PDP-15 58
EAE

IDENTIFICATION

PRODUCT CODE:	MAINDEC-15-DØGB-D (D)
PRODUCT NAME:	PDP-15 [redacted]
DATE REVIEWED:	AUGUST 6, 1970
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	KEITH NELSON/J. KLAPKIW

(1)

2. ABSTRACT

Part 1 of the PDP-15 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 left1 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 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

(ASC11 tape 2A of this test)

3.3 Equipment

Minimum configuration PDP-15 with EAE option installed.

4. USAGE

4.1 Loading

a. Set Bank Mode SW on 1.
b. Set address SW to 17700.
c. Press reset, press READ IN.

4.2 Calling Sequence

Part 1 Section 1 must run in its entirety before running Part 1 Section 2.

Part 1 Section 2 must run in its entirety 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	The processor halts after each sequence of

SW#	Operation	Description
	Processor halts at address 0066 (AC)=S.A. of last sequence	testing an EAE operation ; i.e., after testing that the MQ will complement all patterns, the processor halts.
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 = 0200

Press I/O Reset

Press START

Processor halts at 0201 with MQ = 777777

Set ADDRESS = 0202

Press I/O Reset

Press START

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

4.4.2 Start Up, SC and Basic Shift Test

Set AC switches = 000000

Set ADDRESS = 2200

Press I/O Reset

Press START

4.4.3 Start Up Random Data and Normalize Test

Set AC switches = 000000

Set ADDRESS = 5000

Press I/O Reset

Press START

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:

4.5.1 (Continued)

(1) An address that may be used to determine which test 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 is 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
```

4.5.1 (Continued)

	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.
ACOTOL	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 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 MG
	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
ACOTOL				Operation is AC bit 0 to link
	L	C(AC)	C(MQ)	Header

Example			Explanation
START	1	400000	Initial conditions MQ not pertinent
ACOTOL	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	000000	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

Example				Explanation
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

4.5.3 (Continued)

Example		Explanation
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
LSR	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

Example				Explanation
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

4.5.3 (Continued)

Example	Explanation
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 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

4.6.2.1 (Continued)

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-15 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.

6.1.1 (Continued)

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

6.1.2.1 (Continued)

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
NOPLKI	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
SCTSTI	(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

6.1.2.2 (Continued)

Test Mnemonic	Operation(s) Tested
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?)
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 AC 17 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)

6.1.2.2 (Continued)

Test Mnemonic	Operation(s) Tested
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 (ACO) right 1 to 44_8 places
LRSTS8	Will AC/MQ shift a 0 bit (ACO) right 1 to 44_8 places
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

6.1.2.3 (Continued)

Test Mnemonic	Operation(s) Tested
	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 RANSEQ0 to RANSEQ8. After each sequence the AC and MQ should equal their starting patterns.
RANSQ0	Bit 0 to 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
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 or 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

6.1.2.3 (Continued)

Test Mnemonic	Operation(s) Tested
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
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
	(4) Does the interrupt not occur until 2 instructions after a normalize.

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)

```
                                .TITLE EAE-P1
                                .ABS
                        /EAE  SET UP DIAGNOSTIC
                        /
                        /START AT 200
                        /PROCESSOR HALTS AT 201 WITH MQ=1'S
                        /SET ADDRESS SWITCHES TO 202, THEN DO RESET AND START.
                        /
                        /SW0 - DELETE ERROR TYPEOUTS
                        /SW1 - HALT AFTER EACH EAE OPERATION
                        /SW2 - REPEAT LAST EAE OPERATION
                        /SW3 - HALT AFTER EACH EAE SEQUENCE
                        /SW4 - REPEAT EACH EAE SEQUENCE
                        /SW5 - 0-REPEAT SET UP TEST OR SCA AND SHIFT TESTS
                        /SW5 - 1-CYCLE SET UP AND SC AND SHIFT TEST
                        /
00020                           .LOC 20
                        /
                        /CAL SUBROUTINE
00020   000020                  20                      /20 IN CASE CAL*
00021   200020                  LAC 20                  /GET ADDRESS
00022   040000                  DAC 0                   /SAVE
00023   200027                  LAC .+4                 /RESTORE 20
00024   040020                  DAC 20
00025   200000                  LAC 0
00026   740040                  HLT                     /HLT DISPLAY
00027   000020                  20                      /WILL CAL IF CONTINUE
                        /
                        /
                        /AC, MQ, LINK AND SC FOR TYPEOUTS
00030                           .LOC 30
00030   000000          ACSTRT  0
00031   000000          MQSTRT  0
00032   000000          LKSTRT  0
00033   000000          SCSTRT  0
00034   000000          ACEND   0
00035   000000          MQEND   0
00036   000000          LKEND   0
00037   000000          SCEND   0
                        /
                                .EJECT
```

```
                        /ROUTINES THAT TEST REPEAT AND STOP
                        /STOP AFTER MINOR LOOP (SW1) AND REPEAT MINOR LOOP (SW2)
                        /
00040   600040  SWITCH  JMP .
00041   750004          LAS
00042   501322          AND BIT1
00043   741200          SNA                     /MINOR LOOP HALT?
00044   600047          JMP .+3                 /NO
00045   220040          LAC* SWITCH
00046   740040          HLT
00047   220040          LAC* SWITCH
00050   040057          DAC .+7
00051   440040          ISZ SWITCH
00052   750004          LAS
00053   501323          AND BIT2
00054   740200          SZA                     /REPEAT LOOP?
00055   620057          JMP* .+2  /YES
00056   620040          JMP* SWITCH             /CONTINUE IN SEQUENCE
00057   000000          0
                        /
                        /STOP AFTER MAJOR LOOP (SW3) AND REPEAT MAJOR LOOP (SW4)
00060   600060  SWTCHS  JMP .
00061   750004          LAS
00062   501324          AND BIT3
00063   741200          SNA                     /MAJOR LOOP HALT?
00064   600067          JMP .+3                 /NO
00065   220060          LAC* SWTCHS
00066   740040          HLT
00067   220060          LAC* SWTCHS
00070   040057          DAC SWTCHS-1
00071   440060          ISZ SWTCHS
00072   750004          LAS
00073   501325          AND BIT4
00074   741200          SNA                     /REPEAT MAJOR LOOP?
00075   620060          JMP* SWTCHS             /CONTINUE
00076   620057          JMP* SWTCHS-1           /REPEAT LOOP
                        /
                        .EJECT
```

```
                        /DOES EAE - OR THE MQ TO AC READ 0'S
                        /MQ SHOULD BE ZERO FROM RESET KEY
                        /
00200                           .LOC 200
00200   640004          SETUP   CMQ
00201   740040                  HLT
                        /
00202   754000          EAERMQ  CLA*4000  /CLEAR LINK
00203   040031                  DAC MQSTRT
00204   040030                  DAC ACSTRT
00205   640002                  EAE*2                   /OR MQ 1'S TO AC
00206   040034                  DAC ACEND
00207   741200                  SNA
00210   600221                  JMP .+11
00211   101134                  JMS ERROR
00212   001533                      TYRMQ
00213   001375                      HDR2
00214   600030                  ACSTRT+600000
00215   600031                  MQSTRT+600000
00216   001533                      TYRMQ
00217   600034                  ACEND+600000
00220   000000                  0
00221   100040                  JMS SWITCH
00222   000202                      EAERMQ
00223   201363                  LAC NBIT16
00224   041261                  DAC CHARK               /SET END TEST K
                        /
                        /DOES EAE NOP CLEAR THE AC?
                        /
00225   754001          NOPAC   CLC*4000  /CLEAR LINK
00226   040030                  DAC ACSTRT              /AC AT START
00227   501365                  AND KALL7 /MAKE MB=1#S BEFORE
00230   640000                  EAE
00231   040034                  DAC ACEND /AC AT END
00232   740001                  CMA
00233   741200                  SNA                     /AC ALTERED
00234   600244                  JMP .+10                /NO
00235   101134                  JMS ERROR
00236   001514                      TYNOP
00237   001366                      HDR1
00240   600030                  ACSTRT+600000           /TYPE CONTENTS OF
                        /
00241   001514                      TYNOP               /TYPE TEXT
00242   600034                  ACEND+600000            /TYPE CONTENTS OF
00243   000000                  0
00244   100040                  JMS SWITCH              /REPEAT SET
00245   000225                      NOPAC               /LOOP TO HERE
                        /
                                .EJECT
```

```
                        /DOES EAE AND CLR AC BIT CLR THE AC?
                        /
00246   754001  EAECAC  CLC*4000              /CLEAR LINK
00247   641000          EAE*1000              /SHOULD CLEAR AC
00250   040034          DAC ACEND
00251   741200          SNA
00252   600262          JMP .+10
00253   101134          JMS ERROR
00254   001517              TYCLA
00255   001366              HDR1
00256   600030          ACSTRT+600000
00257   001517              TYCLA
00260   600034          ACEND+600000
00261   000000          0
00262   100040          JMS SWITCH
00263   000246              EAECAC
                        /
                        /
                        /
                        /DOES CLQ CLEAR THE MQ
                        /
00264   754001  EAECLQ  CLC*4000
00265   040031          DAC MQSTRT
00266   640004          EAE*4                 /SET MQ TO 1'S
00267   750000          CLA
00270   040030          DAC ACSTRT
00271   650000          CLQ                   /CLEAR THE MQ
00272   040034          DAC ACEND
00273   750000          CLA
00274   640002          EAE*2                 /OR MQ 1'S TO AC
00275   040035          DAC MQEND
00276   741200          SNA                   /READ 0'S BACK?
00277   600311          JMP .+12
00300   101134          JMS ERROR
00301   001523              TYCLQ
00302   001375              HDR2
00303   600030          ACSTRT+600000
00304   600031          MQSTRT+600000
00305   001523              TYCLQ
00306   600034          ACEND+600000
00307   600035          MQEND+600000
00310   000000          0
00311   100040          JMS SWITCH            /REPEAT SET
00312   000264              EAECLQ                    /START OVER
                        /
                        .EJECT
```

```
                        /DOES MQ COMPLIMENT FROM 0'S TO 1'S
                        /AND MQ 1'S TO AC
                        /
00313   754000  MQ1TAC  CLA+4000
00314   040030          DAC ACSTRT
00315   040031          DAC MQSTRT
00316   650004          CLQ+4                   /CLEAR THE MQ AND COMPLIMENT
00317   040034          DAC ACEND
00320   750000          CLA
00321   640002          EAE+2                   /OR THE MQ TO AC
00322   040035          DAC MQEND
00323   740001          CMA
00324   741200          SNA
00325   600337          JMP .+12
00326   101134          JMS ERROR
00327   001527              TYCMQ
00330   001375              HDR2
00331   600030          ACSTRT+600000
00332   600031          MQSTRT+600000
00333   001527              TYCMQ
00334   600034          ACEND+600000
00335   600035          MQEND+600000
00336   000000          0
00337   100040          JMS SWITCH
00340   000313              MQ1TAC
                        /
                        /
                        /DOES EAE-NOP WITH MQ=1'S ALTER THE AC
                        /
00341   754000  NOPAC1  CLA+4000
00342   040030          DAC ACSTRT
00343   750001          CLC
00344   040031          DAC MQSTRT
00345   650004          CLQ+4                   /SET MQ TO ONES
00346   501365          AND KALL7               /MAKE MB TO 1#S
00347   640000          EAE                     /NOP
00350   040034          DAC ACEND
00351   740001          CMA
00352   741200          SNA                     /ONES FROM MQ TO AC?
00353   600364          JMP .+11
00354   101134          JMS ERROR
00355   001514              TYNOP
00356   001375              HDR2
00357   600030          ACSTRT+600000
00360   600031          MQSTRT+600000
00361   001514              TYNOP
00362   600034          ACEND+600000
00363   000000          0
00364   100040          JMS SWITCH
00365   000341              NOPAC1
                        /
                        .EJECT
```

```
                        /DOES EAE NOP WITH MQ=1'S ALTER THE MQ
                        /
00366   754000  NOPMQ   CLA+4000
00367   650004          CLQ 4                   /SET MQ TO 1'S
00370   501365          AND KALL7               /MAKE MB TO 1#S BEFORE
00371   640000          EAE                     /NOP
00372   040034          DAC ACEND
00373   750000          CLA
00374   640002          EAE+2
00375   040035          DAC MQEND
00376   740001          CMA
00377   741200          SNA                     /MQ STILL 1'S?
00400   600412          JMP .+12
00401   101134          JMS ERROR
00402   001514              TYNOP
00403   001375              HDR2
00404   600030          ACSTRT+600000
00405   600031          MQSTRT+600000
00406   001514              TYNOP
00407   600034          ACEND+600000
00410   600035          MQEND+600000
00411   000000          0
00412   100040          JMS SWITCH
00413   000366              NOPMQ
                        /
                        /
                        /DOES NOP WITH AC=1'S ALTER MQ
                        /
00414   754000  NOPMQ1  CLA+4000
00415   040031          DAC MQSTRT
00416   650000          CLQ
00417   750001          CLC
00420   040030          DAC ACSTRT
                        /
00421   501365          AND KALL7               /MAKE MB TO 1S BEFORE
00422   640000          EAE                     /NOP
00423   040034          DAC ACEND
00424   641002          LACQ                    /GET MQ TO AC
00425   040035          DAC MQEND
00426   741200          SNA                     /ANY 1'S IN MQ
00427   600441          JMP .+12
00430   101134          JMS ERROR
00431   001514              TYNOP
00432   001375              HDR2
00433   600030          ACSTRT+600000
00434   600031          MQSTRT+600000
00435   001514              TYNOP
00436   600034          ACEND+600000
00437   600035          MQEND+600000
00440   000000          0
00441   100040          JMS SWITCH
00442   000414              NOPMQ1
                        /
                        .EJE
```

```
                        /DOES NOP ALTER THE LINK
                        /AC 0'S MQ 0'S, AC 1'S MQ 1'S
                        /
00443   650000  NOPLNK  CLQ
00444   140030          DZM ACSTRT
00445   140031          DZM MQSTRT
00446   140032          DZM LKSTRT
00447   200032          LAC LKSTRT
00450   740020          RAR
00451   200030          LAC ACSTRT          /SET LINK FOR TEST
00452   501365          AND KALL7           /MAKE MB TO ONES BEFORE
00453   640000          EAE                 /NOP
00454   750010          GLK
00455   040036          DAC LKEND
00456   540032          SAD LKSTRT          /LINK ALTERED?
00457   600471          JMP .+12
00460   101134          JMS ERROR
00461   001514              TYNOP
00462   001411              HDR3
00463   700032          LKSTRT+700000       /ZERO SUPPRESS CONTENTS
00464   600030          ACSTRT+600000
00465   600031          MQSTRT+600000
00466   001514              TYNOP
00467   700036          LKEND+700000
00470   000000          0
00471   100040          JMS SWITCH
00472   000447          NOPLNK+4
00473   200032          LAC LKSTRT
00474   440032          ISZ LKSTRT
00475   741200          SNA                 /CHECKED L=0 AND L=1?
00476   600447          JMP NOPLNK+4
00477   200030          LAC ACSTRT
00500   740200          SZA                 /CHECKED FOR AC=1'S
00501   600533          JMP EAESIK          /YES
00502   650004          CLQ+4               /SET MQ TO 1'S
00503   750001          CLC
00504   040030          DAC ACSTRT          /AC START =1'S
00505   040031          DAC MQSTRT
00506   140032          DZM LKSTRT          /LINK START=0
00507   600447          JMP NOPLNK+4
                        /
                        .EJECT
```

```
                        /DOES MQ TO AC ALL 1'S WITH AC=1'S
00510   750001  QONEAC  CLC
00511   040030          DAC ACSTRT
00512   040031          DAC MQSTRT                      /SET MQ TO 1'S
00513   650004          CLQ*4                   /MQ1'S TO AC1'A
00514   640002          OMQ
00515   040034          DAC ACEND
00516   740001          CMA
00517   741200          SNA                     /AC STAY 1'S
00520   600531          JMP .+11
00521   101134          JMS ERROR
00522   001533              TYRMQ
00523   001375              HDR2
00524   600030              ACSTRT+600000
00525   600031              MQSTRT+600000
00526   001533              TYRMQ
00527   600034              ACEND+600000
00530   000000          0
00531   100040          JMS SWITCH
00532   000510              QONEAC
                        /
                        .EJECT
```

```
                        /LINK SET TO 1 AND TO ZERO?
                        /
00533   140032  EAESLK  DZM LKSTRT          /START LINK 0 TO 1
00534   140031          DZM MQSTRT          /MQ 0'S
00535   650000          CLQ
00536   201321          LAC BIT0            /400000
00537   040030          DAC ACSTRT
00540   200032          LAC LKSTRT          /SET LINK INITIAL
00541   740020          RAR
00542   200030          LAC ACSTRT
00543   660000          EAE+20000           /AC BIT 0 TO LINK
00544   040034          DAC ACEND
00545   750010          GLK
00546   040036          DAC LKEND
00547   742020          RTR
00550   540030          SAD ACSTRT                      /LINK SAME AS START?
00551   741000          SKP
00552   600556          JMP .+4             /ERROR
00553   200034          LAC ACEND
00554   540030          SAD ACSTRT          /AC ALTERED?
00555   600570          JMP .+13
00556   101134          JMS ERROR
00557   001537              TYSLK
00560   001411              HDR3
00561   700032          LKSTRT+700000
00562   600030          ACSTRT+600000
00563   000031              MQSTRT
00564   001537              TYSLK
00565   700036          LKEND+700000
00566   600034          ACEND+600000
00567   000000          0
00570   100040          JMS SWITCH          /LOOP SET?
00571   000540          EAESLK+5
00572   440032          ISZ LKSTRT          /NEXT PASS LINK 1 TO ZERO
00573   200030          LAC ACSTRT
00574   140030          DZM ACSTRT
00575   740200          SZA
00576   600540          JMP EAESLK+5
                        /
                        .EJECT
```

```
                        /DOES NOP ALTER MQ=0'S WITH L=1
                        /
00577   140030  NOPLK1  DZM ACSTRT          /START AC 0'S
00600   140031          DZM MQSTRT          /MQ 0'S
00601   650000          CLQ
00602   201342          LAC BIT17           /1=LINK
00603   040032          DAC LKSTRT
00604   744020          RAR+4000            /CLR LINK, SET LINK
00605   501365          AND KALL7           /MAKE MB TO ONES BEFORE
00606   640000          EAE                 /NOP
00607   240034          DAC ACEND
00610   750010          GLK
00611   040036          DAC LKEND
00612   641002          LACQ
00613   040035          DAC MQEND
00614   741200          SNA                 /MQ STILL ZERO'S
00615   200034          LAC ACEND
00616   751200          SNA!CLA             /AC STILL ZERO'S
00617   200036          LAC LKEND
00620   740200          SZA                 /LINK STILL 1
00621   600635          JMP .+14
00622   101134          JMS ERROR
00623   001514              TYNOP
00624   001411              HDR3
00625   700032          LKSTRT+700000
00626   600030          ACSTRT+600000
00627   600031          MQSTRT+600000
00630   001514              TYNOP
00631   700036          LKEND+700000
00632   600034          ACEND+600000
00633   600035          MQEND+600000
00634   000000          0
00635   100040          JMS SWITCH          /CHECK MINOR LOOP SW
00636   000577              NOPLK1
00637   100060          JMS SWTCHS          /MAJOR LOOP SET?
00640   000225              NOPAC           /START NOP THE AC
                        /
                        .EJECT
```

```
                        /WILL AC TO MQ TO AC ALL PATTERNS
                        /WITH MQ INITIALLY = 0  AND LINK = 0
                        /
00641   140037  ACORMQ  DZM ACSTRT          /START AC = 0'S
00642   140031          DZM MQSTRT          /MQ ALWAYS 0'S
00643   754000          CLL!CLA
00644   650000          CLQ
00645   200030          LAC ACSTRT          /GET NEXT SET
00646   643000          EAE*3000            /AC TO MQ
00647   040034          DAC ACEND
00650   641002          LACQ                /MQ TO AC
00651   040035          DAC MQEND
00652   540030          SAD ACSTRT          /MQ TO AC SAME AS START?
00653   741000          SKP
00654   600660          JMP .+4
00655   200034          LAC ACEND           /YES, TRY AC
00656   741200          SNA                 /AC SHOULD BE 0
00657   600676          JMP .+17
00660   101134          JMS ERROR
00661   001543              TYSMQ
00662   001375              HDR2
00663   600030          ACSTRT+600000
00664   600031          MQSTRT+600000
00665   001543              TYSMQ
00666   600031          MQSTRT+600000
00667   600030          ACSTRT+600000
00670   001461              TYCOR
00671   001547              TYLACQ
00672   600034          ACEND+600000
00673   600035          MQEND+600000
00674   001463              TYINCO
00675   000000          0
00676   100040          JMS SWITCH          /CHECK FOR REPEAT LOOP
00677   000643          ACORMQ+2
00700   440030          ISZ ACSTRT          /TO 777777?
00701   600643          JMP ACORMQ+2
00702   100060          JMS SWTCHS
00703   000641              ACORMQ
                        /
                        .EJECT
```

```
                        /WILL AC TO MQ TO AC ALL PATTERNS
                        /WITH MQ = LAST PATTERN AND LINK = 1
                        /
00704   140030  ACLMQ   DZM ACSTRT          /START AC 0'S
00705   140031          DZM MQSTRT          /MQ 0'S
00706   650000          CLQ
00707   201342          LAC BIT17           /LINK 1
00710   040032          DAC LKSTRT
00711   744002          STL                 /SET LINK
00712   200030          LAC ACSTRT          /GET NEXT CONSTANT
00713   652000          LMQ                 /MQ TO 0'S, AC 1'S TO MQ
00714   040034          DAC ACEND           /SAVE AC RESULT
00715   750010          GLK
00716   040036          DAC LKEND           /SAVE LINK RESULT
00717   641002          LACQ                /GET MQ
00720   040035          DAC MQEND
00721   540030          SAD ACSTRT          /MQ = AC AT START?
00722   741000          SKP
00723   600732          JMP ACLMQE          /MQ ERROR
00724   200036          LAC LKEND
00725   741200          SNA                 /LINK=1 AT END?
00726   600732          JMP ACLMQE          /LINK ERROR
00727   200034          LAC ACEND
00730   540030          SAD ACSTRT          /AC END = AC START?
00731   600753          JMP .+22
00732   101134  ACLMQE  JMS ERROR
00733   001622              TYLMQ
00734   001411              HDR3
00735   700032          LKSTRT+700000
00736   600030          ACSTRT+600000
00737   600031          MQSTRT+600000
00740   001622              TYLMQ
00741   700032          LKSTRT+700000
00742   600030          ACSTRT+600000
00743   600030          ACSTRT+600000
00744   001461              TYCOR
00745   001547              TYLACQ
00746   700036          LKEND+700000
00747   600034          ACEND+600000
00750   600035          MQEND+600000
00751   001463              TYINCO
00752   000000          0
00753   200035          LAC MQEND
00754   040031          DAC MQSTRT          /NEW MQ START
00755   100040          JMS SWITCH          /REPEAT SET?
                        /
00756   000711          ACLMQ+5
                        /
00757   440030          ISZ ACSTRT          /TO 777777?
00760   600711          JMP ACLMQ+5
00761   100060          JMS SWTCHS
00762   000704              ACLMQ
                        /
                        .EJECT
```

```
                        /TESTS THE MQ COMPLIMENT ALL PATTERNS
                        /
00763   140763  COMPMQ  DZM ACSTRT
00764   200030          LAC ACSTRT          /GET NEXT PATTERN
00765   040031          DAC MQSTRT
00766   672000          LMQ*20000           /AC TO MQ, AC0 TO L
00767   640004          CMQ                 /-MQ
00770   040034          DAC ACEND           /SAVE AC RESULT
00771   641002          LACQ                /GET MQ
00772   040035          DAC MQEND
00773   740001          CMA                 //MQ
00774   540030          SAD ACSTRT          /-MQ = AC START?
00775   200034          LAC ACEND
00776   540030          SAD ACSTRT          /ACEND = AC START?
00777   601011          JMP .+12
01000   101134          JMS ERROR
01001   001527              TYCMQ
01002   001375              HDR2
01003   600030          ACSTRT+600000
01004   600031          MQSTRT+600000
01005   001527              TYCMQ
01006   600034          ACEND+600000
01007   600035          MQEND+600000
01010   000000          0
01011   100040          JMS SWITCH
01012   000764          COMPMQ+1
01013   440030          ISZ ACSTRT
01014   600764          JMP COMPMQ+1
01015   100060          JMS SWTCHS
01016   000763              COMPMQ
                        /
                        .EJECT
```

```
                        /DOES AC TO MQ ALL 1'S WITH MQ=1'S
01017   750001  ACONEQ  CLC
01020   040031          DAC MQSTRT
01021   040030          DAC ACSTRT
01022   650004          CLQ*4                   /SET MQ=1'S
01023   642000          EAE*2000  /AC 1'S TO MQ1'S
01024   040034          DAC ACEND
01025   641002          LACQ
01026   040035          DAC MQEND
01027   740001          CMA
01030   741200          SNA       /MQ STAY 1'S
01031   601047          JMP .+16
01032   101134          JMS ERROR
01033   001543              TYSMQ
01034   001375              HDR2
01035   600030              ACSTRT+600000
01036   600031              MQSTRT+600000
01037   001543              TYSMQ
01040   600034              ACEND+600000
01041   600035              MQEND+600000
01042   000000          0
01043   100040          JMS SWITCH
01044   001017              ACONEQ
                        /
                        .EJECT
```

```
                        .DOES ABS GET ABSOLUTE AC
                        /AND NOT DISTURB LINK=1 OR 0
01045   140030  EAEABS  DZM ACSTRT          /START AC 0'S
01046   201342          LAC BIT17           /LINK 1
01047   040032          DAC LKSTRT
01050   200032          LAC LKSTRT
01051   740020          RAR                 /SET LINK
01052   200030          LAC ACSTRT          /GET AC START
01053   644000          ABS                 /ABSOLUTE AC
01054   040034          DAC ACEND           /SAVE RESULT
01055   750010          GLK
01056   040036          DAC LKEND
01057   540032          SAD LKSTRT          /LINK SAME?
01060   741000          SKP                 /YES
01061   601067          JMP .+6             /ERROR, LINK CHANGED
01062   200030          LAC ACSTRT
01063   741100          SPA                 /AC POSITIVE AT START?
01064   740001          CMA                 /NO, SHOULD BE POS. ABS
01065   540034          SAD ACEND           /RESULT AC OK?
01066   601100          JMP .+12            /YES
01067   101134          JMS ERROR           /ABS ERROR LINK OR AC
01070   001626              TYABS
01071   001411              HDR3
01072   700032          LKSTRT+700000
01073   600030          ACSTRT+600000
01074   001626              TYABS
01075   700036          LKEND+700000
01076   600034          ACEND+600000
01077   000000          0
01100   100040          JMS SWITCH
01101   001050          EAEABS+3
01102   440030          ISZ ACSTRT
01103   741000          SKP
01104   601112          JMP NDSETU
01105   200032          LAC LKSTRT
01106   740001          CMA
01107   501342          AND BIT17
01110   040032          DAC LKSTRT
01111   601050          JMP EAEABS+3
                        /
                        .EJECT
```

```
01112  100060  NDSETU  JMS SWTCHS          /TEST REPEAT MAJOR
01113  001045          EAEABS
01114  750004          LAS
01115  501327          AND BIT6
01116  741200          SNA
01117  601125          JMP .+6
01120  760057          LAW 57
01121  101716          TYI
01122  441261          ISZ CHARK
01123  601127          JMP .+4
01124  101240          JMS CRLF
01125  201363          LAC NBIT16
01126  041261          DAC CHARK
01127  750004          LAS
01130  501326          AND BIT5
01131  741200          SNA                  /REPEAT ALL SET?
01132  600225          JMP NOPAC            /CYCLE SET UP TEST
01133  602246          JMP SCTO76                   /CYCLE SET UP AND SHIFT
                       /
                       /EAE ERROR TYPEOUT ROUTINE
                       /GENERAL PURPOSE
                       /LINKS TYPTEX AND ALL TYPE CONTENTS
                       /
                       /AC=0     IS END OF TYPEOUT
                       /AC NOT = 0 AND POSITIVE IS TYPETEXT
                       /AC - AND BIT 1=0 IS CR, LF TYPE CONTENTS
                       /AC - AND BIT 1=1 IS TYPE CONTENTS
                       /AC - AND BIT 2=0 IS NO ZERO UPPRESS
                       /AC - AND BIT 2=1 IS ZERO SUPPRESS
                       /AC - AND BIT 3=0 IS ZERO SUPPRESS8
                       /AC - AND BIT 3=1 IS ZERO SUPPRESS4
                       /
                       /
01134  601134  ERROR    JMP .
01135  750004          LAS
01136  741100          SPA
01137  601244          JMP TYDELE
01140  101240          JMS CRLF
01141  201144          LAC .+3
01142  041276          DAC SAVERR
01143  601167          JMP TYPECN                   /CR LF TYPE CONTENTS ERROR
01144  001134                  ERROR
01145  221134  ERLOOP   LAC* ERROR          /GET NEXT TYPE CONSTANT
01146  041276          DAC SAVERR           /FOR INDIRECTS
01147  506471          AND (7777
01150  041277          DAC SVER
01151  441134          ISZ ERROR
01152  740200          SZA                  /END OF MESSAGE?
01153  601163          JMP ERCONT           /NO
01154  750004          LAS                  /GET SWITCHES
01155  501330          AND BIT7
01156  741200          SNA                  /DELETE HALT?
01157  740040          HLT                  /ERROR HALT
01160  700401          TSF
01161  601160          JMP .-1              /WAIT FLAG
```

```
01162   621134                  JMP* ERROR          /EXIT ERROR ROUT.
01163   741100      ERCONT      SPA                 /TYPE TEXT INDICATED?
01164   601167                  JMP TYPECN          /NO, TYE CONTENTS
01165   101673                  TSR
01166   601145                  JMP ERLOOP
                    /
                                .EJECT
```

```
                        /TYPE CONTENTS ROUTINES
                        /
01167   501322  TYPECN  AND BIT1
01170   741200          SNA                     /CARRIAGE RETURN INDICATED
01171   101240          JMS CRLF                /YES
01172   201276          LAC SAVERR
01173   501323          AND BIT2
01174   741200          SNA                     /SUPPRESS ZERO SET?
01175   601215          JMP TCALL               /NO, TYPE ALL
01176   201276          LAC SAVERR
01177   501324          AND BIT3
01200   740200          SZA                     /SUPPRESS 4 0'S SET?
01201   601224          JMP TCTWO               /YES
01202   221277          LAC* SVER
01203   501214          AND .+11
01204   740200          SZA                     /UPPER 5 CHAR = 0
01205   601215          JMP TCALL               /NO, TYPE ALL
01206   221277          LAC* SVER
01207   746020          CLL!RTR
01210   742020          RTR
01211   102026              TWORD
01212   000001          1
01213   601220          JMP TCALL+3                     /SPACE 3
01214   777770          777770
                        /
                        .EJECT
```

```
01215   221277      TCALL   LAC* SVER
01216   102026              TWORD                   /TYPE 6 OCTAL
01217   000006                6
01220   761442              LAW SPACE3
01221   101673              TSR                     /OUTPUT 3 SPACES
01222   601145              JMP ERLOOP
01223   777700              777700
01224   221277      TCTWO   LAC* SVER
01225   501223              AND .-2
01226   740200              SZA                     /FIRST 4 CHARACTERS Ø
01227   601215              JMP TCALL               /NO, TYPE WHOLE WORD
01230   221277              LAC* SVER
01231   746020              CLL!RTR                 /POSITION LS 2
01232   742020              RTR                     /TO UPPER 2
01233   742020              RTR                     /FOR TYPEOUT ROUT
01234   740020              RAR
01235   102026                  TWORD               /TYPE UPPER 2 CHAR
01236   000002              2
01237   601220              JMP TCALL+3                     /SPACE 3
01240   601240      CRLF    JMP .
01241   761465              LAW CRCODE
01242   101673              TSR
01243   621240              JMP* CRLF
                    /
                            .EJECT
```

```
01244  221134  TYDELE  LAC* ERROR
01245  441134          ISZ ERROR
01246  740200          SZA                     /REACHED END OF MESS.
                /
01247  601244          JMP TYDELE                      /NO
01250  750004          LAS
01251  501330          AND BIT7
01252  741200          SNA                     /RING BELL SET?
01253  621134          JMP* ERROR              /NO, EXIT
01254  206472          LAC (207207
01255  102107          JMS OTY
01256  621134          JMP* ERROR
                /
01257  777773  MIN5    777773
01260  777772  MIN6    777772
01261  000000  CHARK   0
01262  000000          0
01263  000000  SVCHAR  0
01264  000000          0
01265  000000          0
01266  000000          0
01267  000000          0
01270  000000          0
01271  000007  SEVEN   7
01272  000240  TWO40   240
01273  000260  TWO60   260
01274  000077  SEVSEV  77
01275  000076  SEVSIX  76
01276  000000  SAVERR  0
01277  000000  SVER    0
01300  777756  K18     777756
01301  000060  SIXTY   60
01302  000070  SEVNTY  70
01303  000074  SEVN4   74
01304  000041  FOUR1   41
01305  000037  THREE7  37
01306  000061  SIXONE  61
01307  000017  ONESEV  17
01310  000071  SEVONE  71
01311  000075  SEVFIV  75
01312  000003  THREE   3
01313  000045  FOUR5   45
01314  000044  FOUR4   44
01315  000043  FOUR3   43
01316  000034  THREE4  34
01317  000056  FIVE6   56
01320  252525  COMBIT  252525
                /
                .EJECT
```

```
                        .BIT AND NO BIT CONSTANTS
                        /
01321   400000  BIT0    400000
01322   200000  BIT1    200000
01323   100000  BIT2    100000
01324   040000  BIT3    40000
01325   020000  BIT4    20000
01326   010000  BIT5    10000
01327   004000  BIT6    4000
01330   002000  BIT7    2000
01331   001000  BIT8    1000
01332   000400  BIT9    400
01333   000200  BIT10   200
01334   000100  BIT11   100
01335   000040  BIT12   40
01336   000020  BIT13   20
                /
01337   000010  BIT14   10
                /
01340   000004  BIT15   4
01341   000002  BIT16   2
01342   000001  BIT17   1
01343   377777  NBIT0   377777
01344   577777  NBIT1   577777
01345   677777  NBIT2   677777
01346   737777  NBIT3   737777
01347   757777  NBIT4   757777
01350   767777  NBIT5   767777
01351   773777  NBIT6   773777
01352   775777  NBIT7   775777
01353   776777  NBIT8   776777
01354   777377  NBIT9   777377
01355   777577  NBIT10  777577
01356   777677  NBIT11  777677
01357   777737  NBIT12  777737
01360   777757  NBIT13  777757
01361   777767  NBIT14  777767
01362   777773  NBIT15  777773
01363   777775  NBIT16  777775
01364   777776  NBIT17  777776
01365   777777  KALL7   777777
                /
                        .EJECT
```

```
                        /MESSAGE CONSTANTS
                        /ERROR TYPEOUT HEADERS
                        /AC CONTENTS
                        /
01366   151203  HDR1    .SIXBT <15><12>'C(AC)'
01367   500103
01370   510000
01371   151223          .SIXBT <15><12>'START  '<77>
01372   240122
01373   244040
01374   770000
                        /AC AND MQ
                        /
01375   151240  HDR2    .SIXBT <15><12>'       C(AC)    C(MQ)'
01376   404040
01377   404040
01400   400350
01401   010351
01402   404040
01403   400350
01404   152151
01405   151223          .SIXBT <15><12>'START  '<77>
01406   240122
01407   244040
01410   770000
                        /LINK AC AND MQ
                        /
01411   151240  HDR3    .SIXBT <15><12>'       L    C(AC)    C(MQ)'
01412   404040
01413   404040
01414   144040
01415   404003
01416   500103
01417   514040
01420   404003
01421   501521
01422   510000
01423   151223          .SIXBT <15><12>'START  '<77>
01424   240122
01425   244040
01426   770000
                        /SC AC
                        /
01427   151240  HDR4    .SIXBT <15><12>'       SC    C(AC)'
01430   404040
01431   404040
01432   230340
01433   404040
01434   035001
01435   035100
01436   151223          .SIXBT <15><12>'START  '<77>
01437   240122
01440   244040
01441   770000
                        /3 SPACES
```

```
                        /
01442   404040          SPACE3  .SIXBT '    '<77>
01443   770000
                        /4 SPACES
                        /
01444   151240          SPACE4  .SIXBT <15><12>'    '<77>
01445   404040
01446   770000
                        /
                                .EJECT
```

```
01447   151214      HDR5    .SIXBT <15><12>'L    C(AC)    C(MQ)'<77>
01450   404040
01451   400350
01452   010351
01453   404040
01454   400350
01455   152151
01456   770000

                    /
01457   200124      TYPATR  .SIXBT 'PAT'<77>
01460   770000
                    /
01461   031722      TYCOR   .SIXBT 'COR'<77>
01462   770000
                    /
01463   111603      TYINCO  .SIXBT 'INCO'<77>
01464   177700
                    /
01465   151277      CRCODE  .SIXBT <15><12><77>
                    /
01466   151216      TYNRMS  .SIXBT <15><12>'NORMS  '<77>
01467   172215
01470   234040
01471   770000
                    /
01472   151216      TYINTE  .SIXBT <15><12>'NO PROGRAM INTERRUPT'<77>
01473   174020
01474   221707
01475   220115
01476   401116
01477   240522
01500   222520
01501   247700
                    /
01502   151211      INDAT   .SIXBT <15><12>'INTERRUPT DATA ERROR'<77>
01503   162405
01504   222225
01505   202440
01506   040124
01507   014005
01510   222217
01511   227700
                    /
01512   232401      TYSTRT  .SIXBT 'START'<77>
01513   222477
                    /
                            .EJECT
```

```
                        /OPERATION TYPEOUTS
                        /EAE NO OPERATION
                        /
01514   151205          TYNOP       .SIXBT <15><12>'EANOP '<77>
01515   211617
01516   204077
                        /
                        /
                        /EAE CLA
01517   151205          TYCLA       .SIXBT <15><12>'EAECLA '<77>
01520   010503
01521   140140
01522   770000
                        /
                        /
                        /CLEAR MQ
01523   151203          TYCLQ       .SIXBT <15><12>'CLQ    '<77>
01524   142140
01525   404040
01526   770000
                        /
                        /
                        /COMPLIMENT Q
01527   151203          TYCMQ       .SIXBT <15><12>'CMQ    '<77>
01530   152140
01531   404040
01532   770000
                        /
                        /
                        /OR MQ TO AC
01533   151217          TYRMQ       .SIXBT <15><12>'ORMQAC '<77>
01534   221521
01535   010340
01536   770000
                        /
                        /
                        /AC0 TO LINK
01537   151201          TYSLK       .SIXBT <15><12>'AC0TOL '<77>
01540   036024
01541   171440
01542   770000
                        /
                        /
                        /OR AC TO MQ
01543   151217          TYSMQ       .SIXBT <15><12>'ORACMQ '<77>
01544   220103
01545   152140
01546   770000
                        /
                        /
                        /LOAD AC WITH MQ
01547   151214          TYLACQ      .SIXBT <15><12>'LACQ   '<77>
01550   010321
01551   404040
01552   770000
```

```
                        /
                        /
                        /LLS
01553   151214          TYLLS     .SIXBT <15><12>'LLS     '<77>
01554   142340
01555   404040
01556   770000
                        /
                        /
                        /LLSS
01557   151214          TYLLSS    .SIXBT <15><12>'LLSS    '<77>
01560   142323
01561   404040
01562   770000
                        /
                                  .EJECT
```

```
                        /LRS
01563   151214  TYLRS   .SIXBT <15><12>'LRS    '<77>
01564   222340
01565   404040
01566   770000
                        /
                        /
                        /RESULT
01567   151222  TYSIMR  .SIXBT <15><12>'RESULT '<77>
01570   052325
01571   142440
01572   770000
                        /
                        /
                        /TYLRSS
01573   151214  TYLRSS  .SIXBT <15><12>'LRSS   '<77>
01574   222323
01575   404040
01576   770000
                        /
                        /
                        /TYRDSQ
01577   151222  TYRDSQ  .SIXBT <15><12>'RANDOM DATA SEQUENCED'<77>
01600   011604
01601   171540
01602   040124
01603   014023
01604   052125
01605   051603
01606   050477
01607   220523  TYRES   .SIXBT 'RESULT'<77>
01610   251424
01611   770000
01612   151211  TYQINT  .SIXBT <15><12>'INTERUPT NOT DELAYED'<77>
01613   162405
01614   222520
01615   244016
01616   172440
01617   040514
01620   013105
01621   047700
                        /
                        .EJECT
```

```
                        /LOAD MQ WITH AC
01622   151214          TYLMQ     .SIXBT <15><12>'LMQ    '<77>
01623   152140
01624   404040
01625   770000
                        /
                        /
                        /
01626   151201          TYABS     .SIXBT <15><12>'ABS    '<77>
01627   022340
01630   404040
01631   770000
                        /
                        /
                        /CLR SC
01632   151203          TYCSC     .SIXBT <15><12>'CLR SC '<77>
01633   142240
01634   230340
01635   770000
                        /
                        /
                        /LACS
01636   151214          TYLACS    .SIXBT <15><12>'LACS   '<77>
01637   010323
01640   404040
01641   770000
                        /
                        /
                        /SC ERROR
01642   151223          TYSCER    .SIXBT <15><12>'SC ERROR '<77>
01643   034005
01644   222217
01645   224077
                        /
                        /
                        /NORM
01646   151216          TYNORM    .SIXBT <15><12>'NORM   '<77>
01647   172215
01650   404040
01651   770000
                        /
                        /
                        /SET SC
01652   151223          TYSSC     .SIXBT <15><12>'SET SC '<77>
01653   052440
01654   230340
01655   770000
                        /
                        /
                        /SC+1
01656   151223          TYPLS1    .SIXBT <15><12>'SC+1   '<77>
01657   235361
01660   404040
01661   770000
                        /
```

```
                        /
                        /ALS MQ TEST
01662   151201  TYALSQ  .SIXBT <15><12>'ALS MQ TEST'<77>
01663   142340
01664   152140
01665   240523
01666   247700
                        /
                        /
                        /ALS
01667   151201  TYALS   .SIXBT <15><12>'ALS    '<77>
01670   142340
01671   404040
01672   770000
                        /TAPE 3A
                        /TYPE STRING OF CHARACTERS
                        /EOM=77=?
01673   601673  TYPTSR  JMP .
01674   506471          AND (7777
01675   046470          DAC TEMY1#
01676   226470          LAC* TEMY1
01677   446470          ISZ TEMY1
01700   041755          DAC TYPSAV
01701   742020          RTR
01702   742020          RTR
01703   742020          RTR
01704   041756          DAC TYPSAV+1
01705   742020          RTR
01706   742020          RTR
01707   742020          RTR
01710   101716          JMS TYPCHR
01711   201756          LAC TYPSAV+1
01712   101716          JMS TYPCHR
01713   201755          LAC TYPSAV
01714   101716          JMS TYPCHR
01715   601676          JMP TYPTSR+3
01716   740040  TYPCHR  HLT
01717   041757          DAC TYPSAV+2        /ACTIVE
01720   201755          LAC TYPSAV                  /TEST FOR CRLF
01721   506473          AND (777700
01722   546474          SAD (151200         /CRLF?
01723   741000          SKP                 /YES
01724   601732          JMP .+6             /NO
01725   201755          LAC TYPSAV                  /CORRECT IT FOR NEXT TIME
01726   506475          AND (000077
01727   041755          DAC TYPSAV
01730   102101          JMS TYCRLF                  /DO CRLF
01731   601713          JMP TYPCHR-3        /TYPE LAST CHARACTER
01732   201757          LAC TYPSAV+2
01733   506475          AND (77
01734   546475          SAD (77             /END OF MESSAGE?
01735   621673          JMP* TYPTSR         /YES
01736   741200          SNA                 /IF ZERO IGNOR
01737   621716          JMP* TYPCHR         /IGNOR
01740   744001          CMA!CLL
```

```
01741   346476                  TAD (40
01742   741400                  SZL
01743   601750                  JMP .+5
01744   201757                  LAC TYPSAV+2
01745   506475                  AND (77
01746   346477                  TAD (200
01747   601753                  JMP TYPSAV+2
01750   201757                  LAC TYPSAV+2
01751   506475                  AND (77
01752   346500                  TAD (300
01753   102107                  JMS OTY
01754   621716                  JMP* TYPCHR
01755   000000          TYPSAV  0                       /3RD
01756   000000                  0                       /2ND
01757   000000                  0                       /ACTIVE CHAR
01760   000000                  2
01761   000000                  0
                                .EJECT
```

```
                        /TYPE CONTENTS OF THE AC IN OCTAL
01762   601762  TYPCON  JMP .
01763   102047          JMS DECONT
01764   102070          JMS TYPOCT
01765   201761          LAC TYPSAV+4
01766   102070          JMS TYPOCT
01767   201760          LAC TYPSAV+3
01770   102070          JMS TYPOCT
01771   201757          LAC TYPSAV+2
01772   102070          JMS TYPOCT
01773   201756          LAC TYPSAV+1
01774   102070          JMS TYPOCT
01775   201755          LAC TYPSAV
01776   102070          JMS TYPOCT
01777   102075          JMS SPACE2
02000   621762          JMP* TYPCON
                        /TYPE OUT LOWEST 3 CHAR IN OCTAL
02001   602001  TYPC03  JMP .
02002   102047          JMS DECONT
02003   201757          LAC TYPSAV+2
02004   102070          JMS TYPOCT
02005   201756          LAC TYPSAV+1
02006   102070          JMS TYPOCT
02007   201755          LAC TYPSAV
02010   102070          JMS TYPOCT
02011   102075          JMS SPACE2
02012   622001          JMP* TYPC03
02013   602013  TYPTYT  JMP .
02014   102022          TSP
02015   102022          TSP
02016   102022          TSP
02017   102022          TSP
02020   102022          TSP
02021   622013          JMP* TYPTYT
02022   602022  SPAC    JMP .
02023   206501          LAC (240
02024   102107          JMS OTY
02025   622022          JMP* SPAC
                        /FORMAT FOR TWORD
                        /LAC WORD
                        /TWORD  /VALUE
                        /N      /NUMBER OF DIGITS TO PRINT FROM LEFT OF WORD
02026   740040  TOCTAL  HLT
02027   046467          DAC NUVAL#                      /VALUE OF WORD
02030   222026          LAC* TOCTAL                     /NUMBER OF WORD
02031   740001          CMA
02032   046466          DAC NUCT#               /SAVE COUNT
02033   446466          ISZ NUCT                /INC COUNT
02034   442026          ISZ TOCTAL                      /PUSH RETURN POINTER
02035   206467  TOCT1   LAC NUVAL              /LOAD VALUE
02036   742010          RTL
02037   740010          RAL                             /SHIFT INTO POSITION
02040   046467          DAC NUVAL               /SAVE SHIFTED VALUE
02041   740010          RAL                             /PASS THE LINK
02042   506502          AND (7)                         /MASK DIGIT
```

```
02043   102070          TDIGIT                  /TYPE DIGIT
02044   446466          ISZ NUCT        /MORE DIGITS
02045   602035          JMP TOCT1       /YES.
02046   622026          JMP* TOCTAL             /NO = EXIT
                        .EJECT
```

```
02047   602047      DECONT  JMP .
02050   041755              DAC TYPSAV
02051   742020              RTR
02052   740020              RAR
02053   041756              DAC TYPSAV+1
02054   742020              RTR
02055   740020              RAR
02056   041757              DAC TYPSAV+2
02057   742020              RTR
02060   740020              RAR
02061   041760              DAC TYPSAV+3
02062   742020              RTR
02063   740020              RAR
02064   041761              DAC TYPSAV+4
02065   742020              RTR
02066   740020              RAR
02067   622047              JMP* DECONT
02070   602070      TYPOCT  JMP .
02071   506502              AND (7
02072   346503              TAD (260
02073   102107              JMS OTY
02074   622070              JMP* TYPOCT
02075   602075      SPACE2  JMP .
02076   766475              LAW (77
02077   101673              TSR
02100   622075              JMP* SPACE2
02101   602101      TYCRLF  JMP .
02102   206504              LAC (215
02103   102107              JMS OTY
02104   206505              LAC (212
02105   102107              JMS OTY
02106   622101              JMP* TYCRLF
        102070      TDIGIT=JMS TYPOCT
        102026      TWORD=JMS TOCTAL
        101716      TY1=JMS TYPCHR
        102022      TSP=JMS SPAC
        101673      TSR=JMS TYPTSR                          /STRING
        102101      TCR=JMS TYCRLF                          /CR,LF
        102101      TIN=TCR
        101762      OPS=JMS TYPCON                  /CONTENTS OF AC IN OCTAL
        102013      TYT=JMS TYPTYT                          /TAB
        101762      OPT=OPS
                    /
02107   000000      OTY     0
02110   707704              LEM
02111   700406              TLS
02112   700401              TSF
02113   602112              JMP .-1
02114   622107              JMP* OTY
                            .EJECT
```

```
                        /
                        /ROTATE LEFT 6
                        /
02115   000000  RL6     0
02116   742010          RTL
02117   742010          RTL
02120   742010          RTL
02121   622115          JMP* RL6
                        /SHIFT COUNTER AND
                        /AC MQ SHIFT TEST
                        /TAPE 3  OF PDP7 EAE TEST
                        /
                        /SHIFT COUNTER TEST
                        /UTILIZES NORMALIZE INSTRUCTION
                        /WITH NO SHIFT TO DATA TEST S.C
02200                   .LOC 2200
02200   201274  SCTST1  LAC SEVSEV
02201   240033          DAC SCSTRT
02202   201322          LAC BIT1            /200000 ALREADY NORMALIZED
02203   640400          NORM=44                     /SET SC TO 00
02204   740000          NOP
02205   641001          LACS                /SC TO AC
02206   040037          DAC SCEND
02207   741200          SNA                 /READ SC=0'S TO AC?
02210   602221          JMP .+11            /YES, CONTINUE
02211   101134          JMS ERROR
02212   001632              TYCSC
02213   001427              HDR4
02214   740033          SCSTRT+740000
02215   001632              TYCSC
02216   001636              TYLACS
02217   740037          SCEND+740000
02220   000000          0
02221   100040          JMS SWITCH
02222   002206          SCTST1+6
02223   201363          LAC NBIT16
02224   041261          DAC CHARK
                        /
                        .EJECT
```

```
                        /DOES EAE NOP ALTER THE SC
                        /
02225   140033  NOPSC   DZM SCSTRT
02226   501365          AND KALL7           /MAKE MB ONES BEFORE
02227   640000          EAE                 /NOP.
02230   641001          LACS                /GET SC TO AC
02231   040037          DAC SCEND
02232   741200          SNA                 /SC STILL ZERO'S
02233   602244          JMP .+11
02234   101134          JMS ERROR
02235   001514              TYNOP
02236   001427              HDR4
02237   740033          SCSTRT+740000
02240   001514              TYNOP
02241   001636              TYLACS
02242   740037          SCEND+740000
02243   000000          0
02244   100040          JMS SWITCH
02245   002225              NOPSC
                        /
                        /DOES SC SET TO 76 AND +1 TO 77
02246   200037  SCT076  LAC SCEND
02247   040033          DAC SCSTRT
02250   201342          LAC BIT17
02251   040031          DAC MQSTRT                  /NORM 01
02252   201322          LAC BIT1
02253   040030          DAC ACSTRT
02254   640401          NORM=43                     /SET SC TO 76+1 TO 77
02255   040034          DAC ACEND
02256   641001          LACS
02257   040037          DAC SCEND
02260   541274          SAD SEVSEV
02261   602263          JMP .+2
02262   102520          JMS SCERR
02263   100040          JMS SWITCH
02264   002246              SCT076
                        /
                        /DOES SC SET TO 74 AND +1 TO 75
02265   200037  SCT074  LAC SCEND
02266   040033          DAC SCSTRT
02267   201312          LAC THREE
02270   040031          DAC MQSTRT
02271   201322          LAC BIT1
02272   640403          NORM=41                     /SC TO 74+1 TO 75
02273   040034          DAC ACEND           /SAVE FOR ERROR TYPE
02274   641001          LACS
02275   040037          DAC SCEND
02276   541311          SAD SEVFIV
02277   602301          JMP .+2
02300   102520          JMS SCERR
02301   100040          JMS SWITCH
02302   002265              SCT074
                        /
                        .EJECT
```

```
                        /DOES SC SET TO 70 AND +1 TO 71
                        /
02303   200037  SCTO70  LAC SCEND
02304   040033          DAC SCSTRT
02305   201271          LAC SEVEN
02306   040031          DAC MQSTRT
02307   201322          LAC BIT1
02310   640407          NORM+35                 /7, SC TO 70 AND +1 TO 71
02311   040034          DAC ACEND               /SAVE FOR ERROR TYPE
02312   641001          LACS
02313   040037          DAC SCEND
02314   541310          SAD SEVONE
02315   602317          JMP .+2
02316   102520          JMS SCERR
02317   100040          JMS SWITCH
02320   002303              SCTO70
                        /
                        /WILL SC SET TO 60 AND +1 TO 61
                        /
02321   200037  SCTO60  LAC SCEND
02322   040033          DAC SCSTRT
02323   201307          LAC ONESEV              /NORM 17
02324   040031          DAC MQSTRT
02325   201322          LAC BIT1
02326   640417          NORM+25                         /SET SC TO 60 AND +1 TO 61
02327   040034          DAC ACEND               /SAVE FOR ERROR TYPE
02330   641001          LACS
02331   040037          DAC SCEND
02332   541306          SAD SIXONE              /REA
02333   602335          JMP .+2
02334   102520          JMS SCERR
02335   100040          JMS SWITCH
02336   002321              SCTO60
                        /
                        /WILL SC SET TO 40 AND +1 TO 41
                        /
02337   200037  SCTO40  LAC SCEND
02340   040033          DAC SCSTRT
02341   201305          LAC THREE7              /NORM 37
02342   040031          DAC MQSTRT
02343   201322          LAC BIT1                /20000 ALREADY NORMALIZED
02344   640437          NORM+5                          /SET SC TO 40 AND +1 TO 41
02345   040034          DAC ACEND
02346   641001          LACS                    /GET SC TO AC
02347   040037          DAC SCEND               /SAVE FOR ERROR TYPE
02350   541304          SAD FOUR1               /READ 41 FROM SC TO AC
02351   602353          JMP .+2                 /YES
02352   102520          JMS SCERR
02353   100040          JMS SWITCH
02354   002337              SCTO40
                        /
                        .EJECT
```

```
                        /WILL SC SET TO 0 AND +1 TO 1
                        /
02355   200037  SCT000  LAC SCEND
02356   040033           DAC SCSTRT
02357   201274           LAC SEVSEV          /NORM 77
02360   040031           DAC MQSTRT
02361   201322           LAC BIT1
02362   640477           NORM +33            /SC TO 00 +1 TO 01
02363   040034           DAC ACEND
02364   641001           LACS
02365   040037           DAC SCEND
02366   541342           SAD BIT17           /SC READ 01?
02367   602371           JMP .+2             /YES
02370   102520           JMS SCERR
02371   100040           JMS SWITCH
02372   002355               SCT000
                        /
                        /WILL SC SET TO 01 AND +1 TO 02
                        /
02373   200037  SCT001  LAC SCEND
02374   040033           DAC SCSTRT
02375   201275           LAC SEVSIX          /NORM 76
02376   040031           DAC MQSTRT
02377   201322           LAC BIT1
02400   640476           NORM 32             /SET SC TO 1 +1 TO 2
02401   040034           DAC ACEND
02402   641001           LACS
02403   040037           DAC SCEND
02404   541341           SAD BIT16
02405   602407           JMP .+2
02406   102520           JMS SCERR
02407   100040           JMS SWITCH
02410   002373               SCT001
                        /
                        /WILL SC SET TO 03 AND +1 TO 04
                        /
02411   200037  SCT003  LAC SCEND
02412   040033           DAC SCSTRT
02413   201303           LAC SEVN4           /NORM 74
02414   040031           DAC MQSTRT
02415   201322           LAC BIT1
02416   640474           NORM +30            /SET SC TO 3 +1 TO 4
02417   040034           DAC ACEND
02420   641001           LACS
02421   040037           DAC SCEND
02422   541340           SAD BIT15 /SC TO AC =4?
02423   602425           JMP .+2             /YES
02424   102520           JMS SCERR
02425   100040           JMS SWITCH
02426   002411               SCT003
                        /
                        .EJECT
```

```
                        /WILL SC SET TO 07 AND +1 TO 10
02427   200037  SCT007  LAC SCEND
02430   240033          DAC SCSTRT
02431   201302          LAC SEVNTY          /NORM 70
02432   240031          DAC MQSTRT
02433   201322          LAC BIT1
02434   640470          NORM +24            /SET SC TO 7 +1 TO 10
02435   240034          DAC ACEND
02436   641001          LACS
02437   240037          DAC SCEND
02440   541337          SAD BIT14           /SC TO AC = 10?
02441   602443          JMP .+2             /YES
02442   122527          JMS SCERR
02443   100040          JMS SWITCH
02444   002427              SCT007
                        /
                        /WILL SC SET TO 17 AND +1 TO 20
02445   200037  SCT017  LAC SCEND
02446   240033          DAC SCSTRT
02447   201301          LAC SIXTY           /NORM 60
02450   240031          DAC MQSTRT
02451   201322          LAC BIT1
02452   640460          NORM +14            /SC TO 17+1 TO 20
02453   240037          DAC SCEND
02454   641001          LACS
02455   240037          DAC SCEND
02456   541336          SAD BIT13           /SC TO AC = 20?
02457   602461          JMP .+2             /YES
02460   102520          JMS SCERR
02461   100040          JMS SWITCH
02462   002445              SCT017
                        /
                        /WILL SC SET TO 37 AND +1 TO 40
                        /
02463   200037  SCT037  LAC SCEND
02464   200033          LAC SCSTRT
02465   201335          LAC BIT12           /NORM 40
02466   240031          DAC MQSTRT
02467   201322          LAC BIT1
02470   640440          NORM=4                      /SET SC TO 37 +1 TO 40
02471   240034          DAC ACEND
02472   641001          LACS
02473   240037          DAC SCEND
02474   541335          SAD BIT12           /SC TO AC = 40?
02475   602477          JMP .+2             /YES
02476   102520          JMS SCERR
02477   100040          JMS SWITCH
02500   002463              SCT037
                        /
                        .EJECT
```

```
                        /WILL SC SET TO 77 AND +1 TO 00
                        /
02501   200037  SCT077  LAC SCEND
02502   742033          DAC SCSTRT
02503   201274          LAC SEVSEV          /NORM 0
02504   742031          DAC MQSTRT
02505   201322          LAC BIT1
02506   640400          NORM+44                     /SET SC TO 77 AND +1 TO 0
02507   742034          DAC ACEND
02510   641001          LACS                /GET SC TO AC
02511   742037          DAC SCEND
02512   741200          SNA                 /SC TO AC = 00?
02513   602515          JMP .+2             /YES
02514   102520          JMS SCERR
02515   100040          JMS SWITCH
02516   002501          SCT077
02517   602555              JMP NOPSC1
                        /
                        .EJECT
```

```
02520   602520      SCERR   JMP .
02521   200031              LAC MQSTRT          /GET SC OF NORM
02522   740001              CMA
02523   501274              AND SEVSEV          /SHOULD SET SC TO
02524   040035              DAC MQEND
02525   341342              TAD BIT17
02526   501274              AND SEVSEV          /SC SHOULD +1 TO
02527   040036              DAC LKEND
02530   101134              JMS ERROR           /TYPE OUT
02531   001642                  TYSCER                  /SC ERROR
02532   602520              SCERR+600000        /ERROR ADDRESS
02533   001427                  HDR4
02534   740033              SCSTRT+740000       /SC AT START
02535   001442                  SPACE3
02536   600030              ACSTRT+600000       /AC AT START
02537   001646                  TYNORM
02540   740031              MQSTRT+740000       /SC PORTION OF NORM
02541   001652                  TYSSC
02542   740035              MQEND+740000        /SHOULD SET SC TO
02543   001656                  TYPLS1
02544   740036              LKEND+740000        /SC SHOULD +1 TO
02545   001461                  TYCOR
02546   001636                  TYLACS
02547   740037              SCEND+740000        /SC TO AC EQUALED
02550   001463                  TYINCO
02551   001442                  SPACE3
02552   600034              ACEND+600000        /AC AFTER NORM
02553   000000              0
02554   622520              JMP* SCERR
                    /
                    /
                    /DOES EAE NOP ALTER SC = 77
                    /
02555   201274      NOPSC1  LAC SEVSEV
02556   040033              DAC SCSTRT
02557   201322              LAC BIT1
02560   640401              NORM+43                 /SET SC TO 77
02561   501365              AND KALL7           /MAKE MB TO ONES BEFORE
02562   640077              EAE+77              /NOP SHOULD NOT ALTER SC
02563   641001              LACS                /GET SC TO AC
02564   040037              DAC SCEND
02565   540033              SAD SCSTRT          /SC TO AC = 77?
02566   622600              JMP .+12
02567   101134              JMS ERROR
02570   001514                  TYNOP
02571   001427                  HDR4
02572   740033              SCSTRT+740000
02573   001514                  TYNOP
02574   740033              SCSTRT+740000
02575   001636                  TYLACS
02576   740037              SCEND+740000
02577   000000              2
02600   100040              JMS SWITCH
02601   002555                  NOPSC1
02602   100060              JMS SWTCHS
```

```
02603  002246                    SCT076
                                 .EJECT
```

```
                        /SHIFT TESTS
                        /ALS - ACCUMULATOR LEFT SHIFT
                        /DOES ALS AC = 0'S ALTER THE AC?
                        /
02604   140030  ALSZER  DZM ACSTRT
02605   140031          DZM MQSTRT
02606   140032          DZM LKSTRT
02607   140033          DZM SCSTRT
02610   651000          CLQ*1000            /CLEAR AC - MQ AND LINK
02611   744000          CLL
02612   640700          ALS
02613   040034          DAC ACEND
02614   750010          GLK
02615   040036          DAC LKEND
02616   641001          LACS
02617   040037          DAC SCEND
02620   200034          LAC ACEND
02621   741200          SNA
02622   741000          SKP
02623   103175          JMS ALSERR
02624   100040          JMS SWITCH
02625   002610              ALSZER+4
                        /
                        /
                        /DOES ALS 01 AC = 0'S OK
                        /
02626   201342  ALS01   LAC BIT17           /ALS 01
02627   040033          DAC SCSTRT
02630   140030          DZM ACSTRT          /AC 0'S TO START
02631   140031          DZM MQSTRT          /MQ 0'S
02632   140032          DZM LKSTRT          /LINK IS ZERO
02633   650000          CLQ
02634   641000          EAE*1000
02635   744000          CLL
02636   640701          ALS 01              /SHIFT AC LEFT 1
02637   040034          DAC ACEND
02640   750010          GLK
02641   040036          DAC LKEND           /LINK FOR TYPEOUTS
02642   641001          LACS
02643   040037          DAC SCEND           /SC FOR TYPEOUTS
02644   200034          LAC ACEND
02645   741200          SNA
02646   741000          SKP
02647   103175          JMS ALSERR
02650   200031          LAC MQSTRT
02651   652000          LMQ
02652   100040          JMS SWITCH
02653   002634              ALS01+6
02654   200031          LAC MQSTRT
02655   740200          SZA
02656   002663          JMP .+5
02657   750001          CLC
02660   040031          DAC MQSTRT          /2ND PASS MQ = 1'S
02661   640004          EAE*4
02662   002634          JMP ALS01+6
```

/

.EJECT

```
                        /LINK TO AC 17
                        /BIT = 0 L=0, BIT =0 L=1, BIT = 1 L = 0, BIT = 1 L = 1
                        /
02663   140030  ALSLNK  DZM ACSTRT          /START AC 0'S
02664   140031          DZM MQSTRT
02665   140032          DZM LKSTRT          /LINK START 0
02666   650000          CLQ
02667   200032          LAC LKSTRT
02670   740020          RAR                 /LINK = 0 OR 1
02671   200030          LAC ACSTRT
02672   640701          ALS 01
02673   040034          DAC ACEND
02674   750010          GLK
02675   040036          DAC LKEND
02676   641001          LACS
02677   040037          DAC SCEND
02700   200030          LAC ACSTRT
02701   740010          RAL
02702   340032          TAD LKSTRT
02703   540034          SAD ACEND
02704   741000          SKP
02705   103175          JMS ALSERR
02706   100040          JMS SWITCH
02707   002666          ALSLNK+3
02710   200032          LAC LKSTRT
02711   440032          ISZ LKSTRT
02712   741200          SNA                 /2ND PASS L=1
02713   602666          JMP ALSLNK+3
02714   140032          DZM LKSTRT
02715   200030          LAC ACSTRT
02716   440030          ISZ ACSTRT
02717   741200          SNA                 /3RD AND 4TH PASS AC=1
02720   602666          JMP ALSLNK+3
                        /
                        .EJECT
```

```
                        /DOES ALS ALTER THE LINK = 1 OR 0
                        /
02721   140031  LNKALS  DZM MQSTRT          /MQ ALWAYS = 0
02722   140030          DZM ACSTRT          /START AC=0
02723   140032          DZM LKSTRT          /LINK START 0
02724   201342          LAC BIT17
02725   040033          DAC SCSTRT          /SC = 01
02726   650000          CLQ
02727   200032          LAC LKSTRT
02730   740020          RAR                 /LINK = 1 O R 0
02731   200030          LAC ACSTRT          /AC = 0 OR 400000
02732   640701  KALS01  ALS 01
02733   040034          DAC ACEND           /SAVE AC RESULT
02734   641001          LACS
02735   040037          DAC SCEND           /SAVE SC RESULT
02736   750010          CLK
02737   040036          DAC LKEND
02740   540032          SAD LKSTRT          /LINK SAME AS STRT?
02741   741000          SKP                 /YES
02742   103175          JMS ALSERR
02743   100040          JMS SWITCH
02744   002726          LNKALS+5
02745   200032          LAC LKSTRT
02746   440032          ISZ LKSTRT
02747   741200          SNA
02750   602726          JMP LNKALS+5        /2ND AND 4TH PAS L = 1
02751   200030          LAC ACSTRT
02752   201321          LAC BIT0
02753   540030          SAD ACSTRT                  /AC 0 ALREADY = 1
02754   602760          JMP .+4
02755   140032          DZM LKSTRT                  /3RD AND 4TH PASS
02756   040030          DAC ACSTRT          /AC=400000
02757   602726          JMP LNKALS+5
                        /
                        .EJECT
```

```
                        /DOES ALS ALTER THE MQ
                        /
02760   140031  ALSMQT  DZM MQSTRT          /1ST PASSES MQ = 0'S
02761   140030          DZM ACSTRT
02762   140032          DZM LKSTRT
02763   201342          LAC BIT17
02764   040033          DAC SCSTRT          /ALS 01 PLACE
02765   200032          LAC LKSTRT
02766   740020          RAR                 /L=1 OR 0
02767   200031          LAC MQSTRT
02770   652000          LMQ                 /MQ = 0'S OR 1'S
02771   200030          LAC ACSTRT          /AC = 0'S OR 1'S
02772   640701          ALS 01
02773   040034          DAC ACEND
02774   750010          CLK
02775   040036          DAC LKEND
02776   641001          LACS
02777   040037          DAC SCEND
03000   641002          LACQ
03001   040035          DAC MQEND
03002   540031          SAD MQSTRT          /MQ SAME AS START?
03003   603017          JMP .+14  /YES
03004   101134          JMS ERROR
03005   001662              TYALSQ
03006   001411              HDR3
03007   700032          LKSTRT+700000
03010   600030          ACSTRT+600000
03011   000031              MQSTRT
03012   001667              TYALS
03013   700036          LKEND+700000
03014   600034          ACEND+600000
03015   600035          MQEND+600000
03016   000000          0
03017   100040          JMS SWITCH
03020   002765          ALSMQT+5
03021   200032          LAC LKSTRT
03022   440032          ISZ LKSTRT          /EVERY OTHER PASS L = 1
03023   741200          SNA
03024   602765          JMP ALSMQT+5
03025   140032          DZM LKSTRT          /NEXT PASS L = 0
03026   200030          LAC ACSTRT
03027   740001          CMA                 /AC=0'S, 1'S, 0'S, 1'S
03030   040030          DAC ACSTRT
03031   740200          SZA
03032   602765          JMP ALSMQT+5
03033   200031          LAC MQSTRT          /MQ = 0'S 4 PASSES 1'S 4 PASSES
03034   740001          CMA
03035   040031          DAC MQSTRT
03036   740200          SZA                 /MQSTRT BACK TO 0'S?
03037   602765          JMP ALSMQT+5        /NO, TEST M1 = 1'S
03040   100060          JMS SWTCHS
03041   002694              ALSZER
                        /
                        .EJECT
```

```
                        /WILL AC0 GO TO LINK PROPERLY
                        /IMMEDIATELY FOLLOWING AN ALS LEFT SHIFT
                        /0-0,0-1,1-0,1-1
03042   140232  SGNSHF  DZM LKSTRT                      /LK TO 0 FIRST
03043   140231          DZM MQSTRT                      /TO COMPARE LINK ONLY
03044   200031          LAC BIT0
03045   040030          DAC ACSTRT                      /FIRST AC0=1 GOES TO 0
03046   201342          LAC BIT17
03047   040033          DAC SCSTRT                      /SHIFT=1 0PLACE
03050   200032          LAC LKSTRT
03051   740020          RAR                     /MAKE L=START
03052   200030          LAC ACSTRT
03053   640701          ALS 01                  /AC0=1 GOES TO 0 OR = 0 GOES TO 1
03054   660000          EAE*20000               /GET SIGN OF AC
03055   040034          DAC ACEND               /SAVE FOR TYPEOUTS
03056   750010          GLK
03057   040036          DAC LKEND               /SAVE FOR TYPEOUTS
03060   540031          SAD MQSTRT                      /L=CORRECT RESULT
03061   603077          JMP NSNERR                      /YES
03062   101134          JMS ERROR
03063   001667              TYALS
03064   740033          SCSTRT+740000
03065   001537              TYSLK
03066   001411              HDR3
03067   700032          LKSTRT+700000
03070   600030          ACSTRT+600000
03071   001667          TYALS
03072   001537          TYSLK
03073   700036          LKEND+700000
03074   600034          ACEND+600000
03075   001607          TYRES
03076   000000          0
03077   100040  NSNERR  JMS SWITCH                      /END SCOPE LOOP
03100   003050          SGNSHF+6
03101   200032          LAC LKSTRT
03102   440032          ISZ LKSTRT
03103   741200          SNA
03104   603050          JMP SGNSHF+6            /THIS PASS L=1
03105   140032          DZM LKSTRT
03106   201322          LAC BIT1
03107   540030          SAD ACSTRT                      /TESTED SIGN=1
03110   603114          JMP HSALS               /YES
03111   040030          DAC ACSTRT
03112   440031          ISZ MQSTRT
03113   603050          JMP SGNSHF+6
                        /
                        .EJECT
```

```
                        /WILL ALS SHIFT 1 TO 18 PLACES?
                        /1ST PASS BIT 2ND PASS NO BIT
                        /
03114   140031  HSALS   DZM MQSTRT
03115   201342          LAC BIT17
03116   040030          DAC ACSTRT
03117   140032          DZM LKSTRT
03120   103216          JMS SIMALS
03121   201300          LAC K18
03122   040010          DAC 10
03123   202732          LAC KALS01
03124   043134          DAC HSALSE
03125   201342          LAC BIT17
03126   040033          DAC SCSTRT
03127   200032  HSALSL  LAC LKSTRT
03130   740020          RAR
03131   200031          LAC MQSTRT          /MQ ALTERNATES
03132   652000          LMQ                 /FROM 1'S TO 0'S
03133   200030          LAC ACSTRT
03134   640701  HSALSE  ALS 01              /1 TO 18 PLACES
03135   040034          DAC ACEND
03136   750010          GLK
03137   040036          DAC LKEND
03140   641001          LACS
03141   040037          DAC SCEND
03142   740200          SZA                 /SC GO TO ZERO?
03143   603151          JMP .+6
03144   200036          LAC LKEND
03145   540032          SAD LKSTRT          /WAS LINK ALTERED
03146   200034          LAC ACEND
03147   563244          SAD* SALSRP         /RESULT OF SHIFT OK?
03150   603152          JMP .+2
03151   103175          JMS ALSERR
03152   100040          JMS SWITCH
03153   003127              HSALSL
03154   200031          LAC MQSTRT
03155   740001          CMA                 /EVEN PASSES MQ = 777777
03156   040031          DAC MQSTRT
03157   443134          ISZ HSALSE                  /INCREMENT COUNT
03160   443244          ISZ SALSRP          /ADVANCE RESULT POINTER
03161   440033          ISZ SCSTRT          /FOR TYPEOUTS SC+1
03162   440010          ISZ 10              /SHIFT 18 TIMES?
03163   603127          JMP HSALSL
03164   440032          ISZ LKSTRT          /NO BIT PASS L = 1
03165   200030          LAC ACSTRT
03166   740001          CMA                 /2ND PASS AC STRT=777776
03167   040030          DAC ACSTRT
03170   741100          SPA                 /MADE 2 PASSES?
03171   603120          JMP HSALS+4         /NO, SHIFT NO BIT
03172   100060          JMS SWTCHS
03173   003114              HSALS
03174   603310          JMP LLSTS1
                        /
                        .EJECT
```

```
                        /ALS INSTRUCTION
                        /COMMON ERROR TYPEOUT
                        /
03175   603175  ALSERR  JMP .
03176   101134          JMS ERROR
03177   001667              TYALS
03200   740033          SCSTRT+740000
03201   403175          ALSERR+400000
03202   001447              HDR5
03203   500032          LKSTRT+500000
03204   600030          ACSTRT+600000
03205   600031          MQSTRT+600000
03206   001457              TYPATR
03207   500036          LKEND+500000
03210   600034          ACEND+600000
03211   001607              TYRES
03212   001636              TYLACS
03213   740037          SCEND+740000
03214   000000          0
03215   623175          JMP* ALSERR
                        /
                        /
                        /SIMULATE ALS OPERATION
                        /STORES SHIFTS 1 TO 18 PLACES
                        /
03216   603216  SIMALS  JMP .
03217   206506          LAC (RESULT-1
03220   040017          DAC 17
03221   040015          DAC 15
03222   341342          TAD BIT17
03223   043244          DAC SALSRP          /SET RESULT POINTER TO START
03224   201300          LAC K18
03225   040016          DAC 16
03226   200032          LAC LKSTRT
03227   740020          RAR
03230   200030          LAC ACSTRT
03231   740010          RAL
03232   060017          DAC* 17
03233   440016          ISZ 16
03234   200032          LAC LKSTRT
03235   740020          RAR
03236   220015          LAC* 15
03237   740010          RAL
03240   060017          DAC* 17
03241   440016          ISZ 16
03242   603234          JMP .-6
03243   623216          JMP* SIMALS
03244   000000  SALSRP  0
03245   000000  RESULT  0
                        /
03267                   .LOC RESULT+22              /RESERVE 17 SHIFT LOCATIONS
                        /
                        .EJECT
```

```
03267   750004   ENDSHF   LAS
03270   501327            AND BIT6
03271   741200            SNA                  /COMMA AT END?
03272   603300            JMP .+6              /NO
03273   760054            LAW 54
03274   101716            TY1
03275   441261            ISZ CHARK
03276   603302            JMP .+4
03277   101240            JMS CRLF
03300   201363            LAC NBIT16
03301   041261            DAC CHARK
03302   750004            LAS
03303   501326            AND BIT5
03304   741200            SNA                  /CYCLE BOTH TESTS
03305   602246            JMP SCTO76           /NO, STAY IN SHIFT TEST
03306   605002            JMP RANSHF           /REPEAT FROM SETUP TEST
03307   000254   COMMA    254
                 /
                 /LLS  AND LRS BASIC TESTS
                 /TAPE 4   OF EAE PDP7 TEST
                 /
                 /
                 /LONG LEFT SHIFT
                 /
                 /
                 /LLS 01  ALL ZERO'S
                 /
03310   140030   LLSTS1   DZM ACSTRT
03311   140031            DZM MQSTRT
03312   140032            DZM LKSTRT
03313   201342            LAC BIT17
03314   040033            DAC SCSTRT
03315   650000            CLQ                  /START SCOPE LOOP
03316   754000            CLA!CLL              /CLR AC AND LINK
03317   640601            LLS 01
03320   040034            DAC ACEND
03321   750010            GLK
03322   040036            DAC LKEND
03323   641001            LACS
03324   040037            DAC SCEND
03325   641002            LACQ
03326   040035            DAC MQEND
03327   741200            SNA                  /MQ STILL 0'S?
03330   200034            LAC ACEND
03331   741200            SNA                  /AC STILL 0'S?
03332   200036            LAC LKEND
03333   741200            SNA                  /LINK STILL 0'S?
03334   200037            LAC SCEND
03335   741200            SNA                  /SC GO TO ZERO?
03336   603340            JMP .+2
03337   103745            JMS LLSERR
03340   100040            JMS SWITCH
03341   003315                LLSTS1+5
                 /
                           .EJECT
```

```
                        /DOES LINK GO TO MQ17 ON AN LLS
                        /0=0, 1=0, 0=1, 1=1
                        /
03342   140031  LLSTS2  DZM MQSTRT
03343   140030          DZM ACSTRT
03344   140032          DZM LKSTRT
03345   201342          LAC BIT17               /LLS  01
03346   040033          DAC SCSTRT
03347   200031          LAC MQSTRT              /2 PASSES = 0  START SCOPE LOOP
03350   652000          LMQ                     /2 PASSES = 1 (MQ17)
03351   200032          LAC LKSTRT
03352   740020          RAR                     /L=1 EVERY 2ND PASS
03353   200030          LAC ACSTRT              /AC ALWAYS = 0
03354   640601          LLS 01
03355   040034          DAC ACEND               /SAVE RESULTS
03356   750010          GLK
03357   040036          DAC LKEND
03360   641001          LACS
03361   040037          DAC SCEND
03362   641002          LACQ
03363   040035          DAC MQEND
03364   200032          LAC LKSTRT
03365   740020          RAR
03366   200031          LAC MQSTRT
03367   740010          RAL
03370   540035          SAD MQEND
03371   741000          SKP
03372   103745          JMS LLSERR
03373   100040          JMS SWITCH                      /END SCOPE LOOP
03374   003347              LLSTS2+5
03375   200032          LAC LKSTRT
03376   440032          ISZ LKSTRT
03377   741200          SNA                     /2ND OR 4TH PASS?
03400   603347          JMP LLSTS2+5
03401   140032          DZM LKSTRT              /NEXT PASS L = 0
03402   200031          LAC MQSTRT
03403   440031          ISZ MQSTRT
03404   741200          SNA                     /MADE WITH MQ17=1?
03405   603347          JMP LLSTS2+5
                        /
                        .EJECT
```

```
                        /DOES LINK NOT GO TO AC17 ON AN LLS
                        /DOES MQ0 GO TO AC17 ON AN LLS
                        /
03406   140030  LLSACT  DZM ACSTRT
03407   140031          DZM MQSTRT
03410   140032          DZM LKSTRT
03411   201342          LAC BIT17
03412   040033          DAC SCSTRT
03413   200031          LAC MQSTRT              /START SCOPE LOOP
03414   652000          LMQ
03415   200032          LAC LKSTRT
03416   740020          RAR                     /L=0, 1, 0, 1
03417   200030          LAC ACSTRT              /AC=0, 0, 1, 1
03420   640601          LLS 01
03421   040034          DAC ACEND
03422   750010          GLK
03423   040036          DAC LKEND
03424   641001          LACS
03425   040037          DAC SCEND               /SAVE SC FOR TYPEOUT
03426   641002          LACQ
03427   040035          DAC MQEND               /MQ FOR TYPEOUT
03430   540032          SAD LKSTRT              /LINK TO MQ17?
03431   741000          SKP                     /YES, OK
03432   603441          JMP .+7                 /MQ ERROR
03433   200031          LAC MQSTRT
03434   740010          RAL
03435   200030          LAC ACSTRT
03436   740010          RAL
03437   540034          SAD ACEND               /AC0 SHOULD BE = MQ0
03440   741000          SKP
03441   103745          JMS LLSERR
03442   100040          JMS SWITCH
03443   003413          LLSACT+5
03444   200032          LAC LKSTRT
03445   440032          ISZ LKSTRT              /L=0,1,0, 1,0, 1, 0, 1
03446   741200          SNA
03447   603413          JMP LLSACT+5
03450   140032          DZM LKSTRT
03451   200030          LAC ACSTRT
03452   440030          ISZ ACSTRT              /AC0 = 0, 0, 1, 1, 0, 0, 1, 1
03453   741200          SNA
03454   603413          JMP LLSACT+5
03455   140030          DZM ACSTRT
03456   201321          LAC BIT0
03457   540031          SAD MQSTRT              /TESTED MQ0 = 1?
03460   603463          JMP .+3                 /YES
03461   040031          DAC MQSTRT              /MQ0 = 0, 4 PASSES
03462   603413          JMP LLSACT+5            /=1,  4 PASSES
03463   100060          JMS SWTCHS
03464   003310              LLST61
                        /
                        .EJECT
```

```
                        /WILL EACH BIT OF THE MQ SHIFT TO THE NEXT
                        /1=0, AND 0=1 LEFT
                        /
03465   201342  LLSTS3  LAC BIT17               /START MQ 17 TO MQ 16
03466   040031          DAC MQSTRT
03467   040033          DAC SCSTRT
03470   140032          DZM LKSTRT
03471   140030          DZM ACSTRT
03472   200031          LAC MQSTRT                      /START SCOPE LOOP
03473   652000          LMQ
03474   754000          CLA!CLL                 /AC AND L ALWAYS 0'S
03475   640601          LLS 01
03476   040034          DAC ACEND
03477   750010          GLK
03500   040036          DAC LKEND               /FOR TYPEOUTS
03501   641001          LACS
03502   040037          DAC SCEND               /FOR TYPEOUTS
03503   641002          LACQ
03504   040035          DAC MQEND
03505   200031          LAC MQSTRT
03506   740010          RAL
03507   540035          SAD MQEND
03510   741000          SKP
03511   603516          JMP .+5
03512   200030          LAC ACSTRT
03513   740010          RAL
03514   540034          SAD ACEND
03515   603517          JMP .+2
03516   103745          JMS LLSERR
03517   100040          JMS SWITCH                      /END SCOPE
03520   003472          LLSTS3+5
03521   200031          LAC MQSTRT              /SET UP NEXT MQ BIT
03522   744010          CLL!RAL
03523   040031          DAC MQSTRT
03524   740400          SNL                     /TESTED MQ0 = 1
03525   603472          JMP LLSTS3+5
                        /
                        .EJECT
```

```
                        /WILL EACH BIT OF THE MQ SHIFT TO THE NEXT
                        /1-1, 0-1, 1-0 LEFT
                        /
03526   201364  LLSTS4  LAC NBIT17          /START 777776
03527   040031          DAC MQSTRT
03530   740001          CMA
03531   040033          DAC SCSTRT          /LLS 01
03532   040032          DAC LKSTRT          /LINK ALWAYS = 1
03533   750001          CLC
03534   040030          DAC ACSTRT          /AC = 1'S ALL
03535   200031          LAC MQSTRT                  /START SCOPE LOOP
03536   652000          LMQ
03537   754003          STL!CLC
03540   640601          LLS 01
03541   040034          DAC ACEND
03542   750010          GLK
03543   040036          DAC LKEND           /L, FOR TYPEOUT
03544   641001          LACS
03545   040037          DAC SCEND           /SC FOR TYPEOUT
03546   641002          LACQ
03547   040035          DAC MQEND
03550   200031          LAC MQSTRT                  /SIMULATE LLS
03551   744002          STL                 /TO GET
03552   740010          RAL                 /COMPARE CONSTANT
03553   540035          SAD MQEND           /MQ SHIFT OK?
03554   741000          SKP                 /YES
03555   603562          JMP .+5
03556   200030          LAC ACSTRT
03557   740010          RAL
03560   540034          SAD ACEND           /AC SHIFT OK?
03561   603563          JMP .+2
03562   103745          JMS LLSERR
03563   100040          JMS SWITCH                  /END SCOPE
03564   003535          LLSTS4+7
03565   744002          STL
03566   200031          LAC MQSTRT
03567   740010          RAL
03570   040031          DAC MQSTRT
03571   741400          SZL                 /TESTED MQ0 = 0
03572   603535          JMP LLSTS4+7
03573   100060          JMS SWTCHS
03574   003465              LLSTS3
                        /
                        .EJECT
```

```
                        /WILL MQ AC SHIFT A 1 BIT 1 TO 44 PLACES
                        /USES LLS SIGNED
                        /
03575   140030  LLSTS5  DZM ACSTRT              /AC START ZEROS
03576   201342          LAC BIT17
03577   040033          DAC SCSTRT      /SC INCREMENTED TO 44
03600   040031          DAC MQSTRT      /MQ START BIT 17 = 1
03601   040032          DAC LKSTRT
03602   044701          DAC MQCOMK
03603   144700          DZM ACCOMK
03604   203767          LAC KLLSS1
03605   043617          DAC LLSSEX      /RESET SHIFT TO 1
03606   204701  LLSSL1  LAC MQCOMK
03607   744010          CLL!RAL
03610   044701          DAC MQCOMK
03611   204700          LAC ACCOMK
03612   740010          RAL
03613   044700          DAC ACCOMK
03614   200031          LAC MQSTRT              /START SCOPE LOOP
03615   652000          LMQ
03616   754002          STL!CLA
03617   660601  LLSSEX  LLSS 01         /SC = 1 TO 44
03620   040034          DAC ACEND
03621   641001          LACS
03622   040037          DAC SCEND
03623   750010          GLK
03624   040036          DAC LKEND
03625   641002          LACQ
03626   040035          DAC MQEND
03627   544701          SAD MQCOMK
03630   741000          SKP
03631   603635          JMP .+4
03632   204700          LAC ACCOMK
03633   540034          SAD ACEND
03634   741000          SKP
03635   603643          JMP .+6
03636   200036          LAC LKEND
03637   741200          SNA             /LINK GO TO  0
03640   200037          LAC SCEND
03641   741200          SNA             /SC END = 0
03642   603644          JMP .+2
03643   103770          JMS LLSSER              /END SCOPE LOOP
03644   100040          JMS SWITCH
03645   003614          LLSSEX-3
03646   443617          ISZ LLSSEX
03647   440033          ISZ SCSTRT
03650   200033          LAC SCSTRT
03651   241313          XOR FOUR5
03652   740200          SZA
03653   603606          JMP LLSSL1
03654   100060          JMS SWTCHS
03655   003575              LLSTS5
                        /
                        .EJECT
```

```
                        /WILL MQ AC SHIFT A NO BIT 1 TO 44 PLACES
                        /
03656   140032  LLSTS6  DZM LKSTRT
03657   201342          LAC BIT17
03660   040033          DAC SCSTRT
03661   740001          CMA
03662   040031          DAC MQSTRT
03663   044701          DAC MQCOMK
03664   750001          CLC
03665   040030          DAC ACSTRT
03666   044700          DAC ACCOMK
03667   203767          LAC KLLSS1
03670   043703          DAC LLSSX2
03671   204701  LLSSL2  LAC MQCOMK          /FORM AC
03672   744002          STL
03673   740010          RAL                 /AND MQ
03674   044701          DAC MQCOMK          /COMPARE CONSTANTS
03675   204700          LAC ACCOMK
03676   740010          RAL
03677   044700          DAC ACCOMK
03700   200031          LAC MQSTRT                  /SET UP SHIFT  START SCOPE LOOP
03701   652000          LMQ
03702   754001          CLL!CLC
03703   660601  LLSSX2  LLSS 01            /SC=1 TO 44 PLACES
03704   040034          DAC ACEND
03705   641001          LACS
03706   040037          DAC SCEND           /GET SC FOR TEST 4
03707   750010          GLK
03710   040036          DAC LKEND           /LINK SHOULD BE 1
03711   641002          LACQ
03712   040035          DAC MQEND
03713   544701          SAD MQCOMK          /MQ SHIFT OK?
03714   741000          SKP
03715   603721          JMP .+4
03716   204700          LAC ACCOMK
03717   540034          SAD ACEND           /AC SHIFT OK?
03720   741000          SKP
03721   603725          JMP .+4
03722   200036          LAC LKEND
03723   541342          SAD BIT17           /LINK SET TO 1?
03724   741000          SKP
03725   603731          JMP .+4
03726   200037          LAC SCEND
03727   741200          SNA                 /SC GO TO 0?
03730   741000          SKP
03731   103770          JMS LLSSER
03732   100040          JMS SWITCH
03733   003700          LLSSX2-3
03734   443703          ISZ LLSSX2          /ADVANCE TO NEXT SHIFT
03735   440033          ISZ SCSTRT
03736   200033          LAC SCSTRT
03737   241313          XOR FOURS           /SHIFTED 44 PLACES?
03740   740200          SZA
03741   603671          JMP LLSSL2
03742   100060          JMS SWICHS          /REPEAT SEQUENCE SET?
```

```
03743   003656                  LLSTS6
03744   604016                JMP LRSTS1

                              .EJECT
```

```
                        /COMMON ERROR TYPEOUT LLS
                        /
03745   603745  LLSERR  JMP .
03746   101134          JMS ERROR
03747   001553              TYLLS
03750   740033          SCSTRT+740000
03751   403745          LLSERR+400000
03752   001447              HDR5
03753   500032          LKSTRT+500000
03754   600030          ACSTRT+600000
03755   600031          MQSTRT+600000
03756   001457              TYPATR
03757   500036          LKEND+500000
03760   600034          ACEND+600000
03761   600035          MQEND+600000
03762   001607              TYRES
03763   001636              TYLACS
03764   740037          SCEND+740000
03765   000000          0
03766   623745          JMP* LLSERR
03767   660601  KLLSS1  LLSS 01             /TO SET UP LONG LEFT SHIFTS
                        /
                        /
                        /COMMON ERROR TYPEOUT
                        /LLS SIGNED
                        /
03770   603770  LLSSER  JMP .
03771   101134          JMS ERROR
03772   001557              TYLLSS
03773   740033          SCSTRT+740000
03774   403770          LLSSER+400000
03775   001447              HDR5
03776   500032          LKSTRT+500000
03777   600030          ACSTRT+600000
04000   600031          MQSTRT+600000
04001   001457              TYPATR
04002   001444              SPACE4
04003   604700          ACCOMK+600000
04004   604701          MQCOMK+600000
04005   001461              TYCOR
04006   500036          LKEND+500000
04007   600034          ACEND+600000
04010   600035          MQEND+600000
04011   001463              TYINCO
04012   001636              TYLACS
04013   740037          SCEND+740000
04014   000000          0
04015   623770          JMP* LLSSER
                        /
                        .EJECT
```

```
                        /LONG RIGHT SHIFT
                        /LRS Ø1 AC, MQ AND L = Ø'S
                        /
04016   140030  LRSTS1  DZM ACSTRT          /SET INITIAL CONDITIONS
04017   140031          DZM MQSTRT
04020   140032          DZM LKSTRT
04021   201342          LAC BIT17
04022   040033          DAC SCSTRT
04023   650000          CLQ                 /START SCOPE LOOP
04024   754000          CLA!CLL
04025   640501          LRS Ø1
04026   040034          DAC ACEND
04027   750010          GLK
04030   040036          DAC LKEND
04031   641001          LACS
04032   040037          DAC SCEND
04033   641002          LACQ
04034   040035          DAC MQEND
04035   741200          SNA                 /MQ SHOULD BE Ø
04036   200034          LAC ACEND
04037   741200          SNA                 /AC=Ø?
04040   200037          LAC SCEND
04041   741200          SNA                 /SC GO TO Ø?
04042   200036          LAC LKEND
04043   741200          SNA                 /LINK STILL Ø?
04044   741000          SKP
04045   104630          JMS LRSERR
04046   100040          JMS SWITCH                  /END SCOPE
04047   004023          LRSTS1+5
                        /
                        .EJECT
```

```
                        /DOES LINK GO TO AC 0 ON AN LRS
                        /0-0, 1-0, 0-1, 1-1
04050   140031  LRSTS2  DZM MQSTRT
04051   140030          DZM ACSTRT
04052   140032          DZM LKSTRT
04053   201342          LAC BIT17
04054   040033          DAC SCSTRT
04055   200032          LAC LKSTRT              /START SCOPE LOOP
04056   740020          RAR
04057   650000          CLQ
04060   200030          LAC ACSTRT              /SET UP COMPLETE
04061   640501          LRS 01
04062   040034          DAC ACEND       /SAVE RESULTS
04063   641002          LACQ
04064   040035          DAC MQEND
04065   641001          LACS
04066   040037          DAC SCEND
04067   750010          GLK
04070   040036          DAC LKEND       /LINK SHOULD NOT CHANGE
04071   540032          SAD LKSTRT
04072   741000          SKP
04073   604105          JMP .+12
04074   200035          LAC MQEND
04075   740200          SZA
04076   604105          JMP .+7
04077   200032          LAC LKSTRT
04100   740020          RAR
04101   200030          LAC ACSTRT
04102   740020          RAR
04103   540034          SAD ACEND
04104   741000          SKP
04105   104630          JMS LRSERR
04106   100040          JMS SWITCH              /END SCOPE
04107   004055          LRSTS2+5
04110   200032          LAC LKSTRT
04111   440032          ISZ LKSTRT
04112   741200          SNA
04113   604055          JMP LRSTS2+5
04114   140032          DZM LKSTRT
04115   201321          LAC BIT0
04116   540030          SAD ACSTRT
04117   604122          JMP .+3
04120   040030          DAC ACSTRT
04121   604055          JMP LRSTS2+5
                /
                        .EJECT
```

```
                        /DOES AC17 GO TO MQ0 ON AN LRS
                        /0-0, 1-0, 0-1, AND 1-1
04122   140032  LRSTS3  DZM LKSTRT          /LINK ALWAYS 0
04123   140031          DZM MQSTRT
04124   140030          DZM ACSTRT
04125   201342          LAC BIT17           /SHIFT OF 1
04126   040033          DAC SCSTRT
04127   744000          CLL
04130   200031          LAC MQSTRT          /SET MQ
04131   652000          LMQ
04132   200030          LAC ACSTRT
04133   640501          LRS 01
04134   040034          DAC ACEND
04135   750010          GLK
04136   040036          DAC LKEND
04137   641001          LACS
04140   040037          DAC SCEND
04141   641002          LACQ
04142   040035          DAC MQEND
04143   200030          LAC ACSTRT          /GENERATE MQ
04144   740020          RAR                 /COMPARE
04145   200031          LAC MQSTRT          /CONSTANT
04146   740020          RAR
04147   540035          SAD MQEND           /AC17 TO MQ0 OK?
04150   741000          SKP
04151   604154          JMP .+3
04152   200034          LAC ACEND
04153   740200          SZA                 /AC GO TO 0?
04154   604160          JMP .+4
04155   200036          LAC LKEND
04156   741200          SNA
04157   741000          SKP
04160   104630          JMS LRSERR
04161   100040          JMS SWITCH
04162   004127          LRSTS3+5
04163   200030          LAC ACSTRT
04164   440030          ISZ ACSTRT
04165   741200          SNA
04166   604127          JMP LRSTS3+5
04167   201321          LAC BIT0
04170   140030          DZM ACSTRT
04171   540031          SAD MQSTRT
04172   604175          JMP .+3
04173   040031          DAC MQSTRT
04174   604127          JMP LRSTS3+5
                        /
                        .EJECT
```

```
                        /DOES AC17 NOT GO TO LINK ON AN LRS
04175   140032  LRSTS4  DZM LKSTRT
04176   140030          DZM ACSTRT
04177   140031          DZM MQSTRT          /MQ ALWAYS ZERO
04200   201342          LAC BIT17
04201   040033          DAC SCSTRT          /SHIFT OF 1
04202   650000          CLQ
04203   200032          LAC LKSTRT
04204   740020          RAR                 /SET LINK INITIAL 0 OR 1
04205   200030          LAC ACSTRT          /AC=1 OR 0
04206   640501          LRS 01
04207   040034          DAC ACEND
04210   641001          LACS
04211   040037          DAC SCEND
04212   641002          LACQ
04213   040035          DAC MQEND
04214   750010          GLK
04215   040036          DAC LKEND
04216   540032          SAD LKSTRT                  /WAS LINK ALTERED
04217   741000          SKP
04220   104630          JMS LRSERR
04221   100040          JMS SWITCH
04222   004202          LRSTS4+5
04223   200032          LAC LKSTRT
04224   440032          ISZ LKSTRT
04225   741200          SNA                 /TESTED L=1?
04226   604202          JMP LRSTS4+5        /NO
04227   140032          DZM LKSTRT
04230   200030          LAC ACSTRT
04231   440030          ISZ ACSTRT
04232   741200          SNA                 /TESTED AC 17=1
04233   604202          JMP LRSTS4+5
04234   100060          JMS SWTCHS
04235   004016          LRSTS1
                /
                        .EJECT
```

```
                        /WILL AC MQ SHIFT A 1 BIT EACH POSITION RIGHT
                        /
04236   140032  LRSTS5  DZM LKSTRT
04237   140031          DZM MQSTRT
04240   144701          DZM MQCOMK
04241   201342          LAC BIT17
04242   040033          DAC SCSTRT
04243   201321          LAC BIT0
04244   040030          DAC ACSTRT
04245   044700          DAC ACCOMK
04246   204700          LAC ACCOMK          /GENERATE COMPARE
04247   744020          CLL!RAR             /CONSTANTS
04250   044700          DAC ACCOMK
04251   204701          LAC MQCOMK
04252   740020          RAR
04253   044701          DAC MQCOMK
04254   200031  LRST5L  LAC MQSTRT
04255   652000          LMQ
04256   744000          CLL
04257   200030          LAC ACSTRT
04260   640501          LRS 01
04261   040034          DAC ACEND
04262   750010          GLK
04263   040036          DAC LKEND
04264   641001          LACS
04265   040037          DAC SCEND
04266   641002          LACQ
04267   040035          DAC MQEND
04270   544701          SAD MQCOMK          /MQ SHIFT OK?
04271   741000          SKP
04272   604276          JMP .+4
04273   204700          LAC ACCOMK
04274   540034          SAD ACEND           /AC SHIFT OK?
04275   741000          SKP
04276   104630          JMS LRSERR
04277   100040          JMS SWITCH
04300   004254          LRST5L
04301   200030          LAC ACSTRT
04302   744020          CLL!RAR
04303   040030          DAC ACSTRT
04304   200031          LAC MQSTRT
04305   740020          RAR
04306   040031          DAC MQSTRT
04307   740400          SNL
04310   604246          JMP LRST5L-6
04311   100060          JMS SWTCHS
04312   004236          LRSTS5
                        /
                        .EJECT
```

```
                        /WILL AC-MQ SHIFT A NO BIT 1 POSITION
                        /RIGHT FROM EACH BIT
04313   201342  LRSTS6  LAC BIT17
04314   040033          DAC SCSTRT
04315   040032          DAC LKSTRT
04316   201343          LAC NBIT0               /377777
04317   040030          DAC ACSTRT
04320   044700          DAC ACCOMK
04321   750001          CLC
04322   040031          DAC MQSTRT
04323   044701          DAC MQCOMK
04324   204700          LAC ACCOMK              /GENERATE NEXT
04325   744002          STL
04326   740020          RAR                     /SET OF
04327   044700          DAC ACCOMK              /AC MQ COMPARE
04330   204701          LAC MQCOMK              /CONSTANTS
04331   740020          RAR
04332   044701          DAC MQCOMK
04333   200031  LRST6L  LAC MQSTRT             /SET UP LRS
04334   652000          LMQ
04335   744002          STL
04336   200030          LAC ACSTRT
04337   640501          LRS 01
04340   040034          DAC ACEND
04341   641001          LACS
04342   040037          DAC SCEND               /FOR TYPEOUTS
04343   750010          GLK
04344   040036          DAC LKEND               /FOR TYPEOUTS
04345   641002          LACQ
04346   040035          DAC MQEND
04347   544701          SAD MQCOMK              /MQ SHIFT OK?
04350   741000          SKP
04351   604355          JMP .+4
04352   204700          LAC ACCOMK
04353   540034          SAD ACEND               /AC SHIFT OK?
04354   741000          SKP
04355   104630          JMS LRSERR
04356   100040          JMS SWITCH
04357   004333          LRST6L
04360   200030          LAC ACSTRT
04361   744002          STL
04362   740020          RAR
04363   040030          DAC ACSTRT
04364   200031          LAC MQSTRT
04365   740020          RAR
04366   040031          DAC MQSTRT
04367   741400          SZL                     /SHIFTED TILL MQ17=0
04370   604324          JMP LRST6L-7
04371   100060          JMS SWTCHS
04372   004313          LRSTS6
                /
                        .EJECT
```

```
                        /WILL AC MQ SHIFT A 1 BIT
                        /RIGHT TO 44 PLACES
                        /
04373   140031  LRSTS7  DZM MQSTRT
04374   140032          DZM LKSTRT
04375   201342          LAC BIT17
04376   040033          DAC SCSTRT
04377   201321          LAC BIT0
04400   040030          DAC ACSTRT
04401   044700          DAC ACCOMK
04402   144701          DZM MQCOMK
04403   204337          LAC LRST6L+4        /LRS 01
04404   044417          DAC LRST7E          /FOR EXECUTE
04405   204700  LRST7L  LAC ACCOMK
04406   744020          CLL!RAR             /GENERATE AC/MQ
04407   044700          DAC ACCOMK                  /COMPARE CONSTANTS
04410   204701          LAC MQCOMK
04411   740020          RAR
04412   044701          DAC MQCOMK
04413   200031          LAC MQSTRT          /SET UP LRS
04414   652000          LMQ
04415   744000          CLL
04416   200030          LAC ACSTRT
04417   640501  LRST7E  LRS 01              /1 TO 44 PLACES
04420   040034          DAC ACEND
04421   750010          GLK
04422   040036          DAC LKEND
04423   641001          LACS
04424   040037          DAC SCEND
04425   641002          LACQ
04426   040035          DAC MQEND
04427   544701          SAD MQCOMK          /MQ SHIFT OK?
04430   741000          SKP
04431   604435          JMP .+4
04432   204700          LAC ACCOMK
04433   540034          SAD ACEND           /AC END OK?
04434   741000          SKP
04435   104652          JMS LRSER1
04436   100040          JMS SWITCH
04437   004413          LRST7E-4
04440   444417          ISZ LRST7E          /INCREMENT SHIFT COUNT
04441   440033          ISZ SCSTRT          /FOR TYPEOUTS
04442   201313          LAC FOUR5
04443   540033          SAD SCSTRT          /SHIFTED 44 PLACES?
04444   741000          SKP                 /YES
04445   604405          JMP LRST7L
04446   100060          JMS SWTCHS
04447   004373          LRSTS7
                        /
                        .EJECT
```

```
                        /WILL AC MQ SHIFT A NO BIT RIGHT
                        /1 TO 44 PLACES
                        /
04450   750001  LRSTS8  CLC
04451   040031          DAC MQSTRT              /MQ START = 1'S
04452   044701          DAC MQCOMK
04453   201343          LAC NBIT0       /AC START BIT 0=0
04454   040030          DAC ACSTRT
04455   044700          DAC ACCOMK
04456   201342          LAC BIT17
04457   040033          DAC SCSTRT
04460   040032          DAC LKSTRT
04461   204337          LAC LRST6L+4            /LRS 01
04462   044476          DAC LRST8E      /FOR EXECUTE
04463   204700  LRST8L  LAC ACCOMK              /GENERATE
04464   744002          STL
04465   740020          RAR                     /NEXT
04466   044700          DAC ACCOMK      /COMPARE CONSTANTS
04467   204701          LAC MQCOMK
04470   740020          RAR
04471   044701          DAC MQCOMK
04472   200031          LAC MQSTRT      /SET UP LRS
04473   652000          LMQ
04474   200030          LAC ACSTRT
04475   744002          STL
04476   640501  LRST8E  LRS 01          /1 TO 44 PLACES
04477   040034          DAC ACEND
04500   750010          GLK
04501   040036          DAC LKEND
04502   641001          LACS
04503   040037          DAC SCEND
04504   641002          LACQ
04505   040035          DAC MQEND
04506   544701          SAD MQCOMK      /MQ SHIFT OK?
04507   741000          SKP
04510   604514          JMP .+4
04511   204700          LAC ACCOMK
04512   540034          SAD ACEND       /AC SHIFT OK?
04513   741000          SKP
04514   104652          JMS LRSER1
04515   100040          JMS SWITCH
04516   004472          LRST8E+4
04517   444476          ISZ LRST8E      /ADVANCE SHIFT
04520   440033          ISZ SCSTRT      /COUNT
04521   201313          LAC FOUR8
04522   540033          SAD SCSTRT      /SHIFTED 44 PLACES
04523   741000          SKP
04524   604463          JMP LRST8L
04525   100060          JMS SWTCHS
04526   004450          LRSTS8
                        /
                        .EJECT
```

```
                        /WILL MQ SHIFT LEFT 1
                        /EVERY COMBINATION OF BITS
04527   140031  LLSSEQ  DZM MQSTRT          /AC AND MQ WILL
04530   140030          DZM ACSTRT          /ALWAYS BE =
04531   201342          LAC BIT17
04532   040033          DAC SCSTRT          /SHIFT IS ALWAYS 1
04533   140032          DZM LKSTRT
04534   200031          LAC MQSTRT
04535   660000          EAE*20000
04536   740010          RAL
04537   044700          DAC ACCOMK                  /AC SHOULD
04540   044701          DAC MQCOMK                  /=MQ
04541   200031          LAC MQSTRT
04542   652000          LMQ
04543   660601          LLSS 01
04544   040034          DAC ACEND
04545   641001          LACS
04546   040037          DAC SCEND
04547   750010          GLK
04550   040036          DAC LKEND
04551   641002          LACQ
04552   040035          DAC MQEND /MQ AND
04553   540034          SAD ACEND /AC SHIFT OK
04554   741000          SKP
04555   103770          JMS LLSSER
04556   100040          JMS SWITCH
04557   004534          LLSSEQ+5
04560   440030          ISZ ACSTRT
04561   740000          NOP
04562   440031          ISZ MQSTRT
04563   604534          JMP LLSSEQ+5
04564   100060          JMS SWTCHS
04565   004527          LLSSEQ
                        /
                        .EJECT
```

```
                        /WILL MQ SHIFT RIGHT 1 EVERY
                        /COMBINATION OF BITS
04566   140030  LRSSEQ  DZM ACSTRT              /AC AND MQ
04567   140031          DZM MQSTRT              /ALWAYS =
04570   201342          LAC BIT17
04571   040033          DAC SCSTRT              /ALWAYS SHIFT OF 1
04572   200030          LAC ACSTRT
04573   501342          AND BIT17
04574   040032          DAC LKSTRT              /LINK = AC 17
04575   740020          RAR         /SO THAT AC WILL = MQ
04576   200031          LAC MQSTRT
04577   740020          RAR
04600   044701          DAC MQCOMK              /AC AND MQ
04601   044700          DAC ACCOMK              /SHOULD BE =
04602   740010          RAL
04603   652000          LMQ
04604   640501          LRS 01
04605   040034          DAC ACEND
04606   641001          LACS
04607   040037          DAC SCEND
04610   750010          GLK
04611   040036          DAC LKEND
04612   641002          LACQ
04613   040035          DAC MQEND
04614   540034          SAD ACEND /AC AND MQ R 1 OK
04615   741000          SKP
04616   104652          JMS LRSER1
04617   100040          JMS SWITCH
04620   004572          LRSSEQ+4
04621   440030          ISZ ACSTRT
04622   740000          NOP
04623   440031          ISZ MQSTRT              /ALL COMBINATIONS
04624   604572          JMP LRSSEQ+4
04625   100060          JMS SWTCHS
04626   004566          LRSSEQ
04627   603267          JMP ENDSHF
                        /
                        .EJECT
```

```
                                /LRS COMMON ERROR TYPEOUT
                                /SHIFT OF 1
04630   624630          LRSERR  JMP .
04631   101134                  JMS ERROR
04632   001563                      TYLRS
04633   740033                  SCSTRT+740000
04634   404630                  LRSERR+400000
04635   001447                      HDR5
04636   500032                  LKSTRT+500000
04637   600030                  ACSTRT+600000
04640   600031                  MQSTRT+600000
04641   001457                      TYPATR
04642   500036                  LKEND+500000
04643   600034                  ACEND+600000
04644   600035                  MQEND+600000
04645   001607                      TYRES
04646   001636                      TYLACS
04647   740037                  SCEND+740000
04650   000000                  0
04651   624630                  JMP* LRSERR
                                /
                                .EJECT
```

```
                        /LRS COMMON ERROR TYPEOUT
                        /SHIFTS OF MORE THAN 1
04652   604652  LRSER1  JMP .
04653   101134          JMS ERROR
04654   001563              TYLRS
04655   740033          SCSTRT+740000
04656   404652          LRSER1+400000
04657   001447              HDR5
04660   500032          LKSTRT+500000
04661   600030          ACSTRT+600000
04662   600031          MQSTRT+600000
04663   001457              TYPATR
04664   001444              SPACE4
04665   604700          ACCOMK+600000
04666   604701          MQCOMK+600000
04667   001461              TYCOR
04670   500036          LKEND+500000
04671   600034          ACEND+600000
04672   600035          MQEND+600000
04673   001463              TYINCO
04674   001636              TYLACS
04675   740037          SCEND+740000
04676   000000          0
04677   624652          JMP* LRSER1
04700   000000  ACCOMK  0
04701   000000  MQCOMK  0
04702   000000  SCCOMK  0
                        /TAPE 5
                        /RANDOM DATA SHIFTS
                        /NORMALIZE TEST
                        /INTERRUPT TEST
                        /
05000                   .LOC 5000
05000   201363          LAC NBIT16
05001   041261          DAC CHARK           /SET PASS K TO -3
                        /
                        /
                        /START RANDOM DATA SHIFTS
                        /LEFT 0 TO 44 PLACES
05002   201350  RANSHF  LAC NBITS
05003   045535          DAC PASSK
05004   105522          JMS RANGEN
05005   040031          DAC MQSTRT
05006   105522          JMS RANGEN          /GENERATE AC START
05007   045540          DAC SHFBUF
05010   206507          LAC (SHFBUF
05011   040010          DAC 10
05012   341342          TAD BIT17
05013   040011          DAC 11
05014   200031          LAC MQSTRT
05015   652000          LMQ
05016   060010          DAC* 10
05017   205540          LAC SHFBUF
05020   040030          DAC ACSTRT
05021   660601          LLSS 01
```

```
05022   760010                  DAC* 10
05023   641002                  LACQ
05024   060010                  DAC* 10
05025   220011          SETLLS  LAC* 11
05026   440011                  ISZ 11
05027   640601                  LLS 01
05030   060010                  DAC* 10
05031   641002                  LACQ
05032   060010                  DAC* 10
05033   206510                  LAC (SHFBUF+111
05034   540010                  SAD 10                  /SHIFTED 44 PLACES?
05035   741000                  SKP
05036   605025                  JMP SETLLS
05037   750010                  GLK
05040   040032                  DAC LKSTRT
05041   140033                  DZM SCSTRT
05042   205537                  LAC KLLSS
05043   045057                  DAC LRANEX
05044   206511                  LAC (SHFBUF-1
05045   040010                  DAC 10
                        /
                                .EJECT
```

```
05046   220010      LRANLP  LAC* 10
05047   044700              DAC ACCOMK
05050   220010              LAC* 10
05051   044701              DAC MQCOMK
05052   200031              LAC MQSTRT
05053   652000              LMQ
05054   200032              LAC LKSTRT
05055   740020              RAR
05056   200030              LAC ACSTRT
05057   660600      LRANEX  LLSS                /0 TO 44 PLACES
05060   040034              DAC ACEND
05061   750010              GLK
05062   040036              DAC LKEND
05063   641001              LACS
05064   040037              DAC SCEND
05065   641002              LACQ
05066   040035              DAC MQEND
05067   544701              SAD MQCOMK          /MQ = PREDICTED?
05070   741000              SKP
05071   605075              JMP .+4
05072   204700              LAC ACCOMK
05073   540034              SAD ACEND           /AC END = PREDICTED?
05074   741000              SKP
05075   103770              JMS LLSSER
05076   100040              JMS SWITCH
05077   005052              LRANLP+4
05100   445057              ISZ LRANEX
05101   440033              ISZ SCSTRT
05102   201313              LAC FOURS           /SHIFTED 44 PLACES?
05103   540033              SAD SCSTRT
05104   741000              SKP
05105   605046              JMP LRANLP
05106   100060              JMS SWTCHS
05107   005010              RANSHF+6
05110   445535      RLSTAY  ISZ PASSK
05111   605004              JMP RANSHF+2
                    /
                            .EJECT
```

```
                        /RANDOM DATA RIGHT 0 TO 44 PLACES
05112   201350  RANRIT  LAC NBITS
05113   045535          DAC PASSK
05114   105522          JMS RANGEN          /GENERATE MQ START
05115   040031          DAC MQSTRT
05116   105522          JMS RANGEN          /GENERATE ACSTRT
05117   040030          DAC ACSTRT
05120   206512          LAC (SHFBUF-1
05121   040010          DAC 10
05122   040011          DAC 11
05123   200030          LAC ACSTRT
05124   060010          DAC* 10
05125   200031          LAC MQSTRT
05126   060010          DAC* 10
05127   652000          LMQ
05130   744000          CLL
05131   220011  SETLRS  LAC* 11
05132   440011          ISZ 11              /GENERATE AC MQ
05133   640501          LRS 01
05134   060010          DAC* 10             /COMPARE CONSTANTS
05135   641002          LACQ
05136   060010          DAC* 10
05137   206513          LAC (SHFBUF+111
05140   540010          SAD 10
05141   741000          SKP
05142   605131          JMP SETLRS
05143   205536          LAC KLRS
05144   045161          DAC RRANEX
05145   140032          DZM LKSTRT
05146   140033          DZM SCSTRT
05147   206514          LAC (SHFBUF-1
05150   040010          DAC 10
05151   220010  RRANLP  LAC* 10
05152   044700          DAC ACCOMK
05153   220010          LAC* 10
05154   044701          DAC MQCOMK
05155   200031          LAC MQSTRT
05156   652000          LMQ
05157   200030          LAC ACSTRT
05160   744000          CLL
05161   640500  RRANEX  LRS 0               /0 TO 44 PLACES
05162   040034          DAC ACEND
05163   750010          GLK
05164   040036          DAC LKEND
05165   641001          LACS
05166   040037          DAC SCEND
05167   641002          LACQ
05170   040035          DAC MQEND
05171   544701          SAD MQCOMK
05172   741000          SKP
05173   605177          JMP .+4
05174   204700          LAC ACCOMK
05175   540034          SAD ACEND
05176   741000          SKP
05177   104652          JMS (RSEB1
```

```
05200   100040          JMS SWITCH
                        .EJECT
```

```
05201   205155                  RRANEX-4
05202   445161                  ISZ RRANEX
05203   440033                  ISZ SCSTRT
05204   201313                  LAC FOURS
05205   540033                  SAD SCSTRT
05206   741000                  SKP
05207   605151                  JMP RRANIP
05210   100060                  JMS SWTCHS
05211   005120                  RANRIT+6
05212   445535          RRSTAY  ISZ PASSW
05213   605114                  JMP RANRIT+2
                        /
                                .EJECT
```

```
                        /RANDOM DATA SEQUENCED
                        /
05214   201350  RANSEQ  LAC NBIT5
05215   045535          DAC PASSK
05216   105522          JMS RANGEN
05217   040030          DAC ACSTRT
05220   661000          EAE!21000           /GET AC SIGN CLR AC
05221   750010          GLK
05222   045446          DAC SVSIGN
05223   742020          RTR
05224   045447          DAC SVSIGN+1
05225   105522          JMS RANGEN
05226   501364          AND NBIT17          /MAKE MQ17=AC0
05227   345446          TAD SVSIGN
05230   040031          DAC MQSTRT
05231   201364          LAC NBIT17
05232   045450          DAC SVMASK
05233   201343          LAC NBIT0
05234   045451          DAC SVMASK+1
05235   140033          DZM SCSTRT
05236   140032          DZM LKSTRT
05237   200031  RANSQ0  LAC MQSTRT          /SEQUENCE 0
05240   652000          LMQ
05241   744000          CLL
05242   200030          LAC ACSTRT
05243   660601          LLSS 1
05244   640502          LRS 2
05245   660602          LLSS 2
05246   640501          LRS 1
05247   105452          JMS SEQCOM
05250   100040          JMS SWITCH
05251   005237          RANSQ0
05252   105417          JMS NXTSEQ
                        /
                        .EJECT
```

```
                        /SEQUENCE 1
                        /RIGHT 2, L4, R4, L2
                        /
05253   200031  RANSQ1  LAC MQSTRT          /SEQUENCE 1 R2, L4, R4, L2
05254   744000          CLL
05255   652000          LMQ                 /SET UP
05256   200030          LAC ACSTRT
05257   660502          LRSS*2
05260   660604          LLSS*4
05261   640504          LRS*4
05262   660602          LLSS*2
05263   105452          JMS SEQCOM          /COMPARE RESULTS
05264   100040          JMS SWITCH
05265   005253              RANSQ1
05266   105417          JMS NXTSEQ
                        /LEFT 3, RIGHT 6, LEFT 6, RIGHT 3
                        /SEQUENCE 2
05267   200031  RANSQ2  LAC MQSTRT
05270   652000          LMQ
05271   744000          CLL
05272   200030          LAC ACSTRT
05273   660603          LLSS*3
05274   640506          LRS*6
05275   660606          LLSS*6
05276   640503          LRS*3
05277   105452          JMS SEQCOM
05300   100040          JMS SWITCH
05301   005267              RANSQ2
05302   105417          JMS NXTSEQ
                        /
                        .EJECT
```

```
                        /SEQUENCE 3
                        /RIGHT 4, LEFT 8, RIGHT 8, LEFT 4
                        /
05303   200031  RANSQ3  LAC MQSTRT
05304   744000          CLL
05305   652000          LMQ
05306   200030          LAC ACSTRT
05307   660504          LRSS*4
05310   660610          LLSS*10
05311   640510          LRS*10
05312   660604          LLSS*4
05313   105452          JMS SEQCOM
05314   100040          JMS SWITCH
05315   005303          RANSQ3
05316   105417          JMS NXTSEQ
                        /
                        /
                        /SEQUENCE 4  LEFT 5, RIGHT 10, LEFT 10, RIGHT 5
                        /
05317   200031  RANSQ4  LAC MQSTRT
05320   744000          CLL
05321   652000          LMQ
05322   200030          LAC ACSTRT
05323   660605          LLSS*5
05324   640512          LRS*12
05325   660612          LLSS*12
05326   640505          LRS*5
05327   105452          JMS SEQCOM
05330   100040          JMS SWITCH
05331   005317          RANSQ4
05332   105417          JMS NXTSEQ
                        /
                        .EJECT
```

```
                        /SEQUENCE 5  RIGHT 6, LEFT 12, RIGHT 12, LEFT 6
                        /
05333   200031  RANSQ5  LAC MQSTRT
05334   652000          LMQ
05335   744000          CLL
05336   200030          LAC ACSTRT
05337   660506          LRSS*6
05340   660614          LLSS*14
05341   640514          LRS*14
05342   660606          LLSS*6
05343   105432          JMS SEQCOM
05344   100040          JMS SWITCH
05345   005333              RANSQ5
05346   105417          JMS NXTSEQ
                        /
                        /
                        /SEQUENCE 6  LEFT 7 RIGHT 14, LEFT 14, RIGHT 7
                        /
05347   200031  RANSQ6  LAC MQSTRT
05350   652000          LMQ
05351   744000          CLL
05352   200030          LAC ACSTRT
05353   660607          LLSS*7
05354   640516          LRS*16
05355   660616          LLSS*16
05356   640507          LRS*7
05357   105432          JMS SEQCOM
05360   100040          JMS SWITCH
05361   005347              RANSQ6
05362   105417          JMS NXTSEQ
                        /
                        .EJECT
```

```
                        /SEQUENCE 7  RIGHT 8, LEFT 16, RIGHT 16, LEFT 8
                        /
05363   200031  RANSQ7  LAC MQSTRT
05364   652000          LMQ
05365   200030          LAC ACSTRT
05366   744000          CLL
05367   660510          LRSS*10
05370   660620          LLSS*20
05371   640520          LRS*20
05372   660610          LLSS*10
05373   105452          JMS SEQCOM
05374   100040          JMS SWITCH
05375   005363              RANSQ7
05376   105417          JMS NXTSEQ
                        /
                        /
                        /SEQUENCE 8  LEFT 9, RIGHT 18, LEFT 18, RIGHT 9
                        /
05377   200031  RANSQ8  LAC MQSTRT
05400   652000          LMQ
05401   200030          LAC ACSTRT
05402   744000          CLL
05403   660611          LLSS*11
05404   640522          LRS*22
05405   660622          LLSS*22
05406   640511          LRS*11
05407   105452          JMS SEQCOM
05410   100040          JMS SWITCH
05411   005377              RANSQ8
05412   445535          ISE PASSK
05413   605216          JMP RANSEQ+2
05414   100060          JMS SWTCHS
05415   005214              RANSEQ
05416   605652          JMP NRMLEE
                        /
                        .EJECT
```

```
                        /SET AC SIGN INTO NEXT AC
                        /AND MQ BITS
                        /
05417   605417  NXTSEQ  JMP .
05420   205446          LAC SVSIGN
05421   744010          CLL!RAL
05422   045446          DAC SVSIGN          /TO FILL MQ
05423   205447          LAC SVSIGN+1
05424   744020          CLL!RAR
05425   045447          DAC SVSIGN+1
05426   205450          LAC SVMASK
05427   744002          STL
05430   740010          RAL
05431   045450          DAC SVMASK
05432   500031          AND MQSTRT          /CLR MQ BIT
05433   345446          TAD SVSIGN          /MAKE MQ = AC 0
05434   040031          DAC MQSTRT
05435   205451          LAC SVMASK+1
05436   744002          STL
05437   740020          RAR
05440   045451          DAC SVMASK+1
05441   500030          AND ACSTRT          /CLR AC BIT
05442   345447          TAD SVSIGN+1        /MAKE ACX = AC 0
05443   040030          DAC ACSTRT
05444   440033          ISZ SCSTRT          /INDICATE NEXT SEQUENCE
05445   625417          JMP* NXTSEQ
05446   000000  SVSIGN  0
05447   000000          0
05450   000000  SVMASK  0
05451   000000          0
                        /
                        .EJECT
```

```
                        /RANDOM DATA SEQUENCED
                        /COMMON COMPARE AND ERROR TYPE
                        /
05452   605452  SEQCOM  JMP .
05453   240034          DAC ACEND
05454   750010          GLK
05455   040036          DAC LKEND
05456   641001          LACS
05457   040037          DAC SCEND
05460   641002          LACQ
05461   040035          DAC MQEND
05462   540031          SAD MQSTRT          /MQ SAME AS START
05463   741000          SKP
05464   605467          JMP .+3             /ERROR MQ
05465   200037          LAC SCEND
05466   740200          SZA
05467   605473          JMP .+4             /ERROR SC
05470   200030          LAC ACSTRT
05471   540034          SAD ACEND
05472   741000          SKP
05473   605500          JMP .+5             /ERROR AC
05474   661000          EAE!21000           /GET AC SIGN CLR AC
05475   750010          GLK
05476   540036          SAD LKEND           /LINK END = AC SIGN?
05477   625452          JMP* SEQCOM         /ALL OK - EXIT
05500   101134          JMS ERROR
05501   001577              TYRDSQ
05502   001442              SPACE3
05503   740033          SCSTRT+740000
05504   405452          SEQCOM+400000
05505   001447              HDR5
05506   500032          LKSTRT+500000
05507   600030          ACSTRT+600000
05510   600031          MQSTRT+600000
05511   001512              TYSTRT
05512   500036          LKEND+500000
05513   600034          ACEND+600000
05514   600035          MQEND+600000
05515   001607              TYRES
05516   001636              TYLACS
05517   740037          SCEND+740000
05520   000000          0
05521   625452          JMP* SEQCOM         /ERROR EXIT
                        /
                        .EJECT
```

```
                        /RANDOM NUMBER GENERATOR
                        /18 BIT
05522   605522  RANGEN  JMP .
05523   205533          LAC RANNO
05524   744020          CLL!RAR
05525   741400          SZL
05526   241321          XOR BIT0
05527   245534          XOR RANNO+1
05530   305534          ADD RANNO+1
05531   045533          DAC RANNO
05532   625522          JMP* RANGEN
05533   736425  RANNO   736425
05534   335671          335671
05535   000000  PASSK   0
05536   640500  KLRS    LRS
05537   660600  KLLSS   LLSS
                        /
                        /
05540   000000  SHFBUF  0
05652                   .LOC SHFBUF+112
                        /
                        /
                        /NORMALIZE TEST
                        /DOES NORMS GET AC 0 = 0 TO L
                        /
05652   140031  NRMLZE  DZM MQSTRT
05653   140033          DZM SCSTRT
05654   201322          LAC BIT1
05655   040030          DAC ACSTRT
05656   140032          DZM LKSTRT
05657   200032          LAC LKSTRT          /START SCOPE LOOP
05660   740020          RAR
05661   650000          CLQ
05662   200030          LAC ACSTRT
05663   660400          NORMS-44            /SC = 0
05664   641002          LACQ
05665   040035          DAC MQEND           /SAVE RESULTS
05666   641001          LACS
05667   040037          DAC SCEND
05670   750010          GLK
05671   144702          DZM SCCOMK
05672   040036          DAC LKEND
05673   740200          SZA                 /AC SIGN IS 0
05674   106242          JMS NORMSE
05675   100040          JMS SWITCH          /END SCOPE LOOP
05676   005657          NRMLZE+5
05677   200032          LAC LKSTRT
05700   440032          ISZ LKSTRT
05701   741200          SNA
05702   605657          JMP NRMLZE+5
                        /
                        .EJECT
```

```
                        /DOES NORMS GET AC0=1 TO L
05703   750001  NRML=1  CLC
05704   040031          DAC MQSTRT
05705   140033          DZM SCSTRT
05706   140032          DZM LKSTRT
05707   750001          CLC
05710   040030          DAC ACSTRT
05711   200032          LAC LKSTRT          /START SCOPE LOOP
05712   740020          RAR
05713   200030          LAC ACSTRT
05714   650004          CLQ*4               /SET MQ = 1'S
05715   660400          NORMS-44
05716   040034          DAC ACEND
05717   641002          LACQ
05720   040035          DAC MQEND
05721   641001          LACS
05722   040037          DAC SCEND
05723   750010          GLK
05724   144702          DZM SCCOMK
05725   040036          DAC LKEND
05726   741200          SNA
05727   106242          JMS NORMSE
05730   100040          JMS SWITCH          /END SCOPE LOOP
05731   005711          NRML=1+6
05732   200032          LAC LKSTRT
05733   440032          ISZ LKSTRT
05734   741200          SNA
05735   005711          JMP NRML=1+6
                        /
                        .EJECT
```

```
                        /WILL NORM STOP SHIFT WITH
                        /AC 0 AND AC 1 UNEQUAL?  01, 10
                        /
05736   140031  NRML22  DZM MQSTRT
05737   140032          DZM LKSTRT
05740   201274          LAC SEVSEV
05741   044702          DAC SCCOMK
05742   201322          LAC BIT1
05743   040030          DAC ACSTRT
05744   201342          LAC BIT17
05745   040033          DAC SCSTRT
05746   200031          LAC MQSTRT          /START SCOPE LOOP
05747   652000          LMQ
05750   744000          CLL
05751   200030          LAC ACSTRT          /SET UP COMPLETE
05752   640401          NORM=43                     /SC = 1
05753   040034          DAC ACEND
05754   641002          LACQ
05755   040035          DAC MQEND           /SAVE RESULTS
05756   750010          GLK
05757   040036          DAC LKEND
05760   641001          LACS
05761   040037          DAC SCEND
05762   541274          SAD SEVSEV          /SC = -19
05763   741000          SKP
05764   106214          JMS NORMER
05765   100040          JMS SWITCH          /END SCOPE LOOP
05766   005746          NRML22+10
05767   200030          LAC ACSTRT
05770   740001          CMA
05771   040030          DAC ACSTRT
05772   200031          LAC MQSTRT
05773   740001          CMA
05774   040031          DAC MQSTRT
05775   740200          SZA
05776   605746          JMP NRML22+10
                        /
                        .EJECT
```

```
                        /DOES NORM NOT STOP SHIFT
                        /ON AC 0 = AC1  00, 11,
                        /
05777   140031  NRML23  DZM MQSTRT
06000   140032          DZM LKSTRT
06001   201341          LAC BIT16           / NORMALIZE SC = 2
06002   040033          DAC SCSTRT
06003   201274          LAC SEVSEV
06004   044702          DAC SCCOMK
06005   201323          LAC BIT2
06006   040030          DAC ACSTRT
06007   200031          LAC MQSTRT          /START SCOPE LOOP
06010   652000          LMQ
06011   744000          CLL
06012   200030          LAC ACSTRT          /COMPLETE SET UP
06013   660402          NORMS-42            /SC = 2
06014   040034          DAC ACEND
06015   641001          LACS
06016   040037          DAC SCEND           /SAVE RESULTS
06017   750010          GLK
06020   040036          DAC LKEND
06021   641002          LACQ
06022   040035          DAC MQEND
06023   741180          SPA
06024   740001          CMA                 /MQ = ALL 0'S OR ALL 1'S
06025   740200          SZA
06026   606034          JMP .+6             /ERROR IN MQ
06027   200034          LAC ACEND
06030   741100          SPA                 /AC NEGATIVE?
06031   740001          CMA                 /MAKE POSITIVE
06032   541322          SAD BIT1            /AC NORMALIZE CORRECT?
06033   741000          SKP
06034   606040          JMP .+4             /AC IN ERROR
06035   200037          LAC SCEND
06036   541274          SAD SEVSEV          /SC = -1?
06037   741000          SKP
06040   106242          JMS NORMSE
06041   100040          JMS SWITCH          /END SCOPE LOOP
06042   005660          NRMLZE+6
06043   200030          LAC ACSTRT
06044   740001          CMA
06045   040030          DAC ACSTRT
06046   200031          LAC MQSTRT
06047   740001          CMA
06050   040031          DAC MQSTRT
06051   740200          SZA
06052   606007          JMP NRMLZ3+10
                        /
                        .EJECT
```

```
                        /WILL NORMALIZE NORMALIZE A POSITIVE
                        /NUMBER WITH A 1 FROM AC BIT 1 TO AC 17 WITH SC=44 AT START
                        /AND A NEGATIVE NUMBER  WITH A0 IN AC BIT 1
                        /TO AC1
                        /AC = MQ AT NORMS START.
                        /AC & MQ SHOULD EQUAL
                        /200000 OR 577777 AT END.
06053   140032  NRMLZ4  DZM LKSTRT
06054   201314          LAC FOUR4
06055   040033          DAC SCSTRT
06056   201322          LAC BIT1
06057   040030          DAC ACSTRT
06060   040031          DAC MQSTRT
06061   201316  NR4A    LAC THREE4
06062   044702          DAC SCCOMK          /TO COMPARE SC
06063   200031  NR4B    LAC MQSTRT          /SCOPE LOOP START
06064   652000          LMQ
06065   744000          CLL
06066   200030          LAC ACSTRT          /SET UP COMPLETE
06067   660444          NORMS               /SC = 44
06070   040034          DAC ACEND
06071   641002          LACQ
06072   040035          DAC MQEND /SAVE RESULTS
06073   750010          GLK
06074   040036          DAC LKEND
06075   641001          LACS
06076   040037          DAC SCEND
06077   544702          SAD SCCOMK
06100   741000          SKP
06101   606115          JMP NR4C  /SC, ERROR
06102   200034          LAC ACEND
06103   741100          SPA
06104   740001          CMA
06105   541322          SAD BIT1  /AC SHOULD BE = 20000.
06106   741000          SKP
06107   606115          JMP NR4C
06110   200035          LAC MQEND
06111   741100          SPA
06112   740001          CMA
06113   541322          SAD BIT1  /MQ SHOULD BE = 200000
06114   741000          SKP
06115   106242  NR4C    JMS NORMSE
06116   100040          JMS SWITCH
06117   006063          NR4B
06120   200031          LAC MQSTRT
06121   652000          LMQ
06122   744000          CLL
06123   200030          LAC ACSTRT          /SHIFT AC & MQ
06124   444702          ISZ SCCOMK          /WHEN AC NOT EQUAL MQ
06125   660501          LRSS+1
06126   040030          DAC ACSTRT          /CHANGE SIGNS
06127   641002          LACQ
06130   040031          DAC MQSTRT
06131   540030          SAD ACSTRT          /AC AND MQ STILL EQUAL
06132   606063          JMP NR4B  /DO, AGAIN.
```

```
06133   740100              SMA                     /DONE ALL NEGATIVES YET,
06134   606141              JMP NRML#5              /YES, DO NEXT TEST.
06135   201344              LAC NBIT1 /2ND SERIES, POSITIVES DONE, DO NEGATIVES.
06136   040030              DAC ACSTRT              /NEGATIVE NUMBERS
06137   040031              DAC MQSTRT
06140   606061              JMP NR4A
                    /
                            .EJECT
```

```
                        /WILL A COMPLEMENT BIT PATTERN NORMALIZE
                        /MQ = 252525 AND 525252 AC = 0'S OR 1'S
                        /
06141   140030          NRMLZ5  DZM ACSTRT
06142   201320                  LAC COMBIT              /252525 PATTERN
06143   040031                  DAC MQSTRT
06144   201314                  LAC FOUR4
06145   040033                  DAC SCSTRT
06146   140032                  DZM LKSTRT
06147   200031                  LAC MQSTRT              /SCOPE LOOP START
06150   652000                  LMQ
06151   744000                  CLL
06152   200030                  LAC ACSTRT
06153   660444                  NORMS
06154   040034                  DAC ACEND
06155   641001                  LACS
06156   040037                  DAC SCEND
06157   750010                  GLK
06160   040036                  DAC LKEND
06161   641002                  LACQ
06162   040035                  DAC MQEND
06163   741100                  SPA
06164   740001                  CMA
06165   740200                  SZA
06166   606172                  JMP .+4
06167   200034                  LAC ACEND               /ACEND SHOULD
06170   540031                  SAD MQSTRT              /= MQSTRT
06171   741000                  SKP
06172   606177                  JMP .+5                 /AC ERROR
06173   201317                  LAC FIVE6
06174   044702                  DAC SCCOMK
06175   540037                  SAD SCEND /SC INDICATE SHIFT 18
06176   741000                  SKP
06177   106242                  JMS NORMSE
06200   100040                  JMS SWITCH              /END SCOPE LOOP
06201   006147                  NRMLZ5+6
06202   750001                  CLC
06203   040030                  DAC ACSTRT
06204   200031                  LAC MQSTRT
06205   740001                  CMA
06206   040031                  DAC MQSTRT
06207   741100                  SPA
06210   606147                  JMP NRMLZ5+6
06211   100060                  JMS SWTCHS              /TEST REPEAT SEQUENCE
06212   005652                      NRMLZE
06213   606270                  JMP INTEST              /GO TO INTERRUPT TEST
                        /
                                .EJECT
```

```
                        /NORMALIZE ERROR TYPEOUTS
                        /
06214   606214  NORMER  JMP .
06215   101134          JMS ERROR
06216   001646              TYNORM
06217   740033          SCSTRT+740000
06220   406214          NORMER+400000       /ERROR ADDRESS
06221   001447              HDR5
06222   500032          LKSTRT+500000
06223   600030          ACSTRT+600000
06224   600031          MQSTRT+600000
06225   001457              TYPATR
06226   500036          LKEND+500000
06227   600034          ACEND+600000
06230   600035          MQEND+600000
06231   001607              TYRES
06232   001636              TYLACS
06233   744702          SCCOMK+740000
06234   001461              TYCOR
06235   001636              TYLACS
06236   740037          SCEND+740000
06237   001607              TYRES
06240   000000          0
06241   626214          JMP* NORMER
                        /
                        /
                        /NORMALIZE SIGNED ERROR TYPEOUTS
                        /
06242   606242  NORMSE  JMP .
06243   101134          JMS ERROR
06244   001466              TYNRMS
06245   740033          SCSTRT+740000
06246   406242          NORMSE+400000
06247   001447              HDR5
06250   500032          LKSTRT+500000
06251   600030          ACSTRT+600000
06252   600031          MQSTRT+600000
06253   001457              TYPATR
06254   500036          LKEND+500000
06255   600034          ACEND+600000
06256   600035          MQEND+600000
06257   001607              TYRES
06260   001636              TYLACS
06261   744702          SCCOMK+740000
06262   001461              TYCOR
06263   001636              TYLACS
06264   740037          SCEND+740000
06265   001607              TYRES
06266   000000          0
06267   626242          JMP* NORMSE
                        /
                        .EJECT
```

```
                        /TEST PROGRAM INTERRUPT
                        /AFTER EAE OPERATIONS
                        /
06270   700401  INTEST  TSF                     /PRINTER FLAG?
06271   741000          SKP                     /NO
06272   606276          JMP .+4
06273   760000          LAW 0
06274   700406          TLS                     /TYPE NULL
06275   700401          TSF                     /WAIT PRINTER FLAG
06276   606275          JMP .-1
06277   206515          LAC (JMP INTS1
06300   040001          DAC 1                   /LOAD INT JMP
06301   700042          ION
06302   640000          EAE
06303   740000          NOP
06304   700002          IOF                     /SHOULD NOT GET HERE
06305   101134          JMS ERROR
06306   001472              TYINTE
06307   001514              TYNOP
06310   406302          400000+.-6
06311   000000          0
06312   700401  INTS1   TSF                     /WAIT IN CASE
06313   606312          JMP .-1                 /OF ERROR
06314   100040          JMS SWITCH
06315   006301          INTEST+11
06316   201342          LAC BIT17
06317   040031          DAC MQSTRT
06320   140030          DZM ACSTRT
06321   140032          DZM LKSTRT
06322   201315          LAC FOUR3
06323   040033          DAC SCSTRT
06324   206516          LAC (JMP INTS2
06325   040001          DAC 1
06326   200031  INTS2L  LAC MQSTRT              /PREPARE FOR LLS
06327   652000          LMQ
06330   754000          CLA!CLL
06331   700042          ION
06332   640643          LLS+43                  /EXECUTE
06333   740000          NOP
06334   700002          IOF                     /SHOULD NOT GET HERE
06335   101134          JMS ERROR
06336   001472              TYINTE
06337   001553              TYLLS
06340   740033          SCSTRT+740000
06341   406332          .-7+400000
06342   000000          0
06343   606401          JMP INTS2E
                        /
                        .EJECT
```

```
06344   040034      INTS2   DAC ACEND           /SAVE RESULTS
06345   641001              LACS
06346   040037              DAC SCEND
06347   641002              LACQ
06350   040035              DAC MQEND
06351   740200              SZA                 /MQ SHIFT OK?
06352   606356              JMP .+4
06353   200034              LAC ACEND
06354   541321              SAD BIT0            /AC SHIFT OK?
06355   741000              SKP
06356   606362              JMP .+4
06357   200037              LAC SCEND           /SC GO TO 0?
06360   741200              SNA
06361   606401              JMP INTS2E
06362   101134              JMS ERROR
06363   001502                  INDAT
06364   001553                  TYLLS
06365   740033              SCSTRT+740000
06366   001447                  HDR5
06367   500032              LKSTRT+500000
06370   600030              ACSTRT+600000
06371   600031              MQSTRT+600000
06372   001457                  TYPATR
06373   500032              LKSTRT+500000
06374   600034              ACEND+600000
06375   600035              MQEND+600000
06376   001636                  TYLACS
06377   740037              SCEND+740000
06400   000000              0
06401   700401      INTS2E  TSF                 /WAIT IN CASE OF
06402   606401              JMP .-1             /ERROR TYPEOUT
06403   100040              JMS SWITCH
06404   006326                  INTS2L
                            .EJECT
```

```
06405   700401                  TSF                             /TESTING INTERUPT BEING DELAYED
06406   741000                  SKP                             /TWO INSTRUCTIONS AFTER
06407   606413                  JMP .+4                         /NORMALIZE IS DONE.
06410   760000                  LAW 0
06411   700406                  TLS
06412   700401                  TSF                             /HAVE FLAG ON TO CAUSE INTERUPT.
06413   606412                  JMP .-1
06414   206517          QNRM    LAC (JMP QNRM2                  /SET JUMP FOR INTERUPT.
06415   040001                  DAC 1
06416   201322                  LAC BIT1
06417   700042                  ION
06420   640444          QNRM1   NORM                            /DO INITIALIZE
06421   440001                  ISZ 1                           /ISZ SHOULD BE DONE BEFORE INTERUPT.
06422   440000                  ISZ 0
06423   700002                  IOF                             /SHOULD NOT COME HERE.
06424   101134                  JMS ERROR               /NO INTERUPT OCCURRED.
06425   001472                      TYINTE
06426   001646                      TYNORM
06427   406420                  QNRM1+400000
06430   000000                  0
06431   606440                  JMP QNRM3
06432   741000          QNRM2   SKP                             /IF INTERUPT HAPPENS BEFORE ISZ, DO SKP.
06433   606440                  JMP QNRM3               /OK, INTERUPT DELAYED ONE INSTRUCTION.
06434   101134                  JMS ERROR               /NO DELAY OF INTERUPT AFTER NORMALIZE.
06435   001612                      TYOINT
06436   001646                      TYNORM
06437   406420                  QNRM1+400000
06440   700401          QNRM3   TSF                             /WAIT FOR TYPING TO END.
06441   606440                  JMP .-1
06442   100040                  JMS SWITCH                      /CHECK LOOP
06443   006414                  QNRM
06444   100060                  JMS SWTCHS
06445   006270                  INTEST
06446   750004                  LAS
06447   501327                  AND BIT6
06450   741200                  SNA                     /TYPE AT END SET?
06451   606455                  JMP .+4
06452   760052                  LAW 52
06453   101716                  TY1
06454   441261                  ISZ CHARK
06455   606461                  JMP .+4
06456   101240                  JMS CRLF
06457   201363                  LAC NBIT16
06460   041261                  DAC CHARK
06461   750004                  LAS
06462   501326                  AND BIT5                /CYCLE ALL TESTS
06463   741200                  SNA                     /=1?
06464   605002                  JMP RANSHF              /NO, STAY IN RANDOMS
06465   600225                  JMP NOPAC               /START SET UP TEST
        000000                  .END
06471   007777  *L
06472   207207  *L
06473   777700  *L
06474   151200  *L
06475   000077  *L
```

```
06476   000040   *L
06477   000200   *L
06500   000300   *L
06501   000240   *L
06502   000007   *L
06503   000260   *L
06504   000215   *L
06505   000212   *L
06506   003244   *L
06507   005540   *L
06510   005651   *L
06511   005537   *L
06512   005537   *L
06513   005651   *L
06514   005537   *L
06515   606312   *L
06516   606344   *L
06517   606432   *L
```

NO ERROR LINES

```
ACCOMK   04700
ACEND    00034
ACLMQ    00704
ACLMQE   00732
ACONEQ   01017
ACORMQ   00641
ACSTRT   00030
ALSERR   03175
ALSLNK   02663
ALSMQT   02760
ALSZER   02604
ALS01    02626
BIT0     01321
BIT1     01322
BIT10    01333
BIT11    01334
BIT12    01335
BIT13    01336
BIT14    01337
BIT15    01340
BIT16    01341
BIT17    01342
BIT2     01323
BIT3     01324
BIT4     01325
BIT5     01326
BIT6     01327
BIT7     01330
BIT8     01331
BIT9     01332
CHARK    01261
CLOF    700004
CLON    700044
CLSF    700001
COMBIT   01320
COMMA    03307
COMPMQ   00763
CRCODE   01465
CRLF     01240
DECONT   02047
EAEABS   01045
EAECAC   00246
EAECLQ   00264
EAERMQ   00202
EAESLK   00533
EEM     707702
ENDSHF   03267
ERCONT   01163
ERLOOP   01145
ERROR    01134
FIVE6    01317
FOUR1    01304
FOUR3    01315
FOUR4    01314
FOUR5    01313
```

```
HDR1      01366
HDR2      01375
HDR3      01411
HDR4      01427
HDR5      01447
HSALS     03114
HSALSE    03134
HSALSL    03127
INDAT     01502
INTEST    06270
INTS1     06312
INTS2     06344
INTS2E    06401
INTS2L    06326
KALL7     01365
KALS01    02732
KLLSS     05537
KLLSS1    03767
KLRS      05536
KRB      700312
KSF      700301
K18       01300
LEM      707704
LKEND     00036
LKSTRT    00032
LLSACT    03406
LLSERR    03745
LLSSEQ    04527
LLSSER    03770
LLSSEX    03617
LLSSL1    03606
LLSSL2    03671
LLSSX2    03703
LLSTS1    03310
LLSTS2    03342
LLSTS3    03465
LLSTS4    03526
LLSTS5    03575
LLSTS6    03656
LNKALS    02721
LRANEX    05057
LRANLP    05046
LRSERR    04630
LRSER1    04652
LRSSEQ    04566
LRSTS1    04016
LRSTS2    04050
LRSTS3    04122
LRSTS4    04175
LRSTS5    04236
LRSTS6    04313
LRSTS7    04373
LRSTS8    04450
LRST5L    04254
LRST6L    04333
```

```
LRST7E   04417
LRST7L   04405
LRST8E   04476
LRST8L   04463
MIN5     01257
MIN6     01260
MQCOMK   04701
MQEND    00035
MQSTRT   00031
MQ1TAC   00313
NBIT0    01343
NBIT1    01344
NBIT10   01355
NBIT11   01356
NBIT12   01357
NBIT13   01360
NBIT14   01361
NBIT15   01362
NBIT16   01363
NBIT17   01364
NBIT2    01345
NBIT3    01346
NBIT4    01347
NBIT5    01350
NBIT6    01351
NBIT7    01352
NBIT8    01353
NBIT9    01354
NDSETU   01112
NOPAC    00225
NOPAC1   00341
NOPLK1   00577
NOPLNK   00443
NOPMQ    00366
NOPMQ1   00414
NOPSC    02225
NOPSC1   02555
NORMER   06214
NORMSE   06242
NRMLZE   05652
NRMLZ1   05703
NRMLZ2   05736
NRMLZ3   05777
NRMLZ4   06053
NRMLZ5   06141
NR4A     06061
NR4B     06063
NR4C     06115
NSNERR   03077
NUCT     06466
NUVAL    06467
NXTSEQ   05417
ONESEV   01307
OPS     101762
OPT     101762
```

```
OTY      02107
PASSK    05535
PCF     700202
PSA     700204
PSB     700244
PSF     700201
QNRM     06414
QNRM1    06420
QNRM2    06432
QNRM3    06440
QONEAC   00510
RANGEN   05522
RANNO    05533
RANRIT   05112
RANSEQ   05214
RANSHF   05002
RANSQ0   05237
RANSQ1   05253
RANSQ2   05267
RANSQ3   05303
RANSQ4   05317
RANSQ5   05333
RANSQ6   05347
RANSQ7   05363
RANSQ8   05377
RCF     700102
RESULT   03245
RLSTAY   05110
RL6      02115
RRANEX   05161
RRANLP   05151
RRB     700112
RRSTAY   05212
RSA     700104
RSB     700144
RSF     700101
SALSRP   03244
SAVERR   01276
SCCOMK   04702
SCEND    00037
SCERR    02520
SCSTRT   00033
SCT000   02359
SCT001   02373
SCT003   02411
SCT007   02427
SCT017   02445
SCT037   02463
SCT040   02337
SCT060   02321
SCT070   02303
SCT074   02265
SCT076   02246
SCT077   02501
SCTST1   02200
```

```
SECCOM   25452
SETLLS   05025
SETLRS   05131
SETUP    00200
SEVEN    01271
SEVFIV   01311
SEVNTY   01302
SEVN4    01303
SEVONE   01310
SEVSEV   01274
SEVSIX   01275
SGNSHF   03042
SHFBUF   05540
SIMALS   03216
SIXONE   01306
SIXTY    01301
SPAC     02022
SPACE2   02075
SPACE3   01442
SPACE4   01444
SVCHAR   01263
SVER     01277
SVMASK   05450
SVSIGN   05446
SWITCH   00040
SWTCHS   00060
TCALL    01215
TCF      700402
TCR      102101
TCTWO    01224
TDIGIT   102070
TEMY1    06470
THREE    01312
THREE4   01316
THREE7   01305
TIN      102101
TLS      700406
TOCTAL   02026
TOCT1    02035
TSF      700401
TSP      102022
TSR      101673
TWORD    102026
TWO40    01272
TWO60    01273
TYABS    01626
TYALS    01667
TYALSQ   01662
TYCLA    01517
TYCLQ    01523
TYCMQ    01527
TYCOR    01461
TYCRLF   02101
TYCSC    01632
TYDELE   01244
```

```
TYINCO   01463
TYINTE   01472
TYLACQ   01547
TYLACS   01636
TYLLS    01553
TYLLSS   01557
TYLMQ    01622
TYLRS    01563
TYLRSS   01573
TYNOP    01514
TYNORM   01646
TYNRMS   01466
TYPATR   01457
TYPCHR   01716
TYPCON   01762
TYPCO3   02001
TYPECN   01167
TYPLS1   01656
TYPOCT   02070
TYPSAV   01755
TYPTSR   01673
TYPTYT   02013
TYQINT   01612
TYRDSQ   01577
TYRES    01607
TYRMQ    01533
TYSCER   01642
TYSIMR   01567
TYSLK    01537
TYSMQ    01543
TYSSC    01652
TYSTRT   01512
TYT     102013
TY1     101716
.EOT     00000
```

```
.EOT    00000
ACSTRT  00030
MQSTRT  00031
LKSTRT  00032
SCSTRT  00033
ACEND   00034
MQEND   00035
LKEND   00036
SCEND   00037
SWITCH  00040
SWTCHS  00060
SETUP   00200
EAERMQ  00202
NOPAC   00225
EAECAC  00246
EAECLQ  00264
MQ1TAC  00313
NOPAC1  00341
NOPMQ   00366
NOPMQ1  00414
NOPLNK  00443
QONEAC  00510
EAESLK  00533
NOPLK1  00577
ACORMQ  00641
ACLMQ   00704
ACLMQE  00732
COMPMQ  00763
ACONEQ  01017
EAEABS  01045
NDSETU  01112
ERROR   01134
ERLOOP  01145
ERCONT  01163
TYPECN  01167
TCALL   01215
TCTWO   01224
CRLF    01240
TYDELE  01244
MIN5    01257
MIN6    01260
CHARK   01261
SVCHAR  01263
SEVEN   01271
TWO40   01272
TWO60   01273
SEVSEV  01274
SEVSIX  01275
SAVERR  01276
SVER    01277
K18     01300
SIXTY   01301
SEVNTY  01302
SEVN4   01303
FOUR1   01304
```

```
THREE7   01305
SIXONE   01306
ONESEV   01307
SEVONE   01310
SEVFIV   01311
THREE    01312
FOUR5    01313
FOUR4    01314
FOUR3    01315
THREE4   01316
FIVE6    01317
COMBIT   01320
BIT0     01321
BIT1     01322
BIT2     01323
BIT3     01324
BIT4     01325
BIT5     01326
BIT6     01327
BIT7     01330
BIT8     01331
BIT9     01332
BIT10    01333
BIT11    01334
BIT12    01335
BIT13    01336
BIT14    01337
BIT15    01340
BIT16    01341
BIT17    01342
NBIT0    01343
NBIT1    01344
NBIT2    01345
NBIT3    01346
NBIT4    01347
NBIT5    01350
NBIT6    01351
NBIT7    01352
NBIT8    01353
NBIT9    01354
NBIT10   01355
NBIT11   01356
NBIT12   01357
NBIT13   01360
NBIT14   01361
NBIT15   01362
NBIT16   01363
NBIT17   01364
KALL7    01365
HDR1     01366
HDR2     01375
HDR3     01411
HDR4     01427
SPACE3   01442
SPACE4   01444
```

HDR5	01447
TYPATR	01457
TYCOR	01461
TYINCO	01463
CRCODE	01465
TYNRMS	01466
TYINTE	01472
INDAT	01502
TYSTRT	01512
TYNOP	01514
TYCLA	01517
TYCLQ	01523
TYCMQ	01527
TYRMQ	01533
TYSLK	01537
TYSMQ	01543
TYLACQ	01547
TYLLS	01553
TYLLSS	01557
TYLRS	01563
TYSIMR	01567
TYLRSS	01573
TYRDSQ	01577
TYRES	01607
TYQINT	01612
TYLMQ	01622
TYABS	01626
TYCSC	01632
TYLACS	01636
TYSCER	01642
TYNORM	01646
TYSSC	01652
TYPLS1	01656
TYALSQ	01662
TYALS	01667
TYPTSR	01673
TYPCHR	01716
TYPSAV	01755
TYPCON	01762
TYPCO3	02001
TYPTYT	02013
SPAC	02022
TOCTAL	02026
TOCT1	02035
DECONT	02047
TYPOCT	02070
SPACE2	02075
TYCRLF	02101
OTY	02107
RL6	02115
SCTST1	02200
NOPSC	02225
SCT076	02246
SCT074	02265
SCT070	02303

```
SCT060  02321
SCT040  02337
SCT000  02355
SCT001  02373
SCT003  02411
SCT007  02427
SCT017  02445
SCT037  02463
SCT077  02501
SCERR   02520
NOPSC1  02555
ALSZER  02604
ALS01   02626
ALSLNK  02663
LNKALS  02721
KALS01  02732
ALSMQT  02760
SGNSHF  03042
NSNERR  03077
HSALS   03114
HSALSL  03127
HSALSE  03134
ALSERR  03175
SIMALS  03216
SALSRP  03244
RESULT  03245
ENDSHF  03267
COMMA   03307
LLSTS1  03310
LLSTS2  03342
LLSACT  03406
LLSTS3  03465
LLSTS4  03526
LLSTS5  03575
LLSSL1  03606
LLSSEX  03617
LLSTS6  03656
LLSSL2  03671
LLSSX2  03703
LLSERR  03745
KLLSS1  03767
LLSSER  03770
LRSTS1  04016
LRSTS2  04050
LRSTS3  04122
LRSTS4  04175
LRSTS5  04236
LRST5L  04254
LRSTS6  04313
LRST6L  04333
LRSTS7  04373
LRST7L  04405
LRST7E  04417
LRSTS8  04450
LRST8L  04463
```

```
LRST8E   04476
LLSSEQ   04527
LRSSEQ   04566
LRSERR   04630
LRSER1   04652
ACCOMK   04700
MQCOMK   04701
SCCOMK   04702
RANSHF   05002
SETLLS   05025
LRANLP   05046
LRANEX   05057
RLSTAY   05110
RANRIT   05112
SETLRS   05131
RRANLP   05151
RRANEX   05161
RRSTAY   05212
RANSEQ   05214
RANSQ0   05237
RANSQ1   05253
RANSQ2   05267
RANSQ3   05303
RANSQ4   05317
RANSQ5   05333
RANSQ6   05347
RANSQ7   05363
RANSQ8   05377
NXTSEQ   05417
SVSIGN   05446
SVMASK   05450
SEQCOM   05452
RANGEN   05522
RANNO    05533
PASSK    05535
KLRS     05536
KLLSS    05537
SHFBUF   05540
NRMLZE   05652
NRMLZ1   05703
NRMLZ2   05736
NRMLZ3   05777
NRMLZ4   06053
NR4A     06061
NR4B     06063
NR4C     06115
NRMLZ5   06141
NORMER   06214
NORMSE   06242
INTEST   06270
INTS1    06312
INTS2L   06326
INTS2    06344
INTS2E   06401
QNRM     06414
```

QNRM1	06420
QNRM2	06432
QNRM3	06440
NUCT	06466
NUVAL	06467
TEMY1	06470
TSR	101673
TY1	101716
OPS	101762
OPT	101762
TYT	102013
TSP	102022
TWORD	102026
TDIGIT	102070
TCR	102101
TIN	102101
CLSF	700001
CLOF	700004
CLON	700044
RSF	700101
RCF	700102
RSA	700104
RRB	700112
RSB	700144
PSF	700201
PCF	700202
PSA	700204
PSB	700244
KSF	700301
KRB	700312
TSF	700401
TCF	700402
TLS	700406
EEM	707702
LEM	707704