

IDENTIFICATION

PRODUCT CODE: MAINDEC-28-DGDRP-A-D

PRODUCT NAME: DR8EA DIAGNOSTIC (FOR USE ON SYSTEMS THAT HAVE A TRADITIONAL PDP-8 PROCESSOR WITH A DMS/E OMNIBUS CONVERTER)

DATE REVISED: JULY 3, 1973

MAINTAINER: DIAGNOSTIC GROUP

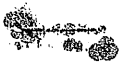
AUTHOR: MIKE HARE

COPYRIGHT (C) 1972, 1973, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A LICENSE FOR USE ON A SINGLE SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DEC'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.



1. ABSTRACT

THIS DIAGNOSTIC HAS BEEN MODIFIED FOR TESTING THE DR8=EA INTERFACE ON SYSTEMS THAT HAVE A TRADITIONAL PDP=8 PROCESSOR WITH A DW8/E BUS CONVERTER.

THIS PROGRAM IS A DIAGNOSTIC AND EXERCISER FOR THE DR8=EA 12 CHANNEL BUFFERED DIGITAL INTERFACE. ALL FUNCTIONS ARE TESTED AND ERRORS ARE REPORTED BY HALTS AND/OR ERROR TYPEOUTS.

2. REQUIREMENTS

2.1 EQUIPMENT

TRADITIONAL PDP=8 COMPUTER WITH 4K OF CORE
ASR=33 TELETYPE (OR EQUIVALENT)
DW8/E BUS CONVERTER
DR8=EA WITH TEST CABLE

2.2 STORAGE

THE PROGRAM USES LOCATIONS 0000=4121

3. LOADING PROCEDURE

THE STANDARD PROCEDURE FOR LOADING BINARY TAPES SHOULD BE USED.

4. STARTING PROCEDURE

4.1 STARTING ADDRESS

200=INPUT DEVICE CONFIGURATION
201=START WITH STANDARD CONFIGURATION

4.2 SWITCH SETTINGS

FOR EITHER STARTING ADDRESS, NORMAL SETTING IS SR0=SR11=0 (DOWN).

4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY
SET SWITCH REGISTER TO DESIRED STARTING ADDRESS
LOAD ADDRESS
CLEAR SWITCHES
PRESS CLEAR AND CONTINUE

4.3.1 FOR STARTING ADDRESS 200

THE PROGRAM WILL TYPE "SET SR FOR DEVICE CODE AND CONT"
AND THEN HALT.

SET SWITCHES TO 00X WHERE X IS AN OCTAL
NUMBER CORRESPONDING TO THE 3 LSB OF THE DEVICE SELECTOR CODE,
E.G. IF DEVICE CODE IS 53, SET SR TO 0003,
PRESS CONTINUE.

PROGRAM WILL RESPOND BY TYPING
"SET SR FOR INTERRUPT JUMPERS AND CONT" AND THEN HALT,
SET SWITCHES FOR ALL INPUT REGISTER BITS JUMPERED TO INTERRUPT,
PRESS CONTINUE.

PROGRAM WILL RESPOND BY TYPING
"SET SR FOR FLIPFLOP JUMPERS AND CONT" AND THEN HALT,
SET SWITCHES FOR ALL INPUT REGISTER FLIPFLOPS,
PRESS CONTINUE.

PROGRAM WILL RESPOND BY TYPING
"SET SR FOR RUN" AND THEN HALT,
SET SWITCHES AS DESIRED. (REFER TO SECTION 5.1)
PRESS CONTINUE.

PROGRAM WILL BEGIN TEST EXECUTION

4.3.2 FOR STARTING ADDRESS 201

SET SWITCHES AS DESIRED. (REFER TO SECTION 5.1)
DEPRESS CONTINUE.

PROGRAM WILL BEGIN TEST EXECUTION

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

SR0=1, SUPPRESS ERROR HALT
SR1=1, SUPPRESS ERROR TYPEOUT
SR2=1, LOOP ON CURRENT TEST
SR3=1, LOOP WITH CURRENT DATA
SR4=1, SUPPRESS BELL OR TYPEOUT AT END OF PASS
SR5=1, SUPPRESS ITERATIONS
SR6=1, ESCAPE TO NEXT TEST ON ERROR

5.2 PROGRAM AND/OR OPERATOR ACTION

5.2.1.1 WITH SWITCHES SET AS IN 4.2, THE PROGRAM WILL RUN ALL TESTS SEQUENTIALLY. EACH IOT TEST WILL BE REPEATED 4096 TIMES. EACH DATA TEST WILL BE REPEATED 50 TIMES. AFTER ALL TESTS HAVE BEEN COMPLETED, THE PROGRAM WILL TYPE "DR" AND START ALL TESTS AGAIN. IF AN ERROR OCCURS, THE PROGRAM WILL HALT AND TYPE AN APPROPRIATE ERROR MESSAGE (SEE SECTION 6 FOR DETAILS).

5.2.1.2 WITH SR0=1 (UP), PROGRAM ACTION WILL BE AS IN 5.2.1.1, EXCEPT NO TYPEOUT WILL OCCUR.

5.2.1.3 WITH SR2=1(UP) PROGRAM ACTION WILL BE AS IN 5.2.1.1. EXCEPT NO TYPEOUT WILL OCCUR. THE ADDRESS OF THE FAILING TEST WILL BE DISPLAYED IN THE COMPUTER AC.

5.2.1.4 WITH SR4=1(UP), PROGRAM ACTION WILL BE AS IN 5.2.1.1 EXCEPT NO END OF PASS TYPEOUT WILL OCCUR.

5.2.1.5 WITH SR5=1 (UP), EACH TEST WILL BE EXECUTED ONLY ONCE. INSTEAD OF TYPING "DR", THE PROGRAM WILL RING THE TTY BELL.

5.2.1.6 WITH SR0=1 AND SR6=1, PROGRAM ACTION WILL BE AS IN 5.2.1.1 IF NO ERRORS OCCUR. IF AN ERROR OCCURS, THE PROGRAM WILL TYPE AN APPROPRIATE ERROR MESSAGE AND IMMEDIATELY TERMINATE ITERATIONS OF THE FAILING TEST. THE PROGRAM WILL THEN START THE NEXT TEST IN SEQUENCE.

6. ERRORS

6.1 NORMAL OPERATION

IF AN ERROR OCCURS WITH SWITCHES SET AS IN 4.2, THE PROGRAM WILL TYPE AN APPROPRIATE ERROR MESSAGE (WITH DATA IF APPLICABLE) AND HALT.

THE FORMAT OF THE ERROR TYPEOUT IS

XXXX MESSAGE
HEADER FOR DATA (IF APPLICABLE)
DATA (IF APPLICABLE)

WHERE XXXX= ADDRESS OF JMS TO ERROR ROUTINE IN TEST THAT FAILED.

6.2 ERROR RECOVERY

SET SR6=1(UP) TO ESCAPE TO NEXT TEST. PRESS CONTINUE.

6.3 ERROR LOOP (IOTS)

SET SR0=1 TO SUPPRESS HALT
SET SR1=1 TO SUPPRESS TYPEOUT
SET SR2=1 TO LOOP ON CURRENT FAILING TEST

6.4 ERROR LOOP (DATA)

SAME AS 6.3 EXCEPT USE SR3 INSTEAD OF SR2 TO LOOP WITH CURRENT DATA.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

TEST JUMPER CABLE MUST BE INSTALLED.
ANY FLOATING INPUTS TO INPUT REGISTER SHOULD BE GROUNDED, OR ERRORS MAY OCCUR.

7.2 OPERATING RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME

EXECUTION TIME IS APPROXIMATELY 9 MINUTES FOR FULL ITERATION AND APPROXIMATELY 10 SECONDS WITH ITERATIONS SUPPRESSED.

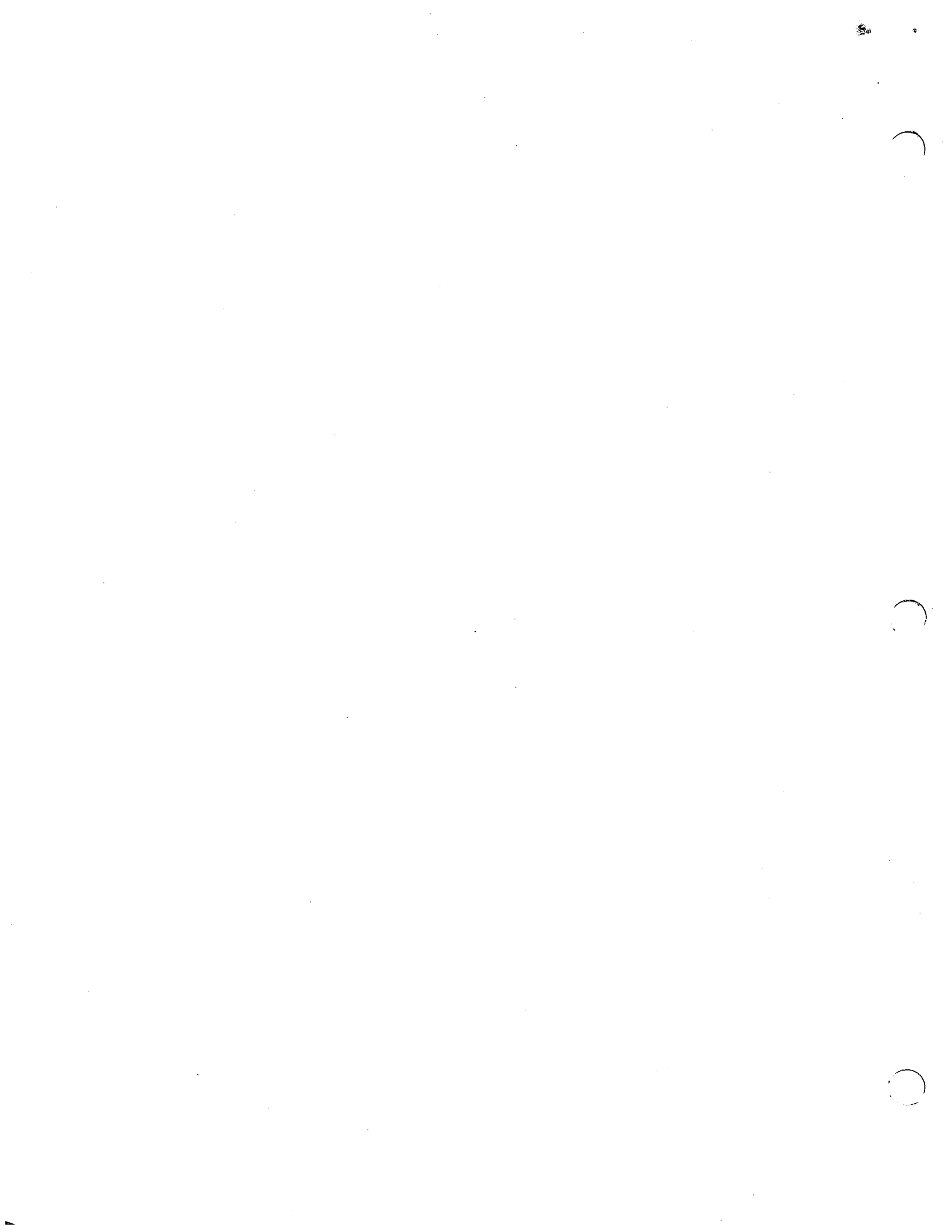
9. PROGRAM DESCRIPTION

THE DR8-EA IS A TEST OF ALL FUNCTIONS OF THE INTERFACE.

THE PROGRAM SEQUENCE IS AS FOLLOWS:

ALL BASIC I/O TESTS ARE EXECUTED 4096 TIMES,
ALL OUTPUT REGISTER FUNCTIONS ARE TESTED WITH BINARY COUNT PATTERNS,
ALL INPUT REGISTER FUNCTIONS ARE TESTED USING BINARY COUNT PATTERNS.
INTERACTION BETWEEN INPUT AND OUTPUT REGISTERS IS TESTED FOR
WITH BINARY COUNT PATTERNS.
ALL SKIPS AND INTERRUPTS ARE TESTED USING BINARY COUNT PATTERNS.

10. LISTING



0031 0000 LPCNT, 0

/INDIRECT POINTERS

0032 0266 XDBEI, DBEIX
 0033 0273 XDBSK, DBSKX
 0034 0300 XDBCI, DBCIX
 0035 0305 XDBRI, DBRIX
 0036 0312 XDBCO, DBCOX
 0037 0317 XDBSO, DBSOX
 0040 0324 XDBRO, DBROX
 0041 3200 XPRINT, PRINT
 0042 3251 XTYPE, TYPE
 0043 2600 XERROR, ERROR
 0044 2667 XLOOP1, LOOP1
 0045 2712 XLOOP2, LOOP2
 0046 0000 SIMBSW, 0
 0047 3113 DCA TYE1
 0050 7012 KTR
 0051 7012 RTR
 0052 7012 RTR
 0053 1113 TAD
 0054 0062 AND K7700
 0055 1113 TAD TYE1
 0056 7006 RTL
 0057 7006 RTL
 0060 7006 RTL
 0061 5446 JMP I SIMBSW
 0062 7700 K7700, 7700

/ SIMCAF, 0
 0063 0000 IOF
 0064 6002 CLA CMA
 0065 7240 DBCI
 0066 4434 SKP
 0067 7410 HLT
 0070 7402 DBCO
 0071 4436 SKP
 0072 7410 HLT
 0073 7402 CLA CLL
 0074 7300 JMP I SIMCAF
 0075 5463

/DBCI SKIPPED
7DBCO SKIPPED

/ SIMSRQ, 0
 0076 0000 DCA
 0077 3113 TAD (RETURN
 0100 1177 DCA 2
 0101 3002 ION
 0102 6001 NOP
 0103 7000 IOF
 0104 6002 SKP
 0105 7410 RETURN, ISZ
 0106 2076 CLA CLL
 0107 7324 DCA 2
 0110 3002 SIMSRQ
 0111 1113 TAD TYE1

/NO INTERRUPT

INIT1

/INITIALIZATION CONSTANTS AND VARIABLES

/BASIC IOT

0247 0000 IOTS,
 0250 0262 TIOT,
 0251 0267 DBDIX+1
 0252 0274 DBEIX+1
 0253 0301 DBSKX+1
 0254 0306 DBCIX+1
 0255 0313 DBRIX+1
 0256 0320 DBCOX+1
 0257 0325 DBSOX+1
 0260 3650 DBROX+1
 DCA I TIOT

/IOT SUBROUTINES

/DISABLE DATA BUFFER INTERRUPT (DBDI,65X0)

0261 0000 DBDIX,
 0262 6500

/TRAP FOR UNDESIRE
/SKIPS

JMP I DBDIX

/ENABLE DATA BUFFER INTERRUPTS (DBEI,65X1)

0266 0000 DBEIX,
 0267 6501

/TRAP FOR UNDESIRE
/SKIPS

JMP I DBEIX

/SKIP ON DATA BUFFER INPUT FLAG (DBSK,65X2)

0273 0000 DBSKX,
 0274 6502

0275 7410 SKP
 0276 2273 ISZ DBSKX
 0277 5673 JMP I DBSKX

/DS TO INPUT REGISTER CORRESPONDING
/TO IS IN AC (DBCI,65X3)

0300 0000 DBCIX,
 0301 6503

/TRAP FOR UNDESIRE
/SKIPS

0302 7410 SKP
 0303 7402 HLT
 0304 5700 JMP I DBCIX

```

/INPUT REGISTER TO AC (DBRI,65X4)
/
DBRIX: 0 6504
0305 0000
0306 6504
0307 7410 SKP /TRAP FOR UNDESIRE
0310 7402 HLT /SKIPS
0311 5705 JMP I DBRIX

```

```

/ZEROS TO OUTPUT REGISTER CORRESPONDING TO
/ONES IN AC (DBCO,65X5)
/
DBCOX: 0 6505
0312 0000 SKP /TRAP FOR UNDESIRE
0313 6505 HLT /SKIPS
0314 7410 JMP I DBCOX
0315 7402
0316 5712

```

```

/IS TO OUTPUT REGISTER CORRESPONDING
/TO 1S IN AC (DBSO,65X6)
/
DBSOX: 0 6506
0317 0000 SKP /TRAP FOR UNDESIRE
0320 6506 HLT /SKIPS
0321 7410 JMP I DBSOX
0322 7402
0323 5712

```

```

/JAM TRANSFER OUTPUT REGISTER TO AC (DBRO 65X7)
/
DBROX: 0 6507
0324 0000 SKP /TRAP FOR UNDESIRE
0325 6507 HLT /SKIPS
0326 7410 JMP I DBROX
0327 7402
0330 5724
0375 7770
0376 6500
0377 0007
0400 0400

```

PAGE

```

/IS OUTPUT REGISTER CLEARED BY INITIALIZE?
/
INITI: DCA TYPEFLG /CLEAR ERROR FLAG
0400 3030 DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
0401 3031 CAF /INITIALIZE INTERFACE
0402 4063 DBRO /READ OUTPUT REGISTER
0403 4440 DCA DATA1 /SAVE REGISTER DATA
0404 3021 TAD DATA1 /GET REGISTER DATA
0405 1021 SNA CLA /WAS REGISTER CLEARED BY INITIALIZE
0406 7650

```

```

0407 5214 JMP .+5 /DATA CORRECT, CONTINUE
0410 4443 JMS I XERROR /NO, ERROR
0411 3645 INIT1E=1 /"OUTPUT REGISTER NOT CLEARED"
0412 3515 DH1=1 /"REGISTER DATA"
0413 7777 =1 /NUMBER OF WORDS TO BE OUTPUT
0414 4444 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0415 5202 JMP INIT1+2 /LOOP ON CURRENT TEST

```

```

/IS INPUT REGISTER CLEARED BY INITIALIZE?
/
INIT2,
0416 3030 DCA TYPFLG /CLEAR ERROR FLAG
0417 3031 DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
0420 7410 SKP /
0421 4063 CAF /INITIALIZE INTERFACE
0422 4435 DBRI /READ INPUT REGISTER
0423 3021 DCA DATA1 /SAVE REGISTER DATA
0424 1021 TAD DATA1 /GET REGISTER DATA
0425 7650 SNA CLA /WAS REGISTER CLEARED
0426 5233 JMP .+5 /DATA CORRECT, CONTINUE
0427 4443 JMS I XERROR /NO, ERROR
0430 3661 INIT2E=1 /"INPUT REGISTER NOT CLEARED"
0431 3515 DH1=1 /"REGISTER DATA"
0432 7777 =1 /NUMBER OF WORDS TO BE OUTPUT
0433 4444 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0434 5221 JMP INIT2+3 /LOOP ON CURRENT TEST

```

```

/IS SKIP FLAG SET AFTER INITIALIZE
/
INIT3,
0435 3030 DCA TYPFLG /CLEAR ERROR FLAG
0436 3031 DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
0437 7410 SKP /
0440 4063 CAF /INITIALIZE INTERFACE
0441 4432 DBEI /ENABLE INTERFACE INTERRUPTS
0442 4433 DBSK /IS INTERFACE FLAG SET
0443 5250 JMP .+5 /FLAG NOT SET, CONTINUE
0444 4443 JMS I XERROR /FLAG SET, ERROR
0445 3674 INIT3E=1 /"SKIP FLAG SET"
0446 3514 DH0=1 /NO DATA HEADER
0447 0000 0 /NO DATA
0450 4444 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0451 5240 JMP INIT3+3 /LOOP ON CURRENT TEST

```

```

/DOES OUTPUT REGISTER JAM TRANSFER TO AC?
/
TRAN1,
0452 3030 DCA TYPFLG /CLEAR ERROR FLAG
0453 3031 DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
0454 4063 CAF /INITIALIZE INTERFACE
0455 7340 CLA CLL CMA /SET AC =7777
0456 4440 DBRO /READ OUTPUT REGISTER
0457 3021 DCA DATA1 /SAVE AC CONTENTS
0460 1021 TAD DATA1 /GET AC CONTENTS
0461 7650 SNA CLA /WAS AC CLEARED BY TRANSFER
0462 5267 JMP .+5 /DATA CORRECT, CONTINUE
0463 4443 JMS I XERROR /NO, ERROR

```

```

0464 3703 /"DBRO DID NOT CLEAR AC"
0465 3525 /"AC CONTENTS"
E466 7777 /NUMBER OF WORDS TO BE OUTPUT
0467 4444 /CHECK FOR LOOP ON CURRENT TEST
0470 5254 /LOOP ON CURRENT TEST

/DOES INPUT REGISTER JAM TRANSFER TO AC
/
TRAN2,
0471 3030 /CLEAR ERROR FLAG
0472 3031 /SET ITERATION COUNT TO 4096(DECIMAL)
0473 4063 /INITIALIZE INTERFACE
0474 7340 /SET AC =7777
0475 4435 /READ INPUT REGISTER
0476 3021 /SAVE AC CONTENTS
0477 1021 /GET AC CONTENTS
0500 7650 /WAS AC CLEARED BY TRANSFER
0501 5306 /DATA CORRECT, CONTINUE
0502 4443 /NO, ERROR
0503 3716 /"DBR1 DID NOT CLEAR AC"
0504 3525 /"AC CONTENTS"
0505 7777 /NUMBER OF WORDS TO BE OUTPUT
0506 4444 /CHECK FOR LOOP ON CURRENT TEST
0507 5273 /LOOP ON CURRENT TEST

```

```

/DOES TRANSFER TO OUTPUT REGISTER CHANGE AC (WITH AC=7777,DBSO)
/
TRAN3,
0510 3030 /CLEAR ERROR FLAG
0511 3031 /SET ITERATION COUNT TO 4096(DECIMAL)
0512 4063 /INITIALIZE INTERFACE
0513 7340 /SET AC =7777
0514 4437 /BIT SET OUTPUT REGISTER
0515 3021 /SAVE AC CONTENTS
0516 1021 /GET AC CONTENTS
0517 7040 /COMPLIMENT DATA TO TEST FOR 7777
0520 7650 /DID AC CHANGE
0521 5326 /DATA CORRECT, CONTINUE
0522 4443 /NO, ERROR
0523 3731 /"DBSO CHANGED AC"
0524 3525 /"AC CONTENTS"
0525 7777 /NUMBER OF WORDS TO BE OUTPUT
0526 4444 /CHECK FOR LOOP ON CURRENT TEST
0527 5312 /LOOP ON CURRENT TEST

```

```

/DOES TRANSFER TO OUTPUT REGISTER CHANGE AC (WITH AC=7777, DBC0)
/
TRAN4,
0530 3030 /CLEAR ERROR FLAG
0531 3031 /SET ITERATION COUNT TO 4096(DECIMAL)
0532 4063 /INITIALIZE INTERFACE
0533 7340 /SET AC =7777
0534 4436 /BIT CLEAR OUTPUT REGISTER
0535 3021 /SAVE AC CONTENTS
0536 1021 /GET AC CONTENTS
0537 7040 /COMPLIMENT DATA TO TEST FOR 7777
0540 7650 /DID AC CHANGE

```

```

0541 5346 JMP .+5
0542 4443 JMS I XERROR
0543 3741 TRAN4E=1
0544 3525 DH2=1
0545 7777 *-1
0546 4444 JMS I XLOOP1
0547 5332 JMP TRAN4+2
/DOES TRANSFER TO INPUT REGISTER CHANGE AC (WITH AC=7777, DBCI)
/
TRAN5,
0550 3030 DCA TYPFLG
0551 3031 DCA LPCNT
0552 4063 CAF
0553 7340 CLA CLL CMA
0554 4434 DBCI
0555 3021 DCA DATA1
0556 1021 TAD DATA1
0557 7040 CMA CLA
0560 7650 SNA CLA
0561 5366 JMP .+5
0562 4443 JMS I XERROR
0563 3751 TRAN5E=1
0564 3525 DH2=1
0565 7777 *-1
0566 4444 JMS I XLOOP1
0567 5352 JMP TRAN5+2
0570 5777 JMP TRAN6
0577 0600
0600 0600

```

PAGE

```

/DOES TRANSFER TO OUTPUT REGISTER CHANGE AC (WITH AC=0,DBSO)
/
TRAN6,
0600 3030 DCA TYPFLG
0601 3031 DCA LPCNT
0602 4063 CAF
0603 4437 DBSO
0604 3021 DCA DATA1
0605 1021 TAD DATA1
0606 7650 SNA CLA
0607 5214 JMP .+5
0610 4443 JMS I XERROR
0611 3731 TRAN3E=1
0612 3525 DH2=1
0613 7777 *-1
0614 4444 JMS I XLOOP1
0615 5202 JMP TRAN6+2
/DOES TRANSFER TO OUTPUT REGISTER CHANGE AC (WITH AC=0,DBCO)
/
TRAN7,
0616 3030 DCA TYPFLG
0617 3031 DCA LPCNT
0620 4063 CAF
0621 4436 DBCO

```



```

0622 3021 DCA /SAVE AC CONTENTS
0623 1021 TAD /GET AC CONTENTS
0624 7650 SNA CLA /IS AC STILL 0
0625 5232 JMP .+5 /DATA CORRECT, CONTINYE
0626 4443 JMS I XERROR /NC, ERROR
0627 3741 TRAN4E=1 /"DBCO CHANGED AC"
0630 3525 DH2=1 /"AC CONTENTS"
0631 7777 =1 /NUMBER OF WORDS TO BE OUTPUT
0632 4444 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0633 5220 JMP TRAN7+2 /LOOP ON CURRENT TEST

```

```

/DOES TRANSFER TO INPUT REGISTER CHANGE AC (WITH AC=0, DBCI)
/
TRAN8, DCA TYPEFLG /CLEAR ERROR FLAG
DCA LPCNT /SET ITERATION COUNT TO 4096 (DECIMAL)
CAF /INITIALIZE INTERFACE
DBCI /BIT CLEAR INPUT REGISTER
DCA DATA1 /SAVE AC CONTENTS
TAD DATA1 /GET AC CONTENTS
SNA CLA /IS AC STILL 0
JMS I .+5 /DATA CORRECT, CONTINUE
JMS I XERROR /AC CHANGED, ERROR
TRAN5E=1 /"DBCI CHANGED AC"
DH2=1 /"AC CONTENTS"
=1 /NUMBER OF WORDS TO BE OUTPUTED
JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
JMP TRAN8+2 /LOOP ON CURRENT TEST
JMP OUT1 /GO TO NEXT TEST

```

PAGE

```

1000 3030 OUT1, DCA /CLEAR ERROR FLAG
1001 3031 DCA /SET ITERATION COUNT TO 4096(DECIMAL)
1002 4065 CAF /INITIALIZE INTERFACE
1003 7340 CLA CLL CMA /SET AC =7777
1004 4437 DBSO /BIT SET OUTPUT REGISTER
1005 4440 DBRO /READ OUTPUT REGISTER
1006 3021 DCA DATA1 /SAVE REGISTER DATA
1007 1021 TAD DATA1 /GET REGISTER DATA
1010 7040 CMA /COMPLEMENT DATA TO TEST FOR 7777
1011 7650 SNA CLA /IS REGISTER=7777
1012 5217 JMP .+5 /DATA CORRECT, CONTINUE
1013 4443 JMS I XERROR /NO, ERROR
1014 3761 OUT1E=1 /"DBSO ERROR"
1015 3515 DH1=1 /"REGISTER DATA"
1016 7777 =1 /NUMBER OF WORDS TO BE OUTPUT
1017 4063 CAF /INITIALIZE INTERFACE
1020 4440 DBRO /READ OUTPUT REGISTER
1021 3021 DCA DATA1 /SAVE REGISTER DATA
1022 1021 TAD DATA1 /GET REGISTER DATA
1023 7650 SNA CLA /WAS REGISTER CLEARED

```

```

1024 5231 JMP .+5 /DATA CORRECT, CONTINUE
1025 4443 JMS I XERROR /NO, ERROR
1026 3645 INITIE-1 /"OUTPUT REG NOT CLEARED"
1027 3515 DH1-1 /"REGISTER DATA"
1030 7777 -1 /NUMBER OF WORDS TO BE OUTPUT
1031 4444 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
1032 5202 JMP OUT1+2 /LOOP ON CURRENT TEST

/ CAN ALL BITS OF OUTPUT REGISTER BE CLEARED (DBCO)
/
1033 3030 OUT2, DCA TYPFLG /CLEAR ERROR FLAG
1034 3031 DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
1035 7340 CLA CLL CMA /SET AC =7777
1036 4437 DBSO /BIT SET OUTPUT REGISTER
1037 4436 DBCO /BIT CLEAR OUTPUT REGISTER
1040 7300 CLA CLL /READ OUTPUT REGISTER
1041 4440 DBRO /SAVE REGISTER DATA
1042 3021 DCA DATA1 /GET REGISTER DATA
1043 1021 TAD DATA1 /WAS OUTPUT REGISTER CLEARED
1044 7650 SNA CLA /DATA CORRECT, CONTINUE
1045 5252 JMP .+5 /NO, ERROR
1046 4443 JMS I XERROR /"OUTPUT REGISTER NOT CLEARED"
1047 3645 INITIE-1 /"REGISTER DATA"
1050 3515 DH1-1 /NUMBER OF WORDS TO BE OUTPUT
1051 7777 -1 /CHECK FOR LOOP ON CURRENT TEST
1052 4444 JMS I XLOOP1 /LOOP ON CURRENT TEST
1053 5235 JMP OUT2*2

```

```

/ CAN EACH BIT OF OUTPUT REGISTER BE SET
/ INDEPENDENTLY (DBSO)
/
1054 3030 OUT3, DCA TYPFLG /CLEAR ERROR FLAG
1055 1176 TAD I=62 /SET ITERATION COUNT
1056 3031 DCA LPCNT /TO 50(DECIMAL)
1057 3021 DCA DATA1 /CLEAR TEST DATA
1060 4063 CAF /INITIALIZE INTERFACE
1061 1021 TAD DATA1 /GET TEST DATA
1062 4437 DBSO /BIT SET OUTPUT REGISTER
1063 7300 CLA CLL /READ OUTPUT REGISTER
1064 4440 DBRO /SAVE REGISTER DATA
1065 3022 DCA DATA2 /GET TEST DATA
1066 1021 TAD DATA1 /COMPARE TO REGISTER CONTENTS
1067 7041 CIA DATA2 /DO THEY COMPARE
1070 1022 TAD SNA CLA /DATA CORRECT, CONTINUE
1071 7650 SNA CLA .+5 /NO, ERROR
1072 5277 JMP I XERROR /"DBSO ERROR"
1073 4443 JMS I /"EXPECTED RECEIVED"
1074 3761 OUT1E-1 /NUMBER OF WORDS TO BE OUTPUT
1075 3560 DH4*1 /TEST FOR LOOP ON SAME DATA, ESCAPE ON DATA ERROR
1076 7776 *2 /LOOP WITH SAME DATA
1077 4445 JMS I XLOOP2 /INCREMENT DATA PATTERN
1100 5260 JMP OUT3A /CONTINUE TEST
1101 2021 ISZ DATA1
1102 5260 JMP OUT3A

```

```

1123 4444 JMS I XLOOP1
1104 5257 JMP OUT3+3
/ CHECK FOR LOOP ON CURRENT TEST
/ LOOP ON CURRENT TEST
/ CAN EACH BIT OF OUTPUT REGISTER BE CLEARED
/ INDEPENDENTLY (DBCO)
/
OUT4:
1105 3030 DCA TYPFLG
1106 1176 TAD L=62
1107 3031 DCA LPCNT
1110 3021 DCA DATA1
OUT4A:
1111 4063 CAF
1112 1021 TAD DATA1
1113 7040 CMA
1114 3022 DCA DATA2
1115 7040 CMA
1116 4437 DBSO
1117 7300 CLA CLL
1120 1021 TAD
1121 4436 DBCO
1122 4440 DBRO
1123 3023 DCA
1124 1022 TAD
1125 7041 CIA
1126 1023 TAD
1127 7650 SNA CLA
1130 5335 JMP
1131 4443 JMS I
1132 3767 OUT4E=1
1133 3534 DB3=1
1134 7775 =3
1135 4445 JMS I
1136 5311 JMP
1137 2021 ISE
1140 5311 JMP
1141 4444 JMS I
1142 5310 JMP
1143 5777 JMP
1177 1200 PAGE
1200 3030 DCA TYPFLG
1201 1176 TAD L=62
1202 3031 DCA LPCNT
1203 3021 DCA DATA1
1204 3022 DCA DATA2
1205 4063 CAF
1206 1021 TAD DATA1
1207 4436 DBCO
1210 7300 CLA CLL
1211 4440 DBRO
1212 3023 DCA
1213 1023 TAD
/ WITH OUTPUT REGISTER CLEARED, DOES CLEARING
/ OUTPUT REGISTER CHANGE ANY BIT IN OUTPUT
/
OUT5:
1200 3030 DCA TYPFLG
1201 1176 TAD L=62
1202 3031 DCA LPCNT
1203 3021 DCA DATA1
1204 3022 DCA DATA2
1205 4063 CAF
1206 1021 TAD DATA1
1207 4436 DBCO
1210 7300 CLA CLL
1211 4440 DBRO
1212 3023 DCA
1213 1023 TAD
/ CLEAR ERROR FLAG
/ SET ITERATION COUNT
/ TO 50 (DECIMAL)
/ CLEAR TEST DATA
/ INITIALIZE INTERFACE
/ GET MASK
/ COMPLIMENT TO GET EXPECTED RESULT
/ SAVE EXPECTED RESULT
/ SET OUTPUT REGISTER TO 7777
/ BIT SET OUTPUT REGISTER
/ GET PATTERN TO CLEAR OUTPUT REGISTER
/ BIT CLEAR OUTPUT REGISTER
/ READ OUTPUT REGISTER
/ SAVE REGISTER DATA
/ GET EXPECTED RESULT
/ COMPARE TO RECEIVED DATA
/ WERE CORRECT BITS IN OUTPUT REGISTER CLEARED
/ DATA CORRECT, CONTINUE
/ NO ERROR
/ "DBCO ERROR"
/ MASK EXPECTED RECEIVED
/ NUMBER OF WORDS TO BE OUTPUT
/ TEST FOR LOOP ON SAME DATA, ESCAPE ON DATA ERROR
/ LOOP WITH SAME DATA
/ INCREMENT DATA PATTERN
/ CONTINUE TEST
/ CHECK FOR LOOP ON CURRENT TEST
/ LOOP ON CURRENT TEST
/ CLEAR ERROR FLAG
/ SET ITERATION COUNT
/ TO 50 (DECIMAL)
/ CLEAR TEST DATA
/ CLEAR EXPECTED RESULT
/ INITIALIZE INTERFACE
/ GET TEST DATA
/ BIT CLEAR OUTPUT REGISTER
/ READ OUTPUT REGISTER
/ SAVE REGISTER DATA
/ GET REGISTER DATA

```

```

1214 7650 SNA CLA
1215 5222 JMP .+5
1216 4443 JMS I XERROR
1217 3767 OUT4E-1
1220 3534 DH3-1
1221 7775 -3
1222 4445 JMS I XLOOP2
1223 5205 JMP OUT5A
1224 2021 ISZ DATA1
1225 5205 JMP OUT5A
1226 4444 JMS I XLOOP1
1227 5203 JMP OUT5+3

```

```

/IS OUTPUT REGISTER 0
/NO, ERROR
/"DBCO ERROR"
/"MASK EXPECTED RECEIVED"
/NUMBER OF DATA WORDS
/TEST FOR LOOP ON SAME DATA
/LOOP WITH SAME DATA
/INCREMENT DATA PATTERN
/CONTINUE
/CHECK FOR LOOP ON CURRENT TEST
/LOOP ON CURRENT TEST

```

```

/DOES SETTING OUTPUT REGISTER TWICE WITH SAME
/DATA CHANGE OUTPUT REGISTER

```

```

OUT6,
1230 3030 DCA TYPFLG
1231 1176 TAD C=62
1232 3031 DCA LPCNT
1233 3021 DCA DATA1
OUT6A,
1234 4063 CAF DATA1
1235 1021 TAD
1236 4437 DBSO
1237 4437 DBSO
1240 7300 CLA CLL
1241 4440 DBRO
1242 3022 DCA DATA2
1243 1021 TAD DATA1
1244 7041 CIA
1245 1022 TAD DATA2
1246 7650 SNA CLA
1247 5254 JMP .+5
1250 4443 JMS I XERROR
1251 3767 OUT4E-1
1252 3560 DH4-1
1253 7776 -2
1254 4445 JMS I XLOOP2
1255 5234 JMP OUT6A
1256 2021 ISZ DATA1
1257 5234 JMP OUT6A
1260 4444 JMS I XLOOP1
1261 5233 JMP OUT6+3

```

```

/CLEAR ERROR FLAG
/SET ITERATION COUNT
/TO 50 (DECIMAL)
/CLEAR TEST DATA
/INITIALIZE INTERFACE
/GET TEST DATA
/BIT SET OUTPUT REGISTER
/BIT SET OUTPUT REGISTER

```

```

/READ OUTPUT REGISTER
/SAVE REGISTER DATA
/GET TEST DATA
/COMPARE TO REGISTER DATA
/AARE THEY THE SAME
/NO, ERROR
/"DBSO ERROR"
/"EXPECTED RECEIVED"
/NUMBER OF DATA WORDS
/TEST FOR LOOP ON SAME DATA
/LOOP WITH SAME DATA
/INCREMENT DATA PATTERN
/CONTINUE
/CHECK FOR LOOP ON CURRENT TEST
/LOOP ON CURRENT TEST

```

```

/DOES READING OUTPUT REGISTER TWICE CHANGE
/OUTPUT REGISTER

```

```

OUT7,
1262 3030 DCA TYPFLG
1263 1176 TAD C=62
1264 3031 DCA LPCNT
1265 3021 DCA DATA1
OUT7A,
1266 4063 CAF DATA1
1267 1021 TAD
1270 4437 DBSO

```

```

/CLEAR ERROR FLAG
/SET ITERATION COUNT
/TO 50 (DECIMAL)
/CLEAR TEST DATA
/INITIALIZE INTERFACE
/GET TEST DATA
/BIT SET OUTPUT REGISTER

```

```

1271 7300 CLA CLL
1272 4440 DBRO
1273 7300 CLA CLL
1274 4440 DBRO
1275 3022 DATA2
1276 1021 DATA1
1277 7041 DATA2
1300 1022 SNA CLA
1301 7650 JMP I
1302 5307 JMP I
1303 4443 JMP I
1304 3775 OUT7E=1
1305 3560 DH4=1
1306 7776 =2
1307 4445 JMP I
1310 5266 ISZ
1311 2021 JMP I
1312 5266 JMP I
1313 4444 JMP I
1314 5265 JMP I
1315 7300 /READ OUTPUT REGISTER
1316 4440 /READ OUTPUT REGISTER
1317 7300 /SAVE REGISTER DATA
1318 4440 /GET TEST DATA
1319 3022 /COMPARE TO REGISTER DATA
1320 1021 /ARE THEY THE SAME
1321 7041 /DATA CORRECT, CONTINUE
1322 1022 /NO, ERROR
1323 7650 /"DBRO ERROR"
1324 5307 /"EXPECTED RECEIVED"
1325 4443 /NUMBER OF DATA WORDS
1326 3775 /TEST FOR LOOP WITH SAME DATA
1327 3560 /LOOP WITH SAME DATA
1328 7776 /INCREMENT DATA PATTERN
1329 4445 /CONTINUE
1330 5266 /CHECK FOR LOOP ON CURRENT TEST
1331 2021 /LOOP ON CURRENT TEST
1332 5266
1333 4444
1334 5265

```

```

1315 7300 /CLEAR ERROR FLAG
1316 4440 /SET ITERATION COUNT
1317 7300 /TO 50 (DECIMAL)
1318 4440 /CLEAR TEST DATA
1319 3022 /CLEAR EXPECTED RESULT
1320 1021 /INITIALIZE INTERFACE
1321 7041 /GET TEST DATA
1322 1022 /BIT SET OUTPUT REGISTER
1323 7650 /BIT CLEAR OUTPUT REGISTER
1324 5307 /BIT CLEAR OUTPUT REGISTER
1325 4443 /READ OUTPUT REGISTER
1326 3775 /SAVE REGISTER DATA
1327 3560 /GET REGISTER DATA
1328 7776 /IS REGISTER 0
1329 4445 /DATA CORRECT, CONTINUE
1330 5266 /NO, ERROR
1331 2021 /"DBCO ERROR"
1332 5266 /"MASK EXPECTED RECEIVED"
1333 4444 /NUMBER OF DATA WORDS
1334 5265 /TEST FOR LOOP WITH SAME DATA
1335 7300 /LOOP WITH SAME DATA
1336 4440 /INCREMENT DATA PATTERN
1337 3534 /CONTINUE
1338 7775 /CHECK FOR LOOP ON CURRENT TEST
1339 1341 /LOOP ON CURRENT TEST
1340 5322
1341 2021
1342 5322
1343 2021
1344 5322
1345 4444
1346 5320
1347 5777
1400 1400

```

```

1400 /CAN ALL BITS IN INPUT REGISTER BE SET
1401 /DOES INITIALIZE CLEAR INPUT REGISTER
1402 /CLEAR ERROR FLAG
1403 /SET ITERATION COUNT TO 4096(DECIMAL)
1404 /INITIALIZE INTERFACE
1405 /SET AC =7777
1406 /BIT SET OUTPUT REGISTER
1407 /READ INPUT REGISTER
1408 /SAVE REGISTER DATA
1409 /GET REGISTER DATA
1410 /COMPLIMENT TO TEST FOR 7777
1411 /WAS INPUT REGISTER SET TO 7777
1412 /DATA CORRECT, CONTINUE
1413 /NO, ERROR
1414 /"INPUT REGISTER NOT CORRECT"
1415 /REGISTER DATA"
1416 /NUMBER OF WORDS TO BE OUTPUT
1417 /INITIALIZE INTERFACE
1420 /READ INPUT REGISTER
1421 /SAVE REGISTER DATA
1422 /DATA CORRECT, CONTINUE
1423 /NO, ERROR
1424 /"INPUT REGISTER NOT CLEARED"
1425 /REGISTER DATA"
1426 /NUMBER OF WORDS TO BE OUTPUT
1427 /CHECK FOR LOOP ON CURRENT TEST
1430 /LOOP ON CURRENT TEST
1431
1432
1433

```

```

IN1,
3030 DCA I TYPFLG
3031 DCA I LPCNT
3032 CAF
3033 CLA CLL CMA
3034 DBSO
3035 CLA CLL
3036 DBRI
3037 DCA DATA1
3038 TAD DATA1
3039 CMA CLA
3040 SNA CLA
3041 JMP .+5
3042 JMS I XERROR
3043 IN3E=1
3044 DH1=1
3045 *-1
3046 CAF
3047 DBRI
3048 DCA DATA1
3049 TAD DATA1
3050 SNA CLA
3051 JMP .+5
3052 JMS I XERROR
3053 INIT2E=1
3054 DH1=1
3055 *-1
3056 JMS I XLOOP1
3057 JMP IN1A
3058
IN2,
3030 DCA I TYPFLG
3031 DCA I LPCNT
3032 CAF
3033 CLA CLL CMA
3034 AND FJUMPER
3035 DCA DATA1
3036 DCA DATA2
3037 TAD DATA1
3038 DBSO
3039 DBCI
3040 CLA CLL
3041 DBRI
3042 DCA DATA3
3043 TAD DATA3
3044 SNA CLA
3045 JMP .+5
3046 JMS I XERROR
3047 IN2E=1
3048 DH3=1
3049
3050

```

1457 7775 /NUMBER OF WORDS TO BE OUTPUT
 1460 4444 /CHECK FOR LOOP ON CURRENT TEST
 1461 5236 /LOOP ON CURRENT TEST

-3
 JMS I XLOOP1
 JMP IN2+2

1462 3030 /CAN EACH BIT OF INPUT REGISTER BE SET INDEPENDENTLY
 1463 1176 /CLEAR ERROR FLAG
 1464 3031 /SET ITERATION COUNT
 1465 3021 /TO 50(DECIMAL)
 1466 4063 /CLEAR TEST DATA
 1467 1021 /INITIALIZE INTERFACE
 1470 4437 /GET TEST DATA
 1471 7300 /BIT SET OUTPUT REGISTER
 1472 4435 /READ INPUT REGISTER
 1473 3022 /SAVE REGISTER DATA
 1474 1021 /GET TEST DATA
 1475 7041 /COMPARE TO RECEIVED DATA
 1476 1022 /ARE THEY THE SAME
 1477 7650 /DATA CORRECT: CONTINUE
 1500 5305 /NO. ERROR
 1501 4443 /"INPUT REGISTER DATA ERROR"
 1502 4011 /"EXPECTED RECEIVED"
 1503 3560 /NUMBER OF WORDS TO BE OUTPUT
 1504 7776 /TEST FOR LOOP ON SAME DATA,ESCAPE ON DATA ERROR
 1505 4445 /LOOP WITH SAME DATA
 1506 5266 /INCREMENT DATA PATTERN
 1507 2021 /CONTINUE TEST
 1510 5266 /CHECK FOR LOOP ON CURRENT TEST
 1511 4444 /LOOP ON CURRENT TEST
 1512 5265 /CHECK FOR LOOP ON CURRENT TEST
 1513 5777 /
 1577 1600 /
 1600 1600

PAGE

1600 3030 /VERIFY THAT ALL LATCHING INPUT LINES HOLD DATA
 1601 1176 /CLEAR ERROR FLAG
 1602 3031 /SET ITERATION COUNT
 1603 3023 /TO 50(DECIMAL)
 1604 4063 /CLEAR TEST DATA
 1605 1023 /INITIALIZE INTERFACE
 1606 0027 /GET TEST DATA
 1607 3021 /MASK OFF NON LATCHING BITS
 1610 1021 /SAVE AS EXPECTED RESULT
 1611 7450 /GET TEST DATA
 1612 5233 /ARE ANY BITS TO BE TESTED
 1613 4437 /NO. GET NEXT DATA WORD
 1614 4436 /BIT SET OUTPUT REGISTER
 1615 7300 /BIT CLEAR OUTPUT REGISTER
 1616 4435 /READ INPUT REGISTER

1600 3030 /CAN EACH BIT OF INPUT REGISTER BE SET INDEPENDENTLY
 1601 1176 /CLEAR ERROR FLAG
 1602 3031 /SET ITERATION COUNT
 1603 3023 /TO 50(DECIMAL)
 1604 4063 /CLEAR TEST DATA
 1605 1023 /INITIALIZE INTERFACE
 1606 0027 /GET TEST DATA
 1607 3021 /MASK OFF NON LATCHING BITS
 1610 1021 /SAVE AS EXPECTED RESULT
 1611 7450 /GET TEST DATA
 1612 5233 /ARE ANY BITS TO BE TESTED
 1613 4437 /NO. GET NEXT DATA WORD
 1614 4436 /BIT SET OUTPUT REGISTER
 1615 7300 /BIT CLEAR OUTPUT REGISTER
 1616 4435 /READ INPUT REGISTER

```

1617 3022 DCA /SAVE REGISTER DATA
1620 1021 TAD /GET EXPECTED RESULT
1621 7041 CIA
1622 1022 TAD DATA
1623 7650 SNA CLA
1624 5231 JMP .#5
1625 4443 JMS I XERROR
1626 4026 IN4E=1
1627 3560 DH4=1
1630 7776 =2
1631 4445 JMS I XLOOP2
1632 5204 JMP IN5A
1633 2023 ISZ DATA3
1634 5204 JMP IN5A
1635 4444 JMS I XLOOP1
1636 5203 JMP IN5+3

```

/VERIFY ALL NON LATCHING DATA BITS DO NOT HOLD DATA

```

IN6, 1637 3030 DCA /CLEAR ERROR FLAG
1640 1176 TAD /SET ITERATION COUNT
1641 3031 LPONT /TO 50(DECIMAL)
1642 3024 DCA /CLEAR TEST DATA
1643 3022 DCA /CLEAR EXPECTED RESULT
1644 4063 CAF /INITIALIZE INTERFACE
1645 1027 TAD /GET MASK FOR NON LATCHING BITS
1646 7040 CMA /CHANGE TO MASK OFF LATCHING BITS
1647 0024 AND
1650 3021 DCA /SAVE FOR TRANSMISSION
1651 1021 TAD /GET TEST DATA
1652 7450 SNA /ARE ANY BITS TO BE TESTED
1653 5272 JMP /NO GET NEXT DATA WORD
1654 4437 DBSO /BIT SET OUTPUT REGISTER
1655 4436 DBCO /BIT CLEAR OUTPUT REGISTER
1656 7300 CLA CLL
1657 4435 DBRI
1660 3023 DCA /READ INPUT REGISTER
1661 1023 TAD /SAVE REGISTER DATA
1662 7650 SNA CLA /GET RECEIVED DATA
1663 5270 JMP /DID ANY BITS HOLD DATA
1664 4443 JMS I XERROR /YES, ERROR
1665 4026 IN4E=1 /"LATCH ERROR"
1666 3534 DH3=1 /"MASK EXPECTED RECEIVED"
1667 7775 JMS I XLOOP2 /NUMBER OF WORDS TO BE OUTPUT
1670 4445 JMP IN6A /TEST FOR LOOP ON SAME DATA,ESCAPE ON DATA ERROR
1671 5244 ISZ DATA4 /LOOP WITH SAME DATA
1672 2024 JMP IN6A /INCREMENT DATA PATTERN
1673 5244 JMS I XLOOP1 /CONTINUE TEST
1674 4444 JMP IN6+3 /CHECK FOR LOOP ON CURRENT TEST
1675 5242 JMP IN6+3 /LOOP ON CURRENT TEST

```

/VERIFY THAT ALL LATCHING LINES CAN BE CLEARED INDEPENDENTLY


```

1676 3030 /DC A /CLEAR ERROR FLAG
1677 1176 /TAD /SET ITERATION COUNT
1700 3031 /DC A /TO 50(DECIMAL)
1701 3024 /DC A /CLEAR TEST
1702 4063 /CAF /INITIALIZE INTERFACE
1703 1027 /TAD /GET MASK FOR LATCHING BITS
1704 0024 /AND /MASK OFF NON LATCHING BITS
1705 3021 /DC A /SAVE FOR TRANSMISSION
1706 3022 /DC A /EXPECTED RESULT
1707 1021 /TAD /SET OUTPUT REGISTER=7777
1710 4437 /DBSO /BIT SET OUTPUT REGISTER
1711 4436 /DBCO /BIT CLEAR OUTPUT REGISTER
1712 7300 /CLA CLL /GET TEST DATA
1713 1021 /TAD /BIT CLEAR INPUT REGISTER
1714 4434 /DBCI /READ INPUT REGISTER
1715 7300 /CLA CLL /SAVE REGISTER DATA
1716 4435 /DBRI /COMPARE TO RECEIVED DATA
1717 3023 /DC A /ARE THEY THE SAME
1720 1023 /TAD /DATA CORRECT, CONTINUE
1721 7650 /SNA CLA /NO, ERROR
1722 5327 /JMP +5 /LATCH ERROR"
1723 4443 /JMS I /MASK EXPECTED RECEIVED"
1724 4026 /IN4E=1 /NUMBER OF WORDS TO BE OUTPUT
1725 3534 /DH3=1 /TEST FOR LOOP ON SAME DATA, ESCAPE ON DATA ERROR
1726 7775 /-3 /LOOP WITH SAME DATA
1727 4445 /JMS I /INCREMENT DATA PATTERN
1730 5302 /JMP /CONTINUE TEST
1731 2024 /ISZ /CHECK FOR LOOP ON CURRENT TEST
1732 5302 /JMP IN7A /LOOP ON CURRENT TEST
1733 4444 /JMS I XLOOP1 /GO TO NEXT TEST
1734 5301 /JMP IN7+3
1735 5777 /JMP IN8
1777 2000 /PAGE
2000 2000

```

```

2000 3030 /DC A /CLEAR ERROR FLAG
2001 1176 /TAD /SET ITERATION COUNT
2002 3031 /DC A /TO 50(DECIMAL)
2003 3021 /DC A /CLEAR TEST DATA
2004 3022 /DC A /CLEAR EXPECTED RESULT
2005 4063 /CAF /INITIALIZE INTERFACE
2006 1021 /TAD /GET TEST DATA
2007 4434 /DBCI /BIT CLEAR INPUT REGISTER
2010 7300 /CLA CLL /READ INPUT REGISTER
2011 4435 /DBRI /SAVE REGISTER DATA
2012 3023 /DC A /GET REGISTER DATA
2013 1023 /TAD /IS INPUT REGISTER 0
2014 7650 /SNA CLA /DATA CORRECT, CONTINUE
2015 5222 /JMP +5 /NO, ERROR
2016 4443 /JMS I

```

```

2017 4003 IN2E=1 /"DBCI ERROR"
2020 3534 DH3=1 /"MASK EXPECTED RECEIVED"
2021 7775 *3 /NUMBER OF DATA WORDS
2022 4445 JMS I /TEST FOR LOOP WITH SAME DATA
2023 5205 JMP IN8A /LOOP WITH SAME DATA
2024 2021 ISZ DATA1 /INCREMENT DATA PATTERN
2025 5205 JMP IN8A /CONTINUE
2026 4444 JMS I /CHECK FOR LOOP ON CURRENT TEST
2027 5203 JMP IN8+3 /LOOP ON CURRENT TEST

```

```

/DOES READING THE INPUT REGISTER TWICE
/CHANGE THE INPUT REGISTER
IN9, DCA TYPFLG /CLEAR ERROR FLAG
TAD L=62 /SET ITERATION COUNT
DCA LPCNT /TO 50 (DECIMAL)
DCA DATA1 /CLEAR TEST DATA
CAF /INITIALIZE INTERFACE
TAD DATA1 /GET TEST DATA
DBSO /BIT SET OUTPUT REGISTER
CLA CLL /READ INPUT REGISTER
DBRI
CLA CLL
DBRI
DCA DATA2 /READ INPUT REGISTER
TAD DATA1 /SAVE REGISTER DATA
CIA /GET TEST DATA
TAD DATA2 /COMPARE TO REGISTER DATA
SNA CLA /ARE THEY THE SAME
JMP +5 /DATA CORRECT, CONTINUE
JMS I XERROR /NO, ERROR
IN9E=1 /"DBCI ERROR"
DH4=1 /"EXPECTED RECEIVED"
=2 /NUMBER OF DAT WORDS
JMS I XLOOP2 /TEST FOR LOOP WITH SAME DATA
JMP IN9A /LOOP WITH SAME DATA
ISZ DATA1 /INCREMENT DATA PATTERN
JMP IN9A /CONTINUE
JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
JMP IN9+3 /LOOP ON CURRENT TEST

```

```

/DOES CLEARING INPUT REGISTER TWICE SET ANY BIT
/IN INPUT REGISTER
IN10, DCA TYPFLG /CLEAR ERROR FLAG
TAD L=62 /SET ITERATION COUNT
DCA LPCNT /TO 50 (DECIMAL)
DCA DATA2 /CLEAR EXPECTED RESULT
DCA DATA1 /CLEAR TEST DATA
CAF /INITIALIZE INTERFACE
TAD DATA1 /GET TEST DATA
DBSO /BIT SET OUTPUT REGISTER
DBCO

```

```

2030 3030 IN2E=1 /"DBCI ERROR"
2031 1176 DH3=1 /"MASK EXPECTED RECEIVED"
2032 3031 *3 /NUMBER OF DATA WORDS
2033 3021 JMS I /TEST FOR LOOP WITH SAME DATA
2034 4063 JMP IN8A /LOOP WITH SAME DATA
2035 1021 ISZ DATA1 /INCREMENT DATA PATTERN
2036 4437 JMP IN8A /CONTINUE
2037 7300 JMS I /CHECK FOR LOOP ON CURRENT TEST
2040 4435 JMP IN8+3 /LOOP ON CURRENT TEST
2041 7300
2042 4435
2043 3022
2044 1021
2045 7041
2046 1022
2047 7650
2050 5255
2051 4443
2052 4034
2053 3560
2054 7776
2055 4445
2056 5234
2057 2021
2060 5234
2061 4444
2062 5233

```

```

2274 4434 DBCI /BIT CLEAR INPUT REGISTER
2275 4434 DBCI /BIT CLEAR INPUT REGISTER
2276 7300 CLA CLL
2277 4435 DBRI /READ INPUT REGISTER
2100 3023 DCA /SAVE REGISTER DATA
2101 1023 TAD /GET REGISTER DATA
2102 7650 SNA CLA /IS INPUT REGISTER 0
2103 5310 JMP /DATA CORRECT, CONTINUE
2104 4443 JMS I /NO, ERROR
2105 4003 IN2E=1 /"DBCI ERROR"
2106 3534 DH3=1 /"MASK EXPECTED RECEIVED"
2107 7775 =3 /NUMBER OF DATA WORDS
2110 4445 JMS I XLOOP2 /TEST FOR LOOP WITH SAME DATA
2111 5270 JMP IN10A /LOOP WITH SAME DATA
2112 2021 ISZ DATA1 /INCREMENT DATA PATTERN
2113 5270 JMP IN10A /CONTINUE
2114 4444 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
2115 5206 JMP IN10+3 /LOOP ON CURRENT TEST
2116 5777 JMP INOU1 /GO TO NEXT TEST
2177 2200
2200 2200

```

PAGE

```

/ WITH BOTH INPUT AND OUTPUT REGISTERS CLEARED
/ DOES CLEARING OUTPUT SET
/ ANY BIT IN INPUT
/
INOU1,
2200 3030 DCA TYPFLG /CLEAR ERROR FLAG
2201 1176 TAD I=62 /SET ITERATION COUNT
2202 3031 DCA LPCNT /TO 50 (DECIMAL)
2203 3021 DCA DATA1 /CLEAR TEST DATA
2204 3022 DCA DATA2 /CLEAR EXPECTED RESULT
2205 4063 CAF /INITIALIZE INTERFACE
2206 1021 TAD DATA1 /GET TEST DATA
2207 4436 DBCO /BIT CLEAR OUTPUT REGISTER
2210 7300 CLA CLL
2211 4435 DBRI /READ INPUT REGISTER
2212 3023 DCA DATAS /SAVE REGISTER DATA
2213 1023 TAD DATAS /GET REGISTER DATA
2214 7650 SNA CLA /IS OUTPUT REGISTER 0
2215 5222 JMP /+5 /DATA CORRECT, CONTINUE
2216 4443 JMS I /NO, ERROR
2217 3767 GUT4E=1 /"DBCO ERROR"
2220 3534 DH3=1 /"MASK EXPECTED RECEIVED"
2221 7775 =3 /NUMBER OF DATA WORDS
2222 4445 JMS I XLOOP2 /TEST FOR LOOP WITH SAME DATA
2223 5205 JMP INOU1A /LOOP WITH SAME DATA
2224 2021 ISZ DATA1 /INCREMENT DATA PATTERN
2225 5205 JMP INOU1A /CONTINUE
2226 4444 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
2227 5203 JMP INOU1+3 /LOOP ON CURRENT TEST
/ WITH BOTH INPUT AND OUTPUT REGISTERS CLEARED
/ DOES CLEARING INPUT SET ANY BIT IN OUTPUT

```

```

2230 3030 /CLEAR ERROR FLAG
2231 1176 /SET ITERATION COUNT
2232 3031 /TO 50 (DECIMAL)
2233 3021 /CLEAR TEST DATA
2234 3022 /CLEAR EXPECTED RESULT
2235 4063 /INITIALIZE INTERFACE
2236 1021 /GET TEST DATA
2237 4434 /BIT CLEAR INPUT REGISTER
2240 7300
2241 4440 /READ OUTPUT REGISTER
2242 3023 /SAVE REGISTER DATA
2243 1023 /GET REGISTER DATA
2244 7650 /IS OUTPUT REGISTER 0
2245 5252 /DATA CORRECT, CONTINUE
2246 4443 /NO, ERROR
2247 4003 /"DBCI ERROR"
2250 3534 /"MASK EXPECTED RECEIVED"
2251 7775 /NUMBER OF DATA WORDS
2252 4445 /TEST FOR LOOP WITH CURRENT DATA
2253 5235 /LOOP WITH SAME DATA
2254 2021 /INCREMENT DATA PATTERN
2255 5235 /CONTINUE
2256 4444 /CHECK FOR LOOP ON CURRENT TEST
2257 5233 /LOOP ON CURRENT TEST

```

```

/ WITH THE OUTPUT REGISTER SET TO ALL 1S, AND
/ THE INPUT REGISTER CLEARED, DOES SELECTIVELY
/ CLEARING THE OUTPUT REGISTER SET ANY BIT IN
/ THE INPUT REGISTER

```

```

2260 3030 /CLEAR ERROR FLAG
2261 1176 /SET ITERATION COUNT
2262 3031 /TO 50 (DECIMAL)
2263 3021 /CLEAR TEST DATA
2264 3022 /CLEAR EXPECTED RESULT
2265 4063 /INITIALIZE INTERFACE
2266 7040 /SET AC=7777
2267 4437 /BIT SET OUTPUT REGISTER
2270 4434 /BIT CLEAR INPUT REGISTER
2271 7300 /GET FLIPFLOP JUMPER MASK
2272 1027
2273 7040
2274 3022 /GET TEST DATA2
2275 1021 /COMPLEMENT
2276 7040 /AND WITH COMPLEMENT OF JUMPER MASK
2277 0022 /TO GET EXPECTED RESULT
2300 3022 /GET TEST DATA
2301 1021 /BIT CLEAR OUTPUT REGISTER
2302 4436 /READ INPUT REGISTER
2303 7300 /SAVE REGISTER DATA
2304 4435 /GET REGISTER DATA
2305 3023
2306 1023
2307 7041

```

```

2310 1022 TAD DATA /COMPARE TO EXPECTED RESULT
2311 7650 SNA CLA /ARE THEY THE SAME
2312 5317 JMP .+5 /DATA CORRECT, CONTINUE
2313 4443 JMS I XERROR /NO, ERROR
2314 3767 OUT4E-1 /"DBCO ERROR"
2315 3534 CH3-1 /MASK EXPECTED RECEIVED
2316 7775 =3 /NUMBER OF DATA WORDS
2317 4445 JMS I XLOOP2 /TEST FOR LOOP WITH SAME DATA
2320 5265 JMP INOU3A /LOOP WITH SAME DATA
2321 2021 ISZ DATA1 /INCREMENT DATA PATTERN
2322 5265 JMP INOU3A /CONTINUE
2323 4444 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
2324 5263 JMP INOU3+3 /LOOP ON CURRENT TEST

```

/ WITH THE INPUT REGISTER SET TO ALL 1S, DOES SELECTIVELY
 / CLEARING THE OUTPUT REGISTER CLEAR ANY BITS IN THE INPUT
 / REGISTER (EXCEPT THOSE NOT FLIPFLOPS)

```

INOU4,
2325 3030 OCA TYPFLG /CLEAR ERROR FLAG
2326 1176 TAD I=62 /SET ITERATION COUNT
2327 3031 DCA LPCNT /TO 50 (DECIMAL)
2330 3021 DCA DATA1 /CLEAR TEST DATA
INOU4A,
2331 4063 CAF /INITIALIZE INTERFACE
2332 7040 CMA /SET AC TO 7777
2333 4437 DBSO /BIT SET OUTPUT REGISTER
2334 7300 CLA CLL /GET FLIPFLOP JUMPER MASK
2335 1027 TAD FJUMPER
2336 7040 CMA
2337 0021 AND /COMBINE WITH MASK
2340 7040 CMA
2341 3022 DCA DATA2 /TO GET EXPECTED RESULT
2342 1021 TAD DATA1 /GET TEST DATA
2343 4436 DBCO /BIT CLEAR OUTPUT REGISTER
2344 7300 CLA CLL
2345 4435 DBRI /READ INPUT REGISTER
2346 3023 DCA DATA3 /SAVE REGISTER DATA
2347 1022 TAD DATA2 /GET EXPECTED RESULT
2350 7041 CIA
2351 1023 TAD DATA3 /COMPARE TO RECEIVED DATA
2352 7650 SNA CLA /ARE THEY THE SAME
2353 5360 JMP .+5 /DATA CORRECT, CONTINUE
2354 4443 JMS I XERROR /NO, ERROR
2355 3767 OUT4E-1 /"DBCO ERROR"
2356 3534 CH3-1 /MASK EXPECTED RECEIVED"
2357 7775 =3 /NUMBER OF DATA WORDS
2360 4445 JMS I XLOOP2 /TEST FOR LOOP WITH CURRENT DATA
2361 5331 JMP INOU4A /LOOP WITH SAME DATA
2362 2021 ISZ DATA1 /INCREMENT DATA PATTERN
2363 5331 JMP INOU4A /CONTINUE
2364 4444 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
2365 5330 JMP INOU4+3 /LOOP ON CURRENT TEST
2366 5777, JMP INT1 /GO TO NEXT TEST
2377 2400

```

```

/VERIFY THAT EACH BIT SET UP TO SKIP DOES
/
INT1, 3030 /CLEAR ERROR FLAG
      2401 1176 /SET ITERATION COUNT
      2402 3031 /TO 50(DECIMAL)
      2403 3022 /INITIALIZE INTERFACE
      2404 4063
      2405 1022
      2406 0026
      2407 7450
      2410 5233
      2411 3021
      2412 1021
      2413 4437
      2414 7300
      2415 4432
      2416 4076
      2417 5223
      2420 4433
      2421 5240
      2422 5231
      2423 4433
      2424 5245
      2425 4443
      2426 4042
      2427 3515
      2430 7777
      2431 4445
      2432 5204
      2433 2022
      2434 5204
      2435 4444
      2436 5203
      2437 5252
      2440 4443
      2441 4065
      2442 3515
      2443 7777
      2444 5231
      2445 4443
      2446 4077
      2447 3515
      2450 7777
      2451 5231

INT1A,
      DCA /CLEAR ERROR FLAG
      TAD /SET ITERATION COUNT
      DCA /TO 50(DECIMAL)
      DCA /INITIALIZE INTERFACE
      CAF
      TAD
      AND
      SNA
      JMP
      DCA
      TAD
      DBS0
      CLA CLL
      DBEI
      SRQ
      JMP
      DBSK
      JMP
      JMP
      DBSK
      JMP
      JMS I
      INT1E-1
      DH1=1
      -1
      INT1OK, JMS I
      JMP
      ISZ
      JMP
      JMS I
      JMP
      JMP
      JMS I
      INT3E-1
      DH1=1
      =1
      JMP
      JMS I
      INT4E-1
      DH1=1
      =1
      JMP

INT1AE,
      INT1AE /ENABLE INTERFACE
      INT1BE /IS INTERRUPT ACTIVE
      INT1OK /NO, ERROR
      INT1CE /IS FLAG SET
      XERROR /NO, ERROR

INT1OK,
      XLOOP2 /NUMBER OF WORDS TO BE OUTPUT
      INT1A /TEST FOR LOOP ON SAME DATA, ESCAPE ON DATA ERROR
      DATA2 /LOOP WITH SAME DATA
      INT1A /INCREMENT DATA PATTERN
      XLOOP1 /CONTINUE
      INT1+3 /CHECK FOR LOOP ON CURRENT TEST
      INT3 /LOOP ON CURRENT TEST
      XERROR /GO TO NEXT TEST
      /NO, ERROR

INT1BE,
      INT1OK /NUMBER OF WORDS TO BE OUTPUT
      XERROR /NO, ERROR

INT1CE,
      INT1OK /NUMBER OF WORDS TO BE OUTPUT
      XERROR /NO, ERROR

INT3,
      DCA /CLEAR ERROR FLAG
      TAD /SET ITERATION COUNT
      DCA /TO 50(DECIMAL)
      DCA /CLEAR TEST DATA
      CAF /INITIALIZE INTERFACE

```

```

2457 1026 TAD IJUMPER /GET JUMPER MASK
2460 7040 CMA /COMPLIMENT FOR NO SKIP BITS
2461 0022 AND DATA2 /GET BITS TO BE TESTED
2462 7450 SNA /ARE ANY BITS TO BE TESTED
2463 5300 JMP INT3C /NO, GET NEXT DATA PATTERN
2464 3021 DCA DATA1 /SAVE FOR OUTPUT
2465 1021 TAD DATA1 /GET TEST DATA
2466 4437 DBSO /BIT SET OUTPUT REGISTER
2467 7300 CLA CLL
2470 4433 DBSK
2471 5276 JMP *5 /IS FLAG SET
2472 4443 JMS I XERROR /NC, CONTINUE
2473 3674 INITSE=1 /YES, ERROR
2474 3515 CH1=1 /"SKIP FLAG SET"
2475 7777 *-1 /"REGISTER DATA"
2476 4445 JMS I XLOOP2
2477 5256 JMP INT3A
2500 2022 ISZ DATA2
2501 5256 JMP INT3A
2502 4444 JMS I XLOOP1
2503 5255 JMP INT3+3
2504 5777 JMP EPASS

```

/ERROR HANDLER

INT3C:

```

2577 3257 PAGE
2600 0000 ERROR:
2601 7300 CLA CLL
2602 1600 TAD I ERROR
2603 3234 DCA MESS /GET POINTER TO ERROR MESSAGE
2604 2200 ISZ ERROR /SAVE POINTER
2605 1600 TAD I ERROR /GET POINTER TO DATA HEADER
2606 3236 DCA DHDER /SAVE HEADER
2607 2200 ISZ ERROR /NUMBER OF WORDS TO BE OUTPUT
2610 1600 TAD I ERROR /GET NUMBER OF DATA WORDS TO BE TYPED
2611 3264 DCA DATCNT /SAVE
2612 1200 TAD ERROR /GET ADDRESS OF TEST THAT FAILED
2613 1377 TAD (*3)
2614 3776' DCA LSTDGT
2615 1776' TAD LSTDGT
2616 3266 DCA ERRAD
2617 7604 LAS
2620 0334 AND SR01
2621 7640 SZA CLA
2622 5254 JMP EHALT
2623 1030 TAD TYPFLG
2624 7640 SZA CLA
2625 5241 JMP DATOUT
2626 7040 CMA
2627 3030 DCA TYPFLG
2630 4775' JMS I OCTASC
2631 4441 JMS I XPRINT
2632 3024 ERADR=1

```

/CLEAR ERROR FLAG

```

2633 4441 JMS I XPRINT
2634 0000 MSG,
2635 4441 JMS I XPRINT
2636 0000 DHDR,
2637 4441 JMS I XPRINT
2640 3512 CRLF-1
2641 1264 TAD DATCNT
2642 7650 SNA CLA
2643 5254 JMP EHALT
2644 1265 TAD DATAP
2645 3010 DCA POINT1
2646 1410 TAD I POINT1
2647 4774 JMS BITOUT
2650 2264 ISZ DATCNT
2651 5246 JMP I BITS
2652 4441 JMS I XPRINT
2653 3512 CRLF-1
2654 7604 LAS EHALT,
2655 0333 AND SR00
2656 7640 SZA CLA .+3
2657 5262 JMP ERRAD
2660 1266 TAD HLT
2661 7402 HLT
2662 2200 ISZ ERROR
2663 5600 JMP I ERROR
2664 0000 DATCNT,
2665 0020 DATAP,
2666 0000 ERRAD,

```

/NUMBER OF WORDS TO BE OUTPUT

/TEST FOR LOOP ON CURRENT TEST

```

2667 0000 LOOP1,
2670 1030 TAD TYPFLG
2671 7650 SNA CLA
2672 5277 JMP LPIEXA
2673 7604 LAS SR06
2674 0341 AND
2675 7640 SZA CLA
2676 5310 JMP LPIEXX-1
2677 7604 LAS SR05
2700 0340 AND
2701 7640 SZA CLA
2702 5305 JMP LPIEXT
2703 2031 ISZ LPCNT
2704 5311 JMP LPIEXX
2705 7604 LAS SR02
2706 0335 AND
2707 7650 SNA CLA
2710 2267 ISZ LOOP1
2711 5667 JMP I LOOP1

2712 0000 LOOP2,
2713 1030 TAD TYPFLG

```

/TEST FOR LOOP ON CURRENT DATA


```

2714 7650 SNA CLA
2715 5326 LP2EXT
2716 7604 LAS
2717 0341 AND SR06
2720 7652 SNA CLA
2721 5326 JMP .+5
2722 1312 TAD LOOP2
2723 1373 TAD (5
2724 3312 DCA LOOP2
2725 5712 JMP I LOOP2
2726 7604 LP2EXT: LAS
2727 0336 AND SR03
2730 7650 SNA CLA
2731 2312 ISZ LOOP2
2732 5712 JMP I LOOP2
2733 4000 SR00:
2734 2000 SR01:
2735 1000 SR02:
2736 0400 SR03:
2737 0200 SR04:
2740 0100 SR05:
2741 0040 SR06:
    
```

/DATA CORRECT, CONTINUE

/OCTAL TO PACKED ASCII CONVERSION

```

2773 3005 PAGE /
2774 3031 OCTASC:
2775 3000 CLA CLL
2776 3027 TAD LSTDGT
2777 7775 RSW JMS
3000 3000 JMS MSTDGT
3001 7300 DCA TAD
3002 1227 TAD JMS
3003 4046 DCA JMP I
3004 4212 AND
3005 3226 MQL (77
3006 1227 MGA
3007 4212 CLL RTL
3010 3227 RAL
3011 5600 AND
3012 0000 MGA
3013 0377 CLL
3014 3114 MGA
3015 4115 RAL
3016 7106 AND
3017 7004 MGA
3020 0376 AND
3021 4115 MGA
3022 0376 AND
3023 1375 TAD
3024 5612 JMP I
3025 3736 TEXT
3026 4040 /
    
```

/GET WORD TO BE CONVERTED
/SWAP HALVES, SEPARATE DIGITS,
/CONVERT MOST SIGNIFICANT
/DIGITS TO ASCII
/CONVERT LEAST SIGNIFICANT
/DIGITS TO ASCII
/RETURN

```

3027 4040
3030 4000

/OUTPUT 12 BIT BINARY WORD
BITOUT, 2
3231 0000 /SAVE DATA IN MQ
3232 3114 /SET UP TO OURPUT
3033 1374 /12 BITS
3034 3020 /GET DATA
3035 4115 /GET MSB INTO LINK
3036 7104 /SAVE REST OF WORD
3037 3114 /GET ASCII 1 INTO AC
3040 1373
3041 7420
3042 0372 /IS BIT=1
3043 4442 /NO, CHANGE TO ASCII 0
3044 2020 /OUTPUT BIT
3045 5235 /CONTINUE
3046 1371 /TYPE 2 SPACES
3047 4442 /AFTER LAST BIT HAS BEEN
3050 1371 /OUTPUTTED
3051 4442 /RETURN
3052 5631
    
```

/CHARACTER STRING OUTPUT ROUTINE

```

3171 0240
3172 0260
3173 0261
3174 7764
3175 6060
3176 0707
3177 0077

PAGE 0
PRINT, 0
CLA CLL
TAD I PRINT
DCA POINT1
ISZ PRINT
TAD I POINT1
MQL
MCA
BSW
JMS TYPSET
MCA
JMS TYPSET
JMP PRINT+5

/UNPACK, DECODE, OUTPUT
TYPSET, 0
AND K0077
SNA
JMP I PRINT

/GET POINTER TO MESSAGE
/SET UP AUTO-INDEX REGISTER
/SET UP RETURN
/GET PACKED WORD
/SAVE IN MQ
/GET WORD
/SWAP HALVES
/DECODE AND OUTPUT
/GET WORD
/DECODE AND OUTPUT
/CONTINUE
    
```

3215 0000 /MASK UNWANTED BITS
 3216 0243 /IS AC=0
 3217 7450 /YES, END OF MESSAGE, EXIT
 3220 5600

3221	1244	TAD	M40	/SUBTRACT 40
3222	7510	SPA		/IS PACKED CHARACTER >40
3223	5226	JMP	+3	/NC
3224	1250	TAD	K240	/YES; CONVERT TO ASCII
3225	5241	JMP	MTP	/OUTPUT
3226	7001	IAC		/ADD 1 TO AC
3227	7440	SZA		/IS CHARACTER=37
3230	5233	JMP	+3	/NO;
3231	1245	TAD	K215	/GET CODE FOR CARRIAGE RETURN
3232	5241	JMP	MTP	/OUTPUT
3233	7001	IAC		/ADD 1 TO AC
3234	7440	SZA		/IS CHARACTER=37
3235	5240	JMP	+3	/NO
3236	1246	TAD	K212	/GET CODE FOR LINE FEED
3237	5241	JMP	MTP	/OUTPUT
3240	1247	TAD	K336	/PACKED CHARACTER >40; CONVERT TO ASCII
3241	4442	JMS I	XTYPE	/OUTPUT
3242	5615	JMP I	TYPSET	
3243	0077	77		
3244	7740	M40,		
3245	0215	K215,		
3246	0212	K212,		
3247	0336	K336,		
3250	0240	K240,		
		/		/OUTPUT ONE CHARACTER TO TTY
		/		
		/		
3251	0000	TYPE,		
3252	6046	TLS		
3253	6041	TSF		
3254	5253	JMP	.-1	
3255	7200	CLA		
3256	5651	JMP I	TYPE	
		/		
		/		
		/		
3257	7604	LAS		
3260	2777	AND	SR04	
3261	7640	SZA	CLA	
3262	5776	JMP	INIT1	
3263	7604	LAS		
3264	0775	AND	SR05	
3265	7640	SZA	CLA	
3266	5272	JMP	EPAS1	
3267	4441	JMS I	XPRINT	
3270	3274	MEP-1		
3271	5776	JMP	INIT1	
3272	1374	TAD	(207	
3273	4251	JMS	TYPE	
3274	5776	JMP	INIT1	
3275	3736	MEP,		
3276	0422	TEXT	/←DR/	
3277	0000			

/TELETYPE MESSAGES

3374	0207		
3375	2740		
3376	3400		
3377	2737		
3400	3400		
3400	3736		
3401	2305		
3422	2440		
3403	2322		
3404	4006		
3405	1722		
3406	4004		
3407	0526		
3410	1103		
3411	0540		
3412	0317		
3413	3405		
3414	4001		
3415	1604		
3416	4003		
3417	1716		
3420	2400		
3421	3736		
3422	2305		
3423	2440		
3424	2322		
3425	4006		
3426	1722		
3427	4011		
3430	1624		
3431	0522		
3432	2225		
3433	2024		
3434	4012		
3435	2515		
3436	2005		
3437	2223		
3440	4001		
3441	1604		
3442	4003		
3443	1716		
3444	2400		
3445	3736		
3446	2305		
3447	2440		
3450	2327		
3451	1124		
3452	0310		
3453	0523		
3454	4006		
3455	1722		
3456	4006		

PAGE M1. TEXT /->SET SR FOR DEVICE CODE AND CONT/

M2. TEXT /->SET SR FOR INTERRUPT JUMPERS AND CONT/

M2A. TEXT /->SET SWITCHES FOR FLIPFLOP JUMPERS AND CONTINUE/

3457 1411
 3460 2006
 3461 1417
 3462 2040
 3463 1225
 3464 1520
 3465 0522
 3466 2340
 3467 0116
 3470 0440
 3471 0317
 3472 1624
 3473 1116
 3474 2505
 3475 0000
 3476 3736
 3477 2305
 3500 2440
 3501 2322
 3502 4006
 3503 1722
 3504 4022
 3525 2516
 3506 4001
 3507 1604
 3510 4003
 3511 1716
 3512 2400
 3513 3736
 3514 0000

H3. TEXT /-SET SR FOR RUN AND CONT/

CRLF. TEXT /-?/

/DATA HEADERS

DH0, DH1, TEXT /-REGISTER DATA/

DH2, TEXT /-PAC CONTENTS/

DH3, TEXT /-MASK EXPECTED RECEIVED/

3515 0000
 3516 3736
 3517 2205
 3520 0711
 3521 2324
 3522 0522
 3523 4004
 3524 0124
 3525 0100
 3526 3736
 3527 0103
 3530 4003
 3531 1716
 3532 2405
 3533 1624
 3534 2300
 3535 3736
 3536 1501
 3537 2313
 3540 4040
 3541 4040
 3542 4040

3543	4040
3544	4040
3545	0530
3546	2005
3547	0324
3550	0504
3551	4040
3552	4040
3553	4040
3554	2205
3555	0305
3556	1126
3557	0504
3560	0000
3561	3736
3562	0530
3563	2005
3564	0324
3565	0504
3566	4040
3567	4040
3570	4040
3571	2205
3572	0305
3573	1126
3574	0504
3575	0000
3576	3736
3577	2205
3600	0711
3601	2324
3602	0522
3603	4040
3604	4040
3605	4040
3606	4004
3607	0124
3610	0140
3611	1725
3612	2440
3613	4040
3614	4040
3615	4040
3616	0401
3617	2401
3620	4011
3621	1600
3622	3736
3623	0103
3624	4003
3625	1716
3626	2405
3627	1624
3630	2340
3631	4040

DH4: TEXT /%EXPECTED RECEIVED/

DH5: TEXT /%REGISTER DATA OUT DATA IN/

DH6: TEXT /%AC CONTENTS DATA OUT DATA IN/

3632 4004
 3633 0124
 3634 0140
 3635 1725
 3636 2440
 3637 4040
 3640 4040
 3641 4040
 3642 0401
 3643 2401
 3644 4011
 3645 1600

/ERROR MESSAGE

INIT1E, TEXT /OUTPUT REG NOT CLEARED/

3646 1725
 3647 2420
 3650 2524
 3651 4022
 3652 0507
 3653 4016
 3654 1724
 3655 4003
 3656 1405
 3657 0122
 3660 0504
 3661 0000
 3662 1116
 3663 2025
 3664 2440
 3665 2205
 3666 0740
 3667 1617
 3670 2440
 3671 0314
 3672 0501
 3673 2205
 3674 0400
 3675 2313
 3676 1120
 3677 4006
 3700 1401
 3701 0740
 3702 2305
 3703 2400
 3704 0402
 3705 2217
 3706 4004
 3707 1104
 3710 4016
 3711 1724
 3712 4003
 3713 1405
 3714 0122
 3715 4001

INIT2E, TEXT /INPUT REG NOT CLEARED/

INIT3E, TEXT /SKIP FLAG SET/

TRAN1E, TEXT /DBRO DID NOT CLEAR AC/

3716 0300	
3717 0402	TRANSE, TEXT /DBRI DID NOT CLEAR AC/
3720 2211	
3721 4004	
3722 1104	
3723 4016	
3724 1724	
3725 4003	
3726 1405	
3727 0122	
3730 4001	
3731 0300	TRANSE, TEXT /DBSO CHANGED AC/
3732 0402	
3733 2317	
3734 4003	
3735 1001	
3736 1607	
3737 0504	
3740 4001	
3741 0300	
3742 0402	TRANSE, TEXT /DBCO CHANGED AC/
3743 0317	
3744 4003	
3745 1001	
3746 1607	
3747 0504	
3750 4001	
3751 0300	TRANSE, TEXT /DBCI CHANGED AC/
3752 0402	
3753 0311	
3754 4003	
3755 1001	
3756 1607	
3757 0504	
3760 4001	
3761 0300	
3762 0402	OUT1E, TEXT /DBSO ERROR/
3763 2317	
3764 4005	
3765 2222	
3766 1722	
3767 0000	
3770 0402	OUT4E, TEXT /DBCO ERROR/
3771 0317	
3772 4005	
3773 2222	
3774 1722	
3775 0000	
3776 0402	OUT7E, TEXT /DBRO ERROR/
3777 2217	
4000 4005	
4001 2222	
4002 1722	
4003 0000	
4004 0402	IN2E, TEXT /DBCI ERROR/

4005	0311		
4006	4005		
4007	2222		
4010	1722		
4011	0000		
4012	1116		
4013	2025		
4014	2440		
4015	2205		
4016	0711		
4017	2324		
4020	0522		
4021	4004		
4022	0124		
4023	0140		
4024	0522		
4025	2217		
4026	2200		
4027	1401		
4030	2403		
4031	1040		
4032	0522		
4033	2217		
4034	2200		
4035	0402		
4036	2211		
4037	4005		
4040	2222		
4041	1722		
4042	0000		
4043	1116		
4044	2405		
4045	2222		
4046	2520		
4047	2440		
4050	0103		
4051	2411		
4052	2605		
4053	0000		
4054	1617		
4055	4011		
4056	1624		
4057	0522		
4060	2225		
4061	2024		
4062	5440		
4063	2313		
4064	1120		
4065	0000		
4066	1116		
4067	2405		
4070	2222		
4071	2520		
4072	2454		
4073	4016		

IN3E, TEXT /INPUT REGISTER DATA ERROR/

IN4E, TEXT /LATCH ERROR/

IN9E, TEXT /DBRI ERROR/

INT1E, TEXT /INTERRUPT ACTIVE/

INT2E, TEXT /NO INTERRUPT, SKIP/

INT3E, TEXT /INTERRUPT, NO SKIP/

	INT4E,	TEXT	/NO INTERRUPT, NO SKIP/
4074		1740	
4075		2313	
4076		1120	
4077		0000	
4100		1617	
4101		4011	
4102		1624	
4103		0522	
4104		2225	
4105		2024	
4106		5440	
4107		1617	
4110		4023	
4111		1311	
4112		2000	
4113		1617	
4114		4023	
4115		1311	
4116		2000	
4117		2313	
4120		1120	
4121		0000	

\$

0176	7716
0177	0106

4000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
4100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

4200
4300

4400
4500

4600
4700

5000
5100

5200
5300

5400
5500

5600
5700

6000
6100

6200
6300

6400
6500

6600
6700

7000
7100

7200
7300

7400
7500

7600
7700

BIT01	3035	IN2	1434	K212	3246	SR00	2733
BITOUT	3031	IN2E	4004	K215	3245	SR01	2734
BITS	2646	IN3	1462	K240	3250	SR02	2735
BSW	4046	IN3A	1466	K336	3247	SR03	2736
CAF	4063	IN3E	4012	K770	0062	SR04	2737
CNTR1	0020	IN4E	4027	LOOP1	2667	SR05	2740
CRLF	3513	IN5	1600	LOCP2	2712	SR06	2741
DATA1	0021	IN5A	1604	LPIEXA	2677	SRQ	4076
DATA2	0022	IN5C	1633	LP1EXT	2705	START1	0202
DATA3	0023	IN6	1637	LP1EXX	2711	START2	0244
DATA4	0024	IN6A	1644	LP2EXT	2726	T1OT	0250
DATA5	0025	IN6C	1672	LPCNT	0031	TRAN1	0452
DATAP	2665	IN7	1676	LSTDGT	3027	TRAN1E	3704
DATCNT	2664	IN7A	1702	M1	3400	TRAN2	0471
DATOUT	2641	IN7C	1731	M2	3421	TRAN2E	3717
DBC1	4434	IN8	2000	M2A	3445	TRAN3	0510
DBCIX	0300	IN8A	2005	M3	3476	TRAN3E	3732
DBCX	4436	IN9	2030	M40	3244	TRAN4	0530
DBC0X	0312	IN9A	2034	MEP	3275	TRAN4E	3742
DBDIX	0261	IN9E	4035	MSG	2634	TRAN5	0550
DBE1	4432	INIT1	0400	HQ	0114	TRAN5E	3752
DBEIX	0266	INIT1E	3646	HQA	6115	TRAN6	0600
DBRI	4435	INIT2	0416	HQL	3114	TRAN7	0616
DBRIX	0305	INIT2E	3662	MSTDGT	3026	TRAN8	0634
DBRO	4440	INIT3	0435	HYP	3241	TYPE	0113
DBROX	0324	INIT3E	3675	OCTASC	3000	TYPFLG	3251
DBSK	4433	INOU1	2200	OUT1	1000	TYPSET	0030
DBSKX	0273	INOU1A	2205	OUT1E	3762	XDBCI	3215
DBSO	4437	INOU2	2230	OUT2	1033	XDBCO	0034
DBSOX	0317	INOU2A	2235	OUT3	1054	XDBEI	0036
DH0	3515	INOU3	2260	OUT3A	1060	XDBRI	0035
DH1	3516	INOU3A	2265	OUT4	1105	XDBRO	0040
DH2	3526	INOU4	2325	OUT4A	1111	XDBSK	0033
DH3	3535	INOU4A	2331	OUT4E	3770	XDBSO	0037
DH4	3561	INT1	2400	OUT5	1200	XERROR	0043
DH5	3576	INT1A	2404	OUT5A	1205	XL00P1	0044
DH6	3622	INT1AE	2423	OUT6	1230	XL00P2	0045
DHDER	2636	INT1BE	2440	OUT6A	1234	XPRINT	0041
D1OT	0260	INT1CE	2445	OUT7	1262	XTYPE	0042
EHALT	2654	INT1D	2433	OUT7A	1266		
EPAS1	3272	INT1E	4043	OUT7E	3776		
EPASS	3257	INT1QK	2431	OUT8	1315		
ERRADR	3025	INT2E	4054	OUT8A	1322		
ERRAD	2666	INT3	2452	PNTR1	0233		
ERROR	2600	INT3A	2456	POINT1	0010		
FJUMPE	0027	INT3C	2500	PRINT	3200		
IJUMPE	0026	INT3E	4066	RETURN	0106		
IN1	1400	INT4	4100	SIMBSW	0046		
IN10	2063	INT5E	4113	SIMCAF	0063		
IN10A	2070	INT6E	4117	SIMMOA	0115		
IN1A	1403	IOTS	0247	SIMSRQ	0076		
IN1B	1420	K0077	3243	SPLIT	3012		

ERRORS DETECTED: 0

LINKS GENERATED: 18

RUN-TIME: 10 SECONDS

3K CORE USED



