

This microfiche card contains a grid of frames. The left side of the card is filled with frames containing data, while the right side is mostly blank. The data in the frames is organized into columns and rows, with some frames containing headers and footers. The data appears to be related to an EIS instruction test, as indicated by the header. The frames contain various alphanumeric characters, including numbers and letters, arranged in a structured format. The overall layout is typical of a microfiche card used for data storage and retrieval.

PDP11/34

EIS INSTRUCTION TESTS
AH-0018A-MC

11-DFKAC-A-DL-A

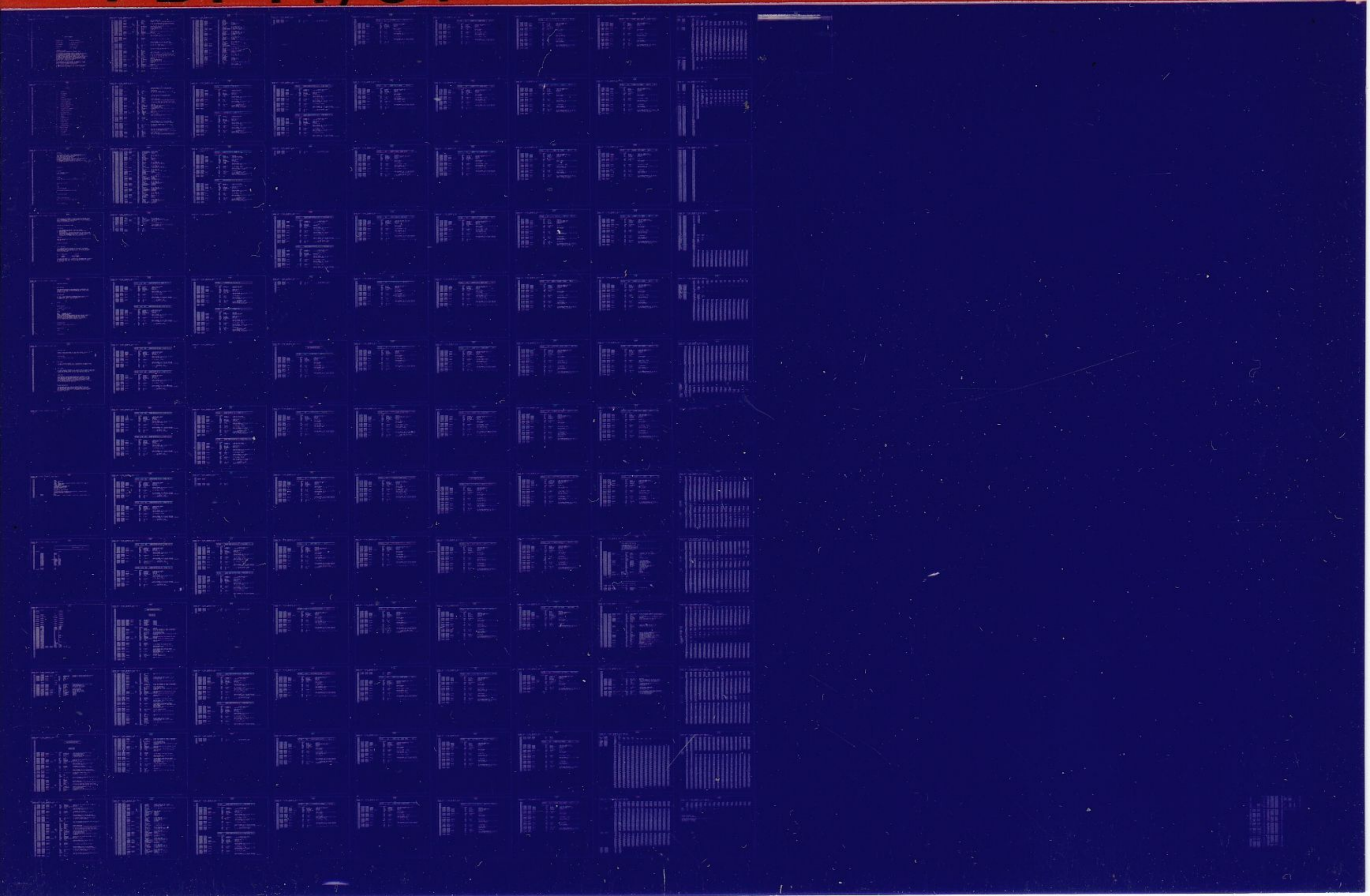
OCT 1976

COPYRIGHT ©1976

digital

FICHE 1 OF 1

Made in U.S.A.



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
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

1. ABSTRACT

THIS PROGRAM TESTS THE 11/34 EXTENDED INSTRUCTION SET
(ASH, ASHC, MUL, AND DIV) USING REGISTERS 0-5 AT-
LEAST ONCE WITH EACH INSTRUCTION.
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.
THIS PROGRAM IS A MODIFICATION OF THE LSI-11 EIS TEST.
IT HAS BEEN MODIFIED TO ACCOUNT FOR ANY LSI-11 - 11/34 DIFFERENCES.

2. REQUIREMENTS

2.1 EQUIPMENT

11/34 STANDARD COMPUTER
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)

4.2 STARTING ADDRESS

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

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) LOAD PROGRAM INTO MEMORY USING ABS LOADER
- 2) SET SWITCHES (SEE SEC 5.1) ALL LOW FOR WORST CASE.
- 3) START AT 200.
- 4) 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 SENVM IS HIGH
- 5) 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).

5. OPERATING PROCEDURE

5.1 SWITCH SETTINGS

IF NO HARDWARE SWITCH REGISTER IS AVAILABLE, THE PROGRAM AUTOMATICALLY USES THE CONTENTS OF LOC. 176 AS THE SOFTWARE SWITCH REGISTER. THE USER SHOULD SET THIS LOCATION BEFORE STARTING THE PROGRAM.

BIT #	OCTAL VALUE	FUNCTION
15	10000.....	HALT ON ERROR
13	02000.....	INHIBIT PRINTOUT

AN 8 BIT BYTE SENVM (I.E. LOCATION 421) HAS BEEN USED TO DEFINE THE OPERATING MODE. ALL TYPEOUTS CAN BE SUPPRESSED BY MAKING BIT 5 OF BYTE SENVM HIGH, IN OTHER WORDS BY PLING A 20000 IN LOCATION 420

206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
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

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 SUPPRESSED 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 SENVM 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

262
263
264
265
266
267
268
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

8.1 EXECUTION TIME

NORMALLY FIRST "END PASS" WILL BE TYPED WITHIN 1 SECOND AND WITHIN 10 SECONDS FOR EVERY CONSECUTIVE 400 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 BYT2 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 11/34 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

H01

DFKACA MACY11 27(732) 08-SEP-76 18:20 PAGE 7
DFKACA.P11

309

010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029

000001
160000

```
.ABS  
.NLIST MD,MC,CND  
.LIST ME  
.MCALL .SEOP, .SACT11, .SAPTBL5, .SAPTHDR, .HEADER, .SETUP, PUSH, POP  
.MCALL STARS, SWRSU  
.TITLE DFKACA  
:*COPYRIGHT (C) DECEMBER 1975  
:*DIGITAL EQUIPMENT CORP.  
:*MAYNARD, MASS. 01754  
:*  
:*PROGRAM BY GLENN JOHNSON  
:*  
:*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC  
:*PACKAGE (MAINDEC-11-DZQAC-A2).  
:*  
$TN=1  
$SWR=160000 ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT
```

330			*****
331			*****
332			*****
333	000000		.=0 ;TRAP CATCHER 0 - 776
334			*****
335			*****
336			*****
0 337	001000		.SACT11
338			
339	000000	DUMMY=	0
340	000001	FERRNM=	1
341	000051	F=	51
342	000176	N=	176
343	000007	PC=	%7
344	000006	SP=	%6
345	010701	SCOPE=	10701
346	010701	SCOPE1=	10701
347	010703	SCOPE3=	10703
348	001000	SW09=	1000
349	002000	SW10=	2000
350	004000	SW11=	4000
351	010000	SW12=	10000
352	000004	TYPE=	IOT
353			
354			
355	000020	.=20	
356	000020 016572	\$TYPE	

K01

0	357		000400		.=400			
	358	000400				.\$APTBL		\$CPUOP
0	359							
	360	000400				.\$APTHDR		3,5
U	361							
	362		000000			.\$APTHD		
	363	000000			COUNT:	.=COUNT+2		
	364		000002		PSWORD:	.=PSWORD+2		
	365	000002			TEMP1:	.=TEMP1+2		
	366		000004		TEMP2:	.=TEMP2+2		
	367	000004			TEMP3:	.=TEMP3+2		
	368		000006		TEMP4:	.=TEMP4+2		
	369	000006			TEMP5:	.WORD		
	370		000010		TEMP6:	.WORD		
	371	000010			TYPCNT:	.BYTE		
	372		000012		\$TPCNT:	.BYTE		
	373	000012			S0:	7		
	374		000014		S1:	-7		
	375	000014	000000		S2:	S1		
	376	000016	000000		S3:	-6		
	377	000020	000		S4:	-1		
	378	000021	000		S5:	40000		
	379	000022	000007		S6:	S5		
	380	000024	177771		S7:	40000		
	381	000026	000024		S8:	-2		
	382	000030	177772		S9:	2		
	383	000032	177777		S10:	S9		
	384	000034	040000		S11:	2		
	385	000036	000034		SWR:	177570		
	386	000040	040000		DISPLAY:	177570		
	387	000042	177776		TTYOUT:	64		
	388	000044	000002		\$TPB:	177566		
	389	000046	000044		\$TPS:	177564		
	390	000050	000002		\$CRLF:	.ASCIZ <15><12>/ /		
	391	000052	177570	020040	POWER:	.ASCIZ <12><15>/POWER/		
	392	000054	177570	047520				
	393	000056	000064					
	394	000060	177566					
	395	000062	177564					
	396	000064	005015					
	397	000072	006412					
	398	000100	000122					
	399							
	400							
	401							
	402							
	403							
	404							
	405							
	406							
	407							
	408							

DFKACA MACY11 27(732) 08-SEP-76 18:20 PAGE 11
 DFKACA.P11 STARTING OF THE PROGRAM

```

409      000200      000200      . =200
U 410 000200 012737 016356 000024      MOV      #SPWRDN,2#24      ;PREPARE TO SERVICE POWER DOWN ROUTINE
U 411 000206 012700 000000      MOV      #SDEVCT,RO      ;PREPARE TO INITIALIZE THE STACK
U 412 000212 005040      2$:      CLR      -(RO)
U 413 000214 022700 000000      CMP      #SMAIL,RO
U 414 000220 001374      BNE      2$
415 000222 000167 000352      RESTRT: JMP      BEGIN
416
417      000600      . =600
U 418
U 419 000600 012705 000000      BEGIN:  MOV      #STESTN,R5      ;MAKE R5 POINT TO THE LOCATION $TESTN
U 420 000604 005037 000000      CLR      2#COUNT      ;CLEAR THE COUNTER
U 421 000610 012715 000001      MOV      #1,(R5)      ;INITIALIZE TEST NUMBER
O 422 000614 012706 000600      MOV      #BEGIN,SP      ;** STACK AT BEGIN **
U 423 000620      SWRSU      1$,3$
U 424 000620 106427 000000      MTPS     #0      ;PLACE #0 IN PSW
U 425 000624 132737 000001 000000      BITB     #1,2#SENV      ;ARE WE UNDER APT ?
U 426 000632 001403      BEQ      2$      ;IF NOT THEN GO TO 2$
U 427 000634 012767 000000 177210      MOV      #SSWREG,SWR      ;USE APT SWITCH REGISTER
U 428 000642 012737 000001 000004      2$:      MOV      #1,2#TEMP1      ;TEMP1=1
U 429 000650 005037 000006      CLR      2#TEMP2      ;TEMP2=0
U 430 000654 012737 000001 000010      MOV      #1,2#TEMP3      ;TEMP3=1
U 431 000662 005037 000012      CLR      2#TEMP4      ;TEMP4=0
432
433

```

MO1

434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489

```
*****
: ASH INSTRUCTION TESTS
: *****
```

```
*****
: TESTS 1-36
: *****
```

000666	010701			START:	SCOPE1		
000670	013700	000004			MOV	2#TEMP1,%0	:LOAD R0 WITH THE CONTENTS OF TEMP1
000674	032737	000001	016336		BIT	#1,2#SPASS	:IS IT AN EVEN PASS ?
000702	001004				BNE	2\$:IF NOT THEN GO TO 2\$
000704	013701	000006			MOV	2#TEMP2,R1	:OTHERWISE EXECUTE THE INSTRUCTION
							:IN MODE 0 USING R1
000710	072001				ASH	R1,R0	
000712	000402				BR	4\$	
000714	072067	177066		2\$:	ASH	TEMP2,%0	:SHIFT R0 BY THE NUMBER SPECIFIED BY TEMP2
000720	106737	000002		4\$:	MFPS	2#PSWORD	:SAVE PS
000724	123737	000012	000002		CMPB	2#TEMP4,2#PSWORD	:IS THE PS = TEMP4 ?
000732	001403				BEQ	.+10	
000734	004767	015446			JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
							:THE PS IS NOT EQUAL TO 0
000740	000001				1		
000742	005237	000000			INC	2#COUNT	:INCREMENT THE COUNTER
000746	023700	000010			CMP	2#TEMP3,%0	:IS THE RESULT IN R0 EQUAL TO TEMP3?
000752	001403				BEQ	.+10	
000754	004767	015426		6\$:	JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
							:EITHER INCORRECT R0 OR INCORRECT SEQUENCE
000760	000002				2		
000762	021537	000000			CMP	(R5),2#COUNT	:IS THE TEST NUMBER EQUAL TO THE
							:COUNTER?
000766	001372				BNE	6\$:IF NOT GO TO THE HLT ABOVE
000770	005215				INC	(R5)	
000772	010701				SCOPE1		
000774	021527	000037			CMP	(R5),#37	:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
							:BY 14. AND RIGHT BY 14.?
001000	002011				BGE	8\$	
001002	005237	000006			INC	2#TEMP2	
001006	006367	176776			ASL	TEMP3	:SHIFT TEMP3 LEFT.
001012	021527	000020			CMP	(R5),#20	:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
001016	001004				BNE	REG1	
001020	000167	000764			JMP	NEGAT	:IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
001024	004767	001006		8\$:	JSR	PC,TST37	:IF SO GO AND CONTINUE THE REST OF THE PROGRAM
001030	010703			REG1:	SCOPE3		
001032	013701	000004			MOV	2#TEMP1,%1	:LOAD R1 WITH THE CONTENTS OF TEMP1
001036	032737	000001	016336		BIT	#1,2#SPASS	:IS IT AN EVEN PASS ?
001044	001004				BNE	2\$:IF NOT THEN GO TO 2\$
001046	013702	000006			MOV	2#TEMP2,R2	:OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
001052	072102				ASH	R2,R1	:USING R1
001054	000402				BR	4\$	

490	001056	072167	176724		2\$:	ASH	TEMP2,%1	;SHIFT R1 BY THE NUMBER SPECIFIED BY TEMP2
491	001062	106737	000002		4\$:	MFPS	@#PSWORD	;SAVE PS
492	001066	123737	000012	000002		CMPB	@#TEMP4,@#PSWORD	;IS THE PS = TEMP4 ?
493	001074	001403				BEQ	.+10	
494	001076	004767	015304			JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
495								;THE PS IS NOT EQUAL TO 0
496	001102	000003				3		
497	001104	005237	000000			INC	@#COUNT	;INCREMENT THE COUNTER
498	001110	023701	000010			CMP	@#TEMP3,%1	;IS THE RESULT IN R1 EQUAL TO TEMP3?
499	001114	001403				BEQ	.+10	
500	001116				6\$:			
501	001116	004767	015264			JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
502								;EITHER INCORRECT R1 OR INCORRECT SEQUENCE
503	001122	000004				4		
504	001124	021537	000000			CMP	(R5),@#COUNT	;IS THE TEST NUMBER EQUAL TO THE COUNTER?
505	001130	001372				BNE	6\$;IF NOT GO TO THE HLT ABOVE
506	001132	005215				INC	(R5)	
507	001134	010703				SCOPE3		
508	001136	021527	000037			CMP	(R5),#37	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
509								;BY 14. AND RIGHT BY 14.?
510	001142	002011				BGE	8\$	
511	001144	005237	000006			INC	@#TEMP2	
512	001150	006367	176634			ASL	TEMP3	;SHIFT TEMP3 LEFT
513	001154	021527	000020			CMP	(R5),#20	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
514	001160	001004				BNE	REG2	
515	001162	000167	000622			JMP	NEGAT	;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
516	001166	004767	000644		8\$:	JSR	PC,TST37	;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
517	001172	010701			REG2:	SCOPE1		
518	001174	013702	000004			MOV	@#TEMP1,%2	;LOAD R2 WITH THE CONTENTS OF TEMP1
519	001200	032737	000001	016336		BIT	#1,@#SPASS	;IS IT AN EVEN PASS ?
520	001206	001004				BNE	2\$;IF NOT THEN GO TO 2\$
521	001210	013703	000006			MOV	@#TEMP2,R3	;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
522	001214	072203				ASH	R3,R2	;USING R2
523	001216	000402				BR	4\$	
524	001220	072267	176562		2\$:	ASH	TEMP2,%2	;SHIFT R2 BY THE NUMBER SPECIFIED BY TEMP2
525	001224	106737	000002		4\$:	MFPS	@#PSWORD	;SAVE PS
526	001230	123737	000012	000002		CMPB	@#TEMP4,@#PSWORD	;IS THE PS = TEMP4 ?
527	001236	001403				BEQ	.+10	
528	001240	004767	015142			JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
529								;THE PS IS NOT EQUAL TO 0
530	001244	000005				5		
531	001246	005237	000000			INC	@#COUNT	
532	001252	023702	000010			CMP	@#TEMP3,%2	;IS THE RESULT IN R2 EQUAL TO TEMP3?
533	001256	001403				BEQ	.+10	
534	001260				6\$:			
535	001260	004767	015122			JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
536								;EITHER INCORRECT R2 OR INCORRECT SEQUENCE
537	001264	000006				6		
538	001266	021537	000000			CMP	(R5),@#COUNT	;IS THE TEST NUMBER EQUAL TO THE COUNTER?
539	001272	001372				BNE	6\$;IF NOT GO TO THE HLT ABOVE
540	001274	005215				INC	(R5)	
541	001276	010701				SCOPE1		
542	001300	021527	000037			CMP	(R5),#37	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
543								;LEFT BY 14, AND RIGHT BY 14.?
544	001304	002011				BGE	8\$	
545	001306	005237	000006			INC	@#TEMP2	

546	001312	006367	176472		ASL	TEMP3	:SHIFTED TEMP3 LEFT
547	001316	021527	000020		CMP	(R5), #20	:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
548	001322	001004			BNE	REG3	
549	001324	000167	000460		JMP	NEGAT	:IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
550	001330	004767	000502		JSR	PC, TST37	:IF SO GO AND CONTINUE THE REST OF THE PROGRAM
551	001334	010701		8\$: REG3:	SCOPE1		
552	001336	013703	000004		MOV	@TEMP1, %3	:LOAD R3 WITH THE CONTENTS OF TEMP1
553	001342	032737	000001	016336	BIT	#1, @SPASS	:IS IT AN EVEN PASS ?
554	001350	001004			BNE	2\$:IF NOT THEN GO TO 2\$
555	001352	013704	000006		MOV	@TEMP2, R4	:OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
556	001356	072304			ASH	R4, R3	:USING R3
557	001360	000402			BR	4\$	
558	001362	072367	176420		ASH	TEMP2, %3	:SHIFT R3 BY THE NUMBER SPECIFIED BY TEMP2
559	001366	106737	000002		MFPS	@PSWORD	:SAVE PS
560	001372	123737	000012	000002	CMPB	@TEMP4, @PSWORD	:IS THE PS = TEMP4 ?
561	001400	001403			BEQ	.+10	
562	001402	004767	015000		JSR	PC, \$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
563							:THE PS IS NOT EQUAL TO 0.
564	001406	000007			7		
565	001410	005237	000000		INC	@COUNT	
566	001414	023703	000010		CMP	@TEMP3, %3	:IS THE RESULT IN R3 EQUAL TO TEMP3?
567	001420	001403			BEQ	.+10	
568	001422			6\$:			
569	001422	004767	014760		JSR	PC, \$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
570							:EITHER INCORRECT R3 OR INCORRECT SEQUENCE
571	001426	000010			10		
572	001430	021537	000000		CMP	(R5), @COUNT	:IS THE TEST NUMBER EQUAL TO THE COUNTER?
573	001434	001372			BNE	6\$:IF NOT GO TO THE HLT ABOVE
574	001436	005215			INC	(R5)	
575	001440	010701			SCOPE1		
576	001442	021527	000037		CMP	(R5), #37	:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
577							:LEFT BY 14, AND RIGHT BY 14.?
578	001446	002010			BGE	8\$	
579	001450	005237	000006		INC	@TEMP2	
580	001454	006367	176330		ASL	TEMP3	:SHIFT TEMP3 LEFT?
581	001460	021527	000020		CMP	(R5), #20	:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
582	001464	001003			BNE	REG4	
583	001466	000550			BR	NEGAT	:IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
584	001470	004767	000342		JSR	PC, TST37	:IF SO GO AND CONTINUE THE REST OF THE PROGRAM
585	001474	010703		8\$: REG4:	SCOPE3		
586	001476	013704	000004		MOV	@TEMP1, %4	:LOAD R4 WITH THE CONTENTS OF TEMP1
587	001502	010501			MOV	R5, R1	:SAVE R5
588	001504	032737	000001	016336	BIT	#1, @SPASS	:IS IT AN EVEN PASS ?
589	001512	001004			BNE	2\$:IF NOT THEN GO TO 2\$
590	001514	013705	000006		MOV	@TEMP2, R5	:OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
591	001520	072405			ASH	R5, R4	:USING R4
592	001522	000402			BR	4\$	
593	001524	072467	176256		ASH	TEMP2, %4	:SHIFT R4 BY THE NUMBER SPECIFIED BY TEMP2
594	001530	106737	000002		MFPS	@PSWORD	:SAVE PS
595	001534	123737	000012	000002	CMPB	@TEMP4, @PSWORD	:IS PS = TEMP4 ?
596	001542	001403			BEQ	.+10	
597	001544	004767	014636		JSR	PC, \$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
598							:THE PS IS NOT EQUAL TO 0
599	001550	000011			11		
600	001552	005237	000000		INC	@COUNT	
601	001556	023704	000010		CMP	@TEMP3, %4	:IS THE RESULT IN R4 EQUAL TO TEMP3?

602	001562	001403			BEQ	+.10	
603	001564			6S:			
604	001564	004767	014616		JSR	PC, \$HLT	: SEEN AN ERROR, GO TO TH HALT ROUTINE
605							: EITHER INCORRECT R4 OR INCORRECT SEQUENCE
606	001570	000012			12		
607	001572	010105			MOV	R1, R5	: RESTORE R5
608	001574	021537	000000		CMP	(R5), @#COUNT	: IS THE TEST NUMBER EQUAL TO THE COUNTER?
609	001600	001371			BNE	6S	: IF NOT GO TO THE HLT ABOVE
610	001602	005215			INC	(R5)	
611	001604	010701			SCOPE1		
612	001606	021527	000037		CMP	(R5), #37	: HAS THE CONTENTS OF REGISTERS BEEN
613							: SHIFTED LEFT BY 14. AND RIGHT BY 14.?
614	001612	002010			BGE	8S	
615	001614	005237	000006		INC	@#TEMP2	
616	001620	006367	176164		ASL	TEMP3	: SHIFT TEMP3 LEFT
617	001624	021527	000020		CMP	(R5), #20	: HAS THE CONTENTS OF REGISTER BEEN SHIFTED BY 14.?
618	001630	001003			BNE	REGS	
619	001632	000466			BR	NEGAT	: IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
620	001634	004767	000176	8S:	JSR	PC, TST37	: IF SO GO AND CONTINUE THE REST OF THE PROGRAM
621	001640	010701		REGS:	SCOPE1		
622	001642	010501			MOV	R5, R1	: SAVE R5
623	001644	013705	000004		MOV	@#TEMP1, %5	: LOAD R5 WITH THE CONTENTS OF TEMP1
624	001650	032737	000001	016336	BIT	#1, @#SPASS	: IS IT AN EVEN PASS ?
625	001656	001004			BNE	2S	: IF NOT THEN GO TO 2S
626	001660	013700	000006		MOV	@#TEMP2, R0	: OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
627	001664	072500			ASH	R0, R5	: USING R5
628	001666	000402			BR	4S	
629	001670	072567	176112	2S:	ASH	TEMP2, %5	: SHIFT R5 BY THE NUMBER SPECIFIED BY TEMP2
630	001674	106737	000002	4S:	MFPS	@#PSWORD	: SAVE PS
631	001700	123737	000012	000002	CMPB	@#TEMP4, @#PSWORD	: IS PS = TEMP4 ?
632	001706	001403			BEQ	+.10	
633	001710	004767	014472		JSR	PC, \$HLT	: SEEN AN ERROR, GO TO TH HALT ROUTINE
634							: THE PS IS NOT EQUAL TO 0.
635	001714	000013			13		
636	001716	005237	000000		INC	@#COUNT	
637	001722	023705	000010		CMP	@#TEMP3, %5	: IS THE RESULT IN R5 EQUAL TO TEMP3?
638	001726	001403			BEQ	+.10	
639	001730			6S:			
640	001730	004767	014452		JSR	PC, \$HLT	: SEEN AN ERROR, GO TO TH HALT ROUTINE
641							: EITHER INCORRECT R5 OR INCORRECT SEQUENCE
642	001734	000014			14		
643	001736	021137	000000		CMP	(R1), @#COUNT	: IS THE TEST NUMBER EQUAL TO THE COUNTER?
644	001742	001372			BNE	6S	: IF NOT GO TO THE HLT ABOVE
645	001744	010105			MOV	R1, R5	: RESTORE R5
646	001746	005215			INC	(R5)	
647	001750	010701			SCOPE1		
648	001752	021527	000037		CMP	(R5), #37	: HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
649							: LEFT BY 14. AND RIGHT BY 14.?
650	001756	002010			BGE	9S	: IF SO GO AND CONTINUE THE REST OF THE PROGRAM
651	001760	005237	000006		INC	@#TEMP2	
652	001764	006367	176020		ASL	TEMP3	: SHIFT TEMP3 LEFT
653	001770	021527	000020		CMP	(R5), #20	: HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
654	001774	001405			BEQ	NEGAT	: IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
655	001776	000402			BR	10S	
656	002000	004767	000032	8S:	JSR	PC, TST37	
657	002004	000167	176656	10S:	JMP	START	: GO BACK TO START

658	002010	012737	040000	000004	NEGAT:	MOV	#40000, @TEMP1	:TEMP1=40000
659	002016	012737	177762	000006		MOV	#177762, @TEMP2	:TEMP2=177762
660	002024	012737	000001	000010		MOV	#1, @TEMP3	:TEMP3=1
661	002032	000167	176630			JMP	START	
662	002036	021527	000037		TST37:	CMP	(R5), #37	: IS IT TEST 37?
663	002042	001013				BNE	TST40	: IF NOT THEN TRY TEST 40
664	002044	005037	000004			CLR	@TEMP1	: 0
665	002050	012737	000020	000006		MOV	#16, @TEMP2	: SHIFTED BY 16
666	002056	005037	000010			CLR	@TEMP3	: IS=0
667	002062	012737	000004	000012		MOV	#4, @TEMP4	: AND PS=4
668	002070	000207				RTS	PC	
669	002072	021527	000040		TST40:	CMP	(R5), #40	: IS IT TEST 40?
670	002076	001003				BNE	TST41	: IF NOT THEN TRY TEST 41
671	002100	005037	000006			CLR	@TEMP2	: 0 SHIFTED BY 0=0 AND PS=4
672	002104	000207				RTS	PC	
673	002106	021527	000041		TST41:	CMP	(R5), #41	: IS IT TEST 41?
674	002112	001004				BNE	TST42	: IF NOT THEN TRY TEST 42
675	002114	012737	177760	000006		MOV	#-16, @TEMP2	: 0 SHIFTED BY -16.=0 AND PS=4
676	002122	000207				RTS	PC	
677	002124	021527	000042		TST42:	CMP	(R5), #42	: IS IT TEST 42?
678	002130	001013				BNE	TST43	: IF NOT THEN TRY TEST 43
679	002132	012737	100000	000004		MOV	#100000, @TEMP1	: 100000
680	002140	005237	000006			INC	@TEMP2	: SHIFTED BY -15
681	002144	005337	000010			DEC	@TEMP3	: IS=-1
682	002150	012737	000010	000012		MOV	#10, @TEMP4	: AND PS=10
683	002156	000207				RTS	PC	
684	002160	021527	000043		TST43:	CMP	(R5), #43	: IS IT TEST 43?
685	002164	001012				BNE	TST44	: IF NOT THEN IF NOT THEN TRY TEST 44
686	002166	012737	125252	000004		MOV	#125252, @TEMP1	: 125252
687	002174	012737	177777	000006		MOV	#-1, @TEMP2	: SHIFTED BY -1
688	002202	012737	152525	000010		MOV	#152525, @TEMP3	: IS=152525 AND PS=10
689	002210	000207				RTS	PC	
690	002212	021527	000044		TST44:	CMP	(R5), #44	: IS IT TEST 44?
691	002216	001012				BNE	TST45	: IF NOT THEN TRY TEST 45
692	002220	012737	000001	000006		MOV	#1, @TEMP2	: 125252 SHIFTED BY 1
693	002226	012737	052524	000010		MOV	#52524, @TEMP3	: IS=52524
694	002234	012737	000003	000012		MOV	#3, @TEMP4	: AND PS=3
695	002242	000207				RTS	PC	
696	002244	021527	000045		TST45:	CMP	(R5), #45	: IS IT TEST 45?
697	002250	001012				BNE	TST46	: IF NOT THEN TRY TEST 46
698	002252	012737	177776	000006		MOV	#-2, @TEMP2	: 125252 SHIFTED BY -2
699	002260	012737	165252	000010		MOV	#165252, @TEMP3	: IS=165252
700	002266	012737	000011	000012		MOV	#11, @TEMP4	: AND PS=11
701	002274	000207				RTS	PC	
702	002276	021527	000046		TST46:	CMP	(R5), #46	: IS IT TEST 46?
703	002302	001014				BNE	TST47	: IF NOT THEN TRY TEST 47
704	002304	012737	177777	000004		MOV	#-1, @TEMP1	: -1
705	002312	012737	000020	000006		MOV	#16, @TEMP2	: SHIFTED BY 15.
706	002320	005037	000010			CLR	@TEMP3	: IS=0
707	002324	012737	000007	000012		MOV	#7, @TEMP4	: AND PS=7
708	002332	000207				RTS	PC	
709	002334	021527	000047		TST47:	CMP	(R5), #47	: IS IT TEST 47?
710	002340	001011				BNE	TST50	: IF NOT THEN TRY TEST 50
711	002342	005337	000006			DEC	@TEMP2	: -1 SHIFTED BY 15
712	002346	012737	100000	000010		MOV	#100000, @TEMP3	: IS=100000
713	002354	012737	000011	000012		MOV	#11, @TEMP4	: AND PS=11

E02

DFKACA MACY11 27(732) 09-SEP-76 18:20 PAGE 17
 DFKACA.P11 ASH INSTRUCTION TESTS

714	002362	000207			RTS	PC		
715	002364	021527	000050		TST50: CMP	(R5), #50		: IS IT TEST 50
716	002370	001007			BNE	ENT51		: IF NOT THEN TRY TEST 51
717	002372	012737	137777	000004	MOV	#137777, @TEMP1		: 137777 SHIFTED BY 15. IS=100000
718	002400	012737	000013	000012	MOV	#13, @TEMP4		: AND PS=13
719	002406	000207			RTS	PC		
720	002410	021527	000051		ENT51: CMP	(R5), #51		: IS IT ENTERING TEST 51?
721	002414	001403			BEQ	+10		
722	002416	004767	013764		JSR	PC, \$HLT		: SEEN AN ERROR, GO TO TH HALT ROUTINE
723								: TEST NUMBER GOOFED
724	002422	000015			15			
725								
726	002424	005726			TST	(SP)+		: RESTORE STACK POINTER
727	002426	012704	177771		MOV	#-7, %4		
728	002432	012702	000024		MOV	#S1, %2		
729	002436	012703	000026		MOV	#S2, %3		

730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779

```

:*****
:TEST:51 11/34 ASH 125252 SHIFTED BY #5 = 52500 PS = 3
:*****
    
```

```

TST51: SCOPE1
        MOV      #125252,%1      ;LOAD R1 WITH 125252
        ASH      #5,%1          ;SHIFT R1 BY #5
        MFPS     @#PSWORD        ;SAVE PS
        CMPB    #3,@#PSWORD      ;IS THE PS 3?
        BEQ     .+10
        JSR     PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                   ;THE PS IS NOT EQUAL TO 3
        16
        CMP     #52500,%1        ;IS THE RESULT 52500?
        BEQ     .+10
    1$:  JSR     PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                   ;R1 IS NOT EQUAL TO 52500 OR INCORRECT SEQUENCE
        17
        CMP     (R5),#51         ;IS $TESTN = #51
        BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
        INC     (R5)
    
```

```

:*****
:TEST:52 11/34 ASH 125252 SHIFTED BY #52 = 177525 PS = 10
:*****
    
```

```

TST52: SCOPE1
        MOV      #125252,%0      ;LOAD R0 WITH 125252
        ASH      #52,%0         ;SHIFT R0 BY #52
        MFPS     @#PSWORD        ;SAVE PS
        CMPB    #10,@#PSWORD     ;IS THE PS 10?
        BEQ     .+10
        JSR     PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                   ;THE PS IS NOT EQUAL TO 10
        20
        CMP     #177525,%0       ;IS THE RESULT 177525?
        BEQ     .+10
    1$:  JSR     PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                   ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
        21
        CMP     (R5),#52         ;IS $TESTN = #52
        BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
        INC     (R5)
    
```

```

780
781
782
783
784 002602 010701
785 002604 012700 125252
786 002610 072037 000024
787 002614 106737 000002
788 002620 122737 000010 000002
789 002626 001403
790 002630 004767 013552
791
792 002634 000022
793 002636 022700 177525
794 002642 001403
795 002644
796 002644 004767 013536
797
798 002650 000023
799 002652 021527 000053
800 002656 001372
801 002660 005215
802
803
804
805
806
807
808
809 002662 010701
810 002664 012700 125252
811 002670 072012
812 002672 106737 000002
813 002676 122737 000010 000002
814 002704 001403
815 002706 004767 013474
816
817 002712 000024
818 002714 022700 177525
819 002720 001403
820 002722
821 002722 004767 013460
822
823 002726 000025
824 002730 021527 000054
825 002734 001372
826 002736 005215
827
828
829

```

```

*****
:TEST:53  11/34  ASH      125252 SHIFTED BY 2#51 = 177525  PS = 10
*****
TST53:  SCOPE1
        MOV      #125252,%0      ;LOAD RO WITH 125252
        ASH     2#51,%0         ;SHIFT RO BY 2#51
        MFPS    2#PSWORD        ;SAVE PS
        CMPB   #10,2#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
                                   ;RO 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)

```

```

*****
:TEST:54  11/34  ASH      125252 SHIFTED BY (2) = 177525  PS = 10
*****
TST54:  SCOPE1
        MOV      #125252,%0      ;LOAD RO WITH 125252
        ASH     (2),%0          ;SHIFT RO BY (2)
        MFPS    2#PSWORD        ;SAVE PS
        CMPB   #10,2#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
                                   ;RO 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)

```

```

830
831
832
833
834 002740 010701
835 002742 012700 125252
836 002746 072022
837 002750 106737 000002
838 002754 122737 000010 000002
839 002762 001403
840 002764 004767 013416
841
842 002770 000026
843 002772 022700 177525
844 002776 001403
845 003000
846 003000 004767 013402
847
848 003004 000027
849 003006 021527 000055
850 003012 001372
851 003014 005215
852
853
854
855
856
857
858
859 003016 010701
860 003020 012700 125252
861 003024 072042
862 003026 106737 000002
863 003032 122737 000010 000002
864 003040 001403
865 003042 004767 013340
866
867 003046 000030
868 003050 022700 177525
869 003054 001403
870 003056
871 003056 004767 013324
872
873 003062 000031
874 003064 021527 000056
875 003070 001372
876 003072 005215
877
878
879
    ;*****
    ;TEST:55 11/34 ASH 125252 SHIFTED BY (2)+ = 177525 PS = 10
    ;*****
TST55: SCOPE1
        MOV #125252,%0 ;LOAD RO WITH 125252
        ASH (2)+,%0 ;SHIFT RO BY (2)+
        MFPS @#PSWORD ;SAVE PS
        CMPB #10,@#PSWORD ;IS THE PS 10?
        BEQ .+10
        JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
        ;THE PS IS NOT EQUAL TO 10
        26
        CMP #177525,%0 ;IS THE RESULT 177525?
        BEQ .+10
    IS:
        JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
        ;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
        27
        CMP (R5),#55 ;IS $TESTN = #55
        BNE IS ;IF NOT THEN GO TO HLT ABOVE
        INC (R5)
    
```

```

859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
    ;*****
    ;TEST:56 11/34 ASH 125252 SHIFTED BY -(2) = 177525 PS = 10
    ;*****
TST56: SCOPE1
        MOV #125252,%0 ;LOAD RO WITH 125252
        ASH -(2),%0 ;SHIFT RO BY -(2)
        MFPS @#PSWORD ;SAVE PS
        CMPB #10,@#PSWORD ;IS THE PS 10?
        BEQ .+10
        JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
        ;THE PS IS NOT EQUAL TO 10
        30
        CMP #177525,%0 ;IS THE RESULT 177525?
        BEQ .+10
    IS:
        JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
        ;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
        31
        CMP (R5),#56 ;IS $TESTN = #56
        BNE IS ;IF NOT THEN GO TO HLT ABOVE
        INC (R5)
    
```

```

880
881
882
883
884 003074 010701
885 003076 012700 125252
886 003102 072063 000002
887 003106 106737 000002
888 003112 122737 000011 000002
889 003120 001403
890 003122 004767 013260
891
892 003126 000032
893 003130 022700 177252
894 003134 001403
895 003136
896 003136 004767 013244
897
898 003142 000033
899 003144 021527 000057
900 003150 001372
901 003152 005215
902
903
904
905
906
907
908
909 003154 010701
910 003156 012700 125252
911 003162 072073 000000
912 003166 106737 000002
913 003172 122737 000010 000002
914 003200 001403
915 003202 004767 013200
916
917 003206 000034
918 003210 022700 177525
919 003214 001403
920 003216
921 003216 004767 013164
922
923 003222 000035
924 003224 021527 000060
925 003230 001372
926 003232 005215
927
928
929

```

```

*****
:TEST:57 11/34 ASH 125252 SHIFTED BY 2(3) = 177252 PS = 11
*****
TST57: SCOPE1
MOV #125252,%0 ;LOAD RO WITH 125252
ASH 2(3),%0 ;SHIFT RO BY 2(3)
MFPS @#PSWORD ;SAVE PS
CMPB #11,@#PSWORD ;IS THE PS 11?
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 11
32
CMP #177252,%0 ;IS THE RESULT 177252?
BEQ .+10
1$: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;RO IS NOT EQUAL TO 177252 OR INCORRECT SEQUENCE
33
CMP (R5),#57 ;IS $TESTN = #57
BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
INC (R5)

```

```

*****
:TEST:60 11/34 ASH 125252 SHIFTED BY 2(3) = 177525 PS = 10
*****
TST60: SCOPE1
MOV #125252,%0 ;LOAD RO WITH 125252
ASH 2(3),%0 ;SHIFT RO BY 2(3)
MFPS @#PSWORD ;SAVE PS
CMPB #10,@#PSWORD ;IS THE PS 10?
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;THE PS IS NOT EQUAL TO 10
34
CMP #177525,%0 ;IS THE RESULT 177525?
BEQ .+10
1$: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
35
CMP (R5),#60 ;IS $TESTN = #60
BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
INC (R5)

```

930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979

:TEST:61 11/34 ASH 125252 SHIFTED BY 2(3)+ = 177525 PS = 10

TST61: SCOPE1
MOV #125252,%0 ;LOAD RO WITH 125252
ASH 2(3)+,%0 ;SHIFT RO BY 2(3)+
MFPS 2#PSWORD ;SAVE PS
CMPB #10,2#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
IS: JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
37
CMP (R5),#61 ;IS \$TESTN = #61
BNE IS ;IF NOT THEN GO TO HLT ABOVE
INC (R5)

:TEST:62 11/34 ASH 125252 SHIFTED BY 2-(3) = 177525 PS = 10

TST62: SCOPE1
MOV #125252,%0 ;LOAD RO WITH 125252
ASH 2-(3),%0 ;SHIFT RO BY 2-(3)
MFPS 2#PSWORD ;SAVE PS
CMPB #10,2#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
IS: JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
41
CMP (R5),#62 ;IS \$TESTN = #62
BNE IS ;IF NOT THEN GO TO HLT ABOVE
INC (R5)




```
*****
: ASHC INSTRUCTION TESTS
*****
```

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

980									
981									
982									
983									
984									
985									
986									
987									
988									
989									
990									
991									
992	003370	012737	000062	000000		MOV	#62, @#COUNT		
993	003376	005037	000004			CLR	@#TEMP1	: TEMP1=0	
994	003402	012737	000001	000006		MOV	#1, @#TEMP2	: TEMP2=1	
995	003410	005037	000010			CLR	@#TEMP3	: TEMP3=0	
996	003414	005037	000012			CLR	@#TEMP4	: TEMP4=0	
997	003420	012737	000001	000014		MOV	#1, @#TEMPS	: TEMPS=1	
998	003426	005037	000016			CLR	@#TEMP6	: 0 1 SHIFTED BY 0=0 1, PS=0	
999									
1000	003432	010703			REG01:	SCOPE3			
1001	003434	010502				MOV	R5, R2	: SAVE R5	
1002	003436	013700	000004			MOV	@#TEMP1, %0	: PLACE THE CONTENTS OF TEMP1 IN REGISTER 0	
1003	003442	013701	000006			MOV	@#TEMP2, %0!1	: PLACE THE CONTENTS OF TEMP2 IN REGISTER 1	
1004	003446	000241				CLC			
1005	003450	032737	000001	016336		BIT	#1, @#SPASS	: IS IT AN EVEN PASS ?	
1006	003456	001004				BNE	2\$: IF NOT THEN GO TO 2\$	
1007	003460	013705	000010			MOV	@#TEMP3, R5	: OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0	
1008	003464	073005				ASHC	R5, R0	: USING R0	
1009	003466	000402				BR	4\$		
1010	003470	073067	174314		2\$:	ASHC	TEMP3, %0	: ASHC REGISTER 0 BY THE CONTENTS OF TEMP3	
1011	003474	106737	000002		4\$:	MFPS	@#PSWORD	: SAVE PS	
1012	003500	123737	000016	000002		CMPB	@#TEMP6, @#PSWORD	: COMPARE PS WITH THE CONTENTS OF TEMP6	
1013	003506	001403				BEQ	.+10		
1014	003510	004767	012672			JSR	PC, \$HLT	: SEEN AN ERROR, GO TO TH HALT ROUTINE	
1015								: WRONG PS	
1016	003514	000042				42			
1017	003516	005237	000000			INC	@#COUNT		
1018	003522	023700	000012			CMP	@#TEMP4, %0	: IS THE RESULT IN R0 SAME AS TEMP4?	
1019	003526	001403				BEQ	.+10		
1020	003530	004767	012652			JSR	PC, \$HLT	: SEEN AN ERROR, GO TO TH HALT ROUTINE	
1021								: WRONG RESULT IN R0	
1022	003534	000043				43			
1023	003536	023701	000014			CMP	@#TEMPS, %1	: IS THE RESULT IN R1 SAME AS TEMPS?	
1024	003542	001403				BEQ	.+10	: TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMPS	
1025								: AND PS=TEMP6	
1026	003544	004767	012636			JSR	PC, \$HLT	: SEEN AN ERROR, GO TO TH HALT ROUTINE	
1027								: WRONG RESULT IN R1	
1028	003550	000044				44			
1029	003552	010205				MOV	R2, R5	: RESTORE R5	
1030	003554	021537	000000			CMP	(R5), @#COUNT	: IS TEST NUMBER=COUNTER?	
1031	003560	001403				BEQ	.+10		
1032	003562	004767	012620			JSR	PC, \$HLT	: SEEN AN ERROR, GO TO TH HALT ROUTINE	
1033								: NO	
1034	003566	000045				45			
1035	003570	005215				INC	(R5)		

1036	003572	021527	000160		CMP	(R5), #160	;HAVE THE FIRST 159 TEST BEEN EXECUTED?
1037	003576	002014			BGE	6\$;YES
1038	003600	005237	000010		INC	@#TEMP3	
1039	003604	000241			CLC		
1040	003606	006137	000014		ROL	@#TEMP5	;ROTATE TEMPS LEFT BY 1 PLACE
1041	003612	006137	000012		ROL	@#TEMP4	;INTRODUCE CARRY FROM TEMP4 IN TEMPS
1042	003616	021527	000121		CMP	(R5), #121	;IS IT TEST 121?
1043	003622	001004			BNE	REG23	
1044	003624	004467	000410		JSR	R4, RITSH	;IF SO THEN GO AND INITIATE RIGHT SHIFT
1045	003630	004767	000440		JSR	%7, TST160	
1046	003634	010701		6\$: REG23:	SCOPE1		
1047	003636	013702	000004		MOV	@#TEMP1, %2	;PLACE THE CONTENTS OF TEMP1 IN REGISTER 2
1048	003642	013703	000006		MOV	@#TEMP2, %2!1	;PLACE THE CONTENTS OF TEMP2 IN REGISTER 3
1049	003646	000241			CLC		
1050	003650	032737	000001	016336	BIT	#1, @#\$PASS	;IS IT AN EVEN PASS ?
1051	003656	001004			BNE	2\$;IF NOT THEN GO TO 2\$
1052	003660	013704	000010		MOV	@#TEMP3, R4	;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
1053	003664	073204			ASHC	R4, R2	;USING R2
1054	003666	000402			BR	4\$	
1055	003670	073267	174114	2\$:	ASHC	TEMP3, %2	;ASHC REGISTER 2 BY THE CONTENTS OF TEMP3
1056	003674	106737	000002	4\$:	MFPS	@#PSWORD	;SAVE PS
1057	003700	123737	000016	000002	CMPB	@#TEMP6, @#PSWORD	;COMPARE PS WITH THE CONTENTS OF TEMP6
1058	003706	001403			BEQ	+10	
1059	003710	004767	012472		JSR	PC, \$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
1060							;WRONG PS
1061	003714	000046			46		
1062	003716	005237	000000		INC	@#COUNT	
1063	003722	023702	000012		CMP	@#TEMP4, %2	;IS THE RESULT IN R2 SAME AS TEMP4?
1064	003726	001403			BEQ	+10	
1065	003730	004767	012452		JSR	PC, \$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
1066							;WRONG RESULT IN R2
1067	003734	000047			47		
1068	003736	023703	000014		CMP	@#TEMP5, %3	;IS THE RESULT IN R3 SAME AS TEMPS?
1069	003742	001403			BEQ	+10	;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMPS
1070							;AND PS=TEMP6
1071	003744	004767	012436		JSR	PC, \$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
1072							;WRONG RESULT IN R1
1073	003750	000050			50		
1074	003752	021537	000000		CMP	(R5), @#COUNT	;IS TEST NUMBER=COUNTER?
1075	003756	001403			BEQ	+10	
1076	003760	004767	012422		JSR	PC, \$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
1077							;NO
1078	003764	000051			51		
1079	003766	005215			INC	(R5)	
1080	003770	021527	000160		CMP	(R5), #160	;HAVE THE FIRST 159 TEST BEEN EXECUTED?
1081	003774	002014			BGE	6\$;YES
1082	003776	005237	000010		INC	@#TEMP3	
1083	004002	000241			CLC		
1084	004004	006137	000014		ROL	@#TEMP5	;ROTATE TEMPS LEFT BY 1 PLACE
1085	004010	006137	000012		ROL	@#TEMP4	;INTRODUCE CARRY FROM TEMPS IN TEMP4
1086	004014	021527	000121		CMP	(R5), #121	;IS IT TEST 121?
1087	004020	001004			BNE	REG45	
1088	004022	004467	000212		JSR	R4, RITSH	;IF SO THEN GO AND INITIATE RIGHT SHIFT
1089	004026	004767	000242	6\$: REG45:	JSR	%7, TST160	
1090	004032	010701			SCOPE1		
1091	004034	010501			MOV	R5, R1	;SAVE R5

1092	004036	013704	000004		MOV	Q#TEMP1,%4	;PLACE THE CONTENTS OF TEMP1 IN REGISTER 4
1093	004042	013705	000006		MOV	Q#TEMP2,%4!1	;PLACE THE CONTENTS OF TEMP2 IN REGISTER 5
1094	004046	000241			CLC		
1095	004050	032737	000001	016336	BIT	#1,Q#SPASS	;IS IT AN EVEN PASS ?
1096	004056	001004			BNE	2\$;IF NOT THEN GO TO 2\$
1097	004060	013700	000010		MOV	Q#TEMP3,R0	;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
1098	004064	073400			ASHC	R0,R4	;USING R4
1099	004066	000402			BR	4\$	
1100	004070	073467	173714		ASHC	TEMP3,%4	;ASHC REGISTER 4 BY THE CONTENTS OF TEMP3
1101	004074	106737	000002		MFPS	Q#PSWORD	;SAVE PS
1102	004100	123737	000016	000002	CMPB	Q#TEMP6,Q#PSWORD	;COMPARE PS WITH THE CONTENTS OF TEMP6
1103	004106	001403			BEQ	.+10	
1104	004110	004767	012272		JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
1105							;WRONG PS
1106	004114	000052			52		
1107	004116	005237	000000		INC	Q#COUNT	
1108	004122	023704	000012		CMP	Q#TEMP4,%4	;IS THE RESULT IN R4 SAME AS TEMP4?
1109	004126	001403			BEQ	.+10	
1110	004130	004767	012252		JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
1111							;WRONG RESULT IN R4
1112	004134	000053			53		
1113	004136	023705	000014		CMP	Q#TEMP5,%5	;IS THE RESULT IN R5 SAME AS TEMP5?
1114	004142	001403			BEQ	.+10	;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMPS
1115							;AND PS=TEMP6
1116	004144	004767	012236		JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
1117							;WRONG RESULT IN R5
1118	004150	000054			54		
1119	004152	021137	000000		CMP	(R1),Q#COUNT	;IS TEST NUMBER=COUNTER?
1120	004156	001403			BEQ	.+10	
1121	004160	004767	012222		JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
1122							;NO
1123	004164	000055			55		
1124	004166	010105			MOV	R1,R5	;RESTORE R5
1125	004170	005215			INC	(R5)	
1126	004172	021527	000160		CMP	(R5),#160	;HAVE THE FIRST 159 TEST BEEN EXECUTED?
1127	004176	002014			BGE	6\$;YES
1128	004200	005237	000010		INC	Q#TEMP3	

1129	004204	000241			CLC		
1130	004206	006137	000014		ROL	@#TEMP5	; ROTATE TEMP5 LEFT BY 1 PLACE
1131	004212	006137	000012		ROL	@#TEMP4	; INTRODUCE CARRY FROM TEMP5 IN TEMP4
1132	004216	021527	000121		CMP	(R5), #121	; IS IT TEST 121?
1133	004222	001004			BNE	8\$	
1134	004224	004467	000010		JSR	R4, RITSH	; IF SO THEN GO AND INITIATE RIGHT SHIFT
1135	004230	004767	000040		JSR	%7, TST160	
1136	004234	000167	177172		JMP	REG01	
1137	004240	022424			RITSH: CMP	(R4)+, (R4)+	; MAKE R4 POINT TO THE NEXT REG TAG
1138	004242	012737	040000	000004	MOV	#40000, @#TEMP1	; TEMP1=4000
1139	004250	005037	000006		CLR	@#TEMP2	; TEMP2=0
1140	004254	012737	177742	000010	MOV	#-30, @#TEMP3	; TEMP3=-30
1141	004262	005037	000012		CLR	@#TEMP4	; TEMP4=0
1142	004266	005237	000014		INC	@#TEMP5	; TEMP5=1
1143	004272	000204			RTS	R4	
1144	004274	021527	000160		TST160: CMP	(R5), #160	; IS IT TEST 160
1145	004300	001010			BNE	TST161	; IF NOT THEN TRY TEST 161
1146	004302	005037	000004		CLR	@#TEMP1	; 0 0 SHIFTED BY 0
1147	004306	005037	000012		CLR	@#TEMP4	; IS EQUAL TO 0 0
1148	004312	012737	000004	000016	MOV	#4, @#TEMP6	; AND PS=4
1149	004320	000207			RTS	%7	
1150	004322	021527	000161		TST161: CMP	(R5), #161	; IS IT TEST 161
1151	004326	001004			BNE	TST162	
1152	004330	012737	177746	000010	MOV	#-32, @#TEMP3	; 0 0 SHIFTED BY -32=0 0, PS=4
1153	004336	000207			RTS	%7	
1154	004340	021527	000162		TST162: CMP	(R5), #162	; IS IT TEST 162
1155	004344	001004			BNE	TST163	; IF NOT THEN TRY TEST 163
1156	004346	012737	000032	000010	MOV	#32, @#TEMP3	; 0 0 SHIFTED BY 32=0 0, PS=4
1157	004354	000207			RTS	%7	
1158	004356	021527	000163		TST163: CMP	(R5), #163	; IS IT TEST 163?
1159	004362	001016			BNE	TST164	; IF NOT THEN TRY TEST 164
1160	004364	012737	052525	000004	MOV	#52525, @#TEMP1	; 52525 0
1161	004372	012737	177760	000010	MOV	#-16, @#TEMP3	; SHIFTED BY -16.
1162	004400	005037	000012		CLR	@#TEMP4	
1163	004404	012737	052525	000014	MOV	#52525, @#TEMP5	; IS EQUAL TO 0 52525
1164	004412	005037	000016		CLR	@#TEMP6	; AND PS = 0
1165	004416	000207			RTS	%7	
1166	004420	021527	000164		TST164: CMP	(R5), #164	; IS IT TEST 164?
1167	004424	001014			BNE	TST165	; IF NOT THEN TRY TEST 165
1168	004426	012737	125252	000004	MOV	#125252, @#TEMP1	; 125252 0 SHIFTED BY -16.
1169	004434	005337	000012		DEC	@#TEMP4	
1170	004440	012737	125252	000014	MOV	#125252, @#TEMP5	; IS EQUAL TO -1 125252
1171	004446	012737	000010	000016	MOV	#10, @#TEMP6	; AND PS=10
1172	004454	000207			RTS	%7	
1173	004456	021527	000165		TST165: CMP	(R5), #165	; IS IT TEST 165?
1174	004462	001007			BNE	TST166	; IF NOT THEN TRY TEST 166
1175	004464	012737	177777	000004	MOV	#-1, @#TEMP1	; -1 0 SHIFTED BY -16
1176	004472	012737	177777	000014	MOV	#-1, @#TEMP5	; IS EQUAL TO -1 -1, AND PS=10
1177	004500	000207			RTS	%7	
1178	004502	021527	000166		TST166: CMP	(R5), #166	; IS IT TEST 166?
1179	004506	001011			BNE	TST167	; IF NOT THEN TRY TEST 167
1180	004510	012737	100000	000004	MOV	#100000, @#TEMP1	; 100000 0
1181	004516	012737	177740	000010	MOV	#-32, @#TEMP3	; SHIFTED BY -32 IS EQUAL TO -1 -1
1182	004524	005237	000016		INC	@#TEMP6	; AND PS=11
1183	004530	000207			RTS	%7	
1184	004532	021527	000167		TST167: CMP	(R5), #167	; IS IT TEST 167?

1185	004536	001014			BNE	TST170	; IF NOT THEN TRY TEST 170
1186	004540	005037	000004		CLR	Q#TEMP1	
1187	004544	005337	000006		DEC	Q#TEMP2	: 0 -1
1189	004550	012737	000020	000010	MOV	#16, Q#TEMP3	: SHIFTED BY 16.
1189	004556	005037	000014		CLR	Q#TEMP5	: IS EQUAL TO -1 0
1190	004562	005237	000016		INC	Q#TEMP6	: AND PS=12
1191	004566	000207			RTS	%7	
1192	004570	021527	000170		TST170: CMP	(R5), #170	: IS IT TEST 170?
1193	004574	001007			BNE	TST171	: IF NOT THEN TRY TEST 171
1194	004576	012737	125252	000006	MOV	#125252, Q#TEMP2	: 0 125252 SHIFTED BY 16
1195	004604	012737	125252	000012	MOV	#125252, Q#TEMP4	: IS EQUAL TO 125252 0, AND PS=12
1196	004612	000207			RTS	%7	
1197	004614	021527	000171		TST171: CMP	(R5), #171	: IS IT TEST 171?
1198	004620	001010			BNE	TST172	: IF NOT THEN TRY TEST 172
1199	004622	005337	000010		DEC	Q#TEMP3	: 0 125252 SHIFTED BY 15
1200	004626	012737	052525	000012	MOV	#52525, Q#TEMP4	: IS EQUAL TO 52525 0
1201	004634	005037	000016		CLR	Q#TEMP6	: AND PS=0
1202	004640	000207			RTS	%7	
1203	004642	021527	000172		TST172: CMP	(R5), #172	: IS IT TEST 172?
1204	004646	001006			BNE	TST173	: IF NOT THEN TRY TEST 173
1205	004650	012737	052525	000006	MOV	#52525, Q#TEMP2	: 0 52525
1206	004656	005237	000010		INC	Q#TEMP3	: SHIFTED BY 16. IS EQUAL TO 52525 0, AND PS=0
1207	004662	000207			RTS	%7	
1208	004664	021527	000173		TST173: CMP	(R5), #173	: IS IT TEST 173?
1209	004670	001014			BNE	TST174	: IF NOT THEN TRY TEST 174
1210	004672	012737	177777	000006	MOV	#-1, Q#TEMP2	: 0 -1
1211	004700	005337	000010		DEC	Q#TEMP3	: SHIFTED BY 15.
1212	004704	012737	077777	000012	MOV	#77777, Q#TEMP4	
1213	004712	012737	100000	000014	MOV	#100000, Q#TEMP5	: IS EQUAL TO 77777 100000, AND PS=0
1214	004720	000207			RTS	%7	
1215	004722	021527	000174		TST174: CMP	(R5), #174	: IS IT TEST 174?
1216	004726	001013			BNE	TST175	: IF NOT THEN TRY TEST 175
1217	004730	012737	100000	000004	MOV	#100000, Q#TEMP1	
1218	004736	005337	000006		DEC	Q#TEMP2	: 100000 -2 SHIFTED BY 15.
1219	004742	005037	000014		CLR	Q#TEMP5	: IS EQUAL TO 77777 0
1220	004746	012737	000002	000016	MOV	#2, Q#TEMP6	: AND PS=2
1221	004754	000207			RTS	%7	
1222	004756	021527	000175		TST175: CMP	(R5), #175	: IS IT TEST 175?
1223	004762	001015			BNE	ENT176	: IF NOT THEN TRY TEST 176
1224	004764	012737	177777	000004	MOV	#-1, Q#TEMP1	
1225	004772	005037	000006		CLR	Q#TEMP2	: -1 0
1226	004776	005237	000010		INC	Q#TEMP3	: SHIFTED BY 16.
1227	005002	005037	000012		CLR	Q#TEMP4	: IS EQUAL TO 0 0
1228	005006	012737	000007	000016	MOV	#7, Q#TEMP6	: AND PS=7
1229	005014	000207			RTS	%7	
1230	005016	021527	000176		ENT176: CMP	(R5), #176	: IS THE PROGRAM ENTERING TEST 176?
1231	005022	001403			BEQ	+10	
1232	005024	004767	011356		JSR	PC, \$HLT	: SEEN AN ERROR, GO TO TH HALT ROUTINE
1233							: TEST NUMBER GOOFED
1234	005030	000056			56		
1235							
1236	005032	005726			TST	(SP)+	: RESTORE STACK POINTER
1237							

```

1238
1239
1240
1241
1242 005034 010701
1243 005036 012701 000000
1244 005042 012701 000001
1245 005046 000241
1246 005050 073127 000010
1247 005054 106737 000002
1248 005060 122737 000000 000002
1249 005066 001403
1250 005070 004767 011312
1251
1252 005074 000057
1253 005076 022701 000400
1254 005102 001403
1255 005104 004767 011276
1256
1257 005110 000060
1258 005112 021527 000176
1259 005116 001403
1260 005120 004767 011262
1261
1262 005124 000061
1263 005126 005215
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
    
```

```

:*****
:TEST:176      1 SHIFTED BY 8. = 400 PS = 0
:*****
    
```

```

TST176: SCOPE1
MOV      #DUMMY,%1      ;LOAD R1 WITH DUMMY
MOV      #1,%1!1        ;LOAD R1!1 WITH 1
CLC
ASHC     #8,%1          ;SHIFT R1,R1!1 BY 8.
MFPS     @#PSWORD       ;SAVE PS
CMPB     #0,@#PSWORD    ;IS THE PS 0?
BEQ      +10
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;THE PS IS NOT EQUAL TO 0
57
CMP      #400,%1        ;IS THE RESULT 400?
BEQ      +10
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;R1 IS NOT EQUAL TO 400
60
CMP      (R5),#176      ;IS $TESTN = #176?
BEQ      +10            ;IF NOT THEN GO TO HLT
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;TEST IS IN WRONG SEQUENCE
61
INC      (R5)
    
```

```

:*****
:TEST:177      -1 SHIFTED BY 15. = 100000 PS = 11
:*****
    
```

```

TST177: SCOPE1
MOV      #DUMMY,%3      ;LOAD R3 WITH DUMMY
MOV      #-1,%3!1       ;LOAD R3!1 WITH -1
CLC
ASHC     #15,%3         ;SHIFT R3,R3!1 BY 15.
MFPS     @#PSWORD       ;SAVE PS
CMPB     #11,@#PSWORD   ;IS THE PS 11?
BEQ      +10
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;THE PS IS NOT EQUAL TO 11
62
CMP      #100000,%3     ;IS THE RESULT 100000?
BEQ      +10
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;R3 IS NOT EQUAL TO 100000
63
CMP      (R5),#177      ;IS $TESTN = #177?
BEQ      +10            ;IF NOT THEN GO TO HLT
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;TEST IS IN WRONG SEQUENCE
64
INC      (R5)
    
```

```

1294
1295
1296
1297
1298 005224 010701
1299 005226 010501
1300 005230 012705 000000
1301 005234 012705 052525
1302 005240 000241
1303 005242 073527 000000
1304 005246 106737 000002
1305 005252 122737 000000 000002
1306 005260 001403
1307 005262 004767 011120
1308
1309 005266 000065
1310 005270 022705 052525
1311 005274 001403
1312 005276 004767 011104
1313
1314 005302 000066
1315 005304 010105
1316 005306 021527 000200
1317 005312 001403
1318 005314 004767 011066
1319
1320 005320 000067
1321 005322 005215
1322
1323
1324
1325
1326
1327
1328
1329 005324 010701
1330 005326 012701 000000
1331 005332 012701 020010
1332 005336 000241
1333 005340 073127 177763
1334 005344 106737 000002
1335 005350 122737 000000 000002
1336 005356 001403
1337 005360 004767 011022
1338
1339 005364 000070
1340 005366 022701 000101
1341 005372 001403
1342 005374 004767 011006
1343
1344 005400 000071
1345 005402 021527 000201
1346 005406 001403
1347 005410 004767 010772
1348
1349 005414 000072
1350 005416 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
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)
    
```

```

:*****
: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
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
        72
INC     (R5)
    
```

E03

DFKACA MACY11 27(732) 09-SEP-76 18:20 PAGE 30
DFKACA.P11 ASHC INSTRUCTION TESTS

1350
1351


```

1352
1353
1354
1355
1356 005420 010701
1357 005422 012703 000000
1358 005426 012703 177777
1359 005432 000241
1360 005434 073327 000020
1361 005440 106737 000002
1362 005444 122737 000011 000002
1363 005452 001403
1364 005454 004767 010726
1365
1366 005460 000073
1367 005462 022703 000000
1368 005466 001403
1369 005470 004767 010712
1370
1371 005474 000074
1372 005476 021527 000202
1373 005502 001403
1374 005504 004767 010676
1375
1376 005510 000075
1377 005512 005215
1378
1379
1380
1381
1382
1383
1384 005514 010701
1385 005516 010501
1386 005520 012705 000000
1387 005524 012705 000001
1388 005530 000241
1389 005532 073527 177777
1390 005536 106737 000002
1391 005542 122737 000001 000002
1392 005550 001403
1393 005552 004767 010630
1394
1395 005556 000076
1396 005560 022705 100000
1397 005564 001403
1398 005566 004767 010614
1399
1400 005572 000077
1401 005574 010105
1402 005576 021527 000203
1403 005602 001403
1404 005604 004767 010576
1405
1406 005610 000100
1407 005612 005215
    
```

```

:*****
:TEST:202      -! 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
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)
    
```

```

:*****
: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
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
100
INC      (R5)
    
```

G03

DFKACA MACY11 27(732) 08-SEP-76 18:20 PAGE 32
DFKACA.P11 ASHC INSTRUCTION TESTS

1408
1409

```

1410
1411
1412
1413
1414 005614 010701
1415 005616 012701 000000
1416 005622 012701 125252
1417 005626 000241
1418 005630 073127 177760
1419 005634 106737 000002
1420 005640 122737 000011 000002
1421 005646 001403
1422 005650 004767 010532
1423
1424 005654 000101
1425 005656 022701 125252
1426 005662 001403
1427 005664 004767 010516
1428
1429 005670 000102
1430 005672 021527 000204
1431 005676 001403
1432 005700 004767 010502
1433
1434 005704 000103
1435 005706 005215
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
    
```

```

*****
:TEST:204      125252 SHIFTED BY -16. = 125252 PS = 11
*****
    
```

```

TST204: SCOPE1
MOV      #DUMMY,%1      ;LOAD R1 WITH DUMMY
MOV      #125252,%1!1   ;LOAD R1!1 WITH 125252
CLC
ASHC     #-16,%1        ;SHIFT R1,R1!1 BY -16.
MFPS     @#PSWORD       ;SAVE PS
CMPB     #11,@#PSWORD   ;IS THE PS 11?
BEQ      +10
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;THE PS IS NOT EQUAL TO 11
101
CMP      #125252,%1     ;IS THE RESULT 125252?
BEQ      +10
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;R1 IS NOT EQUAL TO 125252
102
CMP      (R5),#204      ;IS $TESTN = #204?
BEQ      +10            ;IF NOT THEN GO TO HLT
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;TEST IS IN WRONG SEQUENCE
103
INC      (R5)
    
```

```

*****
:TEST:205      125252 125252 SHIFTED BY 21. = 52500 000000 PS = 3
*****
    
```

```

TST205: SCOPE1
MOV      #125252,%2     ;LOAD R2 WITH 125252
MOV      #125252,%2!1  ;LOAD R2!1 WITH 125252
CLC
ASHC     #21,%2         ;SHIFT R2,R2!1 BY 21.
MFPS     @#PSWORD       ;SAVE PS
CMPB     #3,@#PSWORD   ;IS THE PS 3?
BEQ      +10
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;THE PS IS NOT EQUAL TO 3
104
CMP      #52500,%2      ;IS THE RESULT 52500?
BEQ      +10
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;R2 IS NOT EQUAL TO 52500
105
CMP      #000000,%2!1  ;IS THE RESULT 000000?
BEQ      +10
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;R2!1 IS NOT EQUAL TO 000000
106
CMP      (R5),#205     ;IS $TESTN = #205?
BEQ      +10            ;IF NOT THEN GO TO HLT
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
    
```

DFKACA MACY11 27(732) 08-SEP-76 18:20 PAGE 34
DFKACA.P11 ASHC INSTRUCTION TESTS

;TEST IS IN WRONG SEQUENCE

1466					
1467	006014	000107		107	
1468	006016	005215		INC	(R5)
1469					
1470					
1471					
1472	006020	012702	177771	MOV	#-7,%2
1473	006024	012703	000024	MOV	#51,%3
1474	006030	012704	000026	MOV	#52,%4
1475					

```

1476
1477
1478
1479
1480 006034 010701
1481 006036 012700 125252
1482 006042 012701 125252
1483 006046 000241
1484 006050 073067 171750
1485 006054 106737 000002
1486 006060 122737 000010 000002
1487 006066 001403
1488 006070 004767 010312
1489
1490 006074 000110
1491 006076 022700 177525
1492 006102 001403
1493 006104 004767 010276
1494
1495 006110 000111
1496 006112 022701 052525
1497 006116 001403
1498 006120
1499 006120 004767 010262
1500
1501 006124 000112
1502 006126 021527 000206
1503 006132 001372
1504 006134 005215
1505
1506
1507
1508
1509
1510
1511 006136 010701
1512 006140 012700 125252
1513 006144 012701 125252
1514 006150 000241
1515 006152 073077 171650
1516 006156 106737 000002
1517 006162 122737 000010 000002
1518 006170 001403
1519 006172 004767 010210
1520
1521 006176 000113
1522 006200 022700 177525
1523 006204 001403
1524 006206 004767 010174
1525
1526 006212 000114
1527 006214 022701 052525
1528 006220 001403
1529 006222
1530 006222 004767 010160
1531

```

```

;*****
;TEST:206      125252 125252 SHIFTED BY S1 = 177525 52525 PS = 10
;*****
TST206: SCOPE1
      MOV      #125252,%0          ;LOAD RO WITH 125252
      MOV      #125252,%0!1      ;LOAD RO!1 WITH 125252
      CLC
      ASHC     S1,%0             ;SHIFT RO,RO!1 BY S1
      MFPS     @#PSWORD         ;SAVE PS
      CMPB    #10,@#PSWORD      ;IS THE PS 10?
      BEQ     .+10
      JSR     PC,$HLT           ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                   ;THE PS IS NOT EQUAL TO 10
      CMP     #177525,%0        ;IS THE RESULT 177525?
      BEQ     .+10
      JSR     PC,$HLT           ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                   ;RO IS NOT EQUAL TO 177525
      CMP     #52525,%0!1      ;IS THE RESULT 52525?
      BEQ     .+10
1$:   JSR     PC,$HLT           ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                   ;RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
      CMP     (R5),#206         ;IS THE $TESTN = #206?
      BNE    1$                ;IF NOT THEN GO TO HLT ABOVE
      INC     (R5)
;*****
;TEST:207      125252 125252 SHIFTED BY @S2 = 177525 52525 PS = 10
;*****
TST207: SCOPE1
      MOV      #125252,%0          ;LOAD RO WITH 125252
      MOV      #125252,%0!1      ;LOAD RO!1 WITH 125252
      CLC
      ASHC     @S2,%0           ;SHIFT RO,RO!1 BY @S2
      MFPS     @#PSWORD         ;SAVE PS
      CMPB    #10,@#PSWORD      ;IS THE PS 10?
      BEQ     .+10
      JSR     PC,$HLT           ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                   ;THE PS IS NOT EQUAL TO 10
      CMP     #177525,%0        ;IS THE RESULT 177525?
      BEQ     .+10
      JSR     PC,$HLT           ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                   ;RO IS NOT EQUAL TO 177525
      CMP     #52525,%0!1      ;IS THE RESULT 52525?
      BEQ     .+10
1$:   JSR     PC,$HLT           ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                   ;RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE

```

K03

DFKACA MACY11 27(732) 09-SEP-76 18:20 PAGE 36
DFKACA.P11 ASHC INSTRUCTION TESTS

1532 006226 000115
1533 006230 021527 000207
1534 006234 001372
1535 006236 005215
1536
1537

115
CMP (R5), #207
BNE 1\$
INC (R5)

; IS THE \$TESTN = #207?
; IF NOT THEN GO TO HLT ABOVE

```

1538
1539
1540
1541
1542 006240 010701
1543 006242 012700 125252
1544 006246 012701 125252
1545 006252 000241
1546 006254 073037 000024
1547 006260 106737 000002
1548 006264 122737 000010 000002
1549 006272 001403
1550 006274 004767 010106
1551
1552 006300 000116
1553 006302 022700 177525
1554 006306 001403
1555 006310 004767 010072
1556
1557 006314 000117
1558 006316 022701 052525
1559 006322 001403
1560 006324
1561 006324 004767 010056
1562
1563 006330 000120
1564 006332 021527 000210
1565 006336 001372
1566 006340 005215
1567
1568
1569
1570
1571
1572
1573 006342 010701
1574 006344 012700 125252
1575 006350 012701 125252
1576 006354 000241
1577 006356 073013
1578 006360 106737 000002
1579 006364 122737 000010 000002
1580 006372 001403
1581 006374 004767 010006
1582
1583 006400 000121
1584 006402 022700 177525
1585 006406 001403
1586 006410 004767 007772
1587
1588 006414 000122
1589 006416 022701 052525
1590 006422 001403
1591 006424
1592 006424 004767 007756
1593
    
```

```

:*****
:TEST:210      125252 125252 SHIFTED BY @#S1 = 177525 52525 PS = 10
:*****
    
```

```

TST210: SCOPE1
MOV      #125252,%0          ;LOAD RO WITH 125252
MOV      #125252,%0!1       ;LOAD RO!1 WITH 125252
CLC
ASHC    @#S1,%0             ;SHIFT RO,RO!1 BY @#S1
MFPS    @#PSWORD           ;SAVE PS
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
                                ;RO 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
                                ;RO!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)
    
```

```

:*****
:TEST:211      125252 125252 SHIFTED BY (3) = 177525 52525 PS = 10
:*****
    
```

```

TST211: SCOPE1
MOV      #125252,%0          ;LOAD RO WITH 125252
MOV      #125252,%0!1       ;LOAD RO!1 WITH 125252
CLC
ASHC    (3),%0             ;SHIFT RO,RO!1 BY (3)
MFPS    @#PSWORD           ;SAVE PS
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
                                ;RO IS NOT EQUAL TO 177525
                                122
CMP     #52525,%0!1       ;IS THE RESULT 52525?
BEQ     .+10
1$:     JSR     PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                ;RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
    
```

1594 006430 000123
1595 006432 021527 000211
1596 006436 001372
1597 006440 005215
1598
1599

123
CMP (R5), #211
BNE 1\$
INC (R5)

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


```

1600
1601
1602
1603
1604 006442 010701
1605 006444 012700 125252
1606 006450 012701 125252
1607 006454 000241
1608 006456 073023
1609 006460 106737 000002
1610 006464 122737 000010 000002
1611 006472 001403
1612 006474 004767 007706
1613
1614 006500 000124
1615 006502 022700 177525
1616 006506 001403
1617 006510 004767 007672
1618
1619 006514 000125
1620 006516 022701 052525
1621 006522 001403
1622 006524
1623 006524 004767 007656
1624
1625 006530 000126
1626 006532 021527 000212
1627 006536 001372
1628 006540 005215
1629
1630
1631
1632
1633
1634
1635 006542 010701
1636 006544 012700 125252
1637 006550 012701 125252
1638 006554 000241
1639 006556 073043
1640 006560 106737 000002
1641 006564 122737 000010 000002
1642 006572 001403
1643 006574 004767 007606
1644
1645 006600 000127
1646 006602 022700 177525
1647 006606 001403
1648 006610 004767 007572
1649
1650 006614 000130
1651 006616 022701 052525
1652 006622 001403
1653 006624
1654 006624 004767 007556
1655

```

```

:*****
:TEST:212      125252 125252 SHIFTED BY (3)+ = 177525 52525 PS = 10
:*****

```

```

TST212: SCOPE1
MOV      #125252,%0          ;LOAD RO WITH 125252
MOV      #125252,%0!1      ;LOAD RO!1 WITH 125252
CLC
ASHC    (3)+,%0            ;SHIFT RO,RO!1 BY (3)+
MFPS    @#PSWORD          ;SAVE PS
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
                           ;RO 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
                           ;RO!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)

```

```

:*****
:TEST:213      125252 125252 SHIFTED BY -(3) = 177525 52525 PS = 10
:*****

```

```

TST213: SCOPE1
MOV      #125252,%0          ;LOAD RO WITH 125252
MOV      #125252,%0!1      ;LOAD RO!1 WITH 125252
CLC
ASHC    -(3),%0           ;SHIFT RO,RO!1 BY -(3)
MFPS    @#PSWORD          ;SAVE PS
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
                           ;RO IS NOT EQUAL TO 177525
130
CMP     #52525,%0!1      ;IS THE RESULT 52525?
BEQ     .+10
1$:     JSR     PC,$HLT           ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                           ;RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE

```

1656	006630	000131	
1657	006632	021527	000213
1658	006636	001372	
1659	006640	005215	
1660			
1661			

131
CMP
BNE
INC

(R5), #213
IS
(R5) ; IS THE STESTN = #213?
; IF NOT THEN GO TO HLT ABOVE

```

1662
1663
1664
1665
1666 006642 010701
1667 006644 012700 125252
1668 006650 012701 125252
1669 006654 000241
1670 006656 073064 000002
1671 006662 106737 000002
1672 006666 122737 000011 000002
1673 006674 001403
1674 006676 004767 007504
1675
1676 006702 000132
1677 006704 022700 177252
1678 006710 001403
1679 006712 004767 007470
1680
1681 006716 000133
1682 006720 022701 125252
1683 006724 001403
1684 006726
1685 006726 004767 007454
1686
1687 006732 000134
1688 006734 021527 000214
1689 006740 001372
1690 006742 005215
1691
1692
1693
1694
1695
1696
1697 006744 010701
1698 006746 012700 125252
1699 006752 012701 125252
1700 006756 000241
1701 006760 073074 000000
1702 006764 106737 000002
1703 006770 122737 000010 000002
1704 006776 001403
1705 007000 004767 007402
1706
1707 007004 000135
1708 007006 022700 177525
1709 007012 001403
1710 007014 004767 007366
1711
1712 007020 000136
1713 007022 022701 052525
1714 007026 001403
1715 007030
1716 007030 004767 007352
1717

```

```

*****
: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
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
IS:      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      IS
INC      (R5)           ;IF NOT THEN GO TO HLT ABOVE

```

```

*****
:TEST:215      125252 125252 SHIFTED BY 2(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     2(4),%0        ;SHIFT R0,R0!1 BY 2(4)
MFPS     @#PSWORD       ;SAVE PS
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
IS:      JSR      PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE

```

D04

DFKACA MACY11 27(732) 09-SEP-76 19:20 PAGE 42
DFKACA.P11 ASHC INSTRUCTION TESTS

1718	007034	000137	
1719	007036	021527	000215
1720	007042	001372	
1721	007044	005215	
1722			
1723			

137
CMP
BNE
INC

(R5), #215
IS
(R5)

: IS THE STESTN = #215?
: IF NOT THEN GO TO HLT ABOVE

```

1724
1725
1726
1727
1728 007046 010701
1729 007050 012700 125252
1730 007054 012701 125252
1731 007060 000241
1732 007062 073034
1733 007064 106737 000002
1734 007070 122737 000010 000002
1735 007076 001403
1736 007100 004767 007302
1737
1738 007104 000140
1739 007106 022700 177525
1740 007112 001403
1741 007114 004767 007266
1742
1743 007120 000141
1744 007122 022701 052525
1745 007126 001403
1746 007130
1747 007130 004767 007252
1748
1749 007134 000142
1750 007136 021527 000216
1751 007142 001372
1752 007144 005215
1753
1754
1755
1756
1757
1758
1759 007146 010701
1760 007150 012700 125252
1761 007154 012701 125252
1762 007160 000241
1763 007162 073054
1764 007164 106737 000002
1765 007170 122737 000010 000002
1766 007176 001403
1767 007200 004767 007202
1768
1769 007204 000143
1770 007206 022700 177525
1771 007212 001403
1772 007214 004767 007166
1773
1774 007220 000144
1775 007222 022701 052525
1776 007226 001403
1777 007230
1778 007230 004767 007152
1779

```

```

*****
:TEST:216      125252 125252 SHIFTED BY 2(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     2(4)+,%0          ;SHIFT R0,R0!1 BY 2(4)+
MFPS     2#PSWORD         ;SAVE PS
CMPB     #10,2#PSWORD      ;IS THE PS 10?
BEQ      +10
JSR      PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                          ;THE PS IS NOT EQUAL TO 10
CMP      #177525,%0        ;IS THE RESULT 177525?
BEQ      +10
JSR      PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                          ;R0 IS NOT EQUAL TO 177525
CMP      #52525,%0!1      ;IS THE RESULT 52525?
BEQ      +10
IS:      JSR      PC,$HLT  ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                          ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
CMP      (R5),#216        ;IS THE $TESTN = #216?
BNE      IS
INC      (R5)              ;IF NOT THEN GO TO HLT ABOVE
*****
:TEST:217      125252 125252 SHIFTED BY 2-(4) = 177525 52525 PS = 10
*****
TST217: SCOPE1
MOV      #125252,%0          ;LOAD R0 WITH 125252
MOV      #125252,%0!1      ;LOAD R0!1 WITH 125252
CLC
ASHC     2-(4),%0         ;SHIFT R0,R0!1 BY 2-(4)
MFPS     2#PSWORD         ;SAVE PS
CMPB     #10,2#PSWORD      ;IS THE PS 10?
BEQ      +10
JSR      PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                          ;THE PS IS NOT EQUAL TO 10
CMP      #177525,%0        ;IS THE RESULT 177525?
BEQ      +10
JSR      PC,$HLT          ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                          ;R0 IS NOT EQUAL TO 177525
CMP      #52525,%0!1      ;IS THE RESULT 52525?
BEQ      +10
IS:      JSR      PC,$HLT  ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                          ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE

```

1780	007234	000145	
1781	007236	021527	000217
1782	007242	001372	
1783	007244	005215	
1784			
1785			
1786			
1787			
1788			
1789			
1790			
1791			
1792			

145
CMP (R5), #217
BNE IS
INC (R5)

; IS THE \$TESTN = #217?
; IF NOT THEN GO TO HLT ABOVE

1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828

MUL INSTRUCTION TESTS

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

```

TST220: SCOPE
MOV      #1,%0      ;LOAD MULTIPLICAND WITH 1
MUL      #0,%0      ;MULTIPLY 1 * #0
MFPS     @#PSWORD   ;SAVE PS
CMPB     #4,@#PSWORD ;IS PS = 4
BEQ      +10
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;PS IS WRONG
146
CMP      #0,%0      ;IS HIGH ORDER = 0
BEQ      +10
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;HIGH ORDER IS WRONG
147
CMP      #0,%0!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
150
CMP      (R5),#220
BNE      1$
INC      (R5)
;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE

```

000001
000000
000002
000004 000002
007106
000000
007072
000000
007056
000220

```

1829
1830
1831
1832
1833 007342 010701
1834 007344 012700 177777
1835 007350 070027 000001
1836 007354 106737 000002
1837 007360 122737 000010 000002
1838 007366 001403
1839 007370 004767 007012
1840
1841 007374 000151
1842 007376 022700 177777
1843 007402 001403
1844 007404 004767 006776
1845
1846 007410 000152
1847 007412 022701 177777
1848 007416 001403
1849 007420
1850 007420 004767 006762
1851
1852 007424 000153
1853 007426 021527 000221
1854 007432 001372
1855 007434 005215
1856
1857

```

```

:*****
:TEST:221      MUL      -1 * #1 = -1 -1      PS = 10
:*****
TST221: SCOPE
MOV      #-1,%0      ;LOAD MULTIPLICAND WITH -1
MUL      #1,%0      ;MULTIPLY -1 * #1
MFPS     @#PSWORD    ;SAVE PS
CMPB     #10,@#PSWORD ;IS PS = 10
BEQ      .+10
JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG

151
CMP      #-1,%0      ;IS HIGH ORDER = -1
BEQ      .+10
JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;HIGH ORDER IS WRONG

152
CMP      #-1,%0!1    ;IS LOW ORDER = -1
BEQ      .+10

15:
JSR      PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;LOW ORDER IS WRONG OR WRONG SEQUENCE

153
CMP      (R5),#221
BNE     15:
INC      (R5)
;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE

```



```

1858
1859
1860
1861
1862 007436 010701
1863 007440 012702 000002
1864 007444 070227 000002
1865 007450 106737 000002
1866 007454 122737 000000 000002
1867 007462 001403
1868 007464 004767 006716
1869
1870 007470 000154
1871 007472 022702 000000
1872 007476 001403
1873 007500 004767 006702
1874
1875 007504 000155
1876 007506 022703 000004
1877 007512 001403
1878 007514
1879 007514 004767 006666
1880
1881 007520 000156
1882 007522 021527 000222
1883 007526 001372
1884 007530 005215
1885
1886

```

```

:*****
:TEST:222      MUL      2 * #2 = 0 4      PS = 0
:*****
TST222: SCOPE
MOV      #2,%2      ;LOAD MULTIPLICAND WITH 2
MUL      #2,%2      ;MULTIPLY 2 * #2
MFPS     @#PSWORD   ;SAVE PS
CMPB     #0,@#PSWORD ;IS PS = 0
BEQ      +10
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;PS IS WRONG
                        ;IS HIGH ORDER = 0
CMP      #0,%2
BEQ      +10
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;HIGH ORDER IS WRONG
                        ;IS LOW ORDER = 4
CMP      #4,%2!1
BEQ      +10
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;LOW ORDER IS WRONG OR WRONG SEQUENCE
                        ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
1$:
CMP      (R5),#222
BNE      1$
INC      (R5)

```

```

1887
1888
1889
1890
1891 007532 010701          TST223: SCOPE
1892 007534 010501          MOV      R5,R1          ;SAVE R5
1893 007536 012704 001000    MOV      #1000,%4       ;LOAD MULTIPLICAND WITH 1000
1894 007542 070427 000200    MUL      #200,%4        ;MULTIPLY 1000 * #200
1895 007546 106737 000002    MFPS     @#PSWORD       ;SAVE PS
1896 007552 122737 000001 000002  CMPB     #1,@#PSWORD    ;IS PS = 1
1897 007560 001403          BEQ      .+10
1898 007562 004767 006620    JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1899
1900 007566 000157          157
1901 007570 022704 000001    CMP      #1,%4          ;IS HIGH ORDER = 1
1902 007574 001403          BEQ      .+10
1903 007576 004767 006604    JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1904
1905 007602 000160          160
1906 007604 022705 000000    CMP      #0,%4!1       ;IS LOW ORDER = 0
1907 007610 001403          BEQ      .+10
1908 007612
1909 007612 004767 006570    1$: JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1910
1911 007616 000161          161
1912 007620 021127 000223    CMP      (R1),#223     ;CHECK THE TEST NUMBER
1913 007624 001372          BNE     1$             ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
1914 007626 010105          MOV     R1,R5          ;RESTORE R5
1915 007630 005215          INC     (R5)
1916
1917
    
```

```

1918
1919
1920
1921
1922 007632 010701
1923 007634 012700 000002
1924 007640 070027 077777
1925 007644 106737 000002
1926 007650 122737 000001 000002
1927 007656 001403
1928 007660 004767 006522
1929
1930 007664 000162
1931 007666 022700 000000
1932 007672 001403
1933 007674 004767 006506
1934
1935 007700 000163
1936 007702 022701 177776
1937 007706 001403
1938 007710
1939 007710 004767 006472
1940
1941 007714 000164
1942 007716 021527 000224
1943 007722 001372
1944 007724 005215
1945
1946

;*****
;TEST:224 MUL 2 * #77777 = 0 177776 PS = 1
;*****
TST224: SCOPE
MOV #2,%0 ;LOAD MULTIPLICAND WITH 2
MUL #77777,%0 ;MULTIPLY 2 * #77777
MFPB @#PSWORD ;SAVE PS
CMPB #1,@#PSWORD ;IS PS = 1
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG

162
CMP #0,%0 ;IS HIGH ORDER = 0
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;HIGH ORDER IS WRONG

163
CMP #177776,%0!1 ;IS LOW ORDER = 177776
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;LOW ORDER IS WRONG OR WRONG SEQUENCE

164
CMP (R5),#224
BNE 1$
INC (R5) ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE

1$:
    
```

```

1947
1948
1949
1950
1951 007726 010701
1952 007730 012702 007777
1953 007734 070227 000010
1954 007740 106737 000002
1955 007744 122737 000000 000002
1956 007752 001403
1957 007754 004767 006426
1958
1959 007760 000165
1960 007762 022702 000000
1961 007766 001403
1962 007770 004767 006412
1963
1964 007774 000166
1965 007776 022703 077770
1966 010002 001403
1967 010004
1968 010004 004767 006376
1969
1970 010010 000167
1971 010012 021527 000225
1972 010016 001372
1973 010020 005215
1974
1975

:*****
:TEST:225 MUL 7777 * #10 = 0 77770 PS = 0
:*****

TST225: SCOPE
MOV #7777,%2 ;LOAD MULTIPLICAND WITH 7777
MUL #10,%2 ;MULTIPLY 7777 * #10
MFPS @#PSWORD ;SAVE PS
CMPB #0,@#PSWORD ;IS PS = 0
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG

165
CMP #0,%2 ;IS HIGH ORDER = 0
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;HIGH ORDER IS WRONG

166
CMP #77770,%2!1 ;IS LOW ORDER = 77770
BEQ .+10

1S:
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;LOW ORDER IS WRONG OR WRONG SEQUENCE

167
CMP (R5),#225
BNE 1S ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
INC (R5)
    
```

```

1976
1977
1978
1979
1980 010022 010701
1981 010024 010501
1982 010026 012704 077777
1983 010032 070427 077777
1984 010036 106737 000002
1985 010042 122737 000001 000002
1986 010050 001403
1987 010052 004767 006330
1988
1989 010056 000170
1990 010060 022704 037777
1991 010064 001403
1992 010066 004767 006314
1993
1994 010072 000171
1995 010074 022705 000001
1996 010100 001403
1997 010102
1998 010102 004767 006300
1999
2000 010106 000172
2001 010110 021127 000226
2002 010114 001372
2003 010116 010105
2004 010120 005215
2005
2006

```

;*****
 ;TEST:226 MUL 77777 * #77777 = 37777 1 PS = 1
 ;*****

TST226: SCOPE
 MOV R5,R1 ;SAVE R5
 MOV #77777,%4 ;LOAD MULTIPLICAND WITH 77777
 MUL #77777,%4 ;MULTIPLY 77777 * #77777
 MFPS @#PSWORD ;SAVE PS
 CMPB #1,@#PSWORD ;IS PS = 1
 BEQ .+10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;PS IS WRONG
 170
 CMP #37777,%4 ;IS HIGH ORDER = 37777
 BEQ .+10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;HIGH ORDER IS WRONG
 171
 CMP #1,%4!1 ;IS LOW ORDER = 1
 BEQ .+10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
 172
 CMP (R1),#226 ;CHECK THE TEST NUMBER
 BNE 1\$;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
 MOV R1,R5 ;RESTORE R5
 INC (R5)

1\$:

```

2007
2008
2009
2010
2011 010122 010701
2012 010124 012702 177777
2013 010130 070227 077777
2014 010134 106737 000002
2015 010140 122737 000010 000002
2016 010146 001403
2017 010150 004767 006232
2018
2019 010154 000173
2020 010156 022702 177777
2021 010162 001403
2022 010164 004767 006216
2023
2024 010170 000174
2025 010172 022703 100001
2026 010176 001403
2027 010200
2028 010200 004767 006202
2029
2030 010204 000175
2031 010206 021527 000227
2032 010212 001372
2033 010214 005215
2034
2035

*****
:TEST:227 MUL -1 * #77777 = -1 100001 PS = 10
*****

TST227: SCOPE
MOV #-1,%2 ;LOAD MULTIPLICAND WITH -1
MUL #77777,%2 ;MULTIPLY -1 * #77777
MFPS @#PSWORD ;SAVE PS
CMPB #10,@#PSWORD ;IS PS = 10
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG

173
CMP #-1,%2 ;IS HIGH ORDER = -1
BEQ .+10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;HIGH ORDER IS WRONG

174
CMP #100001,%2!1 ;IS LOW ORDER = 100001
BEQ .+10

1S:
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;LOW ORDER IS WRONG OR WRONG SEQUENCE

175
CMP (R5),#227
BNE 1S ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
INC (R5)
    
```



```

2065 :*****
2066 :TEST:231      MUL      125252 * #2 = -1 52524      PS = 11
2067 :*****
2068
2069 010312 010701      TST231: SCOPE
2070 010314 012702      MOV      #125252,%2      ;LOAD MULTIPLICAND WITH 125252
2071 010320 070227      MUL      #2,%2          ;MULTIPLY 125252 * #2
2072 010324 106737      MFPS    @#PSWORD       ;SAVE PS
2073 010330 122737      CMPB    #11,@#PSWORD   ;IS PS = 11
2074 010336 001403      BEQ     +10
2075 010340 004767      JSR     PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2076                                     ;PS IS WRONG
2077 010344 000201      201
2078 010346 022702      CMP     #-1,%2        ;IS HIGH ORDER = -1
2079 010352 001403      BEQ     +10
2080 010354 004767      JSR     PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2081                                     ;HIGH ORDER IS WRONG
2082 010360 000202      202
2083 010362 022703      CMP     #52524,%2!1   ;IS LOW ORDER = 52524
2084 010366 001403      BEQ     +10
2085                                     1$:
2086 010370 004767      JSR     PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2087                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2088 010374 000203      203
2089 010376 021527      CMP     (R5),#231
2090 010402 001372      BNE    1$
2091 010404 005215      INC     (R5)           ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2092
2093
    
```



```

2094 *****
2095 :TEST:232      MUL      125252 * #40000 = 165252 100000      PS = 11
2096 *****
2097
2098 TST232: SCOPE
2099 MOV      R5,R1      ;SAVE R5
2100 MOV      #125252,%4 ;LOAD MULTIPLICAND WITH 125252
2101 MUL      #40000,%4 ;MULTIPLY 125252 * #40000
2102 MFPS     @#PSWORD  ;SAVE PS
2103 CMPB     #11,@#PSWORD ;IS PS = 11
2104 BEQ      +10
2105 JSR     PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2106                ;PS IS WRONG
2107                204
2108 CMP      #165252,%4 ;IS HIGH ORDER = 165252
2109 BEQ      +10
2110 JSR     PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2111                ;HIGH ORDER IS WRONG
2112                205
2113 CMP      #100000,%4!1 ;IS LOW ORDER = 100000
2114 BEQ      +10
2115                1S:
2116 JSR     PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2117                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2118                206
2119 CMP      (R1),#232    ;CHECK THE TEST NUMBER
2120 BNE     1S           ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2121 MOV     R1,R5        ;RESTORE R5
2122 INC     (R5)
2123
2124

```

```

2125                                     :*****
2126                                     :TEST-233      MUL      107070 * #107070 = 31222 26100      PS = 1
2127                                     :*****
2128
2129 010506 010701                          TST233: SCOPE
2130 010510 012700 107070                   MOV      #107070,%0      ;LOAD MULTIPLICAND WITH 107070
2131 010514 070027 107070                   MUL      #107070,%0      ;MULTIPLY 107070 * #107070
2132 010520 106737 000002                   MFPS     @#PSWORD        ;SAVE PS
2133 010524 122737 000001 000002           CMPB     #1,@#PSWORD     ;IS PS = 1
2134 010532 001403                          BEQ      .+10
2135 010534 004767 005646                   JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2136                                     ;PS IS WRONG
2137 010540 000207                          207
2138 010542 022700 031222                   CMP      #31222,%0      ;IS HIGH ORDER = 31222
2139 010546 001403                          BEQ      .+10
2140 010550 004767 005632                   JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2141                                     ;HIGH ORDER IS WRONG
2142 010554 000210                          210
2143 010556 022701 026100                   CMP      #26100,%0!1    ;IS LOW ORDER = 26100
2144 010562 001403                          BEQ      .+10
2145                                     1$:
2146 010564 004767 005616                   JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2147                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2148 010570 000211                          211
2149 010572 021527 000233                   CMP      (R5),#233
2150 010576 001372                          BNE     1$
2151 010600 005215                          INC      (R5)
2152
2153

```

```

2154
2155
2156
2157
2158 010602 010701
2159 010604 012701 177777
2160 010610 070127 000001
2161 010614 106737 000002
2162 010620 122737 000010 000002
2163 010626 001403
2164 010630 004767 005552
2165
2166 010634 000212
2167 010636 022701 177777
2168 010642 001403
2169 010644 004767 005536
2170
2171 010650 000213
2172 010652 022701 177777
2173 010656 001403
2174 010660
2175 010660 004767 005522
2176
2177 010664 000214
2178 010666 021527 000234
2179 010672 001372
2180 010674 005215
2181
2182

```

 :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
 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
 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)

1\$:

```

2183
2184
2185
2186
2187 010676 010701
2188 010700 012703 177777
2189 010704 070327 000000
2190 010710 106737 000002
2191 010714 122737 000004 000002
2192 010722 001403
2193 010724 004767 005456
2194
2195 010730 000215
2196 010732 022703 000000
2197 010736 001403
2198 010740 004767 005442
2199
2200 010744 000216
2201 010746 022703 000000
2202 010752 001403
2203 010754
2204 010754 004767 005426
2205
2206 010760 000217
2207 010762 021527 000235
2208 010766 001372
2209 010770 005215
2210
2211
    
```

 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
 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
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
 1\$:
 217
 CMP (R5),#235
 BNE 1\$;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
 INC (R5)

```

2212 :*****
2213 :TEST:236      MUL      77777 * #100000 = 100000 100000      PS = 11
2214 :*****
2215
2216 010772 010701      TST236: SCOPE
2217 010774 010501      MOV      R5,R1      ;SAVE R5
2218 010776 012705 077777  MOV      #77777,%5   ;LOAD MULTIPLICAND WITH 77777
2219 011002 070527 100000  MUL      #100000,%5 ;MULTIPLY 77777 * #100000
2220 011006 106737 000002  MFPS    @#PSWORD    ;SAVE PS
2221 011012 122737 000011 000002  CMPB    #11,@#PSWORD ;IS PS = 11
2222 011020 001403      BEQ      +10
2223 011022 004767 005360  JSR     PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2224                                     ;PS IS WRONG
2225 011026 000220      220
2226 011030 022705 100000  CMP     #100000,%5  ;IS HIGH ORDER = 100000
2227 011034 001403      BEQ      +10
2228 011036 004767 005344  JSR     PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2229                                     ;HIGH ORDER IS WRONG
2230 011042 000221      221
2231 011044 022705 100000  CMP     #100000,%5!1 ;IS LOW ORDER = 100000
2232 011050 001403      BEQ      +10
2233 011052                                     1$:
2234 011052 004767 005330  JSR     PC,$HLT     ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2235                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2236 011056 000222      222
2237 011060 021127 000236  CMP     (R1),#236   ;CHECK THE TEST NUMBER
2238 011064 001372      BNE     1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2239 011066 010105      MOV     R1,R5      ;RESTORE R5
2240 011070 005215      INC     (R5)
2241
2242

```

```

2243
2244
2245
2246
2247 011072 010701
2248 011074 012701 177777
2249 011100 070127 077777
2250 011104 106737 000002
2251 011110 122737 000010 000002
2252 011116 001403
2253 011120 004767 005262
2254
2255 011124 000223
2256 011126 022701 100001
2257 011132 001403
2258 011134 004767 005246
2259
2260 011140 000224
2261 011142 022701 100001
2262 011146 001403
2263 011150
2264 011150 004767 005232
2265
2266 011154 000225
2267 011156 021527 000237
2268 011162 001372
2269 011164 005215
2270
2271

```

 :TEST:237 MUL -1 * #77777 = 100001 100001 PS = 10

TST237: SCOPE
 MOV #-1,%1 ;LOAD MULTIPLICAND WITH -1
 MUL #77777,%1 ;MULTIPLY -1 * #77777
 MFPS @#PSWORD ;SAVE PS
 CMPB #10,@#PSWORD ;IS PS = 10
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;PS IS WRONG
 223
 CMP #100001,%1 ;IS HIGH ORDER = 100001
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;HIGH ORDER IS WRONG
 224
 CMP #100001,%1!1 ;IS LOW ORDER = 100001
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
 225
 CMP (R5),#237
 BNE 1\$;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
 INC (R5)

1\$:

```

2272
2273
2274
2275
2276 011166 010701
2277 011170 012703 077777
2278 011174 070327 077777
2279 011200 106737 000002
2280 011204 122737 000001 000002
2281 011212 001403
2282 011214 004767 005166
2283
2284 011220 000226
2285 011222 022703 000001
2286 011226 001403
2287 011230 004767 005152
2288
2289 011234 000227
2290 011236 022703 000001
2291 011242 001403
2292 011244
2293 011244 004767 005136
2294
2295 011250 000230
2296 011252 021527 000240
2297 011256 001372
2298 011260 005215
2299
2300

```

 TEST:240 .. MUL 77777 * #77777 = 1 1 PS = 1

TST240: SCOPE
 MOV #77777,%3 ;LOAD MULTIPLICAND WITH 77777
 MUL #77777,%3 ;MULTIPLY 77777 * #77777
 MFPS @#PSWORD ;SAVE PS
 CMPB #1,@#PSWORD ;IS PS = 1
 BEQ .+10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;PS IS WRONG
 226
 CMP #1,%3 ;IS HIGH ORDER = 1
 BEQ .+10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;HIGH ORDER IS WRONG
 227
 CMP #1,%3!1 ;IS LOW ORDER = 1
 BEQ .+10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
 230
 CMP (R5),#240
 BNE 1\$;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
 INC (R5)

1\$:

```

2301
2302
2303
2304
2305 011262 010701
2306 011264 010501
2307 011266 012705 000002
2308 011272 070527 000002
2309 011276 106737 000002
2310 011302 122737 000000 000002
2311 011310 001403
2312 011312 004767 005070
2313
2314 011316 000231
2315 011320 022705 000004
2316 011324 001403
2317 011326 004767 005054
2318
2319 011332 000232
2320 011334 022705 000004
2321 011340 001403
2322 011342
2323 011342 004767 005040
2324
2325 011346 000233
2326 011350 021127 000241
2327 011354 001372
2328 011356 010105
2329 011360 005215
2330
2331
    
```

 :TEST:241 MUL 2 * #2 = 4 4 PS = 0

TST241: SCOPE

```

                MOV R5,R1 ;SAVE R5
                MOV #2,%5 ;LOAD MULTIPLICAND WITH 2
                MUL #2,%5 ;MULTIPLY 2 * #2
                MFPS @#PSWORD ;SAVE PS
                CMPB #0,@#PSWORD ;IS PS = 0
                BEQ .+10
                JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                ;PS IS WRONG
                231
                CMP #4,%5 ;IS HIGH ORDER = 4
                BEQ .+10
                JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                ;HIGH ORDER IS WRONG
                232
                CMP #4,%5!1 ;IS LOW ORDER = 4
                BEQ .+10
                JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
                233
                CMP (R1),#241 ;CHECK THE TEST NUMBER
                BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
                MOV R1,R5 ;RESTORE R5
                INC (R5)
    
```

1\$:


```

2332 011362 012702 040000
2333 011366 012703 000034
2334 011372 012704 000036
2335
2336
2337
2338
2339
2340 011376 010701
2341 011400 012700 125252
2342 011404 070067 166424
2343 011410 106737 000002
2344 011414 122737 000011 000002
2345 011422 001403
2346 011424 004767 004756
2347
2348 011430 000234
2349 011432 022700 165252
2350 011436 001403
2351 011440 004767 004742
2352
2353 011444 000235
2354 011446 022701 100000
2355 011452 001403
2356 011454
2357 011454 004767 004726
2358
2359 011460 000236
2360 011462 021527 000242
2361 011466 001372
2362 011470 005215
2363
2364
    
```

```

*****
:TEST:242      MUL      125252 * S5 = 165252 100000      PS = 11
*****
    
```

TST242: SCOPE

```

MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
MUL      S5,%0          ;MULTIPLY 125252 * S5
MFPS     @#PSWORD       ;SAVE PS
CMPB     #11,@#PSWORD   ;IS PS = 11
BEQ      .+10
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;PS IS WRONG
234      #165252,%0      ;IS HIGH ORDER = 165252
CMP      .+10
BEQ      JSR            ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;HIGH ORDER IS WRONG
235      #100000,%0!1    ;IS LOW ORDER = 100000
CMP      .+10
BEQ      JSR            ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;LOW ORDER IS WRONG OR WRONG SEQUENCE
1$:      JSR            PC,$HLT
236      (R5),#242
CMP      1$
BNE      INC            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
INC      (R5)
    
```

```

2365
2366
2367
2368
2369 011472 010701
2370 011474 012700 125252
2371 011500 070077 166332
2372 011504 106737 000002
2373 011510 122737 000011 000002
2374 011516 001403
2375 011520 004767 004662
2376
2377 011524 000237
2378 011526 022700 165252
2379 011532 001403
2380 011534 004767 004646
2381
2382 011540 000240
2383 011542 022701 100000
2384 011546 001403
2385 011550
2386 011550 004767 004632
2387
2388 011554 000241
2389 011556 021527 000243
2390 011562 001372
2391 011564 005215
2392
2393

```

```

*****
:TEST:243      MUL      125252 * Q56 = 165252 100000      PS = 11
*****
TST243: SCOPE
MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
MUL      Q56,%0          ;MULTIPLY 125252 * Q56
MFPS     Q#PSWORD        ;SAVE PS
CMPB     #11,Q#PSWORD     ;IS PS = 11
BEQ      .+10
JSR      PC,$HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG

237
CMP      #165252,%0      ;IS HIGH ORDER = 165252
BEQ      .+10
JSR      PC,$HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;HIGH ORDER IS WRONG

240
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

241
CMP      (R5),#243
BNE     1$
INC      (R5)
;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE

```

```

2394 ;*****
2395 ;TEST:244      MUL      125252 * @#55 = 165252 100000      PS = 11
2396 ;*****
2397
2398 011566 010701          TST244: SCOPE
2399 011570 012700 125252      MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
2400 011574 070037 000034      MUL      @#55,%0      ;MULTIPLY 125252 * @#55
2401 011600 106737 000002      MFPS     @#PSWORD     ;SAVE PS
2402 011604 122737 000011 000002  CMPB     #11,@#PSWORD ;IS PS = 11
2403 011612 001403          BEQ      .+10
2404 011614 004767 004566      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2405                                ;PS IS WRONG
2406 011620 000242          242
2407 011622 022700 165252      CMP      #165252,%0    ;IS HIGH ORDER = 165252
2408 011626 001403          BEQ      .+10
2409 011630 004767 004552      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2410                                ;HIGH ORDER IS WRONG
2411 011634 000243          243
2412 011636 022701 100000      CMP      #100000,%0!1 ;IS LOW ORDER = 100000
2413 011642 001403          BEQ      .+10
2414 011644          1$:
2415 011644 004767 004536      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2416                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2417 011650 000244          244
2418 011652 021527 000244      CMP      (R5),#244
2419 011656 001372          BNE     1$
2420 011660 005215          INC      (R5)
2421
2422
    
```

011752
011750
011744
011742
011736
011734
011730
011726
011722
011720
011714
011712
011706
011704
011676
011672
011670
011664
011662

010701
012700 125252
070002
106737 000002
122737 000011 000002
001403
004767 004474

000245
022700 165252
001403
004767 004460

000246
022701 100000
001403
004767 004444

000247
021527 000245
001372
005215

: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
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
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;LOW ORDER IS WRONG OR WRONG SEQUENCE

1\$:
247
CMP (R5),#245
BNE 1\$;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
INC (R5)

2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480

:TEST:246 MUL 125252 * (3)+ = 165252 100000 PS = 11

011754 010701
011756 012700 125252
011762 070023
011764 106737 000002
011770 122737 000011 000002
011776 001403
012000 004767 004402

012004 000250
012006 022700 165252
012012 001403
012014 004767 004366

012020 000251
012022 022701 100000
012026 001403
012030
012030 004767 004352

012034 000252
012036 021527 000246
012042 001372
012044 005215

TST246: SCOPE
MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
MUL (3)+,%0 ;MULTIPLY 125252 * (3)+
MFPS @#PSWORD ;SAVE PS
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
BEQ .+10

1\$: JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;LOW ORDER IS WRONG OR WRONG SEQUENCE

252
CMP (R5),#246
1\$: BNE 1\$;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
INC (R5)

2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509

012046 010701
012050 012700 125252
012054 070043
012056 106737 000002
012062 122737 000011 000002
012070 001403
012072 004767 004310

012076 000253
012100 022700 165252
012104 001403
012106 004767 004274

012112 000254
012114 022701 100000
012120 001403
012122
012122 004767 004260

012126 000255
012130 021527 000247
012134 001372
012136 005215

:TEST:247 MUL 125252 * -(3) = 165252 100000 PS = 11

TST247: SCOPE
MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
MUL -(3),%0 ;MULTIPLY 125252 * -(3)
MFPS @#PSWORD ;SAVE PS
CMPB #11,@#PSWORD ;IS PS = 11
BEQ .+10
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG

253
CMP #165252,%0 ;IS HIGH ORDER = 165252
BEQ .+10
JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;HIGH ORDER IS WRONG

254
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

255
CMP (R5),#247
BNE 1\$;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
INC (R5)

```

2510
2511
2512
2513
2514
2515 012140 010701
2516 012142 012700 125252
2517 012146 070064 000002
2518 012152 106737 000002
2519 012156 122737 000011 000002
2520 012164 001403
2521 012166 004767 004214
2522
2523 012172 000256
2524 012174 022700 165252
2525 012200 001403
2526 012202 004767 004200
2527 012206 000257
2528 012210 022701 100000
2529 012214 001403
2530 012216
2531 012216 004767 004164
2532
2533 012222 000260
2534 012224 021527 000250
2535 012230 001372
2536 012232 005215
2537
2538

```

```

:*****
:TEST:250      MUL      125252 * 2(4) = 165252 100000      PS = 11
:*****
TST250: SCOPE
MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
MUL      2(4),%0        ;MULTIPLY 125252 * 2(4)
MFPS     @#PSWORD        ;SAVE PS
CMPB     #11,@#PSWORD    ;IS PS = 11
BEQ      +10
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;PS IS WRONG
256
CMP      #165252,%0      ;IS HIGH ORDER = 165252
BEQ      +10
JSR      PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
                        ;HIGH ORDER IS WRONG
257
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
260
CMP      (R5),#250
BNE     1$
INC      (R5)
;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE

```

F06

DFKACA MACY11 27(732) 08-SEP-76 18:20 PAGE 70
 DFKACA.P11 MUL INSTRUCTION TESTS

```

2539 ;*****
2540 ;TEST:251      MUL      125252 * 3(4) = 165252 100000      PS = 11
2541 ;*****
2542
2543 TST251: SCOPE
2544 012234 010701      MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
2545 012236 012700 125252      MUL      3(4),%0      ;MULTIPLY 125252 * 3(4)
2546 012242 070074 000000      MFPS     3#PSWORD      ;SAVE PS
2547 012246 106737 000002      CMPB    3#PSWORD      ;IS PS = 11
2548 012252 122737 000011 000002      BEQ     .+10
2549 012260 001403      BEQ     .+10
2550 012262 004767 004120      JSR     PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2551 ;PS IS WRONG
2552 012266 000261      261
2553 012270 022700 165252      CMP     #165252,%0      ;IS HIGH ORDER = 165252
2554 012274 001403      BEQ     .+10
2555 012276 004767 004104      JSR     PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2556 ;HIGH ORDER IS WRONG
2557 012302 000262      262
2558 012304 022701 100000      CMP     #100000,%0!1    ;IS LOW ORDER = 100000
2559 012310 001403      BEQ     .+10
2560 012312 004767 004070      JSR     PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2561 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2562 012316 000263      263
2563 012320 021527 000251      CMP     (R5),#251
2564 012324 001372      BNE    1$
2565 012326 005215      INC     (R5)
2566
2567
  
```



```

2568 ;*****
2569 :TEST:252      MUL      125252 * 2(4)+ = 165252 100000      PS = 11
2570 ;*****
2571
2572 012330 010701          TST252: SCOPE
2573 012332 012700 125252      MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
2574 012336 070034          MUL      2(4)+,%0      ;MULTIPLY 125252 * 2(4)+
2575 012340 106737 000002      MFPS          ;SAVE PS
2576 012344 122737 000011 000002      CMPB     #11,2#PSWORD ;IS PS = 11
2577 012352 001403          BEQ      .+10
2578 012354 004767 004026      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2579                                     ;PS IS WRONG
2580 012360 000264          264
2581 012362 022700 165252      CMP      #165252,%0    ;IS HIGH ORDER = 165252
2582 012366 001403          BEQ      .+10
2583 012370 004767 004012      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2584                                     ;HIGH ORDER IS WRONG
2585 012374 000265          265
2586 012376 022701 100000      CMP      #100000,%0!1 ;IS LOW ORDER = 100000
2587 012402 001403          BEQ      .+10
2588 012404          1$:
2589 012404 004767 003776      JSR      PC,$HLT      ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2590                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2591 012410 000266          266
2592 012412 021527 000252      CMP      (R5),#252
2593 012416 001372          BNE     1$
2594 012420 005215          INC     (R5)
2595
2596
    
```

```

2597 ;*****
2598 ;TEST:253      MUL      125252 * 3-(4) = 165252 100000      PS = 11
2599 ;*****
2600
2601 012422 010701          TST253: SCOPE
2602 012424 012700 125252      MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
2603 012430 070054          MUL      3-(4),%0      ;MULTIPLY 125252 * 3-(4)
2604 012432 106737 000002      MFPS     3#PSWORD      ;SAVE PS
2605 012436 122737 000011 000002      CMPB    #11,3#PSWORD   ;IS PS = 11
2606 012444 001403          BEQ      .+10
2607 012446 004767 003734      JSR     PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2608                                     ;PS IS WRONG
2609 012452 000267          267
2610 012454 022700 165252      CMP     #165252,%0     ;IS HIGH ORDER = 165252
2611 012460 001403          BEQ     .+10
2612 012462 004767 003720      JSR     PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2613                                     ;HIGH ORDER IS WRONG
2614 012466 000270          270
2615 012470 022701 100000      CMP     #100000,%0!1   ;IS LOW ORDER = 100000
2616 012474 001403          BEQ     .+10
2617 012476                                     1$:
2618 012476 004767 003704      JSR     PC,$HLT        ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2619                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2620 012502 000271          271
2621 012504 021527 000253      CMP     (R5),#253
2622 012510 001372          1$:
2623 012512 005215          BNE
2624                                     INC     (R5)
2625                                     ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE

```

 : DIV INSTRUCTION TESTS
 :*****

 : TEST:254 DIV 0 4 / #2 = 2 REM = 0 PS = 0
 :*****

2626
 2627
 2628
 2629
 2630
 2631
 2632
 2633
 2634
 2635
 2636
 2637
 2638
 2639
 2640
 2641
 2642
 2643
 2644
 2645
 2646
 2647
 2648
 2649
 2650
 2651
 2652
 2653
 2654
 2655
 2656
 2657
 2658
 2659
 2660
 2661
 2662
 2663
 2664
 2665

012514	010701		
012516	012700	000000	
012522	012701	000004	
012526	071027	000002	
012532	106737	000002	
012536	122737	000000	000002
012544	001403		
012546	004767	003634	
012552	000272		
012554	022700	000002	
012560	001403		
012562	004767	003620	
012566	000273		
012570	022701	000000	
012574	001403		
012576	004767	003604	
012602	000274		
012604	021527	000254	
012610	001403		
012612	004767	003570	
012616	000275		
012620	005215		

TST254: SCOPE

MOV	#0,%0	:LOAD HIGH ORDER WITH 0
MOV	#4,%0+1	:LOAD LOW ORDER WITH 4
DIV	#2,%0	:DIVIDE BY #2
MFPS	@#PSWORD	:SAVE PS
CMPB	#0,@#PSWORD	:IS PS = 0
BEQ	+10	
JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
		:PS IS WRONG
	272	
CMP	#2,%0	:IS QUOTIENT = 2
BEQ	+10	
JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
		:QUOTIENT IS WRONG
	273	
CMP	#0,%0+1	:IS REMAINDER = 0
BEQ	+10	
JSR	PC,\$HLT	:SEEN AN ERROR, GO TO TH HALT ROUTINE
		:WRONG REMAINDER
	274	
CMP	(R5),#254	
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
	275	
INC	(R5)	

```

2666 ;*****
2667 ;TEST:255 DIV -1 -9. / #3 = -3 REM = 0 PS = 10
2668 ;*****
2669
2670 TST255: SCOPE
2671 012622 010701 MOV #-1,%2 ;LOAD HIGH ORDER WITH -1
2672 012624 012702 177777 MOV #-9,%2+1 ;LOAD LOW ORDER WITH -9.
2673 012630 012703 177767 DIV #3,%2 ;DIVIDE BY #3
2674 012634 071227 000003 MFPS @#PSWORD ;SAVE PS
2675 012640 106737 000002
2676 012644 122737 000010 000002 CMPB #10,@#PSWORD ;IS PS = 10
2677 012652 001403 BEQ +10
2678 012654 004767 003526 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2679 ;PS IS WRCNG
2680 012660 000276 276
2681
2682 012662 022702 177775 CMP #-3,%2 ;IS QUOTIENT = -3
2683 012666 001403 BEQ +10
2684 012670 004767 003512 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2685 ;QUOTIENT IS WRONG
2686 012674 000277 277
2687
2688 012676 022703 000000 CMP #0,%2+1 ;IS REMAINDER = 0
2689 012702 001403 BEQ +10
2690 012704 004767 003476 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2691 ;WRONG REMAINDER
2692 012710 000300 300
2693 012712 021527 000255 CMP (R5),#255
2694 012716 001403 BEQ +10 ;IF IN WRONG SEQUENCE GO TO THE HLT
2695 012720 004767 003462 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2696 ;TEST IS IN WRONG SEQUENCE
2697 012724 000301 301
2698 012726 005215 INC (R5)
2699
  
```

```

2700
2701
2702
2703
2704 012730 010701
2705 012732 010501
2706 012734 012704 000000
2707 012740 012705 000011
2708 012744 071427 000002
2709 012750 106737 000002
2710
2711 012754 122737 000000 000002
2712 012752 001403
2713 012764 004767 003416
2714
2715 012770 000302
2716
2717 012772 022704 000004
2718 012776 001403
2719 013000 004767 003402
2720
2721 013004 000303
2722
2723 013006 022705 000001
2724 013012 001403
2725 013014 004767 003366
2726
2727 013020 000304
2728 013022 010105
2729 013024 021527 000256
2730 013030 001403
2731 013032 004767 003350
2732
2733 013036 000305
2734 013040 005215
2735

```

:*****
 :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 J#PSWORD ;SAVE PS
 CMPB #0,J#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
 JSR PC,\$HLT ;IF IN WRONG SEQUENCE GO TO THE HLT
 ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;TEST IS IN WRONG SEQUENCE
 305
 INC (R5)

```

2736
2737
2738
2739
2740 013042 010701
2741 013044 012700 177777
2742 013050 012701 177767
2743 013054 071027 000002
2744 013060 106737 000002
2745
2746 013064 122737 000010 000002
2747 013072 001403
2748 013074 004767 003306
2749
2750 013100 000306
2751
2752 013102 022700 177774
2753 013106 001403
2754 013110 004767 003272
2755
2756 013114 000307
2757
2758 013116 022701 177777
2759 013122 001403
2760 013124 004767 003256
2761
2762 013130 000310
2763 013132 021527 000257
2764 013136 001403
2765 013140 004767 003242
2766
2767 013144 000311
2768 013146 005215
2769

```

```

:*****
: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
CMPB #10,@#PSWORD ;IS PS = 10
BEQ +10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG
306
CMP #-4,%0 ;IS QUOTIENT = -4
BEQ +10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;QUOTIENT IS WRONG
307
CMP #-1,%0+1 ;IS REMAINDER = -1
BEQ +10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;WRONG REMAINDER
310
CMP (R5),#257
BEQ +10 ;IF IN WRONG SEQUENCE GO TO THE HLT
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;TEST IS IN WRONG SEQUENCE
311
INC (R5)

```

```

2770
2771
2772
2773
2774 013150 010701
2775 013152 012702 000000
2776 013156 012703 000002
2777 013162 071227 177775
2778 013166 106737 000002
2779
2780 013172 122737 000004 000002
2781 013200 001403
2782 013202 004767 003200
2783
2784 013206 000312
2785
2786 013210 022702 000000
2787 013214 001403
2788 013216 004767 003164
2789
2790 013222 000313
2791
2792 013224 022703 000002
2793 013230 001403
2794 013232 004767 003150
2795
2796 013236 000314
2797 013240 021527 000260
2798 013244 001403
2799 013246 004767 003134
2800
2801 013252 000315
2802 013254 005215
2803

```

 :TEST:260 DIV 0 2 / #-3 = 0 REM = 2 PS = 4

TST260: SCOPE
 MOV #0,%2 ;LOAD HIGH ORDER WITH 0
 MOV #2,%2+1 ;LOAD LOW ORDER WITH 2
 DIV #-3,%2 ;DIVIDE BY #-3
 MFPS @#PSWORD ;SAVE PS
 CMPB #4,@#PSWORD ;IS PS = 4
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;PS IS WRONG
 312
 CMP #0,%2 ;IS QUOTIENT = 0
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;QUOTIENT IS WRONG
 313
 CMP #2,%2+1 ;IS REMAINDER = 2
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;WRONG REMAINDER
 314
 CMP (R5),#260
 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
 315
 INC (R5)

```

2804
2805
2806
2807
2808 013256 010701
2809 013260 010501
2810 013262 012704 177777
2811 013266 012705 177776
2812 013272 071427 000003
2813 013276 106737 000002
2814
2815 013302 122737 000004 000002
2816 013310 001403
2817 013312 004767 003070
2818
2819 013316 000316
2820
2821 013320 022704 000000
2822 013324 001403
2823 013326 004767 003054
2824
2825 013332 000317
2826
2827 013334 022705 177776
2828 013340 001403
2829 013342 004767 003040
2830
2831 013346 000320
2832 013350 010105
2833 013352 021527 000261
2834 013356 001403
2835 013360 004767 003022
2836
2837 013364 000321
2838 013366 005215
2839

```

```

;*****
;TEST:261      DIV      -1 -2 / #3 = 0      REM = -2      PS = 4
;*****
TST261: SCOPE
MOV      R5,R1      ;SAVE R5
MOV      #-1,%4     ;LOAD HIGH ORDER WITH -1
MOV      #-2,%4+1   ;LOAD LOW ORDER WITH -2
DIV      #3,%4      ;DIVIDE BY #3
MFPS     @#PSWORD   ;SAVE PS
CMPB     #4,@#PSWORD ;IS PS = 4
BEQ      .+10
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG
316
CMP      #0,%4      ;IS QUOTIENT = 0
BEQ      .+10
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;QUOTIENT IS WRONG
317
CMP      #-2,%4+1   ;IS REMAINDER = -2
BEQ      .+10
JSR      PC,$HLT    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;WRONG REMAINDER
320
MOV      R1,R5      ;RESTORE R5
CMP      (R5),#261
BEQ      .+10
JSR      PC,$HLT    ;IF IN WRONG SEQUENCE GO TO THE HLT
;SEEN AN ERROR, GO TO TH HALT ROUTINE
;TEST IS IN WRONG SEQUENCE
321
INC      (R5)

```


000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073

:TEST:262 DIV -1 -1 / #1 = -1 REM = 0 PS = 10

```

TST262: SCOPE
MOV #-1,%0 ;LOAD HIGH ORDER WITH -1
MOV #-1,%0+1 ;LOAD LOW ORDER WITH -1
DIV #1,%0 ;DIVIDE BY #1
MFPS @#PSWORD ;SAVE PS

013270 010701
013372 012700 177777
013376 012701 177777
013402 071027 000001
013406 106737 000002

013412 122737 000010 000002 CMPB #10,@#PSWORD ;IS PS = 10
013420 001403 BEQ +10
013422 004767 002760 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG

013426 000322 322

013430 022700 177777 CMP #-1,%0 ;IS QUOTIENT = -1
013434 001403 BEQ +10
013436 004767 002744 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;QUOTIENT IS WRONG

013442 000323 323

013444 022701 000000 CMP #0,%0+1 ;IS REMAINDER = 0
013450 001403 BEQ +10
013452 004767 002730 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;WRONG REMAINDER

013456 000324 324
013460 021527 000262 CMP (R5),#262
013464 001403 BEQ +10 ;IF IN WRONG SEQUENCE GO TO THE HLT
013466 004767 002714 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;TEST IS IN WRONG SEQUENCE

013472 000325 325
013474 005215 INC (R5)

```

```

*****
:TEST:263 DIV 0 0 / #1 = 0 REM = 0 PS = 4
*****
2874
2875
2876
2877
2878 013476 010701
2879 013500 012700 000000
2880 013504 012701 000000
2881 013510 071027 000001
2882 013514 106737 000002
2883
2884 013520 122737 000004 000002
2885 013526 001403
2886 013530 004767 002652
2887
2888 013534 000326
2889
2890 013536 022700 000000
2891 013542 001403
2892 013544 004767 002636
2893
2894 013550 000327
2895
2896 013552 022701 000000
2897 013556 001403
2898 013560 004767 002622
2899
2900 013564 000330
2901 013566 021527 000263
2902 013572 001403
2903 013574 004767 002606
2904
2905 013600 000331
2906 013602 005215
2907

```

```

TST263: S...E
MOV #0,%0 ;LOAD HIGH ORDER WITH 0
MOV #0,%0+1 ;LOAD LOW ORDER WITH 0
DIV #1,%0 ;DIVIDE BY #1
MFPS @#PSWORD ;SAVE PS

CMP #4,@#PSWORD ;IS PS = 4
BEQ +10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG

CMP #0,%0 ;IS QUOTIENT = 0
BEQ +10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;QUOTIENT IS WRONG

CMP #0,%0+1 ;IS REMAINDER = 0
BEQ +10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;WRONG REMAINDER

CMP (R5),#263
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

331
INC (R5)

```

```

2908 :*****
2909 :TEST:264 DIV -1 125252 / #2 = 152525 REM = 0 PS = 10
2910 :*****
2911
2912 013604 010701 TST264: SCOPE
2913 013606 012702 MOV #-1,%2 ;LOAD HIGH ORDER WITH -1
2914 013612 012703 MOV #125252,%2+1 ;LOAD LOW ORDER WITH 125252
2915 013616 071227 DIV #2,%2 ;DIVIDE BY #2
2916 013622 106737 MFPS @#PSWORD ;SAVE PS
2917
2918 013626 122737 000010 000002 CMPB #10,@#PSWORD ;IS PS = 10
2919 013634 001403 BEQ +10
2920 013636 004767 002544 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2921 ;PS IS WRONG
2922 013642 000332 332
2923
2924 013644 022702 152525 CMP #152525,%2 ;IS QUOTIENT = 152525
2925 013650 001403 BEQ +10
2926 013652 004767 002530 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2927 ;QUOTIENT IS WRONG
2928 013656 000333 333
2929
2930 013660 022703 000000 CMP #0,%2+1 ;IS REMAINDER = 0
2931 013664 001403 BEQ +10
2932 013666 004767 002514 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2933 ;WRONG REMAINDER
2934 013672 000334 334
2935 013674 021527 000264 CMP (R5),#264
2936 013700 001403 BEQ +10 ;IF IN WRONG SEQUENCE GO TO THE HLT
2937 013702 004767 002500 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2938 ;TEST IS IN WRONG SEQUENCE
2939 013706 000335 335
2940 013710 005215 INC (R5)
2941

```

```

2943
2944
2945
2946 013712 010701
2947 013714 010501
2948 013716 012704 177777
2949 013722 012705 177777
2950 013726 071427 177777
2951 013732 105737 000002
2952
2953 013736 122737 000000 000002
2954 013744 001403
2955 013746 004767 002434
2956
2957 013752 000336 336
2958
2959 013754 022704 000001
2960 013760 001403
2961 013762 004767 002420
2962
2963 013766 000337 337
2964
2965 013770 022705 000000
2966 013774 001403
2967 013776 004767 002404
2968
2969 014002 000340 340
2970 014004 010105
2971 014006 021527 000265
2972 014012 001403
2973 014014 004767 002366
2974
2975 014020 000341 341
2976 014022 005215 INC (R5)
2977
    :*****
    :TEST:265 DIV -1 -1 / #-1 = 1 REM = 0 PS = 0
    :*****
    TST265: SCOPE
    MOV R5,R1 ;SAVE R5
    MOV #-1,%4 ;LOAD HIGH ORDER WITH -1
    MOV #-1,%4+1 ;LOAD LOW ORDER WITH -1
    DIV #-1,%4 ;DIVIDE BY #-1
    MFPS J#PSWORD ;SAVE PS
    CMPB #0,J#PSWORD ;IS PS = 0
    BEQ +10
    JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
    ;PS IS WRONG
    CMP #1,%4 ;IS QUOTIENT = 1
    BEQ +10
    JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
    ;QUOTIENT IS WRONG
    CMP #0,%4+1 ;IS REMAINDER = 0
    BEQ +10
    JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
    ;WRONG REMAINDER
    MOV R1,R5 ;RESTORE R5
    CMP (R5),#265
    BEQ +10
    JSR PC,$HLT ;IF IN WRONG SEQUENCE GO TO THE HLT
    ;SEEN AN ERROR, GO TO TH HALT ROUTINE
    ;TEST IS IN WRONG SEQUENCE
    
```

```

2978 ;*****
2979 ;TEST:266 DIV 25253 1 / #125252 = 100000 REM = 1 PS = 10
2980 ;*****
2981
2982 014024 010701 TST266: SCOPE
2983 014026 012700 MOV #25253,%0 ;LOAD HIGH ORDER WITH 25253
2984 014032 012701 MOV #1,%0+1 ;LOAD LOW ORDER WITH 1
2985 014036 071027 DIV #125252,%0 ;DIVIDE BY #125252
2986 014042 106737 MFPS @#PSWORD ;SAVE PS
2987
2988 014046 122737 000010 000002 CMPB #10,@#PSWORD ;IS PS = 10
2989 014054 001403 BEQ +10
2990 014056 004767 002324 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2991 ;PS IS WRONG
2992 014062 000342 342
2993
2994 014064 022700 100000 CMP #100000,%0 ;IS QUOTIENT = 100000
2995 014070 001403 BEQ +10
2996 014072 004767 002310 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2997 ;QUOTIENT IS WRONG
2998 014076 000343 343
2999
3000 014100 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
3001 014104 001403 BEQ +10
3002 014106 004767 002274 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3003 ;WRONG REMAINDER
3004 014112 000344 344
3005 014114 021527 000266 CMP (R5),#266
3006 014120 001403 BEQ +10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3007 014122 004767 002260 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3008 ;TEST IS IN WRONG SEQUENCE
3009 014126 000345 345
3010 014130 005215 INC (R5)
3011
    
```

```

3012
3013
3014
3015
3016 014132 010701
3017 014134 012702 037777
3018 014140 012703 077777
3019 014144 071227 077777
3020 014150 106737 000002
3021
3022 014154 122737 000000 000002
3023 014162 001403
3024 014164 004767 002216
3025
3026 014170 000346
3027
3028 014172 022702 077777
3029 014176 001403
3030 014200 004767 002202
3031
3032 014204 000347
3033
3034 014206 022703 077776
3035 014212 001403
3036 014214 004767 002166
3037
3038 014220 000350
3039 014222 021527 000267
3040 014226 001403
3041 014230 004767 002152
3042
3043 014234 000351
3044 014236 005215
3045

```

```

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

```

PS = 0

```

TST267: SCOPE
MOV #37777,%2 ;LOAD HIGH ORDER WITH 37777
MOV #77777,%2+1 ;LOAD LOW ORDER WITH 77777
DIV #77777,%2 ;DIVIDE BY #77777
MFPS @#PSWORD ;SAVE PS

CMPB #0,@#PSWORD ;IS PS = 0
BEQ +10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG
346

CMP #77777,%2 ;IS QUOTIENT = 77777
BEQ +10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;QUOTIENT IS WRONG
347

CMP #77776,%2+1 ;IS REMAINDER = 77776
BEQ +10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;WRONG REMAINDER
350

CMP (R5),#267
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
351

INC (R5)

```

```

3046
3047
3048
3049
3050 014240 010701
3051 014242 010501
3052 014244 012704 000000
3053 014250 012705 100000
3054 014254 071427 000002
3055 014260 106737 000002
3056
3057 014264 122737 000000 000002
3058 014272 001403
3059 014274 004767 002106
3060
3061 014300 000352
3062
3063 014302 022704 040000
3064 014306 001403
3065 014310 004767 002072
3066
3067 014314 000353
3068
3069 014316 022705 000000
3070 014322 001403
3071 014324 004767 002056
3072
3073 014330 000354
3074 014332 010105
3075 014334 021527 000270
3076 014340 001403
3077 014342 004767 002040
3078
3079 014346 000355
3080 014350 005215
3081

```

:*****
 :TEST:270 DIV 0 100000 / #2 = 40000 REM = 0 PS = 0
 :*****

TST270: SCOPE
 MOV R5,R1 ;SAVE R5
 MOV #0,%4 ;LOAD HIGH ORDER WITH 0
 MOV #100000,%4+1 ;LOAD LOW ORDER WITH 100000
 DIV #2,%4 ;DIVIDE BY #2
 MFPS @#PSWORD ;SAVE PS

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

352
 CMP #40000,%4 ;IS QUOTIENT = 40000
 BEQ .+10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;QUOTIENT IS WRONG

353
 CMP #0,%4+1 ;IS REMAINDER = 0
 BEQ .+10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;WRONG REMAINDER

354
 MOV R1,R5 ;RESTORE R5
 CMP (R5),#270
 BEQ .+10
 JSR PC,\$HLT ;IF IN WRONG SEQUENCE GO TO THE HLT
 ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;TEST IS IN WRONG SEQUENCE

355
 INC (R5)

```

3082
3083
3084
3085
3086 014352 010701
3087 014354 012700 177777
3088 014360 012701 077777
3089 014364 071027 177776
3090 014370 106737 000002
3091
3092 014374 122737 000000 000002
3093 014402 001403
3094 014404 004767 001776
3095
3096 014410 000356
3097
3098 014412 022700 040000
3099 014416 001403
3100 014420 004767 001762
3101
3102 014424 000357
3103
3104 014426 022701 177777
3105 014432 001403
3106 014434 004767 001746
3107
3108 014440 000360
3109 014442 021527 000271
3110 014446 001403
3111 014450 004767 001732
3112
3113 014454 000361
3114 014456 005215
3115

```

 :TEST:271 DIV 177777 77777 / #177776 = 40000 REM = 177777 PS = 0

TST271: SCOPE
 MOV #177777,%0 ;LOAD HIGH ORDER WITH 177777
 MOV #77777,%0+1 ;LOAD LOW ORDER WITH 77777
 DIV #177776,%0 ;DIVIDE BY #177776
 MFPS @#PSWORD ;SAVE PS
 CMPB #0,@#PSWORD ;IS PS = 0
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;PS IS WRONG
 356
 CMP #40000,%0 ;IS QUOTIENT = 40000
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;QUOTIENT IS WRONG
 357
 CMP #177777,%0+1 ;IS REMAINDER = 177777
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;WRONG REMAINDER
 360
 CMP (R5),#271
 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
 361
 INC (R5)


```

3116                                     :*****
3117                                     :TEST:272      DIV      0 52525 / #52525 = 1      REM = 0      PS = 0
3118                                     :*****
3119
3120 014460 010701 TST272: SCOPE
3121 014462 012702 000000 MOV      #0,%2      ;LOAD HIGH ORDER WITH 0
3122 014466 012703 052525 MOV      #52525,%2+1 ;LOAD LOW ORDER WITH 52525
3123 014472 071227 052525 DIV      #52525,%2 ;DIVIDE BY #52525
3124 014476 106737 000002 MFPS     @#PSWORD ;SAVE PS
3125
3126 014502 122737 000000 000002 CMPB    #0,@#PSWORD ;IS PS = 0
3127 014510 001403 BEQ     .+10
3128 014512 004767 001670 JSR     PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3129                                     ;PS IS WRONG
3130 014516 000362 362
3131
3132 014520 022702 000001 CMP     #1,%2      ;IS QUOTIENT = 1
3133 014524 001403 BEQ     .+10
3134 014526 004767 001654 JSR     PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3135                                     ;QUOTIENT IS WRONG
3136 014532 000363 363
3137
3138 014534 022703 000000 CMP     #0,%2+1 ;IS REMAINDER = 0
3139 014540 001403 BEQ     .+10
3140 014542 004767 001640 JSR     PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3141                                     ;WRONG REMAINDER
3142 014546 000364 364
3143 014550 021527 000272 CMP     (R5),#272
3144 014554 001403 BEQ     .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3145 014556 004767 001624 JSR     PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3146                                     ;TEST IS IN WRONG SEQUENCE
3147 014562 000365 365
3148 014564 005215 INC     (R5)
3149
    
```

```

3150 ;*****
3151 ;TEST:273 DIV 0 77777 / #0 = DUMMY REM = DUMMY PS = 3
3152 ;*****
3153
3154 014566 010701 TST273: SCOPE
3155 014570 010501 MOV R5,R1 ;SAVE R5
3156 014572 012704 000000 MOV #0,%4 ;LOAD HIGH ORDER WITH 0
3157 014576 012705 077777 MOV #77777,%4+1 ;LOAD LOW ORDER WITH 77777
3158 014602 071427 000000 DIV #0,%4 ;DIVIDE BY #0
3159 014606 106737 000002 MFPS @#PSWORD ;SAVE PS
3160 014612 042737 000014 000002 BIC #14,@#PSWORD
3161
3162 014620 122737 000003 000002 CMPB #3,@#PSWORD ;IS PS = 3
3163 014626 001403 BEQ +10
3164 014630 004767 001552 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3165 ;PS IS WRONG
3166 014634 000366 366
3167
3168 014636 010105 MOV R1,R5 ;RESTORE R5
3169 014640 021527 000273 CMP (R5),#273
3170 014644 001403 BEQ +10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3171 014646 004767 001534 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3172 ;TEST IS IN WRONG SEQUENCE
3173 014652 000367 367
3174 014654 005215 INC (R5)
3175

```

```

3176
3177
3178
3179
3180 014656 010701
3181 014660 012700 077777
3182 014664 012701 177777
3183 014670 071027 000002
3184 014674 106737 000002
3185 014700 042737 000014 000002
3186
3187 014706 122737 000002 000002
3188 014714 001403
3189 014716 004767 001464
3190
3191 014722 000370
3192
3193 014724 021527 000274
3194 014730 001403
3195 014732 004767 001450
3196
3197 014736 000371
3198 014740 005215
3199

```

```

:*****
:TEST:274 DIV 77777 177777 / #2 = DUMMY REM = DUMMY PS = 2
:*****
TST274: SCOPE
MOV #77777,%0 ;LOAD HIGH ORDER WITH 77777
MOV #177777,%0+1 ;LOAD LOW ORDER WITH 177777
DIV #2,%0 ;DIVIDE BY #2
MFPS @#PSWORD ;SAVE PS
BIC #14,@#PSWORD

CMPB #2,@#PSWORD ;IS PS = 2
BEQ +10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG

370

CMP (R5),#274
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

371
INC (R5)

```

DFKACA MACY11 27(732) 08-SEP-76 18:20 PAGE 90
 DFKACA.P11 DIV INSTRUCTION TESTS

3200	014742	012702	000002	MOV	#2,%2	
3201	014746	012703	000044	MOV	#S9,%3	
3202	014752	012704	000046	MOV	#S10,%4	
3203						
3204						
3205						*****
3206						:TEST:275 DIV 0 52525 / S9 = 25252 REM = 1 PS = 0
3207						*****
3208	014756	010701		TST275: SCOPE		
3209	014760	012700	000000	MOV	#0,%0	;LOAD HIGH ORDER WITH 0
3210	014764	012701	052525	MOV	#52525,%0+1	;LOAD LOW ORDER WITH 52525
3211	014770	071067	163050	DIV	S9,%0	;DIVIDE BY S9
3212	014774	106737	000002	MFPS	@#PSWORD	;SAVE PS
3213						
3214	015000	122737	000000 000002	CMPB	#0,@#PSWORD	;IS PS = 0
3215	015006	001403		BEQ	+10	
3216	015010	004767	001372	JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
3217						;PS IS WRONG
3218	015014	000372		372		
3219						
3220	015016	022700	025252	CMP	#25252,%0	;IS QUOTIENT = 25252
3221	015022	001403		BEQ	+10	
3222	015024	004767	001356	JSR	PC,\$HLT	;SEEN AN ERROR, GO TO T4 HALT ROUTINE
3223						;QUOTIENT IS WRONG
3224	015030	000373		373		
3225						
3226	015032	022701	000001	CMP	#1,%0+1	;IS REMAINDER = 1
3227	015036	001403		BEQ	+10	
3228	015040	004767	001342	JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
3229						;WRONG REMAINDER
3230	015044	000374		374		
3231	015046	021527	000275	CMP	(R5),#275	
3232	015052	001403		BEQ	+10	;IF IN WRONG SEQUENCE GO TO THE HLT
3233	015054	004767	001326	JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
3234						;TEST IS IN WRONG SEQUENCE
3235	015060	000375		375		
3236	015062	005215		INC	(R5)	
3237						

```

3238
3239
3240
3241
3242 015064 010701
3243 015066 012700 000000
3244 015072 012701 052525
3245 015076 071077 162744
3246 015102 106737 000002
3247
3248 015106 122737 000000 000002
3249 015114 001403
3250 015116 004767 001264
3251
3252 015122 000376
3253
3254 015124 022700 025252
3255 015130 001403
3256 015132 004767 001250
3257
3258 015136 000377
3259
3260 015140 022701 000001
3261 015144 001403
3262 015146 004767 001234
3263
3264 015152 000400
3265 015154 021527 000276
3266 015160 001403
3267 015162 004767 001220
3268
3269 015166 000401
3270 015170 005215
3271

```

;*****
 ;TEST:276 DIV 0 52525 / 3510 = 25252 REM = 1 PS = 0
 ;*****

TST276: SCOPE
 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
 DIV 3510,%0 ;DIVIDE BY 3510
 MFPS 3#PSWORD ;SAVE PS
 CMPB #0,3#PSWORD ;IS PS = 0
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;PS IS WRONG
 376
 CMP #25252,%0 ;IS QUOTIENT = 25252
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;QUOTIENT IS WRONG
 377
 CMP #1,%0+1 ;IS REMAINDER = 1
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;WRONG REMAINDER
 400
 CMP (R5),#276
 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
 401
 INC (R5)

3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305

015172 010701
015174 012700 000000
015200 012701 052525
015204 071037 000044
015210 106737 000002

015214 122737 000000 000002
015222 001403
015224 004767 001156

015230 000402

015232 022700 025252
015236 001403
015240 004767 001142

015244 000403

015246 022701 000001
015252 001403
015254 004767 001126

015260 000404
015262 021527 000277
015266 001403
015270 004767 001112

015274 000405
015276 005215

```
*****  
:TEST:277 DIV 0 52525 / 2#59 = 25252 REM = 1 , PS = 0  
*****  
TST277: SCOPE  
MOV #0,%0 ;LOAD HIGH ORDER WITH 0  
MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525  
DIV 2#59,%0 ;DIVIDE BY 2#59  
MFPS 2#PSWORD ;SAVE PS  
  
CMPB #0,2#PSWORD ;IS PS = 0  
BEQ +10  
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
;PS IS WRONG  
402  
  
CMP #25252,%0 ;IS QUOTIENT = 25252  
BEQ +10  
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
;QUOTIENT IS WRONG  
403  
  
CMP #1,%0+1 ;IS REMAINDER = 1  
BEQ +10  
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
;WRONG REMAINDER  
404  
CMP (R5),#277  
BEQ +10 ;IF IN WRONG SEQUENCE GO TO THE HLT  
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
;TEST IS IN WRONG SEQUENCE  
405  
INC (R5)
```

```

3306 :*****
3307 :TEST:300 DIV 0 52525 / %2 = 25252 REM = 1 PS = 0
3308 :*****
3309
3310 015300 010701 TST300: SCOPE
3311 015302 012700 000000 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
3312 015306 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
3313 015312 071002 DIV %2,%0 ;DIVIDE BY %2
3314 015314 106737 000002 MFPS @#PSWORD ;SAVE PS
3315
3316 015320 122737 000000 000002 CMPB #0,@#PSWORD ;IS PS = 0
3317 015326 001403 BEQ +10
3318 015330 004767 001052 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3319 ;PS IS WRONG
3320 015334 000406 406
3321
3322 015336 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
3323 015342 001403 BEQ +10
3324 015344 004767 001036 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3325 ;QUOTIENT IS WRONG
3326 015350 000407 407
3327
3328 015352 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
3329 015356 001403 BEQ +10
3330 015360 004767 001022 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3331 ;WRONG REMAINDER
3332 015364 000410 410
3333 015366 021527 000300 CMP (R5),#300
3334 015372 001403 BEQ +10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3335 015374 004767 001006 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3336 ;TEST IS IN WRONG SEQUENCE
3337 015400 000411 411
3338 015402 005215 INC (R5)
3339

```

```

3340 :*****
3341 :TEST:301 DIV @ 52525 / (3)+ = 25252 REM = 1 PS = 0
3342 :*****
3343
3344 015404 010701 TST301: SCOPE
3345 015406 012700 000000 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
3346 015412 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
3347 015416 071023 DIV (3)+,%0 ;DIVIDE BY (3)+
3348 015420 106737 000002 MFPS @#PSWORD ;SAVE PS
3349
3350 015424 122737 000000 000002 CMPB #0,@#PSWORD ;IS PS = 0
3351 015432 001403 BEQ +10
3352 015434 004767 000746 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3353 ;PS IS WRONG
3354 015440 000412 412
3355
3356 015442 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
3357 015446 001403 BEQ +10
3358 015450 004767 000732 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3359 ;QUOTIENT IS WRONG
3360 015454 000413 413
3361
3362 015456 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
3363 015462 001403 BEQ +10
3364 015464 004767 000716 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3365 ;WRONG REMAINDER
3366 015470 000414 414
3367 015472 021527 000301 CMP (R5),#301
3368 015476 001403 BEQ +10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3369 015500 004767 000702 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3370 ;TEST IS IN WRONG SEQUENCE
3371 015504 000415 415
3372 015506 005215 INC (R5)
3373
    
```


3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407

015510 010701
015512 012700 000000
015516 012701 052525
015522 071043
015524 106737 000002

015530 122737 000000 000002
015536 001403
015540 004767 000642

015544 000416

015546 022700 025252
015552 001403
015554 004767 000626

015560 000417

015562 022701 000001
015566 001403
015570 004767 000612

015574 000420
015576 021527 000302
015602 001403
015604 004767 000576

015610 000421
015612 005215

```
*****
:TEST:302 DIV 0 52525 / -(3) = 25252 REM = 1 PS = 0
*****
TST302: SCOPE
MOV #0,%0 ;LOAD HIGH ORDER WITH 0
MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
DIV -(3),%0 ;DIVIDE BY -(3)
MFPS @#PSWORD ;SAVE PS

CMPB #0,@#PSWORD ;IS PS = 0
BEQ +10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;PS IS WRONG
416

CMP #25252,%0 ;IS QUOTIENT = 25252
BEQ +10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;QUOTIENT IS WRONG
417

CMP #1,%0+1 ;IS REMAINDER = 1
BEQ +10
JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
;WRONG REMAINDER
420

CMP (R5),#302
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
421
INC (R5)
```

```

3408
3409
3410
3411
3412 015614 010701
3413 015616 012700 000000
3414 015622 012701 052525
3415 015626 071064 000002
3416 015632 106737 000002
3417
3418 015636 122737 000000 000002
3419 015644 001403
3420 015646 004767 000534
3421
3422 015652 000422
3423
3424 015654 022700 025252
3425 015660 001403
3426 015662 004767 000520
3427
3428 015666 000423
3429
3430 015670 022701 000001
3431 015674 001403
3432 015676 004767 000504
3433
3434 015702 000424
3435 015704 021527 000303
3436 015710 001403
3437 015712 004767 000470
3438
3439 015716 000425
3440 015720 005215
3441

```

 :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
 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
 JSR PC,\$HLT ;IF IN WRONG SEQUENCE GO TO THE HLT
 ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;TEST IS IN WRONG SEQUENCE
 425
 INC (R5)

```

3442
3443
3444
3445
3446 015722 010701
3447 015724 012700 000000
3448 015730 012701 052525
3449 015734 071074 000000
3450 015740 106737 000002
3451
3452 015744 122737 000000 000002
3453 015752 001403
3454 015754 004767 000426
3455
3456 015760 000426
3457
3458 015762 022700 025252
3459 015766 001403
3460 015770 004767 000412
3461
3462 015774 000427
3463
3464 015776 022701 000001
3465 016002 001403
3466 016004 004767 000376
3467
3468 016010 000430 430
3469 016012 021527 000304
3470 016016 001403
3471 016020 004767 000362
3472
3473 016024 000431 431
3474 016026 005215 INC (R5)
3475

```

 :TEST:304 DIV 0 52525 / 2(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 2(4),%0 ;DIVIDE BY 2(4)
 MFPS 2#PSWORD ;SAVE PS
 CMPB #0,2#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
 ;QUOTIENT IS WRONG
 427
 CMP #1,%0+1 ;IS REMAINDER = 1
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;WRONG REMAINDER
 430
 CMP (R5),#304
 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
 431
 INC (R5)

```

3476
3477
3478
3479
3480 016030 010701
3481 016032 012700 000000
3482 016036 012701 052525
3483 016042 071034
3484 016044 106737 000002
3485
3486 016050 122737 000000 000002
3487 016056 001403
3488 016060 004767 000322
3489
3490 016064 000432
3491
3492 016066 022700 025252
3493 016072 001403
3494 016074 004767 000306
3495
3496 016100 000433
3497
3498 016102 022701 000001
3499 016106 001403
3500 016110 004767 000272
3501
3502 016114 000434
3503 016116 021527 000305
3504 016122 001403
3505 016124 004767 000256
3506
3507 016130 000435
3508 016132 005215
3509

```

 :TEST:305 DIV 0 52525 / 2(4)+ = 25252 REM = 1 PS = 0

TST305: 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 2#PSWORD ;SAVE PS
 CMPB #0,2#PSWORD ;IS PS = 0
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;PS IS WRONG
 432
 CMP #25252,%0 ;IS QUOTIENT = 25252
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;QUOTIENT IS WRONG
 433
 CMP #1,%0+1 ;IS REMAINDER = 1
 BEQ +10
 JSR PC,\$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
 ;WRONG REMAINDER
 434
 CMP (R5),#305
 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
 435
 INC (R5)

```

3510 :*****
3511 :TEST:306 DIV 0 52525 / 2-(4) = 25252 REM = 1 PS = 0
3512 :*****
3513
3514 016134 010701 TST306: SCOPE
3515 016136 012700 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
3516 016142 012701 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
3517 016146 071054 DIV 2-(4),%0 ;DIVIDE BY 2-(4)
3518 016150 106737 MFPS 2#PSWORD ;SAVE PS
3519
3520 016154 122737 000000 000002 CMPB #0,2#PSWORD ;IS PS = 0
3521 016162 001403 BEQ +10
3522 016164 004767 000216 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3523 ;PS IS WRONG
3524 016170 000436 436
3525
3526 016172 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
3527 016176 001403 BEQ +10
3528 016200 004767 000202 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3529 ;QUOTIENT IS WRONG
3530 016204 000437 437
3531
3532 016206 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
3533 016212 001403 BEQ +10
3534 016214 004767 000166 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3535 ;WRONG REMAINDER
3536 016220 000440 440
3537 016222 021527 000306 CMP (R5),#306
3538 016226 001403 BEQ +10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3539 016230 004767 000152 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3540 ;TEST IS IN WRONG SEQUENCE
3541 016234 000441 441
3542 016236 005215 INC (R5)
3543

```

```

3544 ;*****
3545
3546 .SBTTL END OF PASS ROUTINE
3547
3548 ;*INCREMENT THE PASS NUMBER ($PASS)
3549 ;*TYPE "END PASS"
3550 ;*IF THERES A MONITOR GO TO IT
3551 ;*IF THERE ISN'T JUMP TO BEGIN
3552 ;*IF IT IS DESIRED TO HAVE A BELL INDICATE THE "END OF PASS" LOCATION
3553 ;*SENDMG CAN BE CHANGED TO 7.
3554
3555 $EOP:
3556 016240 010701 SCOPE
3557 016240 005267 INC $PASS ;; INCREMENT THE PASS NUMBER
3558 016246 042767 100000 000062 BIC #10000,$PASS ;; DON'T ALLOW A NEG. NUMBER
3559 016254 005327 DEC (PC)+ ;; LOOP?
3560 016256 000001 $EOPCT: .WORD 1
3561 016260 003024 BGT $DOAGN ;; YES
3562 016262 012737 MOV (PC)+,2(PC)+ ;; RESTORE COUNTER
3563 016264 000001 $ENDCT: .WORD 1
3564 016266 016256 $EOPCT
3565 016270 000004 016340 TYPE $SENDMG ;; TYPE "END PASS"
3566 016274 013700 000042 $GET42: MOV 2#42,RO ;; GET MONITOR ADDRESS
3567 016300 001414 BEQ $DOAGN ;; BRANCH IF NO MONITOR
3568 016302 022700 016322 CMP #SENDAD,RO ;; IS MONITOR ACT11?
3569 016306 001005 BNE SENDAD ;; NO--BRANCH
3570 016310 022760 177777 000002 CMP #-1,2(RO) ;; YES--IS THIS THE LAST PASS?
3571 016316 001005 BNE $DOAGN ;; NO--MAKE ANOTHER PASS
3572 016320 000005 RESET ;; CLEAR THE WORLD
3573 016322 004710 $SENDAD: JSR PC,(RO) ;; GO TO MONITOR
3574 016324 000240 NOP ;; SAVE ROOM
3575 016326 000240 NOP ;; FOR
3576 016330 000240 NOP ;; ACT11
3577 016332 000137 000600 $DOAGN: JMP 2#BEGIN ;; RETURN
3578 016336 000000 $PASS: .WORD 0 ;; NUMBER OF PASSES
3579 016340 035015 047105 020104 $SENDMG: .ASCII <15><12>/END PASS/
3580 016346 040520 051523
3581 016352 377 377 000 $ENULL: .BYTE -1,-1,0 ;; NULL CHARACTER STRING
3582 016356 .EVEN
3583
3584 016264 000400 $ENDCT: 400
3585
3586 ;*****
3587
3588 .SBTTL POWER FAIL ROUTINE
3589
3590
3591 016356 012737 016366 000024 $PWRDN: MOV #PWRUP,2#24
3592 016364 000000 HALT
3593
3594 016366 012706 000600 $PWRUP: MOV #BEGIN,SP ;; RESTORE THE SP
3595 016372 012737 016356 000024 MOV #PWRDN,2#24
3596 016400 060004 000072 TYPE POWER ;; GO AND TYPE "POWER"
3597 016404 000752 BR $DOAGN
    
```

```

3598
3599
3600
3601
3602
3603
3604
3605
U 3606 016406 017637 000000 000000 $HLT: MOV 2(SP), 2#$FATAL ;PLACE THE ERROR NUMBER AT LOCATION $FATAL
3607 016414 032777 020000 161430 BIT #20000, 2$SWR ;HAS THE OPERATOR ASKED TO SUPRESS ERROR TYPE CUTS
3608 016422 001046 BNE 6$
3609 016424 000004 000064 TYPE $SCLF ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
3610 016430 010046 MOV RO, -(SP) ;SAVE RO
3611 016432 112767 000002 161361 MOVVB #2, $TPCNT ;ALLOW TYPE OUTS OF PC AND ERROR NUMBER
3612 016440 016600 000002 MOV 2(SP), RO ;BRING THE RETURN PC IN RO
3613 016444 162700 000004 SUB #4, RO
3614 016450 112737 000006 000020 2$: MOVVB #6, 2$TYPCNT ;ALLOW TYPE OUT OF 6 DIGITS
3615 016456 005046 CLR -(SP)
3616 016460 000241 4$: CLC
3617 016462 006100 ROL RO
3618 016464 006116 ROL (SP) ;BRING THE C BIT FROM RO IN (SP)
3619 016466 052716 000060 BIS #60, (SP) ;PREPARE TO TYPE IT OUT
3620 016472 004767 000130 JSR PC, $TPCHR ;AND GO TO OUT PUT A CHARACTER
3621 016476 005016 CLR (SP)
3622 016500 006100 ROL RO
3623 016502 006116 ROL (SP)
3624 016504 006100 ROL RO
3625 016506 006116 ROL (SP)
3626 016510 105367 161304 DECB TYPCNT ;HAS ALL THE SIX CHARACTERS BEEN TYPED ?
3627 016514 001361 BNE 4$ ;IF NOT THEN REPEAT FROM 4$
3628 016516 005726 TST (SP)+ ;RESTORE STACK POINTER
3629 016520 017600 000002 MOV 22(SP), RO ;PREPARE TO OUT PUT THE ERROR NUMBER
3630 016524 000004 000066 TYPE $SCLF+2 ;GO AND TYPE 3 SPACES
3631 016530 105367 161265 DECB $TPCNT ;IF BOTH PC AND ERROR NUMBER HAS NOT BEEN
3632 016534 001345 BNE 2$ ;REPORTED THEN REPEAT FROM 2$
3633 016536 012600 MOV (SP)+, RO ;RESTORE RO
U 3634 016540 105767 161234 6$: TSTB $ENV ;IF WE ARE NOT UNDER APT. THEN GO TO
3635 016544 001403 BEQ 8$ ;8$
U 3636 016546 005237 000000 INC 2$MSGTY ;OTHERWISE INFORM APT. ABOUT SEEING THE ERROR
3637 016552 000777 BR ;AND LOOP
3638 016554 005777 161272 8$: TST 2$SWR ;IS IT REQUIRED TO HALT ON ERROR ?
3639 016560 100001 BPL 10$ ;IF NOT THEN GO TO 10$
3640 016562 000000 HALT
3641 016564 062716 000002 10$: ADD #2, (SP) ;ADJUST THE RETURN ADDRESS
3642 016570 000207 RTS PC ;AND RETURN

```

```

3643
3644
3645
3646
3647
3648
3649
3650
3651 016572 010046
3652 016574 017500 000002
3653 016600 112046
3654 016602 001005
3655 016604 005726
3656 016606 012600
3657 016610 062716 000002
3658 016614 000002
3659
3660 016616 004767 000004
3661 016622 005726
3662 016624 000765
3663
U 3664 016626 132737 000040 000000 $TPCHR: BITB #40, @#SENV
3665 016634 001006 BNE 4$
3666 016636 105777 161220 2$: TSTB @STPS
3667 016642 100375 BPL 2$
3668 016644 116677 000002 161206 MOV 2(SP), @STPB
3669 016652 000207 4$: RTS PC
3670 000001 .END

```

```

;*
;* TYPE OUT ROUTINE
;*

```

```

;*
;* THIS ROUTINE IS USED TO TYPE ASCIZ MESSAGES
;*

```

```

$TYPE: MOV R0, -(SP) ;SAVE R0
MOV @2(SP), R0 ;GET THE ADDRESS OF THE ASSCIZ STRING
2$: MOV (R0)+, -(SP) ;PUSH THE CHARACTER TO BE TYPED ONTO STACK
BNE 4$ ;BRANCH IF IT IS NOT THE TERMINATOR
TST (SP)+
MOV (SP)+, R0 ;OTHERWISE RESTORE THE STACK AND R0
3$: ADD #2, (SP) ;ADJUST THE RETURN PC
RTI ;AND RETURN

4$: JSR PC, $TPCHR ;GO TO TYPE A CHARACTER
TST (SP)+ ;RESTORE THE STACK POINTER
BR 2$ ;AND RETURN TO 2$

$TPCHR: BITB #40, @#SENV ;HAS THE CONSOLE OUTPUTS BEEN SUPPRESSED?
BNE 4$ ;IF SO THEN RETURN FROM THE SUBROUTINE VIA 4$
2$: TSTB @STPS ;IS THE PRINTER AVAILABLE?
BPL 2$ ;IF NOT THEN LOOP HERE
MOV 2(SP), @STPB ;OUT PUT THE CHARACTER
4$: RTS PC
.END

```


TST220	007246	1904#
TST221	007342	1833#
TST222	007436	1862#
TST223	007532	1891#
TST224	007632	1922#
TST225	007726	1951#
TST226	010022	1980#
TST227	010122	2011#
TST228	010216	2040#
TST229	010312	2069#
TST230	010406	2098#
TST231	010506	2129#
TST232	010602	2158#
TST233	010676	2187#
TST234	010772	2216#
TST235	011072	2247#
TST236	011166	2276#
TST237	011262	2305#
TST238	011376	2340#
TST239	011472	2369#
TST240	011566	2398#
TST241	011662	2427#
TST242	011754	2456#
TST243	012046	2485#
TST244	012140	2514#
TST245	012234	2543#
TST246	012330	2572#
TST247	012422	2601#
TST248	012514	2636#
TST249	012622	2670#
TST250	012730	2704#
TST251	013042	2740#
TST252	013150	2774#
TST253	013256	2808#
TST254	013370	2844#
TST255	013476	2878#
TST256	013604	2912#
TST257	013712	2946#
TST258	014024	2982#
TST259	014132	3016#
TST260	014240	3050#
TST261	014352	3086#
TST262	014460	3120#
TST263	014566	3154#
TST264	014656	3180#
TST265	014756	3208#
TST266	015064	3242#
TST267	015172	3276#
TST268	015300	3310#
TST269	015404	3344#
TST270	015510	3378#
TST271	015614	3412#
TST272	015722	3446#
TST273	016030	3480#
TST274	016134	3514#
TST275	002036	482

TST40	002072	663	669#												
TST41	002106	670	673#												
TST42	002124	674	677#												
TST43	002160	678	684#												
TST44	002212	685	690#												
TST45	002244	691	696#												
TST46	002276	697	702#												
TST47	002334	703	709#												
TST50	002364	710	715#												
TST51	002442	734#													
TST52	002522	759#													
TST53	002602	784#													
TST54	002662	809#													
TST55	002740	834#													
TST56	003016	859#													
TST57	003074	884#													
TST60	003154	909#													
TST61	003234	934#													
TST62	003312	959#													
TTYOUT	000056	393#													
TYPCNT	000020	377#	3614*	3626*											
TYPE =	000004	352#	3565	3596	3609	3630									
\$APTHD=	*****	U	362												
\$CRLF	000064	396#	3609	3630											
\$DEVCT=	*****	U	411												
\$DOAGN	016332	3561	3567	3571	3577#	3597									
\$ENDAD	016322	3568	3569	3573#											
\$ENDCT	016264	3563#	3584												
\$ENDMG	016340	3565	3579#												
\$ENULL	016352	3581#													
\$ENV =	*****	U	425	3634											
\$ENVM =	*****	U	3664												
\$EOP	016240	3555#													
\$EOPCT	016256	3560#	3564												
\$FATAL=	*****	U	3606#												
\$GET42	016274	3566#													
\$HD =	000003	326	327												
\$HLT	016406	459	466	494	501	528	535	562	569	597	604	633	640	722	
		740	746	765	771	790	796	815	821	840	846	865	871	890	
		896	915	921	940	946	965	971	1014	1020	1026	1032	1059	1065	
		1071	1076	1104	1110	1116	1121	1232	1250	1255	1260	1278	1283	1288	
		1307	1312	1318	1336	1341	1346	1364	1369	1374	1393	1398	1404	1422	
		1427	1432	1450	1455	1460	1465	1488	1493	1499	1519	1524	1530	1550	
		1555	1561	1581	1586	1592	1612	1617	1623	1643	1648	1654	1674	1679	
		1685	1705	1710	1716	1736	1741	1747	1767	1772	1778	1810	1815	1821	
		1839	1844	1850	1868	1873	1879	1898	1903	1909	1928	1933	1939	1957	
		1962	1968	1987	1992	1998	2017	2022	2028	2046	2051	2057	2075	2080	
		2086	2105	2110	2116	2135	2140	2146	2164	2169	2175	2193	2198	2204	
		2223	2228	2234	2253	2258	2264	2282	2287	2293	2312	2317	2323	2346	
		2351	2357	2375	2380	2386	2404	2409	2415	2433	2438	2444	2462	2467	
		2473	2491	2496	2502	2520	2525	2531	2549	2554	2560	2578	2583	2589	
		2607	2612	2618	2644	2650	2656	2661	2678	2684	2690	2695	2713	2719	
		2725	2731	2748	2754	2760	2765	2782	2788	2794	2799	2817	2823	2829	
		2835	2852	2858	2864	2869	2886	2892	2898	2903	2920	2926	2932	2937	
		2955	2961	2967	2973	2990	2996	3002	3007	3024	3030	3036	3041	3059	
		3065	3071	3077	3094	3100	3106	3111	3128	3134	3140	3145	3164	3171	

\$MAIL = *****	3189	3195	3216	3222	3228	3233	3250	3256	3262	3267	3284	3290	3296
\$MSGTY = *****	3301	3319	3324	3330	3335	3352	3358	3364	3369	3386	3392	3398	3403
\$PASS 016336	3420	3426	3432	3437	3454	3460	3466	3471	3488	3494	3500	3505	3522
\$PWRDN 016356	3528	3534	3539	3606#									
\$PWRUP 016366	413												
\$SETUP = 000020	3635*												
\$SSTUP = 177777	449	485	519	553	588	624	1005	1050	1095	3557*	3558*	3578#	
\$SWR = 160000	410	3591#	3595										
\$SWREG = *****	3591	3594#											
\$TESTN = *****	400#	3557											
\$TN = 000001	400#												
\$TPB 000060	326	327#	3550	3557	3566	3579							
\$TPCHR 016626	427												
\$TPCNT 000021	419												
\$TPS 000062	326#												
\$TYPE = 016572	394#	3668*											
.	3620	3660	3664#										
	378#	3611*	3631*										
	395#	3666											
	356	3651#											
	333#	336	355#	357#	362#	364#	366#	368#	370#	372#	374#	409#	417#
	458	464	493	499	527	533	561	567	596	602	632	638	721
	739	744	764	769	789	794	814	819	839	844	864	869	889
	894	914	919	939	944	964	969	1013	1019	1024	1031	1058	1064
	1069	1075	1103	1109	1114	1120	1231	1249	1254	1259	1277	1282	1287
	1306	1311	1317	1335	1340	1345	1363	1368	1373	1392	1397	1403	1421
	1426	1431	1449	1454	1459	1464	1487	1492	1497	1518	1523	1528	1549
	1554	1559	1580	1585	1590	1611	1616	1621	1642	1647	1652	1673	1678
	1683	1704	1709	1714	1735	1740	1745	1766	1771	1776	1809	1814	1819
	1838	1843	1848	1867	1872	1877	1897	1902	1907	1927	1932	1937	1956
	1961	1966	1986	1991	1996	2016	2021	2026	2045	2050	2055	2074	2079
	2084	2104	2109	2114	2134	2139	2144	2163	2168	2173	2192	2197	2202
	2222	2227	2232	2252	2257	2262	2281	2286	2291	2311	2316	2321	2345
	2350	2355	2374	2379	2384	2403	2408	2413	2432	2437	2442	2461	2466
	2471	2490	2495	2500	2519	2524	2529	2548	2553	2558	2577	2582	2587
	2606	2611	2616	2643	2649	2655	2660	2677	2683	2689	2694	2712	2718
	2724	2730	2747	2753	2759	2764	2781	2787	2793	2798	2816	2822	2828
	2834	2851	2857	2863	2868	2885	2891	2897	2902	2919	2925	2931	2936
	2954	2960	2966	2972	2989	2995	3001	3006	3023	3029	3035	3040	3058
	3064	3070	3076	3093	3099	3105	3110	3127	3133	3139	3144	3163	3170
	3188	3194	3215	3221	3227	3232	3249	3255	3261	3266	3283	3289	3295
	3300	3317	3323	3329	3334	3351	3357	3363	3368	3385	3391	3397	3402
	3419	3425	3431	3436	3453	3459	3465	3470	3487	3493	3504	3504	3521
	3527	3533	3538	3578	3582#	3584#	3585#	3637					

H09

DFKACA MACY11 27(732) 08-SEP-76 18:20 PAGE 113
DFKACA.P11 CROSS REFERENCE TABLE -- MACRO NAMES

.SEOP 314# 3544

DFKACA MACY11 27(732) 09-SEP-76 18:20 PAGE 120
 DFKACA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

.PAGE	309	310	330	357	409	434	730	780	830	880	930	980	1238	1294	1352
	1410	1476	1538	1600	1662	1724	1793	1829	1858	1887	1918	1947	1976	2007	2036
	2065	2094	2125	2154	2183	2212	2243	2272	2301	2332	2365	2394	2423	2452	2481
	2510	2539	2568	2597	2626	2666	2700	2736	2770	2804	2840	2874	2908	2942	2978
	3012	3046	3082	3116	3150	3176	3200	3238	3272	3306	3340	3374	3408	3442	3476
	3510	3544	3598	3643											
.REPT	1	336													
.SBTTL	409	434	980	1793	2626	3546	3589	3598	3643						
.TITLE	316														
.WORD	375	376	3560	3563	3578										

% ERRORS DETECTED: 14
 DEFAULT GLOBALS GENERATED: 0

*DFKACA,DFKACA.SEG/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:DFKACA.P11
 RUN-TIME: 33 39 6 SECONDS
 RUN-TIME RATIO: 137/75=1.7
 CORE USED: 10K (20 PAGES)

