

IDENTIFICATION

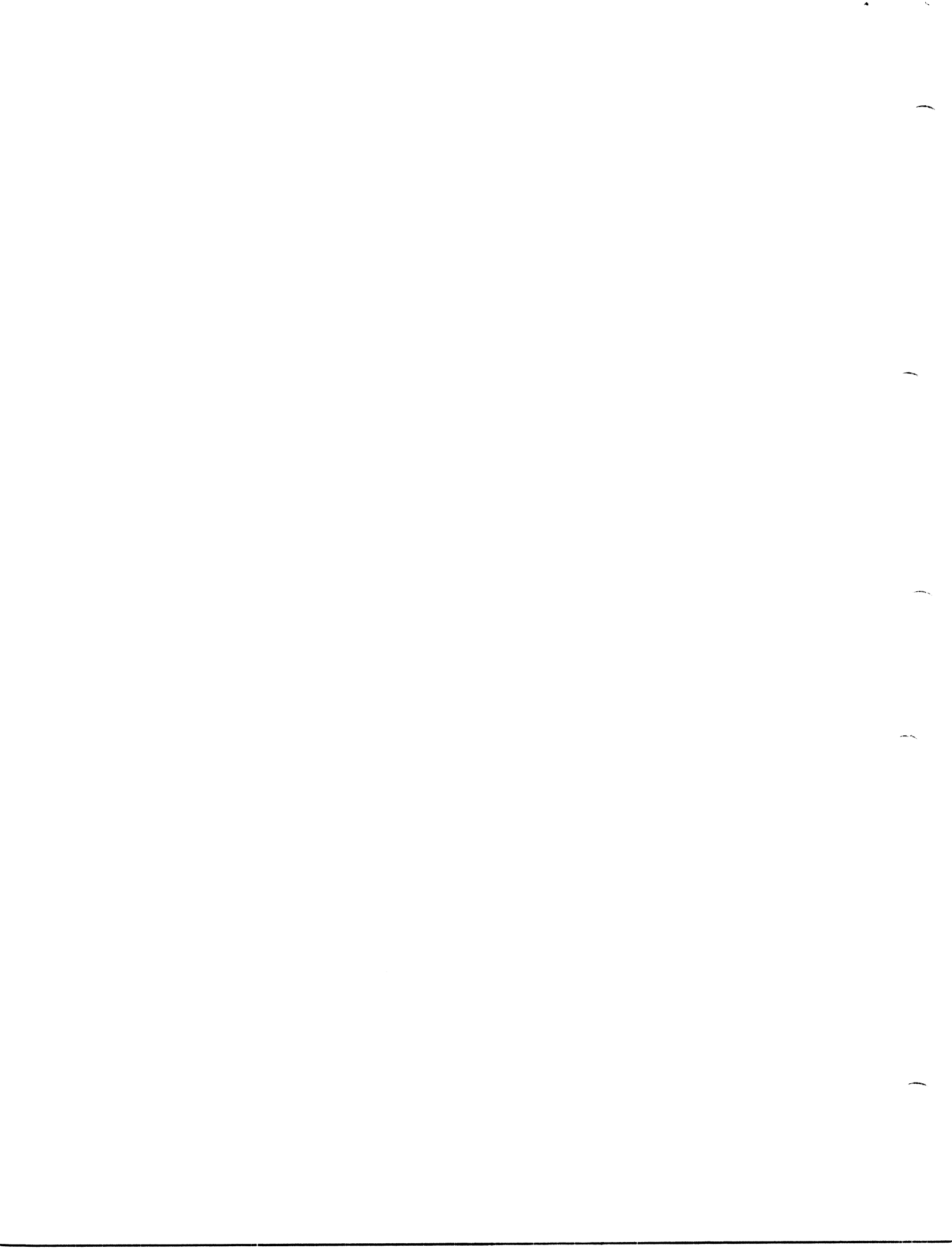
PRODUCT CODE: MAINDEC-08-DHKEA-A-D
 REPLACES: MAINDEC-8E-DORA-D

PRODUCT NAME: KE8-E EAE EXTENDED MEMORY
 EXERCISER

DATE CREATED: AUGUST 31, 1972

AUTHOR: E. FORTMILLER

COPYRIGHT © 1971, 1972
DIGITAL EQUIPMENT CORPORATION



1. ABSTRACT

THE KE8-E EXTENDED MEMORY EXERCISER IS A TEST OF THE KE8-E "B MODE" INSTRUCTIONS WHICH DURING THE DEFER CYCLE USE THE WORD FOLLOWING THE INSTRUCTION TO OBTAIN THE OPERAND. THE CAPABILITY OF EACH INSTRUCTION TO ACCESS EVERY MEMORY FIELD FROM EVERY MEMORY FIELD THROUGH NON-AUTO INDEX AND AUTO INDEX, IS TESTED,

THE FOLLOWING IS A LIST OF THE INSTRUCTIONS IN THE SEQUENCE THEY ARE TESTED,

- A. DOUBLE PRECISION ADD (DAD)
- B. DOUBLE PRECISION STORE (DST)
- C. MULTIPLY (MUY)
- D. DIVIDE (DVI)

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-8/E PROCESSOR WITH AT LEAST 4K OF MEMORY, KE8-E OPTION, AND A TELETYPE ARE REQUIRED,

2.2 STORAGE

LOCATIONS 0000 THROUGH 7300,

2.3 PRELIMINARY PROGRAMS

ALL PROCESSOR AND MEMORY RELATED TEST PROGRAMS, AND KE8-E INSTRUCTION TESTS 1 AND 2 MUST HAVE BEEN RUN SUCCESSFULLY,

3. LOADING PROCEDURE

THE BINARY LOADER IS USED TO LOAD THE PROGRAM INTO ANY DESIRED FIELD. REFER TO THE BINARY LOADER DOCUMENTATION IF UNFAMILIAR WITH ITS USE.

4, STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SR0=0 HALT ON ERROR
SR0=1 NO HALT ON ERROR

SR1=0 NO LOOP
SR1=1 SCOPE LOOP (REPEAT PATTERN)

SR2=0 PRINT ON ERROR
SR2=1 NO PRINT ON ERROR

SR3=0 SEQUENTIALLY RUN TESTS
SR3=1 SELECT TEST ACCORDING TO SR4-6,

SR3=1	SR4	SR5	SR6	
	0	0	0	SELECT DAD TEST
	0	0	1	SELECT DST TEST
	0	1	0	SELECT MUY TEST
	0	1	1	SELECT DIV TEST
	1	0	0	SELECT DAD AUTO INDEX TEST
	1	0	1	SELECT DST AUTO INDEX TEST
	1	1	0	SELECT MUY AUTO INDEX TEST
	1	1	1	SELECT DIV AUTO INDEX TEST

SR7=0 RELOCATE IF SR3 AND 8 ARE BOTH 0,
SR7=1 HOLD IF

SR8=0 SEQUENTIALLY CHANGE DF IF SR3=0
SR8=1 HOLD DF

SR9-11 STARTING DATA FIELD
(ONLY READ AT START TIME)

	9	10	11	
	0	0	0	DF0
	0	0	1	DF1
	0	1	0	DF2
	0	1	1	DF3
	1	0	0	DF4
	1	0	1	DF5
	1	1	0	DF6
	1	1	1	DF7

4.2 STARTING ADDRESS

THIS PROGRAM STARTS AT LOCATION 0200.

4.3 PROGRAM AND/OR OPERATOR ACTION

WITH 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.
- F. TYPE IN THE VALUE OF THE HIGHEST MEMORY IN THE SYSTEM FOLLOWED BY A CARRIAGE RETURN.
(EG. 0 FOR 4K, 1 FOR 8K ON UP TO 7 FOR 32K)
IF THE INCORRECT NUMBER WAS TYPED, TYPE RUBOUT AND THEN RETYPE THE MEMORY FIELD VALUE.

NOTE:

FOR A NORMAL PROGRAM RUN, LOAD THE PROGRAM IN FIELD 0, SET SR TO 0200 AND PRESS LOAD ADDRESS. NOW SET SR TO 0000 AND PRESS CLEAR AND CONTINUE, AND RESPOND TO THE TELETYPE WITH THE VALUE OF THE HIGHEST FIELD FOLLOWED BY A CARRIAGE RETURN. THE PROGRAM WILL NOW ACCESS EVERY FIELD FROM EVERY FIELD. AT THE COMPLETION OF A COMPLETE PROGRAM PASS, "KEB-EME" WILL BE TYPED ON THE TELETYPE.

5. ERRORS

5.1 ERROR HALTS AND DESCRIPTION

LOC 0225	THE SELECTED STARTING DF IN SR9-11 WAS HIGHER THAN THE HIGHEST MEMORY FIELD DEFINED ON THE TELETYPE.
LOC 0256	DOUBLE PRECISION ADD (DAD) INSTRUCTION FAILED.
LOC 0343	DOUBLE PRECISION ADD (DAD) INSTRUCTION FAILED USING AUTO-INDEX.
LOC 0450	DOUBLE PRECISION STORE (DST) INSTRUCTION FAILED.
LOC 0560	DOUBLE PRECISION STORE (DST) INSTRUCTION FAILED USING AUTO-INDEX.
LOC 0637	MULTIPLY INSTRUCTION FAILED.
LOC 0746	MULTIPLY INSTRUCTION FAILED USING AUTO-INDEX.
LOC 1066	DIVIDE INSTRUCTION FAILED.
LOC 1250	DIVIDE INSTRUCTION FAILED USING AUTO-INDEX
LOC 1505	RELOCATION ERROR.
LOC 2411	DIVIDE OVERFLOW IN SIMULATION, THIS ERROR SHOULD NEVER OCCUR. (PROCESSOR ERROR)
LOC 2555	AUTO INDEX ERROR DURING EXECUTION OF AN EAE INSTRUCTION.

5.2 ERROR PRINTOUTS

5.2.1 DAD TEST'S

DAD TEST ERROR

IF 0 DF 1
MQ ADDR 3000
AC ADDR 3001

	L	AC	MQ
ORIGINAL	0	4776	5000
TO BE ADDED		3001	3000
SIMULATED	1	0000	0000
ACTUAL	1	0000	0001

DAD AUTO INDEX TEST ERROR

IF 0 DF 1
MQ ADDR 3000
AC ADDR 3001

	L	AC	MQ
ORIGINAL	0	4776	5000
TO BE ADDED		3001	3000
SIMULATED	1	0000	0000
ACTUAL	1	0000	0001

IF INDICATES THE FIELD THE PROGRAM IS IN,
DF INDICATES THE FIELD "DAD" WAS ACCESSING,
MQ ADDR = ADDRESS IN THE DATA FIELD THAT CONTAINS THE LEAST SIGNIFICANT 12 BITS OF THE 24 BIT WORD TO BE ADDED,
AC ADDR ADDRESS IN THE DATA FIELD THAT CONTAINS THE MOST SIGNIFICANT 12 BITS OF THE 24 BIT WORD TO BE ADDED,
ORIGINAL THE CONTENTS OF THE REGISTER PRIOR TO ISSUING A "DAD" INSTRUCTION,
TO BE ADDED THE CONTENTS OF THE TWO ADDRESSES TO BE ADDED TO THE AC AND MQ,
SIMULATED WHAT THE RESULTS SHOULD BE,
ACTUAL WHAT THE RESULT WAS,
NOTE: THE ACTUAL MQ SHOULD BE 0000,

5,2.2 DST TEST'S

DST TEST ERROR

IF 0 DF 0

MQ ADDR 3000
AC ADDR 3001

REG	BEFORE DST	AFTER DST
C(L)	0	0
C(MQ)	3000	3000
C(MQ ADDR)		3000
C(AC)	3001	3001
C(AC ADDR)		0000

DST AUTO INDEX TEST ERROR

IF 1 DF 1

MQ ADDR 3000
AC ADDR 3001

REG	BEFORE DST	AFTER DST
C(L)	0	0
C(MQ)	3000	3000
C(MQ ADDR)		3000
C(AC)	3001	3001
C(AC ADDR)		0000

IF INDICATES THE FIELD THE PROGRAM IS IN,
 DF INDICATES THE FIELD "DST" WAS STORING IN,
 MQ ADDR ADDRESS IN THE "DF" THAT THE MQ IS TO BE STORED IN,
 AC ADDR ADDRESS IN THE "DF" THAT THE AC IS TO BE STORED IN,
 BEFORE DST WHAT THE REGISTERS WERE PRIOR TO ISSUING A "DST" INSTRUCTION,
 AFTER DST WHAT THE REGISTERS WERE AFTER DST WAS EXECUTED,
 NOTE: THE CONTENTS OF THE AC ADDR SHOULD BE 3001,

5,2,3 MUY TEST'S

MUY TEST ERROR

IF 0 DF0

MULTIPLICAND ADDR		3000				
	L	MTPL	AC	MQ	SC	
ORIGINAL	0	4777	0000	3000		
SIMULATED	0		1677	5000	0014	
ACTUAL	0		1677	5001	0014	

MUY AUTO INDEX TEST ERROR

IF 0 DF 0

MULTIPLICAND ADDR		3000				
	L	MTPL	AC	MQ	SC	
ORIGINAL	0	4777	0000	3000		
SIMULATED	0		1677	5000	0014	
ACTUAL	0		1677	5001	0014	

- IF INDICATES THE FIELD THE PROGRAM IS IN,
- DF INDICATES THE FIELD THE MULTIPLICAND IS IN,
- MULTIPLICAND ADDR ADDRESS WHERE THE MULTIPLICAND IS LOCATED,
- MTPL THE MULTIPLICAND,
- ORIGINAL CONTENTS OF REGISTERS PRIOR TO ISSUING A MULTIPLY INSTRUCTION,
- SIMULATED WHAT THE RESULTS SHOULD BE,
- ACTUAL WHAT THE RESULTS WERE,
- NOTE: ACTUAL MQ SHOULD BE 5000,

5.3.4 DIV TEST'S

DIV TEST ERROR

IF 0 DF0

DIVISOR ADDR 3000

	L	DIVS	AC	MQ	SC
ORIGINAL	0	4777	1377	5000	
SIMULATED	0		4000	3777	0015
ACTUAL	0		4001	3777	0015

DIV AUTO INDEX TEST ERROR

IF 0 DF 0

DIVISOR ADDR 3000

	L	DIVS	AC	MQ	SC
ORIGINAL	0	4777	1377	4777	
SIMULATED	0		4000	3777	0015
ACTUAL	0		4001	3777	0015

IF INDICATES THE FIELD THE PROGRAM IS IN,
 DF INDICATES THE FIELD THE DIVISOR IS IN,
 DIVISOR ADDR ADDRESS WHERE THE DIVISOR IS LOCATED,
 DIVS THE DIVISOR,
 ORIGINAL CONTENTS OF THE REGISTERS PRIOR TO ISSUING
 A DIVIDE INSTRUCTION,
 SIMULATED WHAT THE RESULTS SHOULD BE,
 ACTUAL WHAT THE RESULTS WERE,
 NOTE: ACTUAL AC SHOULD BE 4000.

5.3.5 AUTO-INDEX FAILURE

DAD, (DST), (MUY), OR (DIV) AUTO INDEX TEST ERROR

IF 0 DF0

AI10 SHOULD BE 3001
AI10 IS 3000

"SHOULD BE" IS THE CONTENTS OF AUTO INDEX REGISTER 10 BEFORE EXECUTING A EAE INSTRUCTION IN ADDRSS 0007,

"IS", IS THE CONTENTS OF THE AUTO INDEX REGISTER AFTER THE EXECUTION OF THE EAE INSTRUCTION,

NOTE: AI10 "IS" SHOULD BE 3001,
SWITCH REGISTER SETTINGS DO NOT APPLY TO THIS TYPE OF ERROR, AN ERROR OF THIS TYPE WILL BE PRINTED FOLLOWED BY A HALT, PRESSING CONTINUE (NOT CLEAR AND CONTINUE) WILL ENTER A SCOPE LOOP.

6.0 MISCELLANEOUS

6.1 EXECUTION TIME

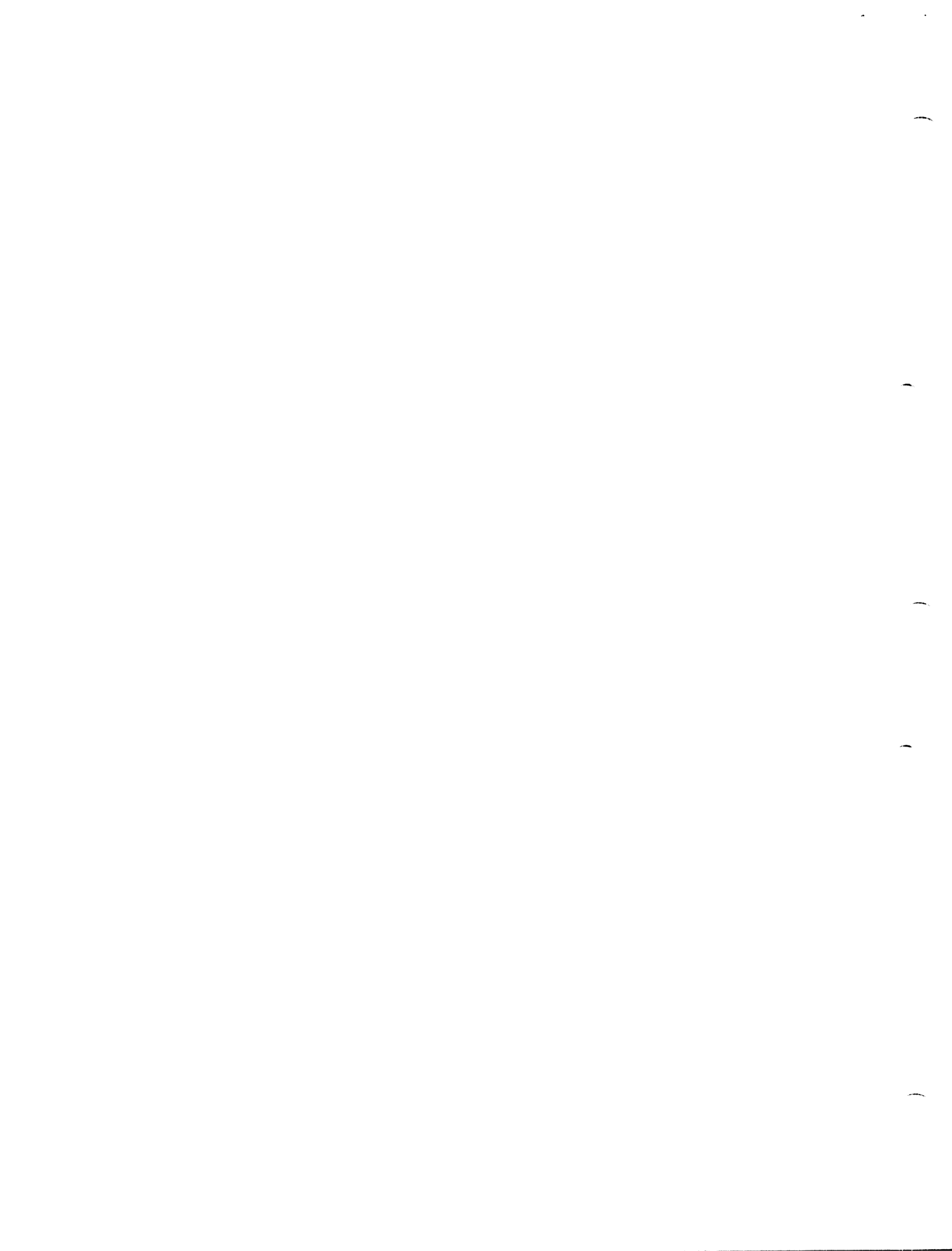
AT THE COMPLETION OF ONE COMPLETE PROGRAM PASS, "KEB EME", WILL BE PRINTED ON THE CONSOLE TELETYPE UNLESS SR2=1, WHICH WOULD ONLY RING THE TELETYPE BELL.

AMOUNT OF MEMORY
(TIME MIN:SEC)

4K	8K	12K	16K
0:2,5	0:11	0:25	0:45
20K	24K	28K	32K
1:15	1:45	2:30	3:30

7.0 PROGRAM DESCRIPTION

SEE DESCRIPTION AT THE BEGINNING OF EACH TEST IN THE PROGRAM LISTING.



/KRB-E EAE EXTENDED MEMORY EXERCISER MAINDEC-08-DHKEA-A=L
 /COPYRIGHT 1971-1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS, 01754
 /PROGRAMMER ED FORTMILLER
 /

/EAE EQUATE STATEMENTS:

/BASIC MOI

7401 MOPE=7401 /EAE NO OPERATION;
 7601 CLAE=7601 /EAE CLEAR AC;
 7421 MQL=7421 /LOAD MQ FROM AC, THEN CLEAR AC;
 7501 MOA=7501 /INCLUSIVE OR THE MQ WITH THE AC,
 7621 CAM=CLAE MQL /CLEAR AC AND MQ;
 7521 SWP=MOA MQL /SWAP AC AND MQ;
 7701 ACL=CLAE MOA /LOAD MQ INTO AC;

/MODEI

7431 SWAB=7431 /CHANGE TO "B" MODE;
 7447 SWBA=7447 /CHANGE TO "A" MODE;

/STEP COUNTER MODE "A":

7403 SCL=7403 /STEP COUNTER LOAD FROM MEMORY,
 7441 SCA=7441 /STEP COUNTER "OR" WITH AC;

/STEP COUNTER, MODE "B":

7403 ACS=7403 /AC 7=11 TO STEP COUNTER THEN CLEAR AC;

/MUY, DIV, NMI, AND SHIFTS:

7405 MUY=7405 /MULTIPLY,
 7407 DIV=7407 /DIVIDE,
 7411 NMI=7411 /NORMALIZE,
 7413 SHL=7413 /SHIFT LEFT,
 7415 ASR=7415 /ARITHMETIC SHIFT RIGHT,
 7417 LSR=7417 /LOGICAL SHIFT RIGHT;

/SUBTRACTION, MODE "B" ONLY:

7457 SAM=7457 /SUBTRACT AC FROM MQ;

/DOUBLE PRECISION, MODE B ONLY:

7443 DAD=7443
 7445 DST=7445
 7573 DPIQ=7573
 7575 DCM=7575
 7451 DPSZ=7451

/COMBINED DOUBLE PRECISION, MODE "B" ONLY:

7663 DLD=DAD CAM /DOUBLE PRECISION LOAD
 7665 DDZ=DST CAM

/EQUATE STATEMENTS FOR PROCESSOR IOT:

6000 SKON=6000 /SKIP IF INTERRUPT ON, AND TURN OFF.
 6001 ION=6001 /TURN INTERRUPT ON;
 6002 IOF=6002 /TURN INTERRUPT OFF;
 6003 SRQ=6003 /SKIP ON INTERRUPT REQUEST;
 6004 GTF=6004 /GET INTERRUPT FLAGS;
 6005 RTF=6005 /RESTORE INTERRUPT FLAGS;
 6006 SGT=6006 /SKIP ON GREATER THAN FLAG;
 6007 CAF=6007 /CLEAR ALL FLAGS;

/EQUATE STATEMENTS FOR MEMORY EXTENSION:

6214 RDF=6214 /READ DATA FIELD
 6224 RIF=6224 /READ INSTRUCTION FIELD;
 6234 RIB=6234 /READ INTERRUPT BUFFER
 6244 RMP=6244 /RESTORE MEMORY FIELD;
 6201 CDF=CDF /CHANGE DATA FIELD

/EQUATE STATEMENTS FOR CREP LISTING:

7402 HLT=7402 /HALT;
 7604 LAS=7604 /READ SWITCHES;

/MISCELLANEOUS EQUATE STATEMENTS:

0000 OPEN=0000 /PROGRAM MODIFIABLE;
 7000 EQUAL=7000 /NOP
 7140 COMPLE=7140 /CHA CLL
 7002 BSW=7002 /BYTE SWAP,
 4577 TYPE=JMS I [UTYPE /TYPE A MESSAGE
 4576 SPACEX=JMS I [XSPACE /SPACE
 4575 TYPEX=JMS I [XTYPE
 4574 INPUT=JMS I [XINPUT

0000 *0
 0000 0000 0
 0001 5001 JMP 1
 0002 0002 2
 0003 0003 3
 0004 0004 4
 0005 0005 5
 0006 0000 INSTRU, OPEN
 0007 0000 ATEST, OPEN /WILL CONTAIN A DAD, DST, MUY, OR DIV.
 0010 0000 *10
 0011 5406 A110, OPEN
 0012 0000 A111 JMP I INSTRU
 0013 0000 A112, OPEN
 0014 0000 A113, OPEN
 0014 0000 A114, OPEN

```

0015 0000 A115, OPEN
0016 0000 A116, OPEN
0017 0000 A117, OPEN

0020 *20

0020 0000 LKTOCK, OPEN
0021 0000 MQTOCK, OPEN
0022 0000 ACTOCK, OPEN
0023 0000 SCTOCK, OPEN
0024 0000 OFTOCK, OPEN
0025 0000 IFTOCK, OPEN
0026 0000 HIFLD, OPEN
0027 0000 BEGIF, OPEN
0030 0000 BEGDF, OPEN
0031 0000 IFFLD, OPEN
0032 0000 DFFLD, OPEN
0033 0000 TEMP, OPEN
0034 0000 TEMP1, OPEN
0035 0000 TEMP2, OPEN
0036 0000 TEMP3, OPEN
0037 0000 TEMP4, OPEN
0040 0000 90TH, OPEN
0041 0000 COUNT, OPEN
0042 0000 NEXT, OPEN
0043 0000 ZERC, 0000
0044 0000 SAVEAC, OPEN
0045 0000 SAVEA1, OPEN
0046 0000 SAVAC, OPEN
0047 0000 SAVLNK, OPEN
    
```

0200 *0200

/START OF PROGRAM

```

0200 0007 START, CAP /CLEAR THE WORLD;
0201 1377 TAO (DADTST /ADDRESS OF FIRST TEST;
0202 3042 DCA NEXT /SAVE THE TEST ADDRESS;
0203 6224 RIF /FIND THE FIELD WE'RE STARTING IN=
0204 3027 DCA BEGIF /AND SAVE IT;
0205 7604 LAS /FIND THE OF=
0206 0173 AND C? /SAVE ONLY SWITCHES 9=1;
0207 7104 RAL CLL /WE'RE TO=
0208 7006 RTL /START IN=
0209 3030 DCA BEGDF /AND SAVE IT;
0210 4572 JMS I (RSETOP /SET OP EQUAL TO IP;
0211 5224 RIF /SET "IFFLD" EQUAL=
0212 3031 DCA IFFLD /TO THE IP;
0213 1030 TAD BEGDF /SET BEGDF EQUAL=
0214 3032 DCA DFFLD /TO THE SELECTED OP;
0215 4776 JMS ASKFLD /FIND OUT HIGHEST FIELD;
0216 1026 TAD HIFLD /GET HIGHEST FIELD JUST DEFINED BY USER;
    
```

```

0221 7140 CLL CMA /CHANGE TO 1'S COMPLEMENT;
0222 1030 TAO BEGDF /ADD BEGINNING OF TO IT;
0223 7620 SNL CLA /IS THE SELECTED STARTING OF HIGHER?
0224 5775 JMP SELTST /NO, SELECT A TEST AND START;
0225 7402 WLT /THE SELECTED STARTING OF IS HIGHER
/THAN THE HIGHEST FIELD JUST DEFINED,
/RESELECT SO OF IS EQUAL TO OR LOWER
/THAN THE HIGHEST FIELD,
/TO CATCH "CONTINUE";

0226 5200 JMP START
    
```

/DOUBLE PRECISION ADD TEST (DAD)
 /LOAD ADDRESSES 3000-3777 OF THE SELECTED OP WITH
 /A NUMBER EQUAL TO THE ADDRESS;
 /WHEN EXECUTING A DAD THE NUMBERS WILL COME

/FROM 3 AREAS,
 / 3000-3777
 / 4000-4777
 / 5000-5777
 /THE SEQUENCE FOR DOUBLE PRECISION ADDITION WILL BE
 /TO LOAD THE MQ WITH THE 2'S COMPLEMENT
 /OF THE CONTENTS OF THE LOCATION FOLLOWING THE DAD INSTRUCTION
 /AND THE AC WITH THE 1'S COMPLEMENT OF THE CONTENTS OF THAT
 /LOCATION +1. THE LOCATION FOLLOWING THE DAD WILL SEQUENTIALLY CHANGE
 /FROM 3XXX TO 4XXX TO 5XXX AND THEN START OVER AGAIN WITH +1
 /TO THE LOCATION, AN EXIT WILL TAKE PLACE AFTER 5976
 /HAS BEEN ACCESSED,

```

0227 4571 DADTST, JMS I (LBCORE /LOAD CORE WITH A NUMBER =
0228 7000 EQUAL /EQUAL TO THE ADDRESS;
0229 1374 TAO (DADA1T /ADDRESS OF THE NEXT TEST;
0230 3042 DCA NEXT /SAVE IT FOR A POINTER;
0231 7431 SWAB /SWITCH TO "B" MODE OF OPERATION;
0232 1170 TAO (3000 /OBTAIN THE INITIAL STARTING ADDRESS=
0233 3245 DCA DADADR /OF THE BUFFER;
0234 1245 DAD, TAD DADADR /OBTAIN THE NUMBER IN DADADR,
0235 7421 MQL /PLACE IT IN THE MQ;
0236 4567 JMS I (SETOP /SET DATA FIELD TO ADD FROM;
0237 7501 MQA /LOAD AC WITH C(MQ);
0238 7101 CLL IAC /+1 TO AC
0239 7575 DCM /CHANGE AC+MQ TO 2'S COMPLEMENT;
0240 7443 DAD /EAE DOUBLE PRECISION ADD;
0241 0000 DADADR, OPEN /THIS SPECIFIES THE LOCATION OF THE NUMBERS TO BE ADDED.
0242 7451 OPSE /RESULTS SHOULD BE L=1, AC+MQ=0,
0243 5252 JMP ,+3 /AC OR MQ NOT 0,
0244 7430 SEL /HAS LINK#1?
0245 5257 JMP DADLOP /L, AC, + MQ=ACK;
0246 4566 JMS I (SAVREG /DAD ERROR, SAVE CONTENTS OF REGISTERS;
0247 4565 JMS I (SW2 /SHALL WE PRINT?
0248 4300 JMS DADERP /YES;
0249 4564 JMS I (SW2 /SHALL WE HALT?
0250 7402 DADHLT, HLT /YES, DAD ERROR;
0251 4572 DADLOP, JMS I (RSETOP /RESET DATA FIELD TO ORIGINAL;
    
```

```

0260 4563 JMS I [SW1 /SMALL WE LOOP?
0261 5236 JMP DAD0 /YES? USE SAME DATA,
0262 1245 TAD DADADR /THE ADDITION OF THESE TWO NUMBERS=
0263 1162 TAD [=5776 /IS TO SEE IF 5776 HAS BEEN ACCESSED,
0264 7650 SNA CLA /AC=0, INDICATING IT HAS BEEN?
0265 5561 JMP I [NXTST /YES, GO TO NEXT SELECTED TEST,
0266 1245 TAD DADADR /THE ADDITION OF THESE TWO NUMBERS=
0267 1162 TAD [1000 /IS TO SELECT THE NEXT BUFFER,
0270 3245 DCA DADADR /NOW SAVE NEXT BUFFER ADDRESS,
0271 1245 TAD DADADR /LETS CHECK AND SEE IF=
0272 7104 RAL CLL /THE NEXT BUFFER ADDRESS IS=
0273 7700 SMA CLA /6000 OR MORE,
0274 5236 JMP DAD0 /IT WASN'T SO GO TO BEGINNING,
0275 1245 TAD DADADR /THE NEXT BUFFER WAS 6000 OR MORE=
0276 1157 TAD [=2777 /SO SUBTRACT 2777 AND THIS WILL=
0277 5235 JMP DAD0-1 /ALLOW BUFFER ADDRESS TO BE 3XXX+1,

0300 0000 DADERP, OPEN
0301 1245 TAD DADADR /OBTAIN ADDRESS OF LEAST SIGNIFICANT HALF,
0302 4556 JMS I [INDMD /PRINT PART 1 OF ERROR,
0303 6755 DADTX /TEST IDENTIFIER,
0304 1245 TAD DADADR /OBTAIN ADDRESS OF LEAST SIGNIFICANT HALF
0305 4773 JMS DADP2 /PRINT PART 2 OF ERROR,
0306 5700 JMP I DADERP /EXIT,

```

/DOUBLE PRECISION ADD TEST (DAD) USING AUTO INDEX;
 /LOAD ADDRESSES 3000-5777 OF THE SELECTED OF WITH
 /A NUMBER EQUAL TO THE ADDRESS.
 /LOAD THE MQ WITH 2'S COMPLEMENT OF THE CONTENTS+1 OF
 /THE LOCATION FOLLOWING THE DAD INSTRUCTION AND THE AC WITH THE
 /1'S COMPLEMENT OF THE CONTENTS OF THAT LOCATION+1;

```

0307 4571 DADAIT, JMS I [LDCORE /LOAD CORE WITH A NUMBER=
0310 7000 EQUAL /EQUAL TO THE ADDRESS,
0311 1372 TAD [DSTTST /ADDRESS OF THE NEXT TEST,
0312 3042 DCA NEXT /SAVE AS A POINTER,
0313 1371 TAD [DAD /GET A DAD INSTRUCTION,
0314 3007 DCA ATEST /AND SAVE FOR EXECUTION,
0315 1155 TAD [2777 /OBTAIN THE INITIAL STARTING ADDRESS
0316 3010 DCA A110 /OF THE BUFFER=1 AND SAVE IT,
0317 1157 TAD [=2777 /SET UP FOR A=
0320 3041 DCA COUNT /COUNT OF 2777 OCTAL,
0321 7431 SWAB /SWITCH TO "B" MODE OF OPERATION,
0322 1010 DADA, TAD A110 /OBTAIN THE NUMBER IN A110=
0323 7001 IAC /AND ADD 1 TO IT THEN PLACE=
0324 7421 MQL /IT IN THE MQ,
0325 4567 JMS I [SETOF /SET DATA FIELD TO ADD FROM,
0326 7501 MDA /LOAD AC WITH C(MQ)
0327 7001 IAC /+1 TO AC,
0330 7575 DCM /CHANGE AC&MQ TO 2'S COMPLEMENT,
0331 4770 JMS CHECAI /EXECUTE A DAD FROM AN AUTO-INDEX REGISTER,
0332 6760 DADATX /TEST IDENTIFIER,
0333 7451 DPBZ /RESULTS SHOULD BE L=1, AC&MQ=0,

```

```

0334 5337 JMP ,+3 /DAD ERROR, AC OR MQ NOT 0
0335 7430 SEL /L=1?
0336 5344 JMP DADALP /EVERYTHING A=OK,
0337 4566 JMS I [SAVREG /DAD ERROR, SAVE CONTENTS OF ALL REGISTERS,
0340 4565 JMS I [SW2 /SMALL WE PRINT?
0341 4352 JMS DADAIE /YES,
0342 4564 JMS I [SW0 /SMALL WE HALT?
0343 7402 HLT /YES, DAD ERROR, AUTO INDEX,
0344 4572 DADA-P, JMS I [RSETDF /RESET THE DF,
0345 4563 JMS I [SW1 /SMALL WE LOOP?
0346 4584 JMS I [RSETAI /YES, NOW RESET A110 SO WE MAY,
0347 2041 ISZ COUNT /FINISHED YET?
0350 5322 JMP DAD0 /NO, DO ANOTHER DAD,
0351 5561 JMP I [NXTST /YES, GO TO NEXT TEST

0352 0000 DADAIE, OPEN
0353 1010 TAD A110 /OBTAIN ADDRESS OF LEAST SIGNIFICANT HALF,
0354 4556 JMS I [INDMD /PRINT,
0355 6760 DADATX /MESSAGE ADDRESS,
0356 1010 TAD A110 /TO BE USED IN THE PRINT ROUTINE,
0357 4773 JMS DADP2 /PRINT,
0360 5752 JMP I DADAIE /EXIT,

0370 2474
0371 7443
0372 0400
0373 6033
0374 0307
0375 2074
0376 2000
0377 0227
0400 PAGE

```

/DOUBLE PRECISION STORE TEST (DST)
 /LOAD ADDRESSES 3000 TO 5777 OF THE SELECTED DP WITH
 /A NUMBER EQUAL TO THE 1'S COMPLEMENT OF THE ADDRESS;
 /WHEN STORING WITH DST THE STORAGE AREA IS DIVIDED
 /INTO 3 AREAS,
 / 3000 TO 3777
 / 4000 TO 4777
 / 5000 TO 5777
 /THE SEQUENCE FOR STORING WILL BE TO "DST" TO EACH ONE IN SEQUENCE AND
 /CHECK EACH ADDRESS FOR THE PROPER CONTENTS AFTER EACH DST,
 /AC AND MQ ARE CHECKED IMMEDIATELY AFTER DST TO VERIFY
 /THAT NO CHANGE OCCURRED IN EITHER,

```

0400 4571 DSTTST, JMS I [LDCORE /LOAD CORE WITH A NUMBER EQUAL=
0401 7140 COMPLE /TO THE 1'S COMPLEMENT OF THE ADDRESS,
0402 1153 TAD [DSTAIT /ADDRESS OF THE NEXT TEST,
0403 3042 DCA NEXT /SAVE FOR A POINTER,
0404 7431 SWAB /SWITCH TO "B" MODE OF OPERATION
0405 1170 TAD [3000 /OBTAIN THE INITIAL STARTING ADDRESS=
0406 3245 DCA DSTADR /OF THE BUFFER,

```

```

0407 1215 DST0, TAD DSTADR /OBTAIN THE NUMBER DSTADR;
0410 7421 MQL /PLACE IT IN THE MQ;
0411 4547 JMS I [SETDF /SET DATA FIELD TO STORE IN;
0412 7501 MQA /LOAD AC WITH C(MQ);
0413 7191 CLL IAC /+1 TO AC;
0414 7445 DST /EAE DOUBLE PRECISION STORE;
0415 0000 DSTADR, OPEN /POINTER TO ADDRESS TO STORE IN;
0416 4566 JMS I [SAVREG /SAVE CONTENTS OF REGISTERS;
0417 1215 TAD DSTADR /LOAD AC WITH ADDRESS OF MQ DATA;
0420 7521 SWP /SWAP AC WITH ADDRESS OF MQ DATA;
0421 7457 SAM /SUBTRACT MQ AFTER DST FROM KNOWN GOOD;
0422 7640 SEA CLA /SKIP IF EQUAL;
0423 5244 JMP DSTERR /DST MODIFIED THE MQ;
0424 4567 JMS I [SETDF /SET DF WE STORED IN;
0425 1615 TAD I DSTADR /GET CONTENTS OF ADDRESS WHERE DST STORED MQ;
0426 7457 SAM /SUBTRACT STORED MQ FROM WHAT IT SHOULD BE;
0427 7640 SEA CLA /SKIP IF EQUAL;
0430 5244 JMP DSTERR /DST DID NOT STORE MQ PROPERLY;
0431 7573 DPIC /+1 TO MQ FOR CHECKING AC DATA;
0432 1022 TAD ACTOCK /LOAD AC WITH AC AFTER DST;
0433 7457 SAM /SUBTRACT AC AFTER DST FROM KNOWN GOOD;
0434 7640 SEA CLA /SKIP IF EQUAL;
0435 5244 JMP DSTERR /DST MODIFIED THE AC;
0436 7501 MQA /MQ TO AC;
0437 3033 DCA TEMP /SAVE IT IN TEMP SO=
0440 1433 TAD I TEMP /WE CAN OBTAIN STORED AC;
0441 7457 SAM /SUBTRACT AC FROM KNOWN GOOD;
0442 7650 SNA CLA /SKIP IF NOT EQUAL;
0443 5251 JMP DSTLOP /EVERYTHING A=OK;

0444 4572 DSTERR, JMS I [RSETDF /RESET DF;
0445 4565 JMS I [SW2 /SHALL WE PRINT?
0446 4272 JMS DSTERP /YES;
0447 4564 JMS I [SW0 /SHALL WE HALT?
0450 7402 DSTHLT, HLT /YES, DST ERROR;
0451 4572 DSTLOP, JMS I [RSETDF /RESET DF;
0452 4563 JMS I [SW1 /SHALL WE LOOP?
0453 5207 JMP /YES! USE SAME DATA;
0454 1215 TAD DSTADR /THE ADDITION OF THESE TWO NUMBERS=
0455 1142 TAD [=5776 /IS TO SEE IF 5776 HAS BEEN ACCESSED;
0456 7650 SNA CLA /AC=0, INDICATING IT HAS BEEN?
0457 5561 JMP I [NXTST /YES, GO TO NEXT SELECTED TEST;
0460 1215 TAD DSTADR /THE ADDITION OF THESE TWO NUMBERS=
0461 1140 TAD [1000 /IS TO SELECT THE NEXT BUFFER;
0462 3215 DCA DSTADR /NO SAVE THE NEXT BUFFER ADDRESS;
0463 1215 TAD DSTADR /LETS CHECK AND SEE IF=
0464 7104 RAL CLL /THE NEXT BUFFER ADDRESS IS=
0465 7700 SMA CLA /6000 OR MORE;
0466 5207 JMP DST0 /IT WASN'T SO GO TO BEGINNING;
0467 1215 TAD DSTADR /THE NEXT BUFFER WAS 6000 OR MORE=
0470 1157 TAD [=-2777 /SO SUBTRACT 2777 AND THIS WILL=
0471 5206 JMP DST0=1 /ALLOW BUFFER ADDRESS TO BE 3XXX+1;

0472 0000 DSTERP, OPEN
0473 1215 TAD DSTADR /OBTAIN ADDRESS OF LEAST SIGNIFICANT HALF;

```

```

0474 4556 JMS I [INDDMO /PRINT;
0475 6771 DSTTX /TEST IDENTIFIER;
0476 1215 TAD DSTADR /TO BE USED IN THE PRINT ROUTINE;
0477 4777 JMS DSTP2 /PRINT;
0500 5672 JMP I DSTERP /EXIT.

```

/DOUBLE PRECISION STORE TEST (DST) USING AUTO INDEX;
 /LOAD ADDRESSES 3000-5777 OF THE SELECTED DF WITH
 /A NUMBER EQUAL TO THE 1'S COMPLEMENT OF THE
 /ADDRESS.
 /DST TO EACH ADDRESS FROM 3000-5777 AND CHECK
 /RESULTS AFTER EACH DST.

```

0501 4571 DSTAIT, JMS I [LDCORE /LOAD CORE WITH A NUMBER EQUAL=
0502 7140 COMPLE /TO THE 1'S COMPLEMENT OF THE ADDRESS;
0503 1152 TAD [MUYTST /ADDRESS OF NEXT TEST;
0504 3042 DCA NEXT /AND SAVE FOR A POINTER;
0505 1151 TAD COST /GET A DST INSTRUCTION;
0506 3077 DCA AITEST /AND SAVE FOR EXECUTION;
0507 1155 TAD [2777 /OBTAIN THE INITIAL STARTING ADDRESS
0510 3010 DCA A110 /OF THE BUFFER +1 AND SAVE IT;
0511 1157 TAD [2777 /SET UP FOR A COUNT OF=
0512 3041 DCA COUNT /2777 OCTAL;
0513 7431 SWAB /SWITCH TO "BM" MODE OF OPERATION;
0514 1010 DSTAIR, TAD A110 /OBTAIN THE NUMBER IN A110=
0515 7001 IAC /AND ADD ONE TO IT THEN PLACE=
0516 7421 MQL /IT IN THE MQ;
0517 4567 JMS I [SETDF /SET DF TO STORE IN;
0520 7501 MQA /LOAD AC WITH C(MQ);
0521 7001 IAC /+1 TO AC;
0522 4776 JMS CHECAI /EXECUTE A DST IN THE AUTO INDEX REG;
0523 6774 DSTATX /TEST IDENTIFIER;
0524 4566 JMS I [SAVREG /SAVE THE REGISTERS;
0525 1010 TAD A110 /PUT CONTENTS OF A110 IN TEMP
0526 3033 DCA TEMP /SO WE DON'T AUTO=INDEX;
0527 1033 TAD TEMP /LOAD AC WITH ADDRESS OF MQ DATA;
0530 7521 SWP /SWAP AC WITH ADDRESS OF MQ DATA;
0531 7457 SAM /SUBTRACT MQ AFTER DST FROM KNOWN GOOD;
0532 7640 SEA CLA /SKIP IF EQUAL;
0533 5354 JMP DSTAIR /DST MODIFIED THE MQ;
0534 4567 JMS I [SETDF /SET DF WE STORED IN;
0535 1433 TAD I TEMP /GET CONTENTS OF ADDRESS WHERE DST STORED MQ;
0536 7457 SAM /SUBTRACT STORED MQ FROM WHAT IT SHOULD BE;
0537 7640 SEA CLA /SKIP IF EQUAL;
0540 5354 JMP DSTAIR /DST DID NOT STORE MQ PROPERLY;
0541 7573 DPIC /+1 TO MQ FOR CHECKING AC DATA;
0542 1022 TAD ACTOCK /LOAD AC WITH AC AFTER DST;
0543 7457 SAM /SUBTRACT AC AFTER DST FROM KNOWN GOOD;
0544 7640 SEA CLA /SKIP IF EQUAL;
0545 5354 JMP DSTAIR /DST MODIFIED THE AC;
0546 7501 MQA /MQ TO AC;
0547 3033 DCA TEMP /SAVE IT IN TEMP SO=
0550 1433 TAD I TEMP /WE CAN OBTAIN STORED AC;

```



```

0551 7457 SAM /SUBTRACT AC FROM KNOWN GOOD,
0552 7650 SNA CLA /SKIP IF NOT EQUAL,
0553 8381 JMP DSTATP /EVERYTHING A=OK,
0554 4572 DSTATP, JMS I CRSETDF /RESET THE DF,
0555 4563 JMS I CSW2 /SHALL WE PRINT?
0556 4367 JMS DSTATE /YES,
0557 4564 JMS I CSW2 /SHALL WE HALT?
0558 7432 HLT /YES, DST IN AN AUTO INDEX REG FAILED,
0559 4572 DSTATP, JMS I CRSETDF /RESET DF,
0560 4563 JMS I CSW1 /SHALL WE LOOP?
0561 4564 JMS I CRSETAI /YES, NOW GO AND RESET A110,
0562 2041 ISZ COUNT /FINISHED YET?
0563 8314 JMP DSTATP /NO, DO ANOTHER DST,
0564 5561 JMP I CNXTST /YES, GO TO NEXT TEST,

0567 0000 DSTATE, OPEN
0570 1010 TAD A110 /OBTAIN ADDRESS OF LEAST SIGNIFICANT HALF,
0571 4566 JMS I CINDDMO /PRINT,
0572 6774 DSTATX /TEST IDENTIFIER,
0573 1010 TAD A110 /TO BE USED IN THE PRINT ROUTINE,
0574 4777 JMS DSTOP2 /PRINT,
0575 5767 JMP I DSTATE /EXIT,

0576 2474
0577 4200
0600 PAGE

```

```

/MULTIPLY TEST (MUY)
/LOAD ADDRESSES 3000-5777 OF THE SELECTED DF WITH
/A NUMBER EQUAL TO THE 1'S COMPLEMENT OF THE ADDRESS,
/SIMULATE THE MULTIPLY INSTRUCTION STORING THE ANSWERS
/IN TWO HOLDING AREA
/ 3400-4377 FOR BUFFER 3000-3377
/ 5000-5777 FOR BUFFER 4400-4777
/EXECUTE A MUY INSTRUCTION AND ALTERNATE BETWEEN BUFFER
/3000-3377 AND 4400-4777,

```

```

0600 4571 MUYTST, JMS I CLOCORE /LOAD CORE WITH A NUMBER=
0601 7140 COMPLE /EQUAL TO THE COMPLEMENT OF THE ADDRESS
0602 1150 TAD CMUYAIT /ADDRESS OF NEXT TEST,
0603 3042 DCA NEXT /SAVE FOR A POINTER,
0604 4777 JMS MUYSIM /SIMULATE MULTIPLY,
0605 7300 CLA CLL /CLEAR AC FROM MUYSIM ROUTINE,
0606 1170 TAD C3000 /OBTAIN THE INITIAL STARTING ADDRESS=
0607 3223 DCA MUYSADR /OF THE BUFFER,
0608 1147 TAD C3400 /OBTAIN THE BEGINNING ADDRESS=
0609 3230 DCA MUYSRS /OF THE SIMULATED RESULTS AND STORE,
0610 1146 TAD C4400 /SET UP FOR A COUNT,
0611 3041 DCA COUNT /A COUNT OF 400 OCTAL,
0612 1146 DCA COUNT /SWITCH TO "B" MODE OF OPERATION,
0613 3041 SHAB /SET UP FOR=
0614 7431 CLA CLL CMA RAL /A COUNT OF 2,
0615 7344 DCA BOTH /SET DATA FIELD TO MULTIPLY FROM,
0616 3040 MUYV, JMS I CRSETDF
0617 4567

```

```

0620 1223 TAD MUYSADR /OBTAIN THE NUMBER IN MUYSADR,
0621 7100 CLL /
0622 7425 MQL MUY /EAE MULTIPLY,
0623 0000 MUYSADR, OPEN /CONTAINS ADDRESS WHERE DATA IS,
0624 4566 JMS I ISAVREG /SAVE THE REGISTER AND RESET DF,
0625 1022 TAD ACTOCK /BRING THE AC BACK,
0626 7575 DCM /NEGATE HQ AND AC,
0627 7443 DAD /ADD THE SIMULATED RESULTS
0630 0000 MUYSRS, OPEN /TO HQ AND AC,
0631 7451 DPEZ /SKIP IF REAL EQUALED SIMULATED,

0632 7610 SKP CLA /UNEQUAL RESULTS; MUY ERROR,
0633 5240 JMP MUYSLOP /EVERYTHING A=OK,
0634 4565 JMS I CSW2 /SHALL WE PRINT?
0635 4265 JMS MUYSRP /YES,
0636 4564 JMS I CSW2 /SHALL WE HALT?
0637 7402 MUYHLT, HLT /YES, MUY ERROR,
0640 4563 MUYSLOP, JMS I CSW1 /SHALL WE LOOP?
0641 5217 JMP MUYS /YES, USE SAME DATA,
0642 2040 ISZ BOTH /HAVE WE DONE 1 MUL FROM EACH BUFFER?
0643 7610 SKP CLA /NO, SET UP FOR NEXT BUFFER
0644 5254 JMP I,+10 /YES, SET UP TO GO TO 3XXX+1 BUFFER,
0645 1145 TAD C1400 /ADD 1400 TO MUYSADR=
0646 1223 TAD MUYSADR /IN ORDER TO PERFORM A MUY=
0647 3223 DCA MUYSADR /FROM THE HIGH BUFFER,
0650 1145 TAD C1400 /ADD 1400 TO MUYSRS IN =
0651 1230 TAD MUYSRS /ORDER TO ACCESS THE SIMULATED =
0652 3230 DCA MUYSRS /RESULTS FOR THE HIGH BUFFER,
0653 5217 JMP MUYS /GO DO A HIGH BUFFER MUY,
0654 1144 TAD C-1377 /SUBTRACT 1377 FROM MUYSADR
0655 1223 TAD MUYSADR /IN ORDER TO DO THE NEXT MUY FROM
0656 3223 DCA MUYSADR /THE LOW BUFFER -1,
0657 1143 TAD C-1376 /SUBTRACT 1376 FROM MUYSRS
0660 1230 TAD MUYSRS /IN ORDER TO OBTAIN SIMULATED RESULTS
0661 3230 DCA MUYSRS /FROM THE LOW BUFFER *2,
0662 2041 ISZ COUNT /FINISHED WITH HIGH BUFFER
0663 5215 JMP MUYS-2 /NO, GO DO MORE,
0664 5561 JMP I CNXTST /YES, GO TO NEXT SELECTED TEST,

0665 0000 MUYSRP, OPEN
0666 1223 TAD MUYSADR /OBTAIN ADDRESS OP=
0667 3033 DCA TEMP /MULTPLICAND,
0670 4542 JMS I CQPO /CONVERT IT TO
0671 0033 TEMP /PRINTABLE ASCII,
0672 7156 WD3
0673 4542 JMS I CQPO /CONVERT 0000
0674 0043 ZERO /TO PRINTABLE ASCII
0675 7152 WD2
0676 1223 TAD MUYSADR /OBTAIN ADDRESS OF THE MULTIPLICAND,
0677 4776 JMS DDP2 /TYPE ERROR DATA
0700 0005 MUYTXT /TEST IDENTIFIER
0701 7264 MULTIP /MULTPLICAND
0702 7242 MA /MA
0703 0630 MUYSRS /ADDRESS OF SIMULATED RESULTS

```

0704 7307 SC14 /SC=14
 0705 5665 JMP I MUYERP /EXIT

/MULTIPLY TEST (MUY) USING AUTO INDEX,
 /LOAD ADDRESSES 3000-5777 OF THE SELECTED OF WITH
 /A NUMBER EQUAL TO THE 1'S COMPLEMENT OF THE ADDRESS,
 /SIMULATE THE MULTIPLY INSTRUCTION STORING THE ANSWERS
 /IN TWO HOLDING AREAS
 / 3400-4377 FOR BUFFER 3000-3377
 / 5000-5777 FOR BUFFER 4400-4777
 /EXECUTE MULTIPLY DOING AREA 3000-3377 AND THEN
 /AREA 4400-4777,

0706 4571 MUYAIT, JMS I [LDCORE /LOAD CORE WITH A NUMBER=
 0707 7140 COMPLE /EQUAL TO THE COMPLEMENT OF THE ADDRESS,
 0710 1141 TAD [DIVTST /ADDRESS OF NEXT TEST,
 0711 3042 DCA NEXT /SAVE FOR A POINTER,
 0712 1155 TAD [2777 /OBTAIN THE INITIAL STARTING ADDRESS OF THE
 0713 3010 DCA A110 /BUFFER =1 AND SAVE IT,
 0714 1147 TAD [3400 /OBTAIN THE BEGINNING ADDRESS
 0715 4337 DCA MUYSAI /OF THE SIMULATED RESULTS AND STORE,
 0716 7431 SWAB /SWITCH TO "B" MODE OF OPERATION,
 0717 4777 JMS MUYSIM /SIMULATE A MULTIPLY,
 0720 7344 CLA CLL CMA RAL /SET UP FOR
 0721 3042 DCA BOTH /A COUNT OF 2,
 0722 1143 TAD [MQL MUY /GET A HOLD MUY INSTRUCTION
 0723 3007 DCA AITEST /AND SAVE FOR EXECUTION,
 0724 1146 TAD [-400 /SET UP FOR A COUNT
 0725 3041 DCA COUNT /OF 400 OCTAL,
 0726 4567 MUYA0, JMS I [SETOF /SET DATA FIELD TO MULTIPLY FROM,
 0727 1010 TAD A110 /OBTAIN THE NUMBER IN A110
 0730 7101 CLL IAC /AND ADD ONE TO IT,
 0731 4775 JMS CHECAI /GO DO A MULTIPLY IN THE AUTO=INDEX REGISTER.
 0732 7010 MUYATX /TEST IDENTIFIER,
 0733 4566 JMS I [SAVREG /AND SAVE ALL THE RESULTS,
 0734 1022 TAD ACTOCK /BRING THE "REAL" AC BACK,
 0735 7575 DCM /NEGATE THE AC AND MQ,
 0736 7443 DAD /ADD THE SIMULATED RESULTS=
 0737 0000 MUYSAI, OPEN /TO THE MQ AND AC,
 0740 7451 DPSZ /SKIP IF REAL EQUALS THE SIMULATED,
 0741 7610 SKP CLA /UNEQUAL RESULTS) MUY ERROR,
 0742 5347 JMP MUYALP /EVERYTHING A=OK,
 0743 4565 JMS I [SW2 /SHALL WE PRINT?
 0744 4774 JMS MUYAIE /YES,
 0745 4564 JMS I [SW0 /SHALL WE HALT?
 0746 7472 HLT /YES, MUY ERROR,
 0747 4563 MUYALP, JMS I [SW1 /SHALL WE LOOP?
 0750 7610 SKP CLA /YES,
 0751 5356 JMP ,+5 /NO,
 0752 7240 CLA CMA /SET THE ADDRESS IN=
 0753 1010 TAD A110 /THE AUTO=INDEX REGISTER=
 0754 3010 DCA A110 /SO WE USE THE SAME ADDRESS,

0755 5326 JMP MUYA0 /REPEAT TEST USING SAME DATA,
 0756 2337 ISZ MUYSAI /INCREMENT SIMULATED RESULTS=
 0757 2337 ISZ MUYSAI /ADDRESS,
 0760 2041 JMP COUNT /FINISHED BUFFER,
 0761 5326 MUYA0 /NO DO ANOTHER MULTIPLY,
 0762 2040 JMP ,+2 /ARE WE FINISHED WITH THE LAST BUFFER?
 0763 5365 JMP ,+2 /NO, JUST FINISHED THE FIRST,
 0764 5561 JMP I [NXTST /YES, GO TO NEXT TEST,
 0765 1137 TAD [4377 /SET UP FOR MULTIPLY
 0766 3010 DCA A110 /TO TAKE PLACE FROM THE HIGH BUFFER,
 0767 1136 TAD [5000 /SET UP FOR THE SIMULATED RESULTS
 0770 3337 DCA MUYSAI /FOR HIGH BUFFER MULTIPLY,
 0771 5324 JMP MUYA0=2 /RESET COUNT AND DO HIGH BUFFER,

0774 1000
 0775 2474
 0776 6400
 0777 2200
 1000

PAGE

1000 0000 MUYAIE, OPEN
 1001 1010 TAD A110 /OBTAIN MULTIPLICAND ADDRESS
 1002 3033 DCA TEMP /AND SAVE IT,
 1003 4542 JMS I [CPO /CONVERT IT TO
 1004 7033 TEMP /PRINTABLE ASCII
 1005 7156 WD3
 1006 4542 JMS I [CPO /CONVERT 0000 TO PRINTABLE ASCII
 1007 0043 ZERO
 1010 7152 WD2
 1011 1010 TAD A110 /OBTAIN MULTIPLICAND ADDRESS
 1012 4777 JMS MDP2 /TYPE ERROR DATA
 1013 7010 MUYATX /TEST IDENTIFIER
 1014 7264 MULTIP /MULTIPLICAND
 1015 7242 MA /MA
 1016 0737 MUYSAI /ADDRESS OF SIMULATED RESULTS,
 1017 7307 SC14 /SC=14
 1020 5600 JMP I MUYAIE /EXIT,

/DIVIDE TEST (DIV)
 /LOAD ADDRESSES 3000-5777 OF THE SELECTED OF WITH
 /A NUMBER EQUAL TO THE 1'S COMPLEMENT OF THE ADDRESS,
 /SIMULATE THE DIVIDE INSTRUCTION STORING THE
 /ANSWERS IN TWO HOLDING AREAS
 / 3400-4377 FOR BUFFER OF 3000-3377
 / 5000-5777 FOR BUFFER OF 4400-4777
 /DIVIDE EXECUTION WILL ALTERNATE BETWEEN BUFFER
 /3000-3377 AND 4400-4777,

1021 4571 DIVTST, JMS I [LDCORE /LOAD CORE WITH A NUMBER=
 1022 7140 COMPLE /EQUAL TO THE COMPLEMENT OF THE ADDRESS,
 1023 1376 TAD [DIVAIT /ADDRESS OF NEXT TEST,
 1024 3042 DCA NEXT /SAVE FOR A POINTER,
 1025 4775 JMS MUYSIM /SIMULATE DIVIDE

```

1026 1170 TAD C3002 /OBTAIN THE INITIAL STARTING ADDRESS =
1027 3247 DCA DIVADR /OF THE BUFFER;
1030 1147 TAD C3400 /OBTAIN THE BEGINNING ADDRESS =
1031 3297 DCA DIVSRS /OF THE SIMULATED RESULTS,
1032 1146 TAD C-400 /SET UP FOR=
1033 3041 DCA COUNT /A COUNT OF 400 OCTAL,
1034 7431 SWAB /SWITCH TO "R" MODE OF OPERATION,
1035 7344 CLA CLL CMA RAL /SET UP FOR=
1036 3040 DCA ROTH /A COUNT OF 2,
1037 4567 DIV0, JMS I CSETOP /SET DATA FIELD TO DIVIDE FROM,
1040 1247 TAD DIVADR
1041 7001 IAC
1042 7421 MQL
1043 7501 MQA
1044 7150 CLL CMA RAR
1045 7100 CLL
1046 7407 DIV /EAE DIVIDE,
1047 0000 DIVADR, OPEN /CONTAINS ADDRESS OF DIVISOR,
1050 4566 JMS I CSAVEG /SAVE REGISTERS
1051 1020 TAD LKTOCK /CHECK FOR A DIVIDE OVERFLOW,
1052 7710 SPA CLA /SKIP IF A NORMAL DIVIDE,
1053 5263 JMP C+10 /DIVIDE OVERFLOW OCCURRED,
1054 1022 TAD ACTOCK /BRING THE AC BACK,
1055 7575 DCM /NEGATE MQ AND AC,
1056 7443 DAD /ADD THE SIMULATED RESULTS
1057 0000 DIVSRS, OPEN /TO MQ AND AC,
1060 7451 DPSE /SKIP IF REAL EQUALED SIMULATED,
1061 7610 SKP CLA /UNEQUAL RESULTS) DIVIDE ERROR
1062 5267 JMP DIVLOP /EVERYTHING A=OK,
1063 4565 JMS I CSW2 /SHALL WE PRINT?
1064 4314 JMS DIVERP /YES,
1065 4564 JMS I CSW0 /SHALL WE HALT?
1066 7402 DIVHLT, HLT /YES, DIVIDE ERROR,
1067 4563 DIVLOP, JMS I CSW1 /SHALL WE LOOP?
1070 5237 JMP DIV0 /YES, USE SAME DATA,
1071 2040 ISZ BOTH /HAVE WE DONE 1 DIV FROM EACH BUFFER,
1072 7610 SKP CLA /NO, SET UP FOR NEXT BUFFER,
1073 5303 JMP C+10 /YES, SET UP TO GO TO 3XXX+1 BUFFER
1074 1145 TAD C1400 /ADD 1400 TO DIVADR=
1075 1247 TAD DIVADR /IN ORDER TO PERFORM A DIV=
1076 3247 DCA DIVADR /FROM THE HIGH BUFFER,
1077 1145 TAD C1400 /ADD 1400 TO DIVSRS IN=
1100 1257 TAD DIVSRS /ORDER TO ACCESS THE SIMULATED=
1101 3257 DCA DIVSRS /RESULTS FOR THE HIGH BUFFER,
1102 5237 JMP DIV0 /GO DO A HIGH BUFFER DIV,
1103 1144 TAD C-1377 /SUBTRACT 1377 FROM DIVADR
1104 1247 TAD DIVADR /IN ORDER TO DO THE NEXT DIV FROM
1105 3247 DCA DIVADR /THE LOW BUFFER +1,
1106 1143 TAD C-1376 /SUBTRACT 1376 FROM DIVSRS
1107 1257 TAD DIVSRS /IN ORDER TO OBTAIN SIMULATED RESULTS
1110 3257 DCA DIVSRS /FROM THE LOW BUFFER +2,
1111 2041 ISZ COUNT /FINISHED WITH HIGH BUFFER,
1112 5235 JMP DIV0-2 /NO, GO DO MORE,
1113 5501 JMP I CNXTST /YES, GO TO NEXT SELECTED TEST,

```

```

1114 0000 DIVERP, OPEN
1115 1247 TAD DIVADR /OBTAIN DIVISOR ADDRESS
1116 7041 CMA IAC /NEGATE IT FOR GOOD MQ
1117 7033 DCA TEMP
1120 4542 JMS I CPO /CONVERT IT TO PRINTABLE ASCII
1121 0033 TEMP
1122 7196 WDS
1123 1033 TAD TEMP /CONVERT GOOD MQ TO GOOD AC
1124 7150 CLL CMA RAR
1125 0033 DCA TEMP
1126 4542 JMS I CPO /CONVERT TEMP TO PRINTABLE ASCII
1127 0033 TEMP
1130 7192 WDS
1131 1247 TAD DIVADR /DIVISOR ADDRESS
1132 4777 JMS MDP2 /PRINT ERROR DATA
1133 7021 DIVTXY /TEST IDENTIFIER
1134 7274 DIVSOR /DIVISOR
1135 7246 DA /DA
1136 1097 DIVSRS /ADDRESS OF SIMULATED RESULTS
1137 7313 SC15 /SC=15
1140 5714 JMP I DIVERP /EXIT

```

```

1175 2242
1176 1200
1177 6400
1200 PAGE

```

```

/DIVIDE TEST (DIV) USING AUTO INDEX,
/LOAD ADDRESSES 3000-3777 OF THE SELECTED OF WITH
/A NUMBER EQUAL TO THE 1'S COMPLEMENT OF THE ADDRESS;
/SIMULATE THE DIVIDE INSTRUCTION STORING THE ANSWER
/IN TWO HOLDING AREAS;
/ 3400-4377 FOR BUFFER 3000-3377
/ 5000-5777 FOR BUFFER 4400-4777
/EXECUTE DIVIDE DOING AREA 3000-3377 AND THEN
/AREA 4400-4777.

```

```

1200 4571 DIVAIT, JMS I CLDCORE /LOAD CORE WITH A NUMBER=
1201 7140 COMPLE /EQUAL TO THE COMPLEMENT OF THE ADDRESS,
1202 1377 TAD CEND /ADDRESS OF NEXT TEST;
1203 3042 DCA NEXT /SAVE AS A POINTER;
1204 4776 JMS DIVSIM /SIMULATE DIVIDE
1205 1155 TAD C2777 /OBTAIN THE INITIAL STARTING ADDRESS OF=
1206 3010 DCA A110 /THE BUFFER -1 AND SAVE IT,
1207 1147 TAD C3400 /OBTAIN THE BEGINNING ADDRESS=
1210 3241 DCA DIVSAI /OF THE SIMULATED RESULTS AND STORE,
1211 7431 SWAB /SWITCH TO "R" MODE OF OPERATION,
1212 7344 CLA CLL CMA RAL /SET UP FOR=
1213 3040 DCA ROTH /A COUNT OF 2;
1214 1135 TAD C01V /GET A DIVIDE INSTRUCTION=
1215 3007 DCA AITEST /AND SAVE FOR EXECUTION,

```

```

1216 1140 TAD C=400 /SET UP FOR A COUNT=
1217 3041 DCA COUNT /OF 400 OCTAL,
1220 4567 DIVA0, JMS I [SETDF /SET DATA FIELD TO DIVIDE FROM,
1221 7101 CLL IAC /OBTAIN THE NUMBER IN A110
1222 7001 IAC
1223 1010 TAD A110 /AND SUBTRACT ONE FROM IT;
1224 7421 MQL /AND LOAD IT IN THE MQ;
1225 7521 MQA /NOW LOAD THE AC WITH THE
1226 7150 CLL CMA RAR /1'S COMPLEMENT OF THE MQ;
1227 7100 CLL
1230 4775 JMS CHECAI /GO DO A DIVIDE IN THE AUTO=INDEX REGISTER;
1231 7024 DIVATX /TEST IDENTIFIER;
1232 4566 JMS I [SAVREG /SAVE ALL THE RESULTS
1233 1020 TAD LKTOCK /CHECK FOR A DIVIDE OVERFLOW
1234 7710 SPA CLA /SKIP IF A NORMAL DIVIDE
1235 5245 JMP ,+10 /DIVIDE OVERFLOW OCCURRED;
1236 1022 TAD ACTOCK /BRING THE REAL AC BACK
1237 7575 DCM /NEGATE AC AND MQ;
1240 7443 DAD /ADD THE SIMULATED RESULTS=
1241 0000 DIVSAI, OPEN /TO THE MQ AND AC;
1242 7451 DPMZ /SKIP IF REAL EQUALLED THE SIMULATED
1243 7610 SKP CLA /UNEQUAL RESULTS; DIV ERROR,
1244 5251 JMP DIVALP /EVERYTHING A=OK;
1245 4545 JMS I [SW2 /SHALL WE PRINT?
1246 4274 JMS DIVAIE /YES;
1247 4564 JMS I [SW0 /SHALL WE HALT?
1250 7402 HLT /YES, DIVIDE ERROR;
1251 4543 DIVALP, JMS I [SW1 /SHALL WE LOOP?
1252 7610 SKP CLA /YES;
1253 5260 JMP ,+5 /NO;
1254 7242 CLA CMA /SET THE ADDRESS IN=
1255 1010 TAD A110 /THE AUTO=INDEX REGISTER=
1256 3010 DCA A110 /SO WE USE THE SAME ADDRESS
1257 5220 JMP DIVA0 /REPEAT TEST USING SAME DATA
1260 2241 ISZ DIVSAI /INCREMENT SIMULATED RESULTS=
1261 2241 ISZ DIVSAI /ADDRESS;
1262 2041 ISZ COUNT /FINISHED BUFFER?
1263 5220 JMP DIVA0 /NO, DO ANOTHER DIVIDE
1264 2040 ISZ BOTH /ARE WE FINISHED WITH THE LAST BUFFER?
1265 5267 JMP ,+2 /NO, JUST FINISHED THE FIRST;
1266 5561 JMP I [NXTST /YES, GO TO NEXT TEST;
1267 1137 TAD [4377 /SET UP FOR DIVIDE TO TAKE=
1270 3010 DCA A110 /PLACE FROM THE HIGH BUFFER;
1271 1136 TAD [5000 /SET UP FOR THE SIMULATED RESULTS=
1272 3241 DCA DIVSAI /FOR HIGH BUFFER DIVIDE;
1273 5216 JMP DIVA0=2 /RESET COUNT AND ON HIGH BUFFER,

1274 0000 DIVAIE, OPEN
1275 1010 TAD A110 /OBTAIN DIVISOR ADDRESS
1276 7041 CMA IAC /NEGATE IT FOR GOOD MQ
1277 3033 DCA TEMP /CONVERT IT=
1300 4542 JMS I [CPO /TO PRINTABLE ASCII!
1301 0033 TEMP
1302 7156 W03
1303 1033 TAD TEMP /

```

```

1304 7150 CLL CMA RAR /CONVERT GOOD MQ TO
1305 3033 DCA TEMP /GOOD AC
1306 4542 JMS I [CPO /AND CONVERT IT=
1307 0033 TEMP /TO ASCII!
1310 7152 W02
1311 1010 TAD A110 /GET DIVISOR ADDRESS
1312 4774 JMS HDP2 /PRINT ERROR DATA
1313 7024 DIVATX /TEST IDENTIFIER
1314 7274 DIVSOR /DIVISOR
1315 7246 DA /DA
1316 1241 DIVSAI /ADDRESS OF SIMULATED RESULTS
1317 7313 SC15 /SC=15
1320 5674 JMP I DIVAIE /EXIT

1374 6400
1375 2474
1376 2242
1377 1400
PAGE

```

/ROUTINE TO TAKE CARE OF CHANGING OF DF
/AND IF AT THE PROPER TIME,

```

1400 7300 END, CLA CLL /
1401 1377 TAD [0ADTST /ADDRESS OF NEXT TEST
1402 3042 DCA NEXT /SAVE AS A POINTER;
1403 7604 LAS /READ SWITCHES
1404 7112 CLL RTR /MOVE SR8 INTO=
1405 7012 RTR /THE LINK;
1406 7430 SEL /IS SR8=1?
1407 5776 JMP SELTST /YES, SELECT TEST AND HOLD DF;
1410 0375 AND [20 /MASK FOR SELECTED TEST HOLD;
1411 7640 SZA CLA /WERE ANY OF THOSE SET?
1412 5776 JMP SELTST /YES SO SELECT TEST AND HOLD DF;
1413 1026 TAD HIFLD /LETS SEE IF THERE IS MORE THAN=
1414 7650 SNA CLA /FIELD 0;
1415 5774 JMP ENDTYP /ONLY FIELD 0 EXISTS; GO TO END OF TEST FIRST;
1416 1032 TAD DFFLD /GET PRESENT DF AND=
1417 1134 TAD [10 /ADD 10 FOR NEXT DF;
1420 7421 MQL /STORE IN MQ;
1421 7501 MQA /GET IT BACK;
1422 7041 CIA /2'S COMPLEMENT AND ADD HIGHEST
1423 1026 TAD HIFLD /FIELD TO IT;
1424 7710 SPA CLA /ARE WE EQUAL TO OR LESS THAN HIGHEST DF;
1425 7621 CAM /NO, OVER SO SET DF FOR 0;
1426 7501 MQA /NEW DF TO AC;
1427 7041 CIA /2'S COMPLEMENT AND ADD THE
1430 1030 TAD BEGDF /BEGINNING OF TO IT;
1431 7650 SNA CLA /IS IT EQUAL TO BEGINNING DF?
1432 5236 JMP ,+4 /YES;
1433 7501 MQA /SET LOCATION DFFLD FOR NEW
1434 3032 DCA DFFLD /DF AND NOW GO AND

```

```

1435 5776' JMP SELTST /SELECT A TEST;
1436 7604 LAS /READ BR;
1437 1102 RSW CLL /PUT SR7 INTO
1440 7006 RTL /THE LINK;
1441 7630 SEL CLA /IS SR7=1?
1442 5233 JMP ,=7 /YES, HOLD IF BUT CHANGE DF;
1443 7501 MQA /GET NEW DF AND
1444 3032 DCA DFFLD /SET DFFLD;
1445 1031 TAD IFFLD /GET PRESENT IF
1446 1134 TAD C10 /AND ADD 10 TO IT;
1447 7421 MQL /NOW SAVE IT IN THE MQ;
1450 7501 MQA /RETRIEVE IT;
1451 7041 CIA /2'S COMPLEMENT IT AND ADD THE VALUE
1452 1026 TAD HIFLD /OF THE HIGHEST IF FIELD TO IT;
1453 7710 SPA CLA /ARE WE OVER THE HIGHEST?
1454 7621 CAM /YES, SET FOR AN IF OF R;
1455 7501 MQA /GET NEW IF AND PLACE IN AC;

1456 1133 TAD CDF 00 /VALUE EQUAL TO CDF 0;
1457 3276 DCA END2 /STORE THE INSTRUCTION;
1460 7501 MQA /NEW IF TO AC;
1461 3031 DCA IFFLD /AND SAVE THE IF;
1462 3015 DCA A115 /CLEAR A115;
1463 3016 DCA A116 /CLEAR A116;
1464 3017 DCA A117 /CLEAR A117;
1465 1132 TAD C=7600 /SET UP SO ALL IS TRANSFERRED
1466 3033 DCA TEMP /EXCEPT LAST PAGE OF MEMORY;
1467 6224 RIF /GET PRESENT IF=
1470 1133 END1. TAD CDF 00 /AND MAKE=
1471 3272 DCA ,*1 /THE DATA FIELD=
1472 0000 OPEN /THE SAME;
1473 1416 TAD I A116 /MOVE THE CONTENTS OF=
1474 3037 DCA TEMP4
1475 1037 TAD TEMP4
1476 0000 END2. OPEN /PRESENT FIELD=
1477 3417 DCA I A117 /TO A NEW FIELD;
1500 1415 TAD I A115 /NOW LET'S CHECK=
1501 7041 CMA !AC /AND SEE IF THE
1502 1037 TAD TEMP4 /WORD WAS TRANSFERRED CORRECTLY;
1503 7650 SNA CLA /WAS IT?
1504 5307 JMP ,*3 /YES;
1505 7402 HLT /RELOCATION FAILURE; ADDRESS 16 OF
/INDICATED "IF" CONTAINS THE ADDRESS
/THAT WAS BEING TRANSFERRED AND
/ADDRESS 17 OF THE INDICATED "IF"
/IS THE ADDRESS THE DATA WAS BEING
/TRANSFERRED TO IN THE INDICATED "DF";
/RESTART THE PROGRAM;
/TO PREVENT PROGRAM FROM BEING
/CONTINUED BECAUSE OF A FATAL ERROR;

1506 5305 JMP ,*1 /ALL OK FINISHED?
1507 2033 ISZ TEMP /NO;
1510 5267 JMP END1 /GET VALUE OF NEW IF
1511 7501 MQA /AND ADD C10 TO IT;
1512 1131 TAD C6202
1513 3314 DCA ,*1 /STORE IT FOR EXECUTION;
    
```

```

1514 0000 OPEN /C10 X;
1515 5316 JMP ,*1 /CHANGE IF;
1516 4572 JMS I CRSETDF /SET DF EQUAL TO IF;
1517 7501 MQA /GET IF AND CHANGE TO=
1520 7041 CIA /2'S COMPLEMENT AND ADD=
1521 1027 TAD BEGIF /THE BEGINNING IF TO IT;
1522 7650 SNA CLA /ARE THEY THE SAME?
1523 5774 JMP ENDTYP /YES, ONE COMPLETE PASS NOW MADE;
1524 5776' JMP SELTST /NO, SELECT A NEW TEST;
    
```

/ROUTINE TO CONVERT A WORD TO PRINTABLE ASCII (2 WORDS);

```

1525 3000 CPO, OPEN
1526 7200 CLA ASCT
1527 1725 TAD I CPO
1530 3362 DCA WASC
1531 2325 ISZ CPO
1532 1725 TAD I CPO
1533 3363 DCA SASC
1534 2325 ISZ CPO
1535 1365 TAD K7700
1536 0762 AND I WASC
1537 7102 BSW CLL
1540 4347 JMS CNV
1541 2363 ISZ SASC
1542 1365 TAD K7700
1543 7040 CMA
1544 0762 AND I WASC
1545 4347 JMS CNV
1546 5725 JMP I CPO /EXIT CPO SUBROUTINE

1547 0000 CNV, OPEN
1550 3364 DCA ASCT
1551 1364 TAD ASCT
1552 7006 RTL
1553 7004 RAL
1554 0366 AND K0707
1555 1364 TAD ASCT
1556 0366 AND K0707
1557 1367 TAD K6060
1560 3763 DCA I SASC
1561 5747 JMP I CNV
1562 0000 WASC, OPEN
1563 0000 SASC, OPEN
1564 0000 ASCT, OPEN
1565 7700 K7700, 7700
1566 0707 K0707, 0707
1567 0062 K6060, 6060
    
```

1574 2135
1575 0020
1576 2074

9

```

1577 0227
1600 1600 PAGE
/SUBROUTINE TO TYPE CHARACTER STRING
1600 0000 UTYPE, OPEN
1601 7200 CLA
1602 1600 TAD I UTYPE /GET AND STORE STRING ADDR,
1603 3266 DCA TEMQ /CLEAR SPECIAL CHAR FLAG,
1604 3270 DCA FLAG /SET UP EXIT ADDR,
1605 2200 ISZ UTYPE
1606 1666 TSC1, TAD I TEMQ
1607 7002 BSW /ROTATE 6 RIGHT,
1610 4215 JMS TSC2 /GO TYPE CHARACTER
1611 1666 TAD I TEMQ /GET DATA
1612 4215 JMS TSC2 /GO TYPE CHARACTER
1613 2266 ISZ TEMQ /*1 TO STRING ADDR,
1614 5206 JMP TSC1 /GO BACK FOR MORE
1615 0000 TSC2, 0
1616 0130 AND L77 /MASK OFF 6 BITS
1617 3267 DCA TEMR /SAVE
1620 1270 TAD FLAG
1621 7640 SZA CLA /FLAG SET?
1622 5232 JMP TYPSP /YES
1623 1267 TAD TEMR /NO,
1624 7450 SNA /TEMR=0?
1625 5230 JMP ,+3 /YES,
1626 4243 TYPAT, JMS PRINT /NO, PRINT IT
1627 5615 JMP I TSC2 /RETURN
1630 2270 ISZ FLAG /TEMR=0, SET FLAG
1631 5615 JMP I TSC2 /EXIT,
1632 3270 TYPSP, DCA FLAG /CLEAR FLAG,
1633 1267 TAD TEMR
1634 7450 SNA /0?
1635 5615 JMP I TSC2 /IGNORE IT
1636 1272 TAD M77
1637 7650 SNA CLA /77?
1640 5600 JMP I UTYPE /YES, EXIT CODE
1641 1267 TAD TEMR
1642 5226 JMP TYPAT
1643 0000 PRINT, OPEN
1644 1273 TAD M45
1645 7640 SZA CLA /IS IT 45?
1646 5252 JMP ,+4 /NO,
1647 1300 TAD CR215 /YES, PRINT CR
1650 4302 JMS TTOUT /GO TYPE
1651 5643 JMP I PRINT
1652 1267 TAD TEMR
1653 1274 TAD M43
1654 7640 SZA CLA /IS IT 43?
1655 5260 JMP ,+3 /NO,
1656 1277 TAD LF212 /YES, OUTPUT LF,
1657 5250 JMP PRINT+5

```

```

1660 1267 TAD TEMR
1661 1275 TAD M40
1662 7510 SPA
1663 1276 TAD P100
1664 1301 TAD SP240
1665 5250 JMP PRINT+5
1666 0000 TEMQ, 0
1667 0000 TEMR, 0
1670 0000 FLAG, 0
1671 0000 PFLAG, 0
1672 7701 M77, -77
1673 7733 M45, -45
1674 7735 M43, -43
1675 7740 M40, -40
1676 0100 P100, 0100
1677 0212 LF212, 212
1700 0215 CR215, 215
1701 0240 SP240, 240
/ACTUAL PRINT ROUTINE,
1702 0000 TTOUT, OPEN
1703 6046 TLA /PRINT CHARACTER IN AC,
1704 6041 TSP /DONE PRINTING YET?
1705 5304 JMP ,+1 /NO,
1706 6042 TCF /CLEAR THE PRINTER FLAG,
1707 7300 CLA CLL /AND ALSO THE AC,
1708 5702 JMP I TTOUT /EXIT, AC + LINK = 0
/ROUTINE TO PRINT 1 SPACE,
1711 0000 SPACE1, OPEN
1712 4577 TYPE /DO JUST=
1713 6661 SPAC1 /ONE SPACE AND
1714 5711 JMP I SPACE1 /THEN EXIT,
/ROUTINE TO DO NUMBER OF SPACES
/IN LOCATION FOLLOWING THE CALL,
1715 0000 XSPACE, OPEN
1716 1715 TAD I XSPACE /GET DESIRED NUMBER OF SPACES
1717 7041 CIA /NEGATE IT,
1720 3037 DCA TEMP4 /STORE IT,
1721 4311 JMS SPACE1 /SPACE 1,
1722 2037 ISZ TEMP4 /DONE REQUESTED SPACES?
1723 5321 JMP ,+2 /NO, THEN DO ANOTHER ONE,
1724 5715 JMP I XSPACE /EXIT,
/ROUTINE TO PRINT CONTENTS OF 1 BIT,
/ENTER WITH BIT IN AC,
/AC=01 PRINT 0,
/AC NON 01 PRINT 1,

```

```

1725 0000 BIT1, OPEN
1726 7640 SZA CLA /WHAT TO PRINT
1727 7081 IAC /AC NON 0 SO SET FOR A 1
1730 1127 TAO [60 /ADD 60 TO PRESENT AC
1731 3777' DCA BITX /SAVE IT
1732 4577 TYPE /TYPE THE BIT,
1733 7142 BITX /THIS BIT,
1734 5725 JMP I BIT1 /EXIT, AC + L = 0;
    
```

/ROUTINE TO TYPE THE NUMBER OF WORDS IN
/THE LOCATION FOLLOWING THE CALL,

```

1735 0000 XTYPE, OPEN
1736 1735 TAD I XTYPE /GET NUMBER OF WORDS TO BE TYPED,
1737 7041 CMA IAC /NEGATE IT,
1740 3352 DCA XINPUT /NOW SAVE THE NEGATIVE COUNT,
1741 2335 ISZ XTYPE /GET READY TO PULL A WORD,
1742 1735 TAD I XTYPE /GET THE WORD,
1743 3345 DCA ,=2 /SAVE THE WORD,
1744 4577 TYPE /GO TYPE THE WORD,
1745 0000 OPEN /THIS WORD,
1746 2352 ISZ XINPUT /FINISHED ALL YET?
1747 5341 JMP ,=6 /NO,
1750 2335 ISZ XTYPE /YES, SET UP FOR AN EXIT,
1751 5735 JMP I XTYPE /NOW EXIT,
    
```

```

1752 0000 XINPUT, OPEN
1753 6032 KCC
1754 6031 KSF
1755 5354 JMP ,=1
1756 6036 KRB
1757 5752 JMP I XINPUT /EXIT WITH CHARACTER IN AC,
    
```

/ROUTINE TO SAVE CONTENTS OF REGISTERS,

```

1760 0000 SAVREG, OPEN
1761 3022 DCA ACTOCK /SAVE AC,
1762 7010 RAR /
1763 3020 DCA LKTOCK /SAVE LINK,
1764 7501 MQA /
1765 3021 DCA MQTOCK /SAVE MQ,
1766 6214 RDF /
1767 3024 DCA DFTOCK /SAVE DF,
1770 6224 RIF /
1771 3025 DCA IFTOCK /SAVE IF,
1772 7441 SCA /
1773 3023 DCA SCTOCK /SAVE SC,
1774 4572 JMS I ERBETDF /RESET DATA FIELD
1775 5760 JMP I SAVREG /EXIT, AC AND LINK=0

1777 7142
2000 PAGE
    
```

/ROUTINE TO FIND OUT HOW MANY MEMORY
/FIELDS EXIST,

```

2000 0000 ASKFLD, OPEN
2001 4577 TYPE /PRINT THE MESSAGE
2002 7220 HF /HIGHEST FIELD (0=7)?
2003 4574 INPUT /INPUT A CHARACTER FROM THE TTY,
2004 1377 TAO (=0 /IS THE CHARACTER VALUE
2005 7710 SPA CLA /LESS THAN 260?
2006 5201 JMP ASKFLD+1 /YES, RETYPE QUESTION,
2007 6034 KRS /READ CHARACTER INTO AC,
2010 1376 TAO (=270 /IS THE CHARACTER VALUE
2011 7700 SMA CLA /MORE THAN 267?
2012 5201 JMP ASKFLD+1 /YES, RETYPE QUESTION,
2013 6034 KRS /READ CHARACTER INTO AC,
2014 3173 AND [7 /SAVE 3 LEAST SIGNIFICANT BITS=
2015 7106 RTL CLL /AND PLACE THEM IN
2016 7004 RAL /AC 6=0,
2017 3026 DCA HIFLD /NOW SAVE THEM,
2020 1026 TAO HIFLD /PRINT IT BACK,
2021 7112 RTR CLL
2022 7010 RAR
2023 1127 TAO [60 /ADD 60 TO MAKE IT PRINTABLE,
2024 4775' JMS TFOUT /PRINT IT,
2025 4574 INPUT /INPUT A CHARACTER,
2026 1374 TAO (=377 /ADD 2'S COMPLEMENT OF A RUBOUT,
2027 7650 SNA CLA /WAS IT A RUBOUT?
2030 5201 JMP ASKFLD+1 /YES, THE OPERATOR DESIRES TO RESELECT,
2031 6036 KRB /GET THE CHARACTER AGAIN,
2032 1373 TAO (=215 /ADD 2'S COMPLEMENT OF A CARRIAGE RETURN,
2033 7640 SZA CLA /WAS IT A CARRIAGE RETURN?
2034 5225 JMP ,=7 /NO, GO TRY AGAIN,
2035 4577 TYPE /GO DO A CR AND LF,
2036 6610 CRLF
2037 5600 JMP I ASKFLD /EXIT, AC=LINK=0;
    
```

/ROUTINE TO CHECK SR0,
/EXIT +1 IF SWITCH IS SET,

```

2040 0000 SW0, OPEN
2041 7604 LAS /READ SR,
2042 7700 SMA CLA /CHECK CONDITION,
2043 5640 JMP I SW0 /SR0=0,
2044 2240 ISZ SW0 /+1 TO EXIT,
2045 5640 JMP I SW0 /SR0=1,
    
```

/ROUTINE TO CHECK SR1,
/EXIT +1 IF SWITCH IS 0

//

```

/KEB-E EAE EXTENDED MEMORY EXERCISER MAINDEC=08-DHKEA=A=L PAL10 V141 31=AUG=72 22149 PAGE 1=23

2046 0000 SW1, OPEN /READ SR;
2047 7604 LAS /AC1 TO AC0;
2050 7104 RAL CLL /CHECK CONDITION;
2051 7710 SPA CLA /SR1=1;
2052 5646 JMP I SW1 /+1 TO EXIT;
2053 2246 ISE SW1 /SR1=0;
2054 5646 JMP I SW1

/ROUTINE TO CHECK SR2,
/EXIT +1 IF SWITCH IS SET;

2055 0000 SW2, OPEN /READ SR;
2056 7604 LAS /AC2 TO AC0;
2057 7106 RTL CLL /CHECK CONDITION;
2060 7700 SMA CLA /SR2=0;
2061 5655 JMP I SW2 /+1 TO EXIT;
2062 2255 ISE SW2 /SR2=1;
2063 5655 JMP I SW2

/ROUTINE TO GO TO NEXT TEST IF NO
/SWITCHES ARE SELECTED FOR A PARTICULAR
/TEST;

2064 7604 NXTST, LAS /READ SWITCHES;
2065 0372 AND (400 /LOOK AT ONLY TEST SELECT SWITCH;
2066 7640 SEA CLA /ARE ANY OF THEM SET?
2067 5526 JMP I [SELTST /YES;
2070 1042 TAD NEXT /GET NEXT TEST POINTER;
2071 3273 DCA ,+2 /STORE IT TO JUMP TO
2072 5673 JMP I ,+1 /NOW JUMP TO IT;
2073 0000 OPEN /POINTER TO JUMP TO;

/ROUTINE TO SELECT A TEST IF A SWITCH
/IS SET;

2074 7604 SELTST, LAS /READ SWITCHES;
2075 7106 RTL CLL /CHECK FOR
2076 7004 RAL /SR 3 SET?
2077 7500 SMA /IS IT SET?
2080 5264 JMP NXTST /NO, GO TO NEXT TEST;
2101 7102 BSW CLL /PUT SR 4=6 IN AC 9=11
2102 7012 RTR /
2103 0173 AND [7 /KEEP ONLY SR 4=6;
2104 1125 TAD [SWTAB /ADD TABLE ADDRESS AND
2105 3037 DCA TEMP4 /SAVE AND NOW
2106 1437 TAD I TEMP4 /GET THE TEST ADDRESS
2107 3037 DCA TEMP4 /AND NOW
2110 5437 JMP I TEMP4 /JUMP TO THAT TEST;

/ROUTINE TO SET DP;

2111 0000 SETDF, OPEN
2112 7300 CLA CLL
2113 1032 TAD DFFLD

```

/KEB-E EAE EXTENDED MEMORY EXERCISER MAINDEC=08-DHKEA=A=L PAL10 V141 31=AUG=72 22149 PAGE 1=23

```

2114 1133 TAD [CDF 00
2115 3316 DCA ,+1
2116 0000 OPEN
2117 5711 JMP I SETDF

/ROUTINE TO SET DP EQUAL TO IF;

2120 0000 RSETDF, OPEN
2121 7200 CLA
2122 6224 RIF
2123 1133 TAD [CDF 00
2124 3325 DCA ,+1
2125 0000 OPEN
2126 5720 JMP I RSETDF

/ROUTINE TO SUBTRACT ONE FROM A110 AND EXIT +1;

2127 0000 RSETA1, OPEN
2130 7240 CLA CMA /AC=-1
2131 1010 TAD A110 /ADD A110 TO IT;
2132 3010 DCA A110 /RESET A110;
2133 2327 ISE RSETA1 /+1 FOR EXIT;
2134 5727 JMP I RSETA1 /NOW EXIT, AC=0;

/ROUTINE TO TYPE OUT TO SIGNIF
/THE END OF ONE PASS OF THE TEST;

2135 4565 ENDTYP, JMS I [SW2 /IF SW 2 SET JUST "DONE";
2136 5342 JMP ,+4 /SW 2 IS NOT SET SO PRINT;
2137 1124 TAD [207 /BELL CODE;
2140 4775 JMS TFOUT /RING THE BELL;
2141 5264 JMP NXTST /GO TO NEXT TEST;
2142 4577 TYPE
2143 7212 KESEME
2144 5264 JMP NXTST /GO TO NEXT TEST;

/ROUTINE TO LOAD CORE FROM ADDRESS 3000 TO 5777
/WITH A NUMBER EQUAL TO THE ADDRESS OR ITS
/COMPLEMENT, THE LOCATION FOLLOWING THE CALL
/WILL BE THE DETERMINER, EQUAL (7000) MEANS
/LOAD CORE WITH CONTENTS EQUAL TO THE ADDRESS, AND
/COMPL (7342) MEANS TO LOAD CORE WITH
/THE 1'S COMPLEMENT OF THE ADDRESS;

2145 0000 LDCORE, OPEN
2146 7300 CLA CLL
2147 4523 JMS I [LZERO
2150 1136 TAD [3000
2151 3041 DCA COUNT /COUNT SET FOR 3000 OCTAL ADDRESSES;
2152 1745 TAD I LDCORE /GET INSTRUCTION FOR SETTING CORE FOR=
2153 3361 DCA ,+6 /EQUAL OR COMPLEMENT AND STORE;
2154 1371 TAD [2777 /SET AUTO INDEX TO START+
2155 3016 DCA A116 /WITH ADDRESS 3000;
2156 4567 JMS I [SETDF /SET DATA PIFLD TO BE SET UP;
2157 1016 TAD A116 /GET C(A116) AND ADD ONE TO=

```



```

2160 7001      IAC          /IT SINCE A110 IS ONE LESS,
2161 0000      OPEN         /EITHER A "NOP" OR "CLA CLL CMA",
2162 3410      DCA !      4116  /STORE THE NUMBER,
2163 2041      ISZ         COUNT  /FINISHED ALL 2000 ADDRESSES YET?
2164 5357      JMP          /NO, REPEAT OPERATION,
2165 4572      JMS I      LRSETOF /YES, RESET THE DATA FIELD,
2166 2345      ISZ         LDCORE  /SETUP FOR EXIT,
2167 5745      JMP I      LDCORE  /EXIT, AC AND LINK EQUAL 0,
    
```

```

2171 2777
2172 0400
2173 7563
2174 7401
2175 1702
2176 7510
2177 7520
    
```

PAGE

/ROUTINE TO LOAD THE TWO BUFFERS, (3400-4377) AND (5000-5777),
 /WITH SIMULATED RESULTS FROM MULTIPLICATION,
 /BUFFER 3400-4377 WILL CONTAIN THE EXPECTED RESULTS
 /FROM HARDWARE MULTIPLY FROM 3000-3377 STORED
 /IN THE FORM (MQ THEN AC), (LOW STORAGE)
 /BUFFER 5000-5777 WILL CONTAIN THE EXPECTED RESULTS
 /FROM HARDWARE MULTIPLY FROM 4400-4777
 /IN THE FORM (MQ THEN AC), (HIGH STORAGE)

```

2200 0000      MUYSIM, OPEN
2201 7431      SWAB
2202 7300      CLA CLL
2203 1170      TAD          [3000  /BEGINNING OPERAND=
2204 3217      DCA          MUYMB  /FOR MULTIPLY,
2205 1147      TAD          [3400  /BEGINNING ADDRESS OF=
2206 3221      DCA          STONUM /BUFFER TO HOLD SIMULATED RESULTS,
2207 7344      CLA CLL CMA RAL /SET A COUNT=
2208 3040      DCA          BOTH  /OF 2,
2209 1146      TAD          [400  /SET UP FOR=
2210 3041      DCA          COUNT  /A COUNT OF 400,
2211 7300      MUYLUP, CLA CLL /CLEAR AC ON RETURN
2212 1217      TAD          MUYMB /
2213 7140      CLL CMA
2214 4324      JMS          MQLMUY /SIMULATE MQL MUY
2215 0000      MUYMB, OPEN /OPERAND
2216 7445      DST          /STORE RESULTS FROM SIMULATED MULTIPLY;
2217 0000      STONUM, OPEN /CONTAINS ADDRESS WHERE MQ WILL BE STORED,
2218 2217      ISZ          MUYMB  /#1 TO OPERAND IN MUYMB,
2219 2221      ISZ          STONUM /
2220 2221      ISZ          STONUM /#2 TO STORAGE ADDRESS;
2221 2041      ISZ          COUNT  /DONE 400(OCTAL) MULTIPLIES?
2222 5213      JMP          MUYLUP /NO, THEN CONTINUE TO DO MORE,
2223 2040      ISZ          BOTH  /DONE WITH HIGH STORAGE?
2224 7610      SKP CLA /NO, LOW WAS JUST COMPLETED, NOW START HIGH,
2225 5600      JMP I      MUYSIM /YES, ALL SIMULATION IS COMPLETE,
2226 1122      TAD          [4400 /SET UP FOR BEGINNING OPERAND OF 4400;
    
```

```

2233 3217      DCA          MUYMB /
2234 1136      TAD          [5000 /SET UP FOR STORAGE TO BEGIN AT=
2235 3221      DCA          STONUM /ADDRESS 5000
2236 5211      JMP          MUYLUP-2 /NOW GO DO THE "HIGH" SIMULATION,

2237 0000      SUDOHQ, OPEN /TEMP
2240 0000      MQLDVI, OPEN /TEMP
2241 0000      PDSLSR, OPEN /TEMP
    
```

/ROUTINE TO LOAD THE TWO BUFFERS, (3400-4377) AND (5000-5777),
 /WITH SIMULATED RESULTS FROM DIVISION,
 /BUFFER 3400-4377 WILL CONTAIN THE EXPECTED RESULTS
 /FROM HARDWARE DIVIDE FROM 3000-3377 STORED
 /IN THE FORM (MQ THEN AC), (LOW STORAGE)
 /BUFFER 5000-5777 WILL CONTAIN THE EXPECTED RESULTS
 /FROM HARDWARE DIVIDE FROM 4400-4777
 /IN THE FORM (MQ THEN AC), (HIGH STORAGE),

```

2242 0000      DIVSIM, OPEN
2243 7431      SWAB
2244 7300      CLA CLL
2245 1121      TAD          [4777  /LET'S MAKE SURE WE'RE IN B MODE;
2246 3264      DCA          DIVMB  /CLEAR AC AND LINK;
2247 1147      TAD          [3400  /BEGINNING DIVISOR=
2248 3266      DCA          DIVNUM /FOR DIVIDE
2249 7344      CLA CLL CMA RAL /BEGINNING ADDRESS OF BUFFER
2250 3040      DCA          BOTH  /TO HOLD SIMULATED RESULTS,
2251 1146      TAD          [400  /SET A COUNT
2252 3041      DCA          COUNT  /OF 2,
2253 7300      DIVLUP, CLA CLL /SET UP FOR=
2254 1264      TAD          DIVMB  /A COUNT OF 400 OCTAL;
2255 7041      DCA IAC       /OBTAIN THE DIVISOR AND=
2256 3237      DCA          SUDOHQ /SUB 1 FROM IT AND THEN=
2257 1237      TAD          SUDOHQ /CHANGE IT TO ITS 2'S COMPLEMENT,
2258 7150      CLL CMA RAR  /STORE IT AS MQ,
2259 4777      JMS          PSDDVI /GET SAVED MQ DATA AND
2260 0000      DIVME, OPEN /#1'S COMPLEMENT IT FOR HIGH DIVIDEND
2261 7445      DST          /SIMULATE A DIVIDE;
2262 0000      DIVNUM, OPEN /DIVISOR,
2263 7340      CLA CLL CMA  /SAVE RESULTS FROM SIMULATED DIVIDE,
2264 1264      TAD          DIVMB  /CONTAINS ADDRESS WHERE MQ WILL BE STORED
2265 3264      DCA          DIVMB
2266 2266      ISZ          DIVNUM
2267 2266      ISZ          DIVNUM /#2 TO STORAGE ADDRESS;
2268 2041      ISZ          COUNT  /DONE 400 OCTAL DIVIDES?
2269 5255      JMP          DIVLUP /NO, DO MORE,
2270 2040      ISZ          BOTH  /DONE WITH HIGH STORAGE?
2271 7610      SKP CLA /NO, LOW WAS JUST COMPLETED, NOW START HIGH,
2272 5642      JMP I      DIVSIM /YES, ALL SIMULATION IS COMPLETE
2273 1120      TAD          [3377 /SET UP FOR DIVISOR TO BEGIN AT 4400
2274 3264      DCA          DIVMB /
2275 1136      TAD          [5000 /SET UP FOR STORAGE TO BEGIN AT =
2276 3266      DCA          DIVNUM /ADDRESS 5000,
2277 5255      JMP          DIVLUP-2 /NOW GO DO HIGH SIMULATION,
    
```

/ROUTINE TO SIMULATE A MULTIPLY (MQL MUY)
 /ENTER WITH NUMBER IN AC TO BE MULTIPLIED
 /BY OPERAND FOLLOWING CALL,

```

2306 0000 MQLMUY, OPEN
2307 3237 DCA SUDOMQ
2310 1306 TAD MQLMUY
2311 3313 DCA PSDMUY
2312 5314 JMP PSDMUY+1
2313 0000 PSDMUY, OPEN
2314 3240 DCA MQLDVI
2315 1376 TAD (7763
2316 3343 DCA PSDLSR
2317 1713 TAD I PSDMUY
2320 3344 DCA PSDMQL
2321 2313 ISE PSDMUY
2322 5332 JMP ,+10
2323 1240 TAD MQLDVI
2324 7420 SNL
2325 5330 JMP ,+3
2326 7100 CLL
2327 1344 TAD PSDMQL
2330 7010 RAR
2331 3240 DCA MQLDVI
2332 1237 TAD SUDOMQ
2333 7010 RAR
2334 3237 DCA SUDOMQ
2335 2343 ISE PSDLSR
2336 5323 JMP ,+13
2337 1237 TAD SUDOMQ
2340 7421 MQL /LOW ORDER PRODUCT IN MQ
2341 1240 TAD MQLDVI /HIGH ORDER PRODUCT IN AC,
2342 5713 JMP I PSDMUY /EXIT,

2343 0000 PSDLSR, OPEN
2344 0000 PSDMQL, OPEN
  
```

/ROUTINE TO SET UP FOR PRINTING IF AND OF,

```

2345 0000 PFIELD, OPEN
2346 1032 TAD DFFLD
2347 7110 RAR CLL
2350 7012 RTR
2351 1127 TAD (60
2352 3775' DCA PDFFLD
2353 1031 TAD IFFLD
2354 7110 RAR CLL
2355 7012 RTR
2356 1127 TAD (60
2357 3774' DCA PIFFLD
  
```

```

2360 5745 JMP I PFIELD /EXIT, AC=0,

2374 7114
2375 7117
2376 7763
2377 2400 PAGE
  
```

/ROUTINE TO SIMULATE A DIVIDE (DIV)
 /ENTER WITH HIGH ORDER DIVIDEND IN THE AC,
 /AND THE DIVISOR FOLLOWING THE CALL,

```

2400 0000 PSDDVI, OPEN
2401 3236 DCA PSDCAM /SAVE HIGH ORDER DIVIDEND
2402 1677 TAD I PSDDVI /DIVISOR
2403 2200 ISE PSDDVI /EXIT POINT,
2404 7141 CLL CMA IAC
2405 3777' DCA MQLMUY /2'S COMPLEMENT OF DIVISOR
2406 1236 TAD PSDCAM /HIGH ORDER DIVIDEND
2407 1777' TAD MQLMUY
2410 7630 SZL CLA
2411 7402 HLT /A DIVIDE OVERFLOW OCCURRED, PROCESSOR "DOWN",
2412 1376 TAD (7763 /7763 TO *
2413 3775' DCA PSDLSR /A COUNTER,
2414 5225 JMP ,+11
2415 1236 TAD PSDCAM
2416 7004 RAL
2417 3236 DCA PSDCAM
2420 1236 TAD PSDCAM
2421 1777' TAD MQLMUY
2422 7430 SZL
2423 3236 DCA PSDCAM
2424 7200 CLA
2425 1774' TAD SUDOMQ
2426 7004 RAL
2427 3774' DCA SUDOMQ
2430 2775' ISE PSDLSR
2431 5215 JMP ,+14
2432 1774' TAD SUDOMQ
2433 7421 MQL /MQ=QUOTIENT,
2434 1236 TAD PSDCAM /AC=REMAINDER
2435 5600 JMP I PSDDVI /EXIT,

2436 0000 PSDCAM, OPEN
  
```

/THIS ROUTINE ZERO'S ADDRESSES 3000-5777 IN EVERY FIELD

```

2437 0000 LZERO, OPEN
2440 7300 CLA CLL
2441 1032 TAD DFFLD
  
```

```

2442 3033 DCA TEMP
2443 3034 DCA OFFLD
2444 1076 TAD WIFLD
2445 7112 RTR CLL
2446 7010 PAR
2447 7040 CMA
2450 3034 DCA TEMP1
2451 1136 LZERO1, TAD E=3000
2452 3041 DCA COUNT
2453 1155 TAD C0777
2454 3016 DCA A116
2455 4507 JMS I CSETDF
2456 3416 DCA I A116
2457 2041 ISZ COUNT
2460 5256 JMP ,=2
2461 2034 ISZ TEMP1
2462 7610 SKP CLA
2463 5270 JMP ,=5
2464 1032 TAD OFFLD
2465 1134 TAD C10
2466 3032 DCA OFFLD
2467 5251 JMP LZERO1
2470 1033 TAD TEMP
2471 3032 DCA OFFLD
2472 4572 JMS I CRSETDF
2473 5637 JMP I LZERO
    
```

/ROUTINE INVOLVED IN AUTO INDEX TEST
 /TO MAKE SURE THE AUTO-INDEX REGISTER
 /WAS TRULY INDEXED.

```

2474 0000 CHECAI, OPEN
2475 3044 DCA SAVEAC /SAVE THE AC,
2476 1010 TAD A110 /GET CONTENTS OF A110=
2477 3045 DCA SAVEAI /AND SAVE IT,
2500 1044 TAD SAVEAC /RESTORE THE AC,
2501 4006 JMS INSTRU /EXECUTE THE INSTRUCTION
2502 3046 DCA SAVAC /SAVE THE AC
2503 7010 RAR /MOVE THE LINK IN A POSITION
2504 3047 DCA SAVLNK /WHERE IT CAN BE SAVED,
2505 1045 TAD SAVEAI /NOW CHECK TO SEE
2506 7040 CMA /IF THE AUTO-INDEX REGISTER
2507 1010 TAD A110 /WAS INDEXED,
2510 7640 SEA CLA /WAS IT?
2511 5317 JMP ,+6 /NO,
2512 2274 ISZ CHECAI /SET UP FOR EXIT,
2513 1047 TAD SAVLNK /GET SAVED LINK,
2514 7104 RAL CLL /RESTORE IT,
2515 1046 TAD SAVAC /RESTORE THE AC,
2516 5674 JMP I CHECAI /EXIT,
2517 4572 JMS I CRSETDF /RESET THE DF,
2520 4517 JMS I CRFIELD /CONVERT FIELD TO PRINTABLE ASCII,
2521 1674 TAD I CHECAI /GET TEST IDENTIFER,
2522 3341 DCA P1 /SAVE IT TO PRINT,
    
```

```

2523 1045 TAD SAVEAI /GET WHAT THE AI WAS BEFORE,
2524 7001 TAC /ADD ONE SO WE KNOW WHAT IT SHOULD BE,
2525 3034 DCA TEMP1 /SAVE FOR A CONVERSION,
2526 4542 JMS I CCPO /CONVERT SHOULD BE AI TO ASCII
2527 1034 TEMP1
2530 7200 A15B
2531 1010 TAD A110 /GET CONTENTS OF A110 AND SAVE
2532 3034 DCA TEMP1 /SO A110 DOESN'T GET AI WHEN ACCESSED,
2533 4542 JMS I CCPO /CONVERT WAS TO ASCII
2534 7034 TEMP1
2535 7163 WD4
2536 4575 TYPEX /TYPE ERROR DATA
2537 0011 DECIMAL 9
2540 5613 OCTAL
2541 7000 P1, CRLF2 /CR+2LF,
2542 6621 OPEN /TEST IDENTIFER,
2543 6613 TSTERR /TEST ERROR
2544 6637 CRLF2 /CR+2LF,
2545 7114 IF /"IF",
2546 6643 PIFFLD /NUMBER OF FIELD,
2547 7117 DF /"DF",
2550 7167 PDFFLD /NUMBER OF FIELD,
2551 4576 MA15B
2552 0011 SPACEX
2553 4577 DECIMAL 9
2554 7163 OCTAL
2555 7402 TYPE
2556 4567 WD4
2557 7300 HLT /EAE INSTRUCTION FAILED TO INCREMENT
2560 1045 /THE AUTO-INDEX REGISTER,
2561 3010 /PRESS CONTINUE TO ENTER
2562 1044 /A SCOPE LOOP, SWITCH
2563 4006 /REGISTER SETTINGS DO NOT
2564 5357 JMP ,=5 /APPLY TO THIS ERROR,
    /SET DF,
    
```

/BUFFER STARTS AT 3000 AND IS 3000 WORDS LONG

```

2574 2237
2575 2343
2576 7763
2577 2306
3000 *3000
    
```

XLIST

6000 *6000

/ROUTINE TO PRINT INITIAL PART OF DAD, DST,
/ERROR DATA.

```

6000 0000 INDDMD, OPEN
6001 0033 DCA TEMP /SAVE HQ ADDRESS;
6002 4542 JMS I [CPO /CONVERT TO ASCII;
6003 0033 TEMP /
6004 6647 ACADDR /ASCII FOR HQ ADDRESS;
6005 2033 ISZ TEMP /+1 TO TEMP FOR AC ADDRESS;
6006 4542 JMS I [CPO /CONVERT TO ASCII;
6007 0033 TEMP
6008 7123 ADDRAC /ASCII FOR AC ADDRESS;
6009 4917 JMS I [PFIELD /CONVERT FIELD TO ASCII;
6010 1000 TAO I INDDMD /GET TEST IDENTIFIER
6011 3220 DCA ,+5 /AND SAVE TO BE PRINTED
6012 2200 ISZ INDDMD /SET FOR EXIT;
6013 4575 TYPEX /TYPE THE FOLLOWING;
6014 0015 DECIMAL
6015 0013 11 /
6016 6613 OCTAL
6017 0000 CRLF2 /CR+2LF
6018 6621 OPEN /DAD, DST;
6019 6621 TSTERR /TEST ERROR;
6020 6613 CRLF2 /CR+2LF
6021 6637 IF /IF
6022 7114 PIFFLD /IF NUMBER
6023 6643 DF /DF
6024 7117 PDFFLD /DF NUMBER
6025 6630 HQADDR /HQ ADDR
6026 6647 ACADDR /HQ ADDRESS AND AC ADDR
6027 7123 ADDRAC /AC ADDRESS, CR+2LF, TERMINATE;
6028 5600 JMP I INDDMD /EXIT, AC=L=0;
    
```

/ROUTINE TO PRINT 2ND PART OF DAD [ERROR INFORMATION,

```

6033 0000 DADP2, OPEN
6034 0033 DCA TEMP
6035 4576 SPACEX /SPACE THIS=
6036 0015 DECIMAL /AMOUNT OF TIMES;
6037 4567 OCTAL
6038 1433 JMS I [SETDF
6039 1433 TAO I TEMP
6040 7041 CMA IAC
6041 3034 DCA TEMP1
6042 2033 ISZ TEMP
6043 0033
    
```

```

6044 1433 TAO I TEMP
6045 7040 CMA
6046 3035 DCA TEMP2
6047 4572 JMS I [RSETDF
6048 4542 JMS I [CPO
6049 0034 TEMP1
6050 7135 PMQ
6051 4542 JMS I [CPO
6052 0035 TEMP2
6053 7130 PAC
6054 4575 TYPEX
6055 0005 5
6056 6700 LACMQ
6057 6663 ORGINL
6058 6675 LINK0
6059 7130 PAC
6060 7135 PMQ
6061 1034 TAO TEMP1
6062 7041 CMA IAC
6063 3034 DCA TEMP1
6064 1035 TAO TEMP2
6065 7040 CMA
6066 3035 DCA TEMP2
6067 4542 JMS I [CPO
6068 0034 TEMP1
6069 7135 PMQ
6070 4542 JMS I [CPO
6071 0035 TEMP2
6072 7130 PAC
6073 4942 JMS I [CPO
6074 0034 TEMP1
6075 7135 PMQ
6076 4542 JMS I [CPO
6077 0035 TEMP2
6078 7130 PAC
6079 4575 TYPEX
6080 0010 10
6081 6710 TBA
6082 7130 PAC
6083 7135 PMQ
6084 6723 SIMUL
6085 6741 LINK1
6086 6734 REG0
6087 6734 REG0
6088 6744 ACTUAL
6089 1020 TAO LKTOCK
6090 4516 JMS I [BIT1
6091 4542 JMS I [CPO
6092 0022 ACTOOCK
6093 7130 PAC
6094 4942 JMS I [CPO
6095 0021 NQTOCK
6096 7135 PMQ
6097 4575 TYPEX
6098 1002 2
6099 7130 PAC
6100 7135 PMQ
6101 5633 JMP I DADP2 /EXIT
    
```

6200 PAGE
 /ROUTINE TO PRINT SECOND PART OF DDT ERROR INFORMATION;
 6200 0000 DSTP2, OPEN
 6201 3033 OCA TEMP
 6202 4577 TYPE
 6203 7035 REG
 6204 4576 SPACEX
 6205 0010 DECIMAL
 8
 6206 4575 OCTAL
 TYPEX
 6207 0007 DECIMAL
 7
 6210 7041 OCTAL
 BEFORE
 6211 6771 DSTXTY
 6212 6616 SPAG2
 6213 7046 AFTER
 6214 6771 DSTXTY
 6215 6610 CALF
 6216 7053 CL
 6217 4576 SPACEX
 DECIMAL
 10
 6220 0012 OCTAL
 6221 4577 TYPE
 6222 6675 LINK0
 6223 4576 SPACEX
 DECIMAL
 10
 6224 0012 OCTAL
 TAD LKTOCK
 6226 4516 JMS I CBIT1
 6227 4577 TYPE
 6230 7060 CMQ
 6231 4576 SPACEX
 DECIMAL
 9
 6232 0011 OCTAL
 JMS I CGPO
 6234 0033 TEMP
 6239 7130 PAC
 6236 4542 JMS I CGPO
 6237 0021 MQTOCK
 6240 7135 PHQ
 6241 4577 TYPE
 6242 7130 PAC
 6243 4576 SPACEX
 DECIMAL
 7
 6244 0007 OCTAL
 6245 4575 TYPEX

6246 0002 DECIMAL
 2
 6247 7135 OCTAL
 PHQ
 6250 7066 CMQA
 6251 4576 SPACEX
 DECIMAL
 17
 6252 0021 OCTAL
 JMS I CSETDF
 6254 1433 TAD I TEMP
 6255 3035 OCA TEMP2
 6256 4572 JMS I CRSETDF
 6257 4542 JMS I CGPO
 6260 0035 TEMP2
 6261 7135 PHQ
 6262 4575 TYPEX
 DECIMAL
 2
 6263 0002 OCTAL
 6264 7135 PHQ
 6265 7076 CAC
 6266 4576 SPACEX
 DECIMAL
 9
 6267 0011 OCTAL
 6270 2033 ISB TEMP
 6271 4542 JMS I CGPO
 6272 0033 TEMP
 6273 7135 PHQ
 6274 4542 JMS I CGPO
 6275 0022 ACTOCK
 6276 7130 PAC
 6277 4577 TYPE
 6300 7135 PHQ
 6301 4576 SPACEX
 DECIMAL
 7
 6302 0007 OCTAL
 TYPEX
 6303 4575 DECIMAL
 2
 6304 0002 OCTAL
 PAC
 6305 7130 CACA
 6306 7104 SPACEX
 6307 4576 DECIMAL
 17
 6310 0021 OCTAL
 JMS I CSETDF
 6311 4567 TAD I TEMP
 6312 1433 OCA TEMP1
 6313 3034 JMS I CRSETDF
 6314 4572 JMS I CGPO
 6315 4542 JMS I CGPO
 6316 0034 TEMP1

6417 7130 PAC
 6420 4577 TYPE
 6421 7132 PAC
 6422 4576 JMP I DSTP2
 6410 PAGE

/ROUTINE TO PRINT ERROR DATA FOR MUY OR DIV TEST;

6420 7030 MDP2, OPEN
 6421 3033 DCA TEMP
 6422 1600 TAD I MDP2
 6423 2200 ISE MDP2
 6424 3217 DCA MDP2A
 6425 4517 JMS I CFFIELD /CONVERT FIELD TO ASCII
 6426 1600 TAD I MDP2
 6427 2200 ISE MDP2
 6410 3226 DCA MDP2B
 6411 4542 JMS I CCPO
 6412 0033 TEMP
 6413 7135 PMQ
 6414 4575 TYPEX
 DECIMAL
 6415 0014 12
 OCTAL
 6416 6613 CRLF2
 6417 0000 MDP2A, OPEN
 6420 6621 TSTERR
 6421 6613 CRLF2
 6422 6637 IF
 6423 7114 PIFFLD
 6424 6643 DF
 6425 7117 PDIFFLD
 6426 0000 MDP2B, OPEN
 6427 7302 ADDR
 6430 7135 PMQ
 6431 6610 CRLF
 6432 4576 SPACEX
 DECIMAL
 6433 7015 13
 OCTAL
 6434 1600 TAD I MDP2
 6435 3291 DCA MDP2C
 6436 2200 ISE MDP2
 6437 4547 JMS I CSETDF
 6440 1433 TAD I TEMP
 6441 3033 DCA TEMP
 6442 4572 JMS I CRSETDF
 6443 4542 JMS I CCPO
 6444 0033 TEMP
 6445 7140 WD1
 6446 4575 TYPEX
 DECIMAL

6447 0010 8
 OCTAL
 6450 7236 MDP2C, MDHED1
 6451 0000 OPEN
 6452 7252 MDHED2
 6453 6663 ORGINL
 6454 6675 LINK0
 6455 7146 WD1
 6456 6723 SIMUL
 6457 6675 LINK0
 6460 4976 SPACEX
 DECIMAL
 6461 0006 6
 OCTAL
 6462 1600 TAD I MDP2
 6463 2200 ISE MDP2
 6464 3034 DCA TEMP1
 6465 1434 TAD I TEMP1
 6466 3034 DCA TEMP1
 6467 1434 TAD I TEMP1
 6470 3035 DCA TEMP2
 6471 2034 ISE TEMP1
 6472 1434 TAD I TEMP1
 6473 3036 DCA TEMP3
 6474 1600 TAD I MDP2
 6475 3310 DCA MDP2D
 6476 2200 ISE MDP2
 6477 4542 JMS I CCPO
 6500 0035 TEMP2
 6501 7156 WD3
 6502 4542 JMS I CCPO
 6503 0036 TEMP3
 6504 7152 WD2
 6505 4575 TYPEX
 DECIMAL
 6506 0003 3
 OCTAL
 6507 7192 MDP2D, WD2
 6510 0000 OPEN
 6511 6744 ACTUAL
 6512 1020 TAD LKTOCK
 6513 4316 JMS I EB1T1
 6514 4576 SPACEX
 DECIMAL
 6515 0006 6
 OCTAL
 6516 4542 JMS I CCPO
 6517 0022 ACTOCK
 6520 7152 WD2
 6521 4542 JMS I CCPO
 6522 0021 MTOCK
 6523 7156 WD3
 6524 4542 JMS I CCPO
 6525 0023 SCTOCK
 6526 7163 WD4

6527 4575 TYPEX
 6530 7002 DECIMAL
 6531 7192 2
 6532 7163 OCTAL
 6533 9602 WQ2
 WQ4
 JMP I MDP2

6600 0227 PAGE
 6601 0400 SWTAB, DADTST
 6602 0600 DSTTST
 6603 1021 MUYTST
 6604 0307 DIVTST
 6605 0501 DADAIT
 6606 0706 DSTAIT
 6607 1200 MUYAIT
 DIVAIT

/TEXT FOR ERROR PRINTOUTS:

6610 4543 CRLF, TEXT 'X#0?'
 6611 0077
 6612 0000
 6613 4543 CRLF2, TEXT 'X##0?' /CR+2LF,
 6614 4300
 6615 7700
 6616 4040 SPAC2, TEXT ' 0?'
 6617 0077
 6620 0000
 6621 4024 TSTERR, TEXT ' TEST ERROR0?'
 6622 0523
 6623 2440
 6624 0522
 6625 2217
 6626 2200
 6627 7700
 6630 4015 HQADDR, TEXT ' HQ ADDR 0?'
 6631 2140
 6632 2104
 6633 0422
 6634 4040
 6635 0077
 6636 0000
 6637 4011 IF, TEXT ' IF 0?'
 6640 0640
 6641 0077
 6642 0000
 6643 4040 DF, TEXT ' DF 0?'
 6644 0406
 6645 4000

6646 7700
 6647 3030 ACADDR, TEXT 'XXXXX# AC ADDR 0?'
 6650 3030
 6651 4543
 6652 4001
 6653 0340
 6654 2104
 6655 0422
 6656 4040
 6657 0077
 6660 0000
 6661 4000 SPAQ1, TEXT ' 0?'
 6662 7700
 6663 4543 ORIGINL, TEXT 'X## ORIGINAL 0?'
 6664 4340
 6665 1722
 6666 1107
 6667 1116
 6670 0114
 6671 4040
 6672 4040
 6673 0077
 6674 0000
 6675 6040 LINK0, TEXT '0 0?'
 6676 4000
 6677 7700
 6700 1440 LACM0, TEXT 'L AC HQ0?'
 6701 4040
 6702 0103
 6703 4040
 6704 4040
 6705 1521
 6706 0077
 6707 0000
 6710 4543 TBA, TEXT 'X# TO BE ADDED 0?'
 6711 4024
 6712 1740
 6713 0205
 6714 4001
 6715 0404
 6716 0504
 6717 4040
 6720 4040
 6721 0077
 6722 0000
 6723 4543 SIMUL, TEXT 'X# SIMULATED 0?'
 6724 4023
 6725 1115
 6726 2514
 6727 1124
 6730 0504
 6731 4040
 6732 4000
 6733 7700
 6734 6060 REG0, TEXT '0000 0?'

6735	0000		
6736	4040		
6737	0077		
6740	0000		
6741	4140	LINK1, TEXT	'1 0?'
6742	4000		
6743	7700		
6744	4543	ACTUAL, TEXT	'%# ACTUAL 0?'
6745	4001		
6746	7324		
6747	0901		
6750	1440		
6751	4040		
6752	4040		
6753	4000		
6754	7700		
6755	0401	DADTXT, TEXT	'DAD0?'
6756	0400		
6757	7700		
6760	0401	DADATX, TEXT	'DAD AUTO INDEX0?'
6761	0440		
6762	0125		
6763	2417		
6764	4011		
6765	1604		
6766	0530		
6767	0077		
6770	0000		
6771	0403	DSTTXT, TEXT	'DST0?'
6772	2400		
6773	7700		
6774	0423	DSTATX, TEXT	'DST AUTO INDEX0?'
6775	2440		
6776	0125		
6777	2417		
7000	4011		
7001	1604		
7002	0530		
7003	0077		
7004	0000		
7005	1925	MUYTXT, TEXT	'MUY0?'
7006	3100		
7007	7700		
7010	1925	MUYATX, TEXT	'MUY AUTO INDEX0?'
7011	3140		
7012	0125		
7013	2417		
7014	4011		
7015	1604		
7016	0530		
7017	0077		
7020	0000		
7021	0411	DIVTXT, TEXT	'DIV0?'
7022	2600		
7023	7700		

7024	0411	DIVATX, TEXT	'DIV AUTO INDEX0?'
7025	2640		
7026	0125		
7027	2417		
7030	4011		
7031	1604		
7032	0530		
7033	0077		
7034	0000		
7035	4022	REG, TEXT	'REG0?'
7036	0507		
7037	0077		
7040	0000		
7041	0205	BEFORE, TEXT	'BEFORE 0?'
7042	0617		
7043	2205		
7044	4000		
7045	7700		
7046	0106	AFTER, TEXT	'AFTER 0?'
7047	2405		
7050	2240		
7051	0077		
7052	0000		
7053	4543	CL, TEXT	'%# C(L)0?'
7054	4003		
7055	5014		
7056	5100		
7057	7700		
7060	4543	CMQ, TEXT	'%# C(MQ)0?'
7061	4003		
7062	5015		
7063	2151		
7064	0077		
7065	0000		
7066	4543	CMQA, TEXT	'%# C(MQ ADDR)0?'
7067	4003		
7070	5015		
7071	2140		
7072	0104		
7073	0422		
7074	5100		
7075	7700		
7076	4543	CAC, TEXT	'%# C(AC)0?'
7077	4003		
7100	5001		
7101	0351		
7102	0077		
7103	0000		
7104	4543	CACA, TEXT	'%# C(AC ADDR)0?'
7105	4003		
7106	5001		
7107	0340		
7110	0104		
7111	0422		
7112	5100		

7113	7700		
7114	3030	PIFFLD, TEXT	'XX0?'
7115	0077		
7116	0000		
7117	3030	POFFLD, TEXT	'XXX##0?'
7120	4543		
7121	4300		
7122	7700		
7123	3030	ADDRAC, TEXT	'XXXX##0?'
7124	3030		
7125	4543		
7126	4300		
7127	7700		
7130	3030	PAC, TEXT	'XXXX 0?'
7131	3030		
7132	4040		
7133	0077		
7134	0000		
7135	3030	PMQ, TEXT	'XXXX 0?'
7136	3030		
7137	4040		
7140	0077		
7141	0000		
7142	3030	BITX, TEXT	'Xx 0?'
7143	4040		
7144	0077		
7145	0000		
7146	3030	WD1, TEXT	'XXXX '
7147	3030		
7150	4040		
7151	0000		
7152	3030	WD2, TEXT	'XXXX '
7153	3030		
7154	4040		
7155	0000		
7156	3030	WD3, TEXT	'XXXX 0?'
7157	3030		
7160	4040		
7161	0077		
7162	0000		
7163	3030	WD4, TEXT	'XXXX0?'
7164	3030		
7165	0077		
7166	0000		
7167	4001	MAISB, TEXT	' A110 SHOULD BE '
7170	1161		
7171	6040		
7172	2310		
7173	1725		
7174	1404		
7175	4002		
7176	4540		
7177	4000		
7200	3030	AISB, TEXT	'XXXX#'
7201	3030		

7202	4543		
7203	0000		
7204	4001	MAIS, TEXT	' A110 IS0?'
7205	1161		
7206	6040		
7207	1173		
7210	0077		
7211	0000		
7212	4543	KEBEME, TEXT	'#KEB EME0?'
7213	1305		
7214	7040		
7215	0515		
7216	0500		
7217	7700		
7220	4543	HF, TEXT	'*##HIGHEST FIELD (0=7)? 0?'
7221	4310		
7222	1107		
7223	1005		
7224	2324		
7225	4006		
7226	1105		
7227	1404		
7230	4050		
7231	6055		
7232	6751		
7233	7740		
7234	4000		
7235	7700		
7236	4314	MDHED1, TEXT	'#L 0?'
7237	4040		
7240	0077		
7241	0000		
7242	1524	MA, TEXT	'MTPLO?'
7243	2014		
7244	0077		
7245	0000		
7246	0411	DA, TEXT	'DIVS0?'
7247	2623		
7250	0077		
7251	0000		
7252	4040	MDHED2, TEXT	' AC HQ SC0?'
7253	4001		
7254	340		
7255	4040		
7256	4015		
7257	2140		
7260	4040		
7261	4023		
7262	1300		
7263	7700		
7264	4015	MULTIP, TEXT	' MULTIPLICAND0?'
7265	2514		
7266	0411		
7267	2014		
7270	1103		

7271	0116		
7272	0400		
7273	7700		
7274	4004	DIVSOR, TEXT	'DIVISOR01'
7275	1126		
7276	1123		
7277	1722		
7300	0077		
7301	0000		
7302	4001	ADDR, TEXT	'ADDR 01'
7303	0404		
7304	2240		
7305	4000		
7306	7700		
7307	0000	SC14, TEXT	'001401'
7310	6164		
7311	0077		
7312	0000		
7313	0000	SC15, TEXT	'001501'
7314	6165		
7315	0077		
7316	0000		

5

0116	1725
0117	2345
0120	3377
0121	4777
0122	4400
0123	2437
0124	0207
0125	6600
0126	2074
0127	0060
0130	0077
0131	6202
0132	0200
0133	6201
0134	0010
0135	7407
0136	5000
0137	4377
0140	7425
0141	1021
0142	1525
0143	6402
0144	6401
0145	1400
0146	7400
0147	3400
0150	0706
0151	7445
0152	0600
0153	0501
0154	2127

0155	2777
0156	0000
0157	5001
0160	1000
0161	2064
0162	2002
0163	2046
0164	2040
0165	2055
0166	1700
0167	2111
0170	3000
0171	2145
0172	2120
0173	0007
0174	1752
0175	1735
0176	1715
0177	1600

ACADDR	8647	DADALP	0344	END	1400	MULTIP	7264
ACL	7701	DADATX	6760	END1	1467	MUY	7435
ACS	7403	DADERP	0300	END2	1476	MUY2	3617
ACTOCK	0022	DADHLT	0256	ENDTYP	2135	MUYA0	0726
ACTUAL	6744	DADLOP	0257	EQUAL	7000	MUYADR	0623
ADDR	7332	DADP2	6033	FLAG	1670	MUYAIE	1000
ADDNAC	7123	DADTST	0227	GTF	6004	MUYAIT	2706
AFTER	7046	DADTXY	6755	HF	7220	MUYALP	0747
AI10	0010	DCM	7575	HIFLD	0026	MUYATX	7010
AI11	0011	DE	7665	HLT	7402	MUYERP	0665
AI12	0012	DF	6643	IF	6637	MUYHLT	0637
AI13	0013	OFFLD	0032	IFFLD	0031	MUYLOP	0640
AI14	0014	OFFOCK	0024	IFTOCK	0025	MUYLUP	2213
AI15	0015	OIV	7407	INDDMO	6000	MUYMB	2217
AI16	0016	OIV0	1037	INPUT	4374	MUYSAI	0737
AI17	0017	OIVA0	1220	INSTRU	0006	MUYSIM	2200
AISB	7200	OIVAOR	1047	IOF	6002	MUYSR5	0630
AITEST	0007	OIVAIE	1274	ION	6001	MUYTST	0600
ASCT	1564	OIVAIT	1200	K0707	1566	MUYTXT	7005
ASKFLD	2000	OIVALP	1251	K6060	1567	NEXT	0042
ASR	7415	OIVATX	7024	K7700	1565	NMI	7411
BEFORE	7041	OIVERP	1114	KEBEME	7212	NOPE	7401
BEQDF	0030	OIVHLT	1066	LACMO	6700	NXTST	2064
BEQIF	0027	OIVLOP	1067	LAS	7604	OPEN	0000
BIT1	1725	OIVLUP	2255	LDCORE	2145	ORGINL	6663
BITX	7142	OIVMB	2264	LF212	1677	P1	2541
BOH	0040	OIVNUM	2266	LINK0	6675	P100	1676
BSW	7002	OIVSAI	1241	LINK1	6741	PAC	7130
CAC	7076	OIVSIM	2242	LKTOCK	0020	PDDFLD	7117
CACA	7104	OIVSOR	7274	LSR	7417	PDSLSR	2241
CAF	6007	OIVSRS	1057	LZERO	2437	PFIELD	2345
CAN	7621	OIVTST	1021	LEERO1	2451	PFLAG	1671
CDF	6201	OIVTXY	7021	M40	1675	PIFFLD	7114
CHECAI	2474	DL0	7663	M43	1674	PH0	7135
CL	7053	DPIC	7573	M45	1673	PRINT	1643
CLAE	7621	DPSZ	7451	M77	1672	PSDCAM	2436
CMO	7060	DST	7445	MA	7242	PSDDVI	2400
CMOA	7066	DST0	0407	MA1IS	7204	PSDLSR	2343
CMV	1547	DSTA0	0514	MAISB	7167	PSDHQL	2344
COMPLE	7140	DSTADR	0415	MDMED1	7236	PSDHUY	2313
COUNT	0041	DSTAER	0554	MDMED2	7292	RF	6214
CP0	1525	DSTAIE	0567	MDP2	6400	REG	7035
CR215	1700	DSTAIT	0501	MDP2A	6417	REG0	6734
CR21	6610	DSTALP	0561	MDP2B	6426	RIB	6234
CRLE2	6613	DSTATX	6774	MDP2C	6451	RIF	6224
CA	7246	DSTERR	0472	MDP2D	6510	RMF	6244
CAD	7443	DSTERR	0444	MQA	7501	RSETAI	2127
CAD0	0236	DSTHLT	0450	MQADDR	6630	RSETDF	2120
CADAC	0322	DSTLOP	0451	SQL	7421	RTF	6005
CADADR	0245	DSTP2	6200	SQLDVI	2240	SAM	7457
CADATE	0352	DSTTST	0400	SQLMUY	2306	SASC	1563
CADAIT	0307	DSTTXY	6771	SQLTOCK	0021	SAVAC	0046

SAVLAC	0044	WD4	7163
SAVEAI	0045	XINPUT	1752
SAVLNK	0047	XSPACE	1715
SAVREG	1760	XTYPE	1735
SG14	7307	ZERO	0043
SG15	7314		
SGA	7441		
SQL	7403		
SQLTOCK	0023		
SELIST	2074		
SETDF	2111		
SGT	6006		
SHL	7413		
SIMUL	6723		
SKON	6000		
SP240	1701		
SPAC1	6661		
SPAC2	6616		
SPACE1	1711		
SPACEX	4576		
SRT	6003		
START	0200		
STONUM	2221		
SUDOMQ	2237		
SW0	2040		
SW1	2046		
SW2	2055		
SWAB	7431		
SWBA	7447		
SWP	7521		
SXTAB	6600		
TBA	6710		
TEMP	0033		
TEMP1	0034		
TEMP2	0035		
TEMP3	0036		
TEMP4	0037		
TEMQ	1666		
TEMR	1667		
TS01	1606		
TS02	1615		
TSTERR	6621		
TTRUT	1702		
TYPAT	1626		
TYPE	4577		
TYPEX	4575		
YPSPP	1632		
LTYPE	1600		
WASC	1562		
W01	7146		
W02	7152		
W03	7156		

/KES-E EAE EXTENDED MEMORY EXERCISER HAINDEC=BB=DHKEA=A=L PAL10 V141 31=AUG*72 22149 PAGE 1=48

ERRORS DETECTED: 0
LINKS GENERATED: 39
RUN-TIME: 18 SECONDS
3K CORE USED

