

MAINDEC-08-DO5B-D

IDENTIFICATION

Product Code: MAINDEC-08-DO5B-D
Product Name: Random JMP-JMS Test
Date Created: December 28, 1967
Maintainer: Diagnostic Group
Author: R. Green

MAINDEC-08-DO5B-D

1. ABSTRACT

This is a diagnostic program to test the JMS instruction of the PDP-8. Random FROM and TO addresses are selected for each test. The JMP instruction is tested in that each test requires a JMP to reach the JMS.

2. REQUIREMENTS

2.1 Equipment

PDP-8 equipped with Teletype.

2.2 Storage

Locations 0000 - 0574

The Binary Loader must be stored in the last memory page.

2.3 Preliminary Programs

It is assumed that MAINDEC 08-D01(n), 08-D02(n), 08-D03(n), and 08-D04(n) have been run successfully.

3. LOADING PROCEDURE

3.1 Method

Use the standard Binary Loader

4. STARTING PROCEDURE

4.1 Control Switch Settings

SR0 Halt on error.
SR2 Hold the FROM address constant (1).
 Select random FROM addresses (0).
SR3 Hold the TO address constant (1).
 Select random TO addresses (0).

4.2 Starting Address

0200

Restart Address - 0215

4.3 Operator Action

- a. Set SR to 0200 and press LOAD ADDRESS.
- b. If it is desired to set either SR2 or SR3, the FROM or TO address may be specified by entering the address into the locations shown below.

FROM = Location 130

TO = Location 126

If SR2 or SR3 is set after the program has been started, the last address taken from the random number generator is used repeatedly.

- c. Push START.

5. OPERATING PROCEDURE

Same as section 4.

6. ERRORS

6.1 Error Halts

All unused memory locations are loaded with HLT instructions. If the program executes one of these background HLTs, it is probable that the interrupt failed to occur following the JMS instruction. The FROM and TO address may be checked at any time to locate the test JMS instructions.

6.2 Error Printouts

F xxxx TO yyyy

(TO) = mmmm

(nnnn) = rrrr

6.2.1 Explanation

(FROM) F xxxx: xxxx = address of JMS instruction being tested.

(TO) TO yyyy: yyyy = address that the JMS instruction is going to.

(TO) = mmmm; mmmm = the contents of the address TO. This should equal xxxx + 1.

(nnnn) = rrrr: nnnn is the address minus one that was stored in location 0000 during the interrupt. rrrr is the content of address nnnn.

6.2.2 Examples

- a. The following is a forced error printout where no error occurred.

```
F 5236 TO 6354
(TO) = 5237
(6354) = 5237
```

The test JMS instruction was in location 5236. The JMS was trying to jump to location 6354. The contents of TO (location 6354) was 5237. This is correct since the PC is stored on a JMS instruction.

To gain any knowledge from the third line of the printout, the user must understand the sequence of events when a JMS instruction is followed by an interrupt. As an end result of this sequence, the address of the location following the cell where the PC is stored is placed into cell 0. To derive this third line of the printout, the address in cell 0 is decremented by one and printed on the Teletype; then the contents of that address are printed.

- b. The following is a typical error printout.

```
F 5236 TO 6354
(TO) = 7402
(4354) = 5237
```

Line 1 is simply a statement of the problem. Line 2 says that the contents of location 6354 are not 5237 as they should be, but are 7402 instead. 7402 is a HLT instruction. Since memory is filled with a background of HLT orders, it is evident that the PC was not stored in location 6354 during the JMS.

Line 3 of the printout reveals where the PC was stored. Since on the interrupt 4355 was stored in location zero and (4354) contains the correctly stored PC, 5237, it is apparent that a jump error occurred. The JMS instruction should have jumped to 6354, but it actually jumped to 4354. Bit 1 was lost.

- c. The following is another typical error printout.

```
F 5236 TO 6354
(TO) = 7237
(6354) = 7237
```

Line 1 is again simply a statement of the problem. Line 2 says that the contents of location 6354 are not 5237 as expected, but are instead 7237. Since the contents are not a HLT order, 7402, it is evident that the PC was stored here, but the number stored was wrong. Comparing the good (5237), and the bad (7237), it is apparent that bit 1 was "picked up" during the store PC operation of the JMS instruction.

6.3 Error Recovery

The program continues testing following an error printout. When enough information has been gathered from the error printouts, a FROM and TO address is selected for use in the scope mode loop. Enter the chosen addresses into proper locations (see section 4.3.b). Enter 5531 into location 1 and restart the program with SR2 and SR3 set.

The scope mode loop is:

<u>Location</u>	<u>Coding</u>
0000	
0001	JMP 1 FROM 1
xxxx	A, ION
xxxx	JMS 1 TO
0131	FROM 1 A

To discontinue the scope mode loop, restore the original contents (7200) of location 1 and restart.

7. RESTRICTIONS

(None)

8. MISCELLANEOUS

8.1 Execution Time

4,726 random tests/second

9. PROGRAM DESCRIPTION

The JMS instruction is checked through use of the interrupt function. A random number generator selects a FROM and a TO address. An ION instruction is then placed at FROM -1 and the JMS instruction at FROM. The program jumps to the address specified by TO. After executing the ION and JMS instructions, an interrupt occurs starting the program counter at location 1. A checking routine located here verifies that the operation was successful before starting the next test.

Random addresses are restricted as follows: $0600 < \text{random address} < 7600$.

The area between 0600 and 7600 is filled with HLT instructions in case the interrupt fails.

"05" is printed after every 61,000 tests.

0200

```

*200
/RANDOM JMP=JMS TEST
/SR0=HALT ON ERROR
/SR2=FIXED FROM
/SR3=FIXED TO
/SPREAD HALTS THROUGH MEMORY
/BETWEEN THE LIMLO AND LIMHI
/LIMITS

```

```

0200 4154
0201 1135
0202 7041
0203 3126
0204 1152
0205 3526
0206 1126
0207 7001
0210 3126
0211 1126
0212 1136
0213 7640
0214 5204
0215 1042
0216 3041
0217 3040

```

```

BEGIN: JMS PATCH /CLA
TAD LIMLO
CIA TU
DCA TU
TAD HALT
DCA I TU
TAD TU
IAC
DCA TU
TAD TO
TAD LIMHI
SEA CLA
JMP GUN
TAD M15
DCA CT1
DCA CT

```

/CHECK FOR FIXED FROM

```

0220 7604
0221 7004
0222 7006
0223 7630
0224 5246

```

```

LOOP: LAS
RAL
RTL
SEL CLA
JMP LOOP1=6

```

/GET RANDOM FROM

```

0225 1133
0226 7104
0227 7430
0230 1134
0231 3133
0232 1133
0233 7510
0234 5241
0235 1135
0236 7710
0237 5225
0240 5244
0241 1136
0242 7700
0243 5225

```

```

GETRAN: TAD RANUM
RAL CLL
SEL
TAD THREE
DCA RANUM
TAD RANUM
SPA
JMP ,+5
TAD LIMLO
SPA CLA
JMP GETRAN
JMP ,+4
TAD LIMHI
SMA CLA
JMP GETRAN

```

```

0244 1133
0245 5230

```

```

TAD RANUM
DCA FROM

```

0246 1130
0247 7001
0250 3132
0251 7040
0252 1130
0253 3131

TAD FROM
IAC
DCA FRMP1
CMA
TAD FROM
DCA FROM1

/CHECK FOR FIXED TO

0254	7604	LOOP1, LAS
0255	7006	RTL
0256	7006	RTL
0257	7630	S&L CLA
0260	5302	JMP CRCK=3

/GET RANDOM TO

0261	1133	GTRAN1, TAD RANUM
0262	7104	RAL CLL
0263	7430	S&L
0264	1134	TAD THREE
0265	3133	DCA RANUM
0266	1133	TAD RANUM
0267	7510	SPA
0270	5275	JMP .+5
0271	1135	TAD LIMLO
0272	7710	SPA CLA
0273	5261	JMP GTRAN1
0274	5300	JMP .+4
0275	1136	TAD LIMHI
0276	7700	SMA CLA
0277	5261	JMP GTRAN1
0300	1133	TAD RANUM
0301	3126	DCA TO
0302	1126	TAD TO
0303	7001	IAC
0304	3127	DCA TOP1
0305	1130	TAD FROM
0306	7041	CJA
0307	1126	TAD TO
0310	7050	SNA CLA
0311	5220	JMP LOOP

CRCK,

/BRING UP THE FLAG

0312	7040	CMA
0313	6041	TSF
0314	6046	TL5
0315	6041	TSF
0316	5315	JMP .-1

/PLACE THE INSTRUCTIONS

0317 7200
 0320 1137
 0321 3531
 0322 1153
 0323 3530
 0324 3000

CLA
 TAD ITON
 DCA I FROM1
 TAD JMP1
 DCA I FROM
 DCA 0

/GO ON IT

0325 5531
 0326 7402

JMP I FROM1
 HLT

/PRINTOUT SUBROUTINE

0327 0000
 0330 3143
 0331 1143
 0332 7012
 0333 7010
 0334 3142
 0335 1142
 0336 7012
 0337 7010
 0340 3141
 0341 1141
 0342 7012
 0343 7010
 0344 3140
 0345 5727

TYPAC, 0
 DCA SAVE*3
 TAD SAVE*3
 RTR
 RAR
 DCA SAVE*2
 TAD SAVE*2
 RTR
 RAR
 DCA SAVE*1
 TAD SAVE*1
 RTR
 RAR
 DCA SAVE
 JMP I TYPAC

0346 1041
 0347 7001
 0350 3041
 0351 1041
 0352 7640
 0353 5437
 0354 1373
 0355 3124
 0356 1124
 0357 7001
 0360 3124
 0361 1524
 0362 6046
 0363 6041
 0364 5363
 0365 1043
 0366 7640
 0367 5356
 0370 1042
 0371 3041

/SUCCESS PRINTOUT
 SUP,
 TAD CT1
 IAC
 DCA CT1
 TAD CT1
 SZA CLA
 JMP I AL00P
 TAD MSG2
 DCA WORK
 TAD WORK
 IAC
 DCA WORK
 TAD I WORK
 TLF
 TSP
 JMP ,=1
 TAD M26P
 SZA CLA
 JMP LP1
 TAD M15
 DCA CT1

LP1,

1/11/68 3120.22

JMP I ALDOP

0372 5437

0373	0373	AMSG2:	.		
0374	0215		215	/CR	
0375	0212		212	/LF	
0376	0260		200	/D	
0377	0265		205	/5	
0000	0000	*0	0	/FOR SCOPE MODE INSERT	
0001	5001		JMP 1	/JMP I FROM 1 (5531) IN LOCI	
0002	0002		2	/GET STORED ADDRESS	
0003	0003		3		
0004	1132		TAD FRMP1		
0005	7640		SEA CLA		
0006	5546		JMP I AER	/ADDRESS STORED IN (TO) WRONG	
0007	1127		TAD TOP1		
0010	7041		CIA 0		
0011	1000		TAD 0		
0012	7640		SEA CLA		
0013	5546		JMP I AER	/ADDRESS STORED IN (0) WRONG	
0014	1152		TAD HALT		
0015	3530		DCA I FROM		
0016	1152	RETURN:	TAD HALT		
0017	3526		DCA I TO		
0020	7040		GMA 0		
0021	1000		TAD 0		
0022	3000		DGA 0		
0023	1152		TAD HALT		
0024	3400		DCA I 0		
0025	1152		TAD HALT		
0026	3531		DGA I FROM1		
0027	7001		IAC		
0030	1040		TAD CT		
0031	3040		DCA CT		
0032	1040		TAD CT		
0033	7640		SEA CLA		
0034	5437		JMP I ALOOP		
0035	5436		JMP I .+1		
0036	0346		SUP		
0037	0220		ALOOP, LUOP		
0040	0000		CT, 0		
0041	0000		CT1, 0		
0042	7763		M15, -15		
0043	7513		M265, -265		

```

0044 0215 /CR
0045 0212 /LF
0046 0212 /LF
0047 0306 /F = FROM
0050 0240 /SPACE
0051 0000 /X ADDRESS OF JMS INSTRUCTION
0052 0000 /X
0053 0000 /X
0054 0000 /X
0055 0240 /SPACE
0056 0324 /T
0057 0317 /U
0060 0240 /SPACE
0061 0000 /X
0062 0000 /X
0063 0000 /X
0064 0000 /X
0065 0215 /CR
0066 0212 /LF
0067 0377 /RUBOUT
0070 0250 /
0071 0324 /T
0072 0317 /U
0073 0251 /
0074 0240 /SPACE
0075 0275 /B
0076 0240 /SPACE
0077 0000 /X STORED ADDRESS
0100 0000 /X S/B FRMP1
0101 0000 /X
0102 0000 /X
0103 0215 /CR
0104 0212 /LF
0105 0377 /RUBOUT
0106 0250 /
0107 0000 /X ADDRESS=1 STORED
0110 0000 /X IN LOC 0 AT INTERRUPT
0111 0000 /X
0112 0000 /X
0113 0251 /
0114 0240 /SPACE
0115 0275 /B
0116 0240 /SPACE
0117 0000 /X CONTENTS OF ABOVE
0120 0000 /X ADDRESS
0121 0000 /X
0122 0000 /X
0123 0207 /END MARK
0124 0000
0125 7571

```

```

MSG1, 215
      212
      306
      240
INS1, 0
INS2, 0
INS3, 0
INS4, 0
      240
      324
      317
      240
      0
INS5, 0
INS6, 0
INS7, 0
INS8, 0
      215
      212
      377
      250
      324
      317
      251
      240
      275
      240
      0
      0
      0
      0
      215
      212
      377
      250
      0
      0
      0
      0
MSG9, 215
INS10, 212
INS11, 377
INS12, 250
      0
      0
      0
      0
      251
      240
      275
      240
      0
      0
      0
      0
      207
      0
      -207

```

/CONSTANTS

0126	0000	TO,	0
0127	0000	TOP1,	0
0130	0000	FROM,	0
0131	0000	FROM1,	0
0132	0000	FRMP1,	0
0133	2525	RANUM,	2525
0134	0003	THREE,	3
0135	7200	LIMLO,	-600
0136	0200	LIMH1,	-7600
0137	0001	ITON,	ION
0140	0000	SAVE,	0
0141	0000		0
0142	0000		0
0143	0000		0
0144	0007	MSK7,	7
0145	0260	TW6,	260
0146	0400	AER,	ER
0147	0327	ATYP,	TYPAC
0150	0330	ATYP1,	TYPAC*1
0151	0044	AMSG1,	MSG1
0152	7402	HALT,	HLT
0153	4526	JMP1,	JMS I TO

0154 0000
 0155 3000
 0156 1167
 0157 3001
 0160 1170
 0161 3002
 0162 1171
 0163 3003
 0164 1172
 0165 3573
 0166 5554
 0167 7200
 0170 1526
 0171 7041
 0172 7200
 0173 0200

PATCH, /RESTORE THEN GO AWAY

0 DCA 0
 TAD X1
 DCA 1
 TAD X2
 DCA 2
 TAD X3
 DCA 3
 TAD X4
 DCA I X5
 JMP I PATCH

X1, 7200
 X2, 1526
 X3, 7041
 X4, CLA
 X5, 200

/TAD I TO

*400

0400

EH,

0400

TAD I+4
 DCA I ATYP
 TAD FROM
 JMP I ATYP1
 I+1
 TAD SAVE
 AND MSK7
 TAD TH6
 DCA INS1
 TAD SAVE+1
 AND MSK7
 TAD TH6
 DCA INS2
 TAD SAVE+2
 AND MSK7
 TAD TH6
 DCA INS3
 TAD SAVE+3
 AND MSK7
 TAD TH6
 DCA INS4

0401 1204
 0402 3547
 0403 1130
 0404 5550
 0405 0405
 0406 1140
 0407 0144
 0410 1145
 0411 3051
 0412 1141
 0413 0144
 0414 1145
 0415 3052
 0416 1142
 0417 0144
 0420 1145
 0421 3053
 0422 1143
 0423 0144
 0424 1145
 0424 3054

TAD I+4
 DCA I ATYP
 TAD TO
 JMP I ATYP1
 I+1
 TAD SAVE
 AND MSK7
 TAD TH6
 DCA INS9
 TAD SAVE+1
 AND MSK7
 TAD TH6
 DCA INS0
 TAD SAVE+2
 AND MSK7
 TAD TH6
 DCA INS7
 TAD SAVE+3
 AND MSK7
 TAD TH6
 DCA INS8
 TAD I+4
 DCA I ATYP
 TAD I TO
 JMP I ATYP1
 I+1

0425 1231
 0426 3547
 0427 1126
 0430 5550
 0431 0432
 0432 1140
 0433 0144
 0434 1145
 0435 3061
 0436 1141
 0437 0144
 0440 1145
 0441 3062
 0442 1142
 0443 0144
 0444 1145
 0445 3063
 0446 1143
 0447 0144
 0450 1145
 0451 3064
 0452 1256
 0453 3547
 0454 1526
 0455 5550
 0456 0457

0457 1140
 0460 0144
 0461 1145
 0462 3077
 0463 1141
 0464 0144
 0465 1145
 0466 3100
 0467 1142
 0470 0144
 0471 1145
 0472 3101
 0473 1143
 0474 0144
 0475 1145
 0476 3102
 0477 7040
 0500 1000
 0501 3000

TAD SAVE
 AND MSK7
 TAD TW6
 DCA INS9
 TAD SAVE#1
 AND MSK7
 TAD TW6
 DCA INS10
 TAD SAVE#2
 AND MSK7
 TAD TW6
 DCA INS11
 TAD SAVE#3
 AND MSK7
 TAD TW6
 DCA INS12
 CMA
 TAD 0
 DCA 0

0502 1506
 0503 3547
 0504 1000
 0505 5550
 0506 0507
 0507 1140
 0510 0144
 0511 1145
 0512 3107
 0513 1141
 0514 0144
 0515 1145
 0516 3110
 0517 1142
 0520 0144
 0521 1145
 0522 3111
 0523 1143
 0524 0144
 0525 1145
 0526 3112
 0527 1333
 0530 3547
 0531 1400
 0532 5550
 0533 0534
 0534 1140
 0535 0144
 0536 1145
 0537 3117
 0540 1141
 0541 0144
 0542 1145
 0543 3120
 0544 1142

TAD #4
 DCA I ATYP
 TAD 0
 JMP I ATYP1
 #1
 TAD SAVE
 AND MSK7
 TAD TW6
 DCA MSG3
 TAD SAVE#1
 AND MSK7
 TAD TW6
 DCA INS13
 TAD SAVE#2
 AND MSK7
 TAD TW6
 DCA INS14
 TAD SAVE#3
 AND MSK7
 TAD TW6
 DCA INS15
 TAD #4
 DCA I ATYP
 TAD 0
 JMP I ATYP1
 #1
 TAD SAVE
 AND MSK7
 TAD TW6
 DCA INS16
 TAD SAVE#1
 AND MSK7
 TAD TW6
 DCA INS17
 TAD SAVE#2

0545 0144
0546 1145
0547 3121
0550 1143
0551 0144
0552 1145
0553 3122

1/11/68 3120.49

PAGE 9-1

AND MSK7
TAD TW6
DCA INS19
TAD SAVE+3
AND MSK7
TAD TW6
DCA INS19

0554 1151
0555 3124
0556 1524
0557 6046
0560 6041
0561 5360
0562 7201
0563 1124
0564 3124
0565 1524
0566 1125
0567 7640
0570 5356
0571 7604
0572 7710
0573 7402
0574 5014

TAD AMSG1
DCA WORK
TAD I WORK
TLS
TSF
JMP :-1
CLA IAC
TAD WORK
DCA WORK
TAD I WORK
TAD M207
SZA CLA
JMP TYPE
LAS
SPA CLA
HLT
JMP RETURN

TYPE,

/HALT ON ERROR

\$

THERE ARE NO ERRORS

SYMBOL TABLE

ALR	0146
ALOOP	0037
AMSG1	0151
AMSG2	0373
ATYP	0147
ATYP1	0150
BEGIN	0200
CHSCK	0305
CT	0040
CT1	0041
EH	0400
FKMP1	0132
FROM	0130
FROM1	0131
GETRAN	0225
GON	0204
GTRAN1	0261
HALT	0152
INST	0051
INS10	0100
INS11	0101
INS12	0102
INS13	0110
INS14	0111
INS15	0112
INS16	0117
INS17	0120
INS18	0121
INS19	0122
INS2	0052
INS3	0053
INS4	0054
INS5	0061
INS6	0062
INS7	0063
INS8	0064
ITON	0077
ITON	0137
JMP1	0153
LIMHI	0136
LIMLO	0135
LOOP	0220
LOOP1	0234
LPI	0356
MSG1	0044
MSG2	0071
MSG3	0107
MSK7	0144
M15	0042
M207	0125
M269	0043
PATCH	0154
RANUM	0133

SYMBOL TABLE

RETURN	0014
SAVE	0140
SUP	0346
THREE	0134
TU	0126
TUP1	0127
TW6	0145
TYPAC	0327
TYPE	0556
WORK	0124
X1	0167
X2	0170
X3	0171
X4	0172
X5	0173

SYMBOL TABLE

RETURN	0014
ALOOP	0037
CT	0040
CT1	0041
M15	0042
M265	0043
MSG1	0044
INS1	0051
INS2	0052
INS3	0053
INS4	0054
INS5	0061
INS6	0062
INS7	0063
INS8	0064
MSG2	0071
INS9	0077
INS10	0100
INS11	0101
INS12	0102
MSG3	0107
INS13	0110
INS14	0111
INS15	0112
INS16	0117
INS17	0120
INS18	0121
INS19	0122
WORK	0124
M207	0125
TU	0126
TOP1	0127
FROM	0130
FROM1	0131
FRMP1	0132
RANUM	0133
THREE	0134
LIMLO	0135
LIMHI	0136
IION	0137
SAVE	0140
MSK7	0144
TW6	0145
AER	0146
ATYP	0147
ATYP1	0150
AMSG1	0151
HALT	0152
JMP1	0153
PATCH	0154
X1	0167
X2	0170
X3	0171

SYMBOL TABLE

X4	0172
X5	0173
BEGIN	0200
GON	0204
LOOP	0220
GETRAN	0225
LOOP1	0254
GTRAN1	0261
CHSCK	0305
TYPAC	0527
SUP	0346
LPI	0556
AMS62	0373
ER	0400
TYPE	0556