

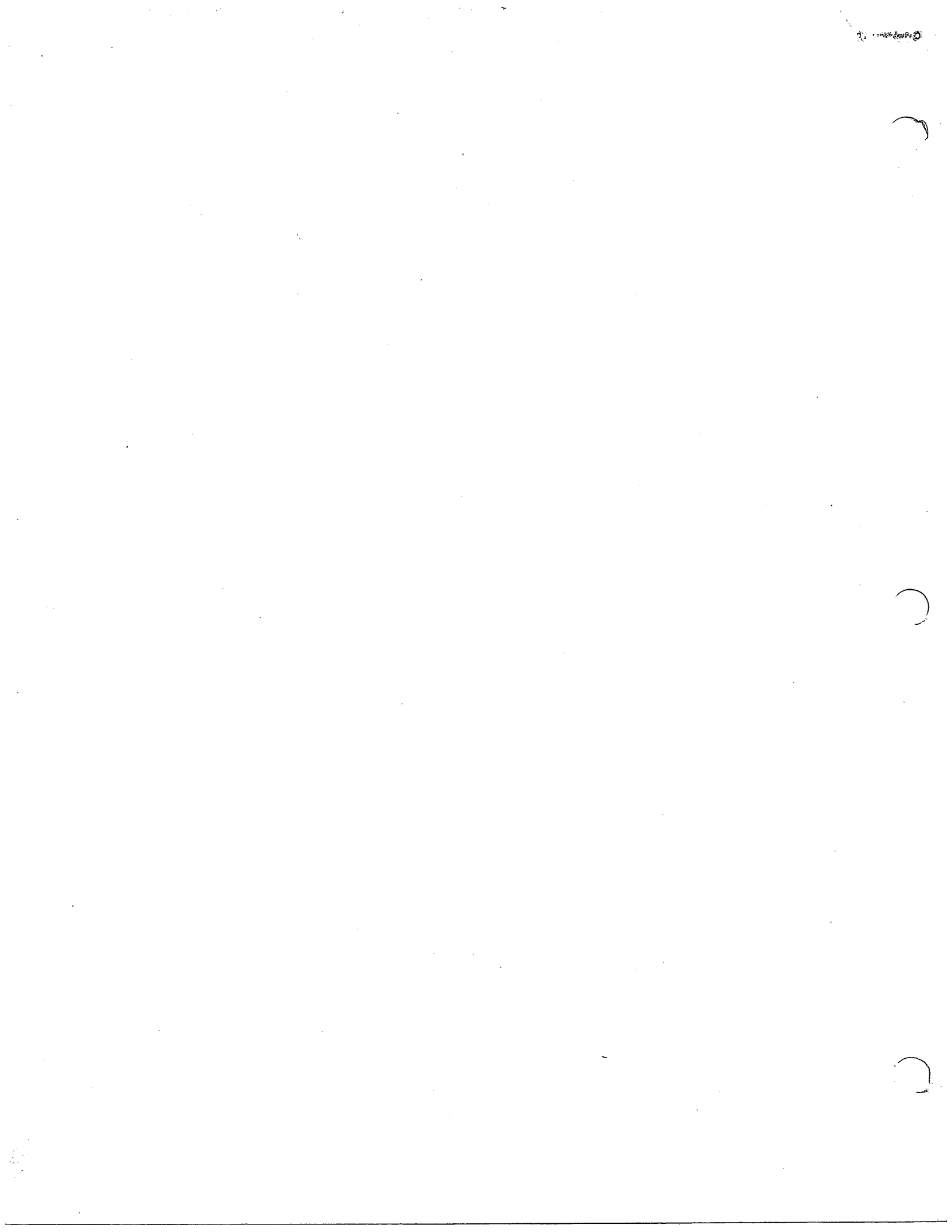
IDENTIFICATION

PRODUCT CODE: MAINDEC-08-DHDRA-A-D
PRODUCT NAME: DR8-EA 12 CHANNEL BUFFERED
DIGITAL INTERFACE
DATE: MARCH, 1972
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: MICHAEL DAVIS/P. COYNE
REPLACES: MAINDEC-8E-D/QB

COPYRIGHT © 1972
DIGITAL EQUIPMENT CORPORATION

ADVANCE COPY

THIS DESCRIPTION IS PRELIMINARY AND
SUBJECT TO CHANGE WITHOUT NOTICE.



1. ABSTRACT

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

PDP8E STANDARD COMPUTER WITH 4K OF CORE
ASR-33 TELETYPE (OR EQUIVALENT)
DR8-EA WITH TEST CABLE

2.2 STORAGE

THE PROGRAM USES LOCATION 0000-3377

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.
PRESS CONTINUE.

PROGRAM WILL RESPOND BY TYPING

"SET SR FOR INTERRUPT JUMPERS AND CONT" AND THEN HALT,
SET SWITCHES FOR ALL INPUT REGISTER BITS JUMPED 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 IN 4.2 OR 5.1
PRESS CONTINUE.

PROGRAM WILL BEGIN TEST EXECUTION

4.3.2 FOR STARTING ADDRESS 201

SET SWITCHES AS IN 4.2 OR 5.1
PRESS CLEAR AND CONTINUE.

PROGRAM WILL BEGIN TEST EXECUTION

OPERATING PROCEDURE

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

2. PROGRAM AND/OR OPERATOR ACTION

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),

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

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.

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.

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

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)
XXX= 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 (LOTS)

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

MISCELLANEOUS

1. EXECUTION TIME

EXECUTION TIME IS APPROXIMATELY 3 MINUTES FOR FULL ITERATION AND APPROXIMATELY 10 SECONDS WITH ITERATIONS SUPPRESSED.

PROGRAM DESCRIPTION

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

THE PROGRAM SEQUENCE IS AS FOLLOWS:

ALL BASIC IOT 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.

0. LISTING

0030 0000 TYPFLG: 0
0031 0000 LPCNT: 0

/ INDIRECT POINTERS

0032 0261 XDBDI; DBDIX
0033 0266 XDBEI; DBEIX
0034 0273 XDBSK; DBSKX
0035 0300 XDBCI; DBCIX
0036 0305 XDBRI; DBRIX
0037 0312 XDBCO; DBCOX
0040 0317 XDBSO; DBSOX
0041 0324 XDBRO; DBROX
0042 3200 XPRINT; PRINT
0043 3251 XTYPE; TYPE
0044 2600 XERROR; ERROR
0045 2667 XLOOP1; LOOP1
0046 2712 XLOOP2; LOOP2

/ TEST INITIALIZATION

0200	*200	JMP	START1	/CLEAR ALL FLAGS
0201		JMP	START2	/TYPE "SET SR FOR DEVICE
0202	START1;	CAF		/CODE AND CONT"
0203		JMS I	XPRINT	/HALT FOR SWITCHES
0204	M1=1	HLT		/GET SWITCHES
0205		LAS		/MASK DEVICE CODE
0206	7604	AND	(7	/POSITION BITS
0207	0377	CLL	RTL	
0210	7106	RAL	(6500	/GENERATE BASIC IOT
0211	7004	TAD	IOTS	/SAVE BASIC IOT
0212	1376	DCA	XPRINT	/TYPE "SET SR FOR JUMPERS
0213	3247	JMS I		/AND CONT"
0214	4442	M2=1		/HALT FOR SWITCHES
0215	3420	HLT		/GET SWITCHES
0216	7402	LAS		/SAVE JUMPER MASK
0217	7604	DCA	I JUMPER	
0220	3026	JMS I	XPRINT	
0221	4442	M2A=1		
0222	3444	HLT		
0223	7402	LAS	F JUMPER	
0224	7604	DCA	(=i0	/8 IOTS WILL BE
0225	3027	DCA	CNTR1	/SET UP
0226	1375	TAD	DCA	/STORE INSTRUCTION FOR
0227	3020	DCA	DIOT	/IOT SET UP
0230	1260	TAD	PNTR1	/GET IOT
0231	3233	DCA	IOTS	/STORE IT
0232	1247	TAD		/PREPARE TO STORE
0233	0000		PNTR1;	/NEXT IOT
0234	2233	ISE		
0235	2247	ISE		
0236	2020	ISE		

/MAINDEC=08=DHORA=A
 /DR8-EA
 /12 CHANNEL BUFFERED I/O DIAGNOSTIC
 /COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
 /STARTING ADDRESS1
 200-INPUT DEVICE CONFIGURATION
 201-USE STANDARD CONFIGURATION

/SWITCH REGISTER OPTIONS

/SR00 =1; SUPPRESS HALT ON ERROR
 /SR01 =1; SUPPRESS ERROR TYPEOUT
 /SR02 =1; LOOP ON CURRENT TEST
 /SR03 =1; LOOP WITH CURRENT DATA
 /SR04 =1; SUPPRESS BELL AT END OF PASS
 /SR05 =1; SUPPRESS ITERATIONS
 /SR06 =1; ESCAPE TO NEXT TEST ON ERROR

/INSTRUCTION DEFINITIONS

7421 MQL=7421
 7501 MQA=7501
 7002 BSW=7002
 6007 CAF=6007
 6003 SRQ=6003
 4432 DBDI=JMS I XDBDI
 4433 DBEI=JMS I XDBEI
 4434 DBSK=JMS I XDBSK
 4435 DBCI=JMS I XDBCI
 4436 DBRI=JMS I XDBRI
 4437 DBCO=JMS I XDBCO
 4440 DBSO=JMS I XDBSO
 4441 DBRO=JMS I XDBRO

/LOCATION EQUIVALENCIES

3026 MSTDGT=ERADR+1
 3027 LSTDGT=ERADR+2

/GENERAL VARIABLES

0010 *10
 0000 POINT1, 0
 0020 *20
 0000 CNTR1, 0
 0000 DATA1, 0
 0000 DATA2, 0
 0000 DATA3, 0
 0000 DATA4, 0
 0000 DATA5, 0
 7777 IJUMPE, 7777 /INTERRUPT JUMPER MASK
 7777 FJUMPE, 7777 /FLIPFLOP JUMPER MASK

/CLEAR TYPE FLAG

```

0237 5232 JMP PNTR1=1
0240 4442 JMS I XPRINT
0241 3475 M3=1
0242 7402 HLT
0243 7300 CLA CLL
0244 3030 START2, DCA TYPFLG
0245 5646 JMP I .=1
0246 0400 INTI

```

/TYPE "SET SR FOR RUN
/AND CONT
/HALT FOR SWITCHES

/CLEAR ERROR FLAG
/GO TO FIRST TEST

/INITIALIZATION CONSTANTS AND VARIABLES

```

0247 0000 /
0250 0262 /
0251 0267 /
0252 0274 /
0253 0301 /
0254 0306 /
0255 0313 /
0256 0320 /
0257 0325 /
0260 3650 /

```

/BASIC IOT

/IOT SUBROUTINES

/DISABLE DATA BUFFER INTERRUPT (DBDI,65X0)

```

0261 0000 DBDIX, 0
0262 6500 6500
2263 7410 SKP /TRAP FOR UNDESIRE0
2264 7402 HLT /SKIPS
2265 5661 JMP I DBDIX

```

/ENABLE DATA BUFFER INTERRUPTS (DBEI,65X1)

```

2266 0000 DBEIX, 0
2267 6501 6501
2270 7410 SKP /TRAP FOR UNDESIRE0
2271 7402 HLT /SKIPS
2272 5666 JMP I DBEIX

```

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

```

2273 0000 DBSKX, 0
2274 6502 6502
2275 7410 SKP
2276 2273 ISZ DBSKX
2277 5673 JMP I DBSKX

```

/08 TO INPUT REGISTER CORRESPONDING
/TO 15 IN AC (DBCI,65X3)

0300 0000 DB0IX, 0
 0301 6503 6503 /TRAP FOR UNDESIRE
 /SKIPS
 0302 7410 SKP
 0303 7402 HLT
 0304 5700 JMP I DB0IX

/ INPUT REGISTER TO AC (DBR1,65X4)
 /
 0305 0000 DBR1X, 0
 0306 6504 6504 /TRAP FOR UNDESIRE
 /SKIPS

0307 7410 SKP /TRAP FOR UNDESIRE
 0310 7402 HLT /SKIPS
 0311 5705 JMP I DBR1X

/ ZEROS TO OUTPUT REGISTER CORRESPONDING TO
 /ONES IN AC (DBCO,65X5)
 0312 0000 DBCOX, 0
 0313 6505 6505

0314 7410 SKP /TRAP FOR UNDESIRE
 0315 7402 HLT /SKIPS
 0316 5712 JMP I DBCOX

/ IS TO OUTPUT REGISTER CORRESPONDING
 /TO 1S IN AC (DBSO,65X6)
 0317 0000 DBSOX, 0
 0320 6506 6506

0321 7410 SKP /TRAP FOR UNDESIRE
 0322 7402 HLT /SKIPS
 0323 5717 JMP I DBSOX

/ JAM TRANSFER OUTPUT REGISTER TO AC (DBRO,65X7)
 /
 0324 0000 DBROX, 0
 0325 6507 6507

0326 7410 SKP /TRAP FOR UNDESIRE
 0327 7402 HLT /SKIPS
 0330 5724 JMP I DBROX

0375 7770
 0376 6500
 0377 0007
 0400 PAGE

/ IS OUTPUT REGISTER CLEARED BY INITIALIZE?
 /

```

0400 3030 INIT1, DCA TYPFLG /CLEAR ERROR FLAG
0401 3031 DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
0402 6007 CAF /INITIALIZE INTERFACE
0403 4441 DBRO /READ OUTPUT REGISTER
0404 3021 DCA DATA1 /SAVE REGISTER DATA
0405 1021 TAD DATA1 /GET REGISTER DATA
0406 7650 SNA CLA /WAS REGISTER CLEARED BY INITIALIZE
0407 5214 JMP ,*5 /DATA CORRECT, CONTINUE
0410 4444 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 4445 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 /INITIALIZE INTERFACE
0421 6007 CAF /READ INPUT REGISTER
0422 4436 DBRI /SAVE REGISTER DATA
0423 3021 DCA DATA1 /GET REGISTER DATA
0424 1021 TAD DATA1 /WAS REGISTER CLEARED
0425 7650 SNA CLA /DATA CORRECT, CONTINUE
0426 5233 JMP ,*5 /NO, ERROR
0427 4444 JMS I XERROR /"INPUT REGISTER NOT CLEARED"
0430 3661 INIT2E=1 /"REGISTER DATA"
0431 3515 DH1=1 /NUMBER OF WORDS TO BE OUTPUT
0432 7777 =1 /CHECK FOR LOOP ON CURRENT TEST
0433 4445 JMS I XLOOP1 /LOOP ON CURRENT TEST
0434 5221 JMP INIT2+3

```

```

/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 /INITIALIZE INTERFACE
0440 6007 CAF /ENABLE INTERFACE INTERRUPTS
0441 4433 DBEI /TEST FOR INTERRUPT ACTIVE
0442 6003 SRQ /NO INTERRUPT, CONTINUE
0443 5250 JMP ,*5 /INTERRUPT ACTIVE, ERROR
0444 4444 JMS I XERROR /"INTERRUPT ACTIVE"
0445 4042 INT1E=1 /NO DATA HEADER
0446 3514 DH0=1 /NO DATA
0447 0000 0 /IS INTERFACE FLAG SET
0450 4434 DBSK /FLAG NOT SET, CONTINUE
0451 5256 JMP ,*5 /FLAG SET, ERROR
0452 4444 JMS I XERROR /"SKIP FLAG SET"
0453 3674 INIT3E=1 /NO DATA HEADER
0454 3514 DH0=1 /NO DATA
0455 0000 0 /CHECK FOR LOOP ON CURRENT TEST
0456 4445 JMS I XLOOP1 /LOOP ON CURRENT TEST
0457 5240 JMP INIT3+3

```

```

0460 3030 TRAN1: /CLEAR ERROR FLAG
0461 3031 DCA /SET ITERATION COUNT TO 4096(DECIMAL)
0462 6007 DCA /INITIALIZE INTERFACE
0463 7340 CAF /SET AC =7777
0464 4441 CLA CLL CMA /READ OUTPUT REGISTER
0465 3021 DCA /SAVE AC CONTENTS
0466 1021 TAD DATA1 /GET AC CONTENTS
0467 7650 SNA CLA /WAS AC CLEARED BY TRANSFER
0470 5275 JMP *5 /DATA CORRECT, CONTINUE
0471 4444 JMS I XERROR /NO, ERROR
0472 3703 TRAN1E=1 /"DBRO DID NOT CLEAR AC"
0473 3525 DH2=1 /"AC CONTENTS"
0474 7777 *1 /NUMBER OF WORDS TO BE OUTPUT
0475 4445 JMP /CHECK FOR LOOP ON CURRENT TEST
0476 5262 JMP /LOOP ON CURRENT TEST

```

```

/DOES OUTPUT REGISTER JAM TRANSFER TO AC?
/
DCA /CLEAR ERROR FLAG
DCA /SET ITERATION COUNT TO 4096(DECIMAL)
CAF /INITIALIZE INTERFACE
CLA CLL CMA /SET AC =7777
DBRI /READ INPUT REGISTER
DCA /SAVE AC CONTENTS
TAD DATA1 /GET AC CONTENTS
SNA CLA /WAS AC CLEARED BY TRANSFER
JMP *5 /DATA CORRECT, CONTINUE
JMS I XERROR /NO, ERROR
TRAN2E=1 /"DBRO DID NOT CLEAR AC"
DH2=1 /"AC CONTENTS"
*1 /NUMBER OF WORDS TO BE OUTPUT
JMS I /CHECK FOR LOOP ON CURRENT TEST
JMP /LOOP ON CURRENT TEST

```

```

/DOES INPUT REGISTER JAM TRANSFER TO AC
/
DCA /CLEAR ERROR FLAG
DCA /SET ITERATION COUNT TO 4096(DECIMAL)
CAF /INITIALIZE INTERFACE
CLA CLL CMA /SET AC =7777
DBRI /READ INPUT REGISTER
DCA /SAVE AC CONTENTS
TAD DATA1 /GET AC CONTENTS
SNA CLA /WAS AC CLEARED BY TRANSFER
JMP *5 /DATA CORRECT, CONTINUE
JMS I XERROR /NO, ERROR
TRAN2E=1 /"DBRO DID NOT CLEAR AC"
DH2=1 /"AC CONTENTS"
*1 /NUMBER OF WORDS TO BE OUTPUT
JMS I /CHECK FOR LOOP ON CURRENT TEST
JMP /LOOP ON CURRENT TEST

```

```

/DOES TRANSFER TO OUTPUT REGISTER CHANGE AC (WITH AC=7777,DBSO)
/
DCA /CLEAR ERROR FLAG
DCA /SET ITERATION COUNT TO 4096(DECIMAL)
CAF /INITIALIZE INTERFACE
CLA CLL CMA /SET AC =7777
DBSO /BIT SET OUTPUT REGISTER
DCA /SAVE AC CONTENTS
TAD DATA1 /GET AC CONTENTS
CMA CLA /COMPLEMENT DATA TO TEST FOR 7777
SNA CLA /DID AC CHANGE
JMP *5 /DATA CORRECT, CONTINUE
JMS I XERROR /NO, ERROR
TRAN3E=1 /"DBSO CHANGED AC"
DH2=1 /"AC CONTENTS"
*1 /NUMBER OF WORDS TO BE OUTPUT
JMS I /CHECK FOR LOOP ON CURRENT TEST
JMP /LOOP ON CURRENT TEST

```

/DOES TRANSFER TO OUTPUT REGISTER CHANGE AC (WITH AC=7777, DBCO)

```

TRAN4,
0536 /CLEAR ERROR FLAG
0537 /SET ITERATION COUNT TO 4096(DECIMAL)
0540 /INITIALIZE INTERFACE
0541 /SET AC =7777
0542 /BIT CLEAR OUTPUT REGISTER
0543 /SAVE AC CONTENTS
0544 /GET AC CONTENTS
0545 /COMPLIMENT DATA TO TEST FOR 7777
0546 /DID AC CHANGE
0547 /NO, ERROR
0548 /"DBCO CHANGED AC"
0551 /"AC CONTENTS"
0552 /NUMBER OF WORDS TO BE OUTPUT
0553 /CHECK FOR LOOP ON CURRENT TEST
0554 /LOOP ON CURRENT TEST
0555 JMP

```

/DOES TRANSFER TO INPUT REGISTER CHANGE AC (WITH AC=7777, DBCI)

```

TRAN5,
0556 /CLEAR ERROR FLAG
0557 /SET ITERATION COUNT TO 4096(DECIMAL)
0560 /INITIALIZE INTERFACE
0561 /SET AC =7777
0562 /BIT CLEAR INPUT REGISTER
0563 /SAVE AC CONTENTS
0564 /GET AC CONTENTS
0565 /COMPLIMENT DATA TO TEST FOR 7777
0566 /DID AC CHANGE
0567 /DATA CORRECT, CONTINUE
0570 /NO, ERROR
0571 /"DBCI CHANGED AC"
0572 /"AC CONTENTS"
0573 /NUMBER OF WORDS TO BE OUTPUT
0574 /CHECK FOR LOOP ON CURRENT TEST
0575 /LOOP ON CURRENT TEST
0576 /GO TO NEXT TEST
0577 JMP

```

PAGE

/DOES TRANSFER TO OUTPUT REGISTER CHANGE AC (WITH AC=0, DBSO)

```

TRAN6,
2620 /CLEAR ERROR FLAG
2621 /SET ITERATION COUNT TO 4096 (DECIMAL)
2622 /INITIALIZE INTERFACE
2623 /BIT SET OUTPUT REGISTER
2624 /SAVE AC CONTENTS
2625 /GET AC CONTENTS
2626 /WAS AC CHANGED
2627 /DATA CORRECT, CONTINUE

```

```

0610 4444 /AC CHANGED, ERROR
0611 3731 /"DBSO CHANGED AC"
0612 3525 /"AC CONTENTS"
0613 7777 /NUMBER OF WORDS TO BE OUTPUT
0614 4445 /CHECK FOR LOOP ON CURRENT TEST
0615 5202 /LOOP ON CURRENT TEST

/
/DOES TRANSFER TO OUTPUT REGISTER CHANGE AC (WITH AC=0,DBC0)
/
TRAN7, DCA TYPFLG /CLEAR ERROR FLAG
DCA LPCNT /SET ITERATION COUNT TO 4096 (DECIMAL)
CAF /INITIALIZE INTERFACE
DBC0 /BIT CLEAR OUTPUT REGISTER
DCA DATA1 /SAVE AC CONTENTS
TAD DATA1 /GET AC CONTENTS
SNA CLA *5 /IS AC STILL 0
JMP *5 /DATA CORRECT, CONTINUE
JMS I XERROR /NO, ERROR
TRAN4E=1 /"DBCO CHANGED AC"
DH2=1 /"AC CONTENTS"
=1 /NUMBER OF WORDS TO BE OUTPUT
JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
JMP TRAN7+2 /LOOP ON CURRENT TEST
/DOES TRANSFER TO INPUT REGISTER CHANGE AC (WITH AC=0, DBCI)
/

```

```

0634 3030 /CLEAR ERROR FLAG
0635 3031 /SET ITERATION COUNT TO 4096 (DECIMAL)
0636 6007 /INITIALIZE INTERFACE
0637 4435 /BIT CLEAR INPUT REGISTER
0640 3021 /SAVE AC CONTENTS
0641 1021 /GET AC CONTENTS
0642 7650 /IS AC STILL 0
0643 5250 /DATA CORRECT, CONTINUE
0644 4444 /AC CHANGED, ERROR
0645 3751 /"DBCI CHANGED AC"
0646 3525 /"AC CONTENTS"
0647 7777 /NUMBER OF WORDS TO BE OUTPUTED
0650 4445 /CHECK FOR LOOP ON CURRENT TEST
0651 5236 /LOOP ON CURRENT TEST
0652 5777 /GO TO NEXT TEST
0777 1000
1000 PAGE

```

```

/DOES TRANSFER TO INPUT REGISTER CHANGE AC (WITH AC=0, DBCI)
/
TRAN8, DCA TYPFLG /CLEAR ERROR FLAG
DCA LPCNT /SET ITERATION COUNT TO 4096 (DECIMAL)
CAF /INITIALIZE INTERFACE
DBC1 /BIT CLEAR INPUT REGISTER
DCA DATA1 /SAVE AC CONTENTS
TAD DATA1 /GET AC CONTENTS
SNA CLA *5 /IS AC STILL 0
JMP *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

```

```

OUT1, DCA TYPFLG /CLEAR ERROR FLAG
DCA LPCNT /SET ITERATION COUNT TO 4096 (DECIMAL)
CAF /INITIALIZE INTERFACE
CLA CLL CMA /SET AC =7777
DBSO /BIT SET OUTPUT REGISTER
DBRO /READ OUTPUT REGISTER
DCA DATA1 /SAVE REGISTER DATA

```

```

1007 1021 TAD DATA /GET REGISTER DATA
1010 7040 CMA CLA /COMPLIMENT DATA TO TEST FOR 7777
1011 7650 SNA CLA /IS REGISTER=7777
1012 5217 JMP I *5 /DATA CORRECT, CONTINUE
1013 4444 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 6007 CAF /INITIALIZE INTERFACE
1020 4441 DBRO /READ OUTPUT REGISTER
1021 3021 DCA DATA /SAVE REGISTER DATA
1022 1021 TAD DATA /GET REGISTER DATA
1023 7650 SNA CLA /WAS REGISTER CLEARED
1024 5231 JMP I *5 /DATA CORRECT, CONTINUE
1025 4444 JMS I XERROR /NO, ERROR
1026 3645 INIT1E=1 /"OUTPUT REG NOT CLEARED"
1027 3515 DH1=1 /"REGISTER DATA"
1030 7777 =1 /NUMBER OF WORDS TO BE OUTPUT
1031 4445 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)

```

OUT2,
1033 3030 DCA TYPFLG /CLEAR ERROR FLAG
1034 3031 DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
1035 7340 CLA CLL CMA /SET AC =7777
1036 4440 DBSO /BIT SET OUTPUT REGISTER
1037 4437 DBCO /BIT CLEAR OUTPUT REGISTER
1040 7300 CLA CLL /READ OUTPUT REGISTER
1041 4441 DBRO /SAVE REGISTER DATA
1042 3021 DCA DATA /GET REGISTER DATA
1043 1021 TAD DATA /WAS OUTPUT REGISTER CLEARED
1044 7650 SNA CLA /DATA CORRECT, CONTINUE
1045 5252 JMP I *5 /NO, ERROR
1046 4444 JMS I XERROR /"OUTPUT REGISTER NOT CLEARED"
1047 3645 INIT1E=1 /"REGISTER DATA"
1050 3515 DH1=1 /NUMBER OF WORDS TO BE OUTPUT
1051 7777 =1 /CHECK FOR LOOP ON CURRENT TEST
1052 4445 JMS I XLOOP1 /LOOP ON CURRENT TEST
1053 5235 JMP OUT2+2

```

/CAN EACH BIT OF OUTPUT REGISTER BE SET INDEPENDENTLY (DBSO)

```

OUT3,
1054 3030 DCA TYPFLG /CLEAR ERROR FLAG
1055 1177 TAD C=62 /SET ITERATION COUNT
1056 3031 DCA LPCNT /TO 50(DECIMAL)
1057 3021 DCA DATA /CLEAR TEST DATA
1060 6007 CAF /INITIALIZE INTERFACE
OUT3A,
1061 1021 TAD DATA /GET TEST DATA
1062 4440 DBRO /BIT SET OUTPUT REGISTER
1063 7300 CLA CLL /READ OUTPUT REGISTER
1064 4441 DBRO /SAVE REGISTER DATA
1065 3022 DCA DATA

```



```

1066 1021 TAD DATA1 /GET TEST DATA
1067 7041 CIA
1070 1022 TAD DATA2 /COMPARE TO REGISTER CONTENTS
1071 7650 SNA CLA /DO THEY COMPARE
1072 5277 JMP :+5 /DATA CORRECT, CONTINUE
1073 4444 XERROR /NO, ERROR
1074 3761 OUT1E=1 /"DBSO ERROR"
1075 3560 DH4=1 /"EXPECTED RECEIVED"
1076 7776 *2 /NUMBER OF WORDS TO BE OUTPUT
1077 4446 JMS I XLOOP2 /TEST FOR LOOP ON SAME DATA;ESCAPE ON DATA ERROR
1100 5260 JMP OUT3A /LOOP WITH SAME DATA
1101 2021 ISZ DATA1 /INCREMENT DATA PATTERN
1102 5260 JMP OUT3A /CONTINUE TEST
1103 4445 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
1104 5257 JMP OUT3+3 /LOOP ON CURRENT TEST
/
/CAN EACH BIT OF OUTPUT REGISTER BE CLEARED
/INDEPENDENTLY (DBCO)
/

```

```

1105 3030 DCA TYPFLG /CLEAR ERROR FLAG
1106 1177 TAD C=62 /SET ITERATION COUNT
1107 3031 DCA LPCNT /TO 50(DECIMAL)
1110 3021 DCA DATA1 /CLEAR TEST DATA
1111 6007 CAF /INITIALIZE INTERFACE
1112 1021 TAD DATA1 /GET MASK
1113 7040 CMA /COMPLIMENT TO GET EXPECTED RESULT
1114 3022 DCA DATA2 /SAVE EXPECTED RESULT
1115 7040 CMA /SET OUTPUT REGISTER TO 7777
1116 4440 DBSO /BIT SET OUTPUT REGISTER
1117 7300 CLA CLL
1120 1021 TAD DATA1 /GET PATTERN TO CLEAR OUTPUT REGISTER
1121 4437 DBCO /BIT CLEAR OUTPUT REGISTER
1122 4441 DBRO /READ OUTPUT REGISTER
1123 3023 DCA DATAS /SAVE REGISTER DATA
1124 1022 TAD DATA2 /GET EXPECTED RESULT
1125 7041 CIA
1126 1023 TAD DATA3 /COMPARE TO RECEIVED DATA
1127 7650 SNA CLA /WERE CORRECT BITS IN OUTPUT REGISTER CLEARED
1130 5335 JMP :+5 /DATA CORRECT, CONTINUE
1131 4444 JMS I XERROR /NO, ERROR
1132 3767 OUT4E=1 /"DBCO ERROR"
1133 3534 DH3=1 /MASK EXPECTED RECEIVED
1134 7775 *3 /NUMBER OF WORDS TO BE OUTPUT
1135 4446 JMS I XLOOP2 /TEST FOR LOOP ON SAME DATA;ESCAPE ON DATA ERROR
1136 5311 JMP OUT4A /LOOP WITH SAME DATA
1137 2021 ISZ DATA1 /INCREMENT DATA PATTERN
1140 5311 JMP OUT4A /CONTINUE TEST
1141 4445 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
1142 5310 JMP OUT4+3 /LOOP ON CURRENT TEST
1143 5777 JMP OUT5
1177 1200

```

```

/ WITH OUTPUT REGISTER CLEARED, DOES CLEARING
/ OUTPUT REGISTER CHANGE ANY BIT IN OUTPUT

```

```

1200 3030 /
1201 1177 /CLEAR ERROR FLAG
1202 3031 /SET ITERATION COUNT
1203 3021 /TO 50 (DECIMAL)
1204 3022 /CLEAR TEST DATA
1205 6007 /CLEAR EXPECTED RESULT
1206 1021 /INITIALIZE INTERFACE
1207 4437 /GET TEST DATA
1210 7300 /BIT CLEAR OUTPUT REGISTER
1211 4441 /
1212 3023 /READ OUTPUT REGISTER
1213 1023 /SAVE REGISTER DATA
1214 7650 /GET REGISTER DATA
1215 5222 /IS OUTPUT REGISTER 0
1216 4444 /DATA CORRECT: CONTINUE
1217 3767 /NO, ERROR
1220 3534 /"DBCO ERROR"
1221 7775 /"MASK EXPECTED RECEIVED"
1222 4446 /NUMBER OF DATA WORDS
1223 5205 /TEST FOR LOOP ON SAME DATA
1224 2021 /LOOP WITH SAME DATA
1225 5205 /INCREMENT DATA PATTERN
1226 4445 /CONTINUE
1227 5203 /CHECK FOR LOOP ON CURRENT TEST
      /LOOP ON CURRENT TEST

```

```

1230 3030 /DOES SETTING OUTPUT REGISTER TWICE WITH SAME
1231 1177 /DATA CHANGE OUTPUT REGISTER
1232 3031 /
1233 3021 /CLEAR ERROR FLAG
1234 6007 /SET ITERATION COUNT
1235 1021 /TO 50 (DECIMAL)
1236 4440 /CLEAR TEST DATA
1237 4440 /INITIALIZE INTERFACE
1240 7300 /GET TEST DATA
1241 4441 /BIT SET OUTPUT REGISTER
1242 3022 /SAVE REGISTER DATA
1243 1021 /COMPARE TO REGISTER DATA
1244 7041 /ARE THEY THE SAME
1245 1022 /DATA CORRECT: CONTINUE
1246 7650 /NO, ERROR
1247 5254 /"DBSO ERROR"
1250 4444 /"EXPECTED RECEIVED"
1251 3767 /NUMBER OF DATA WORDS
1252 3560 /TEST FOR LOOP ON SAME DATA
1253 7776 /LOOP WITH SAME DATA
1254 4446 /INCREMENT DATA PATTERN
1255 5234 /CONTINUE
1256 2021 /CHECK FOR LOOP ON CURRENT TEST
1257 5234 /
1260 4445 /

```

1261 5233

JMP OUT6+3 /LOOP ON CURRENT TEST

/DOES READING OUTPUT REGISTER TWICE CHANGE
/OUTPUT REGISTER

OUT7:

1262 3030 /CLEAR ERROR FLAG
 1263 1177 /SET ITERATION COUNT
 1264 3031 /TO 50 (DECIMAL)
 1265 3021 /CLEAR TEST DATA
 1266 6007 /INITIALIZE INTERFACE
 1267 1021 /GET TEST DATA
 1270 4440 /BIT SET OUTPUT REGISTER

OUT7A:

1271 7300 /READ OUTPUT REGISTER
 1272 4441 /READ OUTPUT REGISTER
 1273 7300 /SAVE REGISTER DATA
 1274 4441 /GET TEST DATA

1275 3022 /COMPARE TO REGISTER DATA
 1276 1021 /ARE THEY THE SAME
 1277 7041 /DATA CORRECT, CONTINUE
 1300 1022 /NO, ERROR
 1301 7650 /"DBRO ERROR"
 1302 5307 /"EXPECTED RECEIVED"
 1303 4444 /NUMBER OF DATA WORDS
 1304 3775 /TEST FOR LOOP WITH SAME DATA
 1305 3560 /LOOP WITH SAME DATA
 1306 7776 /INCREMENT DATA PATTERN
 1307 4446 /CONTINUE
 1310 5266 /CHECK FOR LOOP ON CURRENT TEST
 1311 2021 /LOOP ON CURRENT TEST
 1312 5266
 1313 4445
 1314 5265

/DOES CLEARING OUTPUT REGISTER TWICE
/CHANGE ANY BIT IN OUTPUT REGISTER

OUT8:

1315 3030 /CLEAR ERROR FLAG
 1316 1177 /SET ITERATION COUNT
 1317 3031 /TO 50 (DECIMAL)
 1320 3021 /CLEAR TEST DATA
 1321 3022 /INITIALIZE INTERFACE
 1322 6007 /GET TEST DATA
 1323 1021 /BIT SET OUTPUT REGISTER
 1324 4440 /BIT CLEAR OUTPUT REGISTER
 1325 4437 /BIT CLEAR OUTPUT REGISTER

OUT8A:

1326 4437 /READ OUTPUT REGISTER
 1327 7300 /SAVE REGISTER DATA
 1330 4441 /GET REGISTER DATA
 1331 3023 /IS REGISTER 0
 1332 1023 /DATA CORRECT, CONTINUE
 1333 7650 /NO, ERROR
 1334 5341
 1335 4444

```

1336 3767 OUT4E=1
1337 3534 DH3=1
1340 7775 -3
1341 4446 JMS I XLOOP2
1342 5322 JMP OUT8A
1343 2021 ISZ DATA1
1344 5322 JMP OUT8A
1345 4445 JMS I XLOOP1
1346 5320 JMP OUT8+3
1347 5777 JMP IN1
1377 1400
1400 1400

```

PAGE

```

/ CAN ALL BITS IN INPUT REGISTER BE SET
/ DOES INITIALIZE CLEAR INPUT REGISTER
/
IN1, DCA TYPFLG /CLEAR ERROR FLAG
DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
CAF /INITIALIZE INTERFACE
CLA CLL CMA /SET AC =7777
DBSO /BIT SET OUTPUT REGISTER
CLA CLL
DBRI /READ INPUT REGISTER
DCA DATA1 /SAVE REGISTER DATA
TAD DATA1 /GET REGISTER DATA
CMA CLA /COMPLIMENT TO TEST FOR 7777
SNA CLA /WAS INPUT REGISTER SET TO 7777
JMP ,+5 /DATA CORRECT, CONTINUE
JMS I XERROR /NO, ERROR
INZE-1 /"INPUT REGISTER NOT CORRECT"
DH1=1 /"REGISTER DATA"
*1 /NUMBER OF WORDS TO BE OUTPUT
CAF /INITIALIZE INTERFACE
DBRI /READ INPUT REGISTER
DCA DATA1 /SAVE REGISTER DATA
TAD DATA1
SNA CLA /DATA CORRECT, CONTINUE
JMP ,+5 /NO, ERROR
JMS I XERROR /"INPUT REGISTER NOT CLEARED"
INITZE=1 /"REGISTER DATA"
DH1=1 /NUMBER OF WORDS TO BE OUTPUT
*1 /CHECK FOR LOOP ON CURRENT TEST
JMS I XLOOP1 /LOOP ON CURRENT TEST
JMP IN1A

```

```

/ CAN ALL BITS IN INPUT REGISTER BE CLEARED (DBCI)
/
IN2, DCA TYPFLG /CLEAR ERROR FLAG
DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
CAF /INITIALIZE INTERFACE
CLA CLL CMA /SET AC =7777
AND FJUMPER /MASK TO TEST ONLY FLIPFLOP BITS
DCA DATA1 /SAVE MASK

```

```

1442 3022 /SAVE EXPECTED RESULT
1443 1021 /GET MASK
1444 4440 /BIT SET OUTPUT REGISTER
1445 4435 /BIT CLEAR INPUT REGISTER
1446 7300
1447 4436 /READ INPUT REGISTER
1450 3023 /SAVE REGISTER DATA
1451 1023 /COMPARE TO REGISTER DATA
1452 7650 /WERE CORRECT BITS CLEARED
1453 5260 /DATA CORRECT, CONTINUE
1454 4444 /NO, ERROR
1455 4003 /"DBCI ERROR"
1456 3534 /"MASK EXPECTED RECEIVED"
1457 7775 /NUMBER OF WORDS TO BE OUTPUT
1460 4445 /CHECK FOR LOOP ON CURRENT TEST
1461 5236 /LOOP ON CURRENT TEST

```

```

/ CAN EACH BIT OF INPUT REGISTER BE SET INDEPENDENTLY

```

```

IN3, 1462 3030 /CLEAR ERROR FLAG
1463 1177 /SET ITERATION COUNT
1464 3031 /TO 50(DECIMAL)
IN3A, 1465 3021 /CLEAR TEST DATA
1466 6007 /INITIALIZE INTERFACE
1467 1021 /GET TEST DATA
1470 4440 /BIT SET OUTPUT REGISTER
1471 7300
1472 4436 /READ INPUT REGISTER
1473 3022 /SAVE REGISTER DATA
1474 1021 /GET TEST DATA
1475 7041
1476 1022 /COMPARE TO RECEIVED DATA
1477 7650 /ARE THEY THE SAME
1500 5305 /DATA CORRECT, CONTINUE
1501 4444 /NO, ERROR
1502 4011 /"INPUT REGISTER DATA ERROR"
1503 3560 /"EXPECTED RECEIVED"
1504 7776 /NUMBER OF WORDS TO BE OUTPUT
1505 4446 /TEST FOR LOOP ON SAME DATA, ESCAPE ON DATA ERROR
1506 5266 /LOOP WITH SAME DATA
1507 2021 /INCREMENT DATA PATTERN
1510 5266 /CONTINUE TEST
1511 4445 /CHECK FOR LOOP ON CURRENT TEST
1512 5265 /LOOP ON CURRENT TEST
1513 5777
1577 1600
PAGE 1600

```

```

/ VERIFY THAT ALL LATCHING INPUT LINES HOLD DATA

```

```

IN5, 1600 3030 /CLEAR ERROR FLAG
1601 1177 /SET ITERATION COUNT

```

```

1602 3031 DCA /TO 50(DECIMAL)
1603 3023 DCA /CLEAR TEST DATA
1604 6007 CAF /INITIALIZE INTERFACE
1605 1023 TAD /GET TEST DATA
1606 0027 AND /MASK OFF NON LATCHING BITS
1607 3021 DCA /SAVE AS EXPECTED RESULT
1610 1021 TAD /GET TEST DATA
1611 7450 SNA /ARE ANY BITS TO BE TESTED
1612 5233 JMP /NO, GET NEXT DATA WORD
1613 4440 DBSO /BIT SET OUTPUT REGISTER
1614 4437 DBCO /BIT CLEAR OUTPUT REGISTER
1615 7300 CLA CLL
1616 4436 DBRI
1617 3022 DCA /READ INPUT REGISTER
1620 1021 TAD /SAVE REGISTER DATA
1621 7041 CIA /GET EXPECTED RESULT
1622 1022 TAD /COMPARE TO RECEIVED DATA
1623 7650 SNA CLA /ARE THEY THE SAME
1624 5231 JMP /DATA CORRECT, CONTINUE
1625 4444 JMS I /NO, ERROR
1626 4026 IN4E-1 /"LATCH ERROR"
1627 3560 DH4-1 /"EXPECTED RECEIVED"
1630 7776 -2 /NUMBER OF WORDS TO BE OUTPUT
1631 4446 JMS I /TEST FOR LOOP ON SAME DATA, ESCAPE ON DATA ERROR
1632 5204 JMP /LOOP WITH SAME DATA
1633 2023 ISE /INCREMENT DATA PATTERN
1634 5204 JMP I /CONTINUE TEST
1635 4445 JMS I /CHECK FOR LOOP ON CURRENT TEST
1636 5203 JMP /LOOP ON CURRENT TEST

```

```

/VERIFY ALL NON LATCHING DATA BITS DO NOT HOLD DATA
/
IN6,
1637 3030 DCA /CLEAR ERROR FLAG
1640 1177 TAD /SET ITERATION COUNT
1641 3031 DCA /TO 50(DECIMAL)
1642 3024 DCA /CLEAR TEST DATA
1643 3022 DCA /CLEAR EXPECTED RESULT
1644 6007 CAF /INITIALIZE INTERFACE
1645 1027 TAD /GET MASK FOR NON LATCHING BITS
1646 7040 CMA /CHANGE TO MASK OFF LATCHING BITS
1647 0024 AND /SAVE FOR TRANSMISSION
1650 3021 DCA /GET TEST DATA
1651 1021 TAD /ARE ANY BITS TO BE TESTED
1652 7450 SNA /NO GET NEXT DATA WORD
1653 5272 JMP /BIT SET OUTPUT REGISTER
1654 4440 DBSO /BIT CLEAR OUTPUT REGISTER
1655 4437 DBCO
1656 7300 CLA CLL
1657 4436 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 4444 JMS I /YES, ERROR

```

```

1665 4026
1666 3534
1667 7775
1670 4446
1671 5244
1672 2024
1673 5244
1674 4445
1675 5242

IN6C,
XLOOP2
IN6A
DATA4
IN6A
XLOOP1
IN6+3

/"LATCH 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

```

/VERIFY THAT ALL LATCHING LINES CAN BE CLEARED INDEPENDENTLY

```

IN7,
1676 3030 /DCA /CLEAR ERROR FLAG
1677 1177 /TAD /SET ITERATION COUNT
1700 3031 /DCA /TO 50(DECIMAL)
1701 3024 /DCA /CLEAR TEST
1702 6007 /CAF /INITIALIZE INTERFACE
1703 1027 /TAD /GET MASK FOR LATCHING BITS
1704 0024 /AND /MASK OFF NON LATCHING BITS
1705 3021 /DCA /SAVE FOR TRANSMISSION
1706 3022 /DCA /EXPECTED RESULT
1707 1021 /TAD /SET OUTPUT REGISTER=7777
1710 4440 /DBSO /BIT SET OUTPUT REGISTER
1711 4437 /DBCO /BIT CLEAR OUTPUT REGISTER
1712 7300 /CLA CLL
1713 1021 /TAD /GET TEST DATA
1714 4435 /DBCI /BIT CLEAR INPUT REGISTER
1715 7300 /CLA CLL
1716 4436 /DBRI
1717 3023 /DCA /READ INPUT REGISTER
1720 1023 /TAD /SAVE REGISTER DATA
1721 7650 /SNA CLA /COMPARE TO RECEIVED DATA
1722 5327 /JMP /ARE THEY THE SAME
1723 4444 /JMS I /DATA CORRECT, CONTINUE
1724 4026 /IN4E-1 /NO. ERROR
1725 3534 /DH3-1 /"LATCH ERROR"
1726 7775 /-3 /"MASK EXPECTED RECEIVED"
1727 4446 /JMS I /NUMBER OF WORDS TO BE OUTPUT
1730 5302 /JMP /TEST FOR LOOP ON SAME DATA,ESCAPE ON DATA ERROR
1731 2024 /ISZ /LOOP WITH SAME DATA
1732 5302 /JMP /INCREMENT DATA PATTERN
1733 4445 /JMS I /CONTINUE TEST
1734 5301 /JMP /CHECK FOR LOOP ON CURRENT TEST
1735 5777 /IN7C, /LOOP ON CURRENT TEST
1777 2000 /GO TO NEXT TEST
2000 2000 /PAGE

```

/WITH THE INPUT REGISTER CLEARED, DOES CLEARING /THE INPUT REGISTER SET ANY BIT IN INPUT

```

IN8,
2000 3030 /DCA /CLEAR ERROR FLAG
2001 1177 /TAD /SET ITERATION COUNT

```

```

2002 3031 DCA /TO 50 (DECIMAL)
2003 3021 DCA /CLEAR TEST DATA
2004 3022 DCA /CLEAR EXPECTED RESULT
2005 6007 CAF /INITIALIZE INTERFACE
2006 1021 TAD /GET TEST DATA
2007 4035 DBCI /BIT CLEAR INPUT REGISTER
2010 7300 CLA CLL
2011 4436 DBRI
2012 3023 DCA
2013 1023 TAD
2014 7650 SNA CLA
2015 5222 JMP
2016 4444 JMS I
2017 4003 IN2E-1
2020 3534 DH3-1
2021 7775 *3
2022 4446 JMS I
2023 5205 JMP
2024 2021 ISZ
2025 5205 JMP I
2026 4445 JMP
2027 5203 JMP

/READ INPUT REGISTER
/SAVE REGISTER DATA
/GET REGISTER DATA
/IS INPUT REGISTER 0
/DATA CORRECT, CONTINUE
/NO. ERROR
/"DBCI ERROR"
/"MASK EXPECTED RECEIVED"
/NUMBER OF DATA WORDS
/TEST FOR LOOP WITH SAME DATA
/LOOP WITH SAME DATA
/INCREMENT DATA PATTERN
/CONTINUE
/CHECK FOR LOOP ON CURRENT TEST
/LOOP ON CURRENT TEST

```

```

/DOES READING THE INPUT REGISTER
/CHANGE THE INPUT REGISTER
/

```

```

2030 3030 DCA /CLEAR ERROR FLAG
2031 1177 TAD /SET ITERATION COUNT
2032 3031 DCA /TO 50 (DECIMAL)
2033 3021 DCA /CLEAR TEST DATA
2034 6007 CAF /INITIALIZE INTERFACE
2035 1021 TAD /GET TEST DATA
2036 4440 DBSO /BIT SET OUTPUT REGISTER
2037 7300 CLA CLL
2040 4436 DBRI
2041 7300 CLA CLL
2042 4436 DBRI
2043 3022 DCA
2044 1021 TAD
2045 7041 CIA
2046 1022 TAD
2047 7650 SNA CLA
2050 5255 JMP
2051 4444 JMS I
2052 4034 IN9E-1
2053 3560 DH4-1
2054 7776 *2
2055 4446 JMS I
2056 5234 JMP
2057 1021 ISZ
2060 5234 JMP
2061 4445 JMS I
2062 5233 JMP

/READ INPUT REGISTER
/READ INPUT REGISTER
/SAVE REGISTER DATA
/GET TEST DATA
/COMPARE TO REGISTER DATA
/ARE THEY THE SAME
/DATA CORRECT, CONTINUE
/NO. ERROR
/"DBCI ERROR"
/"EXPECTED RECEIVED"
/NUMBER OF DATA WORDS
/TEST FOR LOOP WITH SAME DATA
/LOOP WITH SAME DATA
/INCREMENT DATA PATTERN
/CONTINUE
/CHECK FOR LOOP ON CURRENT TEST
/LOOP ON CURRENT TEST

```



```

/DOES CLEARING INPUT REGISTER TWICE SET ANY BIT
/IN INPUT REGISTER
/

```

```

2063 3030 /CLEAR ERROR FLAG
2064 1177 /SET ITERATION COUNT
2065 3031 /TO 50 (DECIMAL)
2066 3022 /CLEAR EXPECTED RESULT
2067 3021 /CLEAR TEST DATA
2070 6007 /INITIALIZE INTERFACE
2071 1021 /GET TEST DATA
2072 4440 /BIT SET OUTPUT REGISTER
2073 4437 /BIT CLEAR OUTPUT REGISTER
2074 4435 /BIT CLEAR INPUT REGISTER
2075 4435 /BIT CLEAR INPUT REGISTER
2076 7300 /READ INPUT REGISTER
2077 4436 /SAVE REGISTER DATA
2100 3023 /GET REGISTER DATA
2101 1023 /IS INPUT REGISTER 0
2102 7650 /DATA CORRECT, CONTINUE
2103 5310 /NO, ERROR
2104 4444 /"DBCI ERROR"
2105 4003 /"MASK EXPECTED RECEIVED"
2106 3534 /NUMBER OF DATA WORDS
2107 7775 /TEST FOR LOOP WITH SAME DATA
2110 4446 /LOOP WITH SAME DATA
2111 5270 /INCREMENT DATA PATTERN
2112 2021 /CONTINUE
2113 5270 /CHECK FOR LOOP ON CURRENT TEST
2114 4445 /LOOP ON CURRENT TEST
2115 5266 /GO TO NEXT TEST
2116 5777 /
2177 2200 /
2200 2200 /

```

PAGE

```

/ WITH BOTH INPUT AND OUTPUT REGISTERS CLEARED
/ DOES CLEARING OUTPUT SET
/ ANY BIT IN INPUT
/

```

```

2200 3030 /CLEAR ERROR FLAG
2201 1177 /SET ITERATION COUNT
2202 3031 /TO 50 (DECIMAL)
2203 3021 /CLEAR EXPECTED RESULT
2204 3022 /CLEAR TEST DATA
2205 6007 /INITIALIZE INTERFACE
2206 1021 /GET TEST DATA
2207 4437 /BIT CLEAR OUTPUT REGISTER
2210 7300 /READ INPUT REGISTER
2211 4436 /SAVE REGISTER DATA
2212 3023 /GET REGISTER DATA
2213 1023 /IS OUTPUT REGISTER 0
2214 7650 /DATA CORRECT, CONTINUE
2215 5222 /NO, ERROR
2216 4444 /

```

```

2217 3767      OUT4E=1      /"DBCO ERROR"
2220 3534      DH3=1      /"MASK EXPECTED RECEIVED"
2221 7775      =3      /NUMBER OF DATA WORDS
2222 4446      JMS I      /TEST FOR LOOP WITH SAME DATA
2223 5205      JMP      /LOOP WITH SAME DATA
2224 2021      ISZ      /INCREMENT DATA PATTERN
2225 5205      JMP I      /CONTINUE
2226 4445      JMS I      /CHECK FOR LOOP ON CURRENT TEST
2227 5203      JMP      /LOOP ON CURRENT TEST
  
```

```

/ WITH BOTH INPUT AND OUTPUT REGISTERS CLEARED
/ DOES CLEARING INPUT SET ANY BIT IN OUTPUT
/
INOU2:      DCA      TYPFLG      /CLEAR ERROR FLAG
            TAD      C=62      /SET ITERATION COUNT
            DCA      LPONT      /TO 50 (DECIMAL)
            DCA      DATA1      /CLEAR TEST DATA
            DCA      DATA2      /CLEAR EXPECTED RESULT
INOU2A:      CAF      DATA      /INITIALIZE INTERFACE
            TAD           /GET TEST DATA
            DBCI           /BIT CLEAR INPUT REGISTER
  
```

```

            CLA CLL      /READ OUTPUT REGISTER
            DBRO      /SAVE REGISTER DATA
            DCA      DATA3      /GET REGISTER DATA
            TAD      DATA3      /IS OUTPUT REGISTER 0
            SNA CLA      /DATA CORRECT, CONTINUE
            JMP      /NO, ERROR
            JMS I      /"DBCO ERROR"
            IN2E=1      /"MASK EXPECTED RECEIVED"
            DH3=1      /NUMBER OF DATA WORDS
            =3      /TEST FOR LOOP WITH CURRENT DATA
            JMS I      XLOOP2      /LOOP WITH SAME DATA
            JMP      INOU2A      /INCREMENT DATA PATTERN
            ISZ      DATA1      /CONTINUE
            JMP      INOU2A      /CHECK FOR LOOP ON CURRENT TEST
            JMS I      XLOOP1      /LOOP ON CURRENT TEST
            JMP      INOU2=3
  
```

```

/ WITH THE OUTPUT REGISTER SET TO ALL IS, AND
/ THE INPUT REGISTER CLEARED, DOES SELECTIVELY
/ CLEARING THE OUTPUT REGISTER SET ANY BIT IN
/ THE INPUT REGISTER
/
INOU3:      DCA      TYPFLG      /CLEAR ERROR FLAG
            TAD      C=62      /SET ITERATION COUNT
            DCA      LPONT      /TO 50 (DECIMAL)
            DCA      DATA1      /CLEAR TEST DATA
            DCA      DATA2      /CLEAR EXPECTED RESULT
INOU3A:      CAF      DATA      /INITIALIZE INTERFACE
            CMA           /SET AC=7777
            DBSO           /BIT SET OUTPUT REGISTER
            DBCI           /BIT CLEAR INPUT REGISTER
            CLA CLL      /GET FLIPFLOP JUMPER MASK
            TAD           /
  
```

```

2273 7040 CMA
2274 3022 DCA
2275 1021 TAD
2276 7040 CMA
2277 0022 AND
2280 3022 DCA
2301 1021 TAD
2302 4437 DBCO
2303 7300 CLA CLL
2304 4436 DBRI
2305 3023 DCA
2306 1023 TAD
2307 7041 CIA
2310 1022 TAD
2311 7650 SNA CLA
2312 5317 JMP I
2313 4444 JMS I
2314 3767 OUT4E=1
2315 3934 DH3=1
2316 7775 =3
2317 4446 JMS I
2320 5265 JMP
2321 2021 ISZ
2322 5265 JMP
2323 4445 JMS I
2324 5263 JMP

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

/
/ INOU4,
3030 DCA
3177 TAD
3031 DCA
3021 LPEN
3021 DCA
6007 CAF
7040 CMA
4440 DBSO
4440 CLA CLL
1027 TAD
7040 CMA
0021 AND
7040 CMA
3022 DCA
1021 TAD
4437 DBCO
7300 CLA CLL
4436 DBRI
3023 DCA
1022 TAD
7041 CIA
1023 TAD
1023 SNA CLA
7650 JMP
5360 JMS I
4444 JMS I

/GET TEST DATA2
/COMPLEMENT
/AND WITH COMPLEMENT OF JUMPER MASK
/TO GET EXPECTED RESULT
/GET TEST DATA
/BIT CLEAR OUTPUT REGISTER

/READ INPUT REGISTER
/SAVE REGISTER DATA
/GET REGISTER DATA

/COMPARE TO EXPECTED RESULT
/ARE THEY THE SAME
/NO, ERROR
/DBCO ERROR"
/MASK EXPECTED RECEIVED
/NUMBER OF DATA WORDS
/TEST FOR LOOP WITH SAME DATA
/LOOP WITH SAME DATA
/INCREMENT DATA PATTERN
/CONTINUE
/CHECK FOR LOOP ON CURRENT TEST
/LOOP ON CURRENT TEST

/ CLEAR ERROR FLAG
/SET ITERATION COUNT
/TO 50 (DECIMAL)
/CLEAR TEST DATA
/INITIALIZE INTERFACE
/SET AC TO 7777
/BIT SET OUTPUT REGISTER

/GET FLIPFLOP JUMPER MASK
/COMBINE WITH MASK
/TO GET EXPECTED RESULT
/GET TEST DATA
/BIT CLEAR OUTPUT REGISTER

/READ INPUT REGISTER
/SAVE REGISTER DATA
/GET EXPECTED RESULT
/COMPARE TO RECEIVED DATA
/ARE THEY THE SAME
/NO, ERROR

```

PAGE

```

2355 3767 OUT4E=1 /"DBCO ERROR"
2356 3534 DH3=1 /"MASK EXPECTED RECEIVED"
2357 7775 =3 /NUMBER OF DATA WORDS
2360 4446 JMS I /TEST FOR LOOP WITH CURRENT DATA
2361 5331 INOU4A /LOOP WITH SAME DATA
2362 2021 ISZ DATA1 /INCREMENT DATA PATTERN
2363 5331 JMP INOU4A /CONTINUE
2364 4445 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
2365 5330 JMP INOU4+3 /LOOP ON CURRENT TEST
2366 5777 INT1 /GO TO NEXT TEST
2400
2400
2400

```

```

/VERIFY THAT EACH BIT SET UP TO SKIP DOES
/
INT1,          TYPFLG
2400 3030 DCA /CLEAR ERROR FLAG
2401 1177 TAD C=62 /SET ITERATION COUNT
2402 3031 DCA LPENT /TO 50(DECIMAL)
2403 3022 DCA DATA2
2404 6007 CAF /INITIALIZE INTERFACE
2405 1022 TAD DATA2
2406 0026 AND IJUMPER
2407 7450 SNA
2410 5241 JMP IN1D
2411 3021 DCA DATA1
2412 1021 TAD DATA1
2413 4440 DBSO /SAVE TEST DATA
2414 7300 CLA CLL /BIT SET OUTPUT REGISTER
2415 6003 SRQ /IS INTERRUPT ACTIVE
2416 5223 JMP I+5 /NO, CONTINUE
2417 4444 JMS I XERROR /YES, ERROR
2420 4042 INTIE=1 /INTERRUPT ACTIVE
2421 3515 DH1=1
2422 7777 =1 /NUMBER OF DATA WORDS TO BE OUTPUT
2423 4433 DBEI /ENABLE INTERFACE
2424 6003 SRQ /IS INTERRUPT ACTIVE
2425 5231 JMP /NO, ERROR
2426 4434 DBSK /IS FLAG SET
2427 5246 JMP /NO, ERROR
2430 5237 JMP /INTERRUPT ACTIVE; FLAG SET
2431 4434 DBSK /IS INTERFACE FLAG SET
2432 5233 JMP I /NO, ERROR
2433 4444 INTIE=1
2434 4042 DH1=1
2435 3515 =1 /NUMBER OF WORDS TO BE OUTPUT
2436 7777 /TEST FOR LOOP ON SAME DATA; ESCAPE ON DATA ERROR
2437 4446 JMS I XLOOP2 /LOOP WITH SAME DATA
2440 5204 JMP IN1A /INCREMENT DATA PATTERN
2441 2022 ISZ DATA2 /CONTINUE
2442 5204 JMP IN1A /CHECK FOR LOOP ON CURRENT TEST
2443 4445 JMS I XLOOP1 /LOOP ON CURRENT TEST
2444 5203 JMP IN1+3 /GO TO NEXT TEST
2445 5240 JMP INT3

```

```

2446 4444 INT1BE, JMS I XERROR /NO, ERROR
2447 4065 INT3E=1
2450 3515 DH1=1
2451 7777 =1 /NUMBER OF WORDS TO BE OUTPUT
2452 5237 JMP INT1OK /NO, ERROR
2453 4444 JMS I XERROR
2454 4077 INT4E=1
2455 3515 DH1=1 /NUMBER OF WORDS TO BE OUTPUT
2456 7777 =1
2457 5237 JMP INT1OK
    
```

```

/VERIFY THAT EACH BIT NOT JUMPER TO SKIP DOES NOT
/
INT3, TYPFLG /CLEAR ERROR FLAG
2460 3030 DCA C=62 /SET ITERATION COUNT
2461 1177 TAD LPONT /TO 50 (DECIMAL)
2462 3031 DCA DATA2 /CLEAR TEST DATA
2463 3022 DCA DATA2 /INITIALIZE INTERFACE
2464 6007 CAF /GET JUMPER MASK
INT3A, TAD IJUMPER /COMPLIMENT FOR NO SKIP BITS
2465 1026 CMA /GET BITS TO BE TESTED
2466 7040 AND /ARE ANY BITS TO BE TESTED
2467 0022 SNA /NO, GET NEXT DATA PATTERN
2470 7450 JMP INT3C /SAVE FOR OUTPUT
2471 5306 DCA DATA1 /GET TEST DATA
2472 3021 TAD DATA1 /BIT SET OUTPUT REGISTER
2473 1021 DBSO
2474 4440 CLA CLL
2475 7300 DBSK
2476 4434 JMP .+5
2477 5304 JMS I XERROR
2500 4444 INT3E=1 /"REGISTER DATA"
2501 3674 DH1=1
2502 3515 =1
2503 7777 JMS I XLOOP2
2504 4446 JMP INT3A
2505 5264 ISZ DATA2
2506 2022 JMP INT3A
2507 5264 JMS I XLOOP1
2510 4445 JMP INT3=3
2511 5263 EPASS
2512 5777 /ERROR HANDLER
    
```

```

PAGE ERROR, 0 CLA CLL /GET POINTER TO ERROR MESSAGE
2577 3257 TAD I ERROR /SAVE POINTER
2600 0000 DCA MESC /GET POINTER TO DATA HEADER
2601 7300 ISZ ERROR /SAVE HEADER
2602 1600 TAD I DORDER /NUMBER OF WORDS TO BE OUTPUT
2603 3234 DCA ISZ
2604 2200 TAD I
2605 1600 DCA
2606 3236 ISZ
    
```

```

2610 1600 TAD I ERROR
2611 3264 DCA DATCNT
2612 1200 TAD TAD
2613 1377 TAD TAD
2614 3776 DCA LSTDCI
2615 1776 TAD LSTDCI
2616 3266 DCA ERRAD
2617 7604 LAS SR01
2620 0334 AND
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 OCTASC
2631 4442 JMS I XPRINT
2632 3024 ERADR=1
2633 4442 JMS I XPRINT
2634 0000 MESSG,
2635 4442 JMS I XPRINT
2636 0000 DHDER,
2637 4442 JMS I XPRINT
2640 3512 CRLF=1
2641 1264 TAD TAD
2642 7650 SNA CLA
2643 5254 JMP
2644 1265 TAD
2645 3010 DCA POINT1
2646 1410 TAD I POINT1
2647 4774 JMS BITOUT
2650 2264 JMS DATCNT
2651 5246 JMP BITS
2652 4442 JMS I XPRINT
2653 3512 CRLF=1
2654 7604 LAS
2655 0333 AND SR00
2656 7640 SZA CLA
2657 5262 JMP
2660 1266 TAD ERRAD
2661 7402 HLT
2662 2200 JSE ERROR
2663 5600 JMP I ERROR
2664 0000 DATCNT,
2665 0020 DATAP,
2666 0000 ERRAD,

2667 0000 LOOP1,
2668 1030 TAD TYPFLG
2669 7650 SNA CLA
2672 5277 JMP LPIEXA
2673 7604 LAS

```

/CLEAR ERROR FLAG

/NUMBER OF WORDS TO BE OUTPUT

/TEST FOR LOOP ON CURRENT TEST

2674 0341 AND SR06
 2675 7640 SZA CLA
 2676 5310 JMP LPIEXX-1
 2677 7604 LAS AND SR05
 2700 0340 SZA CLA
 2701 7640 JMP LPIEXT
 2702 5305 ISZ LPCNT
 2703 2031 JMP LPIEXX
 2704 5311 LAS AND SR02
 2705 7604 SZA CLA
 2706 0335 JMP LPIEXT
 2707 7650 ISZ LOOP1
 2710 2267 JMP I LOOP1
 2711 5667 LPIEXX

/TEST FOR LOOP ON CURRENT DATA

LOOP2: 0
 2712 0000 TAD TYPFLG
 2713 1030 SNA CLA
 2714 7650 JMP LP2EXT
 2715 5326 LAS AND SR06
 2716 7604 SNA CLA
 2717 0341 JMP I+5
 2720 7650 TAD LOOP2
 2721 5326 TAD (5
 2722 1312 DCA LOOP2
 2723 1373 JMP I LOOP2
 2724 3312 LAS AND SR03
 2725 5712 SNA CLA
 2726 7604 JMP I LOOP2
 2727 0336 AND SR03
 2730 7650 SNA CLA
 2731 2312 ISZ LOOP2
 2732 5712 JMP I LOOP2
 2733 4000 SR00,
 2734 2000 SR01, 2000
 2735 1000 SR02, 1000
 2736 0400 SR03, 400
 2737 0200 SR04, 200
 2740 0100 SR05, 100
 2741 0040 SR06, 40

/DATA CORRECT, CONTINUE

/OCTAL TO PACKED ASCII CONVERSION

PAGE /
 OCTASC, 0
 2773 0005 CLA CLL
 2774 3031 TAD
 2775 3000 BSW
 2776 3027
 2777 7775
 3000
 3001 7300
 3002 1227
 3003 7002

/GET WORD TO BE CONVERTED
/SWAP HALVES, SEPARATE DIGITS

3004	4212	JMS	SPLIT	/CONVERT MOST SIGNIFICANT
3005	3226	DCA	MSIDGT	/DIGITS TO ASCII
3006	1227	TAD	LSIDGT	/CONVERT LEAST SIGNIFICANT
3007	4212	JMS	SPLIT	/DIGITS TO ASCII
3010	3227	DCA	LSIDGT	/RETURN
3011	5600	JMP I	OCTASC	
3012	0000	Ø		
3013	0377	AND	(77	
3014	7421	MQL		
3015	7501	MQA		
3016	7106	CLL RTL		
3017	7004	RAL		
3020	0376	AND	(707	
3021	7501	MQA		
3022	0376	AND	(707	
3023	1375	TAD	(6060	
3024	5612	JMP I	SPLIT	
3025	3736	TEXT	/5+	
3026	4040			
3027	4040			
3030	4000			

/OUTPUT 12 BIT BINARY WORD

3031	0000	BITOUT,	Ø	/SAVE DATA IN MQ
3032	7421	MQL		/SET UP TO OUTPUT
3033	1374	TAD	(014	/12 BITS
3034	3020	DCA	CNTR1	/GET DATA
3035	7501	MQA		/GET MSB INTO LINK
3036	7104	CLL RAL		/SAVE REST OF WORD
3037	7421	MQL		/GET ASCII 1 INTO AC
3040	1373	TAD	("I	
3041	7420	SNL		/IS BIT=1
3042	0372	AND	(Ø	/NO, CHANGE TO ASCII Ø
3043	4443	JMS I	XTYPE	/OUTPUT BIT
3044	2020	ISZ	CNTR1	/CONTINUE
3045	5235	JMP	BITØ1	/TYPE 2 SPACES
3046	1371	TAD	(240	/AFTER LAST BIT HAS BEEN
3047	4443	JMS I	XTYPE	/OUTPUTED
3050	1371	TAD	(240	/RETURN
3051	4443	JMS I	XTYPE	
3052	5631	JMP I	BITOUT	

/CHARACTER STRING OUTPUT ROUTINE

3171	0240			
3172	0260			
3173	0261			
3174	7764			
3175	6060			
3176	0707			
3177	0097			
3200	0000	PAGE	/	
		PRINT,	Ø	


```

3201 7300 CLA CLL
3202 1600 TAD I
3203 3010 DCA
3204 2200 ISZ
3205 1410 TAD I
3206 7421 MGL
3207 7501 MGA
3210 7002 BSW
3211 4215 JMS
3212 7501 MGA
3213 4215 JMS
3214 5205 JMP
/UNPACK, DECODE, OUTPUT
/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

```

```

TYPSET, 0
3215 0000 AND
3216 0243 SNA
3217 7450 JMP I
3220 5600 TAD
3221 1244 SPA
3222 7510 JMP
3223 5226 TAD
3224 1250 JMP
3225 5241 IAC
3226 7001 SEA
3227 7440 JMP
3230 5283 TAD
3231 1245 JMP
3232 5241 IAC
3233 7001 SEA
3234 7440 JMP
3235 5240 TAD
3236 1246 JMP
3237 5241 TAD
3240 1247 JMP
3241 4443 MTP,
3242 5615 K0077,
3243 0077 M40,
3244 7740 K215,
3245 0215 K212,
3246 0212 K336,
3247 0336 K240,
3250 0240
/OUTPUT ONE CHARACTER TO TTY
/MASK UNWANTED BITS
/IS AC=0
/YES, END OF MESSAGE, EXIT
/SUBTRACT 40
/IS PACKED CHARACTER >40
/NO
/YES, CONVERT TO ASCII
/OUTPUT
/ADD 1 TO AC
/IS CHARACTER=37
/NO
/GET CODE FOR CARRIAGE RETURN
/OUTPUT
/ADD 1 TO AC
/IS CHARACTER=37
/NO
/GET CODE FOR LINE FEED
/OUTPUT
/PACKED CHARACTER $40, CONVERT TO ASCII
/OUTPUT

```

```

TYPSET, 0
3251 0000 TLS
3252 6046 TSF
3253 6041 JMP
3254 5253 CLA
3255 7200 JMP I
3256 5651
/
/
/

```

```

3257 7604 EPASS,
3260 0777'
3261 7640 SR04
3262 5776' SZA CLA INIT1
3263 7604 LAS
3264 0775' AND SR05
3265 7640 SZA CLA EPAS1
3266 5272 JMP XPRINT
3267 4442 JMS I
3270 3274 MEP-1
3271 5776' JMP INIT1
3272 1374 TAD (207
3273 4251 JMS TYPE
3274 5776' JMP INIT1
3275 3736 MEP, /-DR/
3276 0422 TEXT
3277 0000

```

/TELETYPE MESSAGES

```

3374 0207
3375 2740
3376 0400
3377 2737
3400 3400
3401 2305
3402 2440
3403 2322
3404 4006
3405 1722
3406 4004
3407 0526
3410 1103
3411 0540
3412 0317
3413 0405
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

```

PAGE M1.

TEXT /-SET SR FOR DEVICE CODE AND CONT

M2, TEXT /-SET SR FOR INTERRUPT JUMPERS AND CONT/

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
 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
 3505 2516
 3506 4001
 3507 1604
 3510 4003
 3511 1716
 3512 2400
 3513 3736
 3514 0000

M2A: TEXT /-SET SWITCHES FOR FLIPFLOP JUMPERS AND CONTINUE/

M3: TEXT /-SET SR FOR RUN AND CONT/

CRLF: TEXT /- /

/DATA HEADERS

0 TEXT /-REGISTER DATA/

DH0: 0000
 DH1: 3736
 2205

3520	0711				
3521	2324				
3522	0522				
3523	4004				
3524	0124				
3525	0100				
3526	3736	DH2,	TEXT	/*AC CONTENTS/	
3527	0103				
3530	4003				
3531	1716				
3532	2405				
3533	1624				
3534	2300				
3535	3736	DH3,	TEXT	/*MASK	EXPECTED
3536	1501				RECEIVED/
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	0304				
3560	0000				
3561	3736	DH4,	TEXT	/*EXPECTED	RECEIVED/
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	DH5,	TEXT	/*REGISTER	DATA OUT
3577	2205				DATA IN/
3600	0711				
3601	2324				
3602	0522				
3603	4040				
3604	4040				
3605	4040				
3606	4004				

DATA	TEXT	/*AC CONTENTS	DATA OUT	DATA IN/
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			
3632	4004			
3633	0124			
3634	0140			
3635	1725			
3636	2440			
3637	4040			
3640	4040			
3641	4040			
3642	0401			
3643	2401			
3644	4011			
3645	1600			

INITIE: TEXT /ERROR MESSAGE
 /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

INITIE: TEXT /INPUT REG NOT CLEARED/

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
3716 0300
3717 0402
3720 2211
3721 4004
3722 1104
3723 4016
3724 1724
3725 4003
3726 1405
3727 0122
3730 4001
3731 0300
3732 0402
3733 2317
3734 4003
3735 1001
3736 1607
3737 0504
3740 4001
3741 0300
3742 0402
3743 0317
3744 4003
3745 1001
3746 1607
3747 0504
3750 4001
3751 0300
3752 0402
3753 0311
3754 4003
3755 1001
3756 1607
3757 0504
3760 4001
3761 0300

INIT3E, TEXT /SKIP FLAG SET/

TRAN1E, TEXT /DBRO DID NOT CLEAR AC/

TRAN2E, TEXT /DBRI DID NOT CLEAR AC/

TRAN3E, TEXT /DBSO CHANGED AC/

TRAN4E, TEXT /DBCO CHANGED AC/

TRAN5E, TEXT /DBCI CHANGED AC/

3762	0402	OUT1E:	TEXT	/DBSD 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	IN3E:	TEXT	/INPUT REGISTER DATA ERROR/
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	IN4E:	TEXT	/LATCH ERROR/
4030	2403			
4031	1040			
4032	0522			
4033	2217			
4034	2200			
4035	0402	IN9E:	TEXT	/DBRI ERROR/
4036	2211			
4037	4005			
4040	2222			
4041	1722			
4042	0000			
4043	1116	INT1E:	TEXT	/INTERRUPT ACTIVE/
4044	2405			
4045	2222			
4046	2520			
4047	2440			
4050	0103			

4051 2411
 4052 2605
 4053 0000
 INT2E: TEXT /NO INTERRUPT, SKIP/
 4054 1617
 4055 4011
 4056 1624
 4057 0522
 4060 2225
 4061 2024
 4062 5440
 4063 2313
 4064 1120
 4065 0000
 4066 1116
 4067 2605
 4070 2222
 4071 2520
 4072 2434
 4073 4016
 4074 1740
 4075 2313
 4076 1120
 4077 0000

INT3E: TEXT /INTERRUPT, NO SKIP/

INT4E: TEXT /NO INTERRUPT, NO SKIP/

INT5E: TEXT /NO SKIP/

INT6E: TEXT /SKIP/

S

0177 7716

4000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
4100 11111111 11111111 11000000 00000000 00000000 00000000 00000000 00000000

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	IN1B	V141	1420	K0077	3243	SR06	2741
BITOUT	3031	IN2	1434	K212	K212	3246	SR0	6003
BITS	2646	IN2E	4004	K215	K215	3245	START1	0202
B5W	7002	IN3	1462	K240	K240	3250	START2	0244
CAF	6007	IN3A	1466	K336	K336	3247	T10T	0250
CNTRI	0020	IN3E	4012	L00P1	L00P1	2667	TRAN1	0460
CRLF	3513	IN4	4027	L00P2	L00P2	2712	TRAN1E	3704
DATA1	0021	IN5	1600	LP1EXA	LP1EXA	2677	TRAN2	0477
DATA2	0022	IN5A	1604	LP1EXT	LP1EXT	2705	TRAN2E	3717
DATA3	0023	IN5C	1633	LP1EXX	LP1EXX	2711	TRAN3	0516
DATA4	0024	IN6	1637	LP2EXT	LP2EXT	2726	TRAN3E	3736
DATA5	0025	IN6A	1644	LPCNT	LPCNT	0031	TRAN4	0536
DATAP	2665	IN6C	1672	LSTDGT	LSTDGT	3027	TRAN4E	3742
DATCNT	2664	IN7	1676	M1	M1	3400	TRAN5	0556
DATOUT	2641	IN7A	1702	M2	M2	3421	TRAN5E	3752
DBC1	4435	IN7C	1731	M2A	M2A	3445	TRAN6	0600
DBCIX	0300	IN8	2000	M3	M3	3476	TRAN7	0616
DBCO	4437	IN8A	2005	M40	M40	3244	TRAN8	0634
DBCOX	0312	IN9	2030	MEP	MEP	3275	TYPE	3251
DBDI	4432	IN9A	2034	MSG	MSG	2634	TYPEFLG	0030
DBDIX	0261	IN9E	4035	MOA	MOA	7501	TYPSET	3215
DBEI	4433	INIT1	0400	MQL	MQL	7421	XDBCI	0037
DBEIX	0266	INIT1E	3646	MSTDGT	MSTDGT	3026	XDBCO	0032
DBRI	4436	INIT2	0416	MTP	MTP	3241	XBDI	0033
DBRIX	0305	INIT2E	3662	OCTASC	OCTASC	3000	XDBEI	0036
DBRO	4441	INIT3	0435	OUT1	OUT1	1000	XDBRI	0041
DBROX	0324	INIT3E	3675	OUT1E	OUT1E	3762	XDBSK	0034
DBSK	4434	INOU1	2200	OUT2	OUT2	1033	XDBSO	0040
DBSKX	0273	INOU1A	2205	OUT3	OUT3	1054	XERROR	0044
DBSO	4440	INOU2	2230	OUT3A	OUT3A	1060	XL00P1	0045
DBSOX	0317	INOU2A	2235	OUT4	OUT4	1105	XL00P2	0046
DH0	3515	INOU3	2260	OUT4A	OUT4A	1111	XPRINT	0042
DH1	3526	INOU3A	2265	OUT4E	OUT4E	3770	XTYPE	0043
DH2	3535	INOU4	2325	OUT5	OUT5	1200		
DH3	3561	INOU4A	2331	OUT5A	OUT5A	1205		
DH4	3576	INT1	2400	OUT6	OUT6	1230		
DH5	3622	INT1A	2404	OUT6A	OUT6A	1234		
DH6	2636	INT1AE	2431	OUT7	OUT7	1262		
DHDER	0260	INT1BE	2446	OUT7A	OUT7A	1266		
D10T	2654	INT1CE	2453	OUT7E	OUT7E	3776		
EHALT	3272	INT1D	2441	OUT8	OUT8	1315		
EPAS1	3257	INT1E	4043	OUT8A	OUT8A	1322		
EPASS	3025	INT10K	2437	PNTRI	PNTRI	0233		
ERADR	2666	INT2E	4024	POINT1	POINT1	0010		
ERRAD	2600	INT3	2460	PRINT	PRINT	3200		
ERROR	0027	INT3A	2464	SPLIT	SPLIT	3012		
FJUMPE	0026	INT3C	2506	SR00	SR00	2733		
INI	1400	INT3E	4066	SR01	SR01	2734		
INI0	2063	INT4E	4100	SR02	SR02	2735		
INI0A	2070	INT5E	4113	SR03	SR03	2736		
IN1A	1403	INT6E	4117	SR04	SR04	2737		
		IOTS	0247	SR05	SR05	2740		

ERRORS DETECTED: 0

LINKS GENERATED: 18

RUN-TIME: 15 SECONDS

3K CORE USED

