORDER = -1
(1) 007414 001403                    BEQ     .+10
(3) 007416 004767 007472      JSR     PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;HIGH ORDER IS WRONG
(3) 007422 000152                    152
(1) 007424 022701 177777      CMP     #-1,%0!1   ;IS LOW ORDER = -1
(1) 007430 001403                    BEQ     .+10
(1) 007432                    1$:
(3) 007432 004767 007456      JSR     PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 007436 000153                    153
(1) 007440 021527 000221      CMP     (R5),#221
(1) 007444 001372                    BNE    1$           ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 007446 005215                    INC    (R5)
1447
(1)                                ;*****
(1)                                ;TEST:222      MUL      2 * #2 = 0 4      PS - 0
(1)                                ;*****
(1) 007450 010701                    *ST222: SCOPE
(1) 007452 012702 000002      MOV     #2,%2      ;LOAD MULTIPLICAND WITH 2
(1) 007456 070227 000002      MUL     #2,%2      ;MULTIPLY 2 * #2
(1) 007462                    MFPS     @#PSWORD   ;SAVE PS
(2) 007462 106737                    .WORD   106700!..C
(1) 007466 122737 000000 000432  CMPB    #0,@#PSWORD ;IS PS = 0
(1) 007474 001403                    BEQ     .+10
(3) 007476 004767 007412      JSR     PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;PS IS WRONG
(3) 007502 000154                    154
(1) 007504 022702 000000      CMP     #0,%2      ;IS HIGH ORDER = 0
(1) 007510 001403                    BEQ     .+10
(3) 007512 004767 007376      JSR     PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;HIGH ORDER IS WRONG
(3) 007516 000155                    155
(1) 007520 022703 000004      CMP     #4,%2!1   ;IS LOW ORDER - 4
(1) 007524 001403                    BEQ     .+10
(1) 007526                    1$:
(3) 007526 004767 007362      JSR     PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 007532 000156                    156
(1) 007534 021527 000222      CMP     (R5),#222
(1) 007540 001372                    BNE    1$           ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 007542 005215                    INC    (R5)
1448
(1)                                ;*****
(1)                                ;TEST:223      MUL      1000 * #200 1 0      PS 1
(1)                                ;*****
(1) 007544 010701                    TS1223: SCOPE
(1) 007546 010501      MOV     R5,R1      ;SAVE R5

```

```
(1) 007550 012704 001000      MOV      #1000,%4      ;LOAD MULTIPLICAND WITH 1000
(1) 007554 070427 000200      MUL      #200,%4      ;MULTIPLY 1000 * #200
(1) 007560      MFPS      @#PSWORD      ;SAVE PS
(2) 007560 106737      .WORD    106700!..C
(1) 007564 122737 000001 000432  CMPB     #1,@#PSWORD      ;IS PS = 1
(1) 007572 001403      BEQ      .+10
(3) 007574 004767 C07314      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)      ;PS IS WRONG
(3) 007600 000157      157
(1) 007602 022704 000001      CMP      #1,%4      ;IS HIGH ORDER = 1
(1) 007606 001403      BEQ      .+10
(3) 007610 004767 007300      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)      ;HIGH ORDER IS WRONG
(3) 007614 000160      160
(1) 007616 022705 000000      CMP      #0,%4 1      ;IS LOW ORDER = 0
(1) 007622 001403      BEQ      .+10
(1) 007624      1$:
(3) 007624 004767 007264      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)      ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 007630 000161      161
(1) 007632 021127 000223      CMP      (R1),#223      ;CHECK THE TEST NUMBER
(1) 007636 001372      BNE      1$            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 007640 010105      MOV      R1,R5          ;RESTORE R5
(1) 007642 005215      INC      (R5)
```

```
1449
(1) *****
(1) ;TEST:224      MUL      2 * #77777 - 0 177776      PS 1
(1) *****
```

```
(1) 007644 010701      TST224: SCOPE
(1) 007646 012700 000002      MOV      #2,%0      ;LOAD MULTIPLICAND WITH 2
(1) 007652 070027 077777      MUL      #77777,%0      ;MULTIPLY 2 * #77777
(1) 007656      MFPS      @#PSWORD      ;SAVE PS
(2) 007656 106737      .WORD    106700!..C
(1) 007662 122737 000001 000432  CMPB     #1,@#PSWORD      ;IS PS = 1
(1) 007670 001403      BEQ      .+10
(3) 007672 004767 007216      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)      ;PS IS WRONG
(3) 007676 000162      162
(1) 007700 022700 000000      CMP      #0,%0      ;IS HIGH ORDER = 0
(1) 007704 001403      BEQ      .+10
(3) 007706 004767 007202      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)      ;HIGH ORDER IS WRONG
(3) 007712 000163      163
(1) 007714 022701 177776      CMP      #177776,%0!1      ;IS LOW ORDER = 177776
(1) 007720 001403      BEQ      .+10
(1) 007722      1$:
(3) 007722 004767 007166      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)      ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 007726 000164      164
(1) 007730 021527 000224      CMP      (R5),#224
(1) 007734 001372      BNE      1$            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 007736 005215      INC      (R5)
```

```
1450
(1) *****
(1) ;TEST:225      MUL      7777 * #10 = 0 77770      PS = 0
(1) *****
(1)
```

```

(1) 007740 010701          TST225: SCOPE
(1) 007742 012702 007777  MOV      #7777,%2      ;LOAD MULTIPLICAND WITH 7777
(1) 007746 070227 000010  MUL      #10,%2      ;MULTIPLY 7777 * #10
(1) 007752          MFPS     @#PSWORD    ;SAVE PS
(2) 007752 106737          .WORD    106700!..C
(1) 007756 122737 000000 000432  CMPB    #0,@#PSWORD  ;IS PS = 0
(1) 007764 001403          BEQ      .+10
(3) 007766 004767 007122  JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;PS IS WRONG
(3) 007772 000165          165
(1) 007774 022702 000000  CMP      #0,%2      ;IS HIGH ORDER - 0
(1) 010000 001403          BEQ      .+10
(3) 010002 004767 007106  JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;HIGH ORDER IS WRONG
(3) 010006 000166          166
(1) 010010 022703 077770  CMP      #77770,%2.1 ;IS LOW ORDER = 77770
(1) 010014 001403          BEQ      .+10
(1) 010016          1$:
(3) 010016 004767 007072  JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 010022 000167          167
(1) 010024 021527 000225  CMP      (R5),#225
(1) 010030 001372          BNE     1$            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 010032 005215          INC     (R5)
1451 *****
(1) ;TEST:226      MUL      77777 * #77777 = 37777 1      PS = 1
(1) *****
(1)
(1) 010034 010701          TST226: SCOPE
(1) 010036 010501          MOV     R5,R1      ;SAVE R5
(1) 010040 012704 077777  MOV     #77777,%4   ;LOAD MULTIPLICAND WITH 77777
(1) 010044 070427 077777  MUL     #77777,%4   ;MULTIPLY 77777 * #77777
(1) 010050          MFPS     @#PSWORD    ;SAVE PS
(2) 010050 106737          .WORD    106700!..C
(1) 010054 122737 000001 000432  CMPB    #1,@#PSWORD  ;IS PS = 1
(1) 010062 001403          BEQ      .+10
(3) 010064 004767 007024  JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;PS IS WRONG
(3) 010070 000170          170
(1) 010072 022704 037777  CMP     #37777,%4   ;IS HIGH ORDER = 37777
(1) 010076 001403          BEQ     .+10
(3) 010100 004767 007010  JSR     PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;HIGH ORDER IS WRONG
(3) 010104 000171          171
(1) 010106 022705 000001  CMP     #1,%4!1     ;IS LOW ORDER = 1
(1) 010112 001403          BEQ     .+10
(1) 010114          1$:
(3) 010114 004767 006774  JSR     PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 010120 000172          172
(1) 010122 021127 000226  CMP     (R1),#226   ;CHECK THE TEST NUMBER
(1) 010126 001372          BNE     1$            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 010130 010105          MOV     R1,R5      ;RESTORE R5
(1) 010132 005215          INC     (R5)
1452 *****
(1) ;TEST:227      MUL     -1 * #77777  -1 100001      PS = 10

```



```

(1) ;*****
(1)
(1) 010134 010701 TST227: SCOPE
(1) 010136 012702 177777 MOV # -1,%2 ;LOAD MULTIPLICAND WITH -1
(1) 010142 070227 077777 MUL #77777,%2 ;MULTIPLY -1 * #77777
(1) 010146 MFPS @#PSWORD ;SAVE PS
(2) 010146 106737 .WORD 106700!..C
(1) 010152 122737 000010 000432 CMPB #10,@#PSWORD ;IS PS = 10
(1) 010160 001403 BEQ .+10
(3) 010162 004767 006726 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;PS IS WRONG
(3) 010166 000173 173
(1) 010170 022702 177777 CMP # -1,%2 ;IS HIGH ORDER = -1
(1) 010174 001403 BEQ .+10
(3) 010176 004767 006712 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;HIGH ORDER IS WRONG
(3) 010202 000174 174
(1) 010204 022703 100001 CMP #100001,%2!1 ;IS LOW ORDER = 100001
(1) 010210 001403 BEQ .+10
(1) 010212 1$:
(3) 010212 004767 006676 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 010216 000175 175
(1) 010220 021527 000227 CMP (R5),#227
(1) 010224 001372 BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 010226 005215 INC (R5)

```

```

1453 ;*****
(1) ;TEST:230 MUL -2 * #77777 = -1 2 PS = 11
(1) ;*****
(1)

```

```

(1) 010230 010701 TST230: SCOPE
(1) 010232 012700 177776 MOV # -2,%0 ;LOAD MULTIPLICAND WITH -2
(1) 010236 070027 077777 MUL #77777,%0 ;MULTIPLY -2 * #77777
(1) 010242 MFPS @#PSWORD ;SAVE PS
(2) 010242 106737 .WORD 106700!..C
(1) 010246 122737 000011 000432 CMPB #11,@#PSWORD ;IS PS = 11
(1) 010254 001403 BEQ .+10
(3) 010256 004767 006632 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;PS IS WRONG
(3) 010262 000176 176
(1) 010264 022700 177777 CMP # -1,%0 ;IS HIGH ORDER - -1
(1) 010270 001403 BEQ .+10
(3) 010272 004767 006616 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;HIGH ORDER IS WRONG
(3) 010276 000177 177
(1) 010300 022701 000002 CMP #2,%0!1 ;IS LOW ORDER = 2
(1) 010304 001403 BEQ .+10
(1) 010306 1$:
(3) 010306 004767 006602 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 010312 000200 200
(1) 010314 021527 000230 CMP (R5),#230
(1) 010320 001372 BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 010322 005215 INC (R5)

```

```

1454 ;*****
(1) ;TEST:231 MUL 125252 * #2 = -1 52524 PS = 11

```

```

(1) ;*****
(1)
(1) 010324 010701 TST231: SCOPE
(1) 010326 012702 125252 MOV #125252,%2 ;LOAD MULTIPLICAND WITH 125252
(1) 010332 070227 000002 MUL #2,%2 ;MULTIPLY 125252 * #2
(1) 010336 MFPS @#PSWORD ;SAVE PS
(2) 010336 106737 .WORD 106700!..C
(1) 010342 122737 000011 000432 CMPB #11,@#PSWORD ;IS PS = 11
(1) 010350 001403 BEQ .+10
(3) 010352 004767 006536 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;PS IS WRONG
(3) 010356 000201 201
(1) 010360 022702 177777 CMP #-1,%2 ;IS HIGH ORDER = -1
(1) 010364 001403 BEQ .+10
(3) 010366 004767 006522 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;HIGH ORDER IS WRONG
(3) 010372 000202 202
(1) 010374 022703 052524 CMP #52524,%2!1 ;IS LOW ORDER = 52524
(1) 010400 001403 BEQ .+10
(1) 010402 1$:
(3) 010402 004767 006506 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 010406 000203 203
(1) 010410 021527 000231 CMP (R5),#231
(1) 010414 001372 BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 010416 005215 INC (R5)

```

```

1455 ;*****
(1) ;TEST:232 MUL 125252 * #40000 = 165252 100000 PS 11
(1) ;*****
(1)

```

```

(1) TST232: SCOPE
(1) 010420 010701 MOV R5,R1 ;SAVE R5
(1) 010422 010501 MOV #125252,%4 ;LOAD MULTIPLICAND WITH 125252
(1) 010424 012704 125252 MUL #40000,%4 ;MULTIPLY 125252 * #40000
(1) 010430 070427 040000 MFPS @#PSWORD ;SAVE PS
(1) 010434 .WORD 106700!..C
(2) 010434 106737
(1) 010440 122737 000011 000432 CMPB #11,@#PSWORD ;IS PS = 11
(1) 010446 001403 BEQ .+10
(3) 010450 004767 006440 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;PS IS WRONG
(3) 010454 000204 204
(1) 010456 022704 165252 CMP #165252,%4 ;IS HIGH ORDER = 165252
(1) 010462 001403 BEQ .+10
(3) 010464 004767 006424 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;HIGH ORDER IS WRONG
(3) 010470 000205 205
(1) 010472 022705 100000 CMP #100000,%4!1 ;IS LOW ORDER = 100000
(1) 010476 001403 BEQ .+10
(1) 010500 1$:
(3) 010500 004767 006410 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 010504 000206 206
(1) 010506 021127 000232 CMP (R1),#232 ;CHECK THE TEST NUMBER
(1) 010512 001372 BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 010514 010105 MOV R1,R5 ;RESTORE R5
(1) 010516 005215 INC (R5)

```

1456
(1)
(1)
(1)
(1) 010520 010701
(1) 010522 012700 107070
(1) 010526 070027 107070
(1) 010532
(2) 010532 106737
(1) 010536 122737 000001 000432
(1) 010544 001403
(3) 010546 004767 006342
(3)
(3) 010552 000207
(1) 010554 022700 031222
(1) 010560 001403
(3) 010562 004767 006326
(3)
(3) 010566 000210
(1) 010570 022701 026100
(1) 010574 001403
(1) 010576
(3) 010576 004767 006312
(3)
(3) 010602 000211
(1) 010604 021527 000233
(1) 010610 001372
(1) 010612 005215

```
*****  
:TEST:233      MUL      107070 * #107070 = 31222 26100      PS = 1  
*****  
TST233: SCOPE  
MOV      #107070,%0      ;LOAD MULTIPLICAND WITH 107070  
MUL      #107070,%0      ;MULTIPLY 107070 * #107070  
MFPS     @#PSWORD      ;SAVE PS  
        .WORD      106700!..C  
CMPB     #1,@#PSWORD      ;IS PS = 1  
BEQ      .+10  
JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
        ;PS IS WRONG  
        207  
CMP      #31222,%0      ;IS HIGH ORDER = 31222  
BEQ      .+10  
JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
        ;HIGH ORDER IS WRONG  
        210  
CMP      #26100,%0!1      ;IS LOW ORDER = 26100  
BEQ      .+10  
1$:      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
        211  
CMP      (R5),#233  
BNE      1$      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
INC      (R5)
```

1457
(1)
(1)
(1)
(1) 010614 010701
(1) 010616 012701 177777
(1) 010622 070127 000001
(1) 010626
(2) 010626 106737
(1) 010632 122737 000010 000432
(1) 010640 001403
(3) 010642 004767 006246
(3)
(3) 010646 000212
(1) 010650 022701 177777
(1) 010654 001403
(3) 010656 004767 006232
(3)
(3) 010662 000213
(1) 010664 022701 177777
(1) 010670 001403
(1) 010672
(3) 010672 004767 006216
(3)
(3) 010676 000214
(1) 010700 021527 000234
(1) 010704 001372
(1) 010706 005215

```
*****  
:TEST:234      MUL      -1 * #1 = -1 -1      PS = 10  
*****  
TST234: SCOPE  
MOV      #-1,%1      ;LOAD MULTIPLICAND WITH -1  
MUL      #1,%1      ;MULTIPLY -1 * #1  
MFPS     @#PSWORD      ;SAVE PS  
        .WORD      106700!..C  
CMPB     #10,@#PSWORD      ;IS PS = 10  
BEQ      .+10  
JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
        ;PS IS WRONG  
        212  
CMP      #-1,%1      ;IS HIGH ORDER = -1  
BEQ      .+10  
JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
        ;HIGH ORDER IS WRONG  
        213  
CMP      #-1,%1.1      ;IS LOW ORDER = -1  
BEQ      .+10  
1$:      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
        214  
CMP      (R5),#234  
BNE      1$      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
INC      (R5)
```

1458

```
(1)
(1)
(1)
(1) 010710 010701
(1) 010712 012703 177777
(1) 010716 070327 000000
(1) 010722
(2) 010722 106737
(1) 010726 122737 000004 000432
(1) 010734 001403
(3) 010736 004767 006152
(3)
(3) 010742 000215
(1) 010744 022703 000000
(1) 010750 001403
(3) 010752 004767 006136
(3)
(3) 010756 000216
(1) 010760 022703 000000
(1) 010764 001403
(1) 010766
(3) 010766 004767 006122
(3)
(3) 010772 000217
(1) 010774 021527 000235
(1) 011000 001372
(1) 011002 005215
```

```
*****
:TEST:235      MUL      -1 * #0 = 0 0      PS = 4
*****
TST235: SCOPE
MOV      #-1,%3      ;LOAD MULTIPLICAND WITH -1
MUL      #0,%3      ;MULTIPLY -1 * #0
MFPS     @#PSWORD    ;SAVE PS
.WORD    106700!...C
CMPB     #4,@#PSWORD ;IS PS - 4
BEQ      .+10
JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;PS IS WRONG
                215
CMP      #0,%3      ;IS HIGH ORDER = 0
BEQ      .+10
JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;HIGH ORDER IS WRONG
                216
CMP      #0,%3.1    ;IS LOW ORDER = 0
BEQ      .+10
1$:      JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;LOW ORDER IS WRONG OR WRONG SEQUENCE
                217
CMP      (R5),#235
BNE      1$
INC      (R5)
                        ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
```

1459

```
(1)
(1)
(1)
(1) 011004 010701
(1) 011006 010501
(1) 011010 012705 077777
(1) 011014 070527 100000
(1) 011020
(2) 011020 106737
(1) 011024 122737 000011 000432
(1) 011032 001403
(3) 011034 004767 006054
(3)
(3) 011040 000220
(1) 011042 022705 100000
(1) 011046 001403
(3) 011050 004767 006040
(3)
(3) 011054 000221
(1) 011056 022705 100000
(1) 011062 001403
(1) 011064
(3) 011064 004767 006024
(3)
(3) 011070 000222
(1) 011072 021127 000236
(1) 011076 001372
```

```
*****
:TEST:236      MUL      77777 * #100000 = 100000 100000      PS = 11
*****
TST236: SCOPE
MOV      R5,R1      ;SAVE R5
MOV      #77777,%5   ;LOAD MULTIPLICAND WITH 77777
MUL      #100000,%5  ;MULTIPLY 77777 * #100000
MFPS     @#PSWORD    ;SAVE PS
.WORD    106700!...C
CMPB     #11,@#PSWORD ;IS PS = 11
BEQ      .+10
JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;PS IS WRONG
                220
CMP      #100000,%5  ;IS HIGH ORDER = 100000
BEQ      .+10
JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;HIGH ORDER IS WRONG
                221
CMP      #100000,%5.1 ;IS LOW ORDER - 100000
BEQ      .+10
1$:      JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;LOW ORDER IS WRONG OR WRONG SEQUENCE
                222
CMP      (R1),#236
BNE      1$
                        ;CHECK THE TEST NUMBER
                        ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
```

```

(1) 011100 010105          MOV    R1,R5          ;RESTORE R5
(1) 011102 005215          INC    (R5)
1460 (1) *****
(1) :TEST:237          MUL    -1 * #77777 = 100001 100001      PS = 10
(1) :*****
(1) TST237: SCOPE
(1) 011104 010701          MOV    #-1,%1          ;LOAD MULTIPLICAND WITH -1
(1) 011106 012701 177777    MUL    #77777,%1       ;MULTIPLY -1 * #77777
(1) 011112 070127 077777    MFPS  @#PSWORD         ;SAVE PS
(1) 011116          .WORD 106700!..C
(2) 011116 106737          .WORD 106700!..C
(1) 011122 122737 000010 000432  CMPB  #10,@#PSWORD     ;IS PS = 10
(1) 011130 001403          BEQ   .+10
(3) 011132 004767 005756    JSR   PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;PS IS WRONG
(3) 011136 000223          223
(1) 011140 022701 100001    CMP   #100001,%1      ;IS HIGH ORDER = 100001
(1) 011144 001403          BEQ   .+10
(3) 011146 004767 005742    JSR   PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;HIGH ORDER IS WRONG
(3) 011152 000224          224
(1) 011154 022701 100001    CMP   #100001,%1.1    ;IS LOW ORDER = 100001
(1) 011160 001403          BEQ   .+10
(1) 011162          1$:
(3) 011162 004767 005726    JSR   PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 011166 000225          225
(1) 011170 021527 000237    CMP   (R5),#237
(1) 011174 001372          BNE  1$
(1) 011176 005215          INC    (R5)
(1) :*****
1461 (1) :TEST:240          MUL    77777 * #77777 = 1 1      PS = 1
(1) :*****
(1) TST240: SCOPE
(1) 011200 010701          MOV    #77777,%3       ;LOAD MULTIPLICAND WITH 77777
(1) 011202 012703 077777    MUL    #77777,%3       ;MULTIPLY 77777 * #77777
(1) 011206 070327 077777    MFPS  @#PSWORD         ;SAVE PS
(1) 011212          .WORD 106700!..C
(2) 011212 106737          .WORD 106700!..C
(1) 011216 122737 000001 000432  CMPB  #1,@#PSWORD     ;IS PS = 1
(1) 011224 001403          BEQ   .+10
(3) 011226 004767 005662    JSR   PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;PS IS WRONG
(3) 011232 000226          226
(1) 011234 022703 000001    CMP   #1,%3           ;IS HIGH ORDER = 1
(1) 011240 001403          BEQ   .+10
(3) 011242 004767 005646    JSR   PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;HIGH ORDER IS WRONG
(3) 011246 000227          227
(1) 011250 022703 000001    CMP   #1,%3!1         ;IS LOW ORDER = 1
(1) 011254 001403          BEQ   .+10
(1) 011256          1$:
(3) 011256 004767 005632    JSR   PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 011262 000230          230
(1) 011264 021527 000240    CMP   (R5),#240

```

```

(1) 011270 001372          BNE 1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 011272 005215          INC (R5)
1462 :*****
(1) :TEST:241          MUL 2 * #2 = 4 4          PS = 0
(1) :*****
(1)
(1) 011274 010701          TST241: SCOPE
(1) 011276 010501          MOV R5,R1          ;SAVE R5
(1) 011300 012705 000002    MOV #2,%5          ;LOAD MULTIPLICAND WITH 2
(1) 011304 070527 000002    MUL #2,%5          ;MULTIPLY 2 * #2
(1) 011310          MFPS @#PSWORD      ;SAVE PS
(2) 011310 106737          .WORD 106700!..C
(1) 011314 122737 000000 000432    CMPB #0,@#PSWORD  ;IS PS = 0
(1) 011322 001403          BEQ .+10
(3) 011324 004767 005564          JSR PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)          ;PS IS WRONG
(3) 011330 000231          231
(1) 011332 022705 000004          CMP #4,%5          ;IS HIGH ORDER = 4
(1) 011336 001403          BEQ .+10
(3) 011340 004767 005550          JSR PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)          ;HIGH ORDER IS WRONG
(3) 011344 000232          232
(1) 011346 022705 000004          CMP #4,%5!1       ;IS LOW ORDER = 4
(1) 011352 001403          BEQ .+10
(1) 011354          1$:
(3) 011354 004767 005534          JSR PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)          ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 011360 000233          233
(1) 011362 021127 000241          CMP (R1),#241      ;CHECK THE TEST NUMBER
(1) 011366 001372          BNE 1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 011370 010105          MOV R1,R5          ;RESTORE R5
(1) 011372 005215          INC (R5)
1463 011374 012702 040000          MOV #40000,%2
1464 011400 012703 000464          MOV #5,%3
1465 011404 012704 000466          MOV #6,%4
1466 :*****
1467 :TEST:242          MUL 125252 * S5 = 165252 100000          PS = 11
(1) :*****
(1)
(1) 011410 010701          TST242: SCOPE
(1) 011412 012700 125252          MOV #125252,%0     ;LOAD MULTIPLICAND WITH 125252
(1) 011416 070067 167042          MUL S5,%0          ;MULTIPLY 125252 * S5
(1) 011422          MFPS @#PSWORD      ;SAVE PS
(2) 011422 106737          .WORD 106700!..C
(1) 011426 122737 000011 000432    CMPB #11,@#PSWORD ;IS PS = 11
(1) 011434 001403          BEQ .+10
(3) 011436 004767 005452          JSR PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)          ;PS IS WRONG
(3) 011442 000234          234
(1) 011444 022700 165252          CMP #165252,%0     ;IS HIGH ORDER = 165252
(1) 011450 001403          BEQ .+10
(3) 011452 004767 005436          JSR PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)          ;HIGH ORDER IS WRONG
(3) 011456 000235          235
(1) 011460 022701 100000          CMP #100000,%0!1  ;IS LOW ORDER = 100000

```

```
(1) 011464 001403          BEQ      .+10
(1) 011466
(3) 011466 004767 005422    1$:      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 011472 000236          236
(1) 011474 021527 000242    CMP      (R5),#242
(1) 011500 001372          BNE      1$      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 011502 005215          INC      (R5)
1468
(1)
(1)
(1)
(1) 011504 010701          TST243: SCOPE
(1) 011506 012700 125252    MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
(1) 011512 070077 166750    MUL      @S6,%0          ;MULTIPLY 125252 * @S6
(1) 011516          MFPS     @#PSWORD        ;SAVE PS
(2) 011516 106737          .WORD   106700!..C
(1) 011522 122737 000011 000432  CMPB     #11,@#PSWORD    ;IS PS = 11
(1) 011530 001403          BEQ      .+10
(3) 011532 004767 005356    JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;PS IS WRONG
(3) 011536 000237          237
(1) 011540 022700 165252    CMP      #165252,%0      ;IS HIGH ORDER = 165252
(1) 011544 001403          BEQ      .+10
(3) 011546 004767 005342    JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;HIGH ORDER IS WRONG
(3) 011552 000240          240
(1) 011554 022701 100000    CMP      #100000,%0!1    ;IS LOW ORDER = 100000
(1) 011560 001403          BEQ      .+10
(1) 011562
(3) 011562 004767 005326    1$:      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 011566 000241          241
(1) 011570 021527 000243    CMP      (R5),#243
(1) 011574 001372          BNE      1$      ;IF IN WPONG SEQUENCE GO TO THE HLT ABOVE
(1) 011576 005215          INC      (R5)
1469
(1)
(1)
(1)
(1) 011600 010701          TST244: SCOPE
(1) 011602 012700 125252    MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
(1) 011606 070037 000464    MUL      @#S5,%0          ;MULTIPLY 125252 * @#S5
(1) 011612          MFPS     @#PSWORD        ;SAVE PS
(2) 011612 106737          .WORD   106700!..C
(1) 011616 122737 000011 000432  CMPB     #11,@#PSWORD    ;IS PS = 11
(1) 011624 001403          BEQ      .+10
(3) 011626 004767 005262    JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;PS IS WRONG
(3) 011632 000242          242
(1) 011634 022700 165252    CMP      #165252,%0      ;IS HIGH ORDER = 165252
(1) 011640 001403          BEQ      .+10
(3) 011642 004767 005246    JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;HIGH ORDER IS WRONG
(3) 011646 000243          243
(1) 011650 022701 100000    CMP      #100000,%0!1    ;IS LOW ORDER = 100000
```

DVKABA MACY:1 30(1046) 14-SEP-81 16:32
CVKABB.P11 14-SEP-81 16:31

PAGE 4-50
MUL INSTRUCTION TESTS

SEQ 0055

(1) 011654 001403
 (1) 011656
 (3) 011656 004767 005232
 (3)
 (3) 011662 000244
 (1) 011664 021527 000244
 (1) 011670 001372
 (1) 011672 005215
 1470
 (1)
 (1)
 (1) 011674 010701
 (1) 011676 012700 125252
 (1) 011702 070002
 (1) 011704
 (2) 011704 106737
 (1) 011710 122737 000011 000432
 (1) 011716 001403
 (3) 011720 004767 005170
 (3)
 (3) 011724 000245
 (1) 011726 022700 165252
 (1) 011732 001403
 (3) 011734 004767 005154
 (3)
 (3) 011740 000246
 (1) 011742 022701 100000
 (1) 011746 001403
 (1) 011750
 (3) 011750 004767 005140
 (3)
 (3) 011754 000247
 (1) 011756 021527 000245
 (1) 011762 001372
 (1) 011764 005215
 1471
 (1)
 (1)
 (1)
 (1) 011766 010701
 (1) 011770 012700 125252
 (1) 011774 070023
 (1) 011776
 (2) 011776 106737
 (1) 012002 122737 000011 000432
 (1) 012010 001403
 (3) 012012 004767 005076
 (3)
 (3) 012016 000250
 (1) 012020 022700 165252
 (1) 012024 001403
 (3) 012026 004767 005062
 (3)
 (3) 012032 000251
 (1) 012034 022701 100000

```

      BEQ      .+10
1$:      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
      ;LOW ORDER IS WRONG OR WRONG SEQUENCE
      244
      CMP      (R5),#244
      BNE     1$      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
      INC     (R5)
;*****
;TEST:245      MUL      125252 * %2 = 165252 100000      PS = 11
;*****
TST245: SCOPE
      MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
      MUL     %2,%0          ;MULTIPLY 125252 * %2
      MFPS   @#PSWORD      ;SAVE PS
      .WORD  106700!..C
      CMPB   #11,@#PSWORD   ;IS PS = 11
      BEQ    .+10
      JSR    PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
      ;PS IS WRONG
      245
      CMP     #165252,%0     ;IS HIGH ORDER = 165252
      BEQ    .+10
      JSR    PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
      ;HIGH ORDER IS WRONG
      246
      CMP     #100000,%0!1   ;IS LOW ORDER = 100000
      BEQ    .+10
1$:      JSR    PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
      ;LOW ORDER IS WRONG OR WRONG SEQUENCE
      247
      CMP     (R5),#245
      BNE     1$      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
      INC     (R5)
;*****
;TEST:246      MUL      125252 * (3)+ = 165252 100000      PS = 11
;*****
TST246: SCOPE
      MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
      MUL     (3)+,%0        ;MULTIPLY 125252 * (3)+
      MFPS   @#PSWORD      ;SAVE PS
      .WORD  106700!..C
      CMPB   #11,@#PSWORD   ;IS PS = 11
      BEQ    .+10
      JSR    PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
      ;PS IS WRONG
      250
      CMP     #165252,%0     ;IS HIGH ORDER = 165252
      BEQ    .+10
      JSR    PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
      ;HIGH ORDER IS WRONG
      251
      CMP     #100000,%0!1   ;IS LOW ORDER = 100000

```



```

(1) 012040 001403          BEQ      .+10
(1) 012042          1$:      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) 012042 004767 005046          ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3)          252
(3) 012046 000252          CMP      (R5),#246
(1) 012050 021527 000246          BNE     1$      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 012054 001372          INC     (R5)
(1) 012056 005215
1472
(1)          :*****
(1)          :TEST:247      MUL      125252 * -(3) = 165252 100000      PS = 11
(1)          :*****
(1)
(1) 012060 010701          TST247: SCOPE
(1) 012062 012700 125252          MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
(1) 012066 070043          MUL     -(3),%0        ;MULTIPLY 125252 * -(3)
(1) 012070          MFPS   @#PSWORD        ;SAVE PS
(2) 012070 106737          .WORD  106700!..C
(1) 012074 122737 000011 000432          CMPB   #11,@#PSWORD    ;IS PS = 11
(1) 012102 001403          BEQ     .+10
(3) 012104 004767 005004          JSR     PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)          ;PS IS WRONG
(3) 012110 000253          253
(1) 012112 022700 165252          CMP     #165252,%0     ;IS HIGH ORDER = 165252
(1) 012116 001403          BEQ     .+10
(3) 012120 004767 004770          JSR     PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)          ;HIGH ORDER IS WRONG
(3) 012124 000254          254
(1) 012126 022701 100000          CMP     #100000,%0!1   ;IS LOW ORDER = 100000
(1) 012132 001403          BEQ     .+10
(1) 012134          1$:
(3) 012134 004767 004754          JSR     PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)          ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 012140 000255          255
(1) 012142 021527 000247          CMP     (R5),#247
(1) 012146 001372          BNE     1$      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 012150 005215          INC     (R5)
1473
(1)          :*****
(1)          :TEST:250      MUL      125252 * 2(4) = 165252 100000      PS = 11
(1)          :*****
(1)
(1) 012152 010701          TST250: SCOPE
(1) 012154 012700 125252          MOV     #125252,%0     ;LOAD MULTIPLICAND WITH 125252
(1) 012160 070064 000002          MUL     2(4),%0        ;MULTIPLY 125252 * 2(4)
(1) 012164          MFPS   @#PSWORD        ;SAVE PS
(2) 012164 106737          .WORD  106700!..C
(1) 012170 122737 000011 000432          CMPB   #11,@#PSWORD    ;IS PS = 11
(1) 012176 001403          BEQ     .+10
(3) 012200 004767 004710          JSR     PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)          ;PS IS WRONG
(3) 012204 000256          256
(1) 012206 022700 165252          CMP     #165252,%0     ;IS HIGH ORDER = 165252
(1) 012212 001403          BEQ     .+10
(3) 012214 004767 004674          JSR     PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)          ;HIGH ORDER IS WRONG
(3) 012220 000257          257
(1) 012222 022701 100000          CMP     #100000,%0!1   ;IS LOW ORDER = 100000

```

MUL INSTRUCTION TESTS

(1) 012226 001403
(1) 012230
(3) 012230 004767 004660
(3)
(3) 012234 000260
(1) 012236 021527 000250
(1) 012242 001372
(1) 012244 005215

1\$: BEQ .+10
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;LOW ORDER IS WRONG OR WRONG SEQUENCE
260
CMP (R5),#250
BNE 1\$;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
INC (R5)

1474

(1)
(1)
(1)

:TEST:251 MUL 125252 * @ (4) = 165252 100000 PS = 11

(1) 012246 010701
(1) 012250 012700 125252
(1) 012254 070074 000000
(1) 012260
(2) 012260 106737
(1) 012264 122737 000011 000432
(1) 012272 001403
(3) 012274 004767 004614
(3)

TST251: SCOPE
MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
MUL @ (4),%0 ;MULTIPLY 125252 * @ (4)
MFPS @#PSWORD ;SAVE PS
.WORD 106700!..C
CMPB #11,@#PSWORD ;IS PS = 11
BEQ .+10
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG

(3) 012300 000261
(1) 012302 022700 165252
(1) 012306 001403
(3) 012310 004767 004600
(3)

261
CMP #165252,%0 ;IS HIGH ORDER = 165252
BEQ .+10
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;HIGH ORDER IS WRONG

(3) 012314 000262
(1) 012316 022701 100000
(1) 012322 001403
(1) 012324
(3) 012324 004767 004564
(3)

262
CMP #100000,%0!1 ;IS LOW ORDER = 100000
BEQ .+10
1\$: JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;LOW ORDER IS WRONG OR WRONG SEQUENCE

(3) 012330 000263
(1) 012332 021527 000251
(1) 012336 001372
(1) 012340 005215

263
CMP (R5),#251 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
BNE 1\$
INC (R5)

1475

(1)
(1)
(1)

:TEST:252 MUL 125252 * @ (4)+ = 165252 100000 PS = 11

(1) 012342 010701
(1) 012344 012700 125252
(1) 012350 070034
(1) 012352
(2) 012352 106737
(1) 012356 122737 000011 000432
(1) 012364 001403
(3) 012366 004767 004522
(3)

TST252: SCOPE
MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
MUL @ (4)+,%0 ;MULTIPLY 125252 * @ (4)+
MFPS @#PSWORD ;SAVE PS
.WORD 106700!..C
CMPB #11,@#PSWORD ;IS PS = 11
BEQ .+10
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG

(3) 012372 000264
(1) 012374 022700 165252
(1) 012400 001403
(3) 012402 004767 004506
(3)

264
CMP #165252,%0 ;IS HIGH ORDER = 165252
BEQ .+10
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;HIGH ORDER IS WRONG

(3) 012406 000265
(1) 012410 022701 100000

265
CMP #100000,%0!1 ;IS LOW ORDER = 100000

```

(1) 012414 001403          BEQ      .+10
(1) 012416
(3) 012416 004767 004472    1$:     JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 012422 000266          266
(1) 012424 021527 000252    CMP      (R5),#252
(1) 012430 001372          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 012432 005215          INC     (R5)
1476
(1)
(1)
(1)
(1) 012434 010701          TST253: SCOPE
(1) 012436 012700 125252    MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
(1) 012442 070054          MUL     @-(4),%0        ;MULTIPLY 125252 * @-(4)
(1) 012444          MFPS   @#PSWORD        ;SAVE PS
(2) 012444 106737          .WORD  106700!..C
(1) 012450 122737 000011 000432  CMPB   #11,@#PSWORD    ;IS PS = 11
(1) 012456 001403          BEQ     .+10
(3) 012460 004767 004430    JSR     PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;PS IS WRONG
(3) 012464 000267          267
(1) 012466 022700 165252    CMP     #165252,%0      ;IS HIGH ORDER = 165252
(1) 012472 001403          BEQ     .+10
(3) 012474 004767 004414    JSR     PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;HIGH ORDER IS WRONG
(3) 012500 000270          270
(1) 012502 022701 100000    CMP     #100000,%0!1    ;IS LOW ORDER = 100000
(1) 012506 001403          BEQ     .+10
(1) 012510
(3) 012510 004767 004400    1$:     JSR     PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
(3) 012514 000271          271
(1) 012516 021527 000253    CMP     (R5),#253
(1) 012522 001372          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
(1) 012524 005215          INC     (R5)

```

```

:*****
:      DIV INSTRUCTION TESTS
:*****

```

```

1480
1481
1482
1483
1484
1485
1486
(1)
(1)
(1)
:*****
:TFST:254      DIV      0 4 / #2 = 2      REM = 0      PS  0
:*****

```

```

(1) 012526 010701          TST254: SCOPE
(1) 012530 012700 000000    MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
(1) 012534 012701 000004    MOV     #4,%0+1        ;LOAD LOW ORDER WITH 4
(1) 012540 071027 000002    DIV     #2,%0          ;DIVIDE BY #2
(1) 012544          MFPS   @#PSWORD        ;SAVE PS
(2) 012544 106737          .WORD  106700!..C
(1)
(1) 012550 122737 000000 000432  CMPB   #0,@#PSWORD    ;IS PS = 0
(1) 012556 001403          BEQ     .+10
(3) 012560 004767 004430    JSR     PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE

```

```
(3) ;PS IS WRONG
(3) 012564 000272 272
(1) 012566 022700 000002 CMP #2,%0 ;IS QUOTIENT = 2
(1) 012572 001403 BEQ .+10
(3) 012574 004767 004314 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;QUOTIENT IS WRONG
(3) 012600 000273 273
(1) 012602 022701 000000 CMP #0,%0+1 ;IS REMAINDER = 0
(1) 012606 001403 BEQ .+10
(3) 012610 004767 004300 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;WRONG REMAINDER
(3) 012614 000274 274
(1) 012616 021527 000254 CMP (R5),#254
(1) 012622 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
(3) 012624 004767 004264 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;TEST IS IN WRONG SEQUENCE
(3) 012630 000275 275
(1) 012632 005215 INC (R5)
(1)
1487 ;*****
(1) ;TEST:255 DIV -1 -9. / #3 -3 REM - 0 PS - 10
(1) ;*****
(1)
(1) 012634 010701 TST255: SCOPE
(1) 012636 012702 MOV #-1,%2 ;LOAD HIGH ORDER WITH -1
(1) 012642 012703 MOV #-9,%2+1 ;LOAD LOW ORDER WITH -9.
(1) 012646 071227 000003 DIV #3,%2 ;DIVIDE BY #3
(1) 012652 MFPS @#PSWORD ;SAVE PS
(2) 012652 106737 .WORD 106700!..C
(1)
(1) 012656 122737 000010 000432 CMPB #10,@#PSWORD ;IS PS = 10
(1) 012664 001403 BEQ .+10
(3) 012666 004767 004222 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;PS IS WRONG
(3) 012672 000276 276
(1) 012674 022702 177775 CMP #-3,%2 ;IS QUOTIENT = -3
(1) 012700 001403 BEQ .+10
(3) 012702 004767 004206 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;QUOTIENT IS WRONG
(3) 012706 000277 277
(1) 012710 022703 000000 CMP #0,%2+1 ;IS REMAINDER = 0
(1) 012714 001403 BEQ .+10
(3) 012716 004767 004172 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;WRONG REMAINDER
(3) 012722 000300 300
(1) 012724 021527 000255 CMP (R5),#255
(1) 012730 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
(3) 012732 004767 004156 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;TEST IS IN WRONG SEQUENCE
(3) 012736 000301 301
(1) 012740 005215 INC (R5)
(1)
```

1488

```
(1)
(1)
(1)
(1) 012742 010701
(1) 012744 010501
(1) 012746 012704 000000
(1) 012752 012705 000011
(1) 012756 071427 000002
(1) 012762
(2) 012762 106737
(1)
(1) 012766 122737 000000 000432
(1) 012774 001403
(3) 012776 004767 004112
(3)
(3) 013002 000302
(1)
(1) 013004 022704 000004
(1) 013010 001403
(3) 013012 004767 004076
(3)
(3) 013016 000303
(1)
(1) 013020 022705 000001
(1) 013024 001403
(3) 013026 004767 004062
(3)
(3) 013032 000304
(1) 013034 010105
(1) 013036 021527 000256
(1) 013042 001403
(3) 013044 004767 004044
(3)
(3) 013050 000305
(1) 013052 005215
(1)
```

```
*****
:TEST:256 DIV 0 9. / #2 = 4 REM - 1 PS = 0
*****
```

```
TST256: SCOPE
MOV R5,R1 ;SAVE R5
MOV #0,%4 ;LOAD HIGH ORDER WITH 0
MOV #9,%4+1 ;LOAD LOW ORDER WITH 9.
DIV #2,%4 ;DIVIDE BY #2
MFPS @#PSWORD ;SAVE PS
.WORD 106700!..C

CMPB #0,@#PSWORD ;IS PS = 0
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG
302

CMP #4,%4 ;IS QUOTIENT = 4
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;QUOTIENT IS WRONG
303

CMP #1,%4+1 ;IS REMAINDER = 1
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;WRONG REMAINDER
304

MOV R1,R5 ;RESTORE R5
CMP (R5),#256
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;TEST IS IN WRONG SEQUENCE
305
INC (R5)
```

1489

```
(1)
(1)
(1)
(1) 013054 010701
(1) 013056 012700 177777
(1) 013062 012701 177767
(1) 013066 071027 000002
(1) 013072
(2) 013072 106737
(1)
(1) 013076 122737 000010 000432
(1) 013104 001403
(3) 013106 004767 004002
(3)
(3) 013112 000306
(1)
(1) 013114 022700 177774
(1) 013120 001403
```

```
*****
:TEST:257 DIV -1 -9. / #2 - -4 REM = -1 PS = 10
*****
```

```
TST257: SCOPE
MOV #-1,%0 ;LOAD HIGH ORDER WITH -1
MOV #-9,%0+1 ;LOAD LOW ORDER WITH -9.
DIV #2,%0 ;DIVIDE BY #2
MFPS @#PSWORD ;SAVE PS
.WORD 106700!..C

CMPB #10,@#PSWORD ;IS PS = 10
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG
306

CMP #-4,%0 ;IS QUOTIENT - -4
BEQ .+10
```

```
(3) 013122 004767 003766 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;QUOTIENT IS WRONG
(3) 013126 000307 307
(1)
(1) 013130 022701 177777 CMP #0,%2 ;IS REMAINDER = -1
(1) 013134 001403 BEQ .+10
(3) 013136 004767 003752 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;WRONG REMAINDER
(3) 013142 000310 310
(1) 013144 021527 000257 CMP (R5),#257
(1) 013150 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
(3) 013152 004767 003736 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;TEST IS IN WRONG SEQUENCE
(3) 013156 000311 311
(1) 013160 005215 INC (R5)
(1)
```

```
1490
(1) :*****
(1) :TEST:260 DIV 0 2 / #-3 = 0 REM = 2 PS = 4
(1) :*****
```

```
(1) 013162 010701 TST260: SCOPE
(1) 013164 012702 000000 MOV #0,%2 ;LOAD HIGH ORDER WITH 0
(1) 013170 012703 000002 MOV #2,%2+1 ;LOAD LOW ORDER WITH 2
(1) 013174 071227 177775 DIV #-3,%2 ;DIVIDE BY #-3
(1) 013200 MFPS @#PSWORD ;SAVE PS
(2) 013200 106737 .WORD 106700!..C
(1)
(1) 013204 122737 000004 000432 CMPB #4,@#PSWORD ;IS PS = 4
(1) 013212 001403 BEQ .+10
(3) 013214 004767 003674 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;PS IS WRONG
(3) 013220 000312 312
(1)
(1) 013222 022702 000000 CMP #0,%2 ;IS QUOTIENT = 0
(1) 013226 001403 BEQ .+10
(3) 013230 004767 003660 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;QUOTIENT IS WRONG
(3) 013234 000313 313
(1)
(1) 013236 022703 000002 CMP #2,%2+1 ;IS REMAINDER = 2
(1) 013242 001403 BEQ .+10
(3) 013244 004767 003644 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;WRONG REMAINDER
(3) 013250 000314 314
(1) 013252 021527 000260 CMP (R5),#260
(1) 013256 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
(3) 013260 004767 003630 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;TEST IS IN WRONG SEQUENCE
(3) 013264 000315 315
(1) 013266 005215 INC (R5)
(1)
```

```
1491
(1) :*****
(1) :TEST:261 DIV -1 -2 / #3 = 0 REM = -2 PS = 4
(1) :*****
```

```
(1) 013270 010701 TST261: SCOPE
```

```

(1) 013272 010501      MOV      R5,R1      ;SAVE R5
(1) 013274 012704 177777  MOV      #-1,%4     ;LOAD HIGH ORDER WITH -1
(1) 013300 012705 177776  MOV      #-2,%4+1   ;LOAD LOW ORDER WITH -2
(1) 013304 071427 000003  DIV      #3,%4      ;DIVIDE BY #3
(1) 013310      MFPS      @#PSWORD ;SAVE PS
(2) 013310 106737      .WORD    106700!...C
(1)
(1) 013314 122737 000004 000432  CMPB     #4,@#PSWORD ;IS PS = 4
(1) 013322 001403      BEQ      .+10
(3) 013324 004767 003564      JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;PS IS WRONG
(3) 013330 000316      316
(1)
(1) 013332 022704 000000      CMP      #0,%4      ;IS QUOTIENT = 0
(1) 013336 001403      BEQ      .+10
(3) 013340 004767 003550      JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;QUOTIENT IS WRONG
(3) 013344 000317      317
(1)
(1) 013346 022705 177776      CMP      #-2,%4+1   ;IS REMAINDER = -2
(1) 013352 001403      BEQ      .+10
(3) 013354 004767 003534      JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;WRONG REMAINDER
(3) 013360 000320      320
(1) 013362 010105      MOV      R1,R5      ;RESTORE R5
(1) 013364 021527 000261      CMP      (R5),#261
(1) 013370 001403      BEQ      .+10
(3) 013372 004767 003516      JSR      PC,$HLT     ;IF IN WRONG SEQUENCE GO TO THE HLT
(3)                                     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;TEST IS IN WRONG SEQUENCE
(3) 013376 000321      321
(1) 013400 005215      INC      (R5)

```

```

1492
(1)
(1)
(1)
:*****
:TEST:262      DIV      -1 -1 / #1 = -1      REM = 0      PS = 10
:*****

```

```

(1) 013402 010701      TST262: SCOPE
(1) 013404 012700 177777  MOV      #-1,%0     ;LOAD HIGH ORDER WITH -1
(1) 013410 012701 177777  MOV      #-1,%0+1   ;LOAD LOW ORDER WITH -1
(1) 013414 071027 000001  DIV      #1,%0      ;DIVIDE BY #1
(1) 013420      MFPS      @#PSWORD ;SAVE PS
(2) 013420 106737      .WORD    106700!...C
(1)
(1) 013424 122737 000010 000432  CMPB     #10,@#PSWORD ;IS PS = 10
(1) 013432 001403      BEQ      .+10
(3) 013434 004767 003454      JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;PS IS WRONG
(3) 013440 000322      322
(1)
(1) 013442 022700 177777      CMP      #-1,%0     ;IS QUOTIENT = -1
(1) 013446 001403      BEQ      .+10
(3) 013450 004767 003440      JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;QUOTIENT IS WRONG
(3) 013454 000323      323
(1)
(1) 013456 022701 000000      CMP      #0,%0+1   ;IS REMAINDER = 0

```

```

(1) 013462 001403      BEQ      .+10
(3) 013464 004767 003424 JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;WRONG REMAINDER
(3) 013470 000324      324
(1) 013472 021527 000262 CMP      (R5),#262
(1) 013476 001403      BEQ      .+10
(3) 013500 004767 003410 JSR      PC,$HLT      ;IF IN WRONG SEQUENCE GO TO THE HLT
(3)                                     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;TEST IS IN WRONG SEQUENCE
(3) 013504 000325      325
(1) 013506 005215      INC      (R5)

```

1493

```

(1) (1) *****
(1) (1) :TEST:263      DIV      0 0 / #1 = 0      REM = 0      PS = 4
(1) (1) *****

```

```

(1) 013510 010701      TST263: SCOPE
(1) 013512 012700 000000 MOV      #0,%0      ;LOAD HIGH ORDER WITH 0
(1) 013516 012701 000000 MOV      #0,%0+1    ;LOAD LOW ORDER WITH 0
(1) 013522 071027 000001 DIV      #1,%0      ;DIVIDE BY #1
(1) 013526 MFPS      @#PSWORD  ;SAVE PS
(2) 013526 106737      .WORD    106700...C
(1) (1) 013532 122737 000004 000432 CMPB     #4,@#PSWORD ;IS PS = 4
(1) 013540 001403      BEQ      .+10
(3) 013542 004767 003346 JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;PS IS WRONG
(3) 013546 000326      326
(1) (1) 013550 022700 000000 CMP      #0,%0      ;IS QUOTIENT = 0
(1) 013554 001403      BEQ      .+10
(3) 013556 004767 003332 JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;QUOTIENT IS WRONG
(3) 013562 000327      327
(1) (1) 013564 022701 000000 CMP      #0,%0+1    ;IS REMAINDER = 0
(1) 013570 001403      BEQ      .+10
(3) 013572 004767 003316 JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;WRONG REMAINDER
(3) 013576 000330      330
(1) 013600 021527 000263 CMP      (R5),#263
(1) 013604 001403      BEQ      .+10
(3) 013606 004767 003302 JSR      PC,$HLT      ;IF IN WRONG SEQUENCE GO TO THE HLT
(3)                                     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;TEST IS IN WRONG SEQUENCE
(3) 013612 000331      331
(1) 013614 005215      INC      (R5)

```

1494

```

(1) (1) *****
(1) (1) :TEST:264      DIV      -1 125252 / #2 = 152525      REM = 0      PS = 10
(1) (1) *****

```

```

(1) 013616 010701      TST264: SCOPE
(1) 013620 012702 177777 MOV      #-1,%2     ;LOAD HIGH ORDER WITH -1
(1) 013624 012703 125252 MOV      #125252,%2+1 ;LOAD LOW ORDER WITH 125252
(1) 013630 071227 000002 DIV      #2,%2      ;DIVIDE BY #2
(1) 013634 MFPS      @#PSWORD  ;SAVE PS
(2) 013634 106737      .WORD    106700!...C

```



```

(1)
(1) 013640 122737 000010 000432      CMPB   #10,@#PSWORD      ;IS PS = 10
(1) 013646 001403                    BEQ    .+10
(3) 013650 004767 003240              JSR    PC,$HLT           ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;PS IS WRONG
(3) 013654 000332                    332
(1)
(1) 013656 022702 152525              CMP    #152525,%2       ;IS QUOTIENT = 152525
(1) 013662 001403                    BEQ    .+10
(3) 013664 004767 003224              JSR    PC,$HLT           ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;QUOTIENT IS WRONG
(3) 013670 000333                    333
(1)
(1) 013672 022703 000000              CMP    #0,%2+1          ;IS REMAINDER = 0
(1) 013676 001403                    BEQ    .+10
(3) 013700 004767 003210              JSR    PC,$HLT           ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;WRONG REMAINDER
(3) 013704 000334                    334
(1) 013706 021527 000264              CMP    (R5),#264
(1) 013712 001403                    BEQ    .+10
(3) 013714 004767 003174              JSR    PC,$HLT           ;IF IN WRONG SEQUENCE GO TO THE HLT
(3)                                     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;TEST IS IN WRONG SEQUENCE
(3) 013720 000335                    335
(1) 013722 005215                    INC    (R5)
(1)

```

```

1495
(1)
(1)
(1)
:*****
:TEST:265      DIV      -1 -1 / #-1 - 1      REM = 0      PS - 0
:*****

```

```

(1) 013724 010701      TST265: SCOPE
(1) 013726 010501      MOV    R5,R1            ;SAVE R5
(1) 013730 012704 177777  MOV    #-1,%4          ;LOAD HIGH ORDER WITH -1
(1) 013734 012705 177777  MOV    #-1,%4+1        ;LOAD LOW ORDER WITH -1
(1) 013740 071427 177777  DIV    #-1,%4          ;DIVIDE BY #-1
(1) 013744                    MFPS  @#PSWORD         ;SAVE PS
(2) 013744 106737      .WORD 106700!...C
(1)
(1) 013750 122737 000000 000432      CMPB   #0,@#PSWORD      ;IS PS = 0
(1) 013756 001403                    BEQ    .+10
(3) 013760 004767 003130              JSR    PC,$HLT           ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;PS IS WRONG
(3) 013764 000336                    336
(1)
(1) 013766 022704 000001              CMP    #1,%4           ;IS QUOTIENT = 1
(1) 013772 001403                    BEQ    .+10
(3) 013774 004767 003114              JSR    PC,$HLT           ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;QUOTIENT IS WRONG
(3) 014000 000337                    337
(1)
(1) 014002 022705 000000              CMP    #0,%4+1         ;IS REMAINDER = 0
(1) 014006 001403                    BEQ    .+10
(3) 014010 004767 003100              JSR    PC,$HLT           ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;WRONG REMAINDER
(3) 014014 003340                    340
(1) 014016 010105                    MOV    R1,R5            ;RESTORE R5
(1) 014020 021527 000265              CMP    (R5),#265

```

```

(1) 014024 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
(3) 014026 004767 003062 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;TEST IS IN WRONG SEQUENCE
(3) 014032 000341 341
(1) 014034 005215 INC (R5)

```

1496

```

:*****
:TEST:266 DIV 25253 1 / #125252 = 100000 REM = 1 PS = 10
:*****

```

```

(1) 014036 010701 TST266: SCOPE
(1) 014040 012700 025253 MOV #25253,%0 ;LOAD HIGH ORDER WITH 25253
(1) 014044 012701 000001 MOV #1,%0+1 ;LOAD LOW ORDER WITH 1
(1) 014050 071027 125252 DIV #125252,%0 ;DIVIDE BY #125252
(1) 014054 MFPS @#PSWORD ;SAVE PS
(2) 014054 106737 .WORD 106700!..C
(1) 014060 122737 000010 000432 CMPB #10,@#PSWORD ;IS PS = 10
(1) 014066 001403 BEQ .+10
(3) 014070 004767 003020 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;PS IS WRONG
(3) 014074 000342 342
(1) 014076 022700 100000 CMP #100000,%0 ;IS QUOTIENT = 100000
(1) 014102 001403 BEQ .+10
(3) 014104 004767 003004 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;QUOTIENT IS WRONG
(3) 014110 000343 343
(1) 014112 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
(1) 014116 001403 BEQ .+10
(3) 014120 004767 002770 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;WRONG REMAINDER
(3) 014124 000344 344
(1) 014126 021527 000266 CMP (R5),#266
(1) 014132 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
(3) 014134 004767 002754 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;TEST IS IN WRONG SEQUENCE
(3) 014140 000345 345
(1) 014142 005215 INC (R5)

```

1497

```

:*****
:TEST:267 DIV 37777 77777 / #77777 - 77777 REM = 77776 PS = 0
:*****

```

```

(1) 014144 010701 TST267: SCOPE
(1) 014146 012702 037777 MOV #37777,%2 ;LOAD HIGH ORDER WITH 37777
(1) 014152 012703 077777 MOV #77777,%2+1 ;LOAD LOW ORDER WITH 77777
(1) 014156 071227 077777 DIV #77777,%2 ;DIVIDE BY #77777
(1) 014162 MFPS @#PSWORD ;SAVE PS
(2) 014162 106737 .WORD 106700!..C
(1) 014166 122737 000000 000432 CMPB #0,@#PSWORD ;IS PS = 0
(1) 014174 001403 BEQ .+10
(3) 014176 004767 002712 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;PS IS WRONG

```

```
(3) 014202 000346          346
(1)
(1) 014204 022702 077777    CMP      #77777,%2      ;IS QUOTIENT = 77777
(1) 014210 001403          BEQ      .+10
(3) 014212 004767 002676    JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;QUOTIENT IS WRONG
(3) 014216 000347          347
(1)
(1) 014220 022703 077776    CMP      #77776,%2+1  ;IS REMAINDER = 77776
(1) 014224 001403          BEQ      .+10
(3) 014226 004767 002662    JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;WRONG REMAINDER
(3) 014232 000350          350
(1) 014234 021527 000267    CMP      (R5),#267
(1) 014240 001403          BEQ      .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
(3) 014242 004767 002646    JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;TEST IS IN WRONG SEQUENCE
(3) 014246 000351          351
(1) 014250 005215          INC      (R5)
(1)
1498 ;*****
(1) ;TEST:270      DIV      0 100000 / #2 = 40000      REM = 0      PS = 0
(1) ;*****
(1)
(1) 014252 010701          TST270: SCOPE
(1) 014254 010501          MOV      R5,R1          ;SAVE R5
(1) 014256 012704 000000    MOV      #0,%4          ;LOAD HIGH ORDER WITH 0
(1) 014262 012705 100000    MOV      #100000,%4+1  ;LOAD LOW ORDER WITH 100000
(1) 014266 071427 000002    DIV      #2,%4          ;DIVIDE BY #2
(1) 014272          MFPS      @#PSWORD      ;SAVE PS
(2) 014272 106737          .WORD   106700!..C
(1)
(1) 014276 122737 000000 000432  CMPB     #0,@#PSWORD    ;IS PS = 0
(1) 014304 001403          BEQ      .+10
(3) 014306 004767 002602    JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;PS IS WRONG
(3) 014312 000352          352
(1)
(1) 014314 022704 040000    CMP      #40000,%4     ;IS QUOTIENT = 40000
(1) 014320 001403          BEQ      .+10
(3) 014322 004767 002566    JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;QUOTIENT IS WRONG
(3) 014326 000353          353
(1)
(1) 014330 022705 000000    CMP      #0,%4+1      ;IS REMAINDER = 0
(1) 014334 001403          BEQ      .+10
(3) 014336 004767 002552    JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;WRONG REMAINDER
(3) 014342 000354          354
(1) 014344 010105          MOV      R1,R5          ;RESTORE R5
(1) 014346 021527 000270    CMP      (R5),#270
(1) 014352 001403          BEQ      .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
(3) 014354 004767 002534    JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;TEST IS IN WRONG SEQUENCE
(3) 014360 000355          355
(1) 014362 005215          INC      (R5)
```



```

(3)                                ;QUOTIENT IS WRONG
(3) 014544 000363                   363
(1)                                ;IS REMAINDER = 0
(1) 014546 022703 000900           CMP    #0,%2+1
(1) 014552 001403                   BEQ    .+10
(3) 014554 004767 002334           JSR    PC,$HLT                   ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;WRONG REMAINDER
(3) 014560 000364                   364
(1) 014562 021527 000272           CMP    (R5),#272
(1) 014566 001403                   BEQ    .+10
(3) 014570 004767 002320           JSR    PC,$HLT                   ;IF IN WRONG SEQUENCE GO TO THE HLT
(3)                                ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) 014574 000365                   365
(1) 014576 005215                   INC    (R5)
(1)

```

```

1501
(1) :*****
(1) :TEST:273      DIV      0 77777 / #0 = DUMMY      REM = DUMMY      PS      3
(1) :*****
(1)

```

```

(1) 014600 010701           TST273: SCOPE
(1) 014602 010501           MOV    R5,R1                   ;SAVE R5
(1) 014604 012704 000000     MOV    #0,%4                   ;LOAD HIGH ORDER WITH 0
(1) 014610 012705 077777     MOV    #77777,%4+1             ;LOAD LOW ORDER WITH 77777
(1) 014614 071427 000000     DIV    #0,%4                   ;DIVIDE BY #0
(1) 014620           MFPS  @#PSWORD                 ;SAVE PS
(2) 014620 106737           .WORD 106700!..C
(1) 014624 042737 000014 000432 BIC    #14,@#PSWORD
(1)
(1) 014632 122737 000003 000432 CMPB   #3,@#PSWORD             ;IS PS = 3
(1) 014640 001403           BEQ    .+10
(3) 014642 004767 002246     JSR    PC,$HLT                   ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;PS IS WRONG
(3) 014646 000366                   366
(1)
(1) 014650 010105           MOV    R1,R5                   ;RESTORE R5
(1) 014652 021527 000273     CMP    (R5),#273
(1) 014656 001403           BEQ    .+10
(3) 014660 004767 002230     JSR    PC,$HLT                   ;IF IN WRONG SEQUENCE GO TO THE HLT
(3)                                ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) 014664 000367                   367
(1) 014666 005215           INC    (R5)
(1)

```

```

1502
(1) :*****
(1) :TEST:274      DIV      77777 177777 / #2 = DUMMY      REM = DUMMY      PS = 2
(1) :*****
(1)

```

```

(1) 014670 010701           TST274: SCOPE
(1) 014672 012700 077777     MOV    #77777,%0               ;LOAD HIGH ORDER WITH 77777
(1) 014676 012701 177777     MOV    #177777,%0+1           ;LOAD LOW ORDER WITH 177777
(1) 014702 071027 000002     DIV    #2,%0                   ;DIVIDE BY #2
(1) 014706           MFPS  @#PSWORD                 ;SAVE PS
(2) 014706 106737           .WORD 106700!..C
(1) 014712 042737 000014 000432 BIC    #14,@#PSWORD
(1)
(1) 014720 122737 000002 000432 CMPB   #2,@#PSWORD             ;IS PS = 2
(1) 014726 001403           BEQ    .+10

```

```

(3) 014730 004767 002160      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;PS IS WRONG
(3) 014734 000370              370
(1)                               ;
(1) 014736 021527 000274      CMP      (R5),#274
(1) 014742 001403              BEQ      .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
(3) 014744 004767 002144      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;TEST IS IN WRONG SEQUENCE
(3) 014750 000371              371
(1) 014752 005215              INC      (R5)
(1)                               ;
1503 014754 012702 000002      MOV      #2,%2
1504 014760 012703 000474      MOV      #S9,%3
1505 014764 012704 000476      MOV      #S10,%4

```

```

1506
1507
(1) ;*****
(1) ;TEST:275      DIV      0 52525 / S9 = 25252      REM = 1      PS = 0
(1) ;*****

```

```

(1) 014770 010701              TST275: SCOPE
(1) 014772 012700 000000      MOV      #0,%0          ;LOAD HIGH ORDER WITH 0
(1) 014776 012701 052525      MOV      #52525,%0+1    ;LOAD LOW ORDER WITH 52525
(1) 015002 071067 163466      DIV      S9,%0          ;DIVIDE BY S9
(1) 015006                      MFPS      @#PSWORD      ;SAVE PS
(2) 015006 106737              .WORD    106700!..C
(1)                               ;
(1) 015012 122737 000000 000432  CMPB     #0,@#PSWORD     ;IS PS = 0
(1) 015020 001403              BEQ      .+10
(3) 015022 004767 002066      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;PS IS WRONG
(3) 015026 000372              372
(1)                               ;
(1) 015030 022700 025252      CMP      #25252,%0      ;IS QUOTIENT = 25252
(1) 015034 001403              BEQ      .+10
(3) 015036 004767 002052      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;QUOTIENT IS WRONG
(3) 015042 000373              373
(1)                               ;
(1) 015044 022701 000001      CMP      #1,%0+1        ;IS REMAINDER = 1
(1) 015050 001403              BEQ      .+10
(3) 015052 004767 002036      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;WRONG REMAINDER
(3) 015056 000374              374
(1) 015060 021527 000275      CMP      (R5),#275
(1) 015064 001403              BEQ      .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
(3) 015066 004767 002022      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;TEST IS IN WRONG SEQUENCE
(3) 015072 000375              375
(1) 015074 005215              INC      (R5)

```

```

1508
(1) ;*****
(1) ;TEST:276      DIV      0 52525 / @S10 = 25252      REM = 1      PS = 0
(1) ;*****

```

```

(1) 015076 010701              TST276: SCOPE
(1) 015100 012700 000000      MOV      #0,%0          ;LOAD HIGH ORDER WITH 0

```

```

(1) 015104 012701 052525      MOV      #52525,%0+1      ;LOAD LOW ORDER WITH 52525
(1) 015110 071077 163362      DIV      @S10,%0         ;DIVIDE BY @S10
(1) 015114                      MFPS     @#PSWORD        ;SAVE PS
(2) 015114 106737                      .WORD   106700!..C
(1)
(1) 015120 122737 000000 000432  CMPB     #0,@#PSWORD     ;IS PS = 0
(1) 015126 001403                      BEQ     .+10
(3) 015130 004767 001760      JSR     PC,$HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;PS IS WRONG
(3) 015134 000376                      376
(1)
(1) 015136 022700 025252      CMP      #25252,%0       ;IS QUOTIENT = 25252
(1) 015142 001403                      BEQ     .+10
(3) 015144 004767 001744      JSR     PC,$HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;QUOTIENT IS WRONG
(3) 015150 000377                      377
(1)
(1) 015152 022701 000001      CMP      #1,%0+1        ;IS REMAINDER = 1
(1) 015156 001403                      BEQ     .+10
(3) 015160 004767 001730      JSR     PC,$HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;WRONG REMAINDER
(3) 015164 000400                      400
(1) 015166 021527 000276      CMP      (R5),#276
(1) 015172 001403                      BEQ     .+10
(3) 015174 004767 001714      JSR     PC,$HLT         ;IF IN WRONG SEQUENCE GO TO THE HLT
(3)                               ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;TEST IS IN WRONG SEQUENCE
(3) 015200 000401                      401
(1) 015202 005215      INC     (R5)
(1)
1509
(1)
(1)
(1)
(1) 015204 010701      TST277: SCOPE
(1) 015206 012700 000000      MOV      #0,%0
(1) 015212 012701 052525      MOV      #52525,%0+1    ;LOAD HIGH ORDER WITH 0
(1) 015216 071037 000474      DIV      @#S9,%0        ;LOAD LOW ORDER WITH 52525
(1) 015222                      MFPS     @#PSWORD        ;DIVIDE BY @#S9
(2) 015222 106737                      .WORD   106700!..C      ;SAVE PS
(1)
(1) 015226 122737 000000 000432  CMPB     #0,@#PSWORD     ;IS PS = 0
(1) 015234 001403                      BEQ     .+10
(3) 015236 004767 001652      JSR     PC,$HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;PS IS WRONG
(3) 015242 000402                      402
(1)
(1) 015244 022700 025252      CMP      #25252,%0       ;IS QUOTIENT = 25252
(1) 015250 001403                      BEQ     .+10
(3) 015252 004767 001636      JSR     PC,$HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;QUOTIENT IS WRONG
(3) 015256 000403                      403
(1)
(1) 015260 022701 000001      CMP      #1,%0+1        ;IS REMAINDER = 1
(1) 015264 001403                      BEQ     .+10
(3) 015266 004767 001622      JSR     PC,$HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;WRONG REMAINDER

```

```

:*****
:TEST:277      DIV      0 52525 / @#S9 = 25252      REM = 1      PS = 0
:*****

```

```

(3) 015272 000404          404
(1) 015274 021527 000277  CMP      (R5),#277
(1) 015300 001403          BEQ      .+10      ;IF IN WRONG SEQUENCE GO TO THE HLT
(3) 015302 004767 001606  JSR      PC,$HLT  ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;TEST IS IN WRONG SEQUENCE
(3) 015306 000405          405
(1) 015310 005215          INC      (R5)
    
```

```

1510
(1)                                     ;*****
(1)                                     ;TEST:300      DIV      0 52525 / %2 = 25252      REM = 1      PS = 0
(1)                                     ;*****
    
```

```

(1) 015312 010701          TST300: SCOPE
(1) 015314 012700 000000  MOV      #0,%0      ;LOAD HIGH ORDER WITH 0
(1) 015320 012701 052525  MOV      #52525,%0+1 ;LOAD LOW ORDER WITH 52525
(1) 015324 071002          DIV      %2,%0      ;DIVIDE BY %2
(1) 015326          MFPS      @#PSWORD ;SAVE PS
(2) 015326 106737          .WORD   106700!..C
(1)
(1) 015332 122737 000000 000432  CMPB     #0,@#PSWORD ;IS PS = 0
(1) 015340 001403          BEQ      .+10
(3) 015342 004767 001546  JSR      PC,$HLT  ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;PS IS WRONG
(3) 015346 000406          406
(1)
(1) 015350 022700 025252  CMP      #25252,%0  ;IS QUOTIENT = 25252
(1) 015354 001403          BEQ      .+10
(3) 015356 004767 001532  JSR      PC,$HLT  ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;QUOTIENT IS WRONG
(3) 015362 000407          407
(1)
(1) 015364 022701 000001  CMP      #1,%0+1   ;IS REMAINDER = 1
(1) 015370 001403          BEQ      .+10
(3) 015372 004767 001516  JSR      PC,$HLT  ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;WRONG REMAINDER
(3) 015376 000410          410
(1) 015400 021527 000300  CMP      (R5),#300
(1) 015404 001403          BEQ      .+10      ;IF IN WRONG SEQUENCE GO TO THE HLT
(3) 015406 004767 001502  JSR      PC,$HLT  ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     ;TEST IS IN WRONG SEQUENCE
(3) 015412 000411          411
(1) 015414 005215          INC      (R5)
    
```

```

1511
(1)                                     ;*****
(1)                                     ;TEST:301      DIV      0 52525 / (3)+ = 25252      REM = 1      PS = 0
(1)                                     ;*****
    
```

```

(1) 015416 010701          TST301: SCOPE
(1) 015420 012700 000000  MOV      #0,%0      ;LOAD HIGH ORDER WITH 0
(1) 015424 012701 052525  MOV      #52525,%0+1 ;LOAD LOW ORDER WITH 52525
(1) 015430 071023          DIV      (3)+,%0   ;DIVIDE BY (3)+
(1) 015432          MFPS      @#PSWORD ;SAVE PS
(2) 015432 106737          .WORD   106700!..C
(1)
(1) 015436 122737 000000 000432  CMPB     #0,@#PSWORD ;IS PS = 0
(1) 015444 001403          BEQ      .+10
    
```


(3)	015446	004767	001442	JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)						:PS IS WRONG
(3)	015452	000412		412		
(1)	015454	022700	025252	CMP	#25252,%0	:IS QUOTIENT = 25252
(1)	015460	001403		BEQ	+.10	
(3)	015462	004767	001426	JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)						:QUOTIENT IS WRONG
(3)	015466	000413		413		
(1)	015470	022701	000001	CMP	#1,%0+1	:IS REMAINDER = 1
(1)	015474	001403		BEQ	+.10	
(3)	015476	004767	001412	JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)						:WRONG REMAINDER
(3)	015502	000414		414		
(1)	015504	021527	000301	CMP	(R5),#301	
(1)	015510	001403		BEQ	+.10	:IF IN WRONG SEQUENCE GO TO THE HLT
(3)	015512	004767	001376	JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)						:TEST IS IN WRONG SEQUENCE
(3)	015516	000415		415		
(1)	015520	005215		INC	(R5)	
(1)						
1512						
(1)						:*****
(1)						:TEST:302 DIV 0 52525 / -(3) = 25252 REM 1 PS - 0
(1)						:*****
(1)						
(1)	015522	010701		ST302: SCOPE		
(1)	015524	012700	000000	MOV	#0,%0	:LOAD HIGH ORDER WITH 0
(1)	015530	012701	052525	MOV	#52525,%0+1	:LOAD LOW ORDER WITH 52525
(1)	015534	071043		DIV	-(3),%0	:DIVIDE BY -(3)
(1)	015536			MFPSP	@#PSWORD	:SAVE PS
(2)	015536	106737		.WORD	106700!..C	
(1)						
(1)	015542	122737	000000 000432	CMPB	#0,@#PSWORD	:IS PS = 0
(1)	015550	001403		BEQ	+.10	
(3)	015552	004767	001336	JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)						:PS IS WRONG
(3)	015556	000416		416		
(1)						
(1)	015560	022700	025252	CMP	#25252,%0	:IS QUOTIENT = 25252
(1)	015564	001403		BEQ	+.10	
(3)	015566	004767	001322	JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)						:QUOTIENT IS WRONG
(3)	015572	000417		417		
(1)						
(1)	015574	022701	000001	CMP	#1,%0+1	:IS REMAINDER = 1
(1)	015600	001403		BEQ	+.10	
(3)	015602	004767	001306	JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)						:WRONG REMAINDER
(3)	015606	000420		420		
(1)	015610	021527	000302	CMP	(R5),#302	
(1)	015614	001403		BEQ	+.10	:IF IN WRONG SEQUENCE GO TO THE HLT
(3)	015616	004767	001272	JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)						:TEST IS IN WRONG SEQUENCE
(3)	015622	000421		421		
(1)	015624	005215		INC	(R5)	

```
(1)
1513
(1)
(1)
(1) 015626 010701
(1) 015630 012700 000000
(1) 015634 012701 052525
(1) 015640 071064 000002
(1) 015644
(2) 015644 106737
(1)
(1) 015650 122737 000000 000432
(1) 015656 001403
(3) 015660 004767 001230
(3)
(3) 015664 000422
(1)
(1) 015666 022700 025252
(1) 015672 001403
(3) 015674 004767 001214
(3)
(3) 015700 000423
(1)
(1) 015702 022701 000001
(1) 015706 001403
(3) 015710 004767 001200
(3)
(3) 015714 000424
(1) 015716 021527 000303
(1) 015722 001403
(3) 015724 004767 001164
(3)
(3) 015730 000425
(1) 015732 005215
(1)
```

```
*****
:TEST:303 DIV 0 52525 / 2(4) = 25252 REM 1 PS = 0
*****
TST303: SCOPE
MOV #0,%0 ;LOAD HIGH ORDER WITH 0
MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
DIV 2(4),%0 ;DIVIDE BY 2(4)
MFPS @#PSWORD ;SAVE PS
.WORD 106700!..C
CMPB #0,@#PSWORD ;IS PS = 0
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG
422
CMP #25252,%0 ;IS QUOTIENT = 25252
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;QUOTIENT IS WRONG
423
CMP #1,%0+1 ;IS REMAINDER = 1
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;WRONG REMAINDER
424
CMP (R5),#303
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;TEST IS IN WRONG SEQUENCE
425
INC (R5)
```

```
(1)
1514
(1)
(1)
(1) 015734 010701
(1) 015736 012700 000000
(1) 015742 012701 052525
(1) 015746 071074 000000
(1) 015752
(2) 015752 106737
(1)
(1) 015756 122737 000000 000432
(1) 015764 001403
(3) 015766 004767 001122
(3)
(3) 015772 000426
(1)
(1) 015774 022700 025252
(1) 016000 001403
(3) 016002 004767 001106
```

```
*****
:TEST:304 DIV 0 52525 / @ (4) = 25252 REM - 1 PS - 0
*****
TST304: SCOPE
MOV #0,%0 ;LOAD HIGH ORDER WITH 0
MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
DIV @ (4),%0 ;DIVIDE BY @ (4)
MFPS @#PSWORD ;SAVE PS
.WORD 106700!..C
CMPB #0,@#PSWORD ;IS PS - 0
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG
426
CMP #25252,%0 ;IS QUOTIENT = 25252
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
```

```

(3)                                     :QUOTIENT IS WRONG
(3) 016006 000427                       427
(1)
(1) 016010 022701 000001                CMP    #1,%0+1      :IS REMAINDER - 1
(1) 016014 001403                       BEQ    .+10
(3) 016016 004767 001072                JSR    PC,$HLT     :SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     :WRONG REMAINDER
(3) 016022 000430                       430
(1) 016024 021527 000304                CMP    (R5),#304
(1) 016030 001403                       BEQ    .+10        :IF IN WRONG SEQUENCE GO TO THE HLT
(3) 016032 004767 001056                JSR    PC,$HLT     :SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     :TEST IS IN WRONG SEQUENCE
(3) 016036 000431                       431
(1) 016040 005215                       INC    (R5)

1515
(1) :*****
(1) :TEST:305      DIV      0 52525 / @ (4)+ = 25252      REM = 1      PS = 0
(1) :*****
(1)
(1) 016042 010701                       TST305: SCOPE
(1) 016044 012700 000000                MOV    #0,%0      :LOAD HIGH ORDER WITH 0
(1) 016050 012701 052525                MOV    #52525,%0+1 :LOAD LOW ORDER WITH 52525
(1) 016054 071034                       DIV    @ (4)+,%0  :DIVIDE BY @ (4)+
(1) 016056                               MFPS   @#PSWORD   :SAVE PS
(2) 016056 106737                       .WORD  106700!..C
(1)
(1) 016062 122737 000000 000432          CMPB   #0,@#PSWORD :IS PS = 0
(1) 016070 001403                       BEQ    .+10
(3) 016072 004767 001016                JSR    PC,$HLT     :SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     :PS IS WRONG
(3) 016076 000432                       432
(1)
(1) 016100 022700 025252                CMP    #25252,%0  :IS QUOTIENT = 25252
(1) 016104 001403                       BEQ    .+10
(3) 016106 004767 001002                JSR    PC,$HLT     :SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     :QUOTIENT IS WRONG
(3) 016112 000433                       433
(1)
(1) 016114 022701 000001                CMP    #1,%0+1    :IS REMAINDER = 1
(1) 016120 001403                       BEQ    .+10
(3) 016122 004767 000766                JSR    PC,$HLT     :SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     :WRONG REMAINDER
(3) 016126 000434                       434
(1) 016130 021527 000305                CMP    (R5),#305
(1) 016134 001403                       BEQ    .+10        :IF IN WRONG SEQUENCE GO TO THE HLT
(3) 016136 004767 000752                JSR    PC,$HLT     :SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                     :TEST IS IN WRONG SEQUENCE
(3) 016142 000435                       435
(1) 016144 005215                       INC    (R5)

1516
(1) :*****
(1) :TEST:306      DIV      0 52525 / @-(4) = 25252      REM = 1      PS = 0
(1) :*****
(1)
(1) 016146 010701                       TST306: SCOPE
(1) 016150 012700 000000                MOV    #0,%0      :LOAD HIGH ORDER WITH 0
  
```

```

(1) 016154 012701 052525      MOV      #52525,%0+1      ;LOAD LOW ORDER WITH 52525
(1) 016160 071054              DIV      @-(4),%0        ;DIVIDE BY @-(4)
(1) 016162                      MFPS     @#PSWORD        ;SAVE PS
(2) 016162 106737              .WORD   106700!..C
(1)
(1) 016166 122737 000000 000432  CMPB    #0,@#PSWORD      ;IS PS = 0
(1) 016174 001403              BEQ     .+10
(3) 016176 004767 000712      JSR     PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;PS IS WRONG
(3) 016202 000436              436
(1)
(1) 016204 022700 025252      CMP     #25252,%0        ;IS QUOTIENT = 25252
(1) 016210 001403              BEQ     .+10
(3) 016212 004767 000676      JSR     PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;QUOTIENT IS WRONG
(3) 016216 000437              437
(1)
(1) 016220 022701 000001      CMP     #1,%0+1          ;IS REMAINDER = 1
(1) 016224 001403              BEQ     .+10
(3) 016226 004767 000662      JSR     PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;WRONG REMAINDER
(3) 016232 000440              440
(1) 016234 021527 000306      CMP     (R5),#306
(1) 016240 001403              BEQ     .+10              ;IF IN WRONG SEQUENCE GO TO THE HLT
(3) 016242 004767 000646      JSR     PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                               ;TEST IS IN WRONG SEQUENCE
(3) 016246 000441              441
(1) 016250 005215              INC     (R5)
(1)
1520
(1)                               ;*****
(1)                               ;TEST:307      TEST THAT EIS ABORTS PROPERLY WHEN INTERUPTED
(2)                               ;*****
(1)
(1) 016252 132737 000040 000421  TST307: BITB    #40,@#SENV      ;IF TYPE OUTS HAS BEEN SUPPRESSED
(1) 016260 001100              BNE     EASH+2           ;THEN SKIP THIS TEST
(1) 016262 013702 000502      MOV     @#TTYOUT,R2
(1) 016266 012722 016340      MOV     #R^A307,(R2)+   ;SET INTERUPT VECTOR TO RTA307
(1) 016272 012712 000340      MOV     #340,(R2)       ;AND THE INTERUPT PSW AS 340
(1) 016276                      MTPS    #0
(2) 016276 106427              .WORD   106400!..C
(1) 016302 012737 000030 000434  MOV     #30,@#TEMP1     ;PREPARE TO EXECUTE THIS SUB TEST 30 TIMES
(1) 016310 005004              CLR     R4
(1) 016312 112777 000015 162164  MOVB    #15,@$TPB        ;OUT PUT A "CR"
(1) 016320 112777 000100 162160  MOVB    #100,$$TPS       ;ENABLE TTY INTERUPT
(1) 016326 052704 000001      RTASH: BIS     #1,R4     ;PLACE A 1 IN R4
(1) 016332 072427 000020  ASHA:  ASH     #16,,R4    ;SHIFT R4 FOR 16 TIMES
(1) 016336 000773              BR      RTASH           ;STAY IN THE LOOP UNTIL INTERUPTED
(1) 016340 105077 162142  RTA307: CLRB   @$TPS     ;CLEAR TTY INTERUPT
(1) 016344 022716 016332      CMP     #ASHA,(SP)      ;IS THE RETURN ADDRESS = ASHA
(1) 016350 001415              BEQ     4$              ;IF SO THEN GO TO 4$
(1) 016352 012777 000015 162124  1$:  MOV     #15,@$TPB    ;OTHERWISE OUT PUT A "CR"
(1) 016360 105777 162122  2$:  TSTB   @$TPS         ;LOOP HERE UNTIL DONE COMES ON
(1) 016364 100375              BPL     2$
(1) 016366 012777 000015 162110  MOV     #15,@$TPB        ;OUT PUT ANOTHER "CR"
(1) 016374 012777 000100 162104  MOV     #100,$$TPS       ;ENABLE TTY INTERUPT
(1) 016402 000002              RTI
    
```

```

(1) 016404 020427 000001      4$:  CMP      R4,#1      ;CHECK R4 TO CONTAIN PROPER DATA
(1) 016410 001403              BEQ      6$
(3) 016412 004767 000476      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;R4 WAS CHANGED DURING THE EXECUTION OF
(3) 016416 000442              442
(1)                                ;THE INSTRUCTION
(1) 016420 032766 000360 000002 6$:  BIT      #360,2(SP)  ;CHECK THE PSW BEFORE INTERUPT
(1) 016426 001406              BEQ      8$
(3) 016430 004767 000460      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;PSW IS WRONG
(3) 016434 000443              443
(1) 016436 042766 000020 000002 8$:  BIC      #20,2(SP)  ;CLEAR THE T-BIT IF IT IS SEI
(1) 016444 005337 000434      DEC      @#TEMP1
(1) 016450 001340              BNE     1$           ;IF THE SUB TEST HAS BEEN EXECUTED 30 TIMES
(1)                                ;THEN GO TO THE END OF THE TEST
(1)
(1) 016452 010277 162024      MOV      R2,@TTYOUT  ;RESTORE TTY INTERUPT VECTOR
(1) 016456 005012              CLR      (R2)
(1) 016460 022626              EASH:  CMP      (SP)+,(SP)+ ;RESTORE THE STACK POINTER
(1) 016462 021527 000307      CMP      (R5),#307  ;CHECK THE TEST NUMBER
(1) 016466 001403              BEQ      .+10
(3) 016470 004767 000420      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3)                                ;TEST IS IN WRONG SEQUENCE
(3) 016474 000444              444
(1) 016476 005215              INC      (R5)
1521
(1)                                ;*****
(1)                                ;TEST:310      TEST THAT EIS ABORTS PROPERLY WHEN INTERUPTED
(2)                                ;*****
(1)
(1) 016500 132737 000040 000421 TST310: BITB     #40,@$ENVM  ;IF TYPE OUTS HAS BEEN SUPPRESSED
(1) 016506 001116              BNE     EMUL+2      ;THEN SKIP THIS TEST
(1) 016510 013702 000502      MOV      @#TTYOUT,R2
(1) 016514 012722 016572      MOV      #RTA310,(R2)+ ;SET INTERUPT VECTOR TO RTA310
(1) 016520 012712 000340      MOV      #340,(R2)    ;AND THE INTERUPT PSW AS 340
(1) 016524              MTPS     #10
(2) 016524 106427              .WORD   106400!..C
(1) 016530 012737 000030 000434  MOV      #30,@#TEMP1  ;PREPARE TO EXECUTE THIS SUB TEST 30 TIMES
(1) 016536 012704 077777      MOV      #77777,R4   ;PLACE THE MULTIPLIER IN R4
(1) 016542 012700 177777      MOV      #-1,R0      ;AND THE MULTIPLICAND IN R0
(1) 016546 012701 100001      MOV      #100001,R1  ;AND THE LOWER PART OF THE RESULT IN R1
(1) 016552 112777 000015 161724  MOVVB   #15,@$TPB    ;OUT PUT A 'CR'
(1) 016560 112777 000100 161720  MOVVB   #100,@$TPS   ;ENABLE TTY INTERUPT
(1) 016566 070004              RTMUL:  MUL      R4,R0 ;MULTIPLY R0 BY R4
(1) 016570 000776              BR      RTMUL        ;STAY IN THE LOOP UNTIL INTERUPTED
(1) 016572 105077 161710      RTA310: CLRB     @$TPS  ;CLEAR TTY INTERUPT
(1) 016576 022716 016566      CMP      #RTMUL,(SP) ;IS THE RETURN ADDRESS - RTMUL
(1) 016602 001415              BEQ     4$           ;IF SO THEN GO TO 4$
(1) 016604 012777 000015 161672 1$:  MOV      #15,@$TPB   ;OTHERWISE OUT PUT A 'CR'
(1) 016612 105777 161670      2$:  TSTB     @$TPS    ;LOOP HERE UNTIL DONE COMES ON
(1) 016616 100375              BPL     2$
(1) 016620 012777 000015 161656  MOV      #15,@$TPB   ;OUT PUT ANOTHER 'CR'
(1) 016626 012777 000100 161652  MOV      #100,@$TPS  ;ENABLE TTY INTERUPT
(1) 016634 000002              RTI
(1) 016636 020427 077777      4$:  CMP      R4,#77777  ;CHECK R4 TO CONTAIN PROPER DATA
(1) 016642 001403              BEQ     6$
(3) 016644 004767 000244      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
  
```

```
(3) ;R4 WAS CHANGED DURING THE EXECUTION OF
(3) 016650 000445 445 ;THE INSTRUCTION
(1) ;CHECK R0 TO CONTAIN PROPER DATA
(1) 016652 020027 177777 6$: CMP R0,#-1
(1) 016656 001403 8$ BEQ 8$
(3) 016660 004767 000230 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;R0 CONTAINS WRONG VALUE
(3) 016664 000446 446 ;CHECK R1 FOR THE PROPER DATA
(1) 016666 020127 100001 8$: CMP R1,#100001
(1) 016672 001403 10$ BEQ 10$
(3) 016674 004767 000214 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;R1 CONTAINS WRONG VALUE
(3) 016700 000447 447 ;CHECK THE PSW BEFORE INTERRUPT
(1) 016702 032766 000360 000002 10$: BIT #360,2(SP)
(1) 016710 001406 12$ BEQ 12$
(3) 016712 004767 000176 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;PSW IS WRONG
(3) 016716 000450 450 ;CLEAR THE T-BIT IF IT IS SET
(1) 016720 042766 000020 000002 BIC #20,2(SP)
(1) 016726 005337 000434 12$: DEC @TEMP1
(1) 016732 001324 1$ BNE 1$ ;IF THE SUB TEST HAS BEEN EXECUTED 30 TIMES
(1) ;THEN GO TO THE END OF THE TEST
(1) 016734 010277 161542 MOV R2,@TTYOUT ;RESTORE TTY INTERRUPT VECTOR
(1) 016740 005012 CLR (R2)
(1) 016742 022626 EMUL: CMP (SP)+,(SP)+ ;RESTORE THE STACK POINTER
(1) 016744 021527 000310 CMP (R5),#310 ;CHECK THE TEST NUMBER
(1) 016750 001403 BEQ .+10
(3) 016752 004767 000136 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
(3) ;TEST IS IN WRONG SEQUENCE
(3) 016756 000451 451
(1) 016760 005215 INC (R5)
1522 .SBTTL END OF PASS ROUTINE
(1)
(2) ;*****
(1) ;*INCREMENT THE PASS NUMBER ($PASS)
(1) ;*TYPE 'END PASS'
(1) ;*IF THERES A MONITOR GO TO IT
(1) ;*IF THERE ISN'T JUMP TO BEGIN
(1) ;*IF IT IS DESIRED TO HAVE A BELL INDICATE THE 'END OF PASS' LOCATION
(1) ;*$ENDMG CAN BE CHANGED TO 7.
(1) $EOP:
(1) 016762 SCOPE
(1) 016762 010701 INC $PASS ;:INCREMENT THE PASS NUMBER
(1) 016764 005267 161416 BIC #100000,$PASS ;:DON'T ALLOW A NEG. NUMBER
(1) 016770 042767 100000 161410 DEC (PC)+ ;:LOOP?
(1) 016776 005327 $EOPCT: .WORD 1
(1) 017000 000001 BGT $DOAGN ;:YES
(1) 017002 003017 MOV (PC)+,@(PC)+ ;:RESTORE COUNTER
(1) 017004 012737 $ENDCT: .WORD 1
(1) 017006 000001 $EOPCT
(1) 017010 017000 TYPE , $ENDMG ;:TYPE 'END PASS'
(1) 017012 000004 017051 TYPE , $ENULL ;:TYPE A NULL CHARACTER
(1) 017016 000004 017046 $GET42: MOV @#42,R0 ;:GET MONITOR ADDRESS
(1) 017022 013700 000042 BEQ $DOAGN ;:BRANCH IF NO MONITOR
(1) 017026 001405
```

```

(1) 017030 000005          RESET                ;;CLEAR THE WORLD
(1) 017032 004710          $ENDAD: JSR      PC,(R0)          ;;GO TO MONITOR
(1) 017034 000240          NOP                    ;;SAVE ROOM
(1) 017036 000240          NOP                    ;;FOR
(1) 017040 000240          NOP                    ;;ACT11
(1) 017042                $DOAGN:                ;;
(1) 017042 000137          JMP      @(PC)+          ;;RETURN
(1) 017044 000600          $RTNAD: .WORD  BEGIN          ;;
(1) 017046      377      377      000          $ENULL: .BYTE  -1,-1,0          ;;NULL CHARACTER STRING
(1) 017051      015      042412  042116          $ENDMG: .ASCIZ  <15><12>/END PASS/
(1) 017056 050040 051501 000123

1523
1528 017006 000004          ENDCT: 4

1532
1533
1534                ;;*****
1535
1536                .SBTTL  POWER FAIL ROUTINE
1537
1538 017064 012737 017074 000024          $PWRDN: MOV      #$PWRUP,@#24
1539 017072 000000          HALT
1540
1541 017074 012706 000600          $PWRUP: MOV      #BEGIN,SP          ;RESTORE THE SP
1542 017100 012737 017064 000024          MOV      #$PWRDN,@#24
1543 017106 000004 000516          TYPE    ,POWER          ;GO AND TYPE 'POWER'
1544 017112 000753          BR      $DOAGN
1548
1549                ;*      HALT ROUTINE
1550                ;*      -----
1551                ;*
1552                ;*
1553                ;*      PROGRAM COMES HERE ON ENCOUNTERING ANY ERROR
1554                ;*
1555
1556 017114 017637 000000 000402          $HLT:  MOV      @(SP),@#$FATAL          ;PLACE THE ERROR NUMBER AT LOCATION $FATAL
1557 017122 032737 020000 000422          BIT      #20000,@#$SWREG          ;HAS THE OPERATOR ASKED TO SUPPRESS ERROR TYPE OUTS
1558 017130 001046          BNE     6$
1559 017132 000004 000510          TYPE    , $CRLF          ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
1560 017136 010046          MOV     R0,-(SP)          ;SAVE R0
1561 017140 112767 000002 161303          MOV     #2,$TPCNT          ;ALLOW TYPE OUTS OF PC AND ERROR NUMBER
1562 017146 016600 000002          MOV     2(SP),R0          ;BRING THE RETURN PC IN R0
1563 017152 162700 000004          SUB     #4,R0
1564 017156 112737 000006 000450          2$:    MOV     #6,@#TYPCNT          ;ALLOW TYPE OUT OF 6 DIGITS
1565 017164 005046          CLR     -(SP)
1566 017166 000241          4$:    CLC
1567 017170 006100          ROL     R0
1568 017172 006116          ROL     (SP)          ;BRING THE C BIT FROM R0 IN (SP)
1569 017174 052716 000060          BIS     #60,(SP)          ;PREPARE TO TYPE IT OUT
1570 017200 004767 000130          JSR     PC,$TPCHR          ;AND GO TO OUT PUT A CHARACTER
1571 017204 005016          CLR     (SP)
1572 017206 006100          ROL     R0
1573 017210 006116          ROL     (SP)
1574 017212 006100          ROL     R0
1575 017214 006116          ROL     (SP)
1576 017216 105367 161226          DECB   TYPCNT          ;HAS ALL THE SIX CHARACTERS BEEN TYPED ?
1577 017222 001361          BNE    4$          ;IF NOT THEN REPEAT FROM 4$

```

1578	017224	005726			TST	(SP)+		:RESTORE STACK POINTER
1579	017226	017600	000002		MOV	@2(SP),R0		:PREPARE TO OUT PUT THE ERROR NUMBER
1580	017232	000004	000512		TYPE	,SCLF+2		:GO AND TYPE 3 SPACES
1581	017236	105367	161207		DECB	\$TPCNT		:IF BOTH PC AND ERROR NUMBER HAS NOT BEEN
1582	017242	001345			BNE	2\$:REPORTED THEN REPEAT FROM 2\$
1583	017244	012600			MOV	(SP)+,R0		:RESTORE R0
1584	017246	105767	161146	6\$:	TSTB	\$ENV		:IF WE ARE NOT UNDER APT. THEN GO TO
1585	017252	001403			BEQ	8\$:8\$
1586	017254	005237	000400		INC	@#\$MSGTY		:OTHERWISE INFORM APT. ABOUT SEEING THE ERROR
1587	017260	000777			BR	.		:AND LOOP
1588	017262	005737	000422	8\$:	TST	@\$\$SWREG		:IS IT REQUIRED TO HALT ON ERROR ?
1589	017266	100001			BPL	10\$:IF NOT THEN GO TO 10\$
1590	017270	000000			HALT			
1591	017272	062716	000002	10\$:	ADD	#2,(SP)		:ADJUST THE RETURN ADDRESS
1592	017276	000207			RTS	PC		:AND RETURN
1596								
1597					:*	TYPE OUT ROUTINE		
1598					:*	-----		
1599					:*			
1600					:*			
1601					:*	THIS ROUTINE IS USED TO TYPE ASCIZ MESSAGES		
1602					:*			
1603					:*			
1604	017300	010046			\$TYPE:	MOV	R0,-(SP)	:SAVE R0
1605	017302	017600	000002		MOV	@2(SP),R0		:GET THE ADDRESS OF THE ASSCIZ STRING
1606	017306	112046		2\$:	MOVB	(R0)+,-(SP)		:PUSH THE CHARACTER TO BE TYPED ONTO STACK
1607	017310	001005			BNE	4\$:BRANCH IF IT IS NOT THE TERMINATOR
1608	017312	005726			TST	(SP)+		
1609	017314	012600			MOV	(SP)+,R0		:OTHERWISE RESTORE THE STACK AND R0
1610	017316	062716	000002	3\$:	ADD	#2,(SP)		:ADJUST THE RETURN PC
1611	017322	000002			RTI			:AND RETURN
1612								
1613	017324	004767	000004	4\$:	JSR	PC,\$TPCHR		:GO TO TYPE A CHARACTER
1614	017330	005726			TST	(SP)+		:RESTORE THE STACK POINTER
1615	017332	000765			BR	2\$:AND RETURN TO 2\$
1616								
1617	017334	132737	000040	000421	\$TPCHR:	BITB	#40,@\$ENV	:HAS THE CONSOLE OUTPUTS BEEN SUPPRESSED?
1618	017342	001006			BNE	4\$:IF SO THEN RETURN FROM THE SUBROUTINE VIA 4\$
1619	017344	105777	161136	2\$:	TSTB	@\$TPS		:IS THE PRINTER AVAILABLE?
1620	017350	100375			BPL	2\$:IF NOT THEN LOOP HERE
1621	017352	116677	000002	161124	MOVB	2(SP),@\$TPB		:OUT PUT THE CHARACTER
1622	017360	000207		4\$:	RTS	PC		
1623		000001			.END			

TST211	006354	1331#
TST212	006454	1332#
TST213	006554	1333#
TST214	006654	1334#
TST215	006756	1335#
TST216	007060	1336#
TST217	007160	1337#
TST220	007260	1445#
TST221	007354	1446#
TST222	007450	1447#
TST223	007544	1448#
TST224	007644	1449#
TST225	007740	1450#
TST226	010034	1451#
TST227	010134	1452#
TST230	010230	1453#
TST231	010324	1454#
TST232	010420	1455#
TST233	010520	1456#
TST234	010614	1457#
TST235	010710	1458#
TST236	011004	1459#
TST237	011104	1460#
TST240	011200	1461#
TST241	011274	1462#
TST242	011410	1467#
TST243	011504	1468#
TST244	011600	1469#
TST245	011674	1470#
TST246	011766	1471#
TST247	012060	1472#
TST250	012152	1473#
TST251	012246	1474#
TST252	012342	1475#
TST253	012434	1476#
TST254	012526	1486#
TST255	012634	1487#
TST256	012742	1488#
TST257	013054	1489#
TST260	013162	1490#
TST261	013270	1491#
TST262	013402	1492#
TST263	013510	1493#
TST264	013616	1494#
TST265	013724	1495#
TST266	014036	1496#
TST267	014144	1497#
TST270	014252	1498#
TST271	014364	1499#
TST272	014472	1500#
TST273	014600	1501#
TST274	014670	1502#
TST275	014770	1507#
TST276	015076	1508#
TST277	015204	1509#
TST300	015312	1510#

TST301	015416	1511#												
TST302	015522	1512#												
TST303	015626	1513#												
TST304	015734	1514#												
TST305	016042	1515#												
TST306	016146	1516#												
TST307	016252	1520#												
TST310	016500	1521#												
TST37	002050	801	836	871	906	943	980	986#						
TST40	002104	987	993#											
TST41	002120	994	997#											
TST42	002136	998	1001#											
TST43	002172	1002	1008#											
TST44	002224	1009	1014#											
TST45	002256	1015	1020#											
TST46	002310	1021	1026#											
TST47	002346	1027	1033#											
TST50	002376	1034	1039#											
TST51	002454	1052#												
TST52	002534	1053#												
TST53	002614	1054#												
TST54	002674	1055#												
TST55	002752	1056#												
TST56	003030	1057#												
TST57	003106	1058#												
TST60	003166	1059#												
TST61	003246	1060#												
TST62	003324	1061#												
TTYOUT	000502	397#	1520*	1521*										
TYPCNT	000450	383#	1564*	1576*										
TYPE =	000004	357#	1522	1543	1559	1580								
\$APTHD	000430	366#	368											
\$CPUOP	000426	364#												
\$CRLF	000510	400#	1559	1580										
\$DEVCT	000410	364#	718											
\$DOAGN	017042	1522#	1544											
\$ENDAD	017032	342	1522#											
\$ENDCT	017006	1522#	1526											
\$ENDMG	017051	1522#												
\$ENULL	017046	1522#												
\$ENV	000420	364#	737	1584										
\$ENVM	000421	364#	1520	1521	1617									
\$EOP	016762	1522#												
\$EOPCT	017000	1522#												
\$ETABL	000420	364#												
\$ETEND	000430	364#	366											
\$FATAL	000402	364#	1556*											
\$GET42	017022	1522#												
\$HD =	000003	325												
\$HIBTS	000430	366#												
\$HLT	017114	783	787	819	823	854	858	889	893	925	929	962	966	1046
		1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1105	1109	1113
		1117	1148	1152	1156	1159	1191	1195	1199	1202	1311	1315	1316	1317
		1318	1319	1320	1321	1322	1328	1329	1330	1331	1332	1333	1334	1335
		1336	1337	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
		1456	1457	1458	1459	1460	1461	1462	1467	1468	1469	1470	1471	1472

	1473	1474	1475	1476	1486	1487	1488	1489	1490	1491	1492	1493	1494
	1495	1496	1497	1498	1499	1500	1501	1502	1507	1508	1509	1510	1511
	1512	1513	1514	1515	1516	1520	1521	1556#					
SMAIL 000400	364#	366	720										
SMBADR 000432	366#												
SMSGAD 000414	364#												
SMSGLG 000416	364#												
SMSGTY 000400	364#	1586*											
SPASS 000406	364#	767	804	839	874	910	947	1090	1133	1176	1522*		
SPASTM 000436	366#												
SPWRDN 017064	717	1538#	1542										
SPWRUP 017074	1538	1541#											
SRTNAD 017044	1522#												
SSETUP= 000020	403#	1522											
SSUP = 177777	403#												
SSVPC = 001000	342#												
SSWR = 160000	325#	1522											
SSWREG 000422	364#	1557	1588										
STESTN 000404	364#	726											
STN = 000001	325#												
STPB 000504	398#	1520*	1521*	1621*									
STPCHR 017334	1570	1613	1617#										
STPCNT 000451	384#	1561*	1581*										
STPS 000506	399#	739	1520*	1521*	1619								
STSTM 000434	366#												
STYPE 017300	361	1604#											
SUNIT 000412	364#												
SUNITM 000440	366#												
SUSWR 000424	364#												
SSGET4= 000000	1522#												
- 017362	332#	339	342#	360#	363#	366#	368#	370#	372#	374#	376#	378#	380#
	716#	724#	733#	777#	782	786	813#	818	822	848#	853	857	883#
	888	892	919#	924	928	956#	961	965	1045	1052#	1053#	1054#	1055#
	1056#	1057#	1058#	1059#	1060#	1061#	1099#	1104	1108	1111	1116	1142#	1147
	1151	1154	1158	1185#	1190	1194	1197	1201	1310	1315#	1316#	1317#	1318#
	1319#	1320#	1321#	1322#	1328#	1329#	1330#	1331#	1332#	1333#	1334#	1335#	1336#
	1337#	1445#	1446#	1447#	1448#	1449#	1450#	1451#	1452#	1453#	1454#	1455#	1456#
	1457#	1458#	1459#	1460#	1461#	1462#	1467#	1468#	1469#	1470#	1471#	1472#	1473#
	1474#	1475#	1476#	1486#	1487#	1488#	1489#	1490#	1491#	1492#	1493#	1494#	1495#
	1496#	1497#	1498#	1499#	1500#	1501#	1502#	1507#	1508#	1509#	1510#	1511#	1512#
	1513#	1514#	1515#	1516#	1520#	1521#	1522	1525	1526#	1530#	1587		
.SX - 000430	366#												
..A - 016524	733#	777#	813#	848#	883#	919#	956#	1052#	1053#	1054#	1055#	1056#	1057#
	1058#	1059#	1060#	1061#	1099#	1142#	1185#	1315#	1316#	1317#	1318#	1319#	1320#
	1321#	1322#	1328#	1329#	1330#	1331#	1332#	1333#	1334#	1335#	1336#	1337#	1445#
	1446#	1447#	1448#	1449#	1450#	1451#	1452#	1453#	1454#	1455#	1456#	1457#	1458#
	1459#	1460#	1461#	1462#	1467#	1468#	1469#	1470#	1471#	1472#	1473#	1474#	1475#
	1476#	1486#	1487#	1488#	1489#	1490#	1491#	1492#	1493#	1494#	1495#	1496#	1497#
	1498#	1499#	1500#	1501#	1502#	1507#	1508#	1509#	1510#	1511#	1512#	1513#	1514#
	1515#	1516#	1520#	1521#									
..B - 016530	733#	777#	813#	848#	883#	919#	956#	1052#	1053#	1054#	1055#	1056#	1057#
	1058#	1059#	1060#	1061#	1099#	1142#	1185#	1315#	1316#	1317#	1318#	1319#	1320#
	1321#	1322#	1328#	1329#	1330#	1331#	1332#	1333#	1334#	1335#	1336#	1337#	1445#
	1446#	1447#	1448#	1449#	1450#	1451#	1452#	1453#	1454#	1455#	1456#	1457#	1458#
	1459#	1460#	1461#	1462#	1467#	1468#	1469#	1470#	1471#	1472#	1473#	1474#	1475#
	1476#	1486#	1487#	1488#	1489#	1490#	1491#	1492#	1493#	1494#	1495#	1496#	1497#

1498#	1499#	1500#	1501#	1502#	1507#	1508#	1509#	1510#	1511#	1512#	1513#	1514#
1515#	1516#	1520#	1521#									
733#	777#	813#	848#	883#	919#	956#	1052#	1053#	1054#	1055#	1056#	1057#
1058#	1059#	1060#	1061#	1099#	1142#	1185#	1315#	1316#	1317#	1318#	1319#	1320#
1321#	1322#	1328#	1329#	1330#	1331#	1332#	1333#	1334#	1335#	1336#	1337#	1445#
1446#	1447#	1448#	1449#	1450#	1451#	1452#	1453#	1454#	1455#	1456#	1457#	1458#
1459#	1460#	1461#	1462#	1467#	1468#	1469#	1470#	1471#	1472#	1473#	1474#	1475#
1476#	1486#	1487#	1488#	1489#	1490#	1491#	1492#	1493#	1494#	1495#	1496#	1497#
1498#	1499#	1500#	1501#	1502#	1507#	1508#	1509#	1510#	1511#	1512#	1513#	1514#
1515#	1516#	1520#	1521#									

. ABS. 017362 000

ERRORS DETECTED: 0

CVKABB, CVKABB/CRF: SYM/NL: TOC=SYSMAC.SML/ML, CVKABB.P11
 RUN-TIME: 10 11 .8 SECONDS
 RUN-TIME RATIO: 83/23=3.5
 CORE USED: 15K (30 PAGES)