EAE I

B2

IDENTIFICATION

Product Code:	Maindec 9A-D0GA-D
Product Name:	PDP-9 Extended Arithmetic Element Part 1
Data Reviewed:	October 4, 1967
Maintainer:	Diagnostics Group
Author:	Keith Nelson

2. ABSTRACT

Part 1 of the PDP-4/7/9 EAE Diagnostic verifies correct operation of all EAE operations, except multiplies and divides. Part 1 is written in three logical sections. Part 1 Section 1 is the EAE Set-Up Test and verifies that all set-up operations except LACS operate correctly. Part 1 Section 2 is the Shift Counter (LACS is verified) and Basic Shift Test and verification that the AC and MQ will each shift left 1 and shift right 1 all combinations of 18 bits. Part 1 Section 3 is the Random Data, Normalize, and Interrupt Test verifying that random data will shift left and right 0 to 44_8 places, that normalize will "stop shift" on negative and positive data, and that the teleprinter flag will cause a break after an EAE operation. Hardware malfunctions detected by the program result in an error on the teleprinter.

3. REQUIREMENTS

3.1 Storage

CAL subroutine	00020-00027
AC contents initial	00030
MQ contents initial	00031
Link initial	00032
SC of shift instructions	00033
AC contents as result	00034
MQ contents as result	00035
Link as result	00036
SC of LACS instruction	00037
Halt and/or Scope Loop subroutine	00040-00057
Halt and/or Repeat Sequence subroutine	00060-00077
Set-Up Test	00100-01000 (approx.)
Error Typeout subroutine	
Error texts and program constants	01035-02100 (approx.)
SC and Basic Shift Test	02200-04600 (approx.)
Random Data and Normalize	05000-06400 (approx.)

3.2 Subprograms and/or Subroutines

PDP-4/7/9 Teletype Output Package
(ASCII tape 2A of this test)

3.3 Equipment

Minimum configuration PDP-4/7/9 with EAE option installed.

4. USAGE

4.1 Loading

Normal binary loading procedures are to be used.

4.2 Calling Sequence

Part 1 Section 1 must run in its entirety and at all margins before running Part 1 Section 2.

Part 1 Section 2 must run in its entirety and at all margins before running Part 1 Section 3.

4.3 Switch Settings

4.3.1 AC switches = 0 or down. With all AC Switches down the program results in the following:

(1) All hardware malfunctions detected by the program result in an error typeout on the teleprinter.

(2) At the completion of an error typeout the processor halts.

(3) The program repeats whichever section of the test it was started in and sequences from each sub-test of that section to the next without halting.

4.3.2 AC switches = 1 or up

SW #	Operation	Description
0	Delete error typeouts	The program will not type out error messages and will not error halt (see also SW0 and 7, Ring Bell on Error).
1	Halt after EAE operation Processor halts at address 0046 (AC) = S.A. to set up last operation	The processor halts after each EAE operation is initiated and its results are verified. (Note: Press CONTINUE to proceed.)
2	Repeat EAE operation (Scope Loop)	The program repeats the last EAE operation. If SW2 is set during an error typeout or halt, the program repeats the operation that caused the error (Note: SW1 is tested before SW2.)
3	Halt after EAE sequence Processor halts at address 0066 (AC) = S.A. of last sequence	The processor halts after each sequence of testing an EAE operation; i.e., after testing that the MQ will complement all patterns, the processor halts.

SW #	Operation	Description
4	Repeat EAE sequence	The program repeats the last sequence of testing an EAE operation; i.e., the program repeats the LEFT SHIFT ALL COMBINATIONS and does not proceed to RIGHT SHIFT ALL COMBINATIONS. (Note: The program tests SW3 before SW4.) In the Random Data Left and Random Data Right routines SW4 causes the program to repeatedly shift a single pair of random numbers 0 to 44_8 places.
5	Cycle all sections	At the completion of 1 pass through the Set-Up Test the program proceeds to the SC and Basic Shift Test. At the completion of 1 pass through the SC and Basic Shift Test the program proceeds to the Random Data and Normalize Test. At the completion of 1 pass through Random Data and Normalize Test the program repeats the Set-Up Test.
6	Type end of section	At completion of 1 pass through each of the sections a character is typed on the teleprinter as follows: Set-Up Test / SC and Basic Shift Test , Random Data and Normalize *
7	Delete error halt	The processor will not halt after error typeouts.
0 & 7	Ring bell on error	SW0 and SW7 both up. Error typeouts and halts are deleted and the "bell" on the teleprinter is rung (to be used to determine marginal voltage limits, eliminates waiting for long typeouts).

4.4 Start Up and/or Entry

4.4.1 Start Up, Set-Up Test

Set AC switches = 000000

Set ADDRESS = 0100

Press I/O Reset

Press START

Processor halts at 0101 with MQ = 777777

Set ADDRESS = 0102

Press I/O Reset

Press START

Program reads C(MQ) into the AC and tests for 0, then proceeds to rest of test.

NOTE: This section of Part 1 must run at all margins before running Section 2.

4.4.2 Start Up, SC and Basic Shift Test

Set AC switches = 000000

Set ADDRESS = 2200

Press I/O Reset

Press START

Processor halts at 2204 AC = 200000
SC = 77

Set ADDRESS = 2205

Press START

Program reads C(SC) into the AC and tests for 0, then proceeds to rest of test.

NOTE: This section must run at all margins before running Section 3.

4.4.3 Start Up Random Data and Normalize Test

Set AC switches = 000000

Set ADDRESS = 5000

Press START

NOTE: This section must run at all margins before running EAE Part 2.

4.5 Errors in Usage

Hardware malfunctions detected by the program will result in an error typeout on the teleprinter and a processor halt (see section 4.3.2, SW0 and SW7).

4.5.1 Error Typeout Format

All error typeouts are in standard formats and include the following information:

(1) An address that may be used to determine which test the program was in at the program was in at the time the error was detected

(2) A mnemonic describing the operation being tested

(3) The initial condition of registers pertinent to the failure

(4) The expected results of the operation being tested if they are not easily determined from the initial conditions and operation

(5) The resultant register contents that are pertinent to the failure

A common typeout routine called ERROR generates all error typeouts. The first line of every error typeout is the contents of memory register ERROR or the address + 1 of the JMS ERROR instruction.

The second line of every typeout is the mnemonic describing the operation being tested (see paragraph 4.5.2 for definitions of mnemonics used).

The third line of a typeout may be another address. In this case the second address typed should be used to determine which test failed. (Operations such as LRS or LLSS each have common error routines.)

The next information typed is a header to format the typeouts of the contents of pertinent registers. One of five headers may be used for any typeout.

The abbreviations used by the headers are as follows:

Abbr.	Meaning
L	The information under this column is the contents of the link.
C(AC)	The information under this column is the contents of the accumulator.
C(MQ)	The information under this column is the contents of the MQ register.
SC	The information under this column is the contents of the shift counter or the SC portion of shift instructions.
START	The information in this line is the initial condition of pertinent registers.

The five headers are as follows:

```
          C(AC)
START
          C(AC)     C(MQ)
START
          L         C(AC)     C(MQ)
START
          SC        C(AC)
START
L         C(AC)     C(MQ)
```

4.5.2 Error Typeout Mnemonics

Mnemonic	Description
EAENOP	EAE instruction with no other operation specified.
EAECLA	EAE. Clear the accumulator.
CLQ	Clear the MQ register.
CMQ	Complement the MQ register.
ORMQAC	Inclusive OR the MQ to the AC and place the results in the AC.
AC0TOL	Set AC bit 0 into the link.
ORACMQ	Inclusive OR the AC to the MQ and place the results in the MQ (and in test ACORMQ clear the AC).
LACQ	Clear the AC, then MQ 1's to the AC.
LLS	Long left shift.
LLSS	Long left shift signed.
LRS	Long right shift.
LRSS	Long right shift signed.
LMQ	Clear the MQ, then AC 1's to the MQ.
ABS	Complement the AC if it is negative.
CLR Δ SC	Clear the step counter (START).
LACS	Clear the AC and step counter; 1's to the AC.
NORM	Normalize the AC and MQ.
NORMS	Normalize signed.
ALS	Accumulator left shift.
PAT	Pattern being tested.
COR	Results expected from the operation being tested.
INCO	Erroneous results of the operation.

4.5.3 Error Typeout Examples

The following are examples of error typeouts. The addresses indicated by these typeouts should not necessarily be taken as true representations:

Example 1: Complement the MQ Failure

Example			Explanation
000226			JMS ERROR is at 00225
CMQ			Operation is complement the MQ
	C(AC	C(MQ	Header
START	000000	000000	Initial conditions
CMQ	000000	767777	Contents of the AC and MQ after CMQ was executed.

Note: Examine the MQ indicators to be sure they agree with the typeout. If the MQ as indicated does not agree with a typeout, an error was present in MQ 1's to the AC. This is true of all error typeouts that include the MQ as an end condition.

Example 2: EAE NOP AC Failure

Example		Explanation
000135		JMS ERROR is at 00134
EAENOP		Operation is NOP 640000
	C(AC)	Header
START	777777	Initial condition of the AC
EAENOP	000000	Contents of the AC after the NOP was executed

Example 3: AC Sign to Link Failure

Example				Explanation
000455				JMS ERROR is at 00454
AC0TOL				Operation is AC bit 0 to link
	L	C(AC)	C(MQ)	Header
START	1	400000		Initial conditions MQ not pertinent
AC0TOL	0	400000		State of the LINK and AC after the operation was executed

Example 4: AC to MQ to AC Failures

Example				Explanation
000526				JMS ERROR is at 00525
ORACMQ				Operation is AC 1's to MQ
	C(AC)	C(AC)		Header
START	000000	000000		Initial register states
ORACMQ	000000	000000	COR	Expected results
LACQ	000000	040000	INCO	The contents of the AC after ORACMQ and the contents of the MQ as indicated by a LACQ instruction
000526				
ORACMQ				
	C(AC)	C(MQ)		
START	005000	000000		
ORACMQ	000000	005000	COR	
LACQ	000000	004000	INCO	

Note: Again, the contents of the MQ as indicated by the MQ indicators may not necessarily agree with the MQ contents as typed.

Example 5: Step Counter Error

Example				Explanation
002530				JMS ERROR is at 02527
SC ERROR				One of the SC tests failed
002262				JMS SCERR is at 02261
	SC	C(AC)		Header
START	00	200000		Initial register status
NORM	01			Instruction used to set the SC
SET SC	76			NORM 01 should set the SC to 76
SC + 1	77	COR		SC should increment to 77
LACS	67	INCO	200000	Contents of the SC as read to the AC by a LACS instruction and the contents of the AC after the NORM instruction

Example 6: ALS (Accumulator Left Shift) Failure

Example			Explanation
003123			JMS ERROR is at 03122
ALS	05		ALS instruction 5 places
003076			JMS ALSERR is at 03075
L	C(AC)	C(MQ)	Header
1	777776	PAT	Pattern being tested
1	777777	RESULT	Results in AC after the shift
LACS	00		Shift counter read back to the AC

Example 7: Long Left Shift

Example				Explanation
003673				JMS ERROR is at 03672
LLS	01			Long left shift 1 place
003507				JMS LLSERR is at 03506
L	C(AC)	C(MQ)		Header
1	777777	777737	PAT	Initial register states
1	777777	777377	RESULT	Registers at completion of shift
LACS	00			SC as read back to the AC

Example 8: Long Left Shift Signed

Example				Explanation
003716				JMS ERROR is at 03715
LLSS	03			Long left shift signed 3 places
005075				JMS LRSSER is at 05074
L	C(AC)	C(MQ)		Header
0	456701	234567	PAT	Pattern being tested
	567012	345677	COR	Expected results
1	567012	347677	INCO	L, AC, and MQ after the shift
LACS	00			SC as read back to the AC

Example 9: Long Right Shift

Example				Explanation
004600				JMS ERROR is at 004577
LRS	01			Long right shift 1 place
004537				JMS LRSER 1 is at 004536
L	C(AC)	C(MQ)		Header
1	402101	402101	PAT	Pattern being tested
	601200	601200	COR	Expected results
1	601200	601000	INCO	AC and MQ after completion of the shift
LACS	00			SC as read to the AC after completion of the shift

Example 10: Random Data Sequenced

Example				Explanation
005501				JMS ERROR is at 005500
RANDOM DATA SEQUENCED 02				Random sequence 2
005301				JMS SEQCOM is at 005300
L	C(AC)	C(MQ)		Header
0	045670	123450	START	Pattern sequenced
0	045630	123450	RESULT	L, AC, and MQ after shift sequence
LACS	00			SC after shift sequence

```
Note:  Sequence 2 is:  LLSS  03
                       LRS   06
                       LLSS  06
                       LRS   03
```

The AC and MQ results should equal the AC and MQ at START. This is true of all of the Random Data Sequences.

Example 11: Normalize

Example				Explanation
006217				JMS ERROR
NORM	01			Normalize SC = 1
005766				JMS NORMER is at 05765
L	C(AC)	C(MQ)		Header
0	200000	000000	PAT	Pattern being tested
0	400000	000000	RESULT	L, AC, and MQ after NORM
LACS	77	COR		SC expected after the NORM
LACS	00	RESULT		SC read back to the AC

Example 12: Interrupt Failure

Example	Explanation
006310	JMS ERROR is at 06307
NO PROGRAM INTERRUPT	Error is no interrupt
EAE NOP	Instruction tested
006305	Address of NOP instruction

4.6 Recovery From Such Errors

4.6.1 General

At the completion of an error typeout the processor halts. One of the following operations may be necessary if more information about the failure is required to repair the malfunction:

1. Repeat the exact operation that detected the failure (possibly for a scope loop).
2. Continue normally in the test to generate more information about the failure.
3. Repeat the sequence of operations or data patterns that detected the error.

AC switch control is built into the program to allow for any of these operations. Assuming the processor has halted after an error typeout, the operations may be accomplished as follows:

1. Repeat same operation

Set AC switch 2 up or to a 1
Press CONTINUE

Note that AC SW0 allows deletion of error typeouts for a scope loop.

2. Continue normally

Press CONTINUE

3. Repeat Sequence

Set AC switch 4 up or to a 1
Press CONTINUE

In the Random Data Tests, switch 4 a 1 causes the same pair of random numbers to be repeatedly shifted 0 to 44_8 places. This is useful in determining which shift the random data first fails.

4.6.2 To Determine Area in Program that Failed

4.6.2.1 From Error Typeouts

Each error typeout includes an address typeout that may be used to determine the exact test routine that detected the error. Some of the typeouts include an address that points at a common error routine for that type of error and a second address that points at the test routine. (Section 4.5.3, example 3 has only one octal typeout before the header and example 5 has two. The second octal typeout in example 5 (002262) determines which SC test failed.) Determine which address to use, go to the numerically sorted program labels (section 10.4.1) and find the program labels with addresses lower and higher than the one typed. The last program label with an address lower than the one typed is in the test routine that failed.

4.6.2.2 From CAL Routine

This test program includes a halt at address 00026 that indicates a CAL instruction was executed. Pressing CONTINUE at this point causes the processor to CAL at address 00027. At the time of the first HALT the contents of the AC indicate the contents of address 00020 after the CAL or the address + 1 of the CAL. The approximate area of the test program that was being executed may be determined by examining the following memory addresses.

Address	Contents Indicate
00040	Address +1 or +2 of last JMS SWITCH
00057	Starting address of last SCOPE LOOP
00060	Address +1 or +2 of last JMS SWITCH
00077	Starting address of last TEST SEQUENCE

By comparing the contents of these memory locations with the numerically sorted symbol list, the test routine (at the time of a CAL, hang up, or program wipeout) that was being executed may be determined.

5. RESTRICTIONS (Not Applicable)

6. DESCRIPTION

6.1 Discussion

6.1.1 General

The PDP-4/7/9 EAE Diagnostic Part 1 verifies correct operation of all EAE operations except multiplies and divides. Part 1 itself is written in three logical sections as follows:

Section 1: Set-Up Test

Verifies correct operation of all EAE set-up operations except LACS.

Section 2: SC and Basic Shift Test

Verifies correct operation of the SC and LACS instruction and verifies that the AC and MQ will shift left and right 1 place all combinations of 18 bits.

Section 3: Random Data and Normalize Test

This section of Part 1 verifies that the AC and MQ will shift random data left and right 0 to 44_8 places, that the NORM and NORMS instructions operate correctly, and that the processor interrupts after an EAE operation.

The above sections are to be used incrementally. That is, Section 1 must operate at all margins before Section 2 is run. Section 2 must run at all margins before Section 3 is run.

6.1.2 Test Descriptions

6.1.2.1 Set-Up Test

The Set-Up Test incrementally verifies correct operation of all of the EAE set-up instructions except LACS.

The sequence of testing is as follows:

Test Mnemonic	Operation(s) Tested
SETUP	Does CMQ set MQ = 0's to 1's Do all MQ indicators light (visual)
EAERMQ	Does START clear the MQ Does MQ = 0's to AC = 0's
NOPAC	Does EAE NOP not clear the AC
EAECAC	Do EAE and bit 8 clear the AC
EAECLQ	Does bit 5 clear the MQ
MQITAC	Does bit 16 with MQ = 1's set AC to 1's
NOPACI	Does EAE NOP with MQ = 1's alter the AC
NOPMQ	Does EAE NOP with MQ = 1's alter the MQ
NOPMQI	Does EAE NOP with AC = 1's alter the MQ
NOPLNK	Does EAE NOP alter the link

Test Mnemonic	Operation(s) Tested
QONEAC	Does MQ = 1's inclusive OR to AC = 1's
EAESLK	Do EAE and bit 4 get AC sign to link
NOPLK1	Does EAE NOP alter the MQ with link = 1
ACORMQ	Does AC inclusive OR all patterns to MQ = 0's and MQ to AC all patterns
ACLMQ	Does the LMQ instruction operate as specified
COMPMQ	Will the MQ complement all patterns
ACONEQ	Will the AC = 1's inclusive OR to MQ = 1's
EAEABS	Does the ABS instruction operate as specified

6.1.2.2 SC and Basic Shift Test

The SC and Basic Shift Test incrementally verifies correct operation of the SC (including the LACS instruction) and the left and right shifts. The SC Test assumes that a NORM instruction with the AC = 200000 generates a stop shift.

The sequence of testing is as follows:

Test Mnemonic		Operation(s) Tested
SCTST1	(1)	Does NORM "stop shift" with AC = 200000 (visual) SC is set to 77
	(2)	Does START clear the SC
	(3)	Does LACS get SC = 0's to the AC
NOPSC		Does EAE NOP alter the SC = 0's
SCTO76	(1)	Will the SC set to 76 and + 1 to 77
	(2)	Will LACS read SC = 77 to the AC
SCTO74		Will the SC set to 74 and + 1 to 75
SCTO70		Will the SC set to 70 and + 1 to 71
SCTO60		Will the SC set to 60 and + 1 to 61
SCTO40		Will the SC set to 40 and + 1 to 41
SCTO00		Will the SC set to 00 and + 1 to 01
SCTO01		Will the SC set to 01 and + 1 to 02
SCTO03		Will the SC set to 03 and + 1 to 04
SCTO07		Will the SC set to 07 and + 1 to 10 (Is "high count" generated?)

Test Mnemonic	Operation(s) Tested
SCTO17	Will the SC set to 17 and + 1 to 20
SCTO37	Will the SC set to 37 and + 1 to 40
SCTO77	Will the SC set to 77 and + 1 to 00
NOPSC1	Does EAE NOP alter SC = 77
ALSZER	Does ALS with SC = 00 "stop shift"
ALS01	Does ALS 1 place shift AC = 0's
ALSLNK	Does link get to AC17 on an ALS 1 place
LNKALS	Does bit 0 of the AC not go to the link on an ALS 1 place
ALSMQT	Does ALS alter the MQ Does MQ0 not go to AC17
HSALS	Will ALS shift the AC 1 to 18 places bit and no-bit
LLSTS1	Will the AC/MQ shift 0's place left
LLSTS2	Does link go to MQ17 on an LLS
LLSACT	(1) Does link not go to AC17 on an LLS (2) Does MQ0 go to AC17 on an LLS
LLSTS3	Does each bit of the MQ = 1 shift left 1 place (1 bit at a time = 1)
LLSTS4	Does each bit of the MQ = 0 shift left 1 place (1 bit at a time = 0)
LLSTS5	Will MQ/AC shift a 1 bit 1 to 44_8 places left
LLSTS6	Will MQ/AC shift a 0 bit 1 to 44_8 places left
LRSTS1	Will AC/MQ shift right 1 all 0's
LRSTS2	Does link go to AC0 on an LRS
LRSTS3	Does AC17 go to MQ0 on an LRS
LRSTS4	Does AC17 not go to link on an LRS
LRSTS5	Will AC/MQ shift a 1 bit from each position right 1 place (1 bit at a time)
LRSTS6	Will AC/MQ shift a 0 bit right 1 place (1 bit at a time)
LRSTS7	Will AC/MQ shift 1 bit (AC0) right 1 to 44_8 places
LRSTS8	Will AC/MQ shift a 0 bit (AC0) right 1 to 44_8 places

Test Mnemonic	Operation(s) Tested
LLSSEQ	Will the AC and MQ each shift left 1 place every combination of 18 bits
LRSSEQ	Will the AC and MQ each shift right 1 place every combination of 18 bits

6.1.2.3 Random Data and Normalize Test

The Random Data and Normalize Test verifies that the AC/MQ will shift left and right random data 0 to 44_8 places, that the NORM and NORMS instructions operate as specified, and that the processor interrupts after an EAE instruction.

The sequence of testing is as follows:

Test Mnemonic	Operation(s) Tested
RANSHF	Generates 4096 pairs of random numbers, 1 for the AC and 1 for the MQ. Each pair of random numbers is shifted left signed (LLSS) 0 to 44_8 places, and the results are tested against a table generated by 44 left shift 1 place.
RANRIT	Generates 4096 pairs of random numbers 1 for the AC and 1 for the MQ. Each pair of random numbers is shifted right (LRS) 0 to 44_8 places, and the results are tested against a table generated by 44 shift right 1 place.
RANSEQ	Generates 4096 pairs of random numbers 1 for the AC and 1 for the MQ. Each pair of random numbers is used by RANSQ0 to RANSQ8. After each sequence the AC and MQ should equal their starting patterns.
RANSQ0	Bit 0 of AC = bit 17 of MQ. Random numbers are sequenced 1 left signed, 2 right, 2 left signed, 1 right.
RANSQ1	Bit 0 and 1 of AC = bit 16 and 17 of MQ. Sequence is: 2 right signed 4 left signed 4 right 2 left signed
RANSQ2	Bits 0 to 2 of AC = bits 15 to 17 of MQ. Sequence is: 3 left signed 6 right 6 left signed 3 right

Test Mnemonic	Operation(s) Tested
RANSQ3	Bits 0 to 3 of AC = bits 14 to 17 of MQ. Sequence is: 4 right signed; 8 left signed; 8 right; 4 left signed
RANSQ4	Bits 0 to 4 of AC = bits 13 to 17 of MQ. Sequence is: Left 5 signed; Right 10; Left 10 signed; Right 5
RANSQ5	Bits 0 to 5 of AC = bits 12 to 17 of MQ. Sequence is: Right 6 signed; Left 12 signed; Right 12; Left 6 signed
RANSQ6	Bits 0 to 6 of AC = bits 11 to 17 of MQ. Sequence is: Left 7 signed; Right 14; Left 14 signed; Right 7
RANSQ7	Bits 0 to 7 of AC = bits 10 to 17 of MQ. Sequence is: Right 8 signed; Left 16 signed; Right 16; Left 8 signed
RANSQ8	Bits 0 to 8 of AC = bits 9 to 17 of MQ. Sequence is: Left 9 signed; Right 18; Left 18 signed; Right 9
NRMLZE	Does NORMS get AC sign = 0 to link
NRMLZ1	Does NORMS get AC sign = 1 to link
NRMLZ2	Will NORM "stop shift" with AC0 $\neq$ AC1, AC0 = 1, AC1 = 0, or AC0 = 0, AC1 = 0

Test Mnemonic		Operation(s) Tested
NRMLZ3		Does NORM NOT "stop shift" with AC0 = AC1, AC1 = 0, or AC0 = 0, AC1 = 0 or until SC = 77
NRMLZ4		Will NORMS normalize the alternate pattern of 1 and 0 bits for each bit position of the AC and MQ.
NRMLZ5		Will complement bit patterns normalize
INTEST	(1)	Will the teleprinter flag cause an interrupt after an EAE NOP
	(2)	Will the teleprinter flag cause an interrupt after an LLS 43_8 places
	(3)	Does the interrupt not occur until the LLS is complete

7. METHODS (Not Applicable)

8. FORMAT (Not Applicable)

9. EXECUTION TIME (Not Applicable)

10. PROGRAM

10.1 Core Map (None)

10.2 Dimension List (None)

10.3 Macro, Parameter, and Variable Lists (None)