

PRODUCT CODE: MAINDEC-8E-D0LB-D
PRODUCT TEST: KES-E (EAE) INSTRUCTION TEST 1
DATE CREATED: FEBRUARY 9, 1972
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: ED FORTMILLER

COPYRIGHT © 1972
DIGITAL EQUIPMENT CORPORATION

File for Documentation, 26 min.

1. ABSTRACT

THIS PROGRAM IS A TEST OF ALL THE KEB-E EAE INSTRUCTIONS,
(EXCEPT MULTIPLY AND DIVIDE),

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-8/E OR /M PROCESSOR, KEB-E OPTION, AND A TELETYPE ARE REQUIRED,

2.2 STORAGE

LOCATIONS 0000 THROUGH 7600 ARE USED,

2.3 PRELIMINARY PROGRAMS

ALL PROCESSOR RELATED TEST PROGRAMS MUST HAVE BEEN RUN
SUCCESSFULLY,

3. LOADING PROCEDURE

3.1 METHOD

THE BINARY LOADER IS USED TO LOAD THE PROGRAM INTO ANY DE-
SIRED FIELD, REFER TO THE BINARY LOADER DOCUMENTATION IF
UNFAMILIAR WITH ITS USE,

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTING

CONTROL SWITCH SETTINGS DO NOT APPLY TO STEP COUNTER, GT, MODE,
AND COMBINED TESTS, AN ERROR WILL BE INDICATED BY A PROGRAM HALT,

SR0=1 HALT ON ERROR

SR1=1 SCOPE MODE (REPEAT PATTERN AND/OR TEST)

SR2=1 PRINT ERROR INFORMATION

SR3=1 DO NOT EXIT CURRENT TEST,

SR10=11	SR10	SR11	
	0	0	EXECUTE TEST IN "A" AND "B" MODES
	0	1	EXECUTE TEST IN "A" AND "B" MODES,
	1	0	SELECT "A" MODE,
	1	1	SELECT "B" MODE,

4,2 STARTING ADDRESS

THIS PROGRAM STARTS AT LOCATION 0200,

4,3 PROGRAM AND/OR OPERATOR ACTION

WITH THE PROGRAM LOADED IN CORE PROCEED AS FOLLOWS:

- A, INSURE TELETYPE IS ON-LINE,
- B, LOAD ADDRESS 0200,
- C, SET ANY DESIRED OPTIONS IN THE SR,
- D, PRESS CLEAR AND CONTINUE,

NOTE:

FOR A NORMAL PROGRAM RUN, SET SR SWITCHES TO 5000,
PROGRAM WILL RUN CONTINUOUSLY EXECUTING THE TEST IN "A"
AND "B" MODES UNLESS AN ERROR IS DETECTED AT WHICH TIME
THE ERROR INFORMATION WOULD BE PRINTED OUT AND THEN
THE PROGRAM WILL HALT,

5, ERRORS

5,1 PROGRAM AND/OR OPERATOR ACTION;

THE PURPOSE OF THIS PROGRAM IS TO DETECT OPERATION ERRORS
IN THE KE-8 HARDWARE. UPON DETECTION OF AN ERROR, THE
PROGRAM EITHER HALTS, OR GIVES AN ERROR PRINTOUT DE-
PENDING ON THE SETTING OF SR 0 AND 2.

ERROR HALTS AND DESCRIPTION

LOC 0236	MQL FAILED TO CLEAR THE AC OR THE LINK WAS CLEARED,
LOC 0447	MQL FAILED TO CLEAR THE AC OR THE LINK WAS SET,
LOC 0562	MQL FAILED TO LOAD THE MQ OR MQA FAILED TO LOAD THE AC,
LOC 0727	MQL FAILED TO LOAD THE MQ OR MQA FAILED TO LOAD THE AC,
LOC 1054	MQA FAILED TO "INCLUSIVE OR" THE MQ WITH THE AC.
LOC 1222	MQA FAILED TO "INCLUSIVE OR" THE MQ WITH THE AC.
LOC 1301	SAM INSTRUCTION FAILED,
LOC 1414	SAM INSTRUCTION FAILED,
LOC 1477	SHL INSTRUCTION FAILED,
LOC 1631	SHL INSTRUCTION FAILED,
LOC 1716	LSR INSTRUCTION FAILED,
LOC 2031	LSR INSTRUCTION FAILED,
LOC 2116	ASR INSTRUCTION FAILED,
LOC 2231	ASR INSTRUCTION FAILED,
LOC 2516	DPSZ INSTRUCTION FAILED,
LOC 2637	DPIC INSTRUCTION FAILED,

LOC 2703	DPIC INSTRUCTION FAILED,
LOC 3014	DCM INSTRUCTION FAILED,
LOC 3124	DAD INSTRUCTION FAILED,
LOC 3274	DAD INSTRUCTION FAILED,
LOC 3434	DST INSTRUCTION FAILED,
LOC 3531	DST INSTRUCTION FAILED,
LOC 3653	NORMALIZE INSTRUCTION FAILED,
LOC 4336	NORMALIZE INSTRUCTION FAILED,
LOC 4520	NORMALIZE INSTRUCTION FAILED,
LOC 4605	EAE NOP SKIPPED,
LOC 4610	EAE NOP MODIFIED THE AC,
LOC 4614	EAE NOP MODIFIED THE MQ,
LOC 4622	EAE CLA SKIPPED,
LOC 4624	EAE CLA FAILED TO CLEAR THE AC,
LOC 4630	EAE CLA MODIFIED THE MQ,
LOC 4637	AC OR MQ NOT CLEARED BY CAM,
LOC 4650	SWP FAILED,
LOC 4655	SWP FAILED,
LOC 4666	AQL FAILED,
LOC 4710	DLD (CAM DAD) FAILED,
LOC 4715	DLD (CAM DAD) FAILED,
LOC 4736	DDZ (CAM DST) FAILED,
4741	
4744	
LOC 4752	"B" MODE NORMALIZE FAILED TO CLEAR AC,
LOC 5004	CLEAR KEY FAILED TO SET "MODE A" OR DPSZ FAILED,
LOC 5010	SWAB FAILED TO SET "MODE B" OR DPSZ FAILED,
LOC 5015	SWBA FAILED TO SET "MODE A",
LOC 5023	CAF FAILED TO SET "MODE A",

LOC 5031	SCL OR SCA ERROR,
5040	
5047	
5056	
5065	
5074	
5103	
5112	
5117	
5126	
5135	
5145	
5155	
LOC 5163	ACS INSTRUCTION FAILED
5170	
5205	
5210	
LOC 5220	RTF INSTRUCTION FAILED TO SET GT FLAG TO 0 OR GTF FAILED TO GET IT,
LOC 5230	RTF INSTRUCTION FAILED TO SET GT FLAG TO 1 OR GTF FAILED TO GET IT,
LOC 5236	SGT SKIPPED WITH GT FLAG NOT SET,
LOC 5243	SGT FAILED TO SKIP WITH GT FLAG SET,
LOC 5252	SWBA FAILED TO CLEAR THE GT FLAG,

5,3 ERROR PRINTOUTS

5,3,1 MQL TESTS

MQLT MODE A (OR B)

AC 1 000000000011
0-AC 1 000000000001

MQLT MQL INSTRUCTION TEST WITH LINK SET TO A 1,
AC THE ORIGINAL C(AC) AND C(L),
0-AC C(AC) AND C(L) AFTER THE MQL INSTRUCTION WAS
EXECUTED,
NOTE THAT BIT 11 OF AC SHOULD EQUAL 0,

MQLT1 MODE A (OR B)

AC 0 000000000001
0-AC 0 000000000001

MQLT MQL INSTRUCTION TEST WITH LINK SET TO A 0,
AC THE ORIGINAL C(AC) AND C(L),
0-AC C(AC) AND C(L) AFTER THE MQL INSTRUCTION
WAS EXECUTED,
NOTE THAT BIT 11 OF THE AC SHOULD EQUAL 0,

5,3,2 MQA TESTS

MQAT MODE A (OR B)

AC 1 000000000001
MQL)1 000000000000
MQA)

MQAT MQA MQL INSTRUCTIONS TEST WITH THE LINK SET
TO A 1,
AC THE ORIGINAL C(AC) AND C(L),
MQL MQA THE C(AC) AND C(L) AFTER THE EXECUTION OF
AN MQL INSTRUCTION FOLLOWED BY AN MQA IN-
STRUCTION, NOTE THAT BIT 11 OF THE AC SHOULD
BE A 1,

MQAT1 MODE A (OR B)

AC 0 100000000000
MQL) 0 011111111111
MQA)

MQAT1 MQL, MQA INSTRUCTION TEST WITH THE LINK SET TO A 0,
 AC THE ORIGINAL C(AC) AND C(L),
 MQL, MQA THE C(AC) AND C(L) AFTER THE EXECUTION OF AN MQL
 INSTRUCTION FOLLOWED BY AN MQA INSTRUCTION,
 NOTE THAT THE C(AC) SHOULD BE 4000,

MQAT2 MODE A (OR B)

AC 1 111111111110
 MQ 000000000001
 MQVAC 1 000000000000

MQAT2 MQA INSTRUCTION TEST,
 AC ORIGINAL C(AC) AND C(L),
 MQ ORIGINAL C(MQ);
 MQVAC THE C(AC) AND C(L) AFTER THE EXECUTION OF AN MQA
 INSTRUCTION,
 NOTE THAT THE C(AC) SHOULD BE 7777,

MQAT3 MODE A (OR B)

AC 0 111111111110
 MQ 000000000001
 MQVAC 0 000000000000

MQAT3 MQA INSTRUCTION TEST,
 AC ORIGINAL C(L) AND C(AC),
 MQ ORIGINAL C(MQ);
 MQVAC THE C(AC) AND THE C(L) AFTER THE EXECUTION
 OF AN MQA INSTRUCTION,
 NOTE THAT THE C(AC) SHOULD BE 7777,

5,3,3 SAM TESTS

SAM TEST 0 (OR 1) MODE B

	C(L)	C(AC)	C(MQ)	C(GT)	C(SC)
PROBLEM	1	000000111111	000001000000	0	000000000000
SIMULATED	0	000000000001	000001000000	1	000000000000
ACTUAL	0	000000000000	000001000000	1	000000000000

PROBLEM THE CONTENTS OF THE REGISTERS PRIOR TO ISSUING
 A SAM INSTRUCTION,
 SIMULATED WHAT THE RESULTS SHOULD BE,
 ACTUAL WHAT THE RESULTS WERE FROM THE EXECUTION OF
 A SAM INSTRUCTION,
 NOTE THAT AC SHOULD BE 0001 IN THE "ACTUAL".

5,3,4 SHL TESTS

SHL TEST 0 (OR 1) 0003 SHIFTS MODE A (OR B)

	C(L)	C(AC)	C(MQ)	C(GT)	C(SC)
PROBLEM	1	000100100001	000100000001	0	000000000010
SIMULATED	0	100100001000	100000001000	0	000000000000
ACTUAL	0	100100000000	100000001000	0	000000000000

PROBLEM THE CONTENTS OF THE REGISTERS PRIOR TO ISSUING AN SHL INSTRUCTION, THE NUMBER CONTAINED IN C(SC) IS THE CONTENTS OF THE ADDRESS FOLLOWING THE SHL INSTRUCTION,
SIMULATED WHAT THE RESULTS SHOULD BE,
ACTUAL WHAT THE RESULTS WERE FROM THE EXECUTION OF AN SHL INSTRUCTION.

5,3,5 LSR TESTS

LSR TEST 1 (OR 0) 0004 SHIFTS MODE B (OR A)

	C(L)	C(AC)	C(MQ)	C(GT)	C(SC)
PROBLEM	1	000000111111	000000011111	0	000000001000
SIMULATED	0	000000000011	111000000000	1	000000111111
ACTUAL	0	000000000011	111000000000	0	000000111111

PROBLEM THE CONTENTS OF THE REGISTERS PRIOR TO ISSUING AN LSR INSTRUCTION, THE NUMBER CONTAINED IN C(SC) IS THE CONTENTS OF THE ADDRESS FOLLOWING THE LSR INSTRUCTION,
SIMULATED WHAT THE RESULTS SHOULD BE,
ACTUAL WHAT THE RESULTS WERE FROM THE EXECUTION OF AN LSR INSTRUCTION,
NOTE THAT THE C(GT) SHOULD BE 1 IN THE "ACTUAL".

5,3,6 ASR TESTS

ASR TEST 0 (OR 1) 0002 SHIFTS MODE B (OR A)

	C(L)	C(AC)	C(MQ)	C(GT)	C(SC)
PROBLEM	0	000000000110	00000100001	1	00000000010
SIMULATED	1	110000000001	10000001000	0	00000011111
ACTUAL	1	110000000001	00000001000	0	00000011111

PROBLEM THE CONTENTS OF THE REGISTERS PRIOR TO ISSUING AN ASR INSTRUCTION, THE NUMBER CONTAINED IN C(SC) IS THE CONTENTS OF THE ADDRESS FOLLOWING THE ASR INSTRUCTION,
SIMULATED WHAT THE RESULTS SHOULD BE,
ACTUAL WHAT THE RESULTS WERE FROM THE EXECUTION OF AN ASR INSTRUCTION,
NOTE THAT THE C(MQ) SHOULD BE 4010 IN THE "ACTUAL,"

5,3,7 DPSZ TESTS

DPSZ TEST 0 MODE B

NO SKIP OCCURRED

	C(L)	C(AC)	C(MQ)	C(GT)	C(SC)
PROBLEM	0	000000000000	00000000000	0	00000000000
SIMULATED	0	000000000000	00000000000	0	00000000000
ACTUAL	0	000000000000	00000000000	0	00000000000

PROBLEM THE CONTENTS OF THE REGISTERS PRIOR TO ISSUING A DPSZ INSTRUCTION,
SIMULATED WHAT THE REGISTERS SHOULD BE AFTER ISSUING THE DPSZ INSTRUCTION,
ACTUAL WHAT THE REGISTERS WERE AFTER ISSUING THE DPSZ INSTRUCTION,
NOTE WITH AC AND MQ BOTH ZERO A SKIP FAILED TO OCCUR,

DPSZ TEST 0 MODE B

SKIP OCCURRED

	C(L)	C(AC)	C(MQ)	C(GT)	C(SC)
PROBLEM	0	000000000000	00000000001	0	00000000000
SIMULATED	0	000000000000	00000000000	0	00000000000
ACTUAL	0	000000000000	00000000000	0	00000000000

PROBLEM THE CONTENTS OF THE REGISTER PRIOR TO ISSUING A DPSZ INSTRUCTION,
SIMULATED WHAT THE REGISTERS SHOULD BE AFTER ISSUING THE DPSZ INSTRUCTION,
ACTUAL WHAT THE REGISTERS WERE AFTER ISSUING THE DPSZ INSTRUCTION,
NOTE WITH MQ NON ZERO A SKIP OCCURRED,

DPSZ TEST 0 MODE B

REG MODIFIED

	C(L)	C(AC)	C(MQ)	C(GT)	C(SC)
PROBLEM	0	000000000001	000000000000	0	000000000000
SIMULATED	0	000000000001	000000000000	0	000000000000
ACTUAL	0	000000000000	000000000000	0	000000000000

PROBLEM THE CONTENTS OF THE REGISTERS PRIOR TO ISSUING A
 DPSZ INSTRUCTION,
 SIMULATED WHAT THE REGISTERS SHOULD BE AFTER ISSUING THE
 DPSZ INSTRUCTION,
 ACTUAL WHAT THE REGISTERS WERE AFTER ISSUING THE DPSZ
 INSTRUCTION,
 NOTE THAT THE C(AC) SHOULD BE 0001 IN THE "ACTUAL".

5,3,8 DPIC TESTS

DPIC TEST 0 (OR 1) MODE B

	C(L)	C(AC)	C(MQ)	C(GT)	C(SC)
PROBLEM	1	000000000000	000000000000	0	000000000000
SIMULATED	0	000000000000	000000000001	0	000000000000
ACTUAL	0	000000000000	000000000000	0	000000000000

PROBLEM THE CONTENTS OF THE REGISTERS PRIOR TO ISSUING A
 DPIC INSTRUCTION,
 SIMULATED WHAT THE REGISTERS SHOULD BE AFTER THE ISSUING
 OF A DPIC INSTRUCTION,
 ACTUAL WHAT THE REGISTERS WERE AFTER ISSUING THE DPIC
 INSTRUCTION,
 NOTE THAT THE C(AC) SHOULD BE 0001 IN THE "ACTUAL".

5,3,9 DCM TESTS

DCM TEST 0 (OR 1) MODE B

	C(L)	C(AC)	C(MQ)	C(GT)	C(SC)
PROBLEM	1	000000000000	000000000001	0	000000000000
SIMULATED	0	111111111111	111111111111	0	000000000000
ACTUAL	0	111111111111	111111111110	0	000000000000

PROBLEM THE CONTENTS OF THE REGISTERS PRIOR TO ISSUING A
 DCM INSTRUCTION,
 SIMULATED WHAT THE REGISTERS SHOULD BE AFTER THE ISSUING OF
 A DCM INSTRUCTION,
 ACTUAL WHAT THE REGISTERS WERE AFTER ISSUING THE DCM
 INSTRUCTION,
 NOTE THAT THE C(MQ) SHOULD BE 7777.

5,3,10 DAD TESTS

 DAD TEST 0 (OR 1) MODE B

	C(L)	C(AC)	C(MQ)	C(GT)	C(SC)
PROBLEM	0	000000011111	000011110101	0	000000000000
TO BE ADDED		111111000000	11100001010		
SIMULATED	0	111111111111	111111111111	0	000000000000
ACTUAL	0	111111111111	111111111110	0	000000000000

PROBLEM THE CONTENTS OF THE REGISTERS PRIOR TO ISSUING
 A DAD INSTRUCTION,
 TO BE ADDED THE CONTENTS OF THE TWO ADDRESS TO BE ADDED TO
 THE AC AND MQ,
 SIMULATED WHAT THE REGISTERS SHOULD BE AFTER THE ISSUING
 THE DAD INSTRUCTION,
 ACTUAL WHAT THE REGISTERS WERE AFTER ISSUING THE DAD
 INSTRUCTION,
 NOTE THAT C(MQ) SHOULD BE 7777 IN THE ACTUAL,

5,3,11 DST TESTS

 DST TEST 0 (OR 1) MODE B

REG	BEFORE DST	AFTER DST
C(L)	1	1
C(AC)	111111111111	111111111111
C(MSH)		000000000000
C(MQ)	11110111110	11110111110
C(LSH)		11110111110

BEFORE DST THE CONTENTS OF THE REGISTERS BEFORE ISSUING A
 DST INSTRUCTION,
 AFTER DST WHAT THE REGISTERS WERE AFTER ISSUING THE DST
 INSTRUCTION,
 MSH IS WHAT DST STORED FOR THE AC,
 LSH IS WHAT DST STORED FOR THE MQ,
 NOTE THAT THE C(MSH) SHOULD BE 7777,

5,3,12 NORMALIZE TESTS

NMIT	C(AC)	C(MQ)	MODE A (OR B)
	000000000000	010101010101	
NMI	010101010100	000000000000	
SCAT	000000001100		
SCA	000000001100		

NMIT NORMALIZE AND STEP COUNTER TEST, ORIGINAL C(AC) AND
 C(MQ),

(5,3.12 CONT'D)

NMI C(AC) AND C(MQ) AFTER THE NMI INSTRUCTION WAS EXECUTED,
SCAT THE CORRECT COUNT OF THE STEP COUNTER AFTER THE NORMALIZE
INSTRUCTION WAS EXECUTED,
SCA THE ACTUAL COUNT IN THE STEP COUNTER AS READ INTO THE AC
BY THE SCA INSTRUCTION AFTER THE NORMALIZE INSTRUCTION
WAS EXECUTED,

NOTE THAT BIT 11 OF AC IN ERROR, C(AC) SHOULD EQUAL 2525.

5.4 TABLE OF INSTRUCTIONS

THE FOLLOWING TABLE CONTAINS THE TEST MNEMONIC, STARTING ADDRESS,
ERROR HALT ADDRESS AND INSTRUCTION TESTED,

MNEMONIC	INSTRUCTIONS	STARTING ADDRESS	ERROR HALT
MQLT	MQL	0204	0236
MQLT1	MQL	0400	0447
MQAT	MQL, MQA	0503	0562
MQAT1	MQL, MQA	0650	0727
MQAT2	MQA	1000	1054
MQAT3	MQA	1135	1222
SAMTS0	SAM	1245	1301
SAMTS1	SAM	1333	1414
SHLTS0	SHL	1430	1477
SHLTS1	SHL	1600	1631
LSRTS0	LSR	1646	1716
LSRTS1	LSR	2000	2031
ASRTS0	ASR	2046	2116
ASRTS1	ASR	2200	2231
DPSZS0	DPSZ	2246	2516
DPITS0	DPIC	2600	2637
DPITS1	DPIC	2653	2703
DCMTS0	DCM	2717	3014
DADTS0	DAD	3030	3124
DADTS1	DAD	3200	3274
DSTTS0	DST	3310	3434
DSTTS1	DST	3450	3531
NORMT	NMI, SCA	3600	3653
NORMT1	NMI, SCA	4200	4336
NORMT2	NMI	4400	4520
COMTST	NOP, CLA, ACL, CAM SWP, DLD, DDZ, NMI	4600	4605-4744
MDTST	DPSZ, SWAB, SWBA	0200	5004-5023
TSCL	SCL, ACS	0200	5031-5210
GTTST	GTF, RTF, SGT	0200	5220-5252

6. DESCRIPTION

THE KE8 EAE INSTRUCTION TEST 1, TESTS THE FOLLOWING EXTENDED ARITHMETIC ELEMENT INSTRUCTIONS:

MQL, MGA, SHL, LSR, ASR, DPSZ, DPIC, DCM, DAD,
DST, NMI, SWAB, SWBA, SGT, RTF, AND GTF,

THE EXTENDED ARITHMETIC ELEMENT IS TESTED USING PATTERNS NECESSARY TO DETECT AND ISOLATE ERRORS; IF A FAILURE DOES OCCUR, THE TEST WILL PRINTOUT THE ERROR INFORMATION AND/OR HALT AT A PREDETERMINED ERROR HALT ACCORDING TO THE SR SETTING,

7. EXECUTION TIME

ONE COMPLETE PROGRAM PASS TAKES APPROXIMATELY 6 MINUTES AND AT THE END OF EACH PASS "KE8 1" WILL BE PRINTED OUT ON THE TELETYPE.

/KES EAE INSTRUCTION TEST PART 1 MAINDEC-8E-D8LB
 /COPYRIGHT 1972, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS, 01754
 /PROGRAMMER: ED FORTMILLER

/SWITCH REGISTER OPTIONS:

/SR0=1 HALT ON ERROR
 /SR1=1 SCOPE MODE (REPEAT PATTERN AND/OR TEST)
 /SR2=1 PRINT ERROR INFORMATION
 /SR3=1 DO NOT EXIT CURRENT TEST

/SR10=11	SR10	SR11	
/ 01	0	0	EXECUTE TEST IN "A" AND "B" MODES
/ 0	0	1	EXECUTE TEST IN "A" AND "B" MODES
/ 1	1	0	SELECT "A" MODE
/ 1	1	1	SELECT "B" MODE

7421	MLQ=7421	/LOAD MQ,
7501	MQA=7501	/INCLUSIVE OR MQ WITH AC,
7401	NOPM=7401	/EAE NOP,
7601	CLAM=7601	/EAE CLA,
7411	NMI=7411	/NORMALIZE
7413	SHL=7413	/SHIFT LEFT
7415	ASR=7415	/ARITHMETIC SHIFT RIGHT
7417	LSR=7417	/LOGICAL SHIFT RIGHT
7521	SWP=7521	/SWAP AC AND MQ,
7621	CAM=7621	/CLEAR AC AND MQ,
7701	ACL=CLAM MQA	/LOAD AC FROM MQ,
7441	SCA=7441	/STEP COUNTER TO AC,
7431	SWAB=7431	/SWITCH FROM MODE "A" TO "B",
7447	SWBA=7447	/SWITCH FROM MODE "B" TO "A",
7403	SCL=7403	/STEP COUNTER LOAD FROM MEMORY,
7403	ACS=7403	/ACCUMULATOR TO STEP COUNTER,
7497	SAH=7497	/SUBTRACT AC FROM MQ,
7443	DAD=7443	/DOUBLE PRECISION ADD,
7445	DST=7445	/DOUBLE PRECISION STORE,
7573	DPIC=7573	/DOUBLE PRECISION INCREMENT,
7575	DCM=7575	/DOUBLE PRECISION COMPLEMENT,
7491	DPSE=7491	/DOUBLE PRECISION SKIP IF ZERO,
7663	DLD=DAD CAM	/DOUBLE PRECISION LOAD,
7665	DDE=DST CAM	/DOUBLE PRECISION DEPOSIT ZERO,
6001	ION=6001	/TURN THE INTERRUPT ON,
6002	IOP=6002	/TURN THE INTERRUPT OFF,
6004	GTF=6004	/GET THE INTERRUPT FLAGS,
6005	RTP=6005	/RESTORE THE INTERRUPT FLAGS,
6006	SGT=6006	/SKIP ON GREATER THAN FLAG,
6007	CAF=6007	/CLEAR THE WORLD,
6214	RDF=6214	/READ THE DATA FIELD
6224	RIF=6224	/READ THE INSTRUCTION FIELD,
7002	BSW=7002	/SWAP BYTES IN AC,
7400	NOP=7400	/GROUP 2 NOP,
7402	HLT=7402	/HALT,

0000	OPEN=0000	/PROGRAM MODIFIABLE,
0000	*0	
0000	0000	0
0001	0001	JMP 1
0002	0002	2
0003	0003	3
0020	*0020	
0020	0000	FILLER, 0 /SET TO NUMBER OF FILLER CHARACTERS NEEDED,

/THESE STORAGE LOCATIONS FROM "TOLINK THROUGH "ADDR" MUST
 /REMAIN IN THE ORDER SHOWN,

0021	0000	TOLINK, OPEN
0022	0000	TOAC, OPEN
0023	0000	TOHQ, OPEN
0024	0000	TOSHIF, OPEN
0025	0000	TOGT, OPEN
0026	0000	TLINK, OPEN
0027	0000	TAC, OPEN
0030	0000	THQ, OPEN
0031	0000	TSHIF, OPEN
0032	0000	TGT, OPEN
0033	0000	LKTOCK, OPEN
0034	0000	ACTOCK, OPEN
0035	0000	HQTOCK, OPEN
0036	0000	SCTOCK, OPEN
0037	0000	GITOCK, OPEN
0040	0000	TEMPA, OPEN
0041	0000	TEMPB, OPEN
0042	0000	LSIM, OPEN
0043	0000	MSH, OPEN
0044	0000	LSH, OPEN
0045	0000	SCSIM, OPEN
0046	0000	GTSIM, OPEN
0047	0000	ADDR, OPEN
0050	6600	XIYPSY, TYPST
0051	6670	UPSPC, PSPC
0052	6345	UCOMP, COMP
0053	6400	UMOVE, MOVE
0054	6723	UISPC, SPACE1
0055	6727	UISPC, SPACE2
0056	0000	BACK, 0000
0057	0000	NEXT, 0000
0060	0400	XMQLT1, MQLT1
0061	0503	XMQAT, MQAT
0062	0650	XMQAT1, MQAT1

```

0063 0000 ACP, 0 /GOOD AC
0064 0000 LXP, 0 /GOOD LINK
0065 0000 GENX, 0
0066 0000 BLXP, 0 /BAD LINK
0067 0000 BACP, 0 /BAD AC
0070 0215 CR, 0215 /CARRIAGE RETURN
0071 0212 LF, 0212 /LINE FEED
0072 0315 M, 0315 /M
0073 0321 Q, 0321 /Q
0074 0314 LL, 0314 /L
0075 0324 TT, 0324 /T
0076 0301 A, 0301 /A
0077 0303 C, 0303 /C
0100 0261 ONE, 0261 /1
0101 0260 ZERO, 0260 /0
0102 0000 LINK, 0
0103 0255 TO, 0255 /DASH
0104 7763 COUNTX, 7763
0105 0000 STRCNT, 0
0106 0000 BITSTR, 0
0107 1000 XMQAT2, MQAT2
0110 1135 XMQAT3, MQAT3
0111 0326 INCOR, 0326 /V
0112 0263 THREE, 0263 /3
0113 0262 TWO, 0262
0114 0000 SCOUNT, OPEN
0115 0000 MODE, OPEN
0116 0000 ANYUSE, OPEN /GENERAL USE,
0117 0000 COUNT, OPEN
0120 0000 TWICE, OPEN
0121 0000 DPMQ, OPEN
0122 0000 DPAC, OPEN

```

```

0123 7740 K7740, 7740
0124 5600 CRLF, UCRLF
0125 5607 CRLF2, UCRLF2
0126 5613 PRXLOP, RXLOP
0127 5640 PLINK, UPLINK
0130 5656 ZEROR, UZEROR
0131 5663 MESSG, UMESSG
0132 5645 ONZER, UONZER
0133 5652 ONEP, UONEP
0134 5707 TYTST, UTYTST
0135 5274 MOOSEL, MOSEL
0136 5317 ONLYB, UONLYB
0137 7000 PREGS, UPREGS
0140 5520 ASCOMP, SCOMP
0141 5410 SAVREG, USVREG
0142 5546 UGEN, GEN
0143 5325 TSTSW0, SW0TST
0144 5333 TSTSW1, SW1TST

```

```

0145 5342 TSTSW2, SW2TST
0146 5400 TSTSW3, SW3TST
0147 7070 NUMSHF, NUMSH
0150 6525 RANDOM, RANGEN
0151 5751 LDREG, ULDREG
0152 5732 RNDATA, RANDAT
0153 5761 LOGT, ULOGT
0154 6000 RTPX, XRTF
0155 5726 P1BIT, UP1BIT
0156 5503 LDSC, ULDSC

0010 0010 *0010
0010 0000 ACIND, 0
0011 0000 MQIND, 0
0012 0000 XACNMI, 0
0013 0000 XMGNMI, 0
0200 6007 CAF
0201 3115 DCA MODE /MODE "A" INITIALLY.
0202 7621 CAM
0203 4577 JMS I [TSQL /TEST MODE SWITCHING, GT, AND SC,

/TEST THAT MQL WILL CLEAR THE AC AND LEAVE THE LINK SET TO 1,

0204 5244 MQLT, JMP HSE /HOUSE KEEPING
0205 4542 JMS I UGEN
0206 7360 STL CLA CMA /SET LINK
0207 0045 AND GENX
0210 3043 DCA ACP /STORE AC PATTERN
0211 7240 CLA CMA
0212 3044 DCA LXP /STORE LINK TO A ONE
0213 1043 TAD ACP /LOAD AC,
0214 7421 MQL
0215 3067 DCA BACP /STORE AC RESULT
0216 7620 CLA SNL
0217 5345 JMP XPACP+5 /STORE LINK RESULT 0000
0220 7240 CLA CMA
0221 3066 DCA BLXP /STORE LINK RESULT 7777
0222 1047 TAD BACP
0223 7640 SZA CLA
0224 5231 JMP ,+5 /AC NOT EQUAL TO 0000
0225 1046 TAD BLXP
0226 7450 SNA
0227 5231 JMP ,+2 /LINK NOT EQUAL TO A ONE
0230 5237 JMP ,+7 /CONTINUE TEST MQLT
0231 4545 JMS I TSTSW2 /TEST SW2
0232 4254 JMS PMQLT /PRINT ERROR
0233 7704 CLL CLA OSR /TEST SW 0
0234 7004 RAL
0235 7430 SEL
0236 7402 HLT /HALT MQL ERROR
0237 7604 CLA OSR

```



```

0240 7106      RTL CLL      /TEST SW1
0241 7430      SEL
0242 5206      JMP MQLT+2    /PROGRAM LOOP
0243 5205      JMP MQLT+1    /CONTINUE PROGRAM

0244 7300      HSE,      CLA CLL
0245 3065      DCA GENX
0246 1344      TAD XPACP+4
0247 3056      DCA BACK
0250 1000      TAD XMQLT1
0251 3057      DCA NEXT
0252 4335      JMS I      MOOSEL      /PERFORM MODE SELECTION,
0253 5205      JMP MQLT+1

0254 0000      PHQLT, 0      /PRINT ROUTINE
0255 4325      JMS I      CRLF2      /CR AND LF;
0256 4302      JMS HQ
0257 4311      JMS L
0260 4316      JMS T

0261 4576      CP,      JMS I      CTYMOD      /TYPE THE MODE;
0262 4524      JMS I      CRLF      / CR AND LF,
0263 4455      JMS I      U2SPC      /2 SPACES,
0264 4323      JMS AC
0265 4455      JMS I      U2SPC      /2 SPACES,
0266 4332      JMS PLXP
0267 4494      JMS I      U1SPC      /1 SPACE,
0270 4740      JMS I      XPACP
0271 4524      JMS I      CRLF      /CR AND LF;
0272 4530      JMS I      ZEROR
0273 4741      JMS I      XPACP+1    /RIGHT ARROW
0274 4323      JMS AC
0275 4455      JMS I      U2SPC      /2 SPACES,
0276 4742      JMS I      XPACP+2
0277 4494      JMS I      U1SPC      /1 SPACE,
0300 4743      JMS I      XPACP+3
0301 5694      JMP I      PHQLT      /RETURN TO SWITCH ROUTINE

0302 0000      HQ,      0
0303 7240      CLA CMA
0304 0072      AND M
0305 4526      JMS I      PRXLOP      /M      /PRINT,
0306 1073      TAD      C
0307 4526      JMS I      PRXLOP      /"C",    /PRINT,
0310 5702      JMP I      HQ

0311 0000      L,      0
0312 7240      CLA CMA
0313 0074      AND LL
0314 4526      JMS I      PRXLOP      /L      /PRINT,
0315 5711      JMP I      L

0316 0000      T,      0

```

```

0317 7240      CLA CMA
0320 0075      AND TT
0321 4526      JMS I      PRXLOP      /T      /PRINT,
0322 5716      JMP I      T

0323 0000      AC,      0
0324 7240      CLA CMA
0325 0076      AND A
0326 4526      JMS I      PRXLOP      /A      /PRINT,
0327 1077      TAD      C
0330 4526      JMS I      PRXLOP      /"C",    /PRINT,
0331 5723      JMP I      AC

0332 0000      PLXP, 0
0333 7240      CLA CMA
0334 0064      AND LXP      /GOOD LINK
0335 3102      DCA LINK
0336 4527      JMS I      PLINK
0337 5732      JMP I      PLXP

0340 0362      XPACP, PACP
0341 0355      PTO
0342 0347      PBLXP
0343 0370      PBACP
0344 0204      MQLT
0345 3066      DCA BLXP
0346 5222      JMP MQLT-16

0347 0000      PBLXP, 0
0348 7240      CLA CMA
0349 0066      AND BLXP      /BAD LINK
0352 3102      DCA LINK
0353 4527      JMS I      PLINK
0354 5747      JMP I      PBLXP

0355 0000      PTO, 0
0356 7240      CLA CMA
0357 0103      AND TO
0360 4526      JMS I      PRXLOP      /RIGHT ARROW /PRINT,
0361 5755      JMP I      PTO

0362 0000      PACP, 0
0363 7240      CLA CMA
0364 0063      AND ACP
0365 3106      DCA BITSTR
0366 4531      JMS I      MESSG      /PRINT A MESSAGE,
0367 5762      JMP I      PACP

0370 0000      PBACP, 0

```

```

0371 7240      CLA CMA
0372 0067      AND BACP      /BACP
0373 3106      DCA BITSTR
0374 4531      JMS I  MESSG      /PRINT A MESSAGE;
0375 5770      JMP I  PBACP
                PAGE
0400
                /TEST THAT MQL WILL CLEAR THE AC AND LEAVE THE LINK CLEARED,
0400 5227      MQLT1, JMP HSE1
0401 4542      JMS I  UGEN
0402 7340      CLL  CLA CMA      /CLEAR LINK
0403 0065      AND GENX
0404 3063      DCA ACP      /STORE AC PATTERN
0405 3064      DCA LXP      /STORE LINK TO A ZERO
0406 7040      CMA
0407 0063      AND ACP      /LOAD AC
0410 7421      MQL
0411 3067      DCA BACP      /STORE AC RESULT
0412 7620      CLA SNL
0413 5301      JMP XONE+6      /STORE LINK RESULT 0000
0414 7240      CLA CMA
0415 3066      DCA BLXP      /STORE LINK RESULT 7777
0416 7040      CMA
0417 0067      AND BACP
0420 7440      SZA
0421 5237      JMP MQ1SW      /AC NOT EQUAL TO 0000
0422 7240      CLA CMA
0423 0066      AND BLXP
0424 7440      SZA
0425 5237      JMP MQ1SW      /LINK NOT EQUAL TO A ZERO
0426 5250      JMP MQL+4      /CONTINUE TEST MQLT1

0427 7300      HSE1,  CLA CLL
0430 3065      DCA GENX
0431 1060      TAD XMQLT1
0432 3056      DCA BACK
0433 1061      TAD XMCAT
0434 3057      DCA NEXT
0435 4535      JMS I  MODSEL      /PERFORM MODE SELECTION;
0436 5201      JMP MQLT1+1

0437 7604      MQ1SW, CLA OSR      /TEST SW2
0440 7106      RTL CLL
0441 7004      RAL
0442 7430      SZL
0443 5256      JMP XMQL+1      /PRINT ERROR

0444 7604      MQ1,   CLA OSR      /TEST SW0
0445 7104      RAL CLL
0446 7430      SZL
0447 7402      HLT      /MQL ERROR,

```

```

0450 7604      CLA OSR
0451 7106      RTL CLL
0452 7430      SZL
0453 5202      JMP MQLT1+2      /PROGRAM LOOP
0454 5201      JMP MQLT1+1      /CONTINUE PROGRAM

0455 0444      XMQL,  MQ1
0456 7240      CLA CMA
0457 0255      AND XMQL
0460 3700      DCA I  XONE+5
0461 4525      JMS I  CRLF2      /2 CR AND LF;
0462 4670      JMS I  XCP+1
0463 4671      JMS I  XCP+2
0464 4672      JMS I  XCP+3
0465 4273      JMS XONE
0466 5667      JMP I  XCP

0467 0261      XCP,   CP
0470 0302      MQ
0471 0311      L
0472 0316      T

0473 0000      XONE,  0
0474 7240      CLA CMA
0475 0100      AND ONE      /ONE
0476 4526      JMS I  PRXLOP      /PRINT;
0477 5673      JMP I  XONE
0500 0254      PMQLT
0501 3066      DCA BLXP
0502 5216      JMP MQLT1+16

0503 5340      MQAT,  JMP HSE2
0504 4542      JMS I  UGEN
0505 7360      STL  CLA CMA      /SET LINK
0506 0065      AND GENX
0507 3063      DCA ACP      /STORE AC PATTERN
0510 7240      CLA CMA
0511 3064      DCA LXP      /STORE LINK TO A ONE
0512 7040      CMA
0513 0063      AND ACP      /LOAD AC
0514 7421      MQL
0515 7501      MQA      /LOAD MQ FROM AC
0516 3067      DCA BACP      /STORE RESULT OF MQL, MQA
0517 7620      CLA SNL
0520 5777      JMP YA+5      /STORE LINK RESULT 0000
0521 7240      CLA CMA
0522 3066      DCA BLXP      /STORE LINK RESULT 7777

0523 7040      RL2,   CMA
0524 0063      AND ACP      /COMPARE ACP WITH BACP
0525 7140      CLL  CMA
0526 1067      TAD BACP
0527 7040      CMA

```

```

0530 7490 SNA
0531 7430 S2L
0532 5390 JMP HSE2+10 /MQ DID NOT EQUAL AC
0533 7240 CLA CMA
0534 0866 AND BLXP
0535 7490 SNA
0536 5390 JMP HSE2+10 /LINK DID NOT EQUAL A ONE
0537 5363 JMP HSE2A

0540 7300 HSE2, CLA CLL
0541 3065 DCA GENX
0542 1061 TAD XMQAT
0543 3056 DCA BACK
0544 1062 TAD XMQAT1
0545 3057 DCA NEXT
0546 4935 JMS I MOOSEL /PERFORM MODE SELECTION,
0547 5304 JMP MQAT+1
0550 7604 CLA OSR /TEST SW2
0551 7106 RTL CLL
0552 7004 RAL
0553 7420 SNL
0554 5357 JMP ,+3
0555 4776 JMS MQAT /PRINT ERROR
0556 4775 JMS MQA1
0557 7604 CLA OSR /TEST SW0
0560 7104 RAL CLL
0561 7430 S2L
0562 7402 HLT /MQL OR MQA ERROR,
0563 7604 HSE2A, CLA OSR /TEST SW1
0564 7106 RTL CLL
0565 7430 S2L
0566 5305 JMP MQAT+2 /PROGRAM LOOP
0567 5304 JMP MQAT+1 /CONTINUE PROGRAM

0575 0605
0576 0600
0577 0600 PAGE
    
```

```

0600 0800 MQAT, 0
0601 4525 JMS I CRLF2 /2 CR AND LF,
0602 4777 JMS MQ /PRINT "MQ",
0603 4232 JMS AT
0604 5600 JMP I MQAT

0605 0800 MQA1, OPEN
0606 4576 JMS I ITHMOD /TYPE THE MODE,
0607 4924 JMS I CRLF /CR AND LF,
0610 4451 JMS I UPSPC /5 SPACES,
0611 7773 -5
0612 4776 JMS AC /PRINT "AC",
0613 4455 JMS I U2SPC /2 SPACES,
    
```

```

0614 4775 JMS PLXP /
0615 4454 JMS I U1SPC /1 SPACE,
0616 4774 JMS PACP /
0617 4524 JMS I CRLF /CR AND LF,
0620 4777 JMS MQ /PRINT "MQ",
0621 4773 JMS L /PRINT "L",
0622 4454 JMS I U1SPC /1 SPACE,
0623 4777 JMS MQ /PRINT "MQ",
0624 4241 JMS YA /PRINT "YA",
0625 4455 JMS I U2SPC /2 SPACES,
0626 4772 JMS PBLXP /
0627 4454 JMS I U1SPC /1 SPACE,
0630 4771 JMS PBACP /
0631 5605 JMP I MQA1 /EXIT,

0632 0800 AT, 0
0633 7240 CLA CMA
0634 0876 AND A /A
0635 4526 JMS I PRXLOP /PRINT,
0636 1075 TAD TI /"T",
0637 4526 JMS I PRXLOP /PRINT,
0640 5632 JMP I AT

0641 0800 YA, 0
0642 7240 CLA CMA
0643 0876 AND A /A
0644 4526 JMS I PRXLOP /PRINT,
0645 5641 JMP I YA
0646 3066 DCA BLXP
0647 5770 JMP RL2
    
```

/TEST OF MQL WITH THE LINK SET TO 0

```

0650 4304 MQAT1, JMS HSE3
0651 4542 JMS I UGEN
0652 7340 CLL CLA CMA /CLEAR LINK
0653 0865 AND GENX
0654 3063 DCA ACP /STORE AC PATTERN
0655 3064 DCA LXP /STORE LINK TO A ZERO
0656 7040 CMA
0657 0863 AND ACP /LOAD AC
0660 7421 MQL /LOAD MQ FROM AC
0661 7501 MQA /LOAD AC FROM MQ
0662 3067 DCA BACP /STORE RESULT OF MQL, MQA
0663 7620 CLA SNL
0664 5340 JMP NOPR+14
0665 7240 CLA CMA
0666 3066 DCA BLXP /STORE LINK RESULT 7777
0667 7040 CMA
0670 0863 AND ACP /COMPARE ACP WITH BACP
0671 7140 CLL CMA
0672 1067 TAD BACP
    
```

```

0673 7040      CHA
0674 7450      SNA
0675 7430      SEL
0676 5314      JMP MGAER1    /MO DID NOT EQUAL AC
0677 7240      CLA CMA
0700 0066      AND BLXP
0701 7440      SEA
0702 5314      JMP MGAER1    /LINK DID NOT EQUAL A ZERO
0703 5330      JMP NOPR+4

0704 7300      HSE3,  CLA CLL
0705 3065      DCA GENX
0706 1062      TAD XMGAT1
0707 3056      DCA BACK
0710 1107      TAD XMGAT2
0711 3057      DCA NEXT
0712 4535      JMS I  MODSEL    /PERFORM MODE SELECTION.
0713 5251      JMP MGAT1+1

```

```

0714 7604      MGAER1, CLA OSR    /TEST SW2
0715 7106      RTL CLL
0716 7004      RAL
0717 7420      SNL
0720 5324      JMP NOPR
0721 4735      JMS I  NOPR+11   /PRINT ERROR
0722 4736      JMS I  NOPR+12
0723 4737      JMS I  NOPR+13

0724 7604      NOPR,  CLA OSR    /TEST SW0
0725 7104      RAL CLL
0726 7430      SEL
0727 7402      HLT
0730 7604      CLA OSR    /MQL OR MGA ERROR,
0731 7106      RTL CLL    /TEST SW1
0732 7430      SEL
0733 5252      JMP MGAT1+2     /PROGRAM LOOP
0734 5251      JMP MGAT1+1     /CONTINUE PROGRAM
0735 0000      PHQAT
0736 0473      XONE
0737 0605      MGA1
0740 3066      DCA BLXP
0741 5267      JMP MGAT1+17

0770 0523
0771 0370
0772 0347
0773 0311
0774 0362
0775 0332
0776 0323
0777 0302
1000
PAGE
1000

```

```

/TEST OF MGA,
1000 5232      MQAT2,  JMP HSE4
1001 4542      JMS I  UGEN
1002 7360      STL CLA CMA    /SET LINK
1003 0065      AND GENX
1004 7040      CMA
1005 3063      DCA ACP        /COMPLEMENT GENX PATTERN
1006 7040      CMA
1007 3064      DCA LXP        /STORE AC PATTERN
1010 1065      TAD GENX
1011 7421      MQL ACP        /STORE LINK TO A ONE
1012 1063      TAD ACP        /LOAD MQL
1013 7501      MGA
1014 3067      DCA BACP        /LOAD AC WITH COMPLEMENTED GENX
1015 7620      CLA SNL
1016 5333      JMP CLRL4        /STORE RESULT OF MGA
1017 7240      CLA CMA
1020 3066      DCA BLXP        /STORE LINK RESULT 7777

1021 1067      RL4,  TAD BACP        /AC SHOULD EQUAL 7777
1022 7040      CMA
1023 7440      SEA
1024 5242      JMP MGAER2     /MO DID NOT INCLUSIVE OR WITH AC
1025 7040      CMA
1026 0066      AND BLXP
1027 7450      SNA
1030 5242      JMP MGAER2     /LINK DID NOT EQUAL A ONE
1031 5255      JMP LNPR2+4

1032 7300      HSE4,  CLA CLL
1033 3065      DCA GENX
1034 1107      TAD XMGAT2
1035 3056      DCA BACK
1036 1110      TAD XMGAT3
1037 3057      DCA NEXT
1040 4535      JMS I  MODSEL    /PERFORM MODE SELECTION.
1041 5201      JMP MGAT2+1

1042 7604      MGAER2, CLA OSR    /TEST SW2
1043 7106      RTL CLL
1044 7004      RAL
1045 7420      SNL
1046 5251      JMP LNPR2
1047 4662      JMS I  XPMQAT
1050 4263      JMS EMGAT2

1051 7604      LNPR2, CLA OSR    /TEST SW0
1052 7104      RAL CLL

```

```

1053 7430 SEL
1054 7402 HLT /MOA ERROR,
1055 7604 CLA OSR /TEST SW1
1056 7106 RTL CLL
1057 7430 SEL
1060 5202 JMP HQAT2+2 /PROGRAM LOOP
1061 5201 JMP HQAT2+1 /CONTINUE PROGRAM

1062 0600 XPMQAT, PMQAT

1063 0800 EMQAT2, OPEN
1064 4326 JMS PTWO
1065 4576 JMS I CTYMOD /TYPE THE MODE,
1066 4524 AT3, JMS I CRLF /CARRIAGE RETURN AND LINE FEED,
1067 4455 JMS I U2SPC /2 SPACES,
1070 4454 JMS I U1SPC /1 SPACE,
1071 4777 JMS AC /PRINT "AC",
1072 4455 JMS I U2SPC /2 SPACES,
1073 4776 JMS PLXP /
1074 4454 JMS I U1SPC /1 SPACE
1075 4775 JMS PACP /
1076 4524 JMS I CRLF /CR AND LF,
1077 4455 JMS I U2SPC /2 SPACES,
1100 4454 JMS I U1SPC /1 SPACE,
1101 4774 JMS MQ /PRINT "MQ",
1102 4455 JMS I U2SPC /2 SPACES,
1103 4455 JMS I U2SPC /2 SPACES,
1104 7200 CLA /0
1105 1065 TAD GENX /
1106 3063 DCA ACP /
1107 4775 JMS PACP /
1110 4524 JMS I CRLF /CR AND LF,
1111 4774 JMS MQ /PRINT "MQ",
1112 4321 JMS VOR /
1113 4777 JMS AC /PRINT "AC",
1114 4455 JMS I U2SPC /2 SPACES,
1115 4773 JMS PBLXP /
1116 4454 JMS I U1SPC /1 SPACE,
1117 4772 JMS PBACP /
1120 5663 JMP I EMQAT2 /EXIT,

1121 0800 VOR, 0 /PRINT INCLUSIVE OR
1122 7240 CLA CMA
1123 0111 AND INCOR
1124 4526 JMS I PRXL0P /PRINT,
1125 5721 JMP I VOR

1126 0800 PTWO, 0 /PRINT 2
1127 7240 CLA CMA
1130 0113 AND TWO
1131 4526 JMS I PRXL0P /PRINT,
1132 5726 JMP I PTWO

```

```

1133 3066 CLRL4, DCA BLXP
1134 5221 JMP RL4

/TEST OF MOA,

1135 5771 HQAT3, JMP HSES
1136 4542 JMS I UGEN
1137 7340 CLL CLA CMA /CLEAR LINK
1140 0065 AND GENX
1141 7040 CMA /COMPLEMENT GENX PATTERN
1142 3063 DCA ACP /STORE AC PATTERN
1143 3064 DCA LXP /STORE LINK TO A ZERO
1144 7040 CMA
1145 0065 AND GENX
1146 7421 MQL /LOAD MQ
1147 1063 TAD ACP /LOAD AC WITH COMPLEMENTED GENX,
1150 7501 MQA
1151 3067 DCA BACP /STORE RESULT OF MOA
1152 7620 CLA SNL
1153 7410 SKP
1154 7240 CLA CMA
1155 3066 DCA BLXP /STORE LINK RESULT 7777
1156 1067 TAD BACP /AC SHOULD EQUAL 7777,
1157 7040 CMA
1160 7440 SEA
1161 5770 JMP MQAER3 /MQ DID NOT INCLUSIVE OR WITH AC
1162 7040 CMA
1163 0066 AND BLXP
1164 7440 SEA
1165 5770 JMP MQAER3 /LINK DID NOT EQUAL A ZERO
1166 5767 JMP NQPR3+4

1167 1223
1170 1210
1171 1200
1172 0370
1173 0347
1174 0302
1175 0362
1176 0332
1177 0323
1200 1200 PAGE
1200 7300 HSES, CLA CLL
1201 3065 DCA GENX
1202 1110 TAD XHQAT3
1203 3056 DCA BACK
1204 1377 TAD (SAMTS0
1205 3057 DCA NEXT
1206 4535 JMS I MOSEL /PERFORM MODE SELECTION,
1207 5776 JMP HQAT3+1

```

```

1210 7604  MOAERS, CLA OSR      /TEST SW2
1211 7106          RTL CLL
1212 7804          RAL
1213 7420          SNL
1214 5217          JMP NOPRS      /PRINT ERROR
1215 4630          JMS I APMQAT
1216 5233          JMP AMQAT3

1217 7604  NOPRS, CLA OSR      /TEST SW0
1220 7104          RAL CLL
1221 7430          SEL
1222 7402          HLT      /MOA ERROR,
1223 7604          CLA OSR      /TEST SW1
1224 7106          RTL CLL
1225 7430          SEL
1226 5775/        JMP MQAT3+2    /PROGRAM LOOP
1227 5776/        JMP MQAT3+1    /CONTINUE PROGRAM

1230 0600  APMQAT, PMQAT
1231 1217          NOPRS
1232 1063          EMQAT2

1233 4240  AMQAT3, JMS PTHREE
1234 4576          JMS I CTYMOD      /TYPE THE MODE,
1235 1231          TAD APMQAT+1
1236 3632          DCA I APMQAT+2
1237 5774/        JMP ATS

1240 0000  PTHREE, 0
1241 7240          CLA CMA
1242 0112          AND THREE
1243 4526          JMS I PRXLOP      /PRINT,
1244 5640          JMP I PTHREE
    
```

/TEST OF THE SAM INSTRUCTION USING FIXED NUMBERS,

```

1245 4315  SAMTS0, JMS SAMS0H      /GO DO HOUSEKEEPING,
1246 4263  SAM0, JMS SAMGEN      /LOAD
1247 1021          TAD TOLINK
1250 7104          CLL RAL      /LINK LOADED
1251 1023          TAD TOMQ
1252 7421          MQL      /MQ LOADED
1253 1022          TAD TOAC      /AC LOADED
1254 7457          SAM      /EAE SUBTRACT
1255 4541          JMS I SAVREG    /SAVE L,AC,MO,SC,AND GT.
1256 4773/        JMS SAMSIM      /SIMULATE "SAM"
1257 4492          JMS I UCOMP      /COMPARE ACTUAL AGAINST SIMULATED
1260 7773          =5
1261 5276          JMP ESAM0      /ERROR
    
```

```

1262 5302          JMP ESAM0+4      /NO ERROR

1263 0000  SAMGEN, OPEN
1264 4453          JMS I UMOVE      /MOVE DATA TO: TOLINK, TOAC, TOMQ,
1265 0000          OPEN
1266 0021          TOLINK
1267 7775          =3
1270 7325          CLA CLL CML IAC RAL /AC = 3
1271 1265          TAD =4
1272 3265          DCA =5
1273 2114          ISZ SCOUNT
1274 5663          JMP I SAMGEN
1275 5575          JMP I CGEN+3

/Routine to check SR options for SAM test 0.

1276 4545  ESAM0, JMS I TSTSW2    /CHECK SR 2.
1277 4305          JMS SM0ERR      /PRINT ERROR DATA,
1280 4543          JMS I TSTSW0    /CHECK SR 0.
1281 7402          HLT      /SUBTRACT AC FROM MQ ERROR, (SAM),
1282 4544          JMS I TSTSW1    /CHECK SR 1.
1283 5247          JMP SAM0+1      /LOOP THE ROUTINE,
1284 5246          JMP SAM0      /CONTINUE NORMAL TEST,

/Routine to print error information for SAM test 0.

1305 0000  SM0ERR, OPEN
1306 4534          JMS I TYTST      /TYPE THE FOLLOWING:
1307 7775          =3
1310 7924          ESAM      /SAM
1311 7440          TEST      /TEST
1312 7443          ER0      /0
1313 4537          JMS I PREGS      /PRINT HEADING AND CONTENTS OF REGISTERS,
1314 5705          JMP SM0ERR      /EXIT

/INITIALIZATION ROUTINE FOR SAM TEST 0.

1315 0000  SAMS0H, OPEN
1316 4540          JMS I ASCOMP      /HOUSEKEEPING FOR SAMTS0,
1317 1372          TAD (SAMTAB      /SET COMPARE ROUTINE
1320 3265          DCA SAMGEN+2    /GET ADDRESS OF THE TABLE
1321 1377          TAD (SAMTS0      /AND STORE IT AT SAMGEN+2
1322 3056          DCA BACK      /BACK SET TO RETURN TO CURRENT TEST
1323 1371          TAD (SAMTS1
1324 3057          DCA NEXT
1325 1370          TAD (=14
1326 3114          DCA SCOUNT      /NUMBER OF TESTS+1,
1327 4535          JMS I MODSEL      /PERFORM MODE SELECTION,
1330 4936          JMS I ONLYB      /EXIT TEST IF 1 MODE,
1331 7403          ACS      /CLEAR THE STEP COUNTER,
1332 5715          JMP I SAMS0H      /EXIT,

/TEST OF THE SAM INSTRUCTION USING RANDOM NUMBERS,
    
```

```

1333 4767/ SAMTS1, JMS SAMS1H /GO DO HOUSEKEEPING
1334 4592 SAM1, JMS I RNDATA /LOAD WITH RANDOM
1335 1023 TAD TOMQ
1336 7421 MQL /MQ LOADED
1337 4593 JMS I LDGT /LOAD THE GT ACCORDING TO "TOGT";
1340 4596 JMS I LDSC /LOAD THE SC ACCORDING TO "TOSHIF";
1341 1021 TAD TOLINK
1342 7104 CLL RAL /LINK LOADED;
1343 1022 TAD TOAC /AC LOADED
1344 7497 SAM /EAE SUBTRACT AC FROM MQ
1345 4541 JMS I SAVREG /SAVE L,AC,MO,SC, AND GT,
1346 4773/ JMS SAMSIM /SIMULATE "SAM"
1347 4492 JMS I UCOMP /COMPARE ACTUAL AGAINST SIMULATED
1350 7773 =5 /L, AC, MQ, AND SC;
1351 5766/ JMP ESAM1 /ERROR
1352 5769/ JMP ESAM1+4 /NO ERROR OCCURRED;

1365 1415
1366 1411
1367 1400
1370 7764
1371 1333
1372 7244
1373 6013
1374 1066
1375 1137
1376 1136
1377 1245
1400
    
```

PAGE

/INITIALIZATION ROUTINE FOR SAM TEST 1,

```

1400 0000 SAMS1H, OPEN
1401 4540 JMS I ASCOMP /SET COMPARE ROUTINE;
1402 1377 TAD (SHLTS0 /ADDRESS OF THE
1403 3097 DCA NEXT /NEXT TEST TO "NEXT"
1404 1376 TAD (SAMTS1 /BACK SET TO
1405 3096 DCA BACK /RETURN TO CURRENT TEST;
1406 4535 JMS I MODSEL /PERFORM MODE SELECTION;
1407 4536 JMS I ONLYB /EXIT TEST IF MODE "A";
1410 5600 JMP I SAMS1H /EXIT,
    
```

/ROUTINE TO CHECK SR OPTIONS FOR SAM TEST 1

```

1411 4545 ESAM1, JMS I TSTSW2 /CHECK SR 2,
1412 4220 JMS SM1ERR /PRINT ERROR DATA,
1413 4543 JMS I TSTSW0 /CHECK SR 0,
1414 7402 HLT /SUBTRACT AC FROM MQ ERROR; (SAM),
1415 4544 JMS I TSTSW1 /CHECK SR 1,
1416 5775/ JMP SAM1+1 /LOOP THE ROUTINE;
1417 5774/ JMP SAM1 /CONTINUE NORMAL TEST,
    
```

/ROUTINE TO PRINT ERROR INFORMATION FOR SAM TEST 1;

```

1420 0000 SM1ERR, OPEN
1421 4534 JMS I TYTST /TYPE THE FOLLOWING;
1422 7775 =3
1423 7524 ZSAM /SAM
1424 7440 TEST /TEST
1425 7445 ZONE /1
1426 4537 JMS I PREGS /PRINT HEADING AND CONTENTS OF REGISTERS-
1427 5620 JMP I SM1ERR /EXIT,
    
```

/TEST OF THE SHIFT LEFT INSTRUCTION USING AN
/INCREMENTING PATTERN IN THE MQ WITH THE AC 0
/AND SHIFTING EACH PATTERN 0-37 OCTAL SHIFTS;

```

1430 4293 SHLTS0, JMS SLTS0H /GO DO HOUSE KEEPING
1431 4542 SHL0, JMS I UGEN /GENERATE A NUMBER AND STORE IT IN GENX
1432 7331 CLA CLL CML IAC PAR /AC=4000,L=1
1433 3021 DCA TOLINK /SAVE LINK
1434 1065 TAD GENX /GET THE GENERATED NUMBER
1435 3023 DCA TOMQ /SAVE FOR MQ
1436 3022 DCA TOAC /0 FOR AC
1437 1244 TAD NBSHL0 /GET NUMBER OF SHIFTS
1440 3024 DCA TOSHIF /SAVE NUMBER OF SHIFTS
1441 1065 TAD GENX /GET THE GENERATED NUMBER
1442 7421 MQL /LOAD THE MQ;
1443 7413 SHL /EAE SHIFT LEFT
1444 0000 NBSHL0, OPEN /SHIFT THIS AMOUNT OF TIMES,
1445 4541 JMS I SAVREG /SAVE L,AC,MO,SC,GT,
1446 4773/ JMS SHLSIM /SIMULATE SHL;
1447 4492 JMS I UCOMP /COMPARE SIMULATED SHL AGAINST ACTUAL SHL;
1450 7773 =5 /L, AC, MQ, GT, AND SC;
1451 5274 JMP SBERR /SIMULATED AND ACTUAL DID NOT COMPARE;
1452 5300 JMP SBERR+4 /SIMULATED AND ACTUAL COMPARED, CONTINUE TEST;
    
```

/INITIALIZATION SUBROUTINE FOR SHLTS0,

```

1453 0000 SLTS0H, OPEN /HOUSE KEEPING
1454 4540 JMS I ASCOMP /SET COMPARE ROUTINE;
1455 3065 DCA GENX /ZERO TO NUMBER GENERATOR
1456 3244 DCA NBSHL0 /ZERO TO LOCATION CONTAINS SHIFTS,
1457 1372 TAD (SHL0
1460 3096 DCA BACK
1461 1371 TAD (SBINC
1462 3097 DCA NEXT
1463 1174 TAD C=37
1464 3114 DCA SCOUNT
1465 4535 JMS I MODSEL /PERFORM MODE SELECTION;
1466 5653 JMP I SLTS0H /EXIT, AC=0,
    
```

/ROUTINE TO INCREMENT SHIFT COUNT FOR SHL TEST 0;

```

1467 2244 SBINC, ISZ NBSHL0 /INCREMENT SHIFT COUNT;
1470 2114 ISZ SCOUNT /DONE SHIFTING
    
```

```

1471 5231      JMP      SHL0
1472 5673      JMP I   ,+1
1473 1600      SHLTS1

/Routine to check SR options for SHL test 0.

1474 4545      S0ERR, JMS I   TSTSW2      /CHECK SR 2.
1475 4303      JMS      S0ERR1      /PRINT ERROR DATA.
1476 4543      JMS I   TSTSW0      /CHECK SR 0.
1477 7402      HLT          /SHL ERROR.
1500 4544      JMS I   TSTSW1      /CHECK SR 1.
1501 5232      JMP      SHL0+1      /LOOP THE ROUTINE.
1502 5231      JMP      SHL0          /CONTINUE NORMAL TEST.
    
```

```

/Routine to print error information for SHL test 0.

1503 0000      S0ERR1, OPEN
1504 4534      JMS I   TYTST      /TYPE THE FOLLOWING
1505 7775      JMS      -3
1506 7435      ZSHL          /SHL
1507 7440      TEST         /TEST
1510 7443      ZER0          /0
1511 4547      JMS I   NUMSHF      /NUMBER OF SHIFTS IN DECIMAL.
1512 4537      JMS I   PREGS      /HEADING AND REGISTERS.
1513 5703      JMP I   S0ERR1      /EXIT, AC=0.

1571 1467
1572 1431
1573 6042
1574 1334
1575 1335
1576 1333
1577 1430
1600          PAGE
    
```

```

/TEST OF THE SHIFT LEFT INSTRUCTION USING RANDOM DATA.

1600 4216      SHLTS1, JMS      SLTS1H      /GO DO HOUSE KEEPING.
1601 4552      SHL1,  JMS I   RNDATA      /GENERATE RANDOM DATA.
1602 4551      JMS I   LDREG          /LOAD L, MQ, AND GT.
1603 1024      TAD      TOSHIF        /NUMBER OF SHIFTS.
1604 3207      DCA      NBSHL1       /LOAD THE NUMBER OF SHIFTS TO BE DONE.
1605 1022      TAD      TOAC          /AC LOADED.
1606 7413      SHL          /EAE SHIFT LEFT.
1607 0000      NBSHL1, OPEN
1608 4541      JMS I   SAVREG          /SAVE L, AC, MQ, SC, GT.
1609 4777      JMS      SHLSIM        /SIMULATE SHL
1610 4452      JMS I   UCOMP          /COMPARE SIMULATED AGAINST THE ACTUAL.
1611 7773      JMS      -5
1612 5226      JMP      S1ERR         /ERROR
1613 5232      JMP      S1ERR+4       /NO ERRORS ENCOUNTERED.

/INITIALIZATION SUBROUTINE FOR SHLTS1.
    
```

```

1616 0000      SLTS1H, OPEN          /HOUSE KEEPING
1617 4540      JMS I   ASCOMP        /SET COMPARE ROUTINE AND CLEAR TABLE.
1620 1376      TAD      (SHLTS1
1621 3056      DCA      BACK
1622 1375      TAD      (LSRTO
1623 3057      DCA      NEXT
1624 4535      JMS I   MODSEL        /PERFORM MODE SELECTION.
1625 5616      JMP I   SLTS1H         /EXIT, AC=0.
    
```

```

/Routine to check SR options for SHL test 0.

1626 4545      S1ERR, JMS I   TSTSW2      /CHECK SR 2.
1627 4235      JMS      S1ERR1      /PRINT ERROR DATA.
1630 4543      JMS I   TSTSW0      /CHECK SR 0.
1631 7402      HLT          /SHL ERROR.
1632 4544      JMS I   TSTSW1      /CHECK SR 1.
1633 5202      JMP      SHL1+1      /LOOP THE ROUTINE.
1634 5201      JMP      SHL1          /CONTINUE NORMAL TEST.
    
```

```

/Routine to print error information for SHL test 0.

1635 0000      S1ERR1, OPEN
1636 4534      JMS I   TYTST      /TYPE THE FOLLOWING.
1637 7775      JMS      -3
1640 7435      ZSHL          /SHL
1641 7440      TEST         /TEST
1642 7445      ZONE          /1
1643 4547      JMS I   NUMSHF      /NUMBER OF SHIFTS IN DECIMAL
1644 4537      JMS I   PREGS      /HEADING AND REGISTERS
1645 5635      JMP I   S1ERR1      /EXIT, AC=0
    
```

```

/TEST OF THE LOGICAL SHIFT RIGHT INSTRUCTION, (LSR),
/USING A INCREMENTING PATTERN FROM THE
/MOST SIGNIFICANT TO LEAST SIGNIFICANT WITH THE
/MQ=0 AND SHIFTING EACH PATTERN 0=37 OCTAL
/SHIFTS.

1646 4272      LSRTO0, JMS      LRSO0H      /GO DO HOUSE KEEPING
1647 4542      LSR0,  JMS I   UGEN          /GENERATE A NUMBER
1650 1065      TAD      GENX           /GET THE NUMBER
1651 4774      JMS      OBVERS        /CHANGE IT TO THE OBVERSE
1652 3022      DCA      TOAC          /FOR THE AC
1653 3023      DCA      TOMQ          /0 FOR MQ.
1654 1203      TAD      NBLSR0
1655 3024      DCA      TOSHIF
1656 7331      CLA CLL CML IAC RAR      /L=1, AC=4000
1657 3021      DCA      TOLINK        /TOLINK=4000
1660 7421      MQL          /MQ=0
1661 1022      TAD      TOAC          /AC LOADED.
1662 7417      LSR          /EAE LOGICAL SHIFT RIGHT.
    
```



```

1663 0000 NBLSR0, OPEN /DATA TO STEP COUNTER;
1664 4541 JMS I SAVREG /SAVE L, AC, MQ, SC, GT;
1665 4773 JMS LRSIM /SIMULATE LSR
1666 4452 JMS I UCOMP /COMPARE SIMULATED AGAINST ACTUAL,
1667 7773 =5 /L, AC, MQ, GT, AND SC;
1670 5313 JMP L0ERR /ERROR
1671 5317 JMP L0ERR+4 /NO ERRORS ENCOUNTERED

/INITIALIZATION SUBROUTINE FOR LSR0;

1672 0000 LRSR0H, OPEN
1673 4540 JMS I ASCOMP /SET COMPARE ROUTINE;
1674 3065 DCA GENX /ZERO TO NUMBER GENERATOR
1675 3263 DCA NBLSR0 /ZERO TO LOCATION CONTAINING SHIFTS
1676 1372 TAD (LSR0
1677 3056 DCA BACK
1678 1371 TAD (L0INC
1679 3057 DCA NEXT
1680 1174 TAD (=37
1681 3114 DCA SCOUNT
1684 4535 JMS I MODSEL /PERFORM MODE SELECTION;
1685 5672 JMP I LRSR0H /EXIT, AC=0;

```

/ROUTINE TO INCREMENT SHIFT COUNT FOR LSR TEST 0;

```

1706 2263 L0INC, ISZ NBLSR0
1707 2114 ISZ SCOUNT
1710 5247 JMP LSR0
1711 5712 JMP I ,+1
1712 2000 LSR0S1

```

/ROUTINE TO CHECK SR OPTIONS FOR LSR TEST 0;

```

1713 4545 L0ERR, JMS I TSTSW2 /CHECK SR 2,
1714 4322 JMS L0ERR1 /PRINT ERROR DATA;
1715 4543 JMS I TSTSW0 /CHECK SR 0,
1716 7402 HLT /LSR ERROR;
1717 4544 JMS I TSTSW1 /CHECK SR 1,
1720 5250 JMP LSR0+1 /LOOP THE ROUTINE;
1721 5247 JMP LSR0 /CONTINUE NORMAL TEST;

```

/ROUTINE TO PRINT ERROR INFORMATION FOR LSR TEST 0;

```

1722 0000 L0ERR1, OPEN
1723 4534 JMS I TYTST /TYPE THE FOLLOWING
1724 7775 =3
1725 7453 ZLSR /LSR
1726 7440 TEST /TEST
1727 7443 ZER0 /0
1730 4547 JMS I NUMSHF /NUMBER OF SHIFTS IN DECIMAL
1731 4537 JMS I PREGS
1732 5722 JMP I L0ERR1 /EXIT

```

```

1771 1706
1772 1647
1773 6120
1774 6473
1775 1646
1776 1600
1777 6042
2000 PAGE

```

/TEST OF THE LOGICAL SHIFT RIGHT INSTRUCTION USING RANDOM DATA;

```

2000 4216 LSRTS1, JMS LRS1H /GO DO HOUSE KEEPING
2001 4952 LSRTS1, JMS I RNDATA /GENERATE RANDOM DATA;
2002 4551 JMS I LDREG /LOAD L, MQ, AND GT;
2003 1024 TAD TOSHIF /
2004 3207 DCA NBLSR1 /NUMBER OF SHIFTS;
2005 1022 TAD TOAC /AC LOADED;
2006 7417 LSR /LOGICAL SHIFT RIGHT;
2007 0000 NBLSR1, OPEN /NUMBER OF SHIFTS TO BE PERFORMED;
2010 4541 JMS I SAVREG /SAVE L, AC, MQ, SC, GT;
2011 4777 JMS LRSIM /SIMULATE LSR
2012 4452 JMS I UCOMP /CHECK SIMULATED AGAINST ACTUAL
2013 7773 =5 /L, AC, MQ, GT, AND SC;
2014 5226 JMP L1ERR /ERROR
2015 5232 JMP L1ERR+4 /NO ERRORS ENCOUNTERED;

```

/INITIALIZATION SUBROUTINE FOR LSRTS1

```

2016 0000 LRS1H, OPEN
2017 4540 JMS I ASCOMP /SET COMPARE ROUTINE;
2020 1376 TAD (LSRTS1
2021 3056 DCA BACK
2022 1375 TAD (ASRTS0
2023 3057 DCA NEXT
2024 4535 JMS I MODSEL /PERFORM MODE SELECTION;
2025 5616 JMP I LRS1H /EXIT, AC=0;

```

/ROUTINE TO CHECK SR OPTIONS FOR LSR TEST 1.

```

2026 4545 L1ERR, JMS I TSTSW2 /CHECK SR 2,
2027 4235 JMS L1ERR1 /PRINT ERROR DATA;
2030 4543 JMS I TSTSW0 /CHECK SR 0,
2031 7402 HLT /LSR ERROR;
2032 4544 JMS I TSTSW1 /CHECK SR 1,
2033 5202 JMP LSR1+1 /LOOP THE ROUTINE;
2034 5201 JMP LSR1 /CONTINUE NORMAL TEST;

```

/ROUTINE TO PRINT ERROR INFORMATION

```

2035 0000 L1ERR1, OPEN

```

```

2036 4534 JMS I TYTST /PRINT THE FOLLOWING:
2037 7775 -3 /LSR
2040 7453 ZLSR /LSR
2041 7440 TEST /TEST
2042 7445 ZONE /1
2043 4547 JMS I NUMSHF /NUMBER OF SHIFTS IN DECIMAL
2044 4537 JMS I PREGS /HEADING AND REGISTERS
2045 5635 JMP I L1ERR1 /EXIT
    
```

/TEST OF THE ARITHMETIC SHIFT RIGHT INSTRUCTION,
 /USING AN INCREMENTING PATTERN FROM THE
 /MOST SIGNIFICANT TO LEAST SIGNIFICANT WITH THE
 /NO ZERO AND SHIFTING EACH PATTERN 0-37 OCTAL
 /SHIFTS.

```

2046 4272 ASRTS0, JMS ASRS0H /DO INITIALIZATION
2047 4542 ASR0, JMS I UGEN /GENERATE A COUNT
2050 1065 TAD GENX /GET THE NUMBER
2051 4774 JMS OBVERS /CHANGE IT TO THE OBVERSE
2052 3022 DCA TOAC /COUNT PATTERN TO TOAC
2053 3023 DCA TOMQ /0 TO TOMQ
2054 1263 TAD NBASR0
2055 3024 DCA TOSHIF
2056 7331 CLA CLL CML IAC RAR /L*1
2057 3021 DCA TOLINK /1 TO TOLINK
2060 7421 MQL /0 TO MQ
2061 1022 TAD TOAC /AC LOADED
2062 7415 ASR /EAE ARITHMETIC SHIFT RIGHT
2063 0000 NBASR0, OPEN
2064 4541 JMS I SAVREG /SAVE L, AC, MQ, SC, GT
2065 4773 JMS ASRSIM /SIMULATE AN ASR
2066 4452 JMS I UCOMP /COMPARE ACTUAL AGAINST SIMULATED
2067 7773 =5 /L, AC, MQ, GT, AND SC
2070 5313 JMP ABERR /ERROR DETECTED
2071 5317 JMP ABERR+4 /NO ERROR ENCOUNTERED
    
```

/INITIALIZATION SUBROUTINE FOR ASRTS0,

```

2072 0000 ASRS0H, OPEN
2073 4540 JMS I ASCOMP /SET COMPARE ROUTINE
2074 3065 DCA GENX /ZERO TO NUMBER GENERATOR
2075 3263 DCA NBASR0 /ZERO TO LOCATION CONTAINING SHIFTS
2076 1372 TAD (ASR0
2077 3056 DCA BACK
2080 1371 TAD (ABINC
2081 3057 DCA NEXT
2082 1174 TAD [-37 /SET UP FOR
2083 3114 DCA SCOUNT /37 OCTAL SHIFTS
2084 4535 JMS I MODSEL /PERFORM MODE SELECTION
2085 5672 JMP I ASRS0H /EXIT, AC=0
    
```

/ROUTINE TO INCREMENT SHIFT COUNT FOR ASR TEST 0,

```

2106 2263 ABINC, ISZ NBASR0
    
```

```

2107 2114 ISZ SCOUNT /DONE THIS TEST YET?
2110 5247 JMP ASR0 /NO
2111 5712 JMP I ,+1 /GO TO NEXT TEST
2112 2200 ASRTS1
    
```

/ROUTINE TO CHECK SR OPTIONS FOR ASR TEST 0,

```

2113 4545 ABERR, JMS I TSTSW2 /CHECK SR 2
2114 4322 JMS ABERR1 /PRINT ERROR DATA
2115 4543 JMS I TSTSW0 /CHECK SR 0
2116 7402 HLT /ASR ERROR
2117 4544 JMS I TSTSW1 /CHECK SR 1
2120 5250 JMP ASR0+1 /LOOP THE ROUTINE
2121 5247 JMP ASR0 /CONTINUE NORMAL TEST
    
```

/ROUTINE TO PRINT ERROR INFORMATION FOR ASR TEST 0,

```

2122 0000 ABERR1, OPEN
2123 4534 JMS I TYTST /PRINT THE FOLLOWING
2124 7775 -3 /LSR
2125 7462 ZASR /ASR
2126 7440 TEST /TEST
2127 7443 ZER0 /0
2130 4547 JMS I NUMSHF /NUMBER OF SHIFTS IN DECIMAL
2131 4537 JMS I PREGS /HEADING AND REGISTERS
2132 5722 JMP I ABERR1 /EXIT, AC=0
    
```

2171 2106
 2172 2047
 2173 6200
 2174 6473
 2175 2046
 2176 2000
 2177 6120
 2200

PAGE

/TEST OF THE ARITHMETIC SHIFT RIGHT INSTRUCTION
 /USING RANDOM DATA:

```

2200 4216 ASRTS1, JMS ASRS1H /GO DO HOUSEKEEPING
2201 4552 ASR1, JMS I RNDATA /GENERATE RANDOM DATA
2202 4551 JMS I LDREG /LOAD L, MQ, AND GT
2203 1024 TAD TOSHIF
2204 3207 DCA NBASR1 /NUMBER OF SHIFTS LOADED
2205 1022 TAD TOAC /AC LOADED
2206 7415 ASR /EAE ARITHMETIC SHIFT RIGHT
2207 0000 NBASR1, OPEN
2210 4541 JMS I SAVREG /SAVE L, AC, MQ, SC, GT
2211 4773 JMS ASRSIM /SIMULATE ASR
2212 4452 JMS I UCOMP /COMPARE SIMULATED AGAINST ACTUAL
2213 7773 =5
2214 5226 JMP A1ERR /ERROR DETECTED
2215 5232 JMP A1ERR+4 /NO ERRORS ENCOUNTERED
    
```

```

/INITIALIZATION SUBROUTINE FOR ASRST1,
2216 0000 ASRS1H, OPEN
2217 4540 JMS I ASCOMP /SET COMPARE ROUTINE,
2220 1376 TAD (ASRST1
2221 3056 DCA BACK
2222 1375 TAD (DPSES0
2223 3057 DCA NEXT
2224 4535 JMS I MODSEL /PERFORM MODE SELECTION,
2225 5616 JMP I ASRS1H /EXIT, AC=0.

/ROUTINE TO CHECK SR OPTIONS IN ASR TEST 1,
2226 4545 A1ERR, JMS I TSTSW2 /CHECK SR 2,
2227 4235 JMS A1ERR1 /PRINT ERROR DATA,
2230 4543 JMS I TSTSW0 /CHECK SR 0,
2231 7402 HLT /ASR ERROR,
2232 4544 JMS I TSTSW1 /CHECK SR 1,
2233 5202 JMP ASR1+1 /LOOP THE ROUTINE,
2234 5201 JMP ASR1 /CONTINUE NORMAL TEST,

/ROUTINE TO PRINT ERROR INFORMATION,
2235 0000 A1ERR1, OPEN /PRINT THE FOLLOWING:
2236 4534 JMS I TYTST
2237 7775 -3
2240 7462 ZASR /ASR
2241 7440 TEST /TEST
2242 7445 ZONE /1
2243 4547 JMS I NUMSHF /NUMBER OF SHIFTS IN DECIMAL
2244 4537 JMS I PREGS /HEADING AND REGISTERS
2245 5635 JMP I A1ERR1 /EXIT

/TEST OF THE DOUBLE PRECISION SKIP IF ZERO INSTRUCTION, (DPSE),
2246 4774 DPSE0, JMS DPSE0H /GO DO HOUSE KEEPING,
2247 7320 CLA CLL CML
2250 4773 DPSE0, JMS ROTGEN
2251 7300 CLA CLL
2252 1044 TAD LSH /GET DATA THAT WILL BE PLACED IN THE MQ,
2253 1043 TAD MSH /ADD THE AC DATA TO THAT,
2254 7650 SNA CLA /WOULD THE AC AND MQ BE ZERO?
2255 7430 SEL /CHECK FOR A CARRY WHEN AC AND MQ ARE ADDED,
2256 4302 JMS NOSKIP /AC AND MQ WILL BE NON ZERO,
2257 4313 JMS YSKIP /AC AND MQ WILL BE ZERO,
2260 7331 CLA CLL CML IAC RAR /AC = 0000; LINK = 1
2261 3042 DCA LSIM /LOAD THE SIMULATED LINK
2262 1042 TAD LSIM
2263 3021 DCA TOLINK /LOAD THE PROBLEM LINK,
2264 1044 TAD LSH
2265 7421 MQL /MQ NOW LOADED,
2266 1043 TAD MSH /AC NOW LOADED,
2267 7451 DPSE /EAE DOUBLE PRECISION SKIP IF ZERO
2270 0000 NOSKP, OPEN
2271 0000 YESSKP, OPEN
    
```

```

2272 4541 CKDATA, JMS I SAVREG /SAVE L,AC,MQ,SC,GT,
2273 4452 JMS I UCOMP /COMPARE L,AC,MQ
2274 7775 -3
2275 7610 SKP CLA /DPSE MODIFIED A REGISTER,
2276 5772 JMP EDPSZ0+5 /NO ERROR OCCURED,
2277 1371 TAD (DATER /SET PRINTOUT FOR DATA ERROR
2300 3770 DCA DPZPR1
2301 5767 JMP EDPSZ0+1 /DPSE MODIFIED ONE OF THE REGISTERS,

/ROUTINE TO SETUP FOR NO SKIP CONDITION
2302 0000 NOSKIP, OPEN /AC AND MQ NOT 0
2303 1366 TAD (JMP CKDATA
2304 3270 DCA NOSKP
2305 1364 TAD (JMP EDPSZ0
2306 3271 DCA YESSKP
2307 1363 TAD (SO
2310 3770 DCA DPZPR1
2311 2302 ISZ NOSKIP
2312 5702 JMP I NOSKIP /EXIT, AC=0

/ROUTINE TO SET UP FOR A SKIP CONDITION
2313 0000 YSKIP, OPEN /AC AND MQ = 0
2314 1366 TAD (JMP CKDATA
2315 3271 DCA YESSKP
2316 1364 TAD (JMP EDPSZ0
2317 3270 DCA NOSKP
2320 1362 TAD (NSO
2321 3770 DCA DPZPR1
2322 5713 JMP I YSKIP /EXIT, AC = 0,

2362 7554
2363 7545
2364 5765
2365 2512
2366 5272
2367 2513
2370 5544
2371 7565
2372 2517
2373 2476
2374 2400
2375 2246
2376 2200
2377 6200
    
```

PAGE

/INITIALIZATION ROUTINE FOR DPSE0,

```

2400 0000 DPSE0H, OPEN
2401 4540 JMS I ASCOMP /SET COMPARE ROUTINE,
2402 1377 TAD (DPSES0
2403 3056 DCA BACK
    
```

```

2404 1376      TAD  (DZINC
2405 3057      DCA  NEXT
2406 1775      TAD  DSEJMS
2407 3774      DCA  PRCHG      /SET PRINTOUT ROUTINE FOR DPSZ TEST,
2410 7344      CLA  CLL  CMA  RAL  /7776
2411 3273      DCA  CNTR1
2412 7344      CLA  CLL  CMA  RAL  /7776
2413 3274      DCA  CNTR2
2414 7344      CLA  CLL  CMA  RAL  /7776
2415 3275      DCA  CNTR3
2416 1373      TAD  (=32
2417 3114      DCA  SCOUNT
2420 4535      JMS  I  MOOSEL      /PERFORM MODE SELECTION,
2421 1115      TAD  MODE
2422 7700      SMA  CLA
2423 5264      JMP  MODA      /EXIT IF MODE 1
2424 7403      ACS
2425 5600      JMP  I  DPSZ0H      /CLEAR THE STEP COUNTER,
                                          /EXIT, AC=0;
    
```

```

2426 2114      DZINC, ISZ  SCOUNT
2427 5772      JMP  DPSZ0+1
2430 7340      CLA  CMA  CLL
2431 3114      DCA  SCOUNT      /SET SCOUNT TO 7777 SO THE ROTGEN IS NOT USED,
2432 7240      CLA  CMA
2433 3043      DCA  MSH      /7777
2434 3044      DCA  LSH      /0000
2435 2273      ISZ  CNTR1
2436 5772      JMP  DPSZ0+1
2437 7240      CLA  CMA
2440 3114      DCA  SCOUNT      /SET SCOUNT TO 7777 SO THE ROTGEN IS NOT USED,
2441 7240      CLA  CMA
2442 3273      DCA  CNTR1      /SET CNTR1 TO 7777 SO AC=7777 + MQ=0 TEST IS NOT USED,
2443 7240      CLA  CMA
2444 3044      DCA  LSH      /7777
2445 3043      DCA  MSH      /0000
2446 2274      ISZ  CNTR2
2447 5772      JMP  DPSZ0+1
2450 7240      CLA  CMA
2451 3114      DCA  SCOUNT      /SET SCOUNT TO 7777 SO THE ROTGEN IS NOT USED AGAIN,
2452 7040      CMA
2453 3273      DCA  CNTR1      /SET CNTR1 TO 7777 SO AC=7777+MQ=0 TEST IS NOT USED AGAIN,
2454 7040      CMA
2455 3274      DCA  CNTR2      /SET CNTR2 TO 7777 SO AC=0 + MQ=7777 TEST IS NOT USED AGAIN,
2456 7040      CMA
2457 3044      DCA  LSH      /7777
2460 7040      CMA
2461 3043      DCA  MSH      /7777
2462 2275      ISZ  CNTR3
2463 5772      JMP  DPSZ0+1
2464 7004      MODA, LAS
2465 7006      RTL
2466 7004      RAL
    
```

```

2467 7710      SPA  CLA      /SR 3 SET?
2470 5777      JMP  DPSZ0      /YES I! IS; REPEAT TEST,
2471 5672      JMP  I  ,+1
2472 2600      DPITS0
2473 0000      CNTR1, OPEN
2474 0000      CNTR2, OPEN
2475 0000      CNTR3, OPEN
                /ROUTINE TO GENERATE A ROTATING BIT THROUGH THE MQ AND AC;
2476 0000      ROTGEN, OPEN      /GENERATE ROTATING PATTERN
2477 1044      TAD  LSH
2500 7004      RAL
2501 3044      DCA  LSH
2502 1043      TAD  MSH
2503 7004      RAL
2504 3043      DCA  MSH
2505 1043      TAD  MSH
2506 3022      DCA  TOAC
2507 1044      TAD  LSH
2510 3023      DCA  TOMQ
2511 5226      JMP  DZINC      /EXIT, AC=0;
                /ROUTINE TO CHECK SR OPTIONS FOR DPSZ TEST 0;
2512 4541      EDPZ0, JMS  I  SAVREG      /SAVE L,AC,MQ,SC,GT,
2513 4545      JMS  I  TSTSW2      /CHECK SR 2,
2514 4323      JMS  DZERR0      /PRINT ERROR DATA,
2515 4543      JMS  I  TSTSW0      /CHECK SR 0,
2516 7402      HLT
2517 4544      JMS  I  TSTSW1      /DPSZ ERROR,
2520 5772      JMP  DPSZ0+1      /CHECK SR 1,
2521 7100      CLL
2522 5771      JMP  DPSZ0      /LOOP THE ROUTINE,
                                          /CONTINUE NORMAL TEST;
                /ROUTINE TO PRINT ERROR INFORMATION,
2523 0000      DZERR0, OPEN
2524 4534      JMS  I  TYTST
2525 7775      -3
2526 7465      EDPSE
2527 7440      TEST
2530 7443      ZERR0
2531 4537      JMS  I  PREGS
2532 5723      JMP  I  DZERR0      /EXIT
2571 2250
2572 2251
2573 7746
2574 7002
2575 7045
2576 2426
2577 2246
    
```

2600

PAGE

/TEST OF THE DOUBLE PRECISION INCREMENT INSTRUCTION, (DPIC).

```

2600 4221 DPITS0, JMS DPIS0H /GO DO HOUSE KEEPING,
2601 4542 DPI0, JMS I UGEN /GENERATE NUMBERS,
2602 7240 CLA CMA /7777
2603 3022 DCA TOAC /SIMULATED AC = 7777,
2604 3021 DCA TOLINK /SIMULATED LINK = 0,
2605 1065 TAD GENX /GET THE NUMBER GENERATED BY "GEN",
2606 7421 MQL /MQ LOADED,
2607 7701 CLA MQA /MQ TO AC,
2610 3023 DCA TOMQ /SIMULATED MQ = C(GENX),
2611 7240 CLA CMA /AC = 7777
2612 7573 DPIC /DOUBLE PRECISION INCREMENT
2613 4541 JMS I SAVREG /SAVE L,AC,MQ,SC,GT,
2614 4777 JMS DPISIM /SIMULATE DPI,
2615 4452 JMS I UCOMP /COMPARE SIMULATED AGAINST ACTUAL,
2616 7775 -3 /L,AC,MQ,
2617 5234 JMP EDPI0 /ERROR
2620 5240 JMP EDPI0+4

```

/INITIALIZATION ROUTINE FOR DPIC TEST 0.

```

2621 0000 DPIS0H, OPEN /HOUSE KEEPING OF DPITS0,
2622 4540 JMS I ASCOMP /SET COMPARE ROUTINE,
2623 3065 DCA GENX
2624 1376 TAD (DPI0
2625 3056 DCA BACK
2626 1375 TAD (DPITS1
2627 3057 DCA NEXT
2630 4535 JMS I MODSEL /PERFORM MODE SELECTION,
2631 4536 JMS I ONLYB /EXIT IF MODE "A",
2632 7403 ACS /CLEAR THE STEP COUNTER,
2633 5021 JMP I DPIS0H /EXIT

```

/ROUTINE TO CHECK SR OPTIONS FOR DPIC TEST 0.

```

2634 4545 EDPI0, JMS I TSTSW2 /CHECK SR 2,
2635 4243 JMS DI0ERR /PRINT ERROR DATA,
2636 4543 JMS I TSTSW0 /CHECK SR 0,
2637 7402 HLT /DPIC ERROR,
2640 4544 JMS I TSTSW1 /CHECK SR 1,
2641 5202 JMP DPI0+1 /LOOP THE ROUTINE,
2642 5201 JMP DPI0 /CONTINUE NORMAL TEST,

```

/ROUTINE TO PRINT ERROR INFORMATION.

```

2643 0000 DI0ERR, OPEN
2644 4534 JMS I TYTST /PRINT THE FOLLOWING:
2645 7775 -3
2646 7470 ZDPIC /DPIC
2647 7440 TEST /TEST

```

```

2650 7443 ZERO /0
2651 4537 JMS I PREGS /HEADING AND REGISTERS,
2652 5643 JMP I DI0ERR /EXIT, AC=0,

```

/TEST OF THE DOUBLE PRECISION INCREMENT INSTRUCTION,
/USING RANDOM DATA.

```

2653 4267 DPITS1, JMS DPIS1H /GO DO HOUSEKEEPING
2654 4552 DPI1, JMS I RNDATA /GENERATE RANDOM DATA,
2655 4556 JMS I LDSC /LOAD THE STEP COUNTER,
2656 4551 JMS I LDREG /LOAD L, MQ, AND GT,
2657 1022 TAD TOAC /AC LOADED,
2660 7573 DPIC /EAE DOUBLE PRECISION INCREMENT,
2661 4541 JMS I SAVREG /SAVE L,AC,MQ,SC,GT,
2662 4777 JMS DPISIM /SIMULATE DPIC,
2663 4452 JMS I UCOMP /COMPARE SIMULATED AGAINST ACTUAL
2664 7773 -5 /L,AC,MQ,GT, AD SC,
2665 5300 JMP EDPI1 /ERROR
2666 5304 JMP EDPI1+4 /NO ERRORS ENCOUNTERED

```

/INITIALIZATION ROUTINE FOR DPIC TEST 1.

```

2667 0000 DPIS1H, OPEN /HOUSEKEEPING FOR DPIS1
2670 4540 JMS I ASCOMP /SET COMPARE ROUTINE,
2671 1375 TAD (DPITS1
2672 3056 DCA BACK
2673 1374 TAD (DCMTS0
2674 3057 DCA NEXT
2675 4535 JMS I MODSEL /PERFORM MODE SELECTION,
2676 4536 JMS I ONLYB /EXIT IF MODE "A",
2677 5667 JMP I DPIS1H /EXIT

```

/ROUTINE TO CHECK SR OPTIONS IN DPIC TEST 1.

```

2700 4545 EDPI1, JMS I TSTSW2 /CHECK SR 2,
2701 4307 JMS DI1ERR /PRINT ERROR DATA,
2702 4543 JMS I TSTSW0 /CHECK SR 0,
2703 7402 HLT /DPIC ERROR,
2704 4544 JMS I TSTSW1 /CHECK SR 1,
2705 5255 JMP DPI1+1 /LOOP THE ROUTINE,
2706 5254 JMP DPI1 /CONTINUE NORMAL TEST,

```

/ROUTINE TO PRINT ERROR INFORMATION.

```

2707 0000 DI1ERR, OPEN /PRINT THE FOLLOWING:
2710 4534 JMS I TYTST
2711 7775 -3
2712 7470 ZDPIC /DPIC
2713 7440 TEST /TEST
2714 7445 EONE /1
2715 4537 JMS I PREGS /HEADING AND REGISTERS,
2716 5707 JMP I DI1ERR /EXIT, AC=0,

```

/TEST OF THE DOUBLE PRECISION COMPLEMENT INSTRUCTION.

```

2717 4773/ DCMTS0, JMS DCMS0H /GO DO INITIALIZATION,
2720 4552 DCMS0, JMS I RNDATA /GENERATE RANDOM DATA,
2721 4556 JMS I LDSC /LOAD THE SC,
2722 4551 JMS I LDREG /LOAD L, MQ, AND GT,
2723 1022 TAD TOAC /AC LOADED
2724 7575 DCM /EAE DOUBLE PRECISION 2/5 COMPLEMENT
2725 4541 JMS I SAVREG /SAVE L, AC, MQ, SC, GT,
2726 4772/ JMS DCMSIM /SIMULATE DCM
2727 4452 JMS I UCOMP /COMPARE ACTUAL AGAINST SIMULATED,
2730 7775 -3 /L, AC, MQ ONLY
2731 5771/ JMP EDMS0 /ERROR
2732 5770/ JMP EDMS0+4 /NO ERROR

2770 3015
2771 3011
2772 6311
2773 3000
2774 2717
2775 2653
2776 2601
2777 6273
3000

```

PAGE

/ROUTINE TO DO INITIALIZATION FOR DCM TEST 0,

```

3000 0000 DCMS0H, OPEN /HOUSEKEEPING FOR DCMTS0
3001 4540 JMS I ASCOMP /SET COMPARE ROUTINE,
3002 1377 TAD (DCMTS0
3003 3056 DCA BACK
3004 1376 TAD (DADTS0
3005 3057 DCA NEXT
3006 4535 JMS I MOOSEL /PERFORM MODE SELECTION,
3007 4536 JMS I ONLYB /EXIT IF MODE "A",
3010 5600 JMP I DCMS0H /EXIT

```

/ROUTINE TO CHECK SR OPTIONS FOR DCM TEST 0,

```

3011 4545 EDMS0, JMS I TSTSW2 /CHECK SR 2,
3012 4220 JMS DM0ERR /PRINT ERROR DATA,
3013 4543 JMS I TSTSW0 /CHECK SR 0,
3014 7402 HLT /DCM ERROR,
3015 4544 JMS I TSTSW1 /CHECK SR 1,
3016 5775/ JMP DCM0+1 /LOOP THE ROUTINE,
3017 5774/ JMP DCM0 /CONTINUE NORMAL TEST,

```

/ROUTINE TO PRINT ERROR INFORMATION,

```

3020 0000 DM0ERR, OPEN
3021 4534 JMS I TYTST /PRINT THE FOLLOWING:
3022 7775 -3
3023 7473 EDMS0 /DCM
3024 7440 TEST /TEST
3025 7443 ZER0 /0
3026 4537 JMS I PREGS /HEADING AND REGISTERS,

```

```

3027 5620 JMP I DM0ERR /EXIT, AC=0,

/TEST OF THE DOUBLE PRECISION ADD INSTRUCTION
/USING FIXED DATA,

3030 4267 DADTS0, JMS DADS0H /GO DO HOUSEKEEPING
3031 4253 DAD0, JMS DADGEN /GET NUMBERS TO ADD,
3032 1021 TAD TOLINK
3033 7104 CLL RAL /LINK LOADED
3034 1023 TAD TOMQ
3035 7421 MQL /MQ LOADED
3036 1024 TAD TOSHIF
3037 3122 DCA DPAC /MSH LOADED
3040 1025 TAD TOGT
3041 3121 DCA DPMQ /LSH LOADED
3042 1022 TAD TOAC /AC LOADED
3043 7443 DAD /EAE DOUBLE PRECISION ADD
3044 0121 DPMQ /ADDRESS
3045 4541 JMS I SAVREG /SAVE L, AC, MQ, SC, GT,
3046 4773/ JMS DADSIM /SIMULATE DAD
3047 4452 JMS I UCOMP /COMPARE SIMULATED AGAINST ACTUAL
3050 7775 -3 /L, AC, MQ,
3051 5307 JMP EDAD0 /ERROR
3052 5325 JMP EDAD0+16 /NO ERRORS ENCOUNTERED

3053 0000 DADGEN, OPEN
3054 4453 JMS I UMOVE
3055 0000 OPEN
3056 0021 TOLINK
3057 7773 -5
3060 7326 CLA CLL CML RTL /AC=2
3061 7124 CLL CML RAL /AC=5
3062 1255 TAD /ADD THE ADDRESS
3063 3255 DCA /PUT IT BACK WITH 5 ADDED TO IT
3064 2114 ISE /FINISHED WITH ALL STORED PATTERNS
3065 5653 JMP I DADGEN /NO,
3066 5575 JMP I GEN+3 /YES,

/INITIALIZATION ROUTINE FOR DAD TEST 0,

3067 0000 DADS0H, OPEN
3070 4540 JMS I ASCOMP /SET COMPARE ROUTINE,
3071 1372 TAD (DADTAB /GET ADDRESS OF TABLE
3072 3255 DCA DADGEN+2 /AND STORE IT IN DADGEN+2
3073 1376 TAD (DADTS0
3074 3056 DCA BACK
3075 1371 TAD (DADTS1
3076 3057 DCA NEXT
3077 1370 TAD (=11
3100 3114 DCA SCOUNT
3101 1707/ TAD DADJMS /GET A JMS TO MODIFY PRINT ROUTINE,
3102 3766/ DCA WILCHG /PUT THE JMS IN WILCHG,
3103 4535 JMS I MOOSEL /PERFORM MODE SELECTION,
3104 4536 JMS I ONLYB /EXIT IF MODE "A",

```

3105 7403 ACS /CLEAR THE STEP COUNTER,
 3106 5667 JMP I DADS0H /EXIT

/ROUTINE TO CHECK SR OPTIONS FOR DAD TEST 0,

3107 1024 EQAD0, TAD TOSHIF
 3110 3040 DCA TEMPA /MSH TO TEMPA
 3111 1025 TAD TOGT
 3112 3041 DCA TEMPB /LSH TO TEMPB
 3113 3024 DCA TOSHIF /0
 3114 3025 DCA TOGT /0
 3115 4945 JMS I TSTSW2 /CHECK SR 2,
 3116 4330 JMS DABERR /PRINT ERROR DATA
 3117 1040 TAD TEMPA
 3120 3024 DCA TOSHIF /RESTORE MSH
 3121 1041 TAD TEMPB
 3122 3025 DCA TOGT /RESTORE LSH
 3123 4543 JMS I TSTSW0 /CHECK SR 2,
 3124 7402 HLT /DAD ERROR;
 3125 4544 JMS I TSTSW1 /CHECK SR 1,
 3126 5232 JMP DAD0+1
 3127 5231 JMP DAD0

/ROUTINE TO PRINT ERROR INFORMATION,

3130 0000 DABERR, OPEN
 3131 4534 JMS I TYTST /PRINT THE FOLLOWING:
 3132 7775 -3
 3133 7476 EDAD /DAD
 3134 7440 EDTEST /TEST
 3135 7443 ZCR0 /0
 3136 4537 JMS I PREGS /HEADING AND REGISTERS;
 3137 5730 JMP I DABERR /EXIT, AC=0;

3166 7016
 3167 7044
 3170 7767
 3171 3200
 3172 7327
 3173 6332
 3174 2720
 3175 2721
 3176 3030
 3177 2717
 3200

PAGE

/RANDOM DOUBLE PRECISION ADD TEST 1

3200 4223 DADTS1, JMS DADS1H /GO DO HOUSEKEEPING

3201 4241 DAD1, JMS RANDAD /GENERATE RANDOM NUMBERS
 3202 1021 TAD TOLINK
 3203 7104 CLL RAL /LINK LOADED
 3204 1023 TAD TOMQ
 3205 7421 MQL /MQ LOADED
 3206 1024 TAD TOSHIF
 3207 3122 DCA DPAC
 3210 1025 TAD TOGT
 3211 3121 DCA DPMQ /LEAST SIGNIFICANT LOADED
 3212 1022 TAD TOAC /AC LOADED
 3213 7443 DAD /EAE DOUBLE PRECISION ADD
 3214 0121 DPMQ /ADDRESS
 3215 4541 JMS I SAVREG /SAVE L, AC, MQ, SC, GP,
 3216 4777 JMS DADSIM /SIMULATE DAD
 3217 4492 JMS I UCOMP /COMPARE SIMULATED AGAINST ACTUAL
 3220 7775 -3 /L, AC, MQ;
 3221 5257 JMP EDAD1 /ERROR
 3222 5275 JMP EDAD1+16 /NO ERRORS ENCOUNTERED

/INITIALIZATION ROUTINE FOR DAD TEST 1,

3223 0000 DADS1H, OPEN
 3224 4540 JMS I ASCOMP /SET COMPARE ROUTINE;
 3225 1376 TAD (DADTS1
 3226 3056 DCA BACK
 3227 1375 TAD (DSTTS0 /ADDRESS OF NEXT TEST
 3230 3057 DCA NEXT
 3231 1774 TAD DADJMS /GET A JMS TO MODIFY PRINT ROUTINE;
 3232 3773 DCA WILCHG /PUT THE JMS IN WILCHG;
 3233 3045 DCA SCSIM /CLEAR
 3234 3046 DCA GTSIM /CLEAR
 3235 4535 JMS I MODSEL /PERFORM MODE SELECTION;
 3236 4936 JMS I ONLYB /EXIT IF MODE "A";
 3237 7403 ACS /CLEAR THE STEP COUNTER;
 3240 5623 JMP I DADS1H /EXIT AC=0 L=0
 3241 0000 RANDAD, OPEN
 3242 4772 JMS RANGEN
 3243 3022 DCA TOAC /RANDOM DATA FOR "AC"
 3244 4772 JMS RANGEN
 3245 3023 DCA TOMQ /RANDOM DATA FOR "MQ"
 3246 4772 JMS RANGEN
 3247 3024 DCA TOSHIF /RANDOM DATA FOR "MOST SIGNIFICANT"
 3250 4772 JMS RANGEN
 3251 3025 DCA TOGT /RANDOM DATA FOR "LEAST SIGNIFICANT"
 3252 7210 CLA RAR
 3253 3021 DCA TOLINK /RANDOM DATA FOR LINK
 3254 4573 JMS I EDCCOUNT /DONE?
 3255 5641 JMP I RANDAD /NO
 3256 5575 JMP I CGEN+3 /YES

/ROUTINE TO CHECK SR OPTIONS FOR DAD TEST 1,

3257 1024 EDAD1, TAD TOSHIF

```

3260 3040      DCA  TEMPA
3261 1025      TAD  TOGT
3262 3041      DCA  TEMPB
3263 3024      DCA  TOSHIF
3264 3025      DCA  TOGT
3265 4545      JMS  I  TSTSW2      /CHECK SR 2,
3266 4300      JMS  DAIERR      /PRINT ERROR DATA,
3267 1040      TAD  TEMPA
3270 3024      DCA  TOSHIF
3271 1041      TAD  TEMPB
3272 3025      DCA  TOGT
3273 4543      JMS  I  TSTSW0      /CHECK SR 0,
3274 7402      HLT
3275 4544      JMS  I  TSTSW1      /CHECK SR 1,
3276 5202      JMP  DAD1+1
3277 5201      JMP  DAD1
    
```

/ROUTINE TO PRINT ERROR INFORMATION FOR DAD TEST 1.

```

3300 0000      DAIERR, OPEN
3301 4534      JMS  I  TYTST      /PRINT THE FOLLOWING
3302 7775      -3
3303 7476      ZDAD      /DAD
3304 7440      TEST      /TEST
3305 7445      ZONE      /1
3306 4537      JMS  I  PREGS      /HEADING AND REGISTERS
3307 5700      JMP  I  DAIERR      /EXIT, AC=0
    
```

/TEST OF THE DOUBLE PRECISION STORE INSTRUCTION,
/USING FIXED DATA

```

3310 4771/     DSTTS0, JMS  DSTS0H      /GO DO HOUSEKEEPING
3311 4770/     DST0,   JMS  DST0GN      /SET UP NUMBERS TO BE STORED
3312 1042      TAD  LSIM
3313 7104      CLL  RAL      /LINK LOADED
3314 1044      TAD  LSH
3315 7421      MQL      /MQ LOADED;
3316 1043      TAD  MSH      /AC LOADED
3317 7445      DST      /EAE DOUBLE PRECISION STORE
3320 0121      DPMQ      /MQ TO BE STORED IN DPMQ AND AC IN DPAC
3321 4541      JMS  I  SAVREG      /SAVE L, AC, MQ, SC, GP,
3322 1121      TAD  DPMQ
3323 3037      DCA  GTOCK      /SAVE STORED MQ
3324 1122      TAD  DPAC
3325 3036      DCA  SCTOCK      /SAVE STORED AC
3326 4452      JMS  I  UCOMP      /CHECK L, AC, MQ AGAINST ORIGINAL
3327 7775      -3
3330 5767/     JMP  EDST0      /ERROR, L, AC, OR MQ MODIFIED BY DST,
3331 1044      TAD  LSH
3332 7421      MQL      /LOAD MQ WITH GOOD
3333 1043      TAD  MSH      /LOAD AC WITH GOOD
    
```

```

3334 7575      DCM      /2'S COMPLEMENT
3335 7443      DAD      /ADD "TO BE CHECKED"
3336 0121      DPMQ      /STORED AT THIS ADDRESS
3337 7451      DPSZ      /ARE THEY THE SAME?
3340 5767/     JMP  EDST0      /ERROR
3341 5766/     JMP  EDST0+4    /NO ERRORS ENCOUNTERED
    
```

```

3366 3435
3367 3431
3370 3400
3371 3413
3372 6525
3373 7016
3374 7044
3375 3310
3376 3200
3377 6332
3400
    
```

PAGE

```

3400 0000      DST0GN, OPEN
3401 4453      JMS  I  UMOVE
3402 0000      OPEN
3403 0042      LSIM
3404 7775      -3
3405 7325      CLA  CLL  CML  IAC  RAL      /AC = 3
3406 1202      TAD  I=4
3407 3202      DCA  I=5      /+3 TO DST0GN+2
3410 2114      ISE  SCOUNT
3411 5000      JMP  I  DST0GN
3412 5575      JMP  I  CGEN+3
    
```

/INITIALIZATION ROUTINE FOR DST TEST 0

```

3413 0000      DSTS0H, OPEN
3414 4540      JMS  I  ASCOMP      /SET COMPARE ROUTINE;
3415 1377      TAD  (DSTTAB
3416 3202      DCA  DST0GN+2
3417 1376      TAD  (DSTTS0
3420 3056      DCA  BACK
3421 1375      TAD  (DSTYS1
3422 3057      DCA  NEXT
3423 1374      TAD  (=7
3424 3114      DCA  SCOUNT
3425 4535      JMS  I  MODSEL      /PERFORM MODE SELECTION;
3426 4536      JMS  I  ONLYB      /EXIT IF MODE "A";
3427 7403      ACS      /CLEAR THE STEP COUNTER;
3430 5613      JMP  I  DSTS0H      /EXIT AC=L=0
    
```

/ROUTINE TO CHECK SR OPTIONS FOR DST TEST 0

```

3431 4545      EDST0, JMS  I  TSTSW2      /CHECK SR 2,
3432 4240      JMS  DS0ERR      /PRINT ERROR DATA;
3433 4543      JMS  I  TSTSW0      /CHECK SR 0,
3434 7402      HLT      /DST ERROR;
    
```



```

3435 4544      JMS I  TSTSW1      /CHECK SR 1,
3436 5773/      JMP      DST0+1      /LOOP THE ROUTINE,
3437 5772/      JMP      DST0      /CONTINUE NORMAL TEST,

/ROUTINE TO PRINT ERROR INFORMATION

3440 0000      DS0ERR, OPEN
3441 4534      JMS I  TYTST      /PRINT THE FOLLOWING
3442 7775      -3
3443 7501      EDST      /DST
3444 7440      TEST      /TEST
3445 7443      EER0      /0
3446 4771/      JMS      DSTREG      /HEADING AND REGISTERS
3447 5640      JMP I  DS0ERR      /EXIT, AC=0

/TEST OF THE DOUBLE PRECISION STORE INSTRUCTION,
/USING RANDOM DATA,

3450 4314      DSTTS1, JMS      DSTS1H      /GO DO HOUSEKEEPING
3451 4302      DST1,  JMS      DST1RN      /SET UP RANDOM NUMBERS
3452 1042      TAD      LSH
3453 7104      CLL RAL      /LINK LOADED
3454 1044      TAD      LSH
3455 7421      MQL      /MQ LOADED
3456 1043      TAD      MSH      /AC LOADED
3457 7445      DST      /EAE DOUBLE PRECISION STORE
3460 0121      DPMQ      /MQ TO BE STORED IN DPMQ AND AC IN DPAC,
3461 4541      JMS I  SAVREG      /SAVE L,AC,MO,SC,GT,
3462 1121      TAD      DPMQ
3463 3037      DCA      GTTOCK      /SAVE LEAST SIGNIFICANT
3464 1122      TAD      DPAC
3465 3036      DCA      SCTOCK      /SAVE MOST SIGNIFICANT
3466 4452      JMS I  UCOMP      /COMPARE L, AC, AND MQ AGAINST ORIGINAL,
3467 7775      -3
3470 5326      JMP      EDST1      /ERROR, L, AC, OR MQ MODIFIED BY DST,
3471 1044      TAD      LSH
3472 7421      MQL      /ORIGINAL MQ
3473 1043      TAD      MSH      /ORIGINAL AC
3474 7575      DCM      /2'S COMPLEMENT
3475 7443      DAD      /ADD "TO BE CHECKED"
3476 0121      DPMQ      /STORED AT THIS ADDRESS
3477 7451      DPSZ      /ARE THEY THE SAME?
3500 5326      JMP      EDST1      /ERROR
3501 5332      JMP      EDST1+4      /NO ERRORS ENCOUNTERED,

```

/ROUTINE TO GENERATE RANDOM DATA FOR DST TEST 1.

```

3502 0000      DST1RN, OPEN
3503 4770/      JMS      RANGEN
3504 3043      DCA      MSH      /RANDOM DATA TO BE PUT IN AC
3505 4770/      JMS      RANGEN

```

```

3506 3044      DCA      LSH      /RANDOM DATA TO BE PUT IN MQ
3507 7010      RAR
3510 3042      DCA      LSH      /RANDOM DATA TO BE PUT IN LINK
3511 4573      JMS I  DCOUNT      /DONE?
3512 5702      JMP I  DST1RN      /NO
3513 5575      JMP I  CGEN+3      /YES

```

/INITIALIZATION ROUTINE FOR DST TEST 1.

```

3514 0000      DSTS1H, OPEN
3515 4540      JMS I  ASCOMP      /SET COMPARE ROUTINE,
3516 1375      TAD      (DSTTS1
3517 3056      DCA      BACK
3520 1367      TAD      (NORMT
3521 3057      DCA      NEXT
3522 4535      JMS I  MODSEL      /PERFORM MODE SELECTION,
3523 4536      JMS I  ONLYB      /EXIT IF "A" MODE,
3524 7403      ACS      /CLEAR THE STEP COUNTER,
3525 5714      JMP I  DSTS1H      /EXIT, AC=0

```

/ROUTINE TO CHECK SR OPTION FOR DST TEST 1

```

3526 4545      EDST1, JMS I  TSTSW2      /CHECK SR 2,
3527 4335      JMS      DS1ERR      /PRINT ERROR DATA,
3530 4543      JMS I  TSTSW0      /CHECK SR 0,
3531 7402      HLT      /DST ERROR,
3532 4544      JMS I  TSTSW1      /CHECK SR 1,
3533 5252      JMP      DST1+1      /LOOP THE ROUTINE,
3534 5251      JMP      DST1      /CONTINUE NORMAL TEST,

```

/ROUTINE TO PRINT ERROR INFORMATION,

```

3535 0000      DS1ERR, OPEN
3536 4534      JMS I  TYTST      /PRINT THE FOLLOWING
3537 7775      -3
3540 7501      EDST      /DST
3541 7440      TEST      /TEST
3542 7445      EONE      /1
3543 4771/      JMS      DSTREG      /HEADING AND REGISTERS
3544 5735      JMP I  DS1ERR      /EXIT, AC=0,

```

```

3567 3600
3570 6525
3571 7106
3572 3311
3573 3312
3574 7771
3575 3450
3576 3310
3577 7305

```

PAGE

/TEST OF THE NORMALIZE INSTRUCTION,

```

3600 5257 NORMT, JMP HSENHI
3601 4312 JMS GXEN
3602 7240 CLA CMA
3603 0305 AND MQNMIX
3604 7421 MQL /LOAD MQ INDEXED PATTERN
3605 7040 CMA
3606 0804 AND ACNMIX /LOAD AC INDEXED PATTERN
3607 7411 NMI
3610 3307 DCA ACNMIN /STORE AC
3611 7501 MQA
3612 3306 DCA MQNMIN /STORE MQ
3613 7441 SCA
3614 3300 DCA SCAST /STORE SCA COUNT
3615 7040 CMA
3616 0307 AND ACNMIN
3617 7140 CLL CMA
3620 1301 TAD NMIOOD /8000
3621 7040 CMA
3622 7440 SZA
3623 5250 JMP NMIERR /AC DID NOT EQUAL 8000
3624 7430 SEL
3625 5250 JMP NMIERR /AC DID NOT EQUAL 8000
3626 7240 CLA CMA
3627 0306 AND MQNMIN
3630 7440 SZA
3631 5250 JMP NMIERR /MQ DID NOT EQUAL 8000
3632 7040 CMA
3633 0300 AND SCAST
3634 7140 CLL CMA
3635 1303 TAD SCASTX /INDEXED STEP COUNT #
3636 7040 CMA
3637 7440 SZA
3640 5250 JMP NMIERR /SC IN ERROR
3641 7430 SEL
3642 5250 JMP NMIERR /SC IN ERROR
3643 7240 CLA CMA
3644 0303 AND SCASTX /TEST SCA COUNT FOR 0
3645 7440 SZA /TO EXIT
3646 5254 JMP NMIERR+4 /CONTINUE TEST
3647 5272 JMP EXINMI
3650 4545 NMIERR, JMS I TSTSW2 /CHECK SR 2,
3651 4711 JMS I SCAST+11 /JUMP TO PRINT ROUTINE;
3652 4543 JMS I TSTSW0 /CHECK SR 0,
3653 7402 HLT /NORMALIZE FAILED,
3654 4544 JMS I TSTSW1 /CHECK SR 1,
3655 5202 JMP NORMT+2 /LOOP TEST;
3656 5201 JMP NORMT+1 /CONTINUE TEST;

3657 7240 HSENHI, CLA CMA
3660 0327 AND ANCMIQ
3661 3012 DCA XACNMI /AC AUTO START ADDRESS
3662 7040 CMA
3663 0330 AND MQNMIQ
3664 3013 DCA XMQNMI /MQ AUTO START ADDRESS
    
```

```

3665 7040 CMA
3666 0302 AND SCC23 /SC 23
3667 3303 DCA SCASTX /STORE DECIMAL 23
3670 4535 JMS I MOOSEL /PERFORM MODE SELECTION,
3671 5201 JMP NORMT+1

3672 7604 EXINMI, CLA OSR /TEST SW3
3673 7106 RTL CLL
3674 7006 RTL
3675 7430 SEL
3676 5200 JMP NORMT /REPEAT ENTIRE TEST
3677 5710 JMP I SCAST+10 /JMP TO NEXT NMI TEST

3700 0000 SCAST, 0
3701 6000 NMIOOD, 6000
3702 0027 SCC23, 0027 /23 DECIMAL
3703 0000 SCASTX, 0
3704 0000 ACNMIX, 0
3705 0000 MQNMIX, 0
3706 0000 MQNMIN, 0
3707 0000 ACNMIN, 0
3710 4200 NORMT1
3711 4000 PRNMI

3712 0000 GXEN, 0
3713 7240 CLA CMA
3714 0412 AND I XACNMI
3715 3304 DCA ACNMIX /STORE AC PATTERN
3716 7040 CMA
3717 0413 AND I XMQNMI
3720 3305 DCA MQNMIX /STORE MQ PATTERN
3721 7040 CMA
3722 0303 AND SCASTX /SUBTRACT ONE FROM SCA COUNT
3723 7041 CIA
3724 7040 CMA
3725 3303 DCA SCASTX /STORE DECREMENTED SCA COUNT
3726 5331 JMP EXEN
3727 4000 ANCMIQ, ACNMI
3730 4074 MQNMIQ, MQNMI
3731 7240 EXEN, CLA CMA
3732 0303 AND SCASTX
3733 7440 SZA
3734 5712 JMP I GXEN
3735 5272 JMP EXINMI

4000 0000 PRNMI, OPEN
4001 4325 JMS I CRLF2 /2 CR AND LF;
4002 4326 JMS NMITPR /
    
```

```

4003 4451 JMS I UPSPC /6 SPACES,
4004 7772 -6 /
4005 4777 JMS PC /
4006 4776 JMS LPAR /
4007 4775 JMS AC /PRINT "AC"
4010 4774 JMS RPAR /
4011 4451 JMS I UPSPC /
4012 7765 -13 /
4013 4777 JMS PC /PRINT "PC"
4014 4776 JMS LPAR /
4015 4773 JMS HQ /PRINT "HQ"
4016 4774 JMS RPAR /
4017 4576 JMS I [TYMOD /TYPE THE MODE;
4020 4524 JMS I CRLF /CR AND LF;
4021 4451 JMS I UPSPC /6 SPACES,
4022 7772 -6 /
4023 1772 TAD ACNMIX /
4024 4771 JMS P12BIT /PRINT 12 BITS;
4025 4451 JMS I UPSPC /3 SPACES,
4026 7775 -3 /
4027 1770 TAD HQNMIX /
4030 4771 JMS P12BIT /PRINT 12 BITS;
4031 4524 JMS I CRLF /CR AND LF;
4032 4767 JMS NMIXX /
4033 4451 JMS I UPSPC /3 SPACES,
4034 7775 -3 /
4035 1766 TAD ACNMIX /
4036 4771 JMS P12BIT /PRINT 12 BITS;
4037 4451 JMS I UPSPC /3 SPACES,
4040 7775 -3 /
4041 1765 TAD HQNMIX /
4042 4771 JMS P12BIT /PRINT 12 BITS;
4043 4524 JMS I CRLF /CR AND LF;
4044 4764 JMS SCATXX /
4045 4455 JMS I U2SPC /2 SPACES,
4046 1763 TAD SCASTX /
4047 4771 JMS P12BIT /PRINT 12 BITS;
4050 4524 JMS I CRLF /CR AND LF;
4051 4762 JMS SCAXX /
4052 4451 JMS I UPSPC /3 SPACES,
4053 7775 -3 /
4054 1761 TAD SCAST /
4055 4771 JMS P12BIT /PRINT 12 BITS;
4056 4524 JMS I CRLF /CR AND LF;
4057 5000 JMP I PRNMI /EXIT, AC=0,

```

```

4060 0000 ACNMI, 0
4061 7777 7777 /SC22
4062 7777 7777 /SC21
4063 7777 7777 /SC20
4064 7777 7777 /SC19
4065 7777 7777 /SC18
4066 7777 7777 /SC17
4067 7777 7777 /SC16

```

```

4070 7777 7777 /SC15
4071 7777 7777 /SC14
4072 7777 7777 /SC13
4073 7777 7777 /SC12

```

```

4074 7777 HQNMI, 7777 /SC11
4075 7777 7777 /SC10
4076 7776 7776 /SC9
4077 7774 7774 /SC8
4100 7770 7770 /SC7
4101 7760 7760 /SC6
4102 7740 7740 /SC5
4103 7700 7700 /SC4
4104 7600 7600 /SC3
4105 7400 7400 /SC2
4106 7000 7000 /SC1
4107 6000 6000 /SC0
4110 4000 4000
4111 0000 0000
4112 0000 0000
4113 0000 0000
4114 0000 0
4115 0000 0
4116 0000 0
4117 0000 0
4120 0000 0
4121 0000 0
4122 0000 0
4123 0000 0
4124 0000 0
4125 0000 0

```

```

4126 0000 NMITPR, 0
4127 4332 JMS PNORM
4130 4343 JMS XNORMT
4131 5726 JMP I NMITPR
4132 0000 PNORM, 0
4133 7240 CLA CMA
4134 0760 AND N
4135 4526 JMS I PRXLOP /PRINT,
4136 1757 TAD N+1
4137 4526 JMS I PRXLOP /PRINT,
4140 1756 TAD N+2
4141 4526 JMS I PRXLOP /PRINT,
4142 5732 JMP I PNORM
4143 0000 XNORMT, 0
4144 7240 CLA CMA
4145 0755 AND N+3
4146 4526 JMS I PRXLOP /PRINT,
4147 5743 JMP I XNORMT

```

4155	5477		
4156	5476		
4157	5475		
4160	5474		
4161	3700		
4162	5434		
4163	3703		
4164	5430		
4165	3706		
4166	3707		
4167	5425		
4170	3705		
4171	7200		
4172	3704		
4173	0302		
4174	5467		
4175	0323		
4176	5462		
4177	5455		
	4200	PAGE	
4200	5261	NORMT1, JMP HSENM	
4201	4273	JMS GENNMI	
4202	7240	CLA CMA	
4203	0716	AND I TST25+1	/LOAD MQ PATTERN
4204	7421	MQL	
4205	7240	CLA CMA	
4206	0717	AND I TST25+2	/LOAD AC PATTERN
4207	7411	NMI	
4210	3725	DCA I TST25+10	/STORE NORMALIZED AC
4211	7501	MQA	
4212	3726	DCA I TST25+11	/STORE NORMALIZED MQ
4213	7441	SCA	
4214	3727	DCA I TST25+12	/STORE SCA COUNT
4215	7240	CLA CMA	
4216	0725	AND I TST25+10	
4217	7140	CLL CMA	
4220	1716	TAD I TST25+1	
4221	7040	CMA	
4222	7440	SZA	
4223	5333	JMP NMERR	/AC DID NOT EQUAL 2525
4224	7430	SZL	
4225	5333	JMP NMERR	/AC DID NOT EQUAL 2525
4226	7240	CLA CMA	
4227	0726	AND I TST25+11	
4230	7440	SZA	
4231	5333	JMP NMERR	/MQ DID NOT EQUAL 0000
4232	7240	CLA CMA	
4233	0727	AND I TST25+12	
4234	7140	CLL CMA	
4235	1331	TAD DEC12	/DECIMAL 12
4236	7040	CMA	
4237	7440	SZA	
4240	5333	JMP NMERR	/SC DID NOT EQUAL 12

4241	7430	SZL	
4242	5333	JMP NMERR	/SC DID NOT EQUAL 12
4243	2315	ISE TST25	/REPEAT CURRENT TEST PATTERN
4244	5202	JMP NORMT1+2	
4245	7604	CLA OSR	/TEST SW1
4246	7106	RTL CLL	
4247	7430	SZL	
4250	5202	JMP NORMT1+2	
4251	2322	ISE NMFLG	
4252	5201	JMP NORMT1+1	
4253	7604	CLA OSR	/TEST SW3
4254	7106	RTL CLL	
4255	7006	RTL	
4256	7430	SZL	
4257	5200	JMP NORMT1	
4260	5724	JMP I NEXNMI	
4261	7200	HSENM, CLA	
4262	3315	DCA TST25	/CLEAR TEST COUNTER
4263	7400	NOP	
4264	7040	CMA	
4265	0323	AND NM7776	
4266	3322	DCA NMFLG	
4267	1331	TAD DEC12	
4270	3750	DCA I TST25+13	
4271	4535	JMS I MODSEL	/PERFORM MODE SELECTION,
4272	5201	JMP NORMT1+1	
4273	0000	GENNMI, 0	
4274	7240	CLA CMA	
4275	0322	AND NMFLG	
4276	7040	CMA	
4277	7440	SZA	
4300	5302	JMP PA2525	/GENERATE 2525
4301	5307	JMP PA2525+5	/GENERATE 5252
4302	7240	CLA CMA	
4303	0320	AND NM2525	/MQ PATTERN 2525
4304	3716	DCA I TST25+1	
4305	3717	DCA I TST25+2	/AC PATTERN 0000
4306	5673	JMP I GENNMI	
4307	7240	CLA CMA	
4310	0321	AND NM5252	/MQ PATTERN 5252
4311	3716	DCA I TST25+1	
4312	7040	CMA	
4313	3717	DCA I TST25+2	/AC PATTERN 7777
4314	5673	JMP I GENNMI	
4315	0000	TST25, 0	
4316	3705	MOQNMIX	
4317	3704	ACNMIX	
4320	2525	NM2525, 2525	
4321	5252	NM5252, 5252	
4322	0000	NMFLG, 0	

```

4323 7776  NM7776, 7776
4324 4400  NEXNMI, NORMT2
4325 3707  ACNMIN
4326 3706  M0NMIN
4327 3700  SCAST
4330 3703  SCASTX
4331 0014  DEC12, 0014
4332 4000  PRNMI
4333 4545  NMERR, JMS I TSTSW2 /CHECK SR 2,
4334 4732  JMS I DEC12+1
4335 4543  JMS I TSTSW0 /CHECK SR 0,
4336 7402  HLT /NORMALIZE ERROR,
4337 4544  JMS I TSTSW1 /CHECK SR 1,
4340 7610  SKP CLA
4341 5202  JMP NORMT1+2 /CONTINUE TEST,
4342 3315  DCA TST25 /CLEAR CURRENT TEST COUNTER,
4343 5202  JMP NORMT1+2 /LOOP CURRENT TEST,

```

4400 PAGE

```

4400 5305  NORMT2, JMP HKE /HOUSE KEEPING
4401 4253  JMS GEX /PATTERN GENERATOR
4402 7621  CMA
4403 7040  CMA
4404 0725  AND I PAT01
4405 7421  MQL /MQ PATTERN
4406 7140  CLL CMA /AC PATTERN
4407 0726  AND I PAT00 /AC PATTERN
4410 7411  NMI
4411 3727  DCA I SPAT00 /STORE AC NORMALIZED PATTERN
4412 7501  MQA
4413 3730  DCA I SPAT01 /STORE MQ NORMALIZED PATTERN
4414 7441  SCA
4415 3734  DCA I SCANM /STORE SCA COUNT
4416 7040  CMA
4417 0727  AND I SPAT00 /AC PATTERN
4420 7040  CMA
4421 1331  TAD CHKAC /CHECK PATTERN AC
4422 7040  CMA
4423 7440  SZA /TEST AC BITS
4424 5313  JMP MT2ER /SPAT00 NOT EQUAL TO CHKAC
4425 7430  SEL
4426 5313  JMP MT2ER /SPAT00 NOT EQUAL TO CHKAC
4427 7040  CMA
4430 0730  AND I SPAT01 /MQ PATTERN
4431 7040  CMA
4432 1332  TAD CHKMQ /CHECK PATTERN MQ
4433 7040  CMA
4434 7440  SZA /TEST MQ BITS
4435 5313  JMP MT2ER /SPAT01 NOT EQUAL TO CHKMQ
4436 7430  SEL
4437 5313  JMP MT2ER /SPAT01 NOT EQUAL TO CHKMQ

```

```

4440 7040  CMA
4441 0734  AND I SCANM /SCA COUNT PATTERN
4442 7041  CIA
4443 1733  TAD I CHKSCA /CHECK PATTERN SCA
4444 7420  SNL
4445 5313  JMP MT2ER /SCANM NOT EQUAL TO CHKSCA
4446 2336  ISE AGAIN /4096 REPEATS CURRENT TEST
4447 5202  JMP NORMT2+2

```

```

4450 4544  NMTS1, JMS I TSTSW1
4451 5202  JMP NORMT2+2
4452 5345  JMP PATCH /JUMP TO SW3

```

```

4453 0000  GEX, 0
4454 7240  CLA CMA
4455 0337  AND TPFLAG
4456 7040  CMA
4457 7440  SZA
4460 5262  JMP ,+2 /GENERATE 0000 MQ PATTERN
4461 5271  JMP ,+10 /GENERATE 0001 MQ PATTERN
4462 7200  CLA
4463 3726  DCA I PAT00 /STORE AC PATTERN
4464 3725  DCA I PAT01 /STORE MQ PATTERN
4465 3331  DCA CHKAC /STORE AC CHECK
4466 3332  DCA CHKMQ /STORE MQ CHECK
4467 3733  DCA I CHKSCA /STORE SCA CHECK
4470 5653  JMP I GEX
4471 7240  CLA CMA
4472 0335  AND SCANM+1 /MQ PATTERN (0001)
4473 3725  DCA I PAT01 /STORE MQ PATTERN
4474 7040  CMA
4475 0340  AND TPFLAG+1 /22 DECIMAL PLACES (0030)
4476 3733  DCA I CHKSCA
4477 3726  DCA I PAT00 /STORE AC PATTERN
4500 3332  DCA CHKMQ /STORE MQ CHECK
4501 7040  CMA
4502 0341  AND TPFLAG+2 /20000
4503 3331  DCA CHKAC /STORE AC CHECK
4504 5653  JMP I GEX

```

```

4505 7240  HKE, CLA CMA /HOUSE KEEPING
4506 0342  AND TPFLAG+3 /776
4507 3337  DCA TPFLAG /LOAD FLAG
4510 3336  DCA AGAIN /CHECK TEST COUNTER
4511 4535  JMS I MODSEL /PERFORM MODE SELECTION,
4512 5201  JMP NORMT2+1

```

```

4513 4545  MT2ER, JMS I TSTSW2 /CHECK SR 2,
4514 4743  JMS I TPFLAG+4 /PRINT ROUTINE
4515 7004  CLA OSR /TEST SW0
4516 7104  RAL CLL
4517 7430  SEL
4520 7402  HLT /NORMALIZE ERROR,
4521 5250  JMP NMTS1

```

```

4522 4546 NMTS3, JMS I TSTSH3 /CHECK SR 3,
4523 5200 JMP NORMT2 /CONTINUE
4524 5744 JMP I TPFLAG+5
4525 3705 PAT01, MQNMIX
4526 3704 PAT00, ACNMIX
4527 3707 SPAT00, ACNMIX
4530 3706 SPAT01, MQNMIX
4531 0000 CHKAC, 0
4532 0000 CHKMQ, 0
4533 3703 CHKSCA, SCASTX
4534 3700 SCANM, SCAST
4535 0001 0001
4536 0000 AGAIN, 0

```

```

4537 0000 TPFLAG, 0
4540 0026 0026
4541 2000 2000
4542 7776 7776
4543 4000 PRNMI
4544 4600 COMTST
4545 2337 PATCH, ISE TPFLAG
4546 5201 JMP NORMT2+1
4547 5322 JMP NMTS3

```

PAGE 4600

/TEST OF EAE NOP

```

4600 7240 COMTST, CLA CMA /7777
4601 7421 MQL /MQ=7777
4602 7501 MQA /AC=7777
4603 7401 NOPM /EAE NOP
4604 7410 SKP
4605 7402 HLT /NOP SKIPPED
4606 7040 CMA /0
4607 7640 SEA CLA /
4610 7402 HLT /AC MODIFIED BY NOPM
4611 7501 MQA /MQ TO AC
4612 7040 CMA /AC SHOULD NOW BE 0
4613 7440 SEA /WAS IT 0?
4614 7402 HLT /NO, MQ WAS MODIFIED BY NOPM

```

/TEST OF EAE CLA

```

4615 7240 CLA CMA /7777
4616 7421 MQL /MQ=7777
4617 7501 MQA /AC=7777
4620 7601 CLAM /EAE CLA
4621 7410 SKP
4622 7402 HLT /CLAM SKIPPED
4623 7640 SEA CLA /
4624 7402 HLT /CLAM FAILED TO CLEAR THE AC,
4625 7501 MQA /MQ TO AC

```

```

4626 7040 CMA /AC SHOULD NOW BE 0
4627 7440 SEA /WAS IT 0?
4630 7402 HLT /MQ MODIFIED BY CLAM

```

/TEST OF EAE CAM

```

4631 7240 CLA CMA /AC=7777
4632 7421 MQL /MQ=7777
4633 7501 MQA /AC=7777
4634 7621 CAM /CLEAR THE AC AND MQ,
4635 7501 MQA /MQ DIED WITH AC,
4636 7440 SEA /WERE THEY BOTH 0?
4637 7402 HLT /AC OR MQ NOT CLEARED BY CAM,

```

/TEST OF EAE SWP,

```

4640 7200 CLA /0,
4641 1172 TAD [5252 /5252
4642 7421 MQL /MQ=5252
4643 1171 TAD [2525 /AC=2525
4644 7521 SWP /SWAP AC AND MQ; AC=5252 + MQ=2525
4645 1171 TAD [2525 /AC=7777
4646 7040 CMA /AC=0000
4647 7440 SEA
4650 7402 HLT /SWP FAILED,
4651 7501 MQA /AC=2525
4652 1172 TAD [5252 /AC=7777
4653 7040 CMA /AC SHOULD BE 0
4654 7440 SEA
4655 7402 HLT /SWP FAILED,

```

/TEST OF ACL (MQA CLAM),

```

4656 7621 CAM /AC AND MQ = 0
4657 1171 TAD [2525 /AC=2525
4660 7421 MQL /MQ=2525
4661 1172 TAD [5252 /AC=5252
4662 7701 ACL /CLA THE AC AND LOAD AC FROM MQ,
4663 1172 TAD [5252 /AC=7777
4664 7040 CMA /AC=0
4665 7440 SEA
4666 7402 HLT /ACL FAILED,

```

/TEST OF DLD (CAM DAD),

```

4667 7621 CAM /CLEAR AC AND MQ;
4670 1115 TAD MODE /GET THE MODE;
4671 7650 SNA CLA /SKIP IF B MODE,
4672 9353 JMP COMEND /MODE A SO SKIP B TESTS;

4673 7431 SWAB /B MODE,
4674 7621 CAM /AC AND MQ = 0
4675 1171 TAD [2525 /AC=2525
4676 7421 MQL /MQ=2525
4677 1172 TAD [5252 /AC=5252

```

```

4700 7663      DLD          /EAE DOUBLE PRECISION LOAD;
4701 4703      ,+2          /ADDRESS OF LSH OPERAND;
4702 5305      JMP           /GO OVER;
4703 5252      ,+3          /LSH OPERAND
4704 2525      5252          /MSH OPERAND
4705 1172      TAD          [5252] /AC=7777
4706 7040      CMA          /AC=0
4707 7440      SEA
4710 7402      HLT          /DLD FAILED,
4711 7501      MQA          /MQ TO AC,
4712 1171      TAD          [2525] /AC=7777
4713 7040      CMA          /AC=0
4714 7440      SEA
4715 7402      HLT          /DLD FAILED,

/TEST OF DDZ (CAM DST),

4716 7431      SWAB         /B MODE,
4717 7621      CAM          /AC AND MQ ARE 0
4720 1171      TAD          [2525] /AC=2525
4721 7421      MQL          /MQ=2525
4722 7501      MQA          /AC=2525
4723 3332      DCA          ,+7   /LSH=2525
4724 1172      TAD          [5252] /AC=5252
4725 3333      DCA          ,+6   /MSH=5252
4726 1172      TAD          [5252] /AC=5252
4727 7665      DDE          /EAE DOUBLE PRECISION DEPOSIT ZERO;
4730 4732      ,+2          /OPERAND OF LSH
4731 5334      JMP           /GO OVER;
4732 0000      OPEN        /LSH
4733 0000      OPEN        /MSH
4734 7501      MQA          /MQ OR'ED WITH AC,
4735 7440      SEA         /BOTH 0?
4736 7402      HLT          /NO, AC OR MQ NOT 0,
4737 1332      TAD          ,+5   /GET C(LSH)
4740 7440      SEA         /0?
4741 7402      HLT          /NO, DDZ FAILED,
4742 1333      TAD          ,+7   /GET C(MSH)
4743 7440      SEA         /0?
4744 7402      HLT          /DDZ FAILED, AC DID NOT GET STORED AS 0,

/TEST OF NORMALIZE "B" MODE,

4745 7431      SWAB         /GO TO B MODE,
4746 7621      CAM          /CLEAR AC AND MQ;
4747 7330      CLA CLL CML RAR /AC=0000; MQ=0000;
4750 7411      NMI         /NORMALIZE;
4751 7440      SEA         /DID AC GET CLEARED?
4752 7402      HLT          /NMI FAILED TO CLEAR AC,

4753 7447      COMEND, SWBA /GO TO A MODE;
4754 4546      JMS I TSTSW3 /REMAIN IN THIS TEST
4755 5200      JMP          COMTST /YES, SR3=1
4756 2117      ISZ          COUNT /DO THIS TEST 4096 TIMES BEFORE EXIT,
4757 5200      JMP          COMTST /REPEAT THE TESTS UNTIL DONE,
    
```

```

4760 5777/     JMP          E3A          /GO PRINT OR SOMETHING;

4777 5261      PAGE
5000 0000      TSCL, OPEN        /MODE, GT, AND SC TESTS

/TEST OF MODE SWITCHING,

5001 7621      MDTST, CAM        /CLEAR AC AND MQ;
5002 7491      DPSZ          /
5003 7410      SKP           /
5004 7402      HLT          /CLEAR KEY FAILED TO SET TO "A" MODE OR DPSZ FAILED,
5005 7431      SWAB         /SET "B" MODE,
5006 7621      CAM          /
5007 7491      DPSZ          /
5010 7402      HLT          /SWAB FAILED TO SET "B" MODE OR DPSZ FAILED,
5011 7447      SWBA         /SET "A" MODE,
5012 7621      CAM          /
5013 7491      DPSZ          /
5014 7410      SKP           /
5015 7402      HLT          /SWBA FAILED TO SET "A" MODE OR DPSZ FAILED,
5016 7431      SWAB         /SET "B" MODE
5017 6007      CAF          /INITIALIZE,
5020 7621      CAM          /
5021 7491      DPSZ          /
5022 7610      SKP CLA       /
5023 7402      HLT          /INITIALIZE FAILED TO SET TO "A" MODE,

/STEP COUNTER TESTS,

5024 7200      SCL1, CLA        /TEST SCL=0
5025 7403      7403          /SCL
5026 7737      7737          /SC=0
5027 7441      SCA
5030 7640      CLA SEA
5031 7402      HLT          /ERROR! SC NOT=0
5032 7403      SCL2, 7403      /TEST SCL=01
5033 7776      7776          /SC=1
5034 7441      SCA
5035 1233      TAD          ,+2   /
5036 7040      CMA
5037 7640      CLA SEA
5040 7402      HLT          /ERROR! SC NOT=01
5041 7403      SCL3, 7403      /TEST SCL=02
5042 7775      7775          /SC=2
5043 7441      SCA
5044 1242      TAD          ,+2   /
5045 7040      CMA
5046 7640      CLA SEA
5047 7402      HLT          /ERROR! SC NOT=02
5050 7403      SCL4, 7403      /TEST SCL=04
5051 7773      7773          /SC=4
    
```

```

5052 7441 SCA
5053 1251 TAD ,=2
5054 7040 CMA
5055 7640 CLA SZA
5056 7402 ESCL4, HLT /ERROR! SC NOT = 04
5057 7403 SCL5, 7403 /TEST SCL=10
5060 7767 7767 /SC=10
5061 7441 SCA
5062 1260 TAD ,=2
5063 7040 CMA
5064 7640 CLA SZA
5065 7402 ESCL5, HLT /ERROR! SC NOT=10
5066 7403 SCL6, 7403 /TEST SCL=20
5067 7757 7757 /SC=20
5070 7441 SCA
5071 1267 TAD ,=2
5072 7040 CMA
5073 7640 CLA SZA
5074 7402 ESCL6, HLT /ERROR! SC NOT=20

5075 7403 SCL7, 7403 /TEST SCL=12
5076 7765 7765 /SC=12
5077 7441 SCA
5100 1276 TAD ,=2
5101 7040 CMA
5102 7640 CLA SZA
5103 7402 ESCL7, HLT /ERROR! SC NOT=12
5104 7403 SCL8, 7403 /TEST SCL=25
5105 7752 7752
5106 7441 SCA
5107 1305 TAD ,=2
5110 7040 CMA
5111 7640 CLA SZA
5112 7402 ESCL8, HLT /ERROR! SC NOT=25
5113 7403 SCL9, 7403 /TEST SCL=0
5114 0077 0077 /SC=0
5115 7441 SCA
5116 7640 CLA SZA
5117 7402 ESCL9, HLT /ERROR! SC NOT=0
5120 7403 SCL10, 7403 /TEST SCL=37
5121 7700 7700
5122 7441 SCA
5123 1123 TAD K7740
5124 7040 CMA
5125 7640 CLA SZA
5126 7402 ESCL10, HLT /ERROR! SC NOT 37

5127 7403 SCL11, SCL /LOAD THE SC WITH
5130 7777 7777 /0000
5131 7240 CLA CMA /7777
5132 7441 SCA /SC TO AC
5133 7040 CMA
5134 7440 SEA
5135 7402 ESCL11, HLT /SC DID NOT "ORM" WITH AC.
    
```

```

5136 7403 SCL12, SCL /LOAD SC WITH
5137 7752 7752 /25
5140 7200 CLA
5141 1337 TAD ,=2
5142 7441 SCA /SC TO AC
5143 7040 CMA
5144 7440 SEA
5145 7402 ESCL12, HLT /SC DID NOT "ORM" WITH THE AC.

5146 7403 SCL13, SCL /LOAD THE SC
5147 7765 7765 /WITH 12,
5150 7200 CLA
5151 1347 TAD ,=2
5152 7441 SCA /SC TO AC
5153 7040 CMA
5154 7440 SEA /
5155 7402 ESCL13, HLT /SC DID NOT "ORM" WITH THE AC.

/TEST OF THE ACS INSTRUCTION,

5156 7431 ACS1, SWAB /CHANGE TO MODE B
5157 7360 CLA CMA CLL CML /AC=7777 L=1
5160 7403 ACS /AC TO SC
5161 7430 SEL
5162 7440 SEA
5163 7402 HLT /ACS CLEARED THE LINK OR ACS FAILED
/TO CLEAR THE AC,
/SC TO AC

5164 7441 SCA
5165 1123 TAD K7740
5166 7040 CMA
5167 7440 SEA
5170 7402 HLT /ACS FAILED TO LOAD THE STEP COUNTER WITH 37.
5171 5777 JMP ACS2 /

5177 5200 PAGE
5200 7320 ACS2, CLA CML CLL /AC=0, L=1,
5201 1123 TAD K7740
5202 7403 ACS /AC TO SC
5203 7430 SEL
5204 7440 SEA
5205 7402 HLT /ACS CLEARED LINK OR ACS FAILED TO CLEAR AC,
5206 7441 SCA /SC TO AC
5207 7440 SEA /
5210 7402 HLT /ACS FAILED TO LOAD THE STEP COUNTER WITH 0.

/TEST OF THE GT FLAG,

5211 7431 GTTST1, SWAB /B MODE.
5212 7300 CLA CLL
5213 4954 JMS I RTFX /RESTORE FLAGS; WE'RE ONLY CONCERNED WITH THE GT.
5214 0004 GTF /GET THE FLAGS;
5215 0377 AND (2000 /SAVE THE GT FLAG,
    
```



```

5216 7886      RTL
5217 7430      SEL
5220 7482      HLT

5221 7431      GTTST2, SWAB
5222 7332      CLA CLL CML RTR
5223 4954      JMS I RTFX
5224 6884      GTF
5229 8377      AND (2800
5226 7886      RTL
5227 7420      SNL
5230 7482      HLT

5231 7431      GTTST3, SWAB
5232 7332      CLA CLL
5233 4954      JMS I RTFX
5234 6884      SGT
5235 7410      SKP
5236 7482      HLT

5237 7431      GTTST4, SWAB
5240 7332      CLA CLL CML RTR
5241 4954      JMS I RTFX
5242 6886      SGT
5243 7482      HLT

5244 7431      GTTST5, SWAB
5245 7332      CLA CLL CML RTR
5246 4954      JMS I RTFX
5247 7447      SWBA
5249 6886      SGT
5250 7610      SKP CLA
5251 7482      HLT
5252 4946      JMS I TSTSW3
5253 5776      JMP MDTST
5254 5776      ISZ COUNT
5255 5776      JMP MDTST
5256 6887      CAF
5257 5775      JMP MQLT
    
```

```

/PUT THE GT FLAG INTO THE LINK;
/GT A 1?
/YES, RTF FAILED TO REESE? GT OR
/GTF FAILED TO GET IT;
/B MODE;
/2800
/RESTORE FLAGS; SET GT TO A 1.
/GET THE FLAGS;
/SAVE ONLY THE GT FLAG;
/PUT IT IN THE LINK TO CHECK;
/LINK A ZERO INDICATING GT WAS A 0?
/YES, RTF FAILED TO SET GT OR GTF
/FAILED TO GET IT;
/B MODE;

/RESTORE THE FLAGS;
/GT FLAG = 0?
/YES, OK
/SGT SKIPPED ON NO GT FLAG;

/MODE B;
/2800
/RESTORE THE FLAGS;
/GT FLAG = 1?
/SGT DID NOT SKIP WITH GT = 1

/MODE B;
/2800
/RESTORE THE FLAGS;
/GO TO A MODE;
/GT FLAG SET?
/NO, OK
/SWBA FAILED TO CLEAR THE GT FLAG;
/REMAIN IN THIS TEST?
/YES, SR3=1
/FINISHED TEST 4096 TIMES;
/REPEAT INITIAL TESTS
/CLEAR AND SET TO MODE "A";
/EXIT INITIAL TESTS;
    
```

/END OF TEST PRINT-OUT ROUTINE.

```

5261 4524      E3A, JMS I CRLF
5262 1115      TAD MODE
5263 7850      SNA CLA
5264 5267      JMP ,+3
5265 4450      JMS I XTYPST
5266 7332      KEBS?1
5267 1115      TAD MODE
5270 7148      CMA CLL
5271 3115      DCA MODE
5272 6887      CAF
5273 5774      JMP MQLT=2
    
```

```

/CR AND LF;
/GET THE MODE; 0="A"; 7777="B";
/WHICH MODE?
/A MODE SO DON'T PRINT;
/TYPE A MESSAGE
/"KEB 1"
/GET MODE
/CHANGE IT TO THE OPPOSITE MODE.
/RESET MODE.
/INITIALIZE
/START FROM THE BEGINNING
    
```

/ROUTINE TO SELECT MODE.

```

5274 8080      MOSEL, OPEN
5275 7884      LAS
5276 7112      CLL RTR
5277 7430      SEL
5280 5311      JMP S10SET
5281 7208      CLA
5282 1115      TAD MODE
5283 7640      SNA CLA
5284 5307      JMP ,+3
5285 7447      SWBA
5286 5674      JMP I MOSEL
5287 7431      SWAB
5288 5674      JMP I MOSEL
5289 5674      S10SET, SPA CLA
5290 7710      JMP ,+3
5291 5315      DCA MODE
5292 3115      JMP MOSEL+5
5293 5301      JMP CMA
5294 7148      CLL CMA
5295 5313      JMP ,+3
    
```

```

/READ THE SWITCHES;
/SR10 TO LINK; SP11 TO AC0;
/SR 10 SET?
/YES;

/GET MODE
/WHICH MODE?
/"B" MODE
/SET TO "A" MODE
/EXIT SET TO "A" MODE;
/SET TO "B" MODE
/EXIT SET TO "B" MODE;
/EXIT SET TO "R" MODE;
/SR11=0?
/NO, SR11=1, SO SELECT MODE "B";
/SELECT MODE "A"
/EXIT;
/7777 IN ORDER TO SELECT "B" MODE;
/EXIT;
    
```

/ROUTINE TO EXIT TEST IF MODE "A" IS SELECTED;

```

5317 8080      UONLYB, OPEN
5320 7200      CLA
5321 1115      TAD MODE
5322 7780      SNA CLA
5323 5575      JMP I EGEN+3
5324 5717      JMP I UONLYB
    
```

/ROUTINE TO CHECK SR0;

```

5325 8080      SH0TST, OPEN
5326 7884      LAS
5327 7710      SPA CLA
5330 5725      JMP I SH0TST
5331 2325      ISZ SH0TST
5332 5725      JMP I SH0TST
    
```

/ROUTINE TO CHECK SR1;

```

5333 8080      SH1TST, OPEN
5334 7884      LAS
5335 7884      RAL
5336 7710      SPA CLA
5337 5733      JMP I SH1TST
5340 2333      ISZ SH1TST
5341 5733      JMP I SH1TST
    
```

/ROUTINE TO CHECK SR2;

```

5342 8080      SH2TST, OPEN
5343 7884      LAS
5344 7106      RTL CLL
5345 7710      SPA CLA
    
```

```

5346 5742      JMP I SW2TST
5347 2342      ISE SW2TST
5350 5742      JMP I SW2TST

```

```

5374 0202
5375 0204
5376 5001
5377 2000
5400 5400

```

PAGE

/ROUTINE TO CHECK SR3,

```

5400 0000      SW3TST, OPEN
5401 7604      LAS
5402 7106      RTL CLL
5403 7104      RAL CLL
5404 7710      SPA CLA
5405 5600      JMP I SW3TST
5406 2200      ISE SW3TST
5407 5600      JMP I SW3TST

```

/ROUTINE TO SAVE REGISTERS,

```

5410 0000      USVREG, OPEN
5411 3034      DCA ACTOCK /SAVE AC
5412 7701      CLA MQA /SAVE MQ
5413 3035      DCA MQTOCK
5414 7210      CLA RAR /SAVE LINK
5415 3033      DCA LKTOCK
5416 7641      CLA SCA /SAVE STEP COUNTER
5417 3036      DCA SCTOCK
5420 6004      GTF
5421 0377      AND (2000
5422 7104      CLL RAL /SAVE GTFLAG
5423 3037      DCA GTTOCK
5424 5610      JMP I USVREG /EXIT, AC=0

```

```

5425 0000      NMIXX, 0
5426 4776      JMS PNORM
5427 5625      JMP I NMIXX

```

```

5430 0000      SCATXX, 0
5431 4237      JMS PSTEP
5432 4250      JMS PSTEPT
5433 5630      JMP I SCATXX

```

```

5434 0000      SCAXX, 0
5435 4237      JMS PSTEP
5436 5634      JMP I SCAXX

```

```

5437 0000      PSTEP, 0
5440 7240      CLA CMA
5441 0300      AND N+4
5442 4526      JMS I PRXLOP /PRINT,

```

```

5443 1301      TAD N+5
5444 4526      JMS I PRXLOP /PRINT,
5445 1302      TAD N+6
5446 4526      JMS I PRXLOP /PRINT,
5447 5637      JMP I PSTEP

```

```

5450 0000      PSTEPT, 0
5451 7240      CLA CMA
5452 0277      AND N+3
5453 4526      JMS I PRXLOP /PRINT,
5454 5650      JMP I PSTEPT

```

```

5455 0000      PC, OPEN
5456 7200      CLA
5457 1077      TAD C
5460 4526      JMS I PRXLOP /EXIT,
5461 5655      JMP I PC

```

```

5462 0000      LPAR, OPEN
5463 7200      CLA
5464 1375      TAD (0250 /
5465 4526      JMS I PRXLOP /EXIT,
5466 5662      JMP I LPAR

```

```

5467 0000      RPAR, OPEN
5470 7200      CLA
5471 1374      TAD (251
5472 4526      JMS I PRXLOP /EXIT,
5473 5667      JMP I RPAR

```

```

5474 0316      N, 0316 /N
5475 0315      0315 /M
5476 0311      0311 /I
5477 0324      0324 /T
5500 0323      0323 /S
5501 0303      0303 /C
5502 0301      0301 /A

```

/ROUTINE TO LOAD THE STEP COUNTER,

```

5503 0000      ULDSC, OPEN
5504 1115      TAD MODE
5505 7640      SEA CLA
5506 5315      JMP LDSC1
5507 1024      TAD TOSHIF
5510 7040      CMA
5511 3313      DCA ,+2
5512 7403      SCL
5513 0000      OPEN
5514 5703      JMP I ULDSC
5515 1024      LDSC1, TAD TOSHIF
5516 7403      ACS
5517 5703      JMP I ULDSC

```

/ROUTINE TO SET COMPARE ROUTINE AND DO OTHER JOBS;

```

5520 0000   SCOMP, OPEN
5521 7344       CLA CLL CMA RAL
5522 3120       DCA          TWICE
5523 4554       JMS I    RTFX          /CLEAR THE GT
5524 1170       TAD          [LSIM
5525 3773       DCA          C1
5526 1167       TAD          [LKTOCK
5527 3772       DCA          C2
5528 3114       DCA          SCOUNT
5529 3021       DCA          TOLINK
5530 3771       DCA          WILCHG          /MAKE "WILCHG" EFFECTIVLY AN AND 0
5531 3770       DCA          PRCHG          /WHICH IS REALLY A DO NOTHING INSTRUCTION;
5532 3770       DCA          PRCHG          /MAKE "PRCHG" EFFECTIVLY AN AND 0
5533 3770       DCA          PRCHG          /WHICH IS REALLY A DO NOTHING INSTRUCTION;
5534 4453       JMS I    UMOVE
5535 0021       TOLINK
5536 0022       TOAC
5537 7753       -25
5540 5720       JMP I    SCOMP          /EXIT, AC=0, L=0;
    
```

/ROUTINE FOR DPSZ PRINTOUT

```

5541 0000   DPSZPR, OPEN
5542 4925       JMS I    CRLF2          /2 CR AND LF;
5543 4450       JMS I    XTYPST
5544 0000   DPZPR1, OPEN
5545 5741       JMP I    DPSZPR          /EXIT,
    
```

/UP-COUNT GENERATOR

```

5546 0000   GEN, 0
5547 2065       ISZ GENX
5548 5746       JMP I    GEN
5549 7604       CLA OSR          /TEST SW 3
5550 7106       RTL CLL
5551 7006       RTL
5552 7630       SZL CLA
5553 5456       JMP I    BACK
5554 5457       JMP I    NEXT
    
```

```

5570 7002
5571 7016
5572 6371
5573 6370
5574 0251
5575 0250
5576 4132
5577 2000
5600 5600
    
```

PAGE

/ROUTINE TO DO A CR AND LF;

```

5600 0000   UCRLF, OPEN
5601 7240       CLA CMA
5602 0070       AND CR          /CR
5603 4526       JMS I    PRXLOP          /PRINT,
5604 1071       TAD          LF          /"LINE FEED";
5605 4526       JMS I    PRXLOP          /PRINT,
5606 5600       JMP I    UCRLF          /EXIT,
    
```

/ROUTINE TO DO 2 CR AND LF;

```

5607 0000   UCRLF2, OPEN
5610 4524       JMS I    CRLF
5611 4924       JMS I    CRLF
5612 5607       JMP I    UCRLF2          /EXIT, AC = 0;
    
```

```

5613 0000   RXLOP, OPEN
5614 3236       DCA          CHAR          /SAVE CHARACTER TO PRINTER;
5615 1020       TAD          FILLER          /GET NUMBER OF FILLER CHARACTERS,
5616 7040       CMA          /COMPLEMENT,
5617 3237       DCA          FILCNT          /SAVE COMPLEMENTED NUMBER OF FILLER CHAR,
5618 1236       TAD          CHAR          /GET CHARACTER TO BE PRINTED,
5619 6046       TLS
5620 6041       TSF
5621 5222       JMP          ,=1
5622 1166       TAD          [=-215
5623 7640       SZA CLA
5624 5613       JMP I    RXLOP
5625 2237       ISZ          FILCNT
5626 7610       SKP CLA
5627 5613       JMP I    RXLOP
5628 6046       TLS
5629 6041       TSF
5630 5233       JMP          ,=1
5631 5227       JMP          ,=0
5632 0000   CHAR, OPEN
5633 0000   FILCNT, OPEN
    
```

```

5640 0000   UPLINK, 0
5641 7240       CLA CMA
5642 0102       AND LINK          /LINK
5643 4245       JMS UONZER
5644 5640       JMP I    UPLINK
    
```

```

5645 0000   UONZER, 0
5646 7440       SZA
5647 5252       JMP UONEP          /PRINT ONE
5648 4256       JMS UZEROR          /PRINT ZERO
5649 5645       JMP I    UONZER
    
```

```

5652 7240   UONEP, CLA CMA          /ONE
5653 0100       AND ONE
5654 4526       JMS I    PRXLOP          /PRINT,
5655 5645       JMP I    UONZER
    
```

```

5656 0000   UZEROR, 0
    
```

```

5657 7240      CLA CMA
5660 0101      AND ZERO      /ZERO
5661 4526      JMS I PRXLOP  /PRINT,
5662 5656      JMP I UZEROR

5663 0000      UMESSG, OPEN
5664 7240      CLA CMA
5665 0104      AND COUNTX
5666 3105      DCA STRCNT
5667 2105      ISE STRCNT
5670 7410      SKP
5671 5663      JMP I UMESSG  /EXIT,
5672 7240      CLA CMA
5673 0106      AND BITSTR
5674 7100      CLL
5675 7004      RAL
5676 3106      DCA BITSTR
5677 7430      SEL
5678 5303      JMP UPRONE
5679 4256      JMS UZEROR
5682 5267      JMP ,+13

5683 7240      UPRONE, CLA CMA
5684 0100      AND ONE
5685 4526      JMS I PRXLOP  /ONE
5686 5267      JMP UMESSG+4  /PRINT,

5687 0000      UTYTST, OPEN
5688 4525      JMS I CRLF2  /CR AND LF,
5689 1707      TAD I UTYTST /OBTAIN NUMBER OF WORDS
5690 3116      DCA ANYUSE  /SAVE FOR DURATION OF THIS ROUTINE,
5691 2307      ISE UTYTST /SET UP TO GET NEXT WORD,
5692 1707      TAD I UTYTST /GET THE WORD,
5693 3317      DCA ,+2    /STASH IT AWAY,
5694 4450      JMS I XTYPST /NOW PRINT THE WORD JUST STASHED AWAY,
5695 0000      OPEN
5696 4450      JMS I U2SPC /WORD TO BE PRINTED,
5697 2116      ISE ANYUSE /2 SPACES,
5698 5313      JMP ,+7    /DONE ALL THE WORDS SET UP FOR?
5699 4454      JMS I U1SPC /NO, REPEAT,
5700 2307      ISE UTYTST /YES, 1 SPACE,
5701 5707      JMP I UTYTST /SET UP FOR EXIT,
5702 5707      /EXIT THIS ROUTINE, AC EQUALS ZERO.

5726 0000      UP1BIT, OPEN
5727 3102      DCA LINK
5728 4527      JMS I PLINK
5729 5726      JMP I UP1BIT  /EXIT,

/Routine to load registers with random data,

5732 0000      RANDAT, OPEN
5733 4550      JMS I RANDOM /GET RANDOM DATA,
5734 3022      DCA TOAC   /SAVE FOR THE AC
5735 7010      RAR
5736 7010      /LINK TO ACB

```

```

5736 3021      DCA TOLINK /SAVE FOR THE LINK
5737 4550      JMS I RANDOM /GET RANDOM DATA
5740 3023      DCA TOMQ   /SAVE FOR THE MQ
5741 7010      RAR
5742 3025      DCA TOGT   /SAVE FOR THE GT
5743 4550      JMS I RANDOM /GET RANDOM DATA
5744 0105      AND C37    /KEEP AC 7=11
5745 3024      DCA TOSHIF /SAVE FOR THE STEP COUNTER
5746 4573      JMS I DCCOUNT
5747 5732      JMP I RANDAT
5748 5575      JMP I CGEN+3

/Routine to load MQ, GT, and LINK,

5751 0000      ULDREG, OPEN
5752 7300      CLA CLL
5753 1023      TAD TOMQ
5754 7421      MQL
5755 4553      JMS I LDGT  /MQ LOADED,
5756 1021      TAD TOLINK /LOAD THE GT
5757 7104      CLL RAL
5758 5751      JMP I ULDREG /LINK LOADED,

/Routine to load the GT,

5761 0000      ULDGT, OPEN
5762 7200      CLA
5763 1025      TAD TOGT   /GET THE GT DATA
5764 7110      CLL RAR
5765 4554      JMS I RTFX  /MOVE TO AC1
5766 5761      JMP I ULDGT  /NOW LOAD
5767 5761      /EXIT

/Routine to do a test 8192 octal times,

5767 0000      DCCOUNT, OPEN
5768 2114      ISE SCOUNT
5769 5767      JMP I DCCOUNT
5770 2120      ISE TWICE
5771 5767      JMP I DCCOUNT
5772 2367      ISE DCCOUNT
5773 5767      JMP I DCCOUNT
5774 2367      ISE DCCOUNT
5775 5767      JMP I DCCOUNT

6000          PAGE

/SUBROUTINE TO LOAD GT,

6000 0000      XRTF, OPEN
6001 3116      DCA ANYUSE /SAVE DATA TO BE PLACED IN THE GT,
6002 4214      ROP
6003 7112      CLL RTR
6004 7010      RAR
6005 6224      RIF
6006 1110      TAD ANYUSE /READ THE INSTRUCTION FIELD,
6007 6005      RTF
6008 6005      /GT DATA IN AC ALONG WITH IF AND DF,
6009 6005      /RESTORE THE FLAGS,

```

```

6010 0002 IOF /DO AWAY WITH THE ION CAUSED BY RTF.
6011 7300 CLA CLL /CLEAR FOR THE EXIT.
6012 5600 JMP I XRTF /EXIT, AC AND LINK ARE ZERO.

```

/ROUTINE TO SIMULATE THE SUBTRACT AC FROM MQ INSTRUCTION,

```

6013 0000 SAMSIM, OPEN
6014 1022 TAD TOAC /GET ORIGINAL AC.
6015 7500 SMA /IS IT NEGATIVE?
6016 7120 CLL CML /NO, SET LINK.
6017 7041 CMA IAC /NEGATE AC.
6020 3040 DCA TEMPA /SAVE IT.
6021 1023 TAD TOMQ /GET ORIGINAL MQ.
6022 7510 SPA /IS IT POSITIVE?
6023 7020 CML /NO, COMPLEMENT LINK.
6024 1040 TAD TEMPA /ADD NEGATED ORIGINAL AC.
6025 7230 CLA CML RAR /COMPLEMENT LINK AND MOVE TO AC 0.
6026 3046 DCA GTSIM /SAVE SIMULATED GT.
6027 1022 TAD TOAC /GET ORIGINAL AC.
6030 7041 CMA IAC /NEGATE IT.
6031 1023 TAD TOMQ /ADD ORIGINAL MQ TO IT.
6032 3043 DCA MSH /SAVE AS SIMULATED AC.
6033 7010 RAR /MOVE LINK INTO AC 0.
6034 3042 DCA LSH /SAVE AS SIMULATED LINK.
6035 1023 TAD TOMQ /GET ORIGINAL MQ.
6036 3044 DCA LSH /SAVE AS SIMULATED MQ.
6037 1024 TAD TOSHIF /GET SC DATA.
6040 3045 DCA SCSIM /SAVE AS SIMULATED SC.
6041 5613 JMP I SAMSIM /EXIT, L+AC=0.

```

/ROUTINE TO SIMULATE THE SHIFT LEFT INSTRUCTION,

```

6042 0000 SHLSIM, OPEN /DOUBLE PRECISION SHIFT LEFT.
6043 1024 TAD TOSHIF /
6044 1115 TAD MODE /
6045 7140 CMA CLL /
6046 3045 DCA SCSIM /SAVE THE NUMBER OF SHIFTS TO BE PERFORMED.
6047 1022 TAD TOAC /GET ORIGINAL AC
6050 3043 DCA MSH /MOST SIGNIFICANT HALF
6051 1023 TAD TOMQ /GET ORIGINAL MQ
6052 3044 DCA LSH /LEAST SIGNIFICANT HALF
6053 1025 TAD TOGT /GET ORIGINAL GT
6054 0115 AND MODE /"AND" THE MODE
6055 3046 DCA GTSIM /SAVE AS THE SIMULATED GT.
6056 1045 TAD SCSIM /GET STEP COUNTER DATA
6057 1377 TAD (32 /ADD 32.
6060 7710 SPA CLA /IF MORE THAN 31 SHIFTS THE RESULTS ARE PREDICTABLE
6061 5307 JMP SHL031 /GO TO ROUTINE FOR MORE THAN 31 SHIFTS.
6062 1045 TAD SCSIM /GET STEP COUNTER DATA
6063 7650 SNA CLA /IF 0 SHIFTS THE RESULTS ARE PREDICTABLE
6064 5313 JMP SHIFTO /GO TO ROUTINE FOR 0 SHIFTS
6065 1044 TAD LSH /GET CONTENTS OF LSH

```

```

6066 7421 MQL MSH /STORE IN THE MQ
6067 1043 TAD MSH /LOAD AC WITH CONTENTS OF MSH.
6070 7521 SWP /SWAP THE AC AND MQ
6071 7104 RAL CLL /SHIFT LEFT 1.
6072 7521 SWP /SWAP AC AND MQ.
6073 7004 RAL /SHIFT LEFT 1.
6074 2045 SHLA, ISE SCSIM /DONE NECESSARY SHIFTS?
6075 5270 JMP ,=5 /NO.
6076 3043 DCA MSH /SAVE SHIFTED AC.
6077 7501 MQA /MQ TO AC
6080 3044 DCA LSH /SAVE SHIFTED MQ.
6081 7210 CLA RAR /MOVE LINK INTO AC0.
6082 3042 DCA LSH /SAVE AS SIMULATED LINK
6083 1115 TAD MODE /MODE IN AC, 7777=0, 0000=A.
6084 0165 AND (37 /"AND" WITH A 37
6085 3045 DCA SCSIM /SAVE AS SIMULATED SC.
6086 5642 JMP I SHLSIM /EXIT WITH MOST SIGNIFICANT HALF IN "MSH"
/AND LEAST SIGNIFICANT HALF IN "LSH".

```

```

6107 7340 SHL031, CLA CMA CLL /7777 TO
6110 3045 DCA SCSIM /SCSIM SO AN IMMEDIATE EXIT TAKES PLACE
6111 7421 MQL /0 TO MQ.
6112 5274 JMP SHLA /CONTINUE IN MAIN BODY.
6113 1021 SHIFTO, TAD TOLINK /GET ORIGINAL LINK
6114 3042 DCA LSH /SAVE AS SIMULATED LINK
6115 1165 TAD (37 /37
6116 3045 DCA SCSIM /SAVE AS SIMULATED SC.
6117 5642 JMP I SHLSIM /EXIT, AC=0.

```

/ROUTINE TO SIMULATE THE LOGICAL SHIFT RIGHT INSTRUCTION,

```

6120 0000 LRSIM, OPEN /DOUBLE PRECISION LOGICAL SHIFT RIGHT
6121 1024 TAD TOSHIF /
6122 1115 TAD MODE /
6123 7140 CLL CMA /
6124 3045 DCA SCSIM /SAVE THE NUMBER OF SHIFTS TO BE PERFORMED.
6125 1045 TAD SCSIM /GET SHIFT DATA.
6126 1164 TAD (31 /ADD 31 TO IT
6127 7710 SPA CLA /IF MORE THAN 30 SHIFTS THE RESULTS ARE PREDICTABLE.
6130 5367 JMP LRS030 /GO TO ROUTINE FOR MORE THAN 30 SHIFTS.
6131 1022 TAD TOAC /GET ORIGINAL AC.
6132 3043 DCA MSH /STORE IN MSH
6133 1023 TAD TOMQ /GET ORIGINAL MQ.
6134 3044 DCA LSH /STORE IN LSH
6135 1045 TAD SCSIM /GET SHIFT DATA
6136 7650 SNA CLA /IF 0 SHIFTS, THE RESULTS ARE PREDICTABLE.
6137 9364 JMP LRS0SH+4 /JUST DO THE NECESSARY AND EXIT.
6140 1044 TAD LSH /GET CONTENTS OF LSH
6141 7421 MQL /STORE IN THE MQ.
6142 1043 TAD MSH /CONTENTS OF MSH TO AC.
6143 7110 CLL RAR /SHIFT RIGHT ONCE.
6144 7521 SWP /SWAP THE AC AND MQ.
6145 7010 RAR /SHIFT RIGHT ONCE.
6146 7521 SWP /SWAP THE AC AND MQ.

```

```

6147 2045 LSRA, ISZ SCSIM /DONE NECESSARY SHIFTS?
6150 5343 JMP ,=5 /NO,
6151 3043 DCA MSH /SAVE THE SHIFTED AC,
6152 7501 MQA /
6153 3044 DCA LSH /SAVE THE SHIFTED MQ,
6154 3042 DCA LSHM /SAVE SIMULATED LINK
6155 7210 CLA RAR /
6156 0115 AND MODE /
6157 3046 DCA GTSIM /SAVE SIMULATED GT
6160 1165 LSR0SH, TAD [37 /37
6161 0115 AND MODE /"AND" MODE,
6162 3045 DCA SCSIM /SAVE AS SIMULATED SC,
6163 5720 JMP I LSRSIM /EXIT, AC=0,
6164 1025 TAD TOGT /
6165 3046 DCA GTSIM /GT REMAINS SAME ON 0 SHIFTS,
6166 5360 JMP LSR0SH /CONTINUE IN MAIN BODY,

6167 7340 LSRO30, CLA CLL CMA
6170 3045 DCA SCSIM
6171 7421 MQL
6172 5347 JMP LSRA

6177 0032 PAGE
6200

```

/ROUTINE TO SIMULATE THE ARITHMETIC SHIFT RIGHT INSTRUCTION,

```

6200 0000 ASRSIM, OPEN
6201 1024 TAD TOSHIF /
6202 1115 TAD MODE /
6203 7140 CLL CMA /
6204 3045 DCA SCSIM /SAVE THE NUMBER OF SHIFTS TO BE PERFORMED,
6205 1022 TAD TOAC /GET ORIGINAL AC
6206 3043 DCA MSH /MOST SIGNIFICANT HALF,
6207 1023 TAD TOMQ /GET ORIGINAL MQ,
6210 3044 DCA LSH /LEAST SIGNIFICANT HALF,
6211 1045 TAD SCSIM /GET SHIFT DATA
6212 7650 SNA CLA /IF 0 SHIFTS THE RESULTS ARE PREDICTABLE
6213 5251 JMP ASR0SH /JUST DO THE NECESSARY AND EXIT,
6214 1045 TAD SCSIM /GET SHIFT DATA
6215 1164 TAD [31 /31
6216 7710 SPA CLA /IF MORE THAN 30 SHIFTS, THE RESULTS ARE PREDICTABLE,
6217 5257 JMP ASRO30 /GO TO ROUTINE FOR MORE THAN 30 SHIFTS
6220 1044 TAD LSH /CONTENTS OF LSH TO AC,
6221 7421 MQL /LOAD IT IN THE MQ NOW,
6222 1043 TAD MSH /CONTENTS OF MSH TO AC,
6223 7100 SASR1, CLL /CLEAR THE LINK,
6224 7510 SPA /AC0=1?
6225 7020 CML /YES, SET THE LINK,
6226 7010 RAR /SHIFT RIGHT ONCE,
6227 7521 SWP /SWAP AC AND MQ,
6230 7010 RAR /SHIFT RIGHT ONCE,
6231 7521 SWP /SWAP AC AND MQ,
6232 2045 ISZ SCSIM /DONE NECESSARY SHIFTS?

```

```

6233 5223 JMP SASR1 /NO,
6234 3043 DCA MSH /STORE SHIFTED AC,
6235 7501 MQA /MQ TO AC,
6236 3044 DCA LSH /STORE SHIFTED MQ,
6237 7210 SASR1A, CLA RAR /LINK TO AC0
6240 0115 AND MODE /
6241 3046 SASR2, DCA GTSIM /SAVE AS SIMULATED GT
6242 1043 TAD MSH /GET MSH
6243 0163 AND [4000 /KEEP AC0
6244 3042 DCA LSHM /SAVE AS SIMULATED LINK
6245 1165 SASR3, TAD [37 /37
6246 0115 AND MODE /"AND" MODE "A"=0, "R"=7777
6247 3045 DCA SCSIM /SAVE AS SIMULATED SC
6250 5000 JMP I ASRSIM /EXIT, AC=0
6251 1022 ASR0SH, TAD TOAC /GET ORIGINAL AC
6252 0163 AND [4000 /KEEP AC0 ONLY
6253 3042 DCA LSHM /SAVE AS SIMULATED LINK
6254 1025 TAD TOGT /GET ORIGINAL GT
6255 3046 DCA GTSIM /SAVE AS SIMULATED GT
6256 5245 JMP SASR3 /CONTINUE IN MAIN BODY

6257 1043 ASRO30, TAD MSH /GET MSH
6260 0163 AND [4000 /KEEP AC0
6261 7104 CLL RAL /PUT INTO LINK
6262 7620 SNL CLA /LINK=1? INDICATING AC0=1,
6263 5271 JMP ,+6 /NO, AC0 WAS A 0
6264 7040 CMA /YES, SET AC=7777
6265 3044 DCA LSH /SAVE AS SIMULATED MQ
6266 7040 CMA /7777
6267 3043 DCA MSH /SAVE AS SIMULATED AC
6270 5237 JMP SASR1A /CONTINUE IN MAIN BODY
6271 3044 DCA LSH /SAVE AS SIMULATED MQ
6272 5267 JMP ,=3 /SET SIMULATED AC

```

/ROUTINE TO SIMULATE THE DOUBLE PRECISION INCREMENT INSTRUCTION,

```

6273 0000 DPISIM, OPEN
6274 1023 TAD TOMQ /SIMULATE DPI INSTRUCTION,
6275 7101 CLL IAC /GET WHAT WAS LOADED INTO THE MQ
6276 3044 DCA LSH /INCREMENT IT
6277 7004 RAL /SAVE IT AS "SIMULATED MQ"
6300 1022 TAD TOAC /LINK TO AC11 AC0 TO LINK,
6301 3043 DCA MSH /ADD WHAT WAS LOADED INTO THE AC
6302 7010 RAR /STORE IT AS "SIMULATED AC"
6303 3042 DCA LSHM /PUT LINK INTO AC0
6304 1025 TAD TOGT /STORE IT AS "SIMULATED LINK"
6305 3046 DCA GTSIM /GET PROBLEM GT
6306 1024 TAD TOSHIF /STORE IN SIMULATED GT ALSO
6307 3045 DCA SCSIM /GET PROBLEM STEP COUNTER
6310 5673 JMP I DPISIM /STORE IN SIMULATED SC,
/EXIT, AC AND LINK=0

```

/ROUTINE TO SIMULATE THE DOUBLE PRECISION COMPLEMENT INSTRUCTION,

```

6311 0000 DCMSIM, OPEN /DOUBLE PRECISION 2'S COMPLEMENT SIMULATOR,
6312 1023 TAD TOMQ /GET WHAT WAS LOADED INTO THE MQ

```

```

6313 7041 CMA IAC /2'S COMPLEMENT IT
6314 3044 DCA LSH /SAVE AS SIM MQ
6315 1022 TAD TOAC /GET WHAT WAS LOADED INTO THE AC
6316 7040 CMA /1'S COMPLEMENT IT
6317 3043 DCA MSH /SAVE IT
6320 7004 RAL /GET THE CARRY FROM 2'S COMPLEMENT OF MQ
6321 1043 TAD MSH /ADD 1'S COMPLEMENT OF AC
6322 3043 DCA MSH /STORE AS 2'S COMPLEMENT OF AC
6323 7010 RAR /GET LINK
6324 3042 DCA LSIM /SAVE IT
6325 1025 TAD TOCT /GET PROBLEM GT
6326 3046 DCA GTSIM /SAVE AS SIMULATED GT
6327 1024 TAD TOSHIF /GET PROBLEM STEP COUNTER
6330 3045 DCA SCSIM /SAVE AS SIMULATED SC
6331 5711 JMP I DCMSIM /EXIT, AC=0
    
```

/ROUTINE TO SIMULATE THE DOUBLE PRECISION ADD INSTRUCTION

```

6332 0000 DADSIM, OPEN /DOUBLE PRECISION ADD ROUTINE,
6333 1023 TAD TOMQ /GET ORIGINAL MQ
6334 1025 TAD TOCT /ADD ORIGINAL DATA IN "LEAST SIGNIFICANT"
6335 3044 DCA LSH /SAVE THE DATA
6336 7204 CLA RAL /LINK TO AC 11
6337 1022 TAD TOAC /ADD ORIGINAL AC TO IT
6340 1024 TAD TOSHIF /ADD ORIGINAL DATA IN "MOST SIGNIFICANT" TO IT
6341 3043 DCA MSH /SAVE THE DATA
6342 7010 RAR /LINK TO AC 0
6343 3042 DCA LSIM /SAVE AS SIMULATED LINK
6344 5732 JMP I DADSIM /EXIT, AC=LINK=0
    
```

/ROUTINE TO COMPARE THE CONTENTS OF 2 LOCATIONS.

```

6345 0000 COMP, OPEN
6346 1745 TAD I COMP /GET AND STORE NUMBER OF
6347 3374 DCA CMPCTR /WORDS TO COMPARE,
6350 2345 ISE COMP
6351 1370 TAD C1 /ADDRESS IN C1
6352 3372 DCA C1A /C1A,
6353 1371 TAD C2 /ADDRESS IN C2
6354 3373 DCA C2A /TO C2A,
6355 1772 COMP, TAD I C1A /GET "GOOD" WORD
6356 7041 CIA /2'S COMPLEMENT IT
6357 1773 TAD I C2A /ADD RESULT WORD
6360 7040 SZA CLA /RESULT ZERO?
6361 5745 JMP I COMP /NO, EXIT AC=0
6362 2372 ISE C1A /YES, SET
6363 2373 ISE C2A /FOR NEXT COMPARE
6364 2374 ISE CMPCTR /DONE COMPARING
6365 5355 JMP COMPA /NO,
6366 2345 ISE COMP /YES
6367 5745 JMP I COMP /EXIT, AC=0
6370 0000 C1, OPEN /CONTAINS ADDRESS OF "GOOD"
    
```

```

6371 0000 C2, OPEN /CONTAINS ADDRESS OF DATA TO BE COMPARED
6372 0000 C1A, OPEN /WILL CONTAIN "GOOD" DATA
6373 0000 C2A, OPEN /WILL CONTAIN DATA TO BE COMPARED
6374 0000 CMPCTR, OPEN /COUNTER,
    
```

6400 PAGE

```

6400 0000 MOVE, 0
6401 7200 CLA
6402 1000 TAD I MOVE /GET "FROM ADDR" AND
6403 3223 DCA FADDR /STORE AT FDDR
6404 2000 ISE MOVE
6405 1000 TAD I MOVE /GET "TO ADDR" AND
6406 3224 DCA TADDR /STORE AT TDDR
6407 2000 ISE MOVE
6410 1000 TAD I MOVE /GET "MOVE COUNT" AND
6411 3225 DCA MCTR /STORE AT MCTR
6412 2000 ISE MOVE /SET UP EXIT ADDRESS
6413 7200 MOVEA, CLA
6414 1023 TAD I FADDR /GET "FROM" WORD
6415 3024 DCA I TADDR /STORE AT "TO" LOCATION
6416 2223 ISE FADDR /INCREMENT "FROM" ADDRESS
6417 2224 ISE TADDR /INCREMENT "TO" ADDRESS
6420 2225 ISE MCTR /ALL WORDS MOVED?
6421 5213 JMP MOVEA /NO,
6422 5000 JMP I MOVE /YES
    
```

```

6423 0000 FADDR, 0
6424 0000 TADDR, 0
6425 0000 MCTR, 0
    
```

/ROUTINE TO CONVERT FROM BINARY TO DECIMAL;

```

6426 0000 SDCNV, 0
6427 1377 TAD (-4
6430 3271 DCA CNVCTR
6431 1262 TAD ADDR2A
6432 3243 DCA ARROW /INITIALIZE ARROW
6433 1026 TAD I BDCNV
6434 2226 ISE BDCNV
6435 3270 DCA DIGIT
6436 1070 TAD I DIGIT
6437 3267 DCA VALUE
6440 3270 DCA DIGIT /CLEAR DIGIT
6441 7100 CLL
6442 1267 TAD VALUE
6443 1263 ARROW, TAD TENPWR
6444 7420 SNL
6445 5251 JMP (-4
6446 2270 ISE DIGIT /DEVELOP DIGIT
6447 3267 DCA VALUE
6450 5241 JMP ARROW=2
6451 7200 CLA
    
```

```

6492 1270 TAD DIGIT /GET DIGIT
6493 1272 TAD K260 /ADD 260
6494 4926 JMS I PRXLOP /PRINT
6495 7300 CLA CLL
6496 2243 ISE ARROW /POINT ARROW
6497 2271 ISE CNVCTR /DONE?
6480 5240 JMP ARROW=3 /NO, REPEAT
6481 5826 JMP I BDCNV /YES, EXIT
6462 1263 ADDRZA, TAD TENPWR
6463 6030 TENPWR, -1750
6464 7634 -144
6465 7766 -12
6466 7777 -1
6467 0000 VALUE, 0
6470 0000 DIGIT, 0
6471 0000 CNVCTR, 0
6472 0260 K260, 260
    
```

/ROUTINE TO CHANGE A BINARY NUMBER TO IT'S OBVERSE

```

6473 0000 OBVERS, OPEN /ENTER WITH "ABC DEF GHI JKL",
6474 7102 CLL BSW
6475 7421 MQL
6476 7501 MQA
6477 7812 RTR
6500 7810 RAR
6501 0376 AND (787)
6502 7521 SWP
6503 7106 CLL RTL
6504 7804 RAL
6505 0375 AND (7870)
6506 7501 MQA
6507 7421 MQL
6510 7501 MQA
6511 0374 AND (2222)
6512 3324 DCA OBV
6513 7501 MQA
6514 0373 AND (4444)
6515 7112 CLL RTR
6516 7521 SWP
6517 0372 AND (1111)
6520 7106 CLL RTL
6521 1324 TAD OBV
6522 7501 MQA
6523 9673 JMP I OBVERS /EXIT WITH "LKJ IMG FED CBA", AND LINK 0,
    
```

```

6524 0000 OBV, OPEN
/RANDOM NUMBER GENERATOR SUBROUTINE
    
```

```

6525 0000 RANGEN, 0
6526 7200 CLA
6527 1370 TAD RANTND
    
```

```

6530 1395 TAD RANDEX
6531 7640 SEA CLA
6532 9342 JMP RANTAD
6533 1397 TAD RANTBL
6534 3395 DCA RANDEX
6535 1396 TAD RANCON
6536 7104 CLL RAL
6537 7430 SEL
6540 7801 IAC
6541 3396 DCA
6542 1396 RANTAD, TAD RANCON
6543 1795 TAD I RANDEX
6544 3795 DCA I RANDEX
6545 1371 TAD RANSAV
6546 7810 RAR
6547 1795 TAD I RANDEX
6550 2395 ISE RANDEX
6551 7400 NOP
6552 3371 DCA RANSAV
6553 1371 TAD RANSAV
6554 5725 JMP I RANGEN
6555 6570 RANDEX, RANTND
6556 6543 RANCON, 6543
6557 6560 RANTBL, +1
6560 6543 +1
6561 3210 3210
6562 0765 0765
6563 5432 5432
6564 2107 2107
6565 7654 7654
6566 4321 4321
6567 0176 0176
6570 1210 RANTND, =
6571 0000 RANSAV, OPEN
    
```

```

6572 1111
6573 4444
6574 2222
6575 7070
6576 0707
6577 7774
6600 PAGE
/Routine FOR TYPING,
    
```

```

6600 0000 TYPST, OPEN
6601 7200 CLA
6602 1600 TAD I TYPST /GET INITIAL ADDRESS
6603 3263 DCA TEMQ /STORE INITIAL ADDRESS
6604 3265 DCA FLAG /CLEAR FLAG
6605 2200 ISE TYPST /PRESET UP EXIT
6606 1663 TSC1, TAD I TEMQ /PICK UP DATA
6607 7812 RTR
6610 7812 RTR
6611 7812 RTR
    
```



```

6612 4217      JMS      TSC2      /GO TYPE FIRST CHARACTER
6613 1663      TAD I    TEMQ      /PICK-UP DATA
6614 4217      JMS      TSC2      /GO TYPE SECOND CHARACTER
6615 2263      ISZ     TEMQ      /EVEN STRING ADDRESS
6616 5206      JMP     TSC1      /GO BACK FOR MORE
6617 0000      TSC2,   OPEN
6620 0377      AND     (77       /MASK OFF SIX BITS
6621 3264      DCA     TEMR      /SAVE CHARACTER,
6622 1265      TAD     FLAG      /TEST "SPECIAL" FLAG,
6623 7640      SZA     CLA
6624 5234      JMP     TYPSP     /SET! TYPE SPECIAL
6625 1264      TAD     TEMR      /NO, REGULAR CHARACTER
6626 7450      SNA
6627 5232      JMP     ,+3
6630 4253      TYPAT,  JMS     PRINT   /YES, SET FLAG
6631 5617      JMP I   TSC2     /NO, PRINT IT,
6632 2265      ISZ     FLAG     /RETURN
6633 5617      JMP I   TSC2     /SET "SPECIAL" FLAG
6634 3265      TYPSP,  DCA     FLAG     /EXIT
6635 1264      TAD     TEMR      /CLEAR FLAG,
6636 7041      CIA
6637 7450      SNA
6640 5230      JMP     TYPAT     /0!TYPE "0",
6641 7001      IAC
6642 7650      SNA     CLA
6643 5600      JMP I   TYPST     /TEST FOR 01;
6644 1266      TAD     SKIPMA    /YES! EXIT CODE
6645 3255      DCA     SWITCH    /ALTER INSTRUCTION
6646 1264      TAD     TEMR      /TO BE "SMA"
6647 4253      JMS     PRINT     /TYPE CHAR
6650 1267      TAD     SKIPPA    /ALTER INSTRUCTION
6651 3255      DCA     SWITCH    /TO BE "SPA"
6652 5617      JMP I   TSC2     /RETURN,
6653 0000      PRINT,  OPEN
6654 1376      TAD     (=40      /COMPARE WITH 40,
6655 7510      SWITCH, SPA      /OR SMA FOR SPECIAL CODES
6656 1375      TAD     (100
6657 1374      TAD     (240
6660 4526      JMS I   PRXLOP    /PRINT,
6661 5653      JMP I   PRINT
6662 0000      TEMPQ, OPEN
6663 0000      TEMQ,  OPEN
6664 0000      TEMR,  OPEN
6665 0000      FLAG,  OPEN
6666 7500      SKIPMA, SMA
6667 7510      SKIPPA, SPA

6670 0000      PSPC,  OPEN
6671 1670      TAD I   PSPC      /GET NUMBER
6672 3303      DCA     SPCTR     /OF SPACES
6673 2270      ISZ     PSPC
6674 4450      JMS I   XTYPST   /SPACE ONCE
6675 6701      ,+4

```

```

6676 2303      ISZ     SPCTR      /ALL SPACES DONE
6677 5274      JMP     ,+3
6700 5670      JMP I   PSPC      /NO, REPEAT,
6701 4000      4000
6702 0100      0100
6703 0000      SPCTR,  OPEN
6704 0000      /ROUTINE TO PRINT MODE FAILURE
6705 0000      TYMOD,  OPEN
6706 7300      CLA     CLL
6707 1115      TAD     MODE
6708 7040      CMA
6710 1373      TAD     (4002
6711 3321      DCA     MODEX+2
6712 4451      JMS I   UPSPC
6713 7774      -4
6714 4450      JMS I   XTYPST
6715 6717      MODEX
6716 5704      JMP I   TYMOD
6717 1517      MODEX,  1517
6720 0405      0405
6721 0000      OPEN
6722 0001      0001
6723 0000      /ROUTINE TO PRINT 1 SPACE
6724 4451      SPACE1, OPEN
6725 7777      JMS I   UPSPC
6726 5723      -1
6727 0000      JMP I   SPACE1
6728 0000      /ROUTINE TO PRINT 2 SPACES,
6729 0000      SPACE2, OPEN
6730 4451      JMS I   UPSPC
6731 7776      -2
6732 5727      JMP I   SPACE2
6733 0000      /ROUTINE TO PRINT A HEADING IN ERROR PRINTOUTS;
6734 4525      HEADIN, OPEN
6735 4451      JMS I   CRLF2    /2 CR AND LF;
6736 7764      JMS I   UPSPC     /#12 SPACES
6737 4450      -14
6738 7407      JMS I   XTYPST    /C(L)
6739 4451      CL
6740 7407      CL
6741 4451      JMS I   UPSPC    /5 SPACES
6742 7773      -5
6743 4450      JMS I   XTYPST    /C(AC)
6744 7377      CAC
6745 4451      JMS I   UPSPC    /9 SPACES
6746 7767      -11
6747 4450      JMS I   XTYPST    /C(MQ)

```

6750	7403	CMQ		
6751	4451	JMS I	UPSPC	
6752	7774	-4		
6753	4450	JMS I	XTYPST	/C(IGT)
6754	7456	CGT		
6755	4451	JMS I	UPSPC	/6 SPACES
6756	7772	-6		
6757	4450	JMS I	XTYPST	/C(SC)
6760	7431	CSC		
6761	5733	JMP I	HEADIN	/EXIT
6773	4002			
6774	0240			
6775	0100			
6776	7740			
6777	0077			
	7000		PAGE	
7000	0000	UPREGS,	OPEN	
7001	4576	JMS I	CTYMOD	
7002	0000	PRCHG,	OPEN	/THIS LOCATION WILL CONTAIN 0000 EXCEPT /WHEN THE DPSZ TEST ARE BEING EXECUTED AND /AT THAT TIME THERE WILL BE A JMS DPSZPR /IN THIS LOCATION, /PRINT HEADING /2 CR AND LF,
7003	4777/	JMS	HEADIN	
7004	4525	JMS I	CRLF2	
7005	4453	JMS I	UMOVE	
7006	0021	TOLINK		
7007	0026	TLINK		
7010	7773	-5		
7011	4450	JMS I	XTYPST	
7012	7412	PROBLM		
7013	4451	JMS I	UPSPC	
7014	7771	-7		
7015	4246	JMS	PLAMGS	/PRINT C(L),C(IAC),C(MQ),C(IGT),C(SC)
7016	0000	WILCHG,	OPEN	/THIS LOCATION WILL CONTAIN 0000 EXCEPT /WHEN THE DAD TESTS ARE BEING /EXECUTED, THEN THERE WILL RE A JMS FORDAD /IN THIS LOCATION, /2 CR AND LF; /PRINT "SIMULATED"
7017	4525	JMS I	CRLF2	
7020	4450	JMS I	XTYPST	
7021	7417	SIMULT		
7022	4451	JMS I	UPSPC	/5 SPACES
7023	7773	-5		
7024	4453	JMS I	UMOVE	
7025	0042	LSIM		
7026	0026	TLINK		
7027	7773	-5		
7030	4246	JMS	PLAMGS	
7031	4525	JMS I	CRLF2	/2 CR AND LF;
7032	4450	JMS I	XTYPST	/TYPE "ACTUAL"

7033	7425	ACTUAL		
7034	4451	JMS I	UPSPC	/7 SPACES
7035	7770	-10		
7036	4453	JMS I	UMOVE	
7037	0033	LKTOCK		
7040	0026	TLINK		
7041	7773	-5		
7042	4246	JMS	PLAMGS	
7043	5600	JMP I	UPREGS	
7044	4776/	DADJMS,	JMS FORDAD	
7045	4775/	DSEJMS,	JMS DPSZPR	
7046	0000	PLAMGS,	OPEN	
7047	1026	TAD	TLINK	
7050	4555	JMS I	P18IT	
7051	4455	JMS I	U2SPC	/2 SPACES
7052	1027	TAD	TAC	
7053	4774/	JMS	P12BIT	/PRINT CONTENTS OF AC,
7054	4455	JMS I	U2SPC	/2 SPACES
7055	1030	TAD	TMO	
7056	4774/	JMS	P12BIT	/PRINT CONTENTS OF MO,
7057	4451	JMS I	UPSPC	/3 SPACES
7060	7775	-3		
7061	1032	TAD	TGT	
7062	4555	JMS I	P18IT	
7063	4451	JMS I	UPSPC	/4 SPACES
7064	7774	-4		
7065	1031	TAD	TSHIF	
7066	4774/	JMS	P12BIT	/PRINT CONTENTS OF THE STEP COUNTER,
7067	5646	JMP I	PLAMGS	/EXIT, AC=0
/ROUTINE TO PRINT THE NUMBER OF SHIFTS IN DECIMAL;				
7070	0000	NUMSH,	OPEN	/PRINT NUMBER OF SHIFTS IN DECIMAL;
7071	4451	JMS I	UPSPC	
7072	7775	-3		
7073	1024	TAD	TOSHIF	
7074	7001	IAC		
7075	1115	TAD	MODE	
7076	3116	DCA	ANYUSE	
7077	4773/	JMS	BDCNV	
7100	0116	ANYUSE		
7101	4455	JMS I	U2SPC	
7102	4450	JMS I	XTYPST	
7103	7447	SHIFTS		
7104	4455	JMS I	U2SPC	
7105	5670	JMP I	NUMSH	/EXIT
/ROUTINE TO PRINT THE CONTENTS OF THE REGISTERS FOR THE DST INSTRUCTION;				
7106	0000	DSTREG,	OPEN	
7107	4576	JMS I	CTYMOD	
7110	4772/	JMS	DSTHED	/PRINT THE HEADING
7111	4525	JMS I	CRLF2	/2 CR AND LF,

```

7112 4450 JMS I XYPST /PRINT C(L)
7113 7407 CL
7114 4451 JMS I UPSPC /5 SPACES
7115 7773 -5
7116 1042 TAD LSIM
7117 4555 JMS I P1BIT /PRINT ORIGINAL LINK,
7120 4451 JMS I UPSPC /15 SPACES
7121 7761 -17
7122 1033 TAD LKTOCK
7123 4555 JMS I P1BIT /PRINT LINK AFTER EAE INSTRUCTION,
7124 4524 JMS I CRLF /CR AND LF;
7125 4450 JMS I XYPST /PRINT C(AC)
7126 7377 CAC
7127 4451 JMS I UPSPC /4 SPACES
7130 7774 -4
7131 1043 TAD MSH
7132 4774 JMS P12BIT /PRINT ORIGINAL AC
7133 4451 JMS I UPSPC /4 SPACES
7134 7774 -4
7135 1034 TAD ACTOCK
7136 4774 JMS P12BIT /PRINT AC AFTER DST
7137 4524 JMS I CRLF /CR AND LF;
7140 4450 JMS I XYPST /PRINT C(MSH)
7141 7514 CMSH
7142 4451 JMS I UPSPC /19 SPACES
7143 7755 -23
7144 1036 TAD SCTOCK
7145 4774 JMS P12BIT /PRINT THE STORED AC
7146 4524 JMS I CRLF /CR AND LF;
7147 4450 JMS I XYPST /PRINT C(MQ)
7150 7403 CMQ
7151 4451 JMS I UPSPC /4 SPACES
7152 7774 -4
7153 1044 TAD LSH
7154 4774 JMS P12BIT /PRINT ORIGINAL MQ
7155 4451 JMS I UPSPC /4 SPACES
7156 7774 -4
7157 1035 TAD MQTOCK
7160 4774 JMS P12BIT /PRINT MQ AFTER DST
7161 4524 JMS I CRLF /CR AND LF;
7162 4450 JMS I XYPST /PRINT C(LSH)
7163 7520 CLSH
7164 4451 JMS I UPSPC /19 SPACES
7165 7755 -23
7166 1037 TAD GTTOCK
7167 4774 JMS P12BIT /PRINT STORED MQ;
7170 5706 JMP I DSTREG /EXIT,

7172 7204
7173 6426
7174 7200
7175 5541
7176 7230
7177 6733
7200 PAGE

```

```

/ROUTINE TO PRINT THE 12 BITS OF A REGISTER,
7200 0000 P12BIT, OPEN
7201 3106 DCA BITSTR
7202 4531 JMS I MESSG /PRINT A MESSAGE;
7203 5600 JMP I P12BIT

/ROUTINE TO PRINT THE HEADING FOR THE DST INSTRUCTION,
7204 0000 DSTHD, OPEN
7205 4525 JMS I CRLF2 /2 CR AND LF;
7206 4454 JMS I U1SPC /1SPACE
7207 4450 JMS I XYPST /TYPE "REG"
7210 7527 REG
7211 4451 JMS I UPSPC /6 SPACES
7212 7772 -6
7213 4450 JMS I XYPST
7214 7504 BEFORE
7215 4454 JMS I U1SPC
7216 4450 JMS I XYPST
7217 7501 EDST
7220 4451 JMS I UPSPC
7221 7772 -6
7222 4450 JMS I XYPST
7223 7510 AFTER
7224 4454 JMS I U1SPC
7225 4450 JMS I XYPST
7226 7501 EDST
7227 5604 JMP I DSTHD /EXIT, AC = 0;

/ROUTINE TO TYPE THE DATA TO BE ADDED TO THE AC+MQ FOR THE DAD INSTRUCTION;
7230 0000 FORDAD, OPEN
7231 4525 JMS I CRLF2 /2 CR AND LF;
7232 4450 JMS I XYPST /PRINT "TO BE ADDED";
7233 7536 TOREAD
7234 4451 JMS I UPSPC /6 SPACES
7235 7772 -6
7236 1040 TAD TEMPA /MSH TO BE ADDED;
7237 4200 JMS P12BIT /PRINT THE MSH TO BE ADDED;
7240 4455 JMS I U2SPC /2 SPACES;
7241 1041 TAD TEMPB /LSH TO BE ADDED;
7242 4200 JMS P12BIT /PRINT THE LSH TO BE ADDED;
7243 5630 JMP I FORDAD /EXIT,

7244 0000 SANTAB, 0
7245 0000 0000
7246 7777 7777

7247 4000 4000
7250 7777 7777
7251 0000 0000

```

7252	0000	0
7253	7777	7777
7254	7777	7777
7255	0000	0
7256	0000	0000
7257	0000	0000
7260	0000	0
7261	0001	0001
7262	0002	0002
7263	0000	0
7264	3776	3776
7265	3777	3777
7266	4000	4000
7267	3777	3777
7270	3776	3776
7271	4000	4000
7272	4777	4777
7273	4776	4776
7274	0000	0
7275	4776	4776
7276	4777	4777
7277	4000	4000
7300	7777	7777
7301	3776	3776
7302	0000	0
7303	3776	3776
7304	7777	7777
7305	0000	DSTTAB, 0
7306	7777	7777
7307	7777	7777
7310	4000	4000
7311	0000	0000
7312	0000	0000
7313	4000	4000
7314	2525	2525
7315	5252	5252
7316	0000	0
7317	5252	5252
7320	2525	2525
7321	0000	0
7322	7007	7007

/K08 EAE INSTRUCTION TEST PART 1 MAINDEC=0E-00LB PAL10 V141 9-FEB-72 16126 PAGE 1075

7323	0770	0770
7324	4000	4000
7325	0770	0770
7326	7007	7007
7327	0000	DADTAB, 0
7330	0000	0000
7331	0000	0000
7332	0000	0000
7333	0000	0000
7334	4000	4000
7335	7777	7777
7336	7777	7777
7337	0000	0000
7340	0000	0000
7341	4000	4000
7342	0000	0000
7343	0000	0000
7344	7777	7777
7345	7777	7777
7346	0000	0
7347	2525	2525
7350	5252	5252
7351	5252	5252
7352	2525	2525
7353	4000	4000
7354	5252	5252
7355	2525	2525
7356	2525	2525
7357	5252	5252
7360	4000	4000
7361	0770	0770
7362	7007	7007
7363	7007	7007
7364	0770	0770
7365	0000	0
7366	7007	7007
7367	0770	0770
7370	0770	0770
7371	7007	7007
7372	0000	0
7373	7777	7777
7374	7777	7777
7375	7777	7777
7376	7777	7777

/MESSAGES:

7377	0350	CAC,	0350	/C(AC)
7400	0103		0103	
7401	5100		5100	
7402	0100		0100	
7403	0350	CMQ,	0350	/C(MQ)
7404	1521		1521	
7405	5100		5100	
7406	0100		0100	
7407	0350	C.,	0350	/C(L)
7410	1451		1451	
7411	0001		0001	
7412	2022	PROBLM,	2022	
7413	1702		1702	
7414	1405		1405	
7415	1500		1500	
7416	0100		0100	
7417	2311	SIMULT,	2311	/SIMULATED
7420	1525		1525	
7421	1401		1401	
7422	2405		2405	
7423	0400		0400	
7424	0100		0100	
7425	0103	ACTUAL,	0103	/ACTUAL
7426	2425		2425	
7427	0114		0114	
7430	0001		0001	
7431	0350	CSC,	0350	/C(SC)
7432	2303		2303	
7433	5100		5100	
7434	0100		0100	
7435	2310	ZSHL,	2310	/SHL
7436	1400		1400	
7437	0100		0100	
7440	2405	TEST,	2405	/TEST
7441	2324		2324	
7442	0001		0001	
7443	6000	ZER0,	6000	/0
7444	0100		0100	
7445	6100	ZDNE,	6100	/"1"
7446	0100		0100	
7447	2310	SHIFTS,	2310	/SHIFTS
7450	1106		1106	
7451	2423		2423	
7452	0001		0001	
7453	1423	ZLSR,	1423	/LSR
7454	2200		2200	
7455	0100		0100	
7456	0350	CGT,	0350	/C(CT)
7457	0724		0724	
7460	5100		5100	
7461	0100		0100	

7462	0123	ZASR,	0123	/ASR
7463	2200		2200	
7464	0100		0100	
7465	0420	ZDPSZ,	0420	/DPSZ
7466	2332		2332	
7467	0001		0001	
7470	0420	ZDPIC,	0420	/DPIC
7471	1103		1103	
7472	0001		0001	
7473	0403	ZDCM,	0403	/DCM
7474	1500		1500	
7475	0100		0100	
7476	0401	ZDAD,	0401	/DAD
7477	0400		0400	
7500	0100		0100	
7501	0423	ZDST,	0423	/DST
7502	2400		2400	
7503	0100		0100	
7504	0205	BEFORE,	0205	/BEFORE
7505	0617		0617	
7506	2205		2205	
7507	0001		0001	
7510	0106	AFTER,	0106	/AFTER
7511	2405		2405	
7512	2200		2200	
7513	0100		0100	
7514	0350	CMSH,	0350	/C(MSH)
7515	1523		1523	
7516	1051		1051	
7517	0001		0001	
7520	0350	CLSH,	0350	/C(LSH)
7521	1423		1423	
7522	1051		1051	
7523	0001		0001	
7524	2301	ZSAM,	2301	/"SAM"
7525	1500		1500	
7526	0100		0100	
7527	2205	REG,	2205	/"REG"
7530	0700		0700	
7531	0100		0100	
7532	1305	KEBSP1,	1305	/"KEB 1"
7533	7040		7040	
7534	6100		6100	
7535	0100		0100	
7536	2417	TOBEAD,	2417	/TO BE ADDED,
7537	4002		4002	
7540	0540		0540	
7541	0104		0104	
7542	0405		0405	
7543	0400		0400	
7544	0100		0100	

7545	2313	SO,	2313	/SKIP OCCURED.
7546	1120		1120	
7547	4817		4817	
7550	0303		0303	
7551	2522		2522	
7552	0504		0504	
7553	0001		0001	
7554	1617	NSO,	1617	/NO SKIP OCCURED.
7555	4023		4023	
7556	1311		1311	
7557	2040		2040	
7560	1703		1703	
7561	0325		0325	
7562	2205		2205	
7563	0400		0400	
7564	0100		0100	
7565	2205	DATER,	2205	/REG MODIFIED.
7566	0740		0740	
7567	1517		1517	
7570	0411		0411	
7571	0611		0611	
7572	0504		0504	
7573	0001		0001	

S

0163	4000
0164	7031
0165	0037
0166	7503
0167	0033
0170	0042
0171	2525
0172	9252
0173	5767
0174	7741
0175	9551
0176	6704
0177	5000

0000	11110000	11110000	11111111	11111111	11111111	11111111	11111111	11111111	11111111
0100	11111111	11111111	11111111	11111111	11111111	11111111	11111110	00011111	11111111
0200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
0300	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111100
0400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
0500	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	00000111
0600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
0700	11111111	11111111	11111111	11111111	11000000	00000000	00000000	00000000	11111111
1000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
1100	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
1200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
1300	11111111	11111111	11111111	11111111	11111111	11111111	11000000	00000111	11111111
1400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
1500	11111111	11110000	00000000	00000000	00000000	00000000	00000000	00000000	01111111
1600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
1700	11111111	11111111	11111111	11100000	00000000	00000000	00000000	00000000	01111111
2000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
2100	11111111	11111111	11111111	11100000	00000000	00000000	00000000	00000000	01111111
2200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
2300	11111111	11111111	11100000	00000000	00000000	00000000	00000000	00111111	11111111
2400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
2500	11111111	11111111	11111111	11100000	00000000	00000000	00000000	00000000	01111111
2600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
2700	11111111	11111111	11111111	11100000	00000000	00000000	00000000	00000000	11111111
3000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
3100	11111111	11111111	11111111	11111111	00000000	00000000	00000000	00000001	11111111
3200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
3300	11111111	11111111	11111111	11111111	11000000	00000000	00000000	00000001	11111111
3400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
3500	11111111	11111111	11111111	11111111	11110000	00000000	00000000	00000001	11111111
3600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
3700	11111111	11111111	11111111	11111100	00000000	00000000	00000000	00000000	00000000

4000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 4100 11111111 11111111 11111111 11111111 11111111 00000111 11111111 11111111
 4200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 4300 11111111 11111111 11111111 11111111 11110000 00000000 00000000 00000000
 4400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 4500 11111111 11111111 11111111 11111111 11111111 00000000 00000000 00000000
 4600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 4700 11111111 11111111 11111111 11111111 11111111 11111111 10000000 00000001

 5000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 5100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11000001
 5200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 5300 11111111 11111111 11111111 11111111 11111111 10000000 00000000 00001111
 5400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 5500 11111111 11111111 11111111 11111111 11111111 11111110 00000000 11111111
 5600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 5700 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111100

 6000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 6100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11000001
 6200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 6300 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11110000
 6400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 6500 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 6600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 6700 11111111 11111111 11111111 11111111 11111111 11111111 11000000 00011111

 7000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 7100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 10111111
 7200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 7300 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 7400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
 7500 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11100000
 7600
 7700

A	0076	CAC	7377	DEC12	4331	ESCL10	5126
ABERR	2113	CAP	6007	D10ERR	2643	ESCL11	5135
ABERR1	2122	CAM	7621	D11ERR	2707	ESCL12	5145
ABINC	2106	CGT	7456	DIGIT	6470	ESCL13	5155
AIERR	2226	CHAR	5636	DL0	7663	ESCL2	5040
AIERR1	2235	CHKAC	4531	DM0ERR	3020	ESCL3	5047
AC	0323	CHKMO	4532	DPAC	0122	ESCL4	5056
ACIND	0010	CHKSCA	4533	DP10	2601	ESCL5	5065
ACL	7701	CKDATA	2272	DP11	2654	ESCL6	5074
ACNMI	4060	CL	7407	DPIC	7573	ESCL7	5103
ACNMI1	3707	CLAH	7001	DPIS0H	2621	ESCL8	5112
ACNMIN	3704	CLRL4	1133	DPIS1H	2667	ESCL9	5117
ACNMIX	0063	CLS	7520	DPISIM	6273	EXEN	3731
ACP	7403	CMPCTR	6374	DPITS0	2600	EXINMI	3672
ACS	5156	CMQ	7403	DPITS1	2693	FADDR	6423
ACS1	5200	CMSH	7514	DPMO	0121	FILCNT	5637
ACS2	0034	CNTR1	2473	DPSE	7451	FILLER	0020
ACTOCK	7425	CNTR2	2474	DPSE0	2250	FLAG	6665
ACTUAL	0047	CNTR3	2475	DPSE0H	2400	FORDAD	7230
ADDR	6462	CNVCTR	6471	DPSEPH	5541	GEN	5546
ADDR2A	7910	COMEND	4753	DPSE00	2246	GENNMI	4273
AFTER	4536	COMP	6345	DPSEPR1	5544	GENX	0069
AGAIN	1233	COMPA	6355	DS0ERR	3440	GEX	4453
AMQAT3	3727	COMTST	4600	DS1ERR	3535	GFF	6004
ANCMIQ	0116	COUNT	0117	DST	7445	GTS1H	0046
ANYUSE	1230	COUNTX	0104	DST0	3311	GTTCK	0037
APMGAT	6443	CP	0261	DST0GN	3400	GTTST1	5211
ARROW	0140	CR	0070	DST1	3451	GTTST2	5221
ASCOMP	7415	CRLF	0124	DST1RN	3502	GTTST3	5231
ASR	2047	CRLF2	0125	DSTHED	7204	GTTST4	5237
ASR0	6251	CSC	7431	DSTREG	7106	GTTST5	5244
ASR0SH	2201	DA0ERR	3130	DSTS0H	3413	GKEN	3742
ASR0S0	6257	DA1ERR	3300	DSTS1H	3514	HEADIN	6733
ASRS0H	2072	DAD	7443	DSTTAB	7305	HKE	4505
ASRS1H	2216	DAD0	3031	DSTTS0	3310	HLT	7402
ASRS1H	6200	DAD1	3201	DSTTS1	3450	HSE	0244
ASRTS0	2046	DA0GEN	3053	DSEJMS	7045	HSE1	0427
ASRTS1	2200	DA0JMS	7044	DEERR0	2523	HSE2	0540
AT	0032	DA0S0H	3067	DEINC	2426	HSE2A	0563
AT3	1066	DA0S1H	3223	EJA	9261	HSE3	0704
BACK	0056	DA0S1M	6332	EDAD0	3107	HSE4	1032
BACP	0067	DADTAB	7327	EDAD1	3257	HSE5	1200
BDCNV	6426	DADTS0	3030	EDCMP	3011	HSENM	4261
BEFORE	7504	DADTS1	3200	EDPI0	2634	HSENM1	3657
BITSTR	0106	DATER	7565	EDPI1	2700	INCOR	0111
BLXP	0066	DCM	7575	EDPSE0	2512	10F	6002
BSW	7002	DCM0	2720	EDST0	3431	10N	6001
C	0077	DCMS0H	3000	EDST1	3526	K200	6472
C1	6370	DCMS1M	6311	EMQAT2	1063	K770	0123
C1A	6372	DCMTS0	2717	ESAM0	1276	KE0SP1	7532
C2	6371	DCOUNT	5767	ESAM1	1411	L	0311
C2A	6373	DDZ	7665	ESCL1	5031	LOERR	1713

LBERR1	1722	MQL	7421	PA2529	4302	S1ERR	1626
LBINC	1706	MQLT	0204	PACP	0362	S1ERR1	1635
LSERR	2026	MQLT1	0400	PAT00	4926	SAH	7457
LSERR1	2039	MQNMI	4074	PAT01	4925	SAM0	1246
LDGT	0153	MQNMIN	3706	PATCH	4945	SAM1	1334
LDREG	0151	MQNMIQ	3730	PBACP	0370	SAMGEN	1263
LDSC	0156	MQNMIK	3705	PBLXP	0347	SAMS0H	1315
LDSC1	5515	MQTOCK	0039	PC	5495	SAMS1H	1400
LF	0071	MSH	0043	PLAMGS	7046	SAMSIH	0813
LINK	0102	MTZER	4513	PLINK	0127	SAMTAB	7244
LKTOCK	0053	N	5474	PLXP	0332	SAMTS0	1245
LL	0074	NBASR0	2063	PMDAT	0600	SAMTS1	1333
LNPR2	1051	NBASR1	2207	PMQLT	0294	SASR1	6223
LPAR	5462	NBLSR0	1663	PNORM	4132	SASR1A	6237
LSH	0044	NBLSR1	2007	PRCHG	7002	SASR2	6241
LSIM	0042	NBSHL0	1444	PREGS	0137	SASR3	6245
LSR	7417	NBSHL1	1607	PRINT	6693	SAVREG	0141
LSR0	1647	NEXNM1	4324	PRNMI	4000	SCA	7441
LSR0SH	6160	NEXT	0057	PROBLM	7412	SCANM	4534
LSR1	2001	NM2525	4320	PRXLOP	0126	SCAST	3700
LSRA	6147	NM2522	4321	PSPC	6670	SCASTX	3703
LSR030	6167	NM7776	4323	PSTEP	5437	SCATXX	5430
LSRS0H	1672	NMERR	4333	PSTEPT	5490	SCAXX	5434
LSRS1H	2016	NMFLG	4322	PTHREE	1240	SCC23	3702
LSRS1M	6120	NMI	7411	PTO	0355	SCL	7403
LSR1S0	1646	NMIERR	3650	PTWO	1126	SCL1	5024
LSR1S1	2000	NMIODD	3701	Q	0073	SCL10	5120
LXP	0064	NMITPR	4126	RANCON	6596	SCL11	5127
M	0072	NM1XX	5425	RANDAD	3241	SCL12	5136
MCTR	6425	NMTS1	4450	RANDAT	5732	SCL13	5146
MDSEL	5274	NMTS3	4522	RANDEX	6595	SCL2	5032
MDTST	5001	NQP	7400	RANDOM	0150	SCL3	5041
MESSG	0131	NOPM	7401	RANGEN	6595	SCL4	5050
MODA	2464	NOPR	0724	RANSAV	6591	SCL5	5057
MODE	0115	NOPR3	1217	RANTAD	6542	SCL6	5066
MODEX	6717	NORMT	3600	RANTBL	6597	SCL7	5075
MODESL	0135	NORMT1	4200	RANTND	6590	SCL8	5104
MOVE	6400	NORMT2	4400	RDF	6214	SCL9	5113
MOVEA	6413	NOSKIP	2302	REG	7527	SCOMP	5520
MQ	0302	NOSKP	2270	RIF	6224	SCOUNT	0114
MQ1	0444	NSO	7554	RL2	0523	SCSIM	0045
MQ1SW	0437	NUMSH	7070	RL4	1021	SCTOCK	0036
MQA	7501	NUMSHF	0147	RNDATA	0132	SGT	6006
MQA1	0605	OBV	6524	ROTGEN	2476	SHIFTO	6113
MQAER1	0714	OBVERS	6473	RPAR	5467	SHIFTS	7447
MQAER2	1042	ONE	0100	RTF	0005	SHL	7413
MQAER3	1210	ONEP	0133	RTFX	0154	SHL0	1431
MQAT	0503	ONLYB	0136	RXLOP	5613	SHL1	1601
MQAT1	0650	ONZER	0132	S0ERR	1474	SHLA	0074
MQAT2	1000	OPEN	0000	S0ERR1	1503	SHL031	6107
MQAT3	1135	P12BIT	7200	S0INC	1467	SHLSIM	0042
MQIND	0011	P1BIT	0155	S10SET	5311	SHLTS0	1430

SHLTS1	1600	TSTSW2	0145	YSKIP	2313
SIMULT	7417	TSTSW3	0146	ZASR	7402
SKIPMA	6666	TT	0075	ZDAD	7476
SKIPPA	6667	TWICE	0120	ZDCM	7473
SLTS0H	1453	TWO	0113	ZDPIC	7470
SLTS1H	1616	TYMOD	6704	ZDPSZ	7465
SM0ERR	1305	TYPAT	6630	ZDST	7501
SM1ERR	1420	TYPSP	6634	ZER0	7443
SO	7545	TYPST	6600	ZERO	0101
SPACE1	6723	TYTST	0134	ZEROR	0130
SPACE2	6727	U1SPC	0054	ZLSR	7453
SPAT00	4527	U2SPC	0055	ZONE	7445
SPAT01	4530	UCOMP	0052	ZSAM	7524
SPCTR	6703	UCRLF	5600	ZSHL	7435
STRCNT	0105	UCRLF2	5607		
SW0TST	5325	UGEN	0142		
SW1TST	5333	ULDGT	5761		
SW2TST	5342	ULDREG	5791		
SW3TST	5400	ULDSC	5503		
SWAB	7431	UMESSG	5663		
SWBA	7447	UMOVE	0053		
SWITCH	6655	UONEP	5692		
SWP	7521	UONLYB	5317		
T	0316	UONZER	5645		
TAC	0027	UP1BIT	5726		
TADDR	6424	UPLINK	5640		
TEHPA	0040	UPREGS	7000		
TEHPB	0041	UPRONE	5703		
TEHPQ	6662	UPSPC	0051		
TEHQ	6663	USVREG	5410		
TEHR	6664	UTYTST	5707		
TEHPWR	6463	UZEROR	5656		
TEST	7440	VALUE	6467		
TGT	0032	VOR	1121		
THREE	0112	WILCHG	7016		
TLINK	0026	XACNMI	0012		
TMQ	0030	XCP	0467		
TO	0103	XMQ1	0455		
TOAC	0022	XMQAT	0061		
TOBEAD	7536	XMQAT1	0062		
TOGT	0025	XMQAT2	0107		
TOLINK	0021	XMQAT3	0110		
TOMQ	0023	XHOLT1	0060		
TOSHIF	0024	XNONMI	0013		
TPFLAG	4537	XNORM1	4143		
TSC1	6606	XONE	0473		
TSC2	6617	XPACP	0340		
TSC3	5000	XPMQAT	1062		
TSHIF	0031	XRTF	6000		
TST25	4315	XTYPESI	0050		
TSTSW0	0143	YA	0641		
TSTSW1	0144	YESSKP	2271		

/KES EAE INSTRUCTION TEST PART 1 MAINDEC-8E-D8LB

PAL10 V141 9-FEB-72

16126 PAGE 1-84

ERRORS DETECTED: 0

LINKS GENERATED: 144

RUN-TIME: 48 SECONDS

3K CORE USED

