

FP11

LDD/STD EXERCISER
MD-11-DCFPR-C

EP-DCFPR-C-DL
COPYRIGHT © 72-73
FICHE 1 OF 1

MAY 1978
digital
MADE IN USA



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

.REPT 2

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCFPR-C-D
PRODUCT NAME: FP11 LDD/STD EXERCISER
DATE CREATED: NOVEMBER 1973
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: BOB BRAIN

COPYRIGHT (C) 1972, 1973
DIGITAL EQUIPMENT CORPORATION

THIS MATERIAL IN THIS DOCUMENT IS FOR INFORMATIONAL PURPOSES ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OF SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY IT. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS WHICH MAY APPEAR IN THE DOCUMENT.

45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94

MAINDEC-11-DCFPR-C
TABLE OF CONTENTS

LDD/STD EXERCISER

PAGE 2

CONTENTS

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

95
96
97
98
99
100
101
102
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

MAINDEC-11-DCFPR-C
DESCRIPTION

LOD/STD EXERCISER

PAGE 3

1. ABSTRACT

THIS PROGRAM IS AN EXERCISER OF LOD/STD INSTRUCTIONS. IT USES RANDOM NUMBERS, FLOATING 1'S, AND FLOATING 2'S, AND CHECKS ALL MEMORY WITH LOD/STD INTO AN AC.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11/45 STANDARD COMPUTER WITH FP11 OPTION

2.2 STORAGE

PROGRAM STORAGE - THE ROUTINES USE MEMORY 2 - 17776

2.3 PRELIMINARY PROGRAMS

MAINDEC-11-DCFPA TO DCFPL

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

THE PROGRAM SHOULD ALWAYS BE STARTED AT 200.

4.3 PROGRAM AND/OR OPERATOR ACTION

- 1) LOAD PROGRAM INTO MEMORY USING ABS LOADER.
- 2) LOAD ADDRESS 200.
- 3) SET SWITCHES (SEE SEC 5.1.1) ALL DOWN FOR WORST CASE
- 4) PRESS START.

149
150
151
152

5) THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS
6) A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN,
7) THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION CCUNT IN
THE LEFT BYTE AND TEST NUMRER IN THE RIGHT. TO USE, SET THE

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
201
202
203
204
205
206

MAINDEC-11-DCFPR-C
DESCRIPTION

LDD/STD EXERCISER

PAGE 4

DATA DISPLAY SWITCH TO THE DISPLAY POSITION.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AT SA 200 .. ALL SWITCHES DOWN IS WORST CASE TESTING. IF AN ERROR OCCURS, THAT TEST WILL BE LOOPED UPON UNTIL COMPLETION OF 256 CONSECUTIVE PASSES WITH NO ERRORS OF THE SUBTEST IF SW<9> SET TO A 1. THE BELL WILL RING UPON COMPLETION OF A PASS.

5.1.1 SWITCH SETTINGS ARE:

SW<15> = 1 HALT ON ERROR
 SW<14> = 1 SCOPE LOOP
 SW<13> = 1 INHIBIT PRINTOUT
 SW<12> = 1 INHIBIT TRACE TRAPPING
 SW<11> = 1 INHIBIT ITERATIONS OF SUBTEST
 SW<10> = 1 BELL ON ERROR
 0 BELL ON PASS COMPLETE
 SW<09> = 1 LOOP ON ERROR
 SW<08> = 1 LOOP ON TEST IN SW<7:0>
 0 LOAD SW<7:0> INTO UB REGISTER

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED IN LOCATION "LAD". IF A SCOPE LOOP IS REQUESTED, THE CURRENT SUBTEST WILL BE LOOPED UPON. SW<11> ON A 1 INHIBITS ITERATION OF SUBTESTS. THE CONTENTS OF LAD MAY BE USED TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1.) IF A HLT IS EXECUTED, THE SUBTEST WILL BE LOOPED UPON UNTIL 256 CONSECUTIVE GOOD PASSES ARE COMPLETED IF SW<9> IS ON A 1. TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

237

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

MAINDEC-11-DCFPR-C
DESCRIPTION

LOD/STD EXERCISER

PAGE 5

5.2.3 TRTRAP

IF SWK12> IS ON A 0, THE T BIT WILL BE SET ON ALTERNATE PASSES. WHEN SET, IT CAUSES A TRAP AFTER EACH INSTRUCTION. THE FIRST INSTRUCTION EXECUTED UPON TRAPPING IS AN "RTY" WHICH RETURNS TO THE INTERRUPTED SEQUENCE OF INSTRUCTIONS. THIS SEQUENCE IS CONTINUED UNTIL THE END OF THE PROGRAM IS REACHED.

5.2.4 TRAPCATCHER

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.

5.2.5 FLOATING POINT TRAP (TO 244)

THE FP11 INTERRUPT DISABLE BIT IS ALWAYS SET IN ALL OF THESE TESTS SO NO TRAPS TO 244 SHOULD OCCUR. IF AN INTERRUPT OCCURS, THE PROGRAM WILL HALT AT 766 IN THE ROUTINE CALLED FLTERR AND DISPLAY THE FPS REGISTER IN R0.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ADR FPS ANS1 ANS2 ANS3 ANS4 ANS5 ANS6 ANS7 ANS8
FEC FEA

WHERE:

- ADR = ADDRESS OF ERROR HLT
- FPS = FLOATING POINT STATUS
- FEC = FLOATING EXCEPTION CODES (ERROR CODES)
- FEA = FLOATING EXCEPTION ADDRESS (ERROR ADDRESS)
- ANS1-8 = ERROR DATA READ FROM THE FP11. FROM 2-8 OF THESE MAY BE TYPED DEPENDING ON THE NUMBER FOLLOWING THE HLT; I.E., HLT+3 WOULD TYPE ANS1-ANS3. CHECK LISTING FOR MEANING OF THE DATA.

TO FIND THE FAILING TEST, LOOK AT THE LISTING ABOVE THE ADDRESS TYPED.

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

MAINDEC-11-DCFPR-C
DESCRIPTION

LDD/STD EXERCISER

PAGE 6

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

NCNE

8. MISCELLANEOUS

8.1 EXECUTION TIME

A BELL WILL RING WITHIN 15 SECONDS WITH ALL SWITCHES DOWN.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 600

8.3 POWER FAIL

THIS TEST 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 TYPE "POWER" AND CONTINUE TO RUN WITH NO OTHER TYPEOUTS.

9. PROGRAM DESCRIPTION

THIS PROGRAM TESTS ALL THE FP11 AC'S (AC0 - AC5) WITH RANDOM NUMBERS, FLOATING 1'S, AND FLOATING 0'S. IT ALSO TESTS LDF AND STF IN ALL MEMORY LOCATIONS FROM THE END OF THE PROGRAM TO THE BEGINNING OF THE LOADER. ONE RANDOM NUMBER IS GENERATED PER PASS AND LOADED INTO AC0 - AC5, SO EACH BELL SIGNIFIES THE COMPLETION OF ONE LOAD OF A RANDOM NUMBER, FLOATING 1'S, FLOATING 0'S, AND LDF/STF THROUGHOUT MEMORY.
.ENDR

.TITLE MAINDEC-11-DCFPR-C EXERCISER OF LDD AND STD
 ;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS
 ;PROGRAM BY BOB BRAIN
 .REM*

SWITCH	USE
8	0 = LOAD UB REGISTER WITH SWK7:0> 1 = LOOP ON TEST IN SWK7:0>
9	LOOP ON ERROR
10	0 = BELL ON PASS COMPLETE 1 = BELL ON ERROR
11	INHIBIT ITERATIONS
12	INHIBIT TRACE TRAP
13	INHIBIT ERROR TYPEOUTS
14	LOOP ON TEST
15	HALT ON ERROR

OUTPUT FORM:

ADR FPS ANS1 ANS2 ANS3 ANS4 ANS5 ANS6 ANS7 ANS8
 FEC FEA

BIT	FPS	REASON	CODE	FEC	ERROR
2		CARRY	0		ADDRESS ERROR
1		OVERFLOW	2		OPCODE ERROR
2		ZERO	4		DIVIDE BY ZERO
3		NEGATIVE	6		CONVERSION ERROR
4		MAINTAINANCE MODE	10		OVERFLOW
5		TRUNCATE MODE	12		UNDERFLOW
6		LONG INTEGER MODE	14		UNDEFINED VARIABLE (-0)
7		DOUBLE PRECISION MODE	16		UBREAK TRAP
8		INTERUPT ON CONVERSION ERROR			
9		INTERUPT ON OVERFLOW			
10		INTERUPT ON UNDERFLOW			
11		INTERUPT ON UNDEFINED VARIABLE			
12					
13					
14		INTERUPT DISABLE			
15		ERROR FLAG*			

170003		.ENABL	ABS		
000001		LDUB=	170003		
177776		N=	1		
177570		PS=	177776		
104400		SWR=	177570		
104000		SCOPE=	TRAP		
000007		HLT=	EMT		
000004		BELL=	7		
040000		TYPE=	ICT		
020000		SW14=	40000		
010000		SW13=	20000		
004000		SW12=	10000		
002000		SW11=	4000		
001000		SW10=	2000		
000400		SW09=	1000		
000000		SW08=	400		
000000		FPS=	%0		
000000		R0=	%0		
000001		R1=	%1		
000002		R2=	%2		
000003		R3=	%3		
000004		R4=	%4		
000005		TTY=	%5		
000005		R5=	%5		
000006		SP=	%6		
000007		PC=	%7		
000000		AC0=	%0		
000001		AC1=	%1		
000002		AC2=	%2		
000003		AC3=	%3		
000004		AC4=	%4		
000005		AC5=	%5		
000000		.	0		
000200	000167	000622	000		
			JMP	BEG	
			.		
			760		
000760	170200		FLTERR:	STFPS	FPS
000762	170367	000034		STST	FEC
000766	000000			HALT	
000770	000002			RTI	
			.		
			1000		
001000	000000		ICNT:	0	
001002	000000		ANS1:	0	
001004	000000		ANS2:	0	
001006	000000		ANS3:	0	
001010	000000		ANS4:	0	
001012	000000		ANS5:	0	
001014	000000		ANS6:	0	
001016	000000		ANS7:	0	
001020	000000		ANS8:	0	
001022	000000		FEC:	0	
001024	000000		FEA:	0	

;TRAP CATCHER FROM 0 - 776

;PUT FPS INTO R0
;STORE STATUS
;ERROR TRAP - SHOULD NOT BE ENABLED

;PASS COUNT - LH ITERATION COUNT - RH
;FIRST ANSWER (SEE CODE)

;FLOATING EXCEPTION CODES
;FLOATING EXECPTION ADDRESS

```

001026 012706 000600          BEG:  MCV      #600,SP          ; ** STACK AT 600 **
001032 162701 000006          SLB      #6,R1
001036 000425          BR        HAVIT

001040 012737 001106 000004 1S:   MCV      #NEM,0#4
001046 013701 000042          MCV      #042,R1          ;GET MONITOR ADDRESS
001052 001405          BEQ      25              ;SKIP IF 0
001054 162701 000020          SUB      #20,R1          ;DROP IT BACK
001060 022701 006042          CMP      #HICORE,R1     ;IS IT WITHIN THE PROGRAM?
001064 003012          BGT      HAVIT          ;NO - GOT THE ADDRESS
001066 012701 020000          2S:   MCV      #20000,R1
001072 005711          TRYAGN: TST      (1)          ;FIND MAX CORE
001074 062701 020000          ADD      #20000,R1
001100 022701 160000          CMP      #160000,R1
001104 001372          BNE      TRYAGN
001106 162701 000306          NEM:   SUB      #20000-17472,R1 ;MOVE IT TO EXISTANT CORE
001112 010167 004722          HAVIT: MCV      R1,UPLIM     ;SAVE IT
001116 012767 001140 176660          MCV      #BEGIN-6,4      ;FIND OUT WHICH MACHINE THIS IS
001124 005737 177772          TST      #177772        ;TEST FOR PIRQ
001130 012767 000006 003316          MCV      #6,YESRT       ;FUDGE IN RTY IF 11/45
001136 000403          BR
001140 016737 004642 000010          MCV      FPTADR,0#10     ;LOAD THE ILLEGAL INSTR VECTOR
                                ; WITH THE ADDRESS OF THE FPU,
                                ; THE FPU WILL HANDLE THE BAD OPCODES

001146 012706 000600          BEGIN: MCV      #600,SP          ; ** STACK AT 600 **
001152 012737 000006 000004          MCV      #6,0#4          ;RESET 4
001160 012737 004454 000014          MCV      #YESRT,0#14     ;SET TRACE TRAP VECTOR
001166 012777 005506 004620          MCV      #POWDN,0DWNVEC ;POWER DOWN VECTOR
001174 012777 000340 004614          MCV      #340,0DWNVEC+2
001202 012737 005706 000020          MCV      #,IOT,0#20      ;SET UP VECTOR 20
001210 012700 000030          MCV      #30,R0          ;SET R0 TO VECTOR 30
001214 012720 004616          MCV      #,TRP,(0)+      ;SET TRAP VECTOR
001220 012720 000340          MCV      #340,(0)+
001224 012720 004456          MCV      #,EMT,(0)+      ;SET EMT VECTOR
001230 012710 000340          MCV      #340,(0)
001234 012777 000760 004546          MCV      #FLTERR,0FPVECT ;LOAD INTERRUPT VECTOR
001242 012777 000340 004542          MCV      #340,0FPVECT+2 ;LOCK UP PROCESSOR
001250 005067 177524          CLR      ICNT
001254 005067 004550          CLR      LAD
001260 170127 000000          LDFPS   #0

001264 004767 003622          JSR      7,RAND4          ;GET 4 RANDOM NUMBERS INTO ANS1 - ANS4
    
```

.....
 ;TEST 1 USES AC0
 ;.....

001270	104400			SCOPE		
001272	170011			SETD		;SET DOUBLE MODE
001274	172437	001002		LDD	@ANS1,0	;LOAD INTO AC2
001300	174037	001012		STD	0,@ANS5	;STORE INTO ANS5
001304	023737	001002	001012	CMP	@ANS1,@ANS5	;FIRST WORD OK?
001312	001401			BEO	,+4	
001314	104010			HLT+8.		;FIRST WORD WRONG
001316	023737	001004	001014	CMP	@ANS2,@ANS6	;SECOND WORD OK?
001324	001401			BEO	,+4	
001326	104010			HLT+8.		;SECOND WORD WRONG
001330	023737	001006	001016	CMP	@ANS3,@ANS7	;THIRD WORD OK?
001336	001401			BEO	,+4	
001340	104010			HLT+8.		;THIRD WORD WRONG
001342	023737	001010	001020	CMP	@ANS4,@ANS8	;FOURTH WORD OK?
001350	001401			BEO	,+4	
001352	104010			HLT+8.		;FOURTH WORD WRONG

.....
 ;TEST 2 USES AC1
 ;.....

001354	104400			SCOPE		
001356	170011			SETD		;SET DOUBLE MODE
001360	172537	001002		LDD	@ANS1,1	;LOAD INTO AC1
001364	174137	001012		STD	1,@ANS5	;STORE INTO ANS5
001370	023737	001002	001012	CMP	@ANS1,@ANS5	;FIRST WORD OK?
001376	001401			BEO	,+4	
001400	104010			HLT+8.		;FIRST WORD WRONG
001402	023737	001004	001014	CMP	@ANS2,@ANS6	;SECOND WORD OK?
001410	001401			BEO	,+4	
001412	104010			HLT+8.		;SECOND WORD WRONG
001414	023737	001006	001016	CMP	@ANS3,@ANS7	;THIRD WORD OK?
001422	001401			BEO	,+4	
001424	104010			HLT+8.		;THIRD WORD WRONG
001426	023737	001010	001020	CMP	@ANS4,@ANS8	;FOURTH WORD OK?
001434	001401			BEO	,+4	
001436	104010			HLT+8.		;FOURTH WORD WRONG

.....
 ;TEST 3 USES AC2
 ;.....

001440	104400			SCOPE	
001442	170011			SETD	;SET DOUBLE MODE
001444	172637	001002		LDD	;LOAD INTO AC2
001450	174237	001012		STD	;STORE INTO ANS5
001454	023737	001002	001012	CMP	;FIRST WORD OK?
001462	001401			BEQ	,+4
001464	104010			HLT+B.	;FIRST WORD WRONG
001466	023737	001004	001014	CMP	;SECOND WORD OK?
001474	001401			BEQ	,+4
001476	104010			HLT+B.	;SECOND WORD WRONG
001500	023737	001006	001016	CMP	;THIRD WORD OK?
001506	001401			BEQ	,+4
001510	104010			HLT+B.	;THIRD WORD WRONG
001512	023737	001010	001020	CMP	;FOURTH WORD OK?
001520	001401			BEQ	,+4
001522	104010			HLT+B.	;FOURTH WORD WRONG

.....
 ;TEST 4 USES AC3
 ;.....

001524	104400			SCOPE	
001526	170011			SETD	;SET DOUBLE MODE
001530	172737	001002		LDD	;LOAD INTO AC3
001534	174337	001012		STD	;STORE INTO ANS5
001540	023737	001002	001012	CMP	;FIRST WORD OK?
001546	001401			BEQ	,+4
001550	104010			HLT+B.	;FIRST WORD WRONG
001552	023737	001004	001014	CMP	;SECOND WORD OK?
001560	001401			BEQ	,+4
001562	104010			HLT+B.	;SECOND WORD WRONG
001564	023737	001006	001016	CMP	;THIRD WORD OK?
001572	001401			BEQ	,+4
001574	104010			HLT+B.	;THIRD WORD WRONG
001576	023737	001010	001020	CMP	;FOURTH WORD OK?
001604	001401			BEQ	,+4
001606	104010			HLT+B.	;FOURTH WORD WRONG

.....
 ;TEST 5 USES AC4
 ;.....

001610	104400			SCOPE		
001612	170011			SETD		;SET DOUBLE MODE
001614	174004			STD	0,AC4	;PUT INTO AC4
001616	172404			LDD	AC4,0	;BACK INTO AC0
001620	174037	001012		STD	0,0#ANS5	;NOW INTI ANS5
001624	023737	001002	001012	CMP	0#ANS1,0#ANS5	;FIRST WORD OK?
001632	001401			BEQ	,+4	
001634	104010			HLT+8.		;FIRST WORD WRONG
001636	023737	001004	001014	CMP	0#ANS2,0#ANS6	;SECOND WORD OK?
001644	001401			BEQ	,+4	
001646	104010			HLT+8.		;SECOND WORD WRONG
001650	023737	001006	001016	CMP	0#ANS3,0#ANS7	;THIRD WORD OK?
001656	001401			BEQ	,+4	
001660	104010			HLT+8.		;THIRD WORD WRONG
001662	023737	001010	001020	CMP	0#ANS4,0#ANS8	;FOURTH WORD OK?
001670	001401			BEQ	,+4	
001672	104010			HLT+8.		;FOURTH WORD WRONG

.....
 ;TEST 6 USES AC5
 ;.....

001674	104400			SCOPE		
001676	170011			SETD		;SET DOUBLE MODE
001700	174005			STD	0,AC5	;PUT INTO AC5
001702	172405			LDD	AC5,0	;BACK INTO AC0
001704	174037	001012		STD	0,0#ANS5	;NOW INTI ANS5
001710	023737	001002	001012	CMP	0#ANS1,0#ANS5	;FIRST WORD OK?
001716	001401			BEQ	,+4	
001720	104010			HLT+8.		;FIRST WORD WRONG
001722	023737	001004	001014	CMP	0#ANS2,0#ANS6	;SECOND WORD OK?
001730	001401			BEQ	,+4	
001732	104010			HLT+8.		;SECOND WORD WRONG
001734	023737	001006	001016	CMP	0#ANS3,0#ANS7	;THIRD WORD OK?
001742	001401			BEQ	,+4	
001744	104010			HLT+8.		;THIRD WORD WRONG
001746	023737	001010	001020	CMP	0#ANS4,0#ANS8	;FOURTH WORD OK?
001754	001401			BEQ	,+4	
001756	104010			HLT+8.		;FOURTH WORD WRONG

.....
 ;TEST 7 FLOATING 1'S TEST IN AC0
 ;.....

001760	104400		SCOPE		
001762	170011		SETD		
001764	012701	001002	MOV	#ANS1,R1	;LOAD FIRST WORD
001770	012721	000001	MOV	#1,(1)+	;INITIALIZE TO 1
001774	012721	000001	MOV	#1,(1)+	;INITIALIZE TO 1
002000	012721	000001	MOV	#1,(1)+	;INITIALIZE TO 1
002004	012721	000001	MOV	#1,(1)+	;INITIALIZE TO 1
002010	012767	000020	MOV	#16,,COUNT	;SET UP COUNT
002016	172467	176760	LDD	ANS1,0	;LOAD AC0
002022	174067	176764	STD	0,ANS5	;STORE RESULTS
002026	012702	001002	MOV	#ANS1,R2	;SET UP FIRST WORD
002032	022122		CMP	(1)+,(2)+	;IS DATA OK?
002034	001401		BEQ	.+4	
002036	104010		HLT+0.		;FIRST WORD IS WRONG
002040	022122		CMP	(1)+,(2)+	;IS DATA OK?
002042	001401		BEQ	.+4	
002044	104010		HLT+0.		;SECOND WORD IS WRONG
002046	022122		CMP	(1)+,(2)+	;IS DATA OK?
002050	001401		BEQ	.+4	
002052	104010		HLT+0.		;THIRD WORD IS WRONG
002054	022122		CMP	(1)+,(2)+	;IS DATA OK?
002056	001401		BEQ	.+4	
002060	104010		HLT+0.		;LAST WORD IS WRONG
002062	012701	001002	MOV	#ANS1,R1	
002066	006321		ASL	(1)+	;SHIFT ALL BY 1
002070	006321		ASL	(1)+	
002072	006321		ASL	(1)+	
002074	006321		ASL	(1)+	
002076	005367	003734	DEC	COUNT	
002102	001345		BNE	LP7	
002104	105067	176671	CLRB	ICNT+1	

.....
 ;TEST 10 FLOATING 1'S TEST IN AC1
 ;.....

002110	104400		SCOPE		
002112	170011		SETD		
002114	012701	001002	MOV	#ANS1,R1	;LOAD FIRST WORD
002120	012721	000001	MOV	#1,(1)+	;INITIALIZE TO 1
002124	012721	000001	MOV	#1,(1)+	;INITIALIZE TO 1
002130	012721	000001	MOV	#1,(1)+	;INITIALIZE TO 1
002134	012721	000001	MOV	#1,(1)+	;INITIALIZE TO 1
002140	012767	000020	MOV	#16,COUNT	;SET UP COUNT
002146	172567	176630	LDD	ANS1,1	;LOAD AC1
002152	174167	176634	STD	1,ANS5	;STORE RESULTS
002156	012702	001002	MOV	#ANS1,R2	;SET UP FIRST WORD
002162	022122		CMP	(1)+,(2)+	;IS DATA OK?
002164	001401		BEO	.+4	
002166	104010		HLT+0.		;FIRST WORD IS WRONG
002170	022122		CMP	(1)+,(2)+	;IS DATA OK?
002172	001401		BEO	.+4	
002174	104010		HLT+0.		;SECOND WORD IS WRONG
002176	022122		CMP	(1)+,(2)+	;IS DATA OK?
002200	001401		BEO	.+4	
002202	104010		HLT+0.		;THIRD WORD IS WRONG
002204	022122		CMP	(1)+,(2)+	;IS DATA OK?
002206	001401		BEO	.+4	
002210	104010		HLT+0.		;LAST WORD IS WRONG
002212	012701	001002	MOV	#ANS1,R1	
002216	006321		ASL	(1)+	;SHIFT ALL BY 1
002220	006321		ASL	(1)+	
002222	006321		ASL	(1)+	
002224	006321		ASL	(1)+	
002226	005367	003604	DEC	COUNT	
002232	001345		BNE	LP10	
002234	105067	176541	CLRB	ICNT+1	

.....
 ;TEST 11 FLOATING 1'S TEST IN AC2
 ;.....

002240	104400		SCOPE		
002242	170011		SETD		
002244	012701	001002	MOV	#ANS1,R1	;LOAD FIRST WORD
002250	012721	000001	MOV	#1,(1)+	;INITIALIZE TO 1
002254	012721	000001	MOV	#1,(1)+	;INITIALIZE TO 1
002260	012721	000001	MOV	#1,(1)+	;INITIALIZE TO 1
002264	012721	000001	MOV	#1,(1)+	;INITIALIZE TO 1
002270	012767	000020	MOV	#16,COUNT	;SET UP COUNT
002276	172667	176500	LDD	ANS1,2	;LOAD AC2
002302	174267	176504	STD	2,ANS5	;STORE RESULTS
002306	012702	001002	MOV	#ANS1,R2	;SET UP FIRST WORD
002312	022122		CMP	(1)+,(2)+	;IS DATA OK?
002314	001401		BEO	,+4	
002316	104010		HLT+8.		;FIRST WORD IS WRONG
002320	022122		CMP	(1)+,(2)+	;IS DATA OK?
002322	001401		BEO	,+4	
002324	104010		HLT+8.		;SECOND WORD IS WRONG
002326	022122		CMP	(1)+,(2)+	;IS DATA OK?
002330	001401		BEO	,+4	
002332	104010		HLT+8.		;THIRD WORD IS WRONG
002334	022122		CMP	(1)+,(2)+	;IS DATA OK?
002336	001401		BEO	,+4	
002340	104010		HLT+8.		;LAST WORD IS WRONG
002342	012701	001002	MOV	#ANS1,R1	
002346	006321		ASL	(1)+	;SHIFT ALL BY 1
002350	006321		ASL	(1)+	
002352	006321		ASL	(1)+	
002354	006321		ASL	(1)+	
002356	005367	003454	DEC	COUNT	
002362	001345		BNE	LP11	
002364	105067	176411	CLRB	ICNT+1	

003540 LP11:

.....
 ;TEST 12 FLOATING 1'S TEST IN AC3
 ;.....

0J2370	104400			SCOPE		
0J2372	173011			SETD		
0J2374	012701	001002		MOV	#ANS1,R1	;LOAD FIRST WORD
0J2400	012721	000001		MOV	#1,(1)+	;INITIALIZE TO 1
0J2404	012721	000001		MOV	#1,(1)+	;INITIALIZE TO 1
0J2410	012721	000001		MOV	#1,(1)+	;INITIALIZE TO 1
0J2414	012721	000001		MOV	#1,(1)+	;INITIALIZE TO 1
0J2420	012767	000020	003410	MOV	#16,COUNT	;SET UP COUNT
0J2426	172767	176350		LDD	ANS1,3	;LOAD AC3
0J2432	174367	176354	LP12:	STD	3,ANS5	;STORE RESULTS
0J2436	012702	001002		MOV	#ANS1,R2	;SET UP FIRST WORD
0J2442	022122			CMF	(1)+,(2)+	;IS DATA OK?
0J2444	001401			BEQ	,+4	
0J2446	104010			HLT+8.		;FIRST WORD IS WRONG
0J2450	022122			CMF	(1)+,(2)+	;IS DATA OK?
0J2452	001401			BEQ	,+4	
0J2454	104010			HLT+8.		;SECOND WORD IS WRONG
0J2456	022122			CMF	(1)+,(2)+	;IS DATA OK?
0J2460	001401			BEQ	,+4	
0J2462	104010			HLT+8.		;THIRD WORD IS WRONG
0J2464	022122			CMF	(1)+,(2)+	;IS DATA OK?
0J2466	001401			BEQ	,+4	
0J2470	104010			HLT+8.		;LAST WORD IS WRONG
0J2472	012701	001002		MOV	#ANS1,R1	
0J2476	006321			ASL	(1)+	;SHIFT ALL BY 1
0J2500	006321			ASL	(1)+	
0J2502	006321			ASL	(1)+	
0J2504	006321			ASL	(1)+	
0J2506	005367	003324		DEC	COUNT	
0J2512	001345			BNE	LP12	
0J2514	105067	176261		CLRB	ICNT+1	

.....
 ;TEST 13 FLOATING 1'S TEST IN AC4
 ;.....

002520	104400			SCOPE		
002522	170011			SETD		
002524	012701	001002		MOV	#ANS1,R1	;LOAD FIRST WORD
002530	012721	000001		MOV	#1,(1)+	;INITIALIZE TO 1
002534	012721	000001		MOV	#1,(1)+	;INITIALIZE TO 1
002540	012721	000001		MOV	#1,(1)+	;INITIALIZE TO 1
002544	012721	000001		MOV	#1,(1)+	;INITIALIZE TO 1
002550	012767	000020	003260	MOV	#16,COUNT	;SET UP COUNT
002556	172437	001002	LP13:	LDD	#ANS1,0	;LOAD AC0
002562	174004			STD	0,AC4	;PUT INTO AC4
002564	172404			LDD	AC4,0	;BACK INTO AC0
002566	174037	001012		STD	0,#ANS5	;NOW INTO ANS5
002572	012702	001002		MOV	#ANS1,R2	;SET UP FIRST WORD
002576	022122			CMP	(1)+,(2)+	;IS DATA OK?
002600	001401			BEQ	.+4	
002602	104010			HLT+0.		;FIRST WORD IS WRONG
002604	022122			CMP	(1)+,(2)+	;IS DATA OK?
002606	001401			BEQ	.+4	
002610	104010			HLT+0.		;SECOND WORD IS WRONG
002612	022122			CMP	(1)+,(2)+	;IS DATA OK?
002614	001401			BEQ	.+4	
002616	104010			HLT+0.		;THIRD WORD IS WRONG
002620	022122			CMP	(1)+,(2)+	;IS DATA OK?
002622	001401			BEQ	.+4	
002624	104010			HLT+0.		;LAST WORD IS WRONG
002626	012701	001002		MOV	#ANS1,R1	
002632	006321			ASL	(1)+	;SHIFT ALL BY 1
002634	006321			ASL	(1)+	
002636	006321			ASL	(1)+	
002640	006321			ASL	(1)+	
002642	005367	003170		DEC	COUNT	
002646	001343			BNE	LP13	
002650	105067	176125		CLRB	ICNT+1	

.....
 ;TEST 14 FLOATING 1'S TEST IN AC5
 ;.....

032654	104400			SCOPE		
032656	170011			SETC		
032660	012701	001002		MOV	#ANS1,R1	;LOAD FIRST WORD
032664	012721	000001		MOV	#1,(1)+	;INITIALIZE TO 1
032670	012721	000001		MOV	#1,(1)+	;INITIALIZE TO 1
032674	012721	000001		MOV	#1,(1)+	;INITIALIZE TO 1
032700	012721	000001		MOV	#1,(1)+	;INITIALIZE TO 1
032704	012767	000020	003124	MOV	#16.,COUNT	;SET UP COUNT
032712	172437	001002	LP14:	LDD	#ANS1,2	;LOAD AC2
032716	174005			STD	0,AC5	;PUT INTO AC5
032720	172405			LDD	AC5,2	;BACK INTO AC2
032722	174037	001012		STD	0,#ANS5	;NOW INTO ANS5
032726	012702	001002		MOV	#ANS1,R2	;SET UP FIRST WORD
032732	022122			CMP	(1)+,(2)+	;IS DATA OK?
032734	001401			BEQ	.+4	
032736	104010			HLT+8.		;FIRST WORD IS WRONG
032740	022122			CMP	(1)+,(2)+	;IS DATA OK?
032742	001401			BEQ	.+4	
032744	104010			HLT+8.		;SECOND WORD IS WRONG
032746	022122			CMP	(1)+,(2)+	;IS DATA OK?
032750	001401			BEQ	.+4	
032752	104010			HLT+8.		;THIRD WORD IS WRONG
032754	022122			CMP	(1)+,(2)+	;IS DATA OK?
032756	001401			BEQ	.+4	
032760	104010			HLT+8.		;LAST WORD IS WRONG
032762	012701	001002		MOV	#ANS1,R1	
032766	006321			ASL	(1)+	;SHIFT ALL BY 1
032770	006321			ASL	(1)+	
032772	006321			ASL	(1)+	
032774	006321			ASL	(1)+	
032776	005367	003034		DEC	COUNT	
033002	001343			BNE	LP14	
033004	105067	175771		CLRB	ICNT+1	

.....
 ;TEST 15 FLOATING 0'S TEST IN ACC
 ;.....

003010	104400		SCOPE		
003012	170011		SETD		
003014	012701	001002	MOV	#ANS1,R1	;LOAD FIRST WORD
003020	012721	177776	MOV	#-2,(1)+	;INITIALIZE TO 177776
003024	012721	177776	MOV	#-2,(1)+	;INITIALIZE TO 177776
003030	012721	177776	MOV	#-2,(1)+	;INITIALIZE TO 177776
003034	012721	177776	MOV	#-2,(1)+	;INITIALIZE TO 177776
003040	012767	000020	MOV	#16,,COUNT	;SET UP COUNT
003046	172467	175730	LDD	ANS1,0	;LOAD ACC
003052	174067	175734	STD	0,ANS5	;STORE RESULTS
003056	012702	001002	MOV	#ANS1,R2	;SET UP FIRST WORD
003062	022122		CMP	(1)+,(2)+	;IS DATA OK?
003064	001401		BEQ	,+4	
003066	104010		HLT+0.		;FIRST WORD IS WRONG
003070	022122		CMP	(1)+,(2)+	;IS DATA OK?
003072	001401		BEQ	,+4	
003074	104010		HLT+0.		;SECOND WORD IS WRONG
003076	022122		CMP	(1)+,(2)+	;IS DATA OK?
003100	001401		BEQ	,+4	
003102	104010		HLT+0.		;THIRD WORD IS WRONG
003104	022122		CMP	(1)+,(2)+	;IS DATA OK?
003106	001401		BEQ	,+4	
003110	104010		HLT+0.		;LAST WORD IS WRONG
003112	012701	001002	MOV	#ANS1,R1	
003116	000241		CLC		;SHIFT ALL BY 1 (FLOAT 0)
003120	006121		RCL	(1)+	
003122	000241		CLC		
003124	006121		RCL	(1)+	
003126	000241		CLC		
003130	006121		RCL	(1)+	
003132	000241		CLC		
003134	006121		RCL	(1)+	
003136	005367	002674	DEC	COUNT	
003142	001341		BNE	LP15	
003144	105067	175631	CLRB	ICNT+1	

002770 LP15:

.....
 ;TEST 16 FLOATING 3'S TEST IN AC1
 ;.....

003150	104400			SCOPE		
003152	173011			SETC		
003154	012701	001002		MOV	#ANS1,R1	:LOAD FIRST WORD
003160	012721	177776		MOV	#-2,(1)+	:INITIALIZE TO 177776
003164	012721	177776		MOV	#-2,(1)+	:INITIALIZE TO 177776
003170	012721	177776		MOV	#-2,(1)+	:INITIALIZE TO 177776
003174	012721	177776		MOV	#-2,(1)+	:INITIALIZE TO 177776
003200	012767	000020	002630	MOV	#16,,COUNT	:SET UP COUNT
003206	172567	175570	LP161	LDD	ANS1,1	:LOAD AC1
003212	174167	175574		STD	1,ANS5	:STORE RESULTS
003216	012702	001002		MOV	#ANS1,R2	:SET UP FIRST WORD
003222	022122			COMP	(1)+,(2)+	:IS DATA OK?
003224	001401			BEQ	,+4	
003226	104010			HLT+B.		:FIRST WORD IS WRONG
003230	022122			COMP	(1)+,(2)+	:IS DATA OK?
003232	001401			BEQ	,+4	
003234	104010			HLT+B.		:SECOND WORD IS WRONG
003236	022122			COMP	(1)+,(2)+	:IS DATA OK?
003240	001401			BEQ	,+4	
003242	104010			HLT+B.		:THIRD WORD IS WRONG
003244	022122			COMP	(1)+,(2)+	:IS DATA OK?
003246	001401			BEQ	,+4	
003250	104010			HLT+B.		:LAST WORD IS WRONG
003252	012701	001002		MOV	#ANS1,R1	
003256	000241			CLC		:SHIFT ALL BY 1 (FLOAT 0)
003260	006121			ROL	(1)+	
003262	000241			CLC		
003264	006121			ROL	(1)+	
003266	000241			CLC		
003270	006121			ROL	(1)+	
003272	000241			CLC		
003274	006121			ROL	(1)+	
003276	005367	002534		DEC	COUNT	
003302	001341			BNE	LP16	
003304	105067	175471		CLRB	ICNT+1	

.....
 ;TEST 17 FLOATING 0'S TEST IN AC2
 ;.....

003310	104400		SCOPE		
003312	170011		SETD		
003314	012701	001002	MOV	#ANS1,R1	;LOAD FIRST WORD
003320	012721	177776	MOV	#-2,(1)+	;INITIALIZE TO 177776
003324	012721	177776	MOV	#-2,(1)+	;INITIALIZE TO 177776
003330	012721	177776	MOV	#-2,(1)+	;INITIALIZE TO 177776
003334	012721	177776	MOV	#-2,(1)+	;INITIALIZE TO 177776
003340	012767	000020	002470	MOV	#16,COUNT
			LP17:		
003346	172667	175430	LDD	ANS1,2	;LOAD AC2
003352	174267	175434	STD	2,ANS5	;STORE RESULTS
003356	012702	001002	MOV	#ANS1,R2	;SET UP FIRST WORD
003362	022122		CMP	(1)+,(2)+	;IS DATA OK?
003364	001401		BEQ	,+4	
003366	104010		HLT+0.		;FIRST WORD IS WRONG
003370	022122		CMP	(1)+,(2)+	;IS DATA OK?
003372	001401		BEQ	,+4	
003374	104010		HLT+0.		;SECOND WORD IS WRONG
003376	022122		CMP	(1)+,(2)+	;IS DATA OK?
003400	001401		BEQ	,+4	
003402	104010		HLT+0.		;THIRD WORD IS WRONG
003404	022122		CMP	(1)+,(2)+	;IS DATA OK?
003406	001401		BEQ	,+4	
003410	104010		HLT+0.		;LAST WORD IS WRONG
003412	012701	001002	MOV	#ANS1,R1	
003416	000241		CLC		;SHIFT ALL BY 1 (FLOAT 0)
003420	006121		ROL	(1)+	
003422	000241		CLC		
003424	006121		ROL	(1)+	
003426	000241		CLC		
003430	006121		ROL	(1)+	
003432	000241		CLC		
003434	006121		ROL	(1)+	
003436	005367	002374	DEC	COUNT	
003442	001341		BNE	LP17	
003444	105067	175331	CLRB	ICNT+1	

.....
 ;TEST 20 FLOATING 0'S TEST IN AC3
 ;.....

003450	104400		SCOPE		
003452	170011		SETD		
003454	012701	001002	MOV	#ANS1,R1	;LOAD FIRST WORD
003460	012721	177776	MOV	#-2,(1)+	;INITIALIZE TO 177776
003464	012721	177776	MOV	#-2,(1)+	;INITIALIZE TO 177776
003470	012721	177776	MOV	#-2,(1)+	;INITIALIZE TO 177776
003474	012721	177776	MOV	#-2,(1)+	;INITIALIZE TO 177776
003500	012767	000020	MOV	#16,,COUNT	;SET UP COUNT
003506	172767	175270	LDD	ANS1,3	;LOAD AC3
003512	174367	175274	STD	3,ANS5	;STORE RESULTS
003516	012702	001002	MOV	#ANS1,R2	;SET UP FIRST WORD
003522	022122		CMP	(1)+,(2)+	;IS DATA OK?
003524	001401		BEQ	,+4	
003526	104010		HLT+8.		;FIRST WORD IS WRONG
003530	022122		CMP	(1)+,(2)+	;IS DATA OK?
003532	001401		BEQ	,+4	
003534	104010		HLT+8.		;SECOND WORD IS WRONG
003536	022122		CMP	(1)+,(2)+	;IS DATA OK?
003540	001401		BEQ	,+4	
003542	104010		HLT+8.		;THIRD WORD IS WRONG
003544	022122		CMP	(1)+,(2)+	;IS DATA OK?
003546	001401		BEQ	,+4	
003550	104010		HLT+8.		;LAST WORD IS WRONG
003552	012701	001002	MOV	#ANS1,R1	
003556	000241		CLC		;SHIFT ALL BY 1 (FLOAT 0)
003560	006121		ROL	(1)+	
003562	000241		CLC		
003564	006121		ROL	(1)+	
003566	000241		CLC		
003570	006121		ROL	(1)+	
003572	000241		CLC		
003574	006121		ROL	(1)+	
003576	005367	002234	DEC	COUNT	
003602	001341		BNE	LP20	
003604	105067	175171	CLRB	ICNT+1	

.....
 ;TEST 21 FLOATING 0'S TEST IN AC4
 ;.....

003610	104400			SCOPE		
003612	173011			SETD		
003614	012701	001002		MOV	#ANS1,R1	;LOAD FIRST WORD
003620	012721	177776		MOV	#-2,(1)+	;INITIALIZE TO 177776
003624	012721	177776		MOV	#-2,(1)+	;INITIALIZE TO 177776
003630	012721	177776		MOV	#-2,(1)+	;INITIALIZE TO 177776
003634	012721	177776		MOV	#-2,(1)+	;INITIALIZE TO 177776
003640	012767	000020	002170	MOV	#16,COUNT	;SET UP COUNT
003646	172437	001002	LP21:	LDD	#ANS1,0	;LOAD AC0
003652	174004			STD	0,AC4	;PUT INTO AC4
003654	172404			LDD	AC4,0	;BACK INTO AC0
003656	174037	001012		STD	0,#ANS5	;NOW INTO ANS5
003662	012702	001002		MOV	#ANS1,R2	;SET UP FIRST WORD
003666	022122			CMP	(1)+,(2)+	;IS DATA OK?
003670	001401			BEQ	,+4	
003672	104010			HLT+0.		;FIRST WORD IS WRONG
003674	022122			CMP	(1)+,(2)+	;IS DATA OK?
003676	001401			BEQ	,+4	
003700	104010			HLT+0.		;SECOND WORD IS WRONG
003702	022122			CMP	(1)+,(2)+	;IS DATA OK?
003704	001401			BEQ	,+4	
003706	104010			HLT+0.		;THIRD WORD IS WRONG
003710	022122			CMP	(1)+,(2)+	;IS DATA OK?
003712	001401			BEQ	,+4	
003714	104010			HLT+0.		;LAST WORD IS WRONG
003716	012701	001002		MOV	#ANS1,R1	
003722	000241			CLC		;SHIFT ALL BY 1 (FLOAT 0)
003724	006121			RCL	(1)+	
003726	000241			CLC		
003730	006121			RCL	(1)+	
003732	000241			CLC		
003734	006121			RCL	(1)+	
003736	000241			CLC		
003740	006121			RCL	(1)+	
003742	005367	002070		DEC	COUNT	
003746	001337			BNE	LP21	
003750	105067	175025		CLRB	ICNT+1	

.....
 :TEST 22 FLOATING 3'S TEST IN AC5
 :.....

003754	104400			SCOPE		
003756	170211			SETD		
003760	012701	001002		MOV	#ANS1,R1	:LOAD FIRST WORD
003764	012721	177776		MOV	#-2,(1)+	:INITIALIZE TO 177776
003770	012721	177776		MOV	#-2,(1)+	:INITIALIZE TO 177776
003774	012721	177776		MOV	#-2,(1)+	:INITIALIZE TO 177776
004000	012721	177776		MOV	#-2,(1)+	:INITIALIZE TO 177776
004004	012767	000020	002024	MOV	#16,COUNT	:SET UP COUNT
004012	172437	001002		LDD	#ANS1,0	:LOAD AC0
004016	174005		LP22:	STD	0,AC5	:PUT INTO AC5
004020	172405			LDD	AC5,0	:BACK INTO AC0
004022	174037	001012		STD	0,#ANS5	:NOW INTO AN5
004026	012702	001002		MOV	#ANS1,R2	:SET UP FIRST WORD
004032	022122			CMP	(1)+,(2)+	:IS DATA OK?
004034	001401			BEQ	,+4	
004036	104010			HLT+0.		:FIRST WORD IS WRONG
004040	022122			CMP	(1)+,(2)+	:IS DATA OK?
004042	001401			BEQ	,+4	
004044	104010			HLT+0.		:SECOND WORD IS WRONG
004046	022122			CMP	(1)+,(2)+	:IS DATA OK?
004050	001401			BEQ	,+4	
004052	104010			HLT+0.		:THIRD WORD IS WRONG
004054	022122			CMP	(1)+,(2)+	:IS DATA OK?
004056	001401			BEQ	,+4	
004060	104010			HLT+0.		:LAST WORD IS WRONG
004062	012701	001002		MOV	#ANS1,R1	
004066	000241			CLC		:SHIFT ALL BY 1 (FLOAT 0)
004070	006121			RCL	(1)+	
004072	000241			CLC		
004074	006121			RCL	(1)+	
004076	000241			CLC		
004100	006121			RCL	(1)+	
004102	000241			CLC		
004104	006121			RCL	(1)+	
004106	005367	001724		DEC	COUNT	
004112	001337			BNE	LP22	
004114	105067	174661		CLRB	ICNT+1	

.....
 ;TEST 23 MEMORY TEST 1 (MEMORY INTO ITSELF) USING STF (1)+
 ;.....

004120	104400			SCOPE		
004122	170001			SETF		
004124	012701	006042		MOV	#HICORE,R1	;GET END OF PROGRAM
004130	010167	174646		LP23:	R1,ANS1	
004134	010167	174644		MOV	R1,ANS2	
004140	062767	000002	174636	ADD	#2,ANS2	
004146	172467	174630		LDF	ANS1,AC0	;GET THE DATA
004152	174021			STF	AC0,(1)+	;STORE INTO MEMORY
004154	026701	001660		CMP	UPLIM,R1	;END YET?
004160	101363			BHI	LP23	
004162	012701	006042		MOV	#HICORE,X1	
004166	020111			LP23A:	CMP	X1,(1)
004170	001805			BNE	NG23	;CHECK THE RESULTS
004172	005721			LP23B:	TST	(1)+
004174	026701	001640		CMP	UPLIM,R1	;GO TO THE NEXT
004200	101372			RHI	LP23A	;CHECK FOR END
004202	000406			BR	NXT23	
004204	010167	174572		NG23:	MOV	X1,ANS1
004210	011167	174570		MOV	(1),ANS2	
004214	104002			HLT+2		;DATA WRONG
004216	000755			BR	LP23B	
004220	105067	174555		NXT23:	CLRB	ICNT+1
						;ONLY ONCE

.....
 ;TEST 24 MEMORY TEST 2 USING LDF (1)
 ;.....

004224	104400			SCOPE		
004226	170001			SETF		
004230	012701	006042		MOV	#WICORE,X1	;GET END OF PROGRAM
004234	011101		LP24:	MOV	(1),X1	;PUT ADDRESS INTO ITSELF
004236	005721			TST	(1)+	;KLUDGE IT FOR 11/20
004240	026701	001574		CMP	UPLIM,R1	;END?
004244	101373			BHI	LP24	
004246	012701	006042		MOV	#WICORE,X1	
004252	016702	001562		MOV	UPLIM,X2	
004256	011167	174520		MOV	(1),ANS1	;GET FOR
004262	016167	000002	174514	MOV	2(1),ANS2	;TYPING
004270	172411			LDF	(1),AC2	;LOAD FROM CORE
004272	174067	174510		STF	AC0,ANS3	;GET IT BACK
004276	021167	174504		CMP	(1),ANS3	;FIRST WORD RIGHT
004302	001011			BNE	NG24	;NO
004304	026167	000002	174476	CMP	2(1),ANS4	;SECOND WORD RIGHT
004312	001005			BNE	NG24	;NO
004314	022121			CMP	(1)+,(1)+	;GET TO NEXT
004316	026701	001516		LP24B:	CMP	UPLIM,R1
004322	101355			BHI	LP24A	;NO
004324	000400			BR	NXT24	;YES
004326	012167	174454		NG24:	MOV	(1)+,ANS3
004332	012167	174452			MOV	(1)+,ANS4
004336	104004			HLT+4		;DATA WRONG
004340	000766			BR	LP24B	
004342	105067	174433		NXT24:	CLRB	ICNT+1
						;ONLY ONCE

```

034346 104400          DONE:  SCOPE
034350 032737 002000 177570  BIT      #SW12,0#SWR      ;RING THE BELL?
034356 001005          BNE      15        ;NO!
034360 012767 000007 001436  MOV      #BELL,,TYPE ;TYPE A BELL
034366 000004 006024          TYPE      ,,TYPE

034372 005046          15:    CLR      -(6)      ;CLEAR TRACE TRAP
034374 032737 010000 177570  BIT      #SW12,0#SWR ;RUN WITH TRY?
034402 001010          BNE      25
034404 005167 001416          COM      TRPB
034410 100005          BPL      25
034412 052716 000020          BIS      #20,(6)    ;SET TRACE TRAP
034416 012746 001146          MOV      #BEGIN,-(6)
034422 000414          BR      YESRT
034424 012746 004432          25:    MOV      #45,-(6)    ;JUMP TO 45
034430 000002          RTI
034432 013700 000042          45:    MOV      #42,X0    ;GET MONITOR ADDRESS
034436 001404          BEQ      35        ;NONE
034440 004710          JSR      7,(0)    ;GO TO IT
034442 000240          NOP
034444 000240          NOP
034446 000240          NOP
034450 000167 174472          35:    JMP      BEGIN
034454 000002          YESRT: RTI        ;RETURN TO PROGRAM FROM TRAP

034456 032737 000400 177570  .ENT:   BIT      #SW00,0#SWR ;KILL LDUB OR LOOP ON SPEC. TEST
034464 001404          BEQ      15
034466 123767 177570 174304  CMPB    #0#SWR,ICNT ;ON RIGHT TEST?
034474 001437          BEQ      OVER
034476 113703 177570          15:    MOVB    #0#SWR,X3    ;GET UB BITS
034502 170003          LDUB
034504 032737 040000 177570  BIT      #SW14,0#SWR ;LOOP ON TEST
034512 001026          BNE      KIT
034514 032737 004000 177570  BIT      #SW11,0#SWR ;KILL ITERATIONS
034522 001012          BNE      SAVLAD
034524 105767 174251          TSTB    ICNT+1
034530 001404          BEQ      25
034532 126767 001276 174241  CMPB    TIMES,ICNT+1 ;BRANCH IF FIRST
034540 001013          BNE      KIT        ;DONE?
034542 112767 000001 174231  25:    MOVB    #1,ICNT+1 ;BRANCH IF NOT
034550 105267 174224          SAVLAD: INCB    ICNT    ;FIRST ITERATION
034554 011667 001250          MOV      (6),LAD    ;COUNT TEST NUMBERS
034560 016737 174214 177570  MOV      ICNT,0#SWR ;SAVE LOOP ADDRESS
034566 000002          RTI        ;DISPLAY TEST NO. AND ITERATION COUNT
034566          ;RETURN

034570 105267 174205          KIT:   INCB    ICNT+1
034574 016737 174200 177570  OVER:  MOV      ICNT,0#SWR ;SET UP DISPLAY
034602 005767 001222          TST     LAD        ;FIRST ONE?
034606 001760          BEQ     SAVLAD
034610 016716 001214          MOV     LAD,(6)    ;FUDGE RETURN ADDRESS
034614 000002          RTI        ;FIXES PS
    
```

034616	032737	002000	177570	.TRP:	BIT	#SW17,0#SWR	
034624	001405				REQ	15	
034626	012767	000007	001170		MOV	#BELL,,TYPE	;TYPE A BELL
034634	000004	006024			TYPE	,,TYPE	
034640	004767	000600		15:	JSR	PC,ERROR	;COUNT THE NUMBER OF ERRORS
034644	010446				MOV	R4,-(6)	
034646	032737	020000	177570		BIT	#SW13,0#SWR	;SKIP TYPEOUT IF SET
034654	001072				BNE	45	
034656	000004	005772			TYPE	,RETURN	
034662	016646	000002			MOV	2(6),-(6)	;PUT ADDRESS OF INSTRUCTION ON STACK
034666	162716	000002			SUB	#2,(6)	
034672	011605				MOV	(6),TTY	;TYPE (6) IN OCTAL
034674	004767	000404			JSR	X7,PRINTR	;TYPE LEADING ZERO'S
034700	000004	006000			TYPE	,SPACE+3	
034704	010005				MOV	X0,TTY	;TYPE X0 IN OCTAL
034706	004767	000372			JSR	X7,PRINTR	;TYPE LEADING ZERO'S
034712	000004	006001			TYPE	,SPACE+4	
034716	012703	001002			MOV	#ANS1,X3	;ADDRESS OF DATA
034722	113604				MOV	0(6)+,R4	;AMOUNT OF DATA IN TABLE
034724	001426				BEO	35	
034726	100016				BPL	25	;TYPE STACK?
034730	016667	000006	174044		MOV	6(6),ANS1	
034736	016667	000010	174040		MOV	10(6),ANS2	
034744	016667	000012	174034		MOV	12(6),ANS3	
034752	016667	000014	174030		MOV	14(6),ANS4	
034760	042704	177600			BIC	#177600,R4	;CLEAR SIGN
034764	000004	006001		25:	TYPE	,SPACE+4	
034770	012305				MOV	(3)+,TTY	;TYPE (3)+ IN OCTAL
034772	004767	000306			JSR	X7,PRINTR	;TYPE LEADING ZERO'S
034776	005304				DEC	R4	
035000	001371				BNE	25	
035002	005700			35:	TST	FPS	
035004	100016				BPL	45	
035006	000004	005775			TYPE	,SPACE	
035012	170367	174004			STST	FEC	
035016	016705	174000			MOV	FEC,TTY	;TYPE FEC IN OCTAL
035022	004767	000256			JSR	X7,PRINTR	;TYPE LEADING ZERO'S
035026	000004	006000			TYPE	,SPACE+3	
035032	016705	173766			MOV	FEA,TTY	;TYPE FEA IN OCTAL
035036	004767	000242			JSR	X7,PRINTR	;TYPE LEADING ZERO'S
035042	012604			45:	MOV	(6)+,R4	
035044	005737	177570			TST	0#SWR	
035050	100001				BPL	,+4	
035052	000000				HALT		
035054	032737	001000	177570		BIT	#SW09,0#SWR	;CHECK FOR INHIBIT LOOP ON ERROR
035062	001001				BNE	,+4	
035064	000002				RTI		
035066	105067	173707			CLRB	ICNT+1	
035072	032737	000400	177570		BIT	#SW08,0#SWR	;CHECK FOR LOAD MICROBREAK
035100	001233				BNE	KIT	;BRANCH IF NOT
035102	113703	177570			MOV	0#SWR,X3	;PUT MICROBREAK ADDRESS IN R3
035106	170003				LDUB		;LOAD MICROBREAK
035110	000627				BR	KIT	;LOOP ON TEST UNTIL NO ERRORS

005112	004767	000036		RAND4:	JSR	7,RANDOM	
005116	016767	000160	173656		MOV	HINUM,ANS1	
005124	016767	000150	173652		MOV	LONUM,ANS2	
005132	004767	000016			JSR	7,RANDOM	
005136	016767	000140	173642		MOV	HINUM,ANS3	
005144	016767	000130	173636		MOV	LONUM,ANS4	
005152	000207				RTS	7	
005154	010046			RANDOM:	MOV	%0,-(6)	;SAVE R0
005156	010146				MOV	%1,-(6)	;SAVE R1
005160	010246				MOV	%2,-(6)	;SAVE R2
005162	010446				MOV	R4,-(6)	;SAVE R4
005164	016700	000110			MOV	LONUM,%0	;SET R0 WITH LOW
005170	016701	000106			MOV	HINUM,%1	;SET R1 WITH HIGH
005174	012704	177771			MOV	#-7,R4	;SET SHIFT COUNT
005200	005002				CLR	%2	
005202	006300			SHIFT:	ASL	%0	;SHIFT R0 LEFT AND
005204	006101				RCL	%1	;ROTATE CARRY INTO R1 AND
005206	006102				RCL	%2	;ROTATE CARRY INTO R2
005210	005204				INC	R4	;CHECK FOR DONE
005212	001373				BNE	SHIFT	;CONTINUE SHIFT LOOP
005214	066702	000060			ADD	LONUM,%2	;ADD NUMBER TO MAKE X 129
005220	005501				ACC	%1	;PROPOGATE CARRY
005222	066701	000054			ADD	HINUM,%1	;ADD NUMBER TO MAKE X 129
005226	005502				ADC	%2	;PROPOGATE CARRY
005230	062700	001057			ADD	#1057,%0	;ADD LOW CONSTANT
005234	005501				ACC	%1	;PROPOGATE CARRY
005236	005502				ADC	%2	;PROPOGATE CARRY
005240	062701	047401			ADD	#47401,%1	;ADD HIGH CONSTANT
005244	005502				ADC	%2	;PROPOGATE CARRY
005246	062702	000006			ADD	#6,%2	;ADD HIGHEST CONSTANT
005252	060200				ADD	%2,%0	;REPRIME R0 WITH HIGHEST DIGIT
005254	005501				ADC	%1	;PROPOGATE CARRY
005256	010067	000016			MOV	%0,LONUM	;SAVE R0
005262	010167	000014			MOV	%1,HINUM	;SAVE R1
005266	012604				MOV	(6)+,R4	;RESTORE R4
005270	012602				MOV	(6)+,%2	;RESTORE R2
005272	012601				MOV	(6)+,%1	;RESTORE R1
005274	012600				MOV	(6)+,%0	;RESTORE R0
005276	000207				RTS	PC	;RETURN
005300	123456			LONUM:		123456	
005302	176543			HINUM:		176543	

```

005324 112767 000001 000130 PRINTR: MOVB #1,PRS ;SET ZERO FILL SWITCH
005312 000402 RR ,+6
005314 005067 000122 PRINTS: CLR PRS ;SUPPRESS LEADING ZERO'S
005320 112767 177772 000115 MCVB #=6,PRS+1 ;SET COUNT
005326 010446 MCV R4,-(6) ;SAVE R4
005330 012704 005432 MCV #3S,R4 ;SET POINTER TO FIRST ASCII CHAR.
005334 105014 CLRB (4) ;CLEAR FIRST BYTE
005336 000405 RR 2S ;ROTATE FIRST BIT
005340 105014 1S: CLRB (4) ;CLEAR BYTE OF CHARACTER
005342 006105 ROL TTY ;ROTATE BIT INTO C
005344 106114 ROLB (4) ;PACK IT
005346 006105 RCL TTY ;ROTATE BIT INTO C
005350 106114 ROLB (4) ;PACK IT
005352 006105 2S: ROL TTY ;ROTATE BIT INTO C
005354 106114 ROLB (4) ;PACK IT
005356 105714 TSTB (4)
005360 001402 BEQ ,+6
005362 105267 000054 INCB PRS
005366 105767 000050 TSTB PRS ;CHECK FILL SWITCH
005372 001402 BEQ ,+6
005374 152724 000060 RISB #'0,(4)+ ;MAKE INTO ASCII CHAR
005400 105267 000037 INCB PRS+1
005404 001355 BNE 1S ;REPEAT
005406 022704 005432 CMP #3S,R4
005412 001002 RNE ,+6
005414 112724 000060 MOVB #'0,(4)+
005420 105014 CLRB (4)
005422 000004 005432 TYPE ,3S ;TYPE IT
005426 012604 MCV (6)+,R4 ;RESTORE R4
005430 000207 RTS PC

005432 000004 3S: .BLKW 4
005442 000000 PRS: 0

005444 005267 000362 ERROR: INC ERRORS ;COUNT ERRORS
005450 132737 000001 000041 BITB #1,0#41 ;AUTO MODE?
005456 001412 BEQ 1S ;NO!
005460 022767 000010 000344 CMP #10,ERRORS ;TOO MANY?
005466 001006 RNE 1S ;NOT YET
005470 013700 000042 MCV #042,R0 ;GET ADDRESS
005474 001403 BEQ 1S ;FORGET IT IF ZERO
005476 005037 000042 CLR #042 ;ZAP 42
005502 004710 JSR PC,(?) ;CALL THE MONITOR
005504 000207 1S: RTS PC ;RETURN
    
```

```

005506 012777 005702 000304 POWDWN: MOV #ILLUP, @UPVEC ;SET FOR FAST UP
005514 012777 000340 000300 MOV #340, @UPVEC+2 ;PRIO:7
005522 170246 STFPS -(6) ;GET THE FPS
005524 170011 SETD ;
005526 174046 STD AC0, -(6) ;SAVE AC'S
005530 174146 STD AC1, -(6)
005532 174246 STD AC2, -(6)
005534 174346 STD AC3, -(6)
005536 172404 LDD AC4, AC0
005540 174046 STD AC0, -(6)
005542 172405 LDD AC5, AC0
005544 174046 STD AC0, -(6)
005546 010046 MOV R0, -(6) ;SAVE REGISTERS
005550 010146 MOV R1, -(6)
005552 010246 MOV R2, -(6)
005554 010346 MOV R3, -(6)
005556 010446 MOV R4, -(6)
005560 010546 MOV R5, -(6)
005562 010667 000216 MOV SP, SAVE6 ;SAVE SP
005566 012777 005576 000224 MOV #POWUP, @UPVEC ;SET UP VECTOR
005574 000000 HALT

005576 016706 000202 POWUP: MOV SAVE6, SP ;GET SP
005602 005001 CLR R1 ;WAIT LOOP FOR THE TTY
005604 005201 1S: INC R1
005606 001376 BNE 1S
005610 012605 MOV (6)+, R5 ;GET THE REGISTERS
005612 012604 MOV (6)+, R4
005614 012603 MOV (6)+, R3
005616 012602 MOV (6)+, R2
005620 012601 MOV (6)+, R1
005622 012600 MOV (6)+, R0
005624 170011 SETD
005626 172426 LDD (6)+, AC0 ;RESTORE THE AC'S
005630 174005 STD AC0, AC5
005632 172426 LDD (6)+, AC0
005634 174004 STD AC0, AC4
005636 172726 LDD (6)+, AC3
005640 172626 LDD (6)+, AC2
005642 172526 LDD (6)+, AC1
005644 172426 LDD (6)+, AC0
005646 170126 LDFPS (6)+ ;RESTORE FPS
005650 012777 005506 000136 MOV #POWDWN, @DWNVEC ;SET UP THE POWER DOWN VECTOR
005656 012777 000340 000132 MOV #340, @DWNVEC+2
005664 000004 005670 TYPE ;.ASCIZ <15><12>"POWER"
005700 000002 RTI

005702 000000 ILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
005704 000776 BR .-2 ; BEFORE THE POWER DOWN WAS COMPLETE
    
```

```

005706 010546          .IOT:  MCV      TTY,-(6)      ;SAVE TTY
005710 017605 000002    MCV      02(6),TTY    ;GET ADDRESS TO BE TYPED
005714 105715          1S:   TSTB     (TTY)      ;TERMINATOR?
005716 001406          BEQ      2S          ;
005720 112537 177566    MCVB    (TTY)+,0#177566 ;LOAD AND TYPE THE CHARACTER
005724 105737 177564    TSTB    0#177564     ;IS THE PRINTER READY
005730 100375          BPL      ,=4        ;
005732 000770          BR       1S          ;GET THE NEXT CHARACTER

005734 017646 000002    2S:   MCV      02(6),-(6) ;GET ADDRESS TO BE TYPED
005740 062766 000002 000004  ADD      #2,4(6)      ;ADD 2 TO THE ADDRESS
005746 022666 000002    CMP      (6)+,2(6)    ;IS IT ,+2?
005752 001005          RNE      3S          ;NO
005754 005725          TST      (TTY)+      ;ADD 2 TO THE ADDRESS
005756 042705 000001    BIC      #1,TTY      ;BACK UP TO AN EVEN BYTE
005762 010566 000002    MCV      TTY,2(6)    ;RESTORE ADDRESS
005766 012605          3S:   MCV      (6)+,TTY ;RESTORE TTY
005770 000002          RTI                    ;RETURN

005772 005015 000          RETURN: ,ASCIZ <15><12>
005775 015 020012 020040  SPACE:  ,ASCIZ <15><12>"
006002 000
006004          .EVEN
006004 000000  SAVE6:  0
006006 172160  FPTADR: 172160      ;FLOATING POINT ADDRESS ON THE 11/20
006010 000244 000246  FPVECT: 244,246    ;FLOATING POINT VECTOR ADDRESS
006014 000024 000026  DWNVEC: 24,26     ;POWER DOWN VECTOR ADDRESS
006020 000024 000026  UPVEC:  24,26     ;POWER UP VECTOR ADDRESS
006024 000000  .TYPE:  0
006026 000000  TRPB:   0
006030 000000  LAD:    0          ;LOOP ADDRESS
006032 000000  ERRORS: 0          ;ERROR COUNT
006034 000377  TIMES:  377       ;ITERATION COUNT
006036 000000  COUNT:  0
006040 000000  UPLIM:  0
006042 000000  WICORE: 0

000001          .END

```

AC0	=X000000	379#	1042#	1043	1073#	1074	1203	1207#	1208	1289#	1297	1312#	1313	1314#
		1315	1319#											
AC1	=X000001	380#	1204	1318#										
AC2	=X000002	381#	1205	1317#										
AC3	=X000003	382#	1206	1316#										
AC4	=X000004	383#	541#	542	733#	734	963#	964	1207	1315#				
AC5	=X000005	384#	565#	566	770#	771	1004#	1005	1209	1313#				
ANS1	001002	397#	455	457	478	482	498	502	521	523	544	568	586	592
		594	607	621	627	629	642	656	662	664	677	691	697	699
		712	726	732	736	749	763	769	773	786	807	806	808	821
		839	845	847	862	878	884	886	899	917	923	925	938	956
		962	966	979	997	1003	1007	1020	1030#	1042	1053#	1071#	1156	1160#
		1193#												
ANS2	001004	398#	460	483	523	526	547	571	1040#	1041#	1054#	1072#	1161#	1164#
ANS3	001006	399#	463	486	526	529	552	574	1074#	1075	1083#	1162#	1196#	
ANS4	001010	400#	466	489	529	532	553	577	1077	1084#	1163#	1197#		
ANS5	001012	401#	456#	457	479#	482	499#	502	522#	523	543#	544	567#	568
		593#	628#	663#	698#	735#	772#	807#	846#	885#	924#	965#	1026#	
ANS6	001014	402#	468	483	523	526	547	571						
ANS7	001016	403#	463	486	526	529	552	574						
ANS8	001020	404#	466	489	529	532	553	577						
BEG	001026	387	407#											
REGIN	001146	424	427	431#	1120	1110								
BELL	= 000007	360#	1091	1141										
COUNT	006036	591#	612#	626#	647#	661#	682#	696#	717#	731#	754#	768#	791#	825#
		830#	844#	869#	883#	908#	922#	947#	961#	988#	1002#	1029#	1361#	
DONE	004346	1008#												
DWVEC	006014	434#	435#	1321#	1322#	1354#								
ERROR	005444	1143	1269#											
ERRORS	006032	1269#	1272	1359#										
FEA	001024	406#	1177											
FEC	001022	391#	405#	1173#	1174									
FLTERR	000760	390#	442											
FPS	=X000000	369#	390#	1170										
FPTADR	006006	420	1352#											
FPVECT	006010	442#	443#	1353#										
HAVIT	001112	409	416	423#										
HICORE	006042	415	1038	1046	1064	1069	1363#							
HINJM	005302	1193	1196	1205	1215	1226#	1234#							
HLT	= 104000	359#	459	462	465	468	482	485	488	491	502	505	508	511
		525	528	531	534	546	549	552	555	570	573	576	579	587
		600	603	606	632	635	638	641	667	670	673	676	702	705
		708	711	739	742	745	748	776	779	782	785	811	814	817
		820	850	853	856	857	889	892	895	898	920	931	934	937
		969	972	975	978	1010	1013	1016	1019	1055	1085			
ICNT	001000	396#	444#	614#	649#	684#	719#	756#	793#	832#	871#	910#	949#	980#
		1031#	1057#	1087#	1115	1123	1125	1127#	1128#	1130	1133#	1134	1186#	
ILLUP	005702	1279	1326#											
KIT	004570	1120	1126	1133#	1188	1191								
LAD	006030	445#	1129#	1135	1137	1358#								
LDUB	= 170003	354#												
LONJM	005300	1194	1197	1204	1213	1225#	1233#							
LP10	002146	627#	648											
LP11	002276	662#	683											

LP12	002426	697#	718																	
LP13	002596	732#	755																	
LP14	002712	769#	792																	
LP15	003046	806#	831																	
LP16	003206	845#	870																	
LP17	003346	884#	909																	
LP20	003506	923#	948																	
LP21	003646	962#	989																	
LP22	004012	1003#	1030																	
LP23	004130	1039#	1045																	
LP23A	004166	1047#	1051																	
LP23B	004172	1049#	1056																	
LP24	004234	1065#	1068																	
LP24A	004256	1071#	1081																	
LP24B	004316	1080#	1086																	
LP7	002016	592#	613																	
V	= 000001	355#																		
NEM	001106	411	422#																	
NG23	004204	1048	1053#																	
NG24	004326	1076	1078	1083#																
NXT23	004220	1052	1057#																	
NXT24	004342	1082	1087#																	
OVER	004574	1116	1134#																	
PC	=X000007	370#	1143#	1231#	1264#	1277#	1278#													
POWDN	005506	434	1279#	1321																
POWUP	005576	1298	1301#																	
PRINTR	005304	1151	1154	1167	1175	1178	1235#													
PRINTS	005314	1237#																		
PRS	005442	1235#	1237#	1238#	1252#	1253	1256#	1267#												
PS	= 177776	356#																		
RANDOM	005154	1192	1195	1200#																
RAND4	005112	448	1192#																	
RETURN	005772	1147	1347#																	
R0	=X000000	370#	437#	1274#	1291	1310#														
R1	=X000001	371#	408#	412#	414#	415	417#	419#	420	422#	423	586#	607#	621#						
		642#	656#	677#	691#	712#	726#	740#	763#	786#	807#	821#	839#	860#						
		878#	899#	917#	938#	956#	979#	997#	1020#	1038#	1039	1041	1044	1050						
		1067	1080	1292	1302#	1303#	1309#													
R2	=X000002	372#	594#	629#	664#	699#	736#	773#	808#	847#	886#	925#	966#	1007#						
		1293	1308#																	
R3	=X000003	373#	1294	1307#																
R4	=X000004	374#	1144	1157#	1164#	1168#	1179#	1203	1206#	1211#	1227#	1239	1240#	1250						
		1263#	1295	1306#																
R5	=X000005	376#	1296	1305#																
SAVE6	006004	1297#	1301	1351#																
SAVLAD	004550	1122	1128#	1136																
SCOPE	= 104400	358#	453	476	496	519	539	563	584	619	654	689	724	761						
		798	837	876	915	954	995	1036	1062	1088										
SHIFT	005202	1208#	1212																	
SP	=X000006	377#	407#	431#	1297	1301#														
SPACE	005775	1152	1155	1165	1172	1176	1348#													
SWR	= 177570	357#	1089	1095	1113	1115	1117	1119	1121	1130#	1134#	1139	1145	1180						
		1183	1187	1189																
SW08	= 000400	368#	1113	1187																

SW09	= 001000	367#	1183																	
SW10	= 002000	366#	1089	1139																
SW11	= 004000	365#	1121																	
SW12	= 010000	364#	1095																	
SW13	= 020000	363#	1145																	
SW14	= 040000	362#	1119																	
TIMES	006034	1125	1360#																	
TRPH	006026	1097#	1357#																	
TRYAGN	001072	410#	421																	
TTY	=X000005	375#	1150#	1153#	1166#	1174#	1177#	1244#	1246#	1248#	1328	1329#	1332	1332						
		1341	1342#	1343	1344#															
TYPE	= 000004	361#	1092	1142	1147	1152	1155	1165	1172	1176	1262	1323								
UPLIM	006040	423#	1044	1050	1067	1070	1080	1362#												
UPVEC	006020	1279#	1280#	1298#	1355#															
YESRT	004454	426#	433	1101	1111#															
.	= 006044	385#	386#	389#	395#	458	461	464	467	481	484	487	490	491						
		504	507	510	524	527	530	533	545	548	551	554	569	572						
		575	578	596	599	602	605	631	634	637	647	666	669	672						
		675	701	704	707	710	738	741	744	747	775	778	781	784						
		810	813	816	819	849	852	855	858	888	891	894	897	907						
		930	933	936	968	971	974	977	1000	1012	1015	1018	1101	1104						
		1236	1251	1254	1259	1260#	1323	1327	1334	1350#										
.EMT	004456	440	1113#																	
.IOT	005706	436	1328#																	
.TRP	004616	438	1139#																	
.TYPE	006024	1091#	1092	1141#	1142	1356#														

DUMP	3530	1150	1153	1166	1174	1177
PRINT	3530	1323				
SDUMP	3530					
TYPEM	3530	1091	1141			

ADC	1214	1216	1218	1219	1221	1224										
ADD	419	1241	1213	1215	1217	1227	1222	1223	133A							
ASL	62A	639	617	611	643	644	645	64A	67A	679	682	681	713	714	715	
	71A	75A	751	752	753	787	78A	789	79A	122A						
BEJ	413	458	461	464	467	481	484	487	49A	521	524	527	51	524	527	
	53A	533	545	548	551	554	569	572	575	57A	596	599	622	625	631	
	634	637	64A	666	669	672	675	701	704	727	717	73A	741	744	747	
	775	77A	781	784	81A	813	816	819	849	852	855	85A	88A	891	894	
	897	927	93A	933	936	968	971	974	977	1209	1212	121A	121A	1125	1114	
	111A	1124	1136	114A	1158	1251	1254	1271	127A	1331						
BGT	416															
BHI	1245	1251	1368	1381												
BIC	1164	1342														
BIS	1299															
BISB	1255															
BIT	1289	1295	1113	1119	1121	1139	1145	1183	1187							
BITB	1278															
BNE	421	613	648	683	718	755	792	831	87A	929	948	989	123	1248	1276	
	1278	129A	1296	112A	1122	1126	1146	1169	1184	118A	1212	1257	1259	1273	1324	
	134A															
BPL	1298	1159	1171	1181	1334											
BR	429	427	1252	1256	1282	1286	1121	1191	1236	1242	1327	1335				
CLC	822	824	826	82A	861	863	865	867	92A	922	924	926	939	941	943	
	945	98A	982	984	986	1221	1223	1225	1227							
CLR	444	445	1294	1287	1237	1276	1302									
CLRB	614	649	684	719	756	793	832	871	91A	949	998	1231	1257	1287	1286	
	1241	1243	1261													
CMP	415	422	457	462	463	466	482	483	486	489	523	523	526	529	523	
	526	529	532	544	547	552	553	568	571	574	577	59A	598	621	624	
	63A	633	636	639	665	668	671	674	72A	723	726	729	737	740	743	
	746	774	777	782	783	829	812	815	81A	84A	851	854	857	887	852	
	893	896	926	929	932	935	967	97A	973	976	1289	1211	1214	1217	1244	
	1247	1252	1267	1275	1277	1279	1282	1258	1272	1339						
CMPB	1115	1125														
COM	1297															
DEC	612	647	682	717	754	791	832	869	92A	947	989	1229	1168			
EMT	359															
HALT	386	392	1182	1299	1326											
INC	1211	1269	1303													
INCB	1128	1133	1252	1256												
IDT	361															
JMP	387	1112														
JSR	448	1106	1143	1151	1154	1167	1175	1178	1192	1195	1277					
LDD	455	476	498	521	542	566	592	627	662	697	732	734	769	771	826	
	845	884	923	962	964	1223	1225	1287	1289	1312	1314	1316	1317	1318	1319	
LDF	1242	1273														
LDFPS	446	132A														
LDUR	1118	119A														
MOV	427	411	412	417	423	424	426	428	431	432	433	434	435	436	437	
	43A	439	440	441	442	443	586	587	588	589	590	591	594	627	621	
	622	623	624	625	626	629	642	656	657	65A	659	66A	661	664	677	
	691	692	693	694	695	696	699	712	72A	727	72A	729	731	731	736	
	749	763	764	765	766	767	768	773	786	82A	821	822	823	824	825	
	828	821	839	840	841	842	843	844	847	86A	878	879	881	881	882	

	883	886	899	917	918	919	920	921	922	925	938	956	957	958	959
	960	961	966	979	997	998	999	1000	1001	1002	1007	1020	1038	1039	1040
	1046	1053	1054	1064	1065	1069	1070	1071	1072	1083	1084	1091	110	1102	1104
	1129	1130	1134	1137	1141	1144	1148	1150	1153	1156	1160	1161	1162	1163	1166
	1174	1177	1179	1193	1194	1196	1197	1200	1201	1202	1203	1204	1205	1206	1205
	1226	1227	1228	1229	1230	1239	1240	1263	1274	1279	1280	1291	1292	1293	1294
	1295	1296	1297	1298	1301	1305	1306	1307	1308	1309	1310	1321	1322	1328	1329
	1337	1343	1344												
MOV8	1117	1127	1157	1189	1235	1238	1260	1332							
NOP	1107	1108	1109												
ROL	823	825	827	829	862	864	866	868	901	923	925	927	94	942	944
	946	981	983	985	987	1022	1024	1026	1028	1209	1210	1244	1246	1248	
ROLB	1245	1247	1249												
RTI	393	1103	1111	1131	1138	1185	1324	1345							
RTS	1198	1231	1264	1278											
SETD	454	477	497	520	540	564	585	620	655	697	725	762	799	838	877
	916	955	996	1282	1311										
SETF	1037	1063													
STD	456	479	499	522	541	543	565	567	593	628	663	698	733	735	770
	772	807	846	885	924	963	965	1004	1006	1283	1284	1285	1286	1288	1290
	1313	1315													
STF	1043	1074													
STFPS	390	1281													
STST	391	1173													
SUB	408	414	422	1149											
TRAP	358														
TST	418	425	1049	1066	1135	1170	1180	1341							
TSTB	1123	1250	1253	1330	1333										
.ASCIZ	1324	1347	1348												
.BLKW	1266														
.ENABL	353														
.END	1365														
.EVEN	1324	1350													
.LIST	309	353	386	407	449	1088	1139	1192	1235	1279	1324	1328			
.MACR	353														
.MACRO	353														
.NLIST	309	353	386	407	449	1088	1139	1192	1235	1279	1324	1328			
.REM	312														
.REPT	2	386													
.SBTTL	309	353	407	449	1088	1139	1192	1235	1279	1328					
.TITLE	309														

ERRORS DETECTED: 0

•DCFPR,DCFPR/SOL/CRF-DCFPR,P11
RUN-TIME: 5 9 1 SECONDS
CORE USED: 7K