

IDENTIFICATION  
\*\*\*\*\*

PRODUCT CODE: MAINDEC=08=DHDRA=B=D

(REPLACES MAINDEC=0E=D0QA=D)

PRODUCT NAME: DR0=EA 12 CHANNEL BUFFERED DIGITAL INTERFACE ✓

DATE CREATED: MARCH 1972  
JUNE 1975

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: MICHAEL DAVIS  
D. MACOMBER

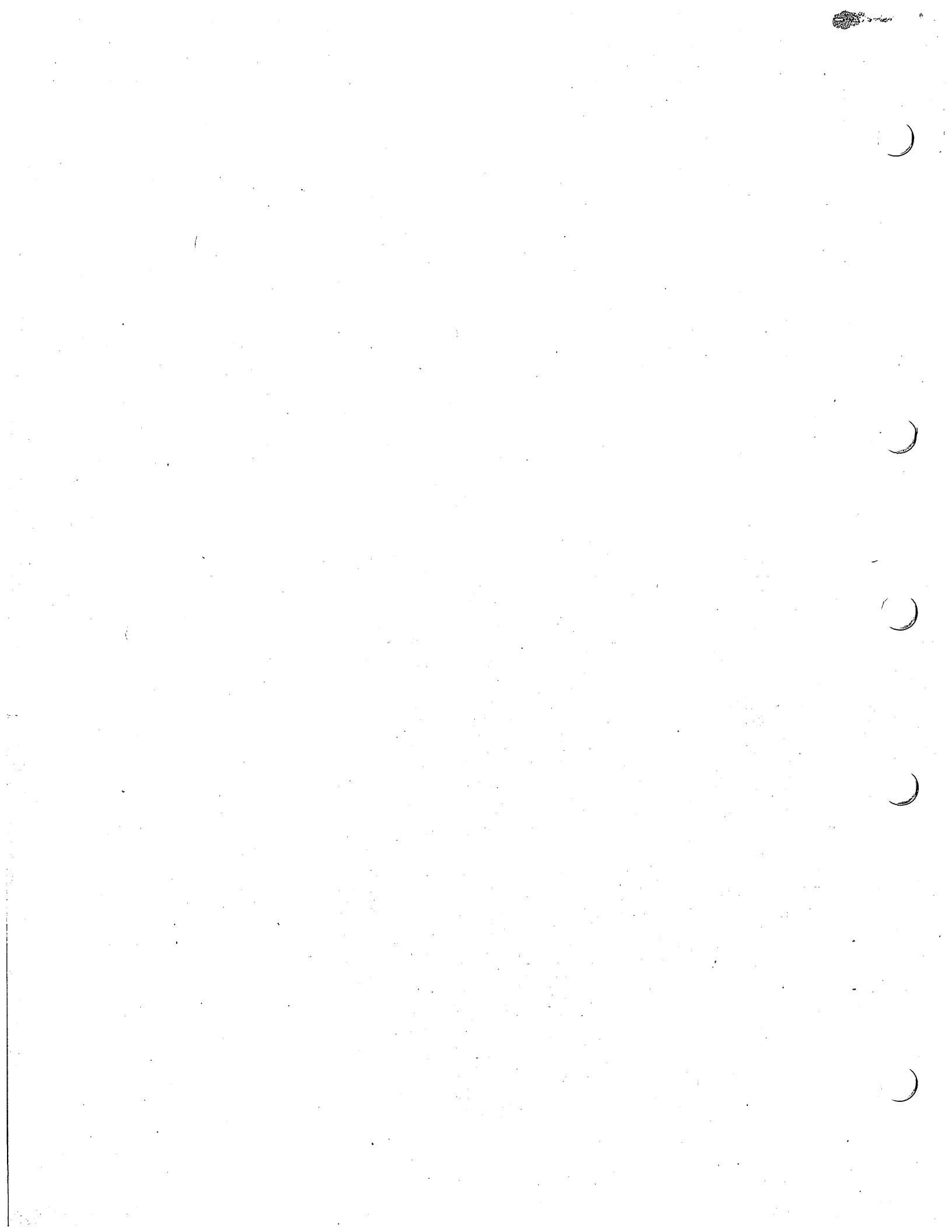
THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

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

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1972, 1975

BY DIGITAL EQUIPMENT CORPORATION



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.

MODIFIED TO RUN ON APT8A SYSTEMS JUNE 1975.  
SEE NOTES AT END OF DOCUMENT.

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 000003377

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.

4.3.1.1 SET SWITCHES TO 000X 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.  
4.2.1.2 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.  
4.3.1.3 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

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 80 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)  
 XXXX ADDRESS OF JMS TO ERROR ROUTINE IN TEST THAT FAILED.
- 6.2 ERROR RECOVERY
  - SET SR6=11UP) 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

8. MISCELLANEOUS

8.1 EXECUTION TIME

EXECUTION TIME IS APPROXIMATELY 3 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 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.

APT NOTES  
\*\*\*\*\*

ALL OF THE FOLLOWING NOTES APPLY ONLY WHEN THE PROGRAM IS BEING RUN ON AN APT SYSTEM.

1. FOR MORE INFORMATION SEE THE FOLLOWING DOCUMENTS.
  - A. STANDARD APT SYSTEM TO PDP8 DIAGNOSTIC INTERFACE.
  - B. APT SYSTEM MANAGERS GUIDE.
2. FOR ANY DIFFERENCES BETWEEN THESE NOTES AND THE REST OF THE DOCUMENT, THESE NOTES WILL PREVAIL. (SEE THE LISTING ALSO.)
3. ALL CODE THAT HAS BEEN ADDED (INSERTED) FOR APT WILL CONTAIN THE EXPRESSION: /APT/  
 ANY ORIGINAL CODE NEGATED FOR APT HAS BEEN 'REMOVED' BY PRECEDING IT WITH THE EXPRESSION: /\*APT\*/
4. IF BIT 0 OF HOW1 IS A '1', THEN THE HARDWARE SWITCH REGISTER WILL BE USED, REGARDLESS OF LOAD METHOD (SCRIPT OR DUMP).

THE FOLLOWING NOTES APPLY ONLY WHEN THE LOAD METHOD WAS SCRIPT LOAD.

5. SWITCHES:

REFER TO SECTIONS 4.3.1 AND 4.3.2 IN THE DOCUMENT.

THE FOLLOWING PROCEDURES ARE AT VARIANCE WITH THE USE OF THE PSEUDO SWITCH REGISTER (PSR); AND HARDWARE CONFIGURATION WORDS 1, 2, & 3 (HOW1, HOW2, & HOW3) AS SET FORTH IN THE "STANDARD APT SYSTEM TO PDP8 DIAGNOSTIC INTERFACE".

FOR EFFECTIVELY STARTING AT LOC 200:

- A. PSR:
  - BITS 0, 1, 4, & 6 (WILL BE FORCED TO ONES BY APT INITIALIZATION.
  - BITS 2,3,5 = (SEE SECTION 3.1)
  - BIT 7 = 0
  - BIT 8 = (NOT USED)
  - BITS 9-11 = (SEE SECTION 4.3.1.1)
- B. HOW1:
  - BITS 0-6 = (\*\*)
  - BITS 7-11 = (INTERUPT JUMPERS - 1ST 5 BITS) (SEE SECTION 4.3.1.2)
- C. HOW2:



BITS 0=3 = (\*\*)

BIT 4 = (NOT USED)

BITS 5=11 = (INTERUPT JUMPERS = NEXT 7 BITS)  
(SEE SECTION 4.3.1.2)

D. HCW3:

BITS 0=11 = (INPUT REGISTER FLIPFLOPS)  
(SEE SECTION 4.3.1.3)

FOR EFFECTIVELY STARTING AT LOC 201:

A. PSR1:

BITS 0,1,4,6 = (WILL BE FORCED TO ONES BY  
APT INITIALIZATION)

BITS 2,3,5 = (SEE SECTION 5.1)

BITS 7 = 1

BITS 8=11 = (NOT USED)

B. HCW1:

BITS 0=6 = (\*\*)

BITS 7=11 = (NOT USED)

C. HCW2:

BITS 0=3 = (\*\*)

BITS 4=11 = (NOT USED)

D. HCW3:

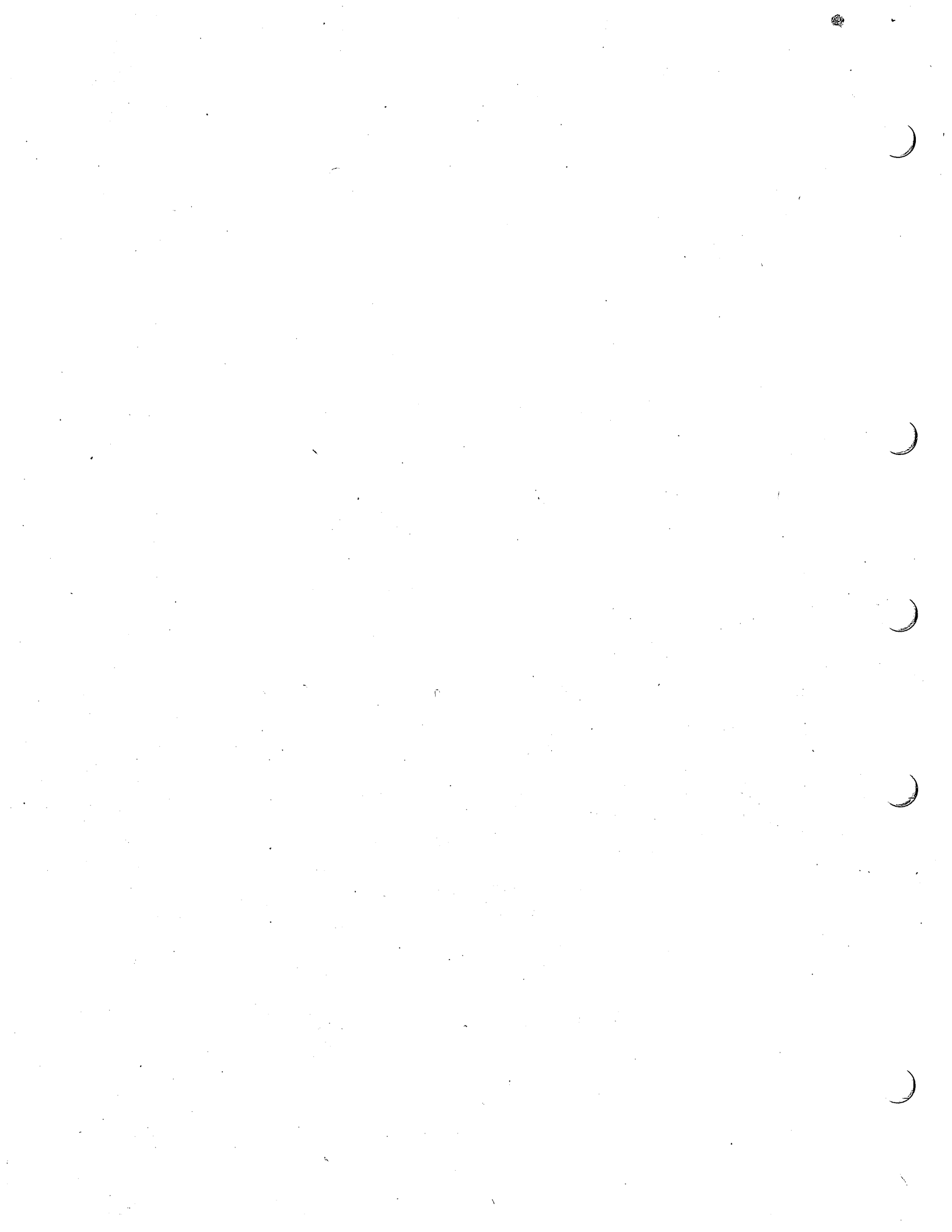
BITS 0=11 = (NOT USED)

\*\* AS DESCRIBED IN THE "STANDARD APT SYSTEM TO PDP8 DIAGNOSTIC  
INTERFACE".

6. ERRORS:

ALL ERRORS CALL APT.

LISTING  
\*\*\*\*\*



/DR8-EA 12 CHANNEL BUFFERED I/O DIAGNOSTIC MAINDEC-08-DHRDA  
 /COPYRIGHT 1972, 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754  
 /STARTING ADDRESS: 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 HQA=7501  
 7002 BSW=7002  
 6007 CAF=6007  
 6003 SRQ=6003  
 4440 DBDI=JMS I XDBDI  
 4441 DBEI=JMS I XDBEI  
 4442 DBSK=JMS I XDBSK  
 4443 DBCI=JMS I XDBCI  
 4444 DBRI=JMS I XDBRI  
 4445 DBCO=JMS I XDBCO  
 4446 DBSO=JMS I XDBSO  
 4447 DBRO=JMS I XDBRO

4425 LAB=JMS I IAPTSR /APT/REDEFINE THE LAB INST,  
 7402 HLT=7402 /APT/MAKE HLT VISIBLE IN CREF,

/LOCATION EQUIVALENCES

3026 MSTDCT=ERADR+1  
 3027 LSTDCT=ERADR+2

/GENERAL VARIABLES

0010 0010 \*10  
 0010 0000 POINT1, 0  
 0020 0020 \*20  
 0020 0000 PSR, 0 /APT/  
 0021 4000 HCH1, 4000 /APT/  
 0022 0000 HCH2, 0 /APT/  
 0023 0000 HCH3, 0  
 0024 4200 IAPTER, APTER /APT/  
 0025 4212 IAPTSR, APTS /APT/  
 0026 0000 CNTR1, 0  
 0027 0000 DATA1, 0  
 0030 0000 DATA2, 0  
 0031 0000 DATA3, 0  
 0032 0000 DATA4, 0  
 0033 0000 DATA5, 0  
 0034 7777 IJUMPE, 7777 /INTERRUPT JUMPER MASK  
 0035 7777 FJUMPE, 7777 /FLIPPLOP JUMPER MASK  
 0036 0000 TYPFLG, 0  
 0037 0000 LPCNT, 0

/INDIRECT POINTERS

0040 0261 XDBDI, DBDIX  
 0041 0266 XDBEI, DBEIX  
 0042 0273 XDBSK, DBSKX  
 0043 0300 XDBCI, DBCIX  
 0044 0305 XDBRI, DBRIX  
 0045 0312 XDBCO, DBCOX  
 0046 0317 XDBSO, DBSOX  
 0047 0324 XDBRO, DBROX  
 0050 3200 XPRINT, PRINT  
 0051 3251 XTYPE, TYPE  
 /APT/ IF UNDER APT CONTROL THE NEXT LOC WILL \* A POINTER TO  
 /APT/ THE APT ERROR HANDLING ROUTINE (APTER).  
 0052 2600 XERROR, ERROR  
 0053 2667 XLOOP1, LOOP1  
 0054 2712 XLOOP2, LOOP2

```

/TEST INITIALIZATION
/
0200 *200
0200 5777' JMP APTI2 /APT/
/*APT*/ JMP START1
0201 5244 JMP START2
0202 6007 START1, CAP /CLEAR ALL FLAGS
0203 4450 JMS I XPRINT /TYPE "SET SR FOR DEVICE
0204 3377 M1=1 /CODE AND CONT"
0205 7402 HLT /HALT FOR SWITCHES
0206 4425 LAS /GET SWITCHES
0207 6376 AND (? /MASK DEVICE CODE
0210 7106 CLL RTL /POSITION BITS
0211 7004 RAL
0212 1375 TAD (6900 /GENERATE BASIC IOT
0213 3247 DCA IOTS /SAVE BASIC IOT
0214 4450 JMS I XPRINT /TYPE "SET SR FOR JUMPERS
0215 3420 M2=1 /AND CONT"
0216 7402 HLT /HALT FOR SWITCHES
0217 4425 LAS /GET SWITCHES
0220 3034 DCA I JUMPER /SAVE JUMPER MASK
0221 4450 JMS I XPRINT
0222 3444 M2A=1
0223 7402 HLT
0224 4425 LAS
0225 3035 DCA FJUMPER
0226 1374 TAD (=10 /8 IOTS WILL BE
0227 3026 DCA CNTR1 /SET UP
0228 1260 TAD DIOT /STORE INSTRUCTION FOR
0231 3233 DCA PNTR1 /IOT SET UP
0232 1247 TAD IOTS /GET IOT
0233 0000 PNTR1, 0 /STORE IT
0234 2233 ISE PNTR1 /PREPARE TO STORE
0235 2247 ISE IOTS /NEXT IOT
0236 2026 ISE CNTR1
0237 5232 JMP PNTR1=1

0240 4450 JMS I XPRINT /TYPE "SET SR FOR RUN
0241 3475 M3=1 /AND CONT
0242 7402 HLT /HALT FOR SWITCHES
0243 7300 CLA CLL
0244 3036 START2, DCA TYPLG /CLEAR ERROR FLAG /CLEAR TYPE FLAG
0245 5646 JMP I .A1 /GO TO FIRST TEST
0246 3400 INITA
    
```

```

/INITIALIZATION CONSTANTS AND VARIABLES
/
0247 0000 IOTS, 0 /BASIC IOT
0250 0262 TIOT,
0251 0267 DBDIX*1
0252 0274 DBSKX*1
0253 0301 DBEIX*1
0254 0306 DBRIX*1
0255 0313 DBCOX*1
0256 0320 DBSOX*1
0257 0325 DBROX*1
0260 3650 DIOT, DCA I TIOT

/IOT SUBROUTINES
/DISABLE DATA BUFFER INTERRUPT (0901,65X0)
0261 0000 DBDIX, 0
0262 6500
0263 7410 SKP /TRAP FOR UNDESIRE
/APT/ IF UNDER APT CONTROL THE NEXT LOC WILL = JMS I IAPTR,
APTE00, /APT/

0264 7402 HLT /SKIPS
0265 5661 JMP I DBDIX
/ENABLE DATA BUFFER INTERRUPTS (DBE1,65X1)
/
0266 0000 DBEIX, 0
0267 6501
0270 7410 SKP /TRAP FOR UNDESIRE
/APT/ IF UNDER APT CONTROL THE NEXT LOC WILL = JMS I IAPTR,
APTE01, /APT/

0271 7402 HLT /SKIPS
0272 5666 JMP I DBEIX
/SKIP ON DATA BUFFER INPUT FLAG (DBSK,65X2)
/
0273 0000 DBSKX, 0
0274 6502
0275 7410 SKP
0276 2273 ISE DBSKX
0277 5673 JMP I DBSKX
/8S TO INPUT REGISTER CORRESPONDING
    
```

```

      /TO IS IN AC (DBGI,65X3)
      /
0302 0000 DBQIX, 0
0301 6503          6503
0302 7410          SKP          /TRAP FOR UNDESIREO
      /APT/ IF UNDER APT CONTROL THE NEXT LOC WILL * JMS I IAPTER,
      APTE02,          /APT/
0303 7402          HLT          /SKIPS
0304 5700          JMP I DBQIX

```

```

      /INPUT REGISTER TO AC (DBRI,65X4)
      /
0305 0000 DBRIX, 0
0306 6504          6504
0307 7410          SKP          /TRAP FOR UNDESIREO
      /APT/ IF UNDER APT CONTROL THE NEXT LOC WILL * JMS I IAPTER,
      APTE03,          /APT/
0310 7402          HLT          /SKIPS
0311 5705          JMP I DBRIX
      /
      /ZEROS TO OUTPUT REGISTER CORRESPONDING TO
      /ONES IN AC (DBCO,65X5)
0312 0000 DBCOX, 0
0313 6505          6505
0314 7410          SKP          /TRAP FOR UNDESIREO
      /APT/ IF UNDER APT CONTROL THE NEXT LOC WILL * JMS I IAPTER,
      APTE04,          /APT/
0315 7402          HLT          /SKIPS
0316 5712          JMP I DBCOX
      /
      /IS TO OUTPUT REGISTER CORRESPONDING
      /TO IS IN AC (DBSO,65X6)
      /
0317 0000 DBSOX, 0
0320 6506          6506
0321 7410          SKP          /TRAP FOR UNDESIREO
      /APT/ IF UNDER APT CONTROL THE NEXT LOC WILL * JMS I IAPTER,
      APTE05,          /APT/
0322 7402          HLT          /SKIPS
0323 5717          JMP I DBSOX
      /
      /JAM TRANSFER OUTPUT REGISTER TO AQ (DBRO 65X7)
      /
0324 0000 DBROX, 0
0325 6507          6507
0326 7410          SKP          /TRAP FOR UNDESIREO
      /APT/ IF UNDER APT CONTROL THE NEXT LOC WILL * JMS I IAPTER,
      APTE06,          /APT/

```

```

0327 7402      HLT          /SKIPS
0330 5724      JMP I   DBROX
0374 7770
0375 6500
0376 0007
0377 4400
2400      PAGE

```

```

/IS OUTPUT REGISTER CLEARED BY INITIALIZE?
0400 3036      INIT1, DCA   TYPFLG  /CLEAN ERROR FLAG
0401 3037      DCA   LPCNT  /SET ITERATION COUNT TO 4096(DECIMAL)
0402 6007      CAF           /INITIALIZE INTERFACE
0403 4447      DBRO          /READ OUTPUT REGISTER
0404 3027      DCA   DATA1 /SAVE REGISTER DATA
0405 1027      TAD   DATA1 /GET REGISTER DATA
0406 7600      SNA CLA  /HAS REGISTER CLEARED BY INITIALIZE
0407 5214      JMP   +0    /DATA CORRECT, CONTINUE
0410 4492      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 4453      JMS I   XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0415 5202      JMP   INIT1+2 /LOOP ON CURRENT TEST

/IS INPUT REGISTER CLEARED BY INITIALIZE?
0416 3036      INIT2, DCA   TYPFLG  /CLEAN ERROR FLAG
0417 3037      DCA   LPCNT  /SET ITERATION COUNT TO 4096(DECIMAL)
0420 7410      SKP           /INITIALIZE INTERFACE
0421 6007      CAF           /INITIALIZE INTERFACE
0422 4444      DBRI          /READ INPUT REGISTER
0423 3027      DCA   DATA1 /SAVE REGISTER DATA
0424 1027      TAD   DATA1 /GET REGISTER DATA
0425 7600      SNA CLA  /HAS REGISTER CLEARED
0426 5233      JMP   +0    /DATA CORRECT, CONTINUE
0427 4492      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 4453      JMS I   XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0434 5221      JMP   INIT2+3 /LOOP ON CURRENT TEST

/IS SKIP FLAG SET AFTER INITIALIZE
0435 3036      INIT3, DCA   TYPFLG  /CLEAN ERROR FLAG
0436 3037      DCA   LPCNT  /SET ITERATION COUNT TO 4096(DECIMAL)
0437 7410      SKP           /INITIALIZE INTERFACE
0440 6007      CAF           /INITIALIZE INTERFACE
0441 4441      DBEI          /ENABLE INTERFACE INTERRUPTS
0442 6003      SRG           /TEST FOR INTERRUPT ACTIVE
0443 5200      JMP   +0    /NO INTERRUPT, CONTINUE
0444 4492      JMS I   XERROR /INTERRUPT ACTIVE, ERROR
0445 4042      INT1E=1 /"INTERRUPT ACTIVE"
0446 3514      DH0=1  /NO DATA HEADER
0447 0000      0          /NO DATA
0450 4442      DBSK          /IS INTERFACE FLAG SET
0451 5206      JMP   +0    /FLAG NOT SET, CONTINUE
0452 4492      JMS I   XERROR /FLAG SET, ERROR
0453 3674      INIT3E=1 /"SKIP FLAG SET"
0454 3514      DH0=1  /NO DATA HEADER

```

```

0455 0000      0      /NO DATA
0456 4453     JMS I  XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0457 5240     JMP      INITS*3  /LOOP ON CURRENT TEST
    
```

```

/DOES OUTPUT REGISTER JAM TRANSFER TO AC?
0460 3036     TRAN1, DCA      TYRFLG /CLEAR ERROR FLAG
0461 3037     DCA      LPCNT  /SET ITERATION COUNT TO 4096(DECIMAL)
0462 6007     CAF      /INITIALIZE INTERFACE
0463 7340     CLA CLL CMA  /SET AC =7777
0464 4447     DBRO     /READ OUTPUT REGISTER
0465 3027     DCA      DATA1 /SAVE AC CONTENTS
0466 1027     TAD      DATA1 /GET AC CONTENTS
0467 7650     SNA CLA  /HAS AC CLEARED BY TRANSFER
0470 5275     JMP      ,+5    /DATA CORRECT, CONTINUE
0471 4452     JMS I  XERROR /NO, ERROR
0472 3705     TRAN1=1 /DBRO DID NOT CLEAR AC
0473 3525     DH2=1  /"AC CONTENTS"
0474 7777     -1      /NUMBER OF WORDS TO BE OUTPUT
0475 4453     JMS I  XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0476 5262     JMP      TRAN1*2 /LOOP ON CURRENT TEST
    
```

```

/DOES INPUT REGISTER JAM TRANSFER TO AC
0477 3036     TRAN2, DCA      TYRFLG /CLEAR ERROR FLAG
0478 3037     DCA      LPCNT  /SET ITERATION COUNT TO 4096(DECIMAL)
0479 6007     CAF      /INITIALIZE INTERFACE
0480 7340     CLA CLL CMA  /SET AC =7777
0481 4444     DBR1     /READ INPUT REGISTER
0482 3027     DCA      DATA1 /SAVE AC CONTENTS
0483 1027     TAD      DATA1 /GET AC CONTENTS
0484 7650     SNA CLA  /HAS AC CLEARED BY TRANSFER
0485 5014     JMP      ,+5    /DATA CORRECT, CONTINUE
0486 4452     JMS I  XERROR /NO, ERROR
0487 3716     TRAN2=1 /DBR1 DID NOT CLEAR AC
0488 3525     DH2=1  /"AC CONTENTS"
0489 7777     -1      /NUMBER OF WORDS TO BE OUTPUT
0490 4453     JMS I  XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0491 5301     JMP      TRAN2*2 /LOOP ON CURRENT TEST
    
```

```

/DOES TRANSFER TO OUTPUT REGISTER CHANGE AC (WITH AC=7777,DBS0)
0516 3036     TRANS, DCA      TYRFLG /CLEAR ERROR FLAG
0517 3037     DCA      LPCNT  /SET ITERATION COUNT TO 4096(DECIMAL)
0518 6007     CAF      /INITIALIZE INTERFACE
0519 7340     CLA CLL CMA  /SET AC =7777
0520 4446     DBB0     /BIT SET OUTPUT REGISTER
0521 3027     DCA      DATA1 /SAVE AC CONTENTS
0522 1027     TAD      DATA1 /GET AC CONTENTS
0523 7040     CMA      /COMPLEMENT DATA TO TEST FOR 7777
0524 7650     SNA CLA  /DID AC CHANGE
0525 5334     JMP      ,+5    /DATA CORRECT, CONTINUE
0526 4452     JMS I  XERROR /NO, ERROR
0527 3731     TRAN3=1 /DBS0 CHANGED AC
0528 3525     DH2=1  /"AC CONTENTS"
0529 7777     -1      /NUMBER OF WORDS TO BE OUTPUT
0530 4453     JMS I  XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0531 5320     JMP      TRANS*2 /LOOP ON CURRENT TEST
    
```

```

/DOES TRANSFER TO OUTPUT REGISTER CHANGE AC (WITH AC=7777, DBC0)
/
0536 3036 TRAN4, DCA TYPFLG /CLEAR ERROR FLAG
0537 3037 DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
0542 6007 CAF /INITIALIZE INTERFACE
0541 7340 CLA CLL CMA /SET AC =7777
0542 4445 DBC0 /BIT CLEAR OUTPUT REGISTER
0543 3027 DCA DATA1 /SAVE AC CONTENTS
0544 1027 TAD DATA1 /GET AC CONTENTS
0545 7040 CMA /COMPLIMENT DATA TO TEST FOR 7777
0546 7650 SNA CLA /DID AC CHANGE
0547 5354 JMP ,*5 /DATA CORRECT, CONTINUE
0550 4452 JMS I XERROR /NO, ERROR
0551 3741 TRAN4E=1 /"DBC0 CHANGED AC"
0552 3525 DH2=1 /"AC CONTENTS"
0553 7777 =1 /NUMBER OF WORDS TO BE OUTPUT
0554 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0555 5340 JMP TRAN4+2 /LOOP ON CURRENT TEST
/

```

```

/DOES TRANSFER TO INPUT REGISTER CHANGE AC (WITH AC=7777, DBC1)
/
0556 3036 TRAN5, DCA TYPFLG /CLEAR ERROR FLAG
0557 3037 DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
0562 6007 CAF /INITIALIZE INTERFACE
0561 7340 CLA CLL CMA /SET AC =7777
0562 4443 DBC1 /BIT CLEAR INPUT REGISTER
0563 3027 DCA DATA1 /SAVE AC CONTENTS
0564 1027 TAD DATA1 /GET AC CONTENTS
0565 7040 CMA /COMPLIMENT DATA TO TEST FOR 7777
0566 7650 SNA CLA /DID AC CHANGE
0567 5374 JMP ,*5 /DATA CORRECT, CONTINUE
0570 4452 JMS I XERROR /NO, ERROR
0571 3751 TRAN5E=1 /"DBC1 CHANGED AC"
0572 3525 DH2=1 /"AC CONTENTS"
0573 7777 =1 /NUMBER OF WORDS TO BE OUTPUT
0574 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0575 5360 JMP TRAN5+2 /LOOP ON CURRENT TEST
0576 5777 JMP TRAN6 /GO TO NEXT TEST
0577 5600 PAGE
0600

```

```

/DOES TRANSFER TO OUTPUT REGISTER CHANGE AC (WITH AC=0, DBC0)
/
0600 3036 TRAN6, DCA TYPFLG /CLEAR ERROR FLAG
0601 3037 DCA LPCNT /SET ITERATION COUNT TO 4096 (DECIMAL)
0602 6007 CAF /INITIALIZE INTERFACE
0603 4446 DBC0 /BIT SET OUTPUT REGISTER
0604 3027 DCA DATA1 /SAVE AC CONTENTS
0605 1027 TAD DATA1 /GET AC CONTENTS
0606 7650 SNA CLA /HAS AC CHANGED
0607 5214 JMP ,*5 /DATA CORRECT, CONTINUE
0612 4452 JMS I XERROR /AC CHANGED, ERROR
0611 3731 TRAN6E=1 /"DBC0 CHANGED AC"
0612 3525 DH2=1 /"AC CONTENTS"
0613 7777 =1 /NUMBER OF WORDS TO BE OUTPUT
0614 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0615 5202 JMP TRAN6+2 /LOOP ON CURRENT TEST
/

```

```

/DOES TRANSFER TO OUTPUT REGISTER CHANGE AC (WITH AC=0, DBC0)
/
0616 3036 TRAN7, DCA TYPFLG /CLEAR ERROR FLAG
0617 3037 DCA LPCNT /SET ITERATION COUNT TO 4096 (DECIMAL)
0620 6007 CAF /INITIALIZE INTERFACE
0621 4445 DBC0 /BIT CLEAR OUTPUT REGISTER
0622 3027 DCA DATA1 /SAVE AC CONTENTS
0623 1027 TAD DATA1 /GET AC CONTENTS
0624 7650 SNA CLA /IS AC STILL 0
0625 5232 JMP ,*5 /DATA CORRECT, CONTINUE
0626 4452 JMS I XERROR /NO, ERROR
0627 3741 TRAN7E=1 /"DBC0 CHANGED AC"
0630 3525 DH2=1 /"AC CONTENTS"
0631 7777 =1 /NUMBER OF WORDS TO BE OUTPUT
0632 4453 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, DBC1)
/
0634 3036 TRAN8, DCA TYPFLG /CLEAR ERROR FLAG
0635 3037 DCA LPCNT /SET ITERATION COUNT TO 4096 (DECIMAL)
0636 6007 CAF /INITIALIZE INTERFACE
0637 4443 DBC1 /BIT CLEAR INPUT REGISTER
0642 3027 DCA DATA1 /SAVE AC CONTENTS
0641 1027 TAD DATA1 /GET AC CONTENTS
0642 7650 SNA CLA /IS AC STILL 0
0643 5230 JMP ,*5 /DATA CORRECT, CONTINUE
0644 4452 JMS I XERROR /AC CHANGED, ERROR
0645 3751 TRAN8E=1 /"DBC1 CHANGED AC"
0646 3525 DH2=1 /"AC CONTENTS"
0647 7777 =1 /NUMBER OF WORDS TO BE OUTPUTED
0652 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
0651 5236 JMP TRAN8+2 /LOOP ON CURRENT TEST
0652 5777 JMP OUT1 /GO TO NEXT TEST
0653 1000 PAGE
0600

```



```

/
/CAN ALL BITS IN OUTPUT REGISTER BE SET (DBSO)
/DOES INITIALIZE CLEAR OUTPUT REGISTER
/
1009 3036 OUT1, DCA TYPFLG /CLEAR ERROR FLAG
1009 3037 DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
1002 6007 CAF /INITIALIZE INTERFACE
1003 7340 CLA CLL CMA /SET AC =7777
1004 4446 DBSO /BIT SET OUTPUT REGISTER
1005 4447 DBRO /READ OUTPUT REGISTER
1006 3027 DCA DATA1 /SAVE REGISTER DATA
1007 1027 TAD DATA1 /GET REGISTER DATA
1010 7040 CMA /COMPLIMENT DATA TO TEST FOR 7777
1011 7050 SNA CLA /IS REGISTER=7777
1012 5257 JMP ,+5 /DATA CORRECT, CONTINUE
1013 4452 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 4447 DBRO /READ OUTPUT REGISTER
1021 3027 DCA DATA1 /SAVE REGISTER DATA
1022 1027 TAD DATA1 /GET REGISTER DATA
1023 7650 SNA CLA /WAS REGISTER CLEARED
1024 5231 JMP ,+5 /DATA CORRECT, CONTINUE
1025 4452 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 4453 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 3036 OUT2, DCA TYPFLG /CLEAR ERROR FLAG
1034 3037 DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
1035 7340 CLA CLL CMA /SET AC =7777
1036 4446 DBSO /BIT SET OUTPUT REGISTER
1037 4445 DBCO /BIT CLEAR OUTPUT REGISTER
1040 7300 CLA CLL
1041 4447 DBRO /READ OUTPUT REGISTER
1042 3027 DCA DATA1 /SAVE REGISTER DATA
1043 1027 TAD DATA1 /GET REGISTER DATA
1044 7650 SNA CLA /WAS OUTPUT REGISTER CLEARED
1045 5252 JMP ,+5 /DATA CORRECT, CONTINUE
1046 4452 JMS I XERROR /NO, ERROR
1047 3645 INIT1E-1 /"OUTPUT REGISTER NOT CLEARED"
1050 3515 DH1-1 /"REGISTER DATA"
1051 7777 -1 /NUMBER OF WORDS TO BE OUTPUT
1052 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
1053 5235 JMP OUT2+2 /LOOP ON CURRENT TEST

```

```

/
/CAN EACH BIT OF OUTPUT REGISTER BE SET
/INDEPENDENTLY (DBSO)
/
1054 3036 OUT3, DCA TYPFLG /CLEAR ERROR FLAG
1055 1177 TAD [=62 /SET ITERATION COUNT
1056 3037 DCA LPCNT /TO 50(DECIMAL)
1057 3027 DCA DATA1 /CLEAR TEST DATA
1060 6007 OUT3A, CAF /INITIALIZE INTERFACE
1061 1027 TAD DATA1 /GET TEST DATA
1062 4446 DBSO /BIT SET OUTPUT REGISTER
1063 7300 CLA CLL
1064 4447 DBRO /READ OUTPUT REGISTER
1065 3030 DCA DATA2 /SAVE REGISTER DATA
1066 1027 TAD DATA1 /GET TEST DATA
1067 7041 CIA
1070 1030 TAD DATA2 /COMPARE TO REGISTER CONTENTS
1071 7650 SNA CLA /DO THEY COMPARE
1072 5277 JMP ,+5 /DATA CORRECT, CONTINUE
1073 4452 JMS I 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 4494 JMS I XLOOP2 /TEST FOR LOOP ON SAME DATA, ESCAPE ON DATA ERROR
1100 5260 JMP OUT3A /LOOP WITH SAME DATA
1101 2027 ISE DATA1 /INCREMENT DATA PATTERN
1102 5260 JMP OUT3A /CONTINUE TEST
1103 4453 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 3036 OUT4, DCA TYPFLG /CLEAR ERROR FLAG
1106 1177 TAD [=62 /SET ITERATION COUNT
1107 3037 DCA LPCNT /TO 50(DECIMAL)
1110 3027 DCA DATA1 /CLEAR TEST DATA
1111 6007 OUT4A, CAF /INITIALIZE INTERFACE
1112 1027 TAD DATA1 /GET MASK
1113 7040 CMA /COMPLIMENT TO GET EXPECTED RESULT
1114 3030 DCA DATA2 /SAVE EXPECTED RESULT
1115 7040 CMA /SET OUTPUT REGISTER TO 7777
1116 4446 DBSO /BIT SET OUTPUT REGISTER
1117 7300 CLA CLL
1120 1027 TAD DATA1 /GET PATTERN TO CLEAR OUTPUT REGISTER
1121 4445 DBCO /BIT CLEAR OUTPUT REGISTER
1122 4447 DBRO /READ OUTPUT REGISTER
1123 3031 DCA DATAS /SAVE REGISTER DATA
1124 1030 TAD DATA2 /GET EXPECTED RESULT
1125 7041 CIA
1126 1031 TAD DATAS /COMPARE TO RECEIVED DATA
1127 7650 SNA CLA /WERE CORRECT BITS IN OUTPUT REGISTER CLEARED
1130 5335 JMP ,+5 /DATA CORRECT, CONTINUE
1131 4452 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 4454      JMS I XLOOP2    /TEST FOR LOOP ON SAME DATA,ESCAPE ON DATA ERROR
1136 5311      JMP OUT4A       /LOOP WITH SAME DATA
1137 2027      ISE DATA1     /INCREMENT DATA PATTERN
1140 5311      JMP OUT4A       /CONTINUE TEST
1141 4453      JMS I XLOOP1    /CHECK FOR LOOP ON CURRENT TEST
1142 5310      JMP OUT4+3      /LOOP ON CURRENT TEST
1143 5777      JMP OUT5
1177 1200

```

PAGE

```

/
/ WITH OUTPUT REGISTER CLEARED; DOES CLEARING
/ OUTPUT REGISTER CHANGE ANY BIT IN OUTPUT
/
1200 3036      OUT5, DCA TYPFLG    /CLEAR ERROR FLAG
1201 1177      TAD I=02       /SET ITERATION COUNT
1202 3037      DCA LPCNT     /TO 50 (DECIMAL)
1203 3027      DCA DATA1    /CLEAR TEST DATA
1204 3030      DCA DATA2    /CLEAR EXPECTED RESULT
1205 6007      OUT5A, CAF      /INITIALIZE INTERFACE
1206 1027      TAD DATA1    /GET TEST DATA
1207 4445      DBCO        /BIT CLEAR OUTPUT REGISTER
1210 7300      CLA CLL
1211 4447      DBRO
1212 3031      DCA DATA3    /READ OUTPUT REGISTER
1213 1031      TAD DATA3    /SAVE REGISTER DATA
1214 7656      SNA CLA     /GET REGISTER DATA
1215 5222      JMP ,+8       /IS OUTPUT REGISTER 0
1216 4492      JMS I XERROR  /DATA CORRECT, CONTINUE
1217 3767      OUT4E-1     /NO, ERROR
1220 3534      DH3-1      /"DBCO ERROR"
1221 7775      -3              /"MASK EXPECTED RECEIVED"
1222 4454      JMS I XLOOP2    /NUMBER OF DATA WORDS
1223 5205      JMP OUT5A       /TEST FOR LOOP ON SAME DATA
1224 2027      ISE DATA1    /LOOP WITH SAME DATA
1225 5205      JMP OUT5A       /INCREMENT DATA PATTERN
1226 4453      JMS I XLOOP1    /CONTINUE
1227 5203      JMP OUT5+3      /CHECK FOR LOOP ON CURRENT TEST

```

```

/
/ DOES SETTING OUTPUT REGISTER TWICE WITH SAME
/ DATA CHANGE OUTPUT REGISTER
/
1230 3036      OUT6, DCA TYPFLG    /CLEAR ERROR FLAG
1231 1177      TAD I=02       /SET ITERATION COUNT
1232 3037      DCA LPCNT     /TO 50 (DECIMAL)
1233 3027      DCA DATA1    /CLEAR TEST DATA
1234 6007      OUT6A, CAF      /INITIALIZE INTERFACE
1235 1027      TAD DATA1    /GET TEST DATA
1236 4446      DBS0        /BIT SET OUTPUT REGISTER
1237 4446      DBS0        /BIT SET OUTPUT REGISTER
1240 7300      CLA CLL
1241 4447      DBRO
1242 3030      DCA DATA2    /READ OUTPUT REGISTER
1243 1027      TAD DATA1    /SAVE REGISTER DATA
1244 7041      CIA         /GET TEST DATA
1245 1030      TAD DATA2    /COMPARE TO REGISTER DATA
1246 7650      SNA CLA     /ARE THEY THE SAME
1247 5254      JMP ,+8       /DATA CORRECT, CONTINE
1250 4492      JMS I XERROR  /NO, ERROR
1251 3767      OUT4E-1     /"DBS0 ERROR"
1252 3560      DH4-1      /"EXPECTED RECEIVED"
1253 7776      -2              /NUMBER OF DATA WORDS
1254 4454      JMS I XLOOP2    /TEST FOR LOOP ON SAME DATA
1255 5234      JMP OUT6A       /LOOP WITH SAME DATA
1256 2027      ISE DATA1    /INCREMENT DATA PATTERN
1257 5234      JMP OUT6A       /CONTINUE
1260 4453      JMS I XLOOP1    /CHECK FOR LOOP ON CURRENT TEST
1261 5233      JMP OUT6+3      /LOOP ON CURRENT TEST

```

```

/
/DOES READING OUTPUT REGISTER TWICE CHANGE
/OUTPUT REGISTER
/
1262 3036 OUT7, DCA TYRFLG /CLEAR ERROR FLAG
1263 1177 TAD C=62 /SET ITERATION COUNT
1264 3037 DCA LRCNT /TO 50 (DECIMAL)
1265 3027 DCA DATA1 /CLEAR TEST DATA
1266 6007 OUT7A, CAF /INITIALIZE INTERFACE
1267 1027 TAD DATA1 /GET TEST DATA
1270 4446 DBSO /BIT SET OUTPUT REGISTER
1271 7300 CLA CLL
1272 4447 DBRO /READ OUTPUT REGISTER
1273 7300 CLA CLL
1274 4447 DBRO /READ OUTPUT REGISTER
1275 3030 DCA DATA2 /SAVE REGISTER DATA
1276 1027 TAD DATA1 /GET TEST DATA
1277 7041 CIA
1280 1030 TAD DATA2 /COMPARE TO REGISTER DATA
1281 7650 SNA CLA /ARE THEY THE SAME
1282 5307 JMP ,+5 /DATA CORRECT, CONTINUE
1283 4452 JMS I XERROR /NO. ERROR
1284 3775 OUT7E-1 /"DBRO ERROR"
1285 3560 DH4-1 /"EXPECTED RECEIVED"
1286 7775 -2 /NUMBER OF DATA WORDS
1287 4454 JMS I XLOOP2 /TEST FOR LOOP WITH SAME DATA
1288 5266 JMP OUT7A /LOOP WITH SAME DATA
1289 2027 ISE DATA1 /INCREMENT DATA PATTERN
1290 5266 JMP OUT7A /CONTINUE
1291 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
1292 5265 JMP OUT7+3 /LOOP ON CURRENT TEST

```

```

/
/DOES CLEARING OUTPUT REGISTER TWICE
/CHANGE ANY BIT IN OUTPUT REGISTER
/
1315 3036 OUT8, DCA TYRFLG /CLEAR ERROR FLAG
1316 1177 TAD C=62 /SET ITERATION COUNT
1317 3037 DCA LRCNT /TO 50 (DECIMAL)
1320 3027 DCA DATA1 /CLEAR TEST DATA
1321 3030 DCA DATA2 /CLEAR EXPECTED RESULT
1322 6007 OUT8A, CAF /INITIALIZE INTERFACE
1323 1027 TAD DATA1 /GET TEST DATA
1324 4446 DBSO /BIT SET OUTPUT REGISTER
1325 4445 DBCO /BIT CLEAR OUTPUT REGISTER
1326 4445 DBCO /BIT CLEAR OUTPUT REGISTER
1327 7300 CLA CLL
1330 4447 DBRO /READ OUTPUT REGISTER
1331 3031 DCA DATA3 /SAVE REGISTER DATA
1332 1031 TAD DATA3 /GET REGISTER DATA
1333 7650 SNA CLA /IS REGISTER 0
1334 5341 JMP ,+5 /DATA CORRECT, CONTINUE
1335 4492 JMS I XERROR /NO. ERROR
1336 3767 OUT4E-1 /"DBCO ERROR"
1337 3534 DH3-1 /"MASK EXPECTED RECEIVED"
1340 7775 -5 /NUMBER OF DATA WORDS
1341 4454 JMS I XLOOP2 /TEST FOR LOOP WITH SAME DATA
1342 5322 JMP OUT8A /LOOP WITH SAME DATA
1343 2027 ISE DATA1 /INCREMENT DATA PATTERN
1344 5322 JMP OUT8A /CONTINUE
1345 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
1346 5320 JMP OUT8+3 /LOOP ON CURRENT TEST
1347 5777 JMP IN1
1400 PAGE

```

```

/
/CAN ALL BITS IN INPUT REGISTER BE SET
/DOES INITIALIZE CLEAR INPUT REGISTER
/
1400 3036 IN1, DCA TYPFLG /CLEAR ERROR FLAG
1401 3037 DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
1402 6007 CAF /INITIALIZE INTERFACE
1403 7340 IN1A, CLA CLL CMA /SET AC #7777
1404 4446 DBSO /BIT SET OUTPUT REGISTER
1405 7300 CLA CLL
1406 4444 DBRI /READ INPUT REGISTER
1407 3027 DCA DATA1 /SAVE REGISTER DATA
1410 1027 TAD DATA1 /GET REGISTER DATA
1411 7040 CMA /COMPLIMENT TO TEST FOR 7777
1412 7650 SNA CLA /HAS INPUT REGISTER SET TO 7777
1413 5220 JMP ,+5 /DATA CORRECT, CONTINUE
1414 4452 JMS I XERROR /NO, ERROR
1415 4011 INSE=1 /"INPUT REGISTER NOT CORRECT"
1416 3515 DH1=1 /"REGISTER DATA"
1417 7777 =1 /NUMBER OF WORDS TO BE OUTPUT
1420 6007 IN1B, CAF /INITIALIZE INTERFACE
1421 4444 DBRI /READ INPUT REGISTER
1422 3027 DCA DATA1 /SAVE REGISTER DATA
1423 1027 TAD DATA1
1424 7650 SNA CLA
1425 5232 JMP ,+5 /DATA CORRECT, CONTINUE
1426 4452 JMS I XERROR /NO, ERROR
1427 3661 INITZE=1 /"INPUT REGISTER NOT CLEARED"
1430 3515 DH1=1 /"REGISTER DATA"
1431 7777 =1 /NUMBER OF WORDS TO BE OUTPUT
1432 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
1433 5203 JMP IN1A /LOOP ON CURRENT TEST

```

```

/
/CAN ALL BITS IN INPUT REGISTER BE CLEARED (DBCI)
/
1434 3036 IN2, DCA TYPFLG /CLEAR ERROR FLAG
1435 3037 DCA LPCNT /SET ITERATION COUNT TO 4096(DECIMAL)
1436 6007 CAF /INITIALIZE INTERFACE
1437 7340 CLA CLL CMA /SET AC #7777
1440 0035 AND FJUMPER /MASK TO TEST ONLY FLIPFLOP BITS
1441 3027 DCA DATA1 /SAVE MASK
1442 3030 DCA DATA2 /SAVE EXPECTED RESULT
1443 1027 TAD DATA1 /GET MASK
1444 4446 DBSO /BIT SET OUTPUT REGISTER
1445 4443 DBCI /BIT CLEAR INPUT REGISTER
1446 7300 CLA CLL
1447 4444 DBRI /READ INPUT REGISTER
1452 3031 DCA DATA3 /SAVE REGISTER DATA
1451 1031 TAD DATA3 /COMPARE TO REGISTER DATA
1452 7650 SNA CLA /WERE CORRECT BITS CLEARED
1453 5200 JMP ,+5 /DATA CORRECT, CONTINUE
1454 4452 JMS I XERROR /NO, ERROR
1455 4003 INZE=1 /"DBCI ERROR"
1456 3534 DH3=1 /"MASK EXPECTED RECEIVED"
1457 7775 =3 /NUMBER OF WORDS TO BE OUTPUT
1460 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
1461 5236 JMP IN2+2 /LOOP ON CURRENT TEST

```

```

/
/CAN EACH BIT OF INPUT REGISTER BE SET INDEPENDENTLY
/
1462 3036 IN3, DCA TYFFLG /CLEAR ERROR FLAG
1463 1177 TAD C=62 /SET ITERATION COUNT
1464 3037 DCA LPCNT /TO 50(DECIMAL)
1465 3027 DCA DATA1 /CLEAR TEST DATA
1466 6007 IN3A, CAF /INITIALIZE INTERFACE
1467 1027 TAD DATA1 /GET TEST DATA
1470 4446 DBSO /BIT SET OUTPUT REGISTER
1471 7300 CLA CLL
1472 4444 DBRI /READ INPUT REGISTER
1473 3030 DCA DATA2 /SAVE REGISTER DATA
1474 1027 TAD DATA1 /GET TEST DATA
1475 7041 CIA
1476 1030 TAD DATA2 /COMPARE TO RECEIVED DATA
1477 7650 SNA CLA /ARE THEY THE SAME
1502 5305 JMP ,+5 /DATA CORRECT, CONTINUE
1501 4452 JMS I XERROR /NO, ERROR
1502 4011 INSE=1 /"INPUT REGISTER DATA ERROR"
1503 3560 DH4-1 /"EXPECTED RECEIVED"
1504 7776 =2 /NUMBER OF WORDS TO BE OUTPUT
1505 4454 JMS I XLOOP2 /TEST FOR LOOP ON SAME DATA, ESCAPE ON DATA ERROR
1506 5266 JMP IN3A /LOOP WITH SAME DATA
1507 2027 ISZ DATA1 /INCREMENT DATA PATTERN
1512 5266 JMP IN3A /CONTINUE TEST
1511 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
1512 5265 JMP IN3+3 /LOOP ON CURRENT TEST
1513 5777 JMP IN3
1577 1600 PAGE
1600

```

```

/
/VERIFY THAT ALL LATCHING INPUT LINES HOLD DATA
/
1600 3036 IN3, DCA TYFFLG /CLEAR ERROR FLAG
1601 1177 TAD C=62 /SET ITERATION COUNT
1602 3037 DCA LPCNT /TO 50(DECIMAL)
1603 3031 IN3A, DCA DATA3 /CLEAR TEST DATA
1604 6007 CAF /INITIALIZE INTERFACE
1605 1031 TAD DATA3 /GET TEST DATA
1606 0035 AND FJUMPER /MASK OFF NON LATCHING BITS
1607 3027 DCA DATA1 /SAVE AS EXPECTED RESULT
1612 1027 TAD DATA1 /GET TEST DATA
1611 7450 SNA /ARE ANY BITS TO BE TESTED
1612 5233 JMP IN3C /NO, GET NEXT DATA WORD
1613 4446 DBSO /BIT SET OUTPUT REGISTER
1614 4445 DBCO /BIT CLEAR OUTPUT REGISTER
1615 7300 CLA CLL
1616 4444 DBRI /READ INPUT REGISTER
1617 3030 DCA DATA2 /SAVE REGISTER DATA
1620 1027 TAD DATA1 /GET EXPECTED RESULT
1621 7041 CIA
1622 1030 TAD DATA2 /COMPARE TO RECEIVED DATA
1623 7650 SNA CLA /ARE THEY THE SAME
1624 5231 JMP ,+5 /DATA CORRECT, CONTINUE
1625 4452 JMS I XERROR /NO, ERROR
1626 4026 IN4E=1 /"LATCH ERROR"
1627 3560 DH4-1 /"EXPECTED RECEIVED"
1630 7776 =2 /NUMBER OF WORDS TO BE OUTPUT
1631 4454 JMS I XLOOP2 /TEST FOR LOOP ON SAME DATA, ESCAPE ON DATA ERROR
1632 5204 JMP IN3A /LOOP WITH SAME DATA
1633 2031 IN3C, ISZ DATA3 /INCREMENT DATA PATTERN
1634 5204 JMP IN3A /CONTINUE TEST
1635 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
1636 5203 JMP IN3+3 /LOOP ON CURRENT TEST

```

```

/VERIFY ALL NON LATCHING DATA BITS DO NOT HOLD DATA
1637 3036 IN6, DCA TYPFLG /CLEAR ERROR FLAG
1642 1177 TAD C=62 /SET ITERATION COUNT
1641 3037 DCA LPCNT /TO 50(DECIMAL)
1642 3032 DCA DATA4 /CLEAR TEST DATA
1643 3030 DCA DATA2 /CLEAR EXPECTED RESULT
1644 6007 IN6A, CAF /INITIALIZE INTERFACE
1645 1035 TAD FJUMPER /GET MASK FOR NON LATCHING BITS
1646 7040 CMA /CHANGE TO MASK OFF LATCHING BITS
1647 3032 AND DATA4
1650 3027 DCA DATA1 /SAVE FOR TRANSMISSION
1651 1027 TAD DATA1 /GET TEST DATA
1652 7450 SNA /ARE ANY BITS TO BE TESTED
1653 5272 JMP IN6C /NO GET NEXT DATA WORD
1654 4446 DBSO /BIT SET OUTPUT REGISTER
1655 4445 DBCO /BIT CLEAR OUTPUT REGISTER
1656 7300 CLA CLL
1657 4444 DBRI /READ INPUT REGISTER
1660 3031 DCA DATA3 /SAVE REGISTER DATA
1661 1031 TAD DATA3 /GET RECEIVED DATA
1662 7650 SNA CLA /DID ANY BITS HOLD DATA
1663 5270 JMP ,+5 /DATA CORRECT, CONTINUE
1664 4452 JMS I XERROR /YES, ERROR
1665 4026 IN4E=1 /"LATCH ERROR"
1666 3534 DHS=1 /"MASK EXPECTED RECEIVED"
1667 7775 =3 /NUMBER OF WORDS TO BE OUTPUT
1670 4454 JMS I XLOOP2 /TEST FOR LOOP ON SAME DATA, ESCAPE ON DATA ERROR
1671 5244 JMP IN6A /LOOP WITH SAME DATA
1672 2032 IN6C, ISE DATA4 /INCREMENT DATA PATTERN
1673 5244 JMP IN6A /CONTINUE TEST
1674 4453 JMS I XLOOP1 /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 3036 IN7, DCA TYPFLG /CLEAR ERROR FLAG
1677 1177 TAD C=62 /SET ITERATION COUNT
1680 3037 DCA LPCNT /TO 50(DECIMAL)
1681 3032 DCA DATA4 /CLEAR TEST DATA
1682 6007 IN7A, CAF /INITIALIZE INTERFACE
1683 1035 TAD FJUMPER /GET MASK FOR LATCHING BITS
1684 3032 AND DATA4 /MASK OFF NON LATCHING BITS
1685 3027 DCA DATA1 /SAVE FOR TRANSMISSION
1686 3030 DCA DATA2 /EXPECTED RESULT
1687 1027 TAD DATA1 /SET OUTPUT REGISTER=7777
1688 4446 DBSO /BIT SET OUTPUT REGISTER
1689 4445 DBCO /BIT CLEAR OUTPUT REGISTER
1690 7300 CLA CLL
1691 1027 TAD DATA1 /GET TEST DATA
1692 4443 DBCI /BIT CLEAR INPUT REGISTER
1693 7300 CLA CLL
1694 4444 DBRI /READ INPUT REGISTER
1695 3031 DCA DATA3 /SAVE REGISTER DATA
1696 1031 TAD DATA3 /COMPARE TO RECEIVED DATA
1697 7650 SNA CLA /ARE THEY THE SAME
1698 5327 JMP ,+5 /DATA CORRECT, CONTINUE
1699 4452 JMS I XERROR /NO, ERROR
1700 4026 IN4E=1 /"LATCH ERROR"
1701 3534 DHS=1 /"MASK EXPECTED RECEIVED"
1702 7775 =3 /NUMBER OF WORDS TO BE OUTPUT
1703 4454 JMS I XLOOP2 /TEST FOR LOOP ON SAME DATA, ESCAPE ON DATA ERROR
1704 5302 JMP IN7A /LOOP WITH SAME DATA
1705 2032 IN7C, ISE DATA4 /INCREMENT DATA PATTERN
1706 5302 JMP IN7A /CONTINUE TEST
1707 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
1708 5301 JMP IN7+3 /LOOP ON CURRENT TEST
1709 5777 JMP IN8 /GO TO NEXT TEST
1710 2000 PAGE
1711 2000
    
```

```

/ WITH THE INPUT REGISTER CLEARED, DOES CLEARING
/ THE INPUT REGISTER SET ANY BIT IN INPUT
/
2000 3036 148, DCA TYPFLG /CLEAR ERROR FLAG
2001 1177 TAD [-62 /SET ITERATION COUNT
2002 3037 DCA LPCNT /TO 50 (DECIMAL)
2003 3027 DCA DATA1 /CLEAR TEST DATA
2004 3030 DCA DATA2 /CLEAR EXPECTED RESULT
2005 6007 148A, CAF /INITIALIZE INTERFACE
2006 1027 TAD DATA1 /GET TEST DATA
2007 4443 DBCI /BIT CLEAR INPUT REGISTER
2010 7300 CLA CLL
2011 4444 DBRI /READ INPUT REGISTER
2012 3031 DCA DATA3 /SAVE REGISTER DATA
2013 1031 TAD DATA3 /GET REGISTER DATA
2014 7650 SNA CLA /IS INPUT REGISTER 0
2015 5222 JMP ,45 /DATA CORRECT, CONTINUE
2016 4452 JMS I XERROR /NO, ERROR
2017 4003 INZE-1 /"DBCI ERROR"
2020 3534 DHS-1 /"MASK EXPECTED RECEIVED"
2021 7775 -3 /NUMBER OF DATA WORDS
2022 4454 JMS I XLOOP2 /TEST FOR LOOP WITH SAME DATA
2023 5205 JMP IN8A /LOOP WITH SAME DATA
2024 2027 ISZ DATA1 /INCREMENT DATA PATTERN
2025 5205 JMP IN8A /CONTINUE
2026 4453 JMS I XLOOP1 /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
/
2030 3036 IN9, DCA TYPFLG /CLEAR ERROR FLAG
2031 1177 TAD [-62 /SET ITERATION COUNT
2032 3037 DCA LPCNT /TO 50 (DECIMAL)
2033 3027 DCA DATA1 /CLEAR TEST DATA
2034 6007 IN9A, CAF /INITIALIZE INTERFACE
2035 1027 TAD DATA1 /GET TEST DATA
2036 4446 DBSO /BIT SET OUTPUT REGISTER
2037 7300 CLA CLL
2040 4444 DBRI /READ INPUT REGISTER
2041 7300 CLA CLL
2042 4444 DBRI /READ INPUT REGISTER
2043 3030 DCA DATA2 /SAVE REGISTER DATA
2044 1027 TAD DATA1 /GET TEST DATA
2045 7041 CIA
2046 1030 TAD DATA2 /COMPARE TO REGISTER DATA
2047 7650 SNA CLA /ARE THEY THE SAME
2050 5255 JMP ,45 /DATA CORRECT, CONTINUE
2051 4452 JMS I XERROR /NO, ERROR
2052 4034 IN9E-1 /"DBRI ERROR"
2053 3560 DH4-1 /"EXPECTED RECEIVED"
2054 7776 -2 /NUMBER OF DAT WORDS
2055 4454 JMS I XLOOP2 /TEST FOR LOOP WITH SAME DATA
2056 5234 JMP IN9A /LOOP WITH SAME DATA
2057 2027 ISZ DATA1 /INCREMENT DATA PATTERN
2060 5234 JMP IN9A /CONTINUE
2061 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
2062 5233 JMP IN9+3 /LOOP ON CURRENT TEST

```

```

/DOES CLEARING INPUT REGISTER TWICE SET ANY BIT
/IN INPUT REGISTER
2063 3036 IN10, DCA TYPFLG /CLEAR ERROR FLAG
2064 1177 TAD C=62 /SET ITERATION COUNT
2065 3037 DCA LPCNT /TO 50 (DECIMAL)
2066 3030 DCA DATA2 /CLEAR EXPECTED RESULT
2067 3027 DCA DATA1 /CLEAR TEST DATA
2070 6007 IN10A, CAF /INITIALIZE INTERFACE
2071 1027 TAD DATA1 /GET TEST DATA
2072 4446 DBSO /BIT SET OUTPUT REGISTER
2073 4445 DBCO /BIT CLEAR OUTPUT REGISTER
2074 4443 DBCI /BIT CLEAR INPUT REGISTER
2075 4443 DBCI /BIT CLEAR INPUT REGISTER
2076 7300 CLA CLL
2077 4444 DBRI /READ INPUT REGISTER
2100 3031 DCA DATA3 /SAVE REGISTER DATA
2101 1031 TAD DATA3 /GET REGISTER DATA
2102 7600 SNA CLA /IS INPUT REGISTER 0
2103 5310 JMP ,+8 /DATA CORRECT, CONTINUE
2104 4492 JMS I XERROR /NO, ERROR
2105 4003 IN2E-1 /"DBCI ERROR"
2106 3534 DHS-1 /"MASK EXPECTED RECEIVED"
2107 7775 -3 /NUMBER OF DATA WORDS
2110 4454 JMS I XLOOP2 /TEST FOR LOOP WITH SAME DATA
2111 5270 JMP IN10A /LOOP WITH SAME DATA
2112 2027 ISE DATA1 /INCREMENT DATA PATTERN
2113 5270 JMP IN10A /CONTINUE
2114 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
2115 5266 JMP IN10+3 /LOOP ON CURRENT TEST
2116 5777 JMP IN0U1 /GO TO NEXT TEST
2177 2200
PAGE
    
```

```

/WITH BOTH INPUT AND OUTPUT REGISTERS CLEARED
/DOES CLEARING OUTPUT SET
/ANY BIT IN INPUT
2200 3036 IN0U1, DCA TYPFLG /CLEAR ERROR FLAG
2201 1177 TAD C=62 /SET ITERATION COUNT
2202 3037 DCA LPCNT /TO 50 (DECIMAL)
2203 3027 DCA DATA1 /CLEAR TEST DATA
2204 3030 DCA DATA2 /CLEAR EXPECTED RESULT
2205 6007 IN0U1A, CAF /INITIALIZE INTERFACE
2206 1027 TAD DATA1 /GET TEST DATA
2207 4445 DBCO /BIT CLEAR OUTPUT REGISTER
2210 7300 CLA CLL
2211 4444 DBRI /READ INPUT REGISTER
2212 3031 DCA DATA3 /SAVE REGISTER DATA
2213 1031 TAD DATA3 /GET REGISTER DATA
2214 7600 SNA CLA /IS OUTPUT REGISTER 0
2215 5222 JMP ,+8 /DATA CORRECT, CONTINUE
2216 4452 JMS I XERROR /NO, ERROR
2217 3767 OUT4E-1 /"DBCO ERROR"
2220 3534 DHS-1 /"MASK EXPECTED RECEIVED"
2221 7775 -3 /NUMBER OF DATA WORDS
2222 4454 JMS I XLOOP2 /TEST FOR LOOP WITH SAME DATA
2223 5205 JMP IN0U1A /LOOP WITH SAME DATA
2224 2027 ISE DATA1 /INCREMENT DATA PATTERN
2225 5205 JMP IN0U1A /CONTINUE
2226 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
2227 5203 JMP IN0U1+3 /LOOP ON CURRENT TEST
    
```



```

/ WITH BOTH INPUT AND OUTPUT REGISTERS CLEARED
/ DOES CLEARING INPUT SET ANY BIT IN OUTPUT
/
2230 3036 INOU2, DCA TYPFLG /CLEAR ERROR FLAG
2231 1177 TAD L=62 /SET ITERATION COUNT
2232 3037 DCA LPCNT /TO 50 (DECIMAL)
2233 3027 DCA DATA1 /CLEAR TEST DATA
2234 3030 DCA DATA2 /CLEAR EXPECTED RESULT
2235 6007 INOU2A, CAF /INITIALIZE INTERFACE
2236 1027 TAD DATA1 /GET TEST DATA
2237 4443 DBCI /BIT CLEAR INPUT REGISTER
2240 7300 CLA CLL
2241 4447 DBRO /READ OUTPUT REGISTER
2242 3031 DCA DATA3 /SAVE REGISTER DATA
2243 1031 TAD DATA3 /GET REGISTER DATA
2244 7650 SNA CLA /IS OUTPUT REGISTER 0
2245 5252 JMP ,05 /DATA CORRECT, CONTINUE
2246 4452 JMS I XERROR /NO, ERROR
2247 4063 IN2E=1 /"DBCI ERROR"
2250 3534 DHS=1 /"MASK EXPECTED RECEIVED"
2251 7775 =3 /NUMBER OF DATA WORDS
2252 4454 JMS I XLOOP2 /TEST FOR LOOP WITH CURRENT DATA
2253 5235 JMP INOU2A /LOOP WITH SAME DATA
2254 2027 ISZ DATA1 /INCREMENT DATA PATTERN
2255 5235 JMP INOU2A /CONTINUE
2256 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
2257 5233 JMP INOU2+3 /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 3036 INOU3, DCA TYPFLG /CLEAR ERROR FLAG
2261 1177 TAD L=62 /SET ITERATION COUNT
2262 3037 DCA LPCNT /TO 50 (DECIMAL)
2263 3027 DCA DATA1 /CLEAR TEST DATA
2264 3030 DCA DATA2 /CLEAR EXPECTED RESULT
2265 6007 INOU3A, CAF /INITIALIZE INTERFACE
2266 7040 CMA /SET AC=7777
2267 4446 DBSO /BIT SET OUTPUT REGISTER
2270 4443 DBCI /BIT CLEAR INPUT REGISTER
2271 7300 CLA CLL
2272 1035 TAD FJUMPER /GET FLIPFLOP JUMPER MASK
2273 7040 CMA
2274 3030 DCA DATA2
2275 1027 TAD DATA1 /GET TEST DATA2
2276 7040 CMA /COMPLEMENT
2277 3030 AND DATA2 /AND WITH COMPLEMENT OF JUMPER MASK
2300 3030 DCA DATA2 /TO GET EXPECTED RESULT
2301 1027 TAD DATA1 /GET TEST DATA
2302 4445 DBCO /BIT CLEAR OUTPUT REGISTER
2303 7300 CLA CLL
2304 4444 DBRI /READ INPUT REGISTER

```

```

2305 3031 DCA DATA3 /SAVE REGISTER DATA
2306 1031 TAD DATA3 /GET REGISTER DATA
2307 7041 CIA
2310 1030 TAD DATA2 /COMPARE TO EXPECTED RESULT
2311 7650 SNA CLA /ARE THEY THE SAME
2312 5317 JMP ,05 /DATA CORRECT, CONTINUE
2313 4452 JMS I XERROR /NO, ERROR
2314 3767 OUT4E=1 /"DBCO ERROR"
2315 3534 DHS=1 /"MASK EXPECTED RECEIVED"
2316 7775 =3 /NUMBER OF DATA WORDS
2317 4454 JMS I XLOOP2 /TEST FOR LOOP WITH SAME DATA
2320 5265 JMP INOU3A /LOOP WITH SAME DATA
2321 2027 ISZ DATA1 /INCREMENT DATA PATTERN
2322 5265 JMP INOU3A /CONTINUE
2323 4453 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)
/
2325 3036 INOU4, DCA TYPFLG /CLEAR ERROR FLAG
2326 1177 TAD L=62 /SET ITERATION COUNT
2327 3037 DCA LPCNT /TO 50 (DECIMAL)
2328 3027 DCA DATA1 /CLEAR TEST DATA
2331 6007 INOU4A, CAF /INITIALIZE INTERFACE
2332 7040 CMA /SET AC TO 7777
2333 4446 DBSO /BIT SET OUTPUT REGISTER
2334 7300 CLA CLL
2335 1035 TAD FJUMPER /GET FLIPFLOP JUMPER MASK
2336 7040 CMA
2337 2027 AND DATA1 /COMBINE WITH MASK
2340 7040 CMA
2341 3030 DCA DATA2 /TO GET EXPECTED RESULT
2342 1027 TAD DATA1 /GET TEST DATA
2343 4445 DBCO /BIT CLEAR OUTPUT REGISTER
2344 7300 CLA CLL
2345 4444 DBRI /READ INPUT REGISTER
2346 3031 DCA DATA3 /SAVE REGISTER DATA
2347 1030 TAD DATA2 /GET EXPECTED RESULT
2350 7041 CIA
2351 1031 TAD DATA3 /COMPARE TO RECEIVED DATA
2352 7650 SNA CLA /ARE THEY THE SAME
2353 5360 JMP ,05 /DATA CORRECT, CONTINUE
2354 4452 JMS I XERROR /NO, ERROR
2355 3767 OUT4E=1 /"DBCO ERROR"
2356 3534 DHS=1 /"MASK EXPECTED RECEIVED"
2357 7775 =3 /NUMBER OF DATA WORDS
2360 4454 JMS I XLOOP2 /TEST FOR LOOP WITH CURRENT DATA
2361 5331 JMP INOU4A /LOOP WITH SAME DATA
2362 2027 ISZ DATA1 /INCREMENT DATA PATTERN
2363 5331 JMP INOU4A /CONTINUE
2364 4453 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
2400 3036 INT1, DCA TYPELG /CLEAR ERROR FLAG
2401 1177 TAD L=02 /SET ITERATION COUNT
2402 3037 DCA LPCNT /TO 50(DECIMAL)
2403 3030 DCA DATA2
2404 6007 INT1A, CAF /INITIALIZE INTERFACE
2405 1030 TAD DATA2
2406 0034 AND IJUMPER
2407 7450 SNA
2410 5241 JMP INT1D
2411 3027 DCA DATA1 /SAVE TEST DATA
2412 1027 TAD DATA1
2413 4446 DBSO /BIT SET OUTPUT REGISTER
2414 7300 CLA CLL
2415 6003 SRQ /IS INTERRUPT ACTIVE
2416 5223 JMP ,+5 /NO, CONTINUE
2417 4452 JMS I XERROR /YES, ERROR
2420 4042 INT1E-1 /INTERRUPT ACTIVE:
2421 3515 DH1-1
2422 7777 -1 /NUMBER OF DATA WORDS TO BE OUTPUT
2423 4441 DBEI /ENABLE INTERFACE
2424 6003 SRQ /IS INTERRUPT ACTIVE
2425 5231 JMP INT1AE /NO, ERROR
2426 4442 DBSK /IS FLAG SET
2427 5246 JMP INT1BE /NO, ERROR
2430 5237 JMP INT10K /INTERRUPT ACTIVE, FLAG SET
2431 4442 INT1AE, DBSK /IS INTERFACE FLAG SET
2432 5283 JMP INT1CE
2433 4452 JMS I XERROR /NO, ERROR
2434 4042 INT1E-1
2435 3515 DH1-1
2436 7777 -1 /NUMBER OF WORDS TO BE OUTPUT
2437 4454 INT10K, JMS I XLOOP2 /TEST FOR LOOP ON SAME DATA, ESCAPE ON DATA ERROR
2440 5204 JMP INT1A /LOOP WITH SAME DATA
2441 2030 INT1D, ISZ DATA2 /INCREMENT DATA PATTERN
2442 5204 JMP INT1A /CONTINUE
2443 4453 JMS I XLOOP1 /CHECK FOR LOOP ON CURRENT TEST
2444 5203 JMP INT1+3 /LOOP ON CURRENT TEST
2445 5260 JMP INT3 /GO TO NEXT TEST
2446 4452 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 INT10K
2453 4452 INT1CE, JMS I XERROR /NO, ERROR
2454 4077 INT4E-1
2455 3515 DH1-1
2456 7777 -1 /NUMBER OF WORDS TO BE OUTPUT
2457 5237 JMP INT10K
    
```

```

/VERIFY THAT EACH BIT NOT JUMPERD TO SKIP DOES NOT
2460 3036 INT3, DCA TYPFLG /CLEAR ERROR FLAG
2461 1177 TAD L=62 /SET ITERATION COUNT
2462 3037 DCA LPCNT /TO 50 (DECIMAL)
2463 3030 DCA DATA2 /CLEAR TEST DATA
2464 6007 INT3A, CAF /INITIALIZE INTERFACE
2465 1034 TAD IJUMPER /GET JUMPER MASK
2466 7040 CMA /COMPLIMENT FOR NO SKIP BITS
2467 0030 AND DATA2 /GET BITS TO BE TESTED
2470 7450 SNA /ARE ANY BITS TO BE TESTED
2471 5306 JMP INT3C /NO, GET NEXT DATA PATTERN
2472 3027 DCA DATA1 /SAVE FOR OUTPUT
2473 1027 TAD DATA1 /GET TEST DATA
2474 4446 DBSO /BIT SET OUTPUT REGISTER
2475 7300 CLA CLL
2476 4442 DBSK /IS FLAG SET
2477 5304 JMP ,+5 /NO, CONTINUE
2500 4492 JMS I XERROR /YES, ERROR
2501 3674 INIT3E=1 /"SKIP FLAG SET"
2502 3515 DH1=1 /"REGISTER DATA"
2503 7777 -1
2504 4454 JMS I XLOOP2
2505 5244 JMP INT3A
2506 2030 INT3C, ISZ DATA2
2507 5244 JMP INT3A
2510 4493 JMS I XLOOP1
2511 5243 JMP INT3+3
2512 5777 JMP EPASS
    
```

```

/ERROR HANDLER
2577 3297 PAGE
2578 2600 ERROR, 0
2600 0000 CLA CLL
2601 7300 TAD I ERROR /GET POINTER TO ERROR MESSAGE
2602 1600 DCA MSG /SAVE POINTER
2603 3234 ISZ ERROR
2604 2200 TAD I ERROR /GET POINTER TO DATA HEADER
2605 1600 DCA DHDR /SAVE HEADER
2606 3236 ISZ ERROR /NUMBER OF WORDS TO BE OUTPUT
2607 2200 TAD I ERROR /GET NUMBER OF DATA WORDS TO BE TYPED
2610 1600 DCA DATCNT /SAVE
2611 3264 TAD ERROR /GET ADDRESS OF TEST THAT FAILED
2612 1200 TAD (=3
2613 1377 DCA LSTOGT
2614 3776 TAD LSTOGT
2615 1776 DCA ERRAD
2616 3266 LAS
2617 4425 AND SR01
2620 0334 SEA CLA
2621 7640 JMP EHALT
2622 5254 TAD TYPFLG
2623 1036 SEA CLA
2624 7640 JMP DATOUT
2625 5241 CMA
2626 7040 DCA TYPFLG /CLEAR ERROR FLAG
2627 3036 JMS I OCTASC
2630 4775 JMS I XPRINT
2631 4490 ERADR=1
2632 3024 JMS I XPRINT
2633 4450 MSG, 0
2634 0000 JMS I XPRINT
2635 4450 DHDR, 0
2636 0000 JMS I XPRINT /NUMBER OF WORDS TO BE OUTPUT
2637 4450 DATOUT, TAD DATCNT
2640 3512 CRLF=1
2641 1264 TAD SNA CLA
2642 7650 JMP EHALT
2643 5254 TAD DATAP
2644 1265 DCA POINT1
2645 3010 BITS, TAD I POINT1
2646 1410 JMS BITOUT
2647 4774 ISZ DATCNT
2650 2264 JMP BITS
2651 5246 JMS I XPRINT
2652 4450 CRLF=1
2653 3512 EHALT, LAS
2654 4425 AND SR00
2655 0333 SEA CLA
2656 7640 JMP ,+3
2657 5262 TAD ERRAD
2660 1266 HLT
2661 7402
    
```

```

2662 2200      ISE  ERROR
2663 5600      JMP  I  ERROR
2664 0000      DATCNT, 0
2665 0026      DATAP, DATA1-1
2666 0000      ERRAD, 0
          /
          /TEST FOR LOOP ON CURRENT TEST
          /
2667 0000      LOOP1, 0
2672 1036      TAD  TYPFLG
2671 7650      SNA  CLA
2672 5277      JMP  LP1EXA
2673 4425      LAS
2674 0341      AND  SR06
2675 7640      SZA  CLA
2676 5310      JMP  LP1EXX=1
2677 4425      LP1EXA, LAS
2700 0340      AND  SR05
2701 7640      SZA  CLA
2702 5305      JMP  LP1EXT
2703 2037      ISE  LPCNT
2704 5311      JMP  LP1EXX
2705 4425      LP1EXT, LAS
2706 0335      AND  SR02
2707 7650      SNA  CLA
2710 2267      ISE  LOOP1
2711 5667      LP1EXX, JMP I  LOOP1
          /
          /TEST FOR LOOP ON CURRENT DATA
          /
2712 0000      LOOP2, 0
2713 1036      TAD  TYPFLG
2714 7650      SNA  CLA
2715 5326      JMP  LP2EXT
2716 4425      LAS
2717 0341      AND  SR06
2720 7650      SNA  CLA
2721 5326      JMP  ,+5          /DATA CORRECT, CONTINUE
2722 1312      TAD  LOOP2
2723 1373      TAD  (5
2724 3312      DCA  LOOP2
2725 5712      JMP  I  LOOP2
2726 4425      LP2EXT, LAS
2727 0336      AND  SR03
2730 7650      SNA  CLA
2731 2312      ISE  LOOP2
2732 5712      JMP  I  LOOP2
2733 4000      SR00, 4000
2734 2000      SR01, 2000
2735 1000      SR02, 1000
2736 0400      SR03, 400
2737 0200      SR04, 200
2740 0100      SR05, 100
2741 0040      SR06, 40
    
```

```

          /OCTAL TO PACKED ASCII CONVERSION
2773 0005
2774 3031
2775 3000
2776 3027
2777 7775
3000 3000      PAGE /
3001 7300      OCTASC, 0
3002 1227      TAD  LSTDGT          /GET WORD TO BE CONVERTED
3003 7002      BSW          /SWAP HALVES, SEPARATE DIGITS,
3004 4212      JMS  SPLIT          /CONVERT MOST SIGNIFICANT
3005 3226      DCA  HSTDGT          /DIGITS TO ASCII
3006 1227      TAD  LSTDGT          /CONVERT LEAST SIGNIFICANT
3007 4212      JMS  SPLIT          /DIGITS TO ASCII
3010 3227      DCA  LSTDGT
3011 5600      JMP  I  OCTASC          /RETURN
3012 0000      SPLIT, 0
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      ERADR, TEXT /??
3026 4040
3027 4040
3030 4000
          /OUTPUT 12 BIT BINARY WORD
3031 0000      BITOUT, 0
3032 7421      MQL
3033 1374      TAD  (014          /SAVE DATA IN HQ
3034 3026      DCA  CNTR1          /SET UP TO OUTPUT
3035 7501      MGA          /12 BITS
3036 7104      CLL  RAL          /GET MSB INTO LINK
3037 7421      MQL          /SAVE REST OF WORD
3040 1373      TAD  ("1          /GET ASCII 1 INTO AC
3041 7420      SNL
3042 0372      AND  ("0
3043 4451      JMS  I  XTYPE          /NO, CHANGE TO ASCII 0
3044 2026      ISE  CNTR1          /OUTPUT BIT
3045 5235      JMP  BIT01          /CONTINUE
3046 1371      TAD  (240          /TYPE 2 SPACES
3047 4451      JMS  I  XTYPE          /AFTER LAST BIT HAS BEEN
3050 1371      TAD  (240          /OUTPUTTED
3051 4451      JMS  I  XTYPE
3052 5631      JMP  I  BITOUT          /RETURN
    
```

/CHARACTER STRING OUTPUT ROUTINE

```

3171 0240
3172 0260
3173 0261
3174 7764
3175 6060
3176 0707
3177 0077
3200 0000 PAGE /
PRINT, 0 CLA CLL
3201 7300 TAD I PRINT /GET POINTER TO MESSAGE
3202 1600 DCA POINT1 /SET UP AUTO-INDEX REGISTER
3203 3010 ISE PRINT /SET UP RETURN
3204 2200 TAD I POINT1 /GET PACKED WORD
3205 1410 MQL /SAVE IN MQ
3206 7421 MQA /GET WORD
3207 7581 BSW /SWAP HALVES
3210 7002 JMS TYPSET /DECODE AND OUTPUT
3211 4215 MQA /GET WORD
3212 7501 JMS TYPSET /DECODE AND OUTPUT
3213 4215 JMP PRINT+5 /CONTINUE
3214 5205
/UNPACK, DECODE, OUTPUT
/
3215 0000 TYPSET, 0
3216 0243 AND K0077 /MASK UNWANTED BITS
3217 7490 SNA /IS AQ=0
3220 5600 JMP I PRINT /YES, END OF MESSAGE, EXIT
3221 1244 TAD M40 /SUBTRACT 40
3222 7510 SPA /IS PACKED CHARACTER >40
3223 5226 JMP ,+3 /NO
3224 1250 TAD K240 /YES, CONVERT TO ASCII
3225 5241 JMP MTP /OUTPUT
3226 7001 IAC /ADD 1 TO AC
3227 7440 SEA /IS CHARACTER=57
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 SEA /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 4451 MTP, JMS I XTYPE /OUTPUT
3242 5615 JMP I TYPSET
3243 0077 K0077, 77
3244 7740 M40, -40
3245 0215 K215, 215
3246 0212 K212, 212
3247 0536 K336, 336
3250 0240 K240, 240

```

/OUTPUT ONE CHARACTER TO TTY

```

3251 0000 TYPE, 0
3252 6046 TLS
3253 6041 TSF
3254 5253 JMP ,+1
3255 7200 CLA
3256 5651 JMP I TYPE
/
/
3257 4425 EPASS, LAS
3260 0777 AND SR04
3261 7640 SEA CLA
3262 5776 JMP INIT1
3263 4425 LAS
3264 0775 AND SR05
3265 7640 SEA CLA
3266 5272 JMP EPAS1
3267 4450 JMS I XPRINT
3270 3274 MEP=1
3271 5776 JMP INIT1
3272 1374 EPAS1, TAD (207
3273 4251 JMS TYPE
3274 5776 JMP INIT1
3275 3736 MEP, TEXT /*DR/
3276 0422
3277 0000
3300 5776 APTPAS, JMP INIT1 /APT/

```

/TELETYPE MESSAGES

3374 2207  
3375 2740  
3376 0400  
3377 2737  
3400 3400  
3401 3736  
3402 2305  
3403 2440  
3404 2322  
3405 4006  
3406 1722  
3407 4004  
3408 0526  
3409 1103  
3410 0540  
3411 0317  
3412 0405  
3413 4001  
3414 1604  
3415 4003  
3416 1716  
3417 2400  
3418 3736  
3419 2305  
3420 2440  
3421 2322  
3422 4006  
3423 1722  
3424 4011  
3425 1624  
3426 0522  
3427 2225  
3428 2024  
3429 4012  
3430 2515  
3431 2005  
3432 2223  
3433 4001  
3434 1604  
3435 4003  
3436 1716  
3437 2400  
3438 3736  
3439 2305  
3440 2440  
3441 2327  
3442 1124  
3443 0310  
3444 0523  
3445 4006  
3446 1722

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/

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 2000  
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 2000

M3, TEXT /\*+SET SR FOR RUN AND CONT/

CRLF, TEXT /\*+//

/DATA HEADERS

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

DH0, 0  
DH1, TEXT /\*+REGISTER DATA/

DH2, TEXT /\*+AC CONTENTS/

DH3, TEXT /\*+MASK EXPECTED RECEIVED/

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

DH4, TEXT /\*EXPECTED RECEIVED/

DH5, TEXT /\*REGISTER DATA OUT DATA IN/

DH6, TEXT /\*AC CONTENTS DATA OUT DATA IN/

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

/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

INIT2E, TEXT /\*INPUT REG NOT CLEARED/

INIT3E, TEXT /\*SKIP FLAG SET/

TRAN1E, TEXT /\*DBRO DID NOT CLEAR AC/

3715 4001  
3716 0300  
3717 0402 TRAN2E, 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  
3732 4002 TRAN3E, TEXT /DBSO CHANGED AC/  
3733 2317  
3734 4003  
3735 1001  
3736 1607  
3737 0504  
3740 4001  
3741 0300 TRAN4E, TEXT /DBCO CHANGED AC/  
3742 0402  
3743 0317  
3744 4003  
3745 1001  
3746 1607  
3747 0504  
3750 4001 TRAN5E, TEXT /DBCI CHANGED AC/  
3751 0300  
3752 0402  
3753 0311  
3754 4003  
3755 1001  
3756 1607  
3757 0504  
3760 4001  
3761 0300 OUT1E, TEXT /DBSO ERROR/  
3762 0402  
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 0424  
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  
4054 1617 INT2E, TEXT /NO INTERRUPT, SKIP/  
4055 0011  
4056 1624  
4057 0522  
4060 2225  
4061 2024  
4062 0440  
4063 2313  
4064 1120  
4065 0000  
4066 1116 INT3E, TEXT /INTERRUPT, NO SKIP/  
4067 2405  
4070 2222  
4071 2520  
4072 2454



```

4073 4016
4074 1740
4075 2313
4076 1120
4077 0000
4100 1617 INT4E, TEXT /NO INTERRUPT, NO SKIP/
4101 4011
4102 1624
4103 0522
4104 2225
4105 2024
4106 5440
4107 1617
4110 4023
4111 1311
4112 2000
4113 1617 INT5E, TEXT /NO SKIP/
4114 4023
4115 1311
4116 2000
4117 2313 INT6E, TEXT /SKIP/
4120 1120
4121 0000
    
```

```

4200 PAGE /APT/
/APT/ ROUTINE TO HANDLE ERRORS UNDER APT CONTROL.
4200 0000 APTER, 0 /APT/
4201 7200 CLA /APT/
4202 6224 RIF /APT/CREATE CDF INST TO CURRENT FIELD.
4203 1377 TAD (6201 /APT/
4204 3207 DCA /APT/MODIFY NEXT CDF INST.
4205 7040 CMA /APT/
4206 1200 TAD APTER /APT/AC=ERROR PC.
4207 6201 CDF /APT/(MODIFIED CDF) DF=CURRENT FIELD.
4210 6272 CIF 70 /APT/IF=FIELD 7.
4211 5776 JMP I (6200 /APT/CALL APT = 'ERROR'.

/APT/ ROUTINE TO GET SWITCHES INTO THE AG.
4212 0000 APTSR, 0 /APT/
4213 2262 ISE APTOKC /APT/BUMP OK CALL COUNT.
4214 7410 SKP /APT/
4215 4226 JMS APTOK /APT/GO CALL APT = 'PROG OK'.
4216 7200 CLA /APT/
4217 1021 TAD HCH1 /APT/USE PSR?
4220 7710 SPA CLA /APT/SKP IF YES.
4221 5224 JMP ,+3 /APT/
4222 1020 TAD PSR /APT/AC=PSR.
4223 7410 SKP /APT/
4224 7604 7604 /APT/(LAS) AC=HARD SR.
4225 5612 JMP I APTSR /APT/RTN TO CALL+1.

/APT/ ROUTINE TO 'NOTIFY' APT THAT THE PROGRAM IS RUNNING OK.
4226 0000 APTOK, 0 /APT/
4227 7200 CLA /APT/
4230 1022 TAD HCH2 /APT/UNDER APT CONTROL?
4231 7700 SHA CLA /APT/SKP IF YES.
4232 5626 JMP I APTOK /APT/RTN TO CALL+1.
4233 1261 TAD APTOKK /APT/INIT OK COUNTER.
4234 3262 DCA APTOKC /APT/
4235 6002 IOF /APT/
4236 1295 TAD APTIMX /APT/DELAY FOR 100MS.
4237 3297 DCA APTCTX /APT/
4240 1296 TAD APTIMY /APT/
4241 3260 DCA APTCTY /APT/
4242 2260 ISE APTCTY /APT/
4243 5242 JMP ,=1 /APT/
4244 2257 ISE APTCTX /APT/
4245 5240 JMP ,=5 /APT/
4246 6224 RIF /APT/CREATE A CDF INST TO CURRENT FIELD.
4247 1377 TAD (6201 /APT/
4250 3231 DCA ,+3 /APT/MODIFY NEXT CDF INST.
4251 6201 CDF /APT/(MODIFIED CDF) DF=CURRENT FIELD.
4252 6272 CIF 70 /APT/IF=FIELD 7.
4253 4775 JMS I (6000 /APT/CALL APT = 'PROG OK'.
    
```

```

4254 5626      JMP I  APTOK      /APT/RTN FROM APT - RTN TO CALL+1.

4255 7771      APTIMX, -7      /APT/
4256 0000      APTINY, 0       /APT/
4257 0000      APTCTX, 0       /APT/
4260 0000      APTCTY, 0       /APT/
4261 0000      APTOKK, 0      /APT/OK COUNT CONSTANT,
4262 0000      APTOKC, 0      /APT/OK COUNTER,

4375 6500
4376 6520
4377 6201
4400 4400      PAGE          /APT/
    
```

/APT/ ROUTINE TO INITIALIZE FOR RUNNING UNDER APT CONTROL.

```

4400 7200      APTI2, CLA      /APT/
4401 1022      TAD      HCW2      /APT/RUN UNDER APT CONTROL?
4402 7700      SMA CLA      /APT/SKP IF YES,
4403 5777'     JMP      START1     /APT/
4404 1020      TAD      PSR      /APT/FORCE BITS 0,1,4,6 TO '1'.
4405 2376      AND      (1037     /APT/
4406 1375      TAD      (6240     /APT/
4407 3020      DCA      PSR      /APT/
4410 1020      TAD      PSR      /APT/EFFECTIVELY START AT LOC 201?
4411 0374      AND      (20      /APT/SR?,
4412 7650      SNA CLA      /APT/SKP IF YES,
4413 5221      JMP      APTI21     /APT/NO, EFFECTIVELY START AT LOC 200,
4414 7461      CHA      APTI21     /APT/FORCE IJUMPER (LOC 34) TO: 7777,
4415 3034      DCA      IJUMPER     /APT/
4416 7040      CHA      FJUMPER     /APT/FORCE FJUMPER (LOC 35) TO: 7777,
4417 3035      DCA      FJUMPER     /APT/
4420 5265      JMP      APTI20     /APT/
4421 6224      APTI21, RIF      /APT/CREATE A CDF TO CURRENT FIELD,
4422 1373      TAD      (6201     /APT/
4423 3226      DCA      ,*3      /APT/MODIFY THE 2ND CDF FOLLOWING,
4424 6271      CDF      70      /APT/DF=FIELD 7,
4425 1772'     TAD      6730     /APT/RETRIEVE HARDWARE CONFIG WD 3,
4426 6201      CDF      /APT/(MODIFIED CDF) DF=CURRENT FIELD,
4427 3023      DCA      HCW3      /APT/SAVE HCW3,
4430 0007      CAF      /APT/CLEAR ALL FLAGS,
4431 1020      TAD      PSR      /APT/GET & MASK DEVICE CODE,
4432 0371      AND      (7      /APT/
4433 7106      CLL RTL      /APT/POSITION BITS,
4434 7004      RAL      /APT/
4435 1370      TAD      (6500     /APT/GENERATE BASIC IOT,
4436 3767'     DCA      IOTS      /APT/SAVE BASIC IOT,
4437 1021      TAD      HCW1      /APT/GET & SAVE INTERRUPT JUMPERS,
4440 0366      AND      (37      /APT/1ST 3 BITS,
4441 7002      BSW      /APT/POSITION THEM,
4442 7104      CLL RAL      /APT/
4443 3034      DCA      IJUMPER     /APT/TEMP STORE,
4444 1022      TAD      HCW2      /APT/
    
```

```

4445 0365      AND      (177      /APT/NEXT 7 BITS,
4446 1034      TAD      IJUMPER     /APT/GET ALL 12 BITS TOGETHER,
4447 3034      DCA      IJUMPER     /APT/
4450 1023      TAD      HCW3      /APT/GET & SAVE FLIPFLOP JUMPERS,
4451 3035      DCA      FJUMPER     /APT/
4452 1364      TAD      (-10      /APT/8 IOTS WILL BE SET UP,
4453 3026      DCA      CNTR1     /APT/
4454 1367      TAD      (TIOT-1   /APT/
4455 3010      DCA      10      /APT/
4456 1410      APTI22, TAD I    10      /APT/
4457 3316      DCA      ART000     /APT/
4460 1767'     TAD      IOTS      /APT/
4461 3716      DCA I  APT000     /APT/
4462 2767'     ISZ      IOTS      /APT/
4463 2026      ISZ      CNTR1     /APT/
4464 5256      JMP      APTI22     /APT/
4465 1363      APTI20, TAD      (IAPTER /APT/CHANGE SOME LOGS TO: JMS I IAPTER,
4466 0365      AND      (177      /APT/
4467 1362      TAD      (4400     /APT/
4470 3761'     DCA      APTE00     /APT/
4471 1761'     TAD      APTE00     /APT/
4472 3760'     DCA      APTE01     /APT/
4473 1761'     TAD      APTE00     /APT/
4474 3757'     DCA      APTE02     /APT/
4475 1761'     TAD      APTE00     /APT/
4476 3756'     DCA      APTE03     /APT/
4477 1761'     TAD      APTE00     /APT/
4500 3755'     DCA      APTE04     /APT/
4501 1761'     TAD      APTE00     /APT/
4502 3754'     DCA      APTE05     /APT/
4503 1761'     TAD      APTE00     /APT/
4504 3753'     DCA      APTE06     /APT/
4505 1352      TAD      (APTER     /APT/CHANGE THE POINTER FOR ERROR HANDLING,
4506 3092      DCA      XERROR     /APT/
4507 1351      TAD      (APTPAS    /APT/CHANGE A LOC TO: JMP APTPAS,
4510 0365      AND      (177      /APT/
4511 1350      TAD      (5200     /APT/
4512 3747'     DCA      EPASS      /APT/
4513 7300      CLA CLL      /APT/
4514 3036      DCA      TYPFLG     /APT/CLEAR ERROR FLAG, CLEAR TYPE FLAG,
4515 5746'     JMP      INIT1      /APT/

4516 0000      APT000, 0      /APT/
    
```

SSS

```

4546 0400
4547 3257
4550 5200
4551 3300
4552 4200
4553 3327
4554 3322
    
```



```

4000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
4100 11111111 11111111 11000000 00000000 00000000 00000000 00000000 00000000
4200 11111111 11111111 11111111 11111111 11111111 11111111 11100000 00000000
4300 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000111
4400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
4500 11111111 11111110 00000000 00000000 00000001 11111111 11111111 11111111
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
    
```

```

APT000 4516 DBS0X 0317 INIT2E 3662 MTP 3241
APTCTY 4257 DH0 3515 INIT3 0435 OCTASC 0000
APTCTY 4260 DH1 3516 INIT3E 3675 OUT1 1000
APTE00 0264 DH2 3526 INOU1 2200 OUT1E 3762
APTE01 0271 DH3 3535 INOU1A 2205 OUT2 1033
APTE02 0303 DH4 3541 INOU2 2230 OUT3 1054
APTE03 0310 DH5 3576 INOU2A 2235 OUT3A 1060
APTE04 0315 DH6 3622 INOU3 2260 OUT4 1105
APTE05 0322 DHDER 2656 INOU3A 2265 OUT4A 1111
APTE06 0327 DIOT 0290 INOU4 2325 OUT4E 3770
APTER 4200 EHALT 2654 INOU4A 2331 OUT5 1200
APTIMX 4255 EPAS1 3272 INT1 2400 OUT5A 1205
APTIMY 4256 EPASS 3297 INT1A 2404 OUT6 1250
APTIZ 4400 ERADR 3025 INT1AE 2431 OUT6A 1254
APTIZ0 4465 ERRAD 2666 INT1ABE 2446 OUT7 1262
APTIZ1 4421 ERROR 2600 INT1ACE 2453 OUT7A 1266
APTIZ2 4456 FJUMPE 0035 INT1AD 2441 OUT7E 3776
APTOK 4226 HCA1 0021 INT1AE 4043 OUT8 1315
APTOK0 4262 HCA2 0022 INT1OK 2437 OUT8A 1322
APTOKK 4261 HCA3 0023 INT2E 4054 PTR1 0253
APTPAS 3300 HLT 7402 INT3 2460 POINT1 0010
APTSR 4212 IAPTER 0024 INT3A 2464 PRINT 3200
BIT01 3035 IAPTSR 0025 INT3C 2505 PSR 0020
BITOUT 3031 IJUMPE 0034 INT3E 4060 SPLIT 3012
BITS 2646 IN1 1400 INT4E 4100 SR00 2733
BSW 7002 IN10 2063 INT5E 4143 SR01 2734
CAF 6007 IN10A 2070 INT6E 4137 SR02 2735
CNTR1 0026 IN1A 1403 IOTS 0247 SR03 2736
CRLF 3513 IN1B 1420 K0077 3245 SR04 2737
DATA1 0027 IN2 1434 K212 3246 SR05 2740
DATA2 0030 IN2E 4004 K215 3245 SR06 2741
DATA3 0031 IN3 1462 K240 3250 SR0 0003
DATA4 0032 IN3A 1466 K336 3247 START1 0202
DATA5 0033 IN3E 4012 LAS 4425 START2 0244
DATAP 2665 IN4E 4027 LOOP1 2667 T1OT 0250
DATCNT 2664 IN5 1600 LOOP2 2712 TRAN1 0400
DATOUT 2641 IN5A 1604 LP1EXA 2677 TRAN1E 3704
DBCI 4443 IN6 1633 LP1EXT 2705 TRAN2 0477
DBCI1 0300 IN6 1637 LP1EXX 2711 TRAN2E 3717
DBCO 4445 IN6A 1644 LP2EXT 2726 TRAN3 0516
DBCOX 0312 IN6G 1672 LPCNT 0037 TRAN3E 3732
DBD1 4440 IN7 1676 LSTDGT 3027 TRAN4 0536
DBDIX 0261 IN7A 1702 M1 3400 TRAN4E 3742
DBE1 4441 IN7G 1731 M2 3421 TRAN5 0556
DBEIX 0266 IN8 2000 M2A 3445 TRAN5E 3752
DBR1 4444 IN8A 2005 M3 3476 TRAN6 0600
DBRIX 0305 IN9 2030 M40 3249 TRAN7 0616
DBRO 4447 IN9A 2034 MEP 3275 TRAN8 0634
DBROX 0324 IN9E 4035 MSG 2634 TYPE 3251
DBSK 4442 INIT1 0400 MGA 7501 TYPFLG 0036
DBSKX 0273 INIT1E 3646 MGL 7421 TYPSET 3215
DBSO 4446 INIT2 0416 HSTDGT 3026 XDBC1 0043
    
```

XDBC0 0045  
XDBD1 0040  
XDBE1 0041  
XDBR1 0044  
XDBR0 0047  
XDBSK 0042  
XDBSO 0046  
XERROR 0052  
XLOOP1 0053  
XLOOP2 0054  
XPRINT 0050  
XTYPE 0051

ERRORS DETECTED: 0

LINKS GENERATED: 40

RUN-TIME: 11 SECONDS

3K CORE USED

