EAE 3A

1. IDENTIFICATION

1.1 Maindec 801-3A

1.2 PDP-8 Instruction Test (EAE Type 182)—Part 3A

1.3 July 19, 1965

2. ABSTRACT

This program is a test of the Extended Arithmetic Element Type 182. The following instructions are tested:

MQL, MQA, SHL, LSR, ASR, NMI, SCA

An attempt is made to detect and isolate errors to its most basic fault and to the minimum number of logic cards. Multiply and divide are tested Maindec 801-3B.

3. REQUIREMENTS

3.1 Storage

Memory Locations 10_8 - 5410_8

3.2 Subprograms and/or Subroutines

High RIM Loader, High Binary Loader

3.3 Equipment

PDP-8, EAE Type 182, keyboard reader and teleprinter.

4. USAGE

4.1 Loading

4.1.1 If the Binary Loader beginning at 7777_8 is in memory, go to paragraph 4.1.2; otherwise the RIM Loader beginning at 7756_8 and/or the Binary Loader must be loaded into memory.

The PDP-8 Instruction Test—Part 3A, may now be loaded as follows:

4.1.2 Set 7777_8 in the SWITCH REGISTER.

4.1.3 Press LOAD ADDRESS key.

4.1.4 Place Instruction Test Part 3A in the keyboard reader.

4.1.5 Press START key on the operator console.

4.1.6 Engage the keyboard reader.

4.2 Calling Sequence (Not Applicable)

4.3 Switch Settings

Set the SWITCH REGISTER keys to 5000_8 before starting the program.

4.4 Startup and/or Entry

4.4.1 The starting address of the PDP-8 Part 3A Instruction Test is 0200_8.

4.4.2 Set 0200_8 in the SWITCH REGISTER keys and press the LOAD ADDRESS key.

4.4.3 Set 5000_8 in the SWITCH REGISTER keys and press the START key.

These initial switch settings will cause the program to print error messages and halt on an error. See paragraph 4.6 for other switch options.

4.5 Errors in Usage

Errors detected by the program cause the following:

Print error information
Halt on error

The following are typical examples of error print information:

4.5.1

	NMIT	C(AC)	C(MQ)
		000000000000	010101010101
	NMI	010101010100	000000000000
	SCAT	000000001100	
	SCA	000000001100	

NMIT = Normalize and step counter test. Original C(AC) and C(MQ).

NMI = C(AC) and C(MQ) after the NMI instruction was executed.

SCAT = The correct count of the step counter after the normalize instruction was executed.

SCA = The actual count in the step counter as read into the AC by the SCA instruction after the normalize instruction was executed.

Note that Bit 11 of AC in error. C(AC) should equal 2525_8.

4.5.2

SHIFT	11	
C(MQ)	001111001011	
	C(AC)	C(MQ)
SHL0	000011100101	100000000000
LSR0	000000000000	000111001011

SHIFT 11 = Number of shifts to be executed.

C(MQ) = Original C(MQ). (The original C(AC) equal all 0's.)

SHL = C(L), C(AC), and C(MQ) after the SHL instruction was executed.

LSR = C(L), C(AC), and C(MQ) after the results of the SHL instructions were acted upon by the LSR instruction.

Note that Bit 3 of AC after a SHL instruction is in error. C(AC) should equal 0745_8.

4.5.3

SHIFT 2		
C(AC) 1	011111111111	
	C(AC)	C(MQ)
ASR 1	111111111111	000000000000

SHIFT 2 = Number of shifts to be executed.

C(AC) = Original C(AC) and C(L). (The original C(MQ) equal to all 0's.)

ASR = C(AC), C(L), and C(MQ) after the ASR instruction was executed.

Note that C(L) should equal a 0, C(AC) should equal 0777_8, C(MQ) should equal 6000_8 after the ASR instruction was executed.

4.5.4

```
MQLT
AC     1     000000000011
0-AC   1     000000000001
```

MQLT = MQL instruction test with a link set to a 1.

AC = The original C(AC) and C(L).

0-AC = C(AC) and C(L) after the MQL instruction was executed.

Note that bit 11 of AC should equal 0.

4.5.5

```
MQLT1
AC     0     000000000001
0-AC   0     000000000001
```

MQLT1 = MQL instruction test with a link cleared to a 0.

AC = The original C(AC) and C(L).

0-AC = C(AC) and C(L) after the MQL instruction was executed.

Note that bit 11 of AC should equal 0.

4.5.6

```
MQAT
AC     1     000000000001
MQL)
MQA)   1     000000000000
```

MQAT = MQA, MQL instructions test with a link set to a 1.

AC = Original C(AC) and C(L).

MQL MQA = The C(AC) and C(L) after the execution of an MQL instruction.

Note that bit 11 of AC should equal a 1.

4.5.7

```
MQAT1
AC     0     100000000000
MQL)
MQA)   0     011111111111
```

MQAT1 = MQA, MQL instructions test with a link set to a 1.

AC = Original C(AC) and C(L).

MQL MQA = The C(AC) and C(L) after the execution of an MQL instruction followed by an MQA instruction.

Note that C(AC) should equal 4000_8.

4.5.8 MQAT2

AC 1	111111111110
MQ	000000000001
MQVAC 1	000000000000

MQAT2 = MQA instruction test.

AC = Original C(AC) and C(L).

MQ = Original C(MQ).

MQVAC = The C(AC) and C(L) after the execution of an MQA instruction.

Note that C(AC) should equal 7777_8.

4.5.9 MQAT3

AC 0	111111111110
MQ	000000000001
MQVAC 0	000000000000

MQAT3 = MQA instruction test.

AC = Original C(L) and C(AC).

MQ = Original C(MQ).

MQVAC = The C(AC) and the C(L) after the execution of an MQA instruction.

Note that C(AC) should equal 7777_8.

4.5.10 The following table contains the test mnemonic, starting address, error halt address, and instructions tested.

Mnemonic	Instructions	Starting Address	Error Halt
MQLT	MQL	0200	0240
MQLT1	MQL	0427	0500
MQAT	MQL, MQA	0600	0661
MQAT1	MQL, MQA	1000	1061
MQAT2	MQA	1200	1261
MQAT3	MQA	1400	1460
STEST	SHL, LSR	1600	1671
STEST1	SHL, LSR	2400	2723
STEST2	ASR	3200	3355
STEST3	ASR	4000	4060
NORMT	NMI, SCA	4200	4261
NORMT1	NMI, SCA	5000	5143
NORMT2	NMI	5200	5324

The module table (see paragraph 11.4) should be used after a visual check of the program error printout.

4.5.11 Table 1 should be used on testing MQAT, MQAT1, etc, after determining which bit of the AC and/or MQ is in error.

4.5.12 Table 1 should also be used on testing STEST1, STEST2, etc. After determining which two bits may be in error in the AC and/or MQ, reference Table 1 for the module identification of the two bits in question.

4.5.13 Table 2 should be used in conjunction with Table 1 when a step count error may exist.

4.6 Recovery from such Errors

Press CONTINUE or set up one of the following switch register controls followed by pressing the CONTINUE key.

4.6.1 SWITCH REGISTER keys 0, 1, 2, 3 are used for error recovery as follows:

Switch 0 = Halt on error.
Switch 1 = Scope mode (repeat pattern and/or test).
Switch 2 = Print error information.
Switch 3 = Inhibit exiting current test.

4.6.2 Multiple switch settings are as follows:

Switches 0, 2 = Print error information and halt at error stop.
Switches 1, 2 = Scope mode and print error information.
Switches 0, 2, 3 = Inhibit exiting current test, print error information and halt at error stop.
Switches 0, 1, 2 = Scope mode, print error information, and halt at error stop.

5. RESTRICTIONS

Before running this test, all basic PDP-8 processor tests should have been run successfully.

6. DESCRIPTION

6.1 Discussion

The PDP-8 Instruction Test—Part 3A tests the following extended arithmetic element instructions:

MQL, MQA, SHL, LSR, SCA, ASR, NMI

The extended arithmetic element is tested using patterns necessary to detect and isolate errors to the basic cause and minimum logic card involved. If a failure does occur, the test will stop at a predetermined error halt.

Two look-up tables are provided for error repair methods containing the following information:

AC, MQ, and SC bit numbers
Processor logic drawing number
Logic board type
Logic board location

The program starts at memory address 0200_8 and will print "3A" at the completion of the test and jump back to location 0200 to repeat the entire test. The test is assembled in binary format.

7. METHODS (Not Applicable)

8. FORMAT (Not Applicable)

9. EXECUTION TIME

35 sec.

10. PROGRAM

10.4 Program Listing

```
                /EAE PART 3A OF INSTRUCTION TEST

                CAM=7621
                SCA=7441
                NMI=7411
                ASR=7415
                MQL=7421
                MQA=7501
                LSR=7417
                SHL=7413

                *0020
0020  0000  GEN,    0
0021  2125          ISZ Z GENX
0022  5420          JMP I GEN
0023  7604          CLA OSR                 /TEST SW 3
0024  7106          RTL CLL
0025  7006          RTL
0026  7430          SZL
0027  5431          JMP I BACK
0030  5432          JMP I NEXT
0031  0000  BACK,    0000
0032  0000  NEXT,    0000
0033  0427  XMQLT1,           MQLT1
0034  0600  XMQAT,  MQAT
0035  1000  XMQAT1,           MQAT1

0036  0000  CRLF,   0
0037  7240          CLA CMA
0040  0130          AND CR                  /CR
0041  4046          JMS PRXLOP
0042  7240          CLA CMA
0043  0131          AND LF                  /LF
0044  4046          JMS PRXLOP
0045  5436          JMP I CRLF

0046  0000  PRXLOP,          0
0047  6046          TLS                     /PRINT LOOP
0050  6041          TSF
0051  5050          JMP .-1
0052  7200          CLA
0053  5446          JMP I PRXLOP

0054  0000  PLINK,  0
0055  7240          CLA CMA
0056  0143          AND LINK                /LINK
0057  4061          JMS ONZER
0060  5454          JMP I PLINK
```

11. DIAGRAMS

11.4 Error Graphs for Functions

11.4.1 Table Number 1

AC, MQ				
Bit 11	R212	R210	R111	R111 (Module Type)
	PF22	PA18 PB18	PE21	PE20 (Module Position)
	D182-0-2	D802	D182-0-2	D182-0-2 (Drawing No.)
Bit 10	R212	R210	R111	R111
	PF22	PA17 PB17	PE21	PE20
	D182-0-2	D802	D182-0-2	D182-0-2
Bit 9	R212	R210	R111	R111
	PF21	PA16 PB16	PE21	PE20
	D182-0-2	D802	D182-0-2	D182-0-2
Bit 8	R212	R210	R111	
	PF21	PA15 PB15	PE19	
	D182-0-2	D802	D182-0-2	
Bit 7	R212	R210	R111	R111
	PF20	PA14 PB14	PE18	PE19
	D182-0-2	D802	D182-0-2	D182-0-2
Bit 6	R212	R210	R111	
	PF20	PA13 PB13	PE18	
	D182-0-2	D802	D182-0-2	
Bit 5	R212	R210	R111	
	PF19	PA12 PB12	PE18	
	D182-0-2	D802	D182-0-2	

Bit 4	R212	R210	R111
	PF19	PA11 PB11	PE17
	D182-0-2	D802	D182-0-2
Bit 3	R212	R210	
	PF18	PA10 PB10	
	D182-0-2	D802	
Bit 2	R212	R210	
	PF18	PA09 PB09	
	D182-0-2	D802	
Bit 1	R212	R210	
	PF17	PA08 PB08	
	D182-0-2	D802	
Bit 0	R212	R210	R111
	PF17	PA07 PB07	PE17
	D182-0-2	D802	D182-0-2
AC, Link	R210		
	PA06 PB06		
	D802		

11.4.2 Table Number 2

SC Bit 0	R205	R111	
	PF25	PE19	
	D182-0-2	D182-0-2	
Bit 1	R205	R111	
	PF25	PE19	
	D182-0-2	D182-0-2	

Bit 2	R205	R111
	PF27	PE20
	D182-0-2	D182-0-2
Bit 3	R205	R111
	PF27	PE20
	D182-0-2	D182-0-2
Bit 4	R205	R111
	PF27	PE20
	D182-0-2	D182-0-2