

IDENTIFICATION

PRODUCT CODE: MAINDEC-12-DIAC-D
PRODUCT NAME: EXTENDED MEMORY CONTROL
(EXTMC12)
DATE CREATED: JUNE 19, 1970
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: HAROLD LONG

RSW = 7001 in BK

8 MODE
START 20

RSW 5-1 inhibit bell
RSW 6-1

1. ABSTRACT

PDP-12 Extended Memory Control Test (Version 2) is designed to exercise all functions of memory control available to a PDP-12 with at least 4K of additional memory. This includes data field control, data handling, interrupts, data field-instruction field control during an interrupt, auto indexing in extended memory, subroutine handling (both with and without interrupts), and non-existent memory detect handling. All these tests are performed both in P mode and L mode whenever possible.

Program Control is handled by a monitor resident in bank 0. Several options are available to the user for control of error handling.

2. REQUIREMENTS

2.1 EQUIPMENT

- a) Any PDP-12 with at least 4K of extended memory.
- b) An ASR-33 teletype or equivalent

2.2 PRELIMINARY PROGRAMS

- a) All basic processor - memory tests should have been run successfully.

3. LOADING PROCEDURES

3.1 METHOD

This program must be loaded with the binary loader. If you are unfamiliar with the proper binary loading procedures refer to "Appendix A" of this program, otherwise proceed with the following:

- a) Set the teletype reader switch to FREE.
- b) Open the teletype reader and insert the program tape so that the arrows on the tape are visible to and pointing toward the operator.
- c) Close the reader and set the reader switch on START.
- d) Set the teletype front panel switch on ON LINE.
- e) Set the LEFT switches to 7777.
- f) Set the RIGHT switches to 4000.
- g) Set the MODE switch to 8 mode.
- h) Depress I/O preset.

- i) Depress START LS.
- j) When the program tape has been read the ACCUMULATOR must be 0000 if it is not, a read-in error has occurred and one might try reloading the binary loader.
- k) Remove the program tape from the reader.

4. STARTING PROCEDURES

- a) Set the right switches as outlined in section 5.1, switch settings.
- b) Set the mode switch to 8 mode.
- c) Depress I/O preset.
- d) Depress start 20.
- e) The program, when properly running, will type the contents of the pass counter at the completion of each pass.
- f) Attempting to test non-existent memory may result in false error printout or program destruction.

5. ERROR ROUTINE

5.1 SWITCH SETTINGS

- a) In general, RSW0-6 allow selection of the error mode. With all switches equal to zero, the sequence would be: (error typeout and halt) - operator selects any additional error modes and depresses continue; machine will respond as directed by right switches.

RSW 00 = 1, SUPPRESS ERROR HALT
RSW 01 = 1, SUPPRESS ERROR PRINTOUT
RSW 02 = 1, SCOPE LOOP ON FAILING ROUTINE
RSW 03 = 1, SCOPE LOOP ON NON-FAILING ROUTINE
RSW 05 = 1, INHIBIT BELL
RSW 06 = 1, INHIBIT PASS COUNTER

- b) RSW 08-11 must contain the amount of memory available, within the range of 8 to 32K.

9K: 001
12K: 010
16K: 011
20K: 100
24K: 101
28K: 110
32K: 111

5.2 ERROR PRINTOUT

- a) The error printout has the following general form:

```
TESTNAME TEST MESSAGE FAILED  
REGISTER REGISTER REGISTER...  
(CONTENTS) (CONTENTS) (CONTENTS)...
```

The message is interpreted as follows:

TESTNAME - The mnemonic code used to identify each test in the listing.

TEST MESSAGE FAILED - What the test is attempting to check, along with the identifier "failed".

REGISTERS - The registers associated with this test; this may be the L mode data field register, the L mode save field register, etc.

(CONTENTS) - The contents of each register identified above.

Consult the listing for further explanation of any error condition encountered.

- b) Following is a list of all possible error printouts:

```
TST01  
CDF OR RDF FAILED (PMODE)  
SENT RCVD
```

```
TST02  
CDF OR RDF FAILED (PMODE)  
SENT RCVD
```

```
TST03  
LDF OR RDF FAILED (LMODE)  
SENT RCVD
```

```
TST04  
LDF OR RDF FAILED (LMODE)  
SENT RCVD
```

```
TST05  
CDF OR RDF FAILED (PMODE)  
SENT RCVD
```

```
TST06  
LDF OR RDF FAILED (LMODE)  
SENT RCVD
```

TST07
PMode INTERRUPT FAILED

TST08
PMode LOAD SF OR RIB FAILED
DF SF

TST9A
LMode INTERRUPT FAILED

TST09
LMode LOAD SF OR RIB FAILED
DF SF

TST 10
PMode DF FAILED TO ZERO ON AN INTERRUPT
SENT SF RCVD

TST11
LMode DF FAILED TO ZERO ON AN INTERRUPT
SENT SF RCVD

TST12
DCA I - TAD I FAILED
BANK LOCN SENT RCVD

TST13
STA - LDA FAILED
BANK LOCN SENT RCVD

TST14
LMode JUMP SAVE RETURN FAILED FOR NORMAL JUMP

TST15
DJR FAILED TO INHIBIT JUMP SAVE

TST16
LMode JMP FAILED TO CLEAR DJR

TST17
PMode JUMP ALTERED CELL ~~0000~~

TST18
PMode TOF ALTERED CELL ~~0000~~

TST19
LMode TOF ALTERED CELL ~~0000~~

TST20
PMode JUMP CLEARED DJR

TST21
DJR INHIBITED PMode INTERRUPT SAVE

TST22
NON EXISTANT MEMORY READ-BACK FAILED
BANK DATA

TST24
CIF FAILED TO LOAD PROPER IF
SENT TCVD

TST25
LIF FAILED TO LOAD PROPER IF
SENT TCVD

TST26
CIF FAILED TO FIND PROPER MEMORY
SENT RCVD

TST27
PMODE INTERRUPTS NOT INHIBITED BY CIF
BANK

TST28
LMODE LIF FAILED TO INHIBIT INTERRUPTS
BANK

TST29
IMODE JMP Ø FAILED TO CLEAR INTERRUPT INHIBIT
BANK

TST30
LMODE DJR-JMP Ø FAILED TO LOAD IF
BANK

TST 32
LMODE ION-LIF FAILED TO INHIBIT INTERRUPTS

TST32
LMODE LIF-JMP N FAILED TO LOAD SF
IF DF SF

TST34
LMODE RMF IN EXTENDED BANK FAILED
BANK SF

TST35
PMODE AUTO-INDEX FAILED
BANK CELL ADDR

TST36 LMODE AUTO-INDEX FAILED
FIELD LOCN

EXT MEM TST PASS --- ØØØØ

SPURIOUS INTERRUPT
(CHECK IOC I/O PRESET)


```
/PDP-12 EXTENDED MEMORY TEST, VERSION 2, MAINDEC 12-D1AC-L
/COPYRIGHT, 1970, DIGITAL EQUIPMENT CORP., MAYNARD, MASS,
/
/AUTHOR: HAROLD LONG
/
/THIS TEST IS DESIGNED TO EXERCISE ALL MEMORY
/REFERENCE INSTRUCTIONS AVAILABLE ON A PDP-12
/COMPUTER WITH EXTENDED MEMORY, IT OPERATES
/IN BOTH P MODE AND L MODE, IN ALL AVAILABLE
/MEMORY; A MINIMUM OF 8K OF CORE IS REQUIRED.
/
/
/RIGHT SWITCH REGISTER OPTIONS:
/SR00=1, INHIBIT ERROR HALT
/SR01=1, INHIBIT ERROR PRINTOUT
/SR02=1, SCOPE LOOP ON FAILING ROUTINE
/SR03=1, SCOPE LOOP ON NON-FAILING ROUTINE
/SR05=1, INHIBIT BELL
/SR06=1, INHIBIT PASS COUNTER PRINTOUT
/SR09,10,11 -- EXTENDED BANKS
/
/
/NORMAL SWITCH SETTING IS RSW=000N, WHERE
/(N)=AMOUNT OF EXTENDED MEMORY AVAILABLE AND
/IS WITHIN THE RANGE OF 0<N<10 OCTAL
/
/
/PROGRAM CONTROL IS HANDLED BY A MONITOR RESIDENT IN BANK 0,
/LOCATIONS 5000 TO 5177, ALL ROUTINES VISIT THE MONITOR 4096 TIMES
/AT THE COMPLETION OF A TEST, AN ERROR WILL CAUSE THE
/PROGRAM TO TYPE OUT THE ERROR MESSAGE AND HALT, THE
/HALT IS AT LOCATION 5033, THE HALTS IN THE PROGRAM
/BLOCKS ARE NOT, REPEAT NOT, EXECUTED, THEY ARE
/THERE FOR MANUAL PROGRAM CONTROL ONLY,
/
/TO REDEFINE AMOUNT OF MEMORY AVAILABLE, THE
/PROGRAM MUST BE RESTARTED,
/
/I/O PRESET TO PMODE, START 20
/
```

```

/PDP-12 INSTRUCTION DEFINITIONS
/L MODE MEMORY REFERENCE
064 LDF=0640 /LOAD DATA FIELD 0-37
090 LIF=0600 /LOAD INSTRUCTION FIELD 0-37
0006 DJR=0006 /DISABLE JUMP RETURN
/MODE CHANGE
0002 POP=0002 /SWITCH TO P MODE
6141 LINC=6141 /SWITCH TO L MODE
/L MODE PROGRAMMING INSTRUCTIONS
6000 LJMP=6000 /JMP
0011 CLR=0011
0450 AZE=0450
2000 ADD=2000
0500 IOB=0500
0016 LNOP=0016 /NOP
0300 ROR=0300
0456 LSKP=0456
0240 ROL=0240
1600 BSE=1600
1540 BCL=1540
0060 SET=0060 / (REALLY SET I)
4000 STC=4000
1500 SRO=1500 /USED AS A SWITCH CHECK
1000 LDA=1000
1040 STA=1040
0220 XSK=0220 / (REALLY XSK I)
/DATA MATRIX SWITCHES
7777 EXITA=7777
4444 EXITB=4444 /SPECIAL RESTART SWITCH
0000 EXIT=0000

```

```

/P MODE INTERRUPT HANDLER
* 020
0000 0000 PINTR, 0020 /INTERRUPT RETURN STORAGE (ALSO LINK JUMP SAVE)
0001 732 CLA CLL CML /SET LINK, CLEAR AC
0002 6234 RIB /READ SF
0003 3143 DCA PREG /SAVE IT
0004 1142 TAD PPOINT /GET SWITCH
0005 7657 SNA CLA /SET?
0006 5555 JMP I RETURN /NO, RETURN THROUGH PRESET LINKUP
0007 3142 DCA PPOINT /CLEAR SWITCH
0010 6244 RMF /RESTORE MEMORY
0011 2000 ISZ 0 /ENABLE RETURN
0012 5400 JMP I PINTR /BACK TO MAINLINE VIA INTERRUPT RETURN LINKUP

/AUTO-INDEX REGISTERS
0013 0000 LREG1, 0000 /DATA POINTER
0014 0000 PINT, 2000 /MESSAGE POINTER
0015 0000 AUTO11, 0000
0016 0000 AUTO12, 0000
0017 0000 COUNT, 0000

/CROSS-PAGE REFERENCE TAGS AND CONSTANTS
/
* 020
0020 5176 JMP 176 /MINOR START
0021 003 K 003, 0003
0022 0007 K 007, 0007
0023 0010 K 010, 0010
0024 0017 K 017, 0017
0025 0020 K 020, 0020
0026 0040 K 040, 0040
0027 0070 K 070, 0070
0030 0077 K 077, 0077
0031 0100 K 100, 0100
0032 0177 K 177, 0177
0033 0207 K 207, 0207
0034 0400 K 400, 0400
0035 1026 K1026, 1026
0036 1777 K1777, 1777
0037 2727 K2000, 2727

```

```

/LMODE INTERRUPT HANDLER
*040
0240 000 LINTR, 0000 /INTERRUPT RETURN STORAGE
0241 011 CLR /CLEAR LINK, CLEAR AC
0242 150 LHAN, IOB /
0243 6234 RIR /READ SAVE FIELD REG
0244 4064 STC LREG /SAVE IT
0245 1500 SRO /SWITCH SET?
0246 0065 LPOINT /
0247 0456 LSKP /TO HERE IF BIT 0=1
0250 605 LSET, LJMP , /NO, RETURN TO BANK 0 THROUGH PRESET LINKUP
0251 150 IOB /
0252 6244 RMF /YES, RESTORE MEMORY FIELDS
0253 0220 XSK 0 /INCREMENT
0254 234 ADD LINTR /GET RETURN
0255 1620 BSE 20 /MAKE IT A LINK JUMP (BSE 1)
0256 6000 /
0257 4063 STC ,+4 /STORE FOR EXECUTION
0260 4065 STC LPOINT /CLEAR SWITCH
0261 150 IOB /
0262 6001 ION /ENABLE INTERRUPTS
0263 6263 LJMP , /BACK TO BANK 0 VIA INTERRUPT RETURN LINKUP
0264 0000 LREG, 0000
0265 1000 LPOINT, 0000

```

```

/
/MORE TAGS AND CONSTANTS
/
0066 1000 BANK, 0000 /AMOUNT OF EXTENDED MEMORY
0067 5464 BELL, BELLS /CROSS PAGE TO BELL RINGER
0070 5326 BNKSET, LOCSET /BANK SET
0071 5020 ERROR, ERRORS /CROSS PAGE TO ERROR MONITOR
0072 2120 EXDF33, XDF33
0073 2121 EXIF33, XIF33
0074 5431 GETBNK, GETNXT /CROSS PAGE TO PMODE FIND BANK
0075 5445 GETBNL, GETNXL /CROSS PAGE TO LMODE FIND BANK
0076 5252 K5252, 5252
0077 6020 K6020, 6020
0100 7774 K7774, 7774
0101 6201 KCDF, CDF /PMODE CDF
0102 6202 KCIF, CIF /PMODE CIF
0103 7402 KHLT, HLT /PMODE HLT
0104 1647 KLDF, LDF /LMODE LDF
0105 0500 KIOB, IOB
0106 6002 KIOF, IOF
0107 1016 KLNOP, LNOP
0110 6224 KRIF, RIF /LMODE NOP
0111 1632 KLIF, LIF /LMODE LIF
0112 6007 KLJMP, LJMP /LMODE JMP
0113 0003 LBANK, 0003 /LINC FIELD IN USE (>4K)
0114 1037 LMASK, 0037 /LIF/LDF MASK
0115 1000 LSTERR, 0000 /LAST ERROR POINTER
0116 5000 NERROR, NERROS /CROSS PAGE TO NON-ERROR MONITOR
0117 4206 PASSN, PASS
0120 0000 FBANK, 0000 /PMODE BANK IN USE (<32K)
0121 0070 PMASK, 0070 /CIF/CDF MASK
0122 1502 PNTA, LOCA /INTERRUPT RETURN TEST 07
0123 0527 PNTB, LOCB /INTERRUPT RETURN TEST 08
0124 1627 PNTC, LOCC /INTERRUPT RETURN TEST 09
0125 1564 PNTCA, LOCCA /INTERRUPT RETURN
0126 1667 PNTD, LOCD /INTERRUPT RETURN TEST 10
0127 1722 PNTE, LOCE /INTERRUPT RETURN TEST 11
0130 1246 PNTE, LOCF /INTERRUPT RETURN TEST 21
0131 1437 PNTG, LOCG
0132 1473 PNTH, LOCH
0133 1675 PNTI, LOCI
0134 1651 PNTJ, LOCJ
0135 1734 PNTK, LOCK
0136 2012 PNTL, LOCL
0137 4222 PNTD, LOCD
0140 5423 PNTP, LOCP
0141 2007 PNT3IN, PNT3M
0142 0000 PPOINT, 0000

```

0143	0000	PREG,	0000	/HOLDS SF
0144	5200	RANDOM,	RANDY	/CROSS PAGE TO RANDOM GENERATOR
0145	0000	REGA,	0000	/DATA
0146	0000	REGB,	0000	/DATA
0147	0000	REGC,	0000	/DATA
0150	0147	REGCN,	REGC	
0151	0000	REGD,	0000	/DATA
0152	0000	REGE,	0000	/DATA
0153	5261	RELOC,	RELOC	/CROSS PAGE TO RELOCATOR SUBR
0154	0000	RELPT,	PINTR	/CROSS BANK TO INTERRUPT RETURN STORAGE
0155	0000	RETURN,	0000	/PMODE INTERRUPT RETURN IF SWITCH=0
0156	5253	SETFLG,	FLAG	/CROSS PAGE TO FLAG SET ROUTINE
0157	0000	SPACE,	0000	/DATA I/O BUFFER
0160	5400	TSTINT,	INTTST	
0161	0746	TST12N,	TST12	/CROSS PAGE TO TEST 12
0162	1004	TST13N,	TST13	/CROSS PAGE TO TEST 13
0163	1370	TST23N,	TST23	
0164	1403	TST24N,	TST24	
0165	1556	TST27N,	TST27	/CROSS PAGE TO TEST 27
0166	1613	TST28N,	TST28	
0167	1747	TST30N,	TST30	
0170	2033	TST32N,	TST32	
0171	2072	TST33N,	TST33	
0172	2152	TST34N,	TST34	
0173	2400	TST35N,	TST35	
0174	5244	TYPE,	TYPOUT	/CROSS PAGE TO TYPEOUT SUBR

```

/
/TO HERE FROM MINOR START
/
*176
P176 7417 SKP /DON'T RING ON STARTUP, INITIALIZE TEST
P177 4467 *START, JMS I BELL /GO RING BELL, RETURN TO TST01
/
/MAJOR START P MODE; INITIALIZATION ROUTINE
/
* 200
P200 7604 START, LAS /READ SWITCHES
P201 1022 AND K0007 /SAVE BANK BITS
P202 3066 DCA BANK /AMOUNT OF EXTENDED MEMORY
P203 3145 DCA REGA /CLEAR LOOP COUNTER
P204 3115 DCA LSTERR /CLEAR OLD ERROR
P205 3127 DCA PBANK
P206 3017 DCA COUNT /CLEAR PASS COUNTER
P207 1021 TAD K0003 /SET LBANK TO UPPER MEMORY
P210 3113 DCA LBANK
P211 6201 CDF 00 /RESTORE DATA FIELD
P212 4560 JMS I TSTINT /TEST FOR NO INTERRUPT
/PMODE
/CAN THE DATA FIELD REGISTER BE LOADED WITH ALL NUMBERS (BINARY COUNT)
/
TST01, TAD REGA /FETCH TEST NUMBER
P214 1021 AND PMASK /SAVE BITS 06-08
P215 3146 DCA REGB /SAVE FOR OBSERVATION
P216 1146 TAD REGB /FETCH IT
P217 1101 TAD KODF /ADD CDF
P220 3221 DCA ,+1 /PLACE IT IN ROUTINE
P221 1000 0002 /EXECUTE CDF N
P222 6214 RDF /GET DATA FIELD
P223 121 AND PMASK /SAVE BITS 06-08
P224 3147 DCA REGC /SAVE FOR TYPING
P225 1147 TAD REGC /FETCH IT
P226 7041 CIA /2'S COMPLEMENT
P227 1146 TAD REGB /COMPARE WITH DATA SENT
P230 6201 CDF 00 /RESTORE DATA FIELD
P231 765 SMA CLA /INCORRECT IF NOT ZERO
P232 4516 JMS I ERROR /CHECK WITH MONITOR
P233 4471 JMS I ERROR /CDF OR RDF FAILED
P234 5475 TST01M /MESSAGE POINTER
P235 7402 HLT /ERROR HALT
P236 761 SKP CLA /GO TO NEXT TEST
P237 213 TST01 /SCOPE LOOP, JSR LOOP

```

/PMODE

/CAN THE DATA FIELD BE LOADED WITH RANDOM NUMBERS

```

/
TST02,  JMS I  RANDOM      /GET A RANDOM NUMBER
        AND    PMASK      /SAVE BITS 06-08
        DCA    REGB       /SAVE FOR OBSERVATION
        TAD    REGR       /FETCH IT
        TAD    KCDF       /ADD CDF
        DCA    ,+1        /PLACE IT IN ROUTINE
        0000    0000      /EXECUTE CDF R
        RDF                    /GET DATA FIELD
        AND    PMASK      /SAVE BITS 06-08
        DCA    REGC       /SAVE FOR TYPING
        TAD    REGC       /FETCH IT
        CIA                    /2'S COMPLEMENT
        TAD    REGB       /COMPARE
        CDF    00         /RESTORE DATA FIELD
        SNA CLA            /INCORRECT IF NOT ZERO
        JMS I  NERROR     /CHECK WITH MONITOR
        JMS I  ERROR      /CDF FAILED
        TST02M            /MESSAGE POINTER
        HLT                    /ERROR HLT
        SKP CLA            /NEXT TEST
        TST02            /SCOPE LOOP; ISZ LOOP
    
```

/LMODE

/CAN THE DATA FIELD REGISTER BE LOADED WITH BINARY COUNT

```

/
TST03,  TAD    REGA      /FETCH TEST NUMBER
        AND    LMASK     /SAVE BITS 07-11
        DCA    REGB     /SAVE FOR OBSERVATION
        TAD    REGR     /FETCH IT
        TAD    KLDF     /ADD LDF
        DCA    ,+2      /PLACE IN ROUTINE
        LINC                    /GO TO LINC MODE
        0000    0000    /EXECUTE LDF
        JCB                    /PREPARE TO GET DATA FIELD
        RDF                    /GET DATA FIELD
        PDP                    /BACK TO PMODE
        RAR CLL            /JUSTIFY RIGHT TO AGREE WITH REGR
        DCA    REGC     /SAVE FOR TYPING
        TAD    REGC     /FETCH IT
        CIA                    /2'S COMPLEMENT
        TAD    REGR     /COMPARE
        CDF    00         /RESTORE DATA FIELD
        SNA CLA            /INCORRECT IF NOT ZERO
        JMS I  NERROR     /CHECK WITH MONITOR
        JMS I  ERROR      /LDF FAILED
        TST03M            /MESSAGE POINTER
        HLT                    /ERROR HLT
        SKP CLA            /GO TO NEXT TEST
        TST03            /SCOPE LOOP; ISZ LOOP
    
```



```

/LMODE
/CAN THE DATA FIELD REGISTER BE LOADED WITH RANDOM NUMBERS
/
TST04:  JMS I  RANDOM      /GET RANDOM NUMBER
        AND    LMASK      /SAVE BITS 07-11
        DCA    REGR      /SAVE FOR OBSERVATION
        TAD    REGB      /FETCH IT
        TAD    KLDF      /ADD LF
        DCA    ,+2       /PLACE IN ROUTINE
        LINC   LINC      /GO TO LINC MODE
        0000   0000      /EXECUTE LDF
        IOR    IOR       /PREPARE TO GET DATA FIELD
        RDF    RDF       /GET DATA FIELD
        PDP    PDP       /BACK TO PMODE
        RAR    CLL      /JUSTIFY RIGHT TO AGREE WITH REGR
        DCA    REGC      /SAVE FOR TYPING
        TAD    REGC      /FETCH IT
        CIA    CIA       /2'S COMPLEMENT
        TAD    REGB      /COMPARE
        CDF    0?       /RESTORE DATA FIELD
        SNA    CLA      /INCORRECT IF NOT ZERO
        JMS I  NERROR    /CHECK WITH MONITOR
        JMS I  ERROR     /LDF FAILED
        TST04M          /MESSAGE POINTER
        HLT           /ERROR HALT
        SKP    CLA      /GO TO NEXT TEST
        TST04          /SCOPE LOOP; ISZ LOOP

```

```

/PMODE
/GATE SHAKER TEST
/
0345 4544 TST05, JMS I RANDOM /GET A RANDOM NUMBER
0346 0121 AND PMASK /SAVE BITS 06-08
0347 3146 DCA REG8 /SAVE FOR OBSERVATION
0350 1146 TAD REG8 /FETCH IT
0351 1101 TAD KCDF /ADD CDF
0352 3374 DCA NOW1 /STORE FOR EXECUTION
0353 6201 CDF 00 /FOLLOWING IS A SERIES OF CDF
0354 6241 CDF 40 /NOISE MAKERS,
0355 6221 CDF 20
0356 6211 CDF 10
0357 6271 CDF 70
0360 6261 CDF 60
0361 6251 CDF 50
0362 6241 CDF 40
0363 6231 CDF 30
0364 6221 CDF 20
0365 6211 CDF 10
0366 6221 CDF 20
0367 6231 CDF 30
0370 6241 CDF 40
0371 6251 CDF 50
0372 6261 CDF 60
0373 6271 CDF 70
0374 0000 NOW1, 0000 /EXECUTE ACTUAL CDF
0375 6214 RDF /GET DATA FIELD
0376 0121 AND PMASK /SAVE BITS 06-08
0377 3147 DCA REGC /SAVE FOR TYPING
0400 1147 TAD REGC /FETCH IT
0401 7041 CIA /2'S COMPLEMENT
0402 1146 TAD REG8 /COMPARE
0403 6201 CDF 00 /RESTORE DATA FIELD
0404 7657 SNA CLA /INCORRECT IF NOT ZERO
0405 4516 JMS I NERROR /CHECK WITH MONITOR
0406 4471 JMS I ERROR /PROBLEMS WITH NOISY DATA FIELD
0407 5645 TST05M /MESSAGE POINTER
0410 7402 HLT /ERROR HALT
0411 761 SKP CLA /GO TO NEXT TEST
0412 0345 TST05 /SCOPE LOOP; ISZ LOOP

```

/LMODE

/GATE SHAKER TEST

```

/
0413 4544 TST06, JMS I RANDOM /GET A RANDOM NUMBER
0414 0114 AND LMASK /SAVE BITS 07-11
0415 3146 DCA REGB /SAVE FOR OBSERVATION
0416 1146 TAD REGB /FETCH IT
0417 1104 TAD KLDF /ADD LDF
0420 3252 DCA NOW2 /STORE FOR EXECUTION
0421 6141 LINC /GO TO LINC MODE
0422 0641 LDF 00 /TRY SOME DATA FIELD
0423 0677 LDF 37 /NOISEMAKERS
0424 0660 LDF 20
0425 0650 LDF 10
0426 0644 LDF 04
0427 0642 LDF 02
0430 0641 LDF 01
0431 0665 LDF 25
0432 0652 LDF 12
0433 0647 LDF 07
0434 0670 LDF 30
0435 0640 LDF 00
0436 0641 LDF 01
0437 0642 LDF 02
0440 0643 LDF 03
0441 0644 LDF 04
0442 0645 LDF 05
0443 0646 LDF 06
0444 0646 LDF 06
0445 0647 LDF 07
0446 0650 LDF 10
0447 0657 LDF 17
0450 0667 LDF 27
0451 0677 LDF 37
0452 0000 NOW2, 0000 /EXECUTE ACTUAL LDF
0453 0500 IOB /PREPARE TO GET DATA FIELD
0454 6214 RDF /GET DATA FIELD
0455 0002 POP /GO TO PMODE
0456 7110 RAR CLL /JUSTIFY WITH REGB
0457 3147 DCA REGB /SAVE FOR TYPING
0460 1147 TAD REGB /FETCH IT
0461 7041 CIA /2'S COMPLEMENT
0462 1146 TAD REGB /COMPARE
0463 6201 CDF 00 /RESTORE DATA FIELD
0464 765 SNA CLA /INCORRECT IF NOT ZERO
0465 4516 JMS I NERROR /CHECK WITH MONITOR
0466 4471 JMS I ERROR /PROBLEMS WITH NOISY DATA FIELD
0467 5677 TST06M /MESSAGE POINTER
0470 7402 HLT /ERROR HALT
0471 7610 SKP CLA /GO TO NEXT TEST
0472 0413 TST06 /SCOPE LOOP: ISZ LOOP

```

/THE DATA FIELD IS NOW CONSIDERED TO BE TESTED,
 /NOW CHECK RIB
 /PMODE
 /CHECK INTERRUPT FACILITY.
 /

0473	6041	TST07,	TSF		/CHECK FOR FLAG
0474	4556		JMS I	SETFLG	/NOT UP; GO SET IT
0475	1122		TAD	PNTA	/GET ADDRESS RETURN
0476	3155		DCA	RETURN	/STORE IT
0477	3142		DCA	PPOINT	/ZERO THE PMODE SWITCH
0500	6001		ION		/ENABLE INTERRUPT
0501	7000		NOP		/WAIT
0502	6002	LOCA,	IOF		/DISABLE INTERRUPT
0503	7430		SZL		/CHECK LINK; INCORRECT IF ZERO
0504	4516		JMS I	NERROR	/CHECK WITH MONITOR
0505	4471		JMS I	ERROR	/INTERRUPT FAILED
0506	5731		TST07M		/MESSAGE POINTER
0507	7402		HLT		/ERROR HALT
0510	7610		SKP CLA		/GO TO NEXT TEST
0511	0473		TST07		/SCOPE LOOP; ISZ LOOP

/PMODE
 /NOW CHECK RIB
 /

0512	6041	TST08,	TSF		/CHECK FOR FLAG
0513	4556		JMS I	SETFLG	/NOT UP; GO SET IT
0514	1123		TAD	PNTB	/GET RETURN ADDRESS
0515	3155		DCA	RETURN	/STORE IT
0516	4544		JMS I	RANDOM	/GET RANDOM NUMBER
0517	0121		AND	PMASK	/SAVE BITS 06-08
0520	3146		DCA	REGB	/SAVE FOR OBSERVATION
0521	1146		TAD	REGB	/FETCH IT
0522	1101		TAD	KCDF	/ADD CDF
0523	3324		DCA	,+1	/STORE FOR EXECUTION
0524	0000		0000		/EXECUTE CDF
0525	6001		ION		/ENABLE INTERRUPT
0526	7000		NOP		/WAIT
0527	6002	LOCB,	IOF		/DISABLE INTERRUPT
0530	6234		RIB		/READ INTERRUPT BUFFER
0531	7006		RTL		/JUSTIFY WITH REGB
0532	7104		RAL CLL		/SOME MORE
0533	0121		AND	PMASK	/SAVE BITS 06-08
0534	3147		DCA	REGC	/SAVE FOR TYPING
0535	1147		TAD	REGC	/FETCH IT
0536	7041		CIA		/2'S COMPLEMENT
0537	1146		TAD	REGB	/COMPARE
0540	6201		COF	00	/RESTORE DATA FIELD
0541	7650		SNA CLA		/INCORRECT IF NOT ZERO
0542	4516		JMS I	NERROR	/CHECK WITH MONITOR
0543	4471		JMS I	ERROR	/LOAD SF OR RIB FAILED
0544	5752		TST08M		/MESSAGE POINTER
0545	7402		HLT		/ERROR HALT
0546	7610		SKP CLA		/GO TO NEXT TEST
0512	0512		TST08		/SCOPE LOOP; ISZ LOOP

```

/LMODE
/CHECK INTERRUPT FACILITY
/
TST9A, TSF /CHECK FOR FLAG
0550 6041 JMS I SETFLG /NOT UP; GO SET IT
0551 4556 TAD PNTCA /GET RETURN ADDRESS
0552 1125 AND K1777 /10 BIT ADDRESS
0553 0036 TAD KLJMP /ADD LINC JUMP
0554 1112 DCA LSET /STORE FOR EXECUTION
0555 3050 DCA LPOINT /ZERO THE LMODE SWITCH
0556 3065 CLL CML /SET LINK
0557 7120 LINC /GO TO LINC MODE
0562 6141 IOB /PREPARE TO EXECUTE IOT
0561 0500 ION /ENABLE INTERRUPTS
0562 6001 LNOP /WAIT
0563 0016 LOCCA, IOB /PREPARE TO EXECUTE IOT
0564 0500 IOF /DISABLE INTERRUPTS
0565 6002 PDP /BACK TO PMODE
0566 0002 SNL /CHECK LINK, INCORRECT IF SET
0567 7420 JMS I NERROR /CHECK WITH MONITOR
0570 4516 JMS I ERROR /INTERRUPT FAILED
0571 4471 TST9AM /MESSAGE POINTER
0572 6004 HLT /ERROR HALT
0573 7402 SKP CLA /GO TO NEXT TEST
0574 7610 TST9A /ISZ LOOP; SCOPE LOOP
0575 550

```

```

/LMODE
/CHECK RIB
TST09, TSF          /CHECK FOR FLAG
0576 6041           /NOT UP; GO SET IT
0577 4556           /GET RETURN ADDRESS
0600 1124           /10 BIT ADDRESS
0601 0036           /ADD LINC JUMP
0602 1112           /STORE IN RETURN ADDRESS
0603 3050           /GET RANDOM NUMBER
0604 4544           /SAVE BITS 07-11
0605 0114           /SAVE FOR COMPARISON
0606 3146           /FETCH IT
0607 1146           /ADD LDF
0610 1104           /STORE FOR EXECUTION
0611 3214           /SET LINK
0612 7120           /GO TO LINC MODE
0613 6141           /EXECUTE LDF
0614 0000           /PREPARE FOR IOT
0615 0500           /ENABLE INTERRUPT
0616 6001           /WAIT
0617 0016           /PREPARE FOR IOT
0620 0500           /DISABLE INTERRUPT
LOCC, IOB          /PREPARE FOR IOT
0621 6002           /READ INTERRUPT BUFFER
0622 0500           /JUSTIFY WITH REGB
0623 6234           /BACK TO PMODE
0624 0242           /SAVE BITS 07-11
0625 0002           /SAVE FOR TYPING
0626 0114           /FETCH IT
0627 3147           /2'S COMPLEMENT
0630 1147           /COMPARE
0631 7041           /RESTORE DATA FIELD
0632 1146           /INCORRECT IF NOT ZERO
0633 6201           /CHECK WITH MONITOR
0634 7657           /LMODE RIB FAILED
0635 4516           /MESSAGE POINTER
0636 4471           /ERROR HALT
0637 6025           /GO TO NEXT TEST
0640 7402           /SCOPE LOOP: ISZ LOOP
0641 7617
0642 0576

```

```

/PMODE
/DOES THE DATA FIELD SET TO ZERO FOR AN INTERRUPT
/
0643 6041 TST10, TSF /CHECK FLAG
0644 4556 JMS I SETFLG /NO UP; GO SET IT
0645 1126 TAD PNTD /GET RETURN ADDRESS
0646 3155 DCA RETURN1 /STORE IT
0647 4544 JMS I RANDOM /GET RANDOM NUMBER
0650 1121 AND PMASK /SAVE BITS 06-08
0651 3146 DCA REGB /SAVE FOR TYPING
0652 1146 TAD REGB /FETCH IT
0653 1101 TAD KCDF /ADD CDF
0654 3255 DCA ,+1 /STORE FOR EXECUTION
0655 0000 /EXECUTE CDF
0656 6001 ION /ENABLE INTERRUPT
0657 7000 NOP /WAIT
0660 6002 LOCD, IOF /DISABLE INTERRUPT
0661 6234 RIR /GET INTERRUPT BUFFER
0662 7006 RTL /JUSTIFY WITH REGB
0663 7004 RAL /SOME MORE
0664 3147 DCA REGC /SAVE FOR TYPING
0665 6214 RDF /READ DATA FIELD
0666 0121 AND PMASK /SAVE BITS 06-08
0667 3151 DCA REGD /STORE FOR TYPING
0670 1151 TAD REGD /FETCH IT
0671 6201 CDF 00 /RESTORE DATA FIELD
0672 7650 SNA CLA /INCORRECT IF NOT ZERO
0673 4516 JMS I NERROR /CHECK WITH MONITOR
0674 4471 JMS I ERROR /DATA FIELD FAILED TO ZERO
0675 6057 TST10M /MESSAGE POINTER
0676 7402 HLT /ERROR HALT
0677 7611 SKP CLA /GO TO NEXT TEST
0700 0643 TST10 /SCOPE LOOP; ISZ LOOP

```

```

/LMODE
/DOES THE DATA FIELD SET TO ZERO FOR AN INTERRUPT
/
0701 6041 TST11, TSF /CHECK FLAG
0702 4556 JMS I SETFLG /NOT UP; GO SET IT
0703 1127 TAD PNTE /GET RETURN ADDRESS
0704 0036 AND K1777 /12 BIT ADDRESS
0705 1112 TAD KLJMP /ADD LINC MODE JMP
0706 3057 DCA LSET /STORE IT
0707 4544 JMS I RANDOM /GET RANDOM NUMBER
0710 0114 AND LMASK /SAVE BITS 07-11
0711 3146 DCA REGB /STORE FOR TYPING
0712 1146 TAD REGB /FETCH IT
0713 1104 TAD KLDF /ADD LDF
0714 3316 DCA ,+2 /STORE FOR EXECUTION
0715 6141 LINC /GO TO LINC MODE
0716 0000 0000 /EXECUTE LDF
0717 0500 IOB /PREPARE FOR IOT
0720 6001 IOB /ENABLE INTERRUPT
0721 0016 LNOP /WAIT
0722 0500 LOCE, IOB /PREPARE FOR IOT
0723 6002 IOF /DISABLE INTERRUPT
0724 0500 IOB /PREPARE FOR IOT
0725 6234 RIB /READ INTERRUPT BUFFER
0726 0242 ROL 2 /JUSTIFY WITH REGB
0727 4147 STC REGD /SAVE FOR TYPING
0730 0500 IOB /PREPARE FOR IOT
0731 6214 RDF /READ DATA FIELD
0732 0002 PDP /BACK TO PMODE
0733 7117 RAR CLL /JUSTIFY WITH REGB
0734 3151 DCA REGD /SAVE FOR TYPING
0735 1151 TAD REGD /FETCH IT
0736 6201 CDF 00 /RESTORE DATA FIELD
0737 7657 SNA CLA /INCORRECT IF NOT ZERO
0740 4516 JMS I NERROR /CHECK WITH MONITOR
0741 4471 JMS I ERROR /DATA FIELD FAILED TO ZERO ON INTERRUPT
0742 6123 TST11M /MESSAGE POINTER
0743 7402 HLT /ERROR HALT
0744 761 SKP CLA /GO TO NEXT TEST
0745 0701 TST11 /SCOPE LOOP; ISZ LOOP

```



```

/PMODE
/DOES DCA I--TAD I WORK FOR ALL DATA FIELDS
/
0746 7300  TST12:  CLA CLL           /CLEAR AC
0747 4474          JMS I   GETBNK        /GET NEXT BANK
0750 7450          SNA           /DONE?
0751 5562          JMP I   TST13N       /YES, NEXT TEST VIA PAGE 0
0752 3146          DCA   REGB        /SAVE BANK
0753 1025          TAD   K0020       /GET CONSTANT
0754 3145          DCA   REGA        /SET REGA = 20
0755 1146  TST12A, TAD   REGB        /GET CURRENT BANK
0756 7006          RTL           /JUSTIFY
0757 7004          RAL           /JUSTIFY
0760 1101          TAD   KCDF        /GET CDF
0761 3363          DCA   EXC12       /STORE FOR EXECUTION
0762 1076          TAD   K5252       /GET CONSTANT
0763 0000  EXC12, 0000          /EXECUTE CDF
0764 3545          DCA I  REGA        /STORE IN TEST BANK
0765 1545          TAD I  REGA        /GET IT
0766 6201          CDF   00          /RESTORE DATA FIELD
0767 3147          DCA   REGC        /SAVE DATA
0770 1147          TAD   REGC        /FETCH IT
0771 7041          CIA           /2'S COMPLEMENT
0772 1076          TAD   K5252       /COMPARE
0773 6201          CDF   00          /RESTORE DATA FIELD
0774 7650          SNA CLA        /INCORRECT IF NOT ZERO
0775 4516          JMS I  NERROR      /CHECK WITH MONITOR
0776 4471          JMS I  ERROR      /DCA I OR TAD I FAILED
0777 6167          TST12M         /MESSAGE POINTER
1000 7402          HLT           /ERROR HALT
1001 7610          SKP CLA        /TO NEXT BANK
1002 1755          TST12A         /SCOPE LOOP; ISZ LOOP
1003 5561          JMP I   TST12N       /NEXT BANK VIA PAGE 0

```

/LMODE

/DOES STA-LDA WORK FOR ALL DATA FIELDS

/

1004	7300	TST13,	CLA	CLL	/CLEAR AC
1005	4475		JMS	I GETBNL	/FIND NEXT BANK
1006	7450		SNA		/DONE
1007	5244		JMP	TST14	/YES, GO TO NEXT TEST
1010	3146		DCA	REG8	/SAVE BANK
1011	1077		TAD	K6020	/GET CONSTANT
1012	3145		DCA	REGA	/SET REGA TO 6020
1013	1146		TAD	REG8	/GET CURRENT BANK
1014	1104		TAD	KLDF	/ADD LDF
1015	3222		DCA	EXC13	/STORE FOR EXECUTION
1016	1145	TST13A,	TAD	REGA	/GET ADDRESS
1017	3013		DCA	LREG1	/STORE FOR INDIRECT ACCESS
1020	1076		TAD	K5252	/GET CONSTANT
1021	6141		LINC		/GO TO LMODE
1022	0000	EXC13,	0000		/EXECUTE LDF
1023	1053		STA	LREG1	/STORE INDIRECT TO OF
1024	1013		LDA	LREG1	/FETCH NUMBER
1025	0640		LDF		/RESTORE DATA FIELD
1026	0002		PDP		/TO PMODE
1027	3147		DCA	REGC	/SAVE FOR TYPING
1030	1147		TAD	REGC	/FETCH IT
1031	7041		CIA		/2'S COMPLEMENT
1032	1076		TAD	K5252	/COMPARE
1033	6201		CDF	00	/RESTORE DATA FIELD
1034	7650		SNA	CLA	/INCORRECT IF NOT ZERO
1035	4516		JMS	I ERROR	/CHECK WITH MONITOR
1036	4471		JMS	I ERROR	/STA OR LDA FAILED
1037	6225		TST13M		/MESSAGE POINTER
1040	7402		HLT		/ERROR HALT
1041	7610		SKP	CLA	/NEXT TEST
1042	1016		TST13A		/SCOPE LOOP; ISZ LOOP
1043	5204		JMP	TST13	/NEXT BANK

/
 /TEST THE DJR FUNCTION FOR ALL COMBINATIONS

/
 /LMODE
 /DOES DJR NOT FUNCTION WHEN NOT SET?

1044	7300	TST14,	CLA CLL	/CLEAR AC
1045	1076		TAD K5252	/GET CONSTANT
1046	3000		DCA 0	/SET 0
1047	6141		LINC	/GO TO LINC MODE
1050	7051		LJMP ,+1	/DO A LINC JUMP
1051	0002		PDP	/BACK TO P MODE
1052	1000		TAD 0	/GET 0
1053	7041		CIA	/2'S COMPLEMENT
1054	1076		TAD K5252	/ADD CONSTANT
1055	7640		SZA CLA	/WAS LOCATION 0 CHANGED?
1056	4516		JMS I NERROR	/YES; CHECK WITH MONITOR
1057	4471		JMS I ERROR	/LINC JUMP SAVE RETURN FAILED
1060	6261		TST14M	/MESSAGE POINTER
1061	7402		HLT	/ERROR HALT
1062	7610		SKP CLA	/TO NEXT TEST
1063	1044		TST14	/SCOPE LOOP; ISZ LOOP

/LMODE
 /DOES DJR FUNCTION WHEN IT'S SET?

1064	7300	TST15,	CLA CLL	/CLEAR AC
1065	1076		TAD K5252	/GET CONSTANT
1066	3000		DCA 0	/SET 0
1067	6141		LINC	/TO L MODE
1070	0006		DJR	/DISABLE JUMP SAVE RETURN
1071	7072		LJMP ,+1	/DO A LINC JUMP
1072	0002		PDP	/BACK TO P MODE
1073	1000		TAD 0	/GET 0
1074	7041		CIA	/2'S COMPLEMENT
1075	1076		TAD K5252	/COMPARE WITH CONSTANT
1076	7650		SNA CLA	/DID DJR WORK?
1077	4516		JMS I NERROR	/CHECK WITH MONITOR
1100	4471		JMS I ERROR	/DJR FAILED
1101	6316		TST15M	/MESSAGE POINTER
1102	7402		HLT	/ERROR HALT
1103	7610		SKP CLA	/TO NEXT TEST
1104	1064		TST15	/SCOPE LOOP; ISZ LOOP

```

/LMODE
/DOES A LINC JUMP CLEAR DJR?
/
1105 7302 TST16, CLA CLL /CLEAR AC
1106 1076 TAD K5252 /GET CONSTANT
1107 3000 DCA 0 /SET 0
1110 6141 LINC /TO LMODE
1111 0000 DJR /DISABLE JUMP SAVE RETURN
1112 7113 LJMP ,+1 /DO A LINC JUMP
1113 7114 LJMP ,+1 /DO ANOTHER LINC JUMP
1114 0002 PDP /BACK TO PMODE
1115 1000 TAD 0 /GET 0
1116 7041 CIA /2'S COMPLEMENT
1117 1076 TAD K5252 /COMPARE WITH CONSTANT
1120 7640 SZA CLA /DID DJR CLEAR?
1121 4516 JMS I NERROR /CHECK MONITOR
1122 4471 JMS I ERROR /DJR FAILED TO CLEAR
1123 6344 TST16M /MESSAGE POINTER
1124 7402 HLT /ERROR HALT
1125 7610 SKP CLA /TO NEXT TEST
1126 1105 TST16 /SCOPE LOOP; ISZ LOOP

/PMODE
/DOES JUMP SAVE RETURN WORK FOR 8 MODE JUMPS?
/
1127 7300 TST17, CLA CLL /CLEAR AC
1130 1076 TAD K5252 /GET CONSTANT
1131 3000 DCA 0 /SET 0
1132 5333 JMP ,+1 /DO AN 8 MODE JUMP
1133 1000 TAD 0 /GET 0
1134 7041 CIA /2'S COMPLEMENT
1135 1076 TAD K5252 /COMPARE WITH CONSTANT
1136 7650 SNA CLA /DID WE SAVE IN ERROR?
1137 4516 JMS I NERROR /CHECK MONITOR
1140 4471 JMS I ERROR /JUMP SAVE RETURN OPERATED IN ERROR
1141 6371 TST17M /MESSAGE POINTER
1142 7402 HLT /ERROR HALT
1143 7610 SKP CLA /TO NEXT TEST
1144 1127 TST17 /ISZ LOOP; SCOPE LOOP

```

/PMODE

/DOES JUMP SAVE RETURN WORK FOR NON-JUMP COMMANDS?

/

1145	7300	TST18,	CLA	CLL		/CLEAR AC
1146	1076		TAD		K5252	/GET CONSTANT
1147	3000		DCA		0	/SET 0
1150	6002		IOF			/IOF LOOKS LIKE LINC JUMP
1151	1000		TAD		0	/GET 0
1152	7041		CIA			/2'S COMPLEMENT
1153	1076		TAD		K5252	/COMPARE WITH CONSTANT
1154	7650		SNA	CLA		/DID CELL 0 CHANGE?
1155	4516		JMS	I	NERROR	/CHECK MONITOR
1156	4471		JMS	I	ERROR	/IOF CHANGED CELL 0
1157	6415		TST18M			/MESSAGE POINTER
1160	7402		HLT			/ERROR HALT
1161	7610		SKP	CLA		/TO NEXT TEST
1162	1145		TST18			/SCOPE LOOP; ISZ LOOP

/LMODE

/DOES JUMP SAVE RETURN WORK FOR NON-JUMP COMMANDS?

/

1163	7300	TST19,	CLA	CLL		/CLEAR AC
1164	1076		TAD		K5252	/GET CONSTANT
1165	3000		DCA		0	/SET 0
1166	6141		LINC			/GO TO LMODE
1167	0500		IOB			/PREPARE FOR IOT
1170	6002		IOF			/DISABLE INTERRUPTS
1171	0002		PDP			/BACK TO PMODE
1172	1000		TAD		0	/FETCH 0
1173	7041		CIA			/2'S COMPLEMENT
1174	1076		TAD		K5252	/ADD CONSTANT
1175	7650		SNA	CLA		/EQUAL?
1176	4516		JMS	I	NERROR	/CHECK MONITOR
1177	4471		JMS	I	ERROR	/IOB/IOF CAUSED LOC 0000 TO ALTER
1200	6441		TST19M			/MESSAGE POINTER
1201	7402		HLT			/ERROR HALT
1202	7610		SKP	CLA		/TO NEXT TEST
1203	1163		TST19			/ISZ LOOP; SCOPE LOOP

```

/LMODE
/DOES DJR CLEAR WITH 8 MODE JUMP?
/
1204 7300 TST20, CLA CLL /CLEAR AC
1205 1076 TAD K5252 /GET CONSTANT
1206 300 DCA 0 /SET 0
1207 6141 LINC /TO LMODE
1210 0006 DJR /DISABLE JUMP RETURN SAVE
1211 0002 PDP /TO PMODE
1212 5213 JMP ,+1 /JUMP
1213 6141 LINC /TO LMODE
1214 7215 LJMP ,+1 /JUMP
1215 0002 PDP /TO PMODE
1216 1000 TAD 0 /FETCH 0
1217 7041 CIA /2'S COMPLEMENT
1220 1076 TAD K5252 /ADD CONSTANT
1221 7650 SNA CLA /EQUAL?
1222 4516 JMS I NERROR /CHECK MONITOR
1223 4471 JMS I ERROR /8 MODE JUMP CLEARED DJR
1224 6465 TST20M /MESSAGE POINTER
1225 7402 HLT /ERROR HALT
1226 7610 SKP CLA /TO NEXT TEST
1227 1204 TST20 /ISZ LOOP; SCOPE LOOP

```

```

/PMODE
/DOES DJR INHIBIT 8 MODE INTERRUPT SAVE?
/
1230 730      TST21,  CLA CLL          /CLEAR AC
1231 1130     TAD      PNIF      /GET RETURN POINTER TO LOCF
1232 3155     OCA      RETURN    /SET UP INTERRUPT HANDLER
1233 1076     TAD      K5252     /GET CONSTANT
1234 3000     OCA      0         /STORE IN 0
1235 6041     TSF              /FLAG SET?
1236 4556     JMS I   SETFLG     /NO, GO SET IT
1237 6141     LINC             /TO LMODE
1240 0006     DJR              /SET DJR
1241 0002     PDP              /TO PMODE
1242 6001     ION              /ENABLE INTERRUPTS
1243 7000     NOP              /WAIT
1244 6002     IOF              /DISABLE INTERRUPTS
1245 7410     SKP              /IF NO INTERRUPT, THIS CAUSES ERROR
1246 1000     LOCF,  TAD      0   /GET 0
1247 7041     CIA              /2'S COMPLEMENT
1250 1076     TAD      K5252     /ADD CONSTANT
1251 7640     SZA CLA          /EQUAL?
1252 4516     JMS I   NERROR    /CHECK MONITOR
1253 4471     JMS I   ERROR     /DJR INHIBITED 8 MODE INTERRUPT
1254 6506     TST21M          /MESSAGE POINTER
1255 7402     HLT              /ERROR HALT
1256 7610     SKP CLA          /TO NEXT TEST
1257 1230     TST21           /ISZ LOOP; SCOPE LOOP
1260 7340     CLA CLL  CMA      /SET AC=7777
1261 3145     OCA      REGA     /PRESET REGA FOR NEXT TEST

```

```

/PMODE
/WILL NON-EXISTANT MEMORY DETECT WORK FOR ALL BANKS?
/
1262 7300 TST22, CLA CLL /CLEAR AC
1263 1066 TAD BANK /GET AVAILABLE MEMORY
1264 3151 DCA REGD /SAVE IT
1265 1151 TAD REGD /FETCH IT
1266 7041 CIA /2'S COMPLEMENT
1267 1022 TAD K0007 /ADD MAXIMUM MEMORY
1270 3146 DCA REGB /SAVE IT
1271 1146 TAD REGB /FETCH IT
1272 7450 SNA /HOW MUCH WAS LEFT?
1273 5370 JMP TST23 /NONE; 32K MACHINE
1274 7010 RAR /CHECK BIT 11
1275 7620 SNL CLA /IS MEMORY ODD OR EVEN?
1276 5342 JMP READ1 /NEXT BANK IS EVEN
1277 5316 JMP READ0 /NEXT BANK IS ODD
1300 7300 BAK22, CLA CLL /CLEAR AC
1301 6201 CDF 00 /RESTORE DATA FIELD
1302 1151 TAD REGD /GET LAST BANK TESTED
1303 7041 CIA /2'S COMPLEMENT
1304 1022 TAD K0007 /COMPARE WITH MAXIMUM
1305 7640 SZA CLA /DONE?
1306 5342 JMP READ1 /NO, TEST NEXT BANK
1307 4516 JMS I NERROR /CHECK MONITOR
1310 4471 FAL22, JMS I ERROR /NON-EXIST DETECT FAILED
1311 6535 TST22M /MESSAGE POINTER
1312 7402 HLT /ERROR HALT
1313 7410 SKP /TO NEXT TEST
1314 1262 TST22 /ISZ LOOP; SCOPE LOOP
1315 5370 JMP TST23 /JUMP OVER READ ROUTINES

```


/PMODE

/READ 0 ROUTINE FOR TST22 (USED ONLY ONCE PER PASS)

```

/
READ0,  CLA  CLL           /CLEAR AC
        ISZ   REGD         /INCREMENT NON-EXIST BANK
1316  7300  TAD   REGD         /FETCH BANK NUMBER
1317  2151  RTL           /JUSTIFY
1320  1151  RAL           /JUSTIFY
1321  7006  TAD   KCDF         /FETCH CDF 00
1322  7004  DCA   ,+1         /STORE FOR EXECUTION
1323  1101  CMA           /EXECUTE CDF
1324  3325  DCA   AUTO11      /SET AC=7777
1325  0000  DCA   0           /SETUP POINTER REGISTER
1326  7040  TAD   I AUTO11    /SETUP COUNTER
1327  3015  SZA           /FETCH NON-EXISTANT WORD
1330  3000  JMP   ,+4         /ZERO?
1331  1415  JMP   ,+4         /NO, ENTER ERROR ROUTINE
1332  7440  ISZ   0           /INCREMENT COUNTER
1333  5337  JMP   ,+4         /TRY NEXT LOCATION
1334  2000  JMP   BAK22      /BANK FINISHED; RETURN
1335  5331  CDF   00         /RESTORE DATA FIELD
1336  5300  DCA   REGC         /SAVE AC FOR TYPEOUT
1337  6201  JMP   FAL22      /TO ERROR MONITOR
1340  3147
1341  5310
    
```

/PMODE

/READ 1 ROUTINE FOR TEST 22

```

/
READ1,  CLA  CLL           /CLEAR AC
        ISZ   REGD         /INCREMENT NON-EXIST BANK
1342  7300  TAD   REGD         /FETCH IT
1343  2151  RTL           /JUSTIFY
1344  1151  RAL           /JUSTIFY
1345  7006  TAD   KCDF         /FETCH CDF 00
1346  7004  DCA   ,+1         /STORE FOR EXECUTION
1347  1101  CMA           /EXECUTE CDF
1350  3351  DCA   AUTO11      /SET AC=7777
1351  0000  DCA   0           /SETUP POINTER REGISTER
1352  7040  TAD   I AUTO11    /SETUP COUNTER
1353  3015  SZA           /FETCH NON-EXISTANT WORD
1354  3000  JMP   ,+4         /COMPLEMENT
1355  1415  JMP   ,+4         /ZERO?
1356  7040  JMP   ,+4         /NO, ENTER ERROR ROUTINE
1357  7440  ISZ   0           /INCREMENT COUNTER
1360  5364  JMP   ,+5         /TRY NEXT LOCATION
1361  2000  JMP   BAK22      /BANK FINISHED; RETURN,
1362  5355  CMA           /RESTORE AC
1363  5300  DCA   REGC         /SAVE FOR TYPEOUT
1364  7040  CDF   00         /RESTORE DATA FIELD
1365  3147  JMP   FAL22      /TO ERROR MONITOR
1366  6201
1367  5310
    
```

```
/PMODE
/NOW SET UP EXTENDED MEMORY FOR FURTHER TESTING
/
1370 7307 TST23, CLA CLL /CLEAR AC
1371 4474 JMS I GETBNK /GO FIND NEXT BANK
1372 7457 SNA /DONE?
1373 5564 JMP I TST24N /YES, EXIT
1374 3376 DCA ,+2 /NO SAVE BANK FOR EXECUTION
1375 4553 JMS I RELOCR /GO RELOCATE ALL OF MEMORY
1376 0000 0000 /TARGET BANK
1377 7777 7777 /ORG,
1420 7777 7777 /DEST,
1421 7777 7777 /LENGTH
1422 5563 JMP I TST23N /DO IT AGAIN
```

/
 /TRY A CIF-ION-JMP TO ALL BANKS
 /

1403	7320	TST24,	CLA	CLL	/CLEAR AC
1404	3142		DCA	PPOINT	/ZERO THE PMODE SWITCH
1405	4474		JMS I	GETBNK	/GO GET THE NEXT BANK
1406	7450		SNA		/DONE?
1407	5246		JMP	TST25	/EXIT
1410	7006		RTL		/JUSTIFY
1411	7004		RAL		/JUSTIFY
1412	3146		DCA	REGB	/SAVE IT
1413	1146		TAD	REGB	/FETCH IT
1414	1102		TAD	KCIF	/ADD CIF
1415	3222		DCA	,+5	/STORE FOR EXECUTION
1416	1131		TAD	PNTG	/GET RETURN ADDRESS
1417	3155		DCA	RETURN	/SET UP HANDLER
1420	6041		TSF		/FLAG SET?
1421	4556		JMS I	SETFLG	/NO, GO SET IT
1422	0000			0000	/EXECUTE CIF
1423	6001		ION		/ENABLE INTERRUPTS
1424	5224		JMP	.	/WAIT
1425	6002		IOF		/DISABLE INTERRUPTS
1426	6202		CIF	00	/BACK TO BANK 0
1427	5555		JMP I	RETURN	/JUMP DOWN
1430	1143	LOCG,	TAD	PREG	/GET INTERRUPT SF
1431	0027		AND	K0070	/CLEAR OUT ALL BUT 06,07,08
1432	3147		DCA	REGC	/SAVE IT
1433	1147		TAD	REGC	/FETCH IT
1434	7041		CIA		/2'S COMPLEMENT
1435	1146		TAD	REGB	/COMPARE
1436	7650		SNA	CLA	/EQUAL?
1437	4516		JMS I	NERROR	/CHECK MONITOR
1440	4471		JMS I	ERROR	/IF FAILED TO LOAD
1441	6575		TST24M		/MESSAGE POINTER
1442	7402		HLT		/ERROR HALT
1443	7410		SKP		/TO NEXT TEST
1444	1413		TST24+10		/ISZ LOOP; SCOPE LOOP
1445	5203		JMP	TST24	

```

/LMODE
/TRY A LIF-IOB-ION-NOP TO ALL BANKS
/
1446 7300 TST25, CLA CLL /CLEAR AC
1447 3065 DCA LPOINT /ZERO THE LMODE SWITCH
1450 4474 JMS I GETBNK /GET NEXT BANK
1451 7450 SNA /DONE?
1452 5314 JMP TST26 /EXIT
1453 3146 DCA REGB /SAVE FIELD
1454 1146 TAD REGB /FETCH IT
1455 7006 RTL /JUSTIFY
1456 1111 TAD KLIF /MAKE IT A LIF
1457 3266 DCA ,+7 /STORE FOR EXECUTION
1460 1132 TAD PNTH /GET RETURN ADDRESS
1461 1112 TAD KLJMP /MAKE IT A LINC JUMP
1462 3050 DCA LSET /STORE FOR RETURN
1463 6041 TSF /FLAG SET?
1464 4556 JMS I SETFLG /NO, GO SET IT
1465 6141 LINC /GO TO LMODE
1466 0000 0000 /EXECUTE LIF
1467 0500 IOB
1470 6001 ION /ENABLE INTERRUPTS
1471 0016 LNOP /WAIT
1472 7472 LJMP /WAIT
1473 2064 LOCH, ADD LREG /GET SAVE FIELD
1474 1560 BCL 20 /CLEAR OUT ALL BUT IF
1475 6037 6037 /
1476 0305 ROR 5 /JUSTIFY
1477 4147 STC REGC /SAVE IT
1500 0002 PDP /BACK TO PMODE
1501 1146 TAD REGB /GET TARGET IF
1502 7041 CIA /2'S COMPLEMENT
1503 1147 TAD REGC /GET CURRENT IF
1504 7650 SNA CLA /EQUAL?
1505 4516 JMS I NERROR /CHECK MONITOR
1506 4471 JMS I ERROR /IF FAILED TO LOAD
1507 6631 TST25M /MESSAGE POINTER
1510 7402 HLT /ERROR HALT
1511 741 SKP /TO NEXT TEST
1512 1454 TST25+6 /ISZ LOOP; SCOPE LOOP
1513 5246 JMP TST25

```

```

/PMODE
/NOW GO TO EXTENDED MEMORY AND TEST RMF
/
1514 7300 TST26 CLA CLL /CLEAR AC
1515 4474 JMS I GETBNK /GET NEXT BANK
1516 7450 SNA /DONE?
1517 5565 JMP I TST27N /YES, NEXT TEST
1520 7006 RTL /JUSTIFY
1521 7004 RAL /JUSTIFY
1522 3146 DCA REGB /SAVE BANK
1523 6041 TSF /FLAG SET?
1524 4556 JMS I SETFLG /NO, GO SET IT
1525 7040 CMA /SET AC=7777
1526 3142 DCA PPOINT /SET P SWITCH#1
1527 1146 TAD REGB /GET BANK
1530 1102 TAD KCIF /MAKE IT A CIF N
1531 3332 DCA ,+1 /STORE FOR EXECUTION
1532 0000 0000 /EXECUTE CIF
1533 6001 ION /ENABLE INTERRUPTS
1534 5334 JMP , /GO TO EXTENDED MEMORY AND WAIT
1535 6002 IOF /DISABLE INTERRUPTS
1536 6224 RIF /GET INSTRUCTION FIELD
1537 3550 DCA I REGCN /SAVE IT
1540 6202 CIF 00 /BACK TO FIELD 0
1541 5342 JMP ,+1 /CHANGE FIELDS
1542 7300 CLA CLL
1543 1146 TAD REGB /GET TARGET FIELD
1544 7041 CIA /2'S COMPLEMENT
1545 1147 TAD REGC /COMPARE WITH ACTUAL FIELD
1546 7650 SNA CLA /EQUAL?
1547 4516 JMS I NERROR /CHECK MONITOR
1550 4471 JMS I ERROR /CIF FAILED TO FIND PROPER IF
1551 6664 TST26M /MESSAGE POINTER
1552 7402 HLT /ERROR HALT
1553 7410 SKP /TO NEXT TEST
1554 1523 TST26+7 /ISZ LOOP; SCOPE LOOP
1555 5314 JMP TST26 /DO NEXT BANK

```

/PMODE
 /INTERRUPT INHIBIT TEST BANK 0 - BANK N - BANK 0
 /

1556	7300	TST27,	CLA	CLL		/CLEAR AC
1557	1133		TAD	PNTI		/GET RETURN
1560	3155		DCA	RETURN		/SET UP HANDLER
1561	4474		JMS I	GETBNK		/GET NEXT BANK
1562	7450		SNA			/DONE
1563	5566		JMP I	TST26N		/YES, GO TO NEXT TEST
1564	3146		DCA	REGB		/SAVE BANK
1565	1146		TAD	REGB		/FETCH IT
1566	7006		RTL			/JUSTIFY FOR CIF
1567	7004		RAL			/JUSTIFY
1570	1102		TAD	KCIF		/MAKE IT A CIF
1571	3374		DCA	,+3		/STORE FOR EXECUTION
1572	6241		TSF			/FLAG SET?
1573	4556		JMS I	SETFLG		/NO, GO SET IT
1574	7000		0000			/EXECUTE CIF
1575	7000		NOP			/SPACER
1576	5377		JMP	,+1		/GO TO UPPER MEMORY
1577	7000		NOP			/WAIT FOR INTERRUPT
1600	7000		NOP			/WAIT FOR INTERRUPT
1601	6002		IOF			/TO HERE IF NO INTERRUPT
1602	6202		CIF	00		/BACK TO BANK 0
1603	5204		JMP	,+1		/JUMP INTO MONITOR
1604	4516		JMS I	NERROR		/INTERRUPT OK; CHECK MONITOR
1605	4471	LOCI,	JMS I	ERROR		/PMODE INTERRUPT INHIBIT FAILED
1606	6721		TST27M			/MESSAGE POINTER
1607	7402		HLT			/ERROR HALT
1610	7410		SKP			/TO NEXT TEST
1611	1565		TST27+7			/ISZ LOOP; SCOPE LOOP
1612	5565		JMP I	TST27N		/DO NEXT BANK

```

/LMODE
/INTERRUPT INHIBIT TEST BANK 0 -BANK N- BANK Z
/
1613 7300 TST28, CLA CLL /CLEAR AC
1614 3065 DCA LPOINT /CLEAR HANDLER SWITCH
1615 1134 TAD PNTJ /GET ERROR RETURN
1616 1112 TAD KLJMP /MAKE IT A LINC JUMP
1617 3050 DCA LSET /PLACE IT IN HANDLER
1620 4474 JMS I GETBNK /GET NEXT 4K BANK
1621 7450 SNA /DONE?
1622 5263 JMP TST29 /YES, NEXT TEST
1623 3146 DCA REGB /SAVE BANK
1624 1146 TAD REGB /FETCH IT
1625 7006 RTL /JUSTIFY FOR LMODE LIF
1626 1111 TAD KLIF /MAKE IT A LIF N
1627 3233 DCA ,+4 /STORE FOR EXECUTION
1630 6041 TSF /FLAG SET?
1631 4556 JMS I SETFLG /NO, GO SET IT
1632 6141 LINC /TO LINC MODE
1633 0600 LIF /EXECUTE LIF N
1634 0500 IOB /
1635 6001 ION /ENABLE INTERRUPTS (SHOULD INHIBIT)
1636 7637 LJMP ,+1 /TO EXTENDED MEMORY
1637 0016 LNOP /WAIT FOR INTERRUPT
1640 0600 LIF 0 /LOAD IB
1641 0500 IOB /
1642 6001 ION /ENABLE INTERRUPT AGAIN
1643 7644 LJMP ,+1 /BACK TO BANK 0
1644 0016 LNOP /WAIT FOR INTERRUPT
1645 0500 IOB /
1646 6002 IOF /DISABLE INTERRUPT
1647 0002 PDP /BACK TO PMODE
1650 5254 JMP ,+4 /TO NON-ERROR
1651 0002 LOCJ, PDP /BACK HERE IF INTERRUPT OCCURS
1652 6002 IOF /DISABLE INTERRUPT
1653 7410 SKP /SKIP INTO ERROR
1654 4516 JMS I NERROR /CHECK MONITOR
1655 4471 JMS I ERROR /LIF FAILED TO INHIBIT INTERRUPT
1656 6756 TST28M /MESSAGE POINTER
1657 7402 HLT /ERROR HALT
1660 7610 SKP CLA /TO NEXT TEST
1661 1624 TST28+11 /ISZ LOOP; SCOPE LOOP
1662 5213 JMP TST28 /NEXT BANK

```

/LMODE

/INTERRUPT INHIBIT TEST; DOES JMP 0 CLEAR INT INH?

/

1663	7300	TST29,	CLA	CLL	/CLEAR AC
1664	3065		DCA	LPOINT	/SET L SWITCH TO OFF
1665	1135		TAD	PNTK	/GET RETURN
1666	1112		TAD	KLJMP	/MAKE IT A LINC JUMP
1667	3050		DCA	LSET	/PUT IT IN HANDLER
1670	4474		JMS	I GETBNK	/GET NEXT BANK
1671	7450		SNA		/DONE?
1672	5347		JMP	TST30	/YES, NEXT TEST
1673	3146		DCA	REGB	/SAVE TARGET
1674	1146		TAD	REGB	/FETCH IT
1675	7006		RTL		/JUSTIFY FOR LIF
1676	1111		TAD	KLIF	/MAKE IT A LIF N
1677	3325		DCA	EX29	/STORE FOR EXECUTION
1700	6041		TSF		/FLAG SET?
1701	4556		JMS	I SETFLG	/NO, GO SET IT
1702	3014		DCA	PINT	/SET UP AUTO-INDEX
1703	1146		TAD	REGB	/GET TARGET
1704	7006		RTL		/JUSTIFY FOR CDF
1705	7004		RAL		/JUSTIFY
1706	1101		TAD	KCDF	/MAKE IT A CDF N
1707	3310		DCA	,+1	/STORE FOR EXECUTION
1710	0000		0000		/EXECUTE CDF
1711	7020		CML		/SET LINK
1712	1105		TAD	KIOB	/GET IOB
1713	3414		DCA	I PINT	/CELL 0001, BANK N
1714	1106		TAD	KIOF	/GET IOF
1715	3414		DCA	I PINT	/CELL 0002, BANK N
1716	1325		TAD	,+7	/GET LIF N
1717	3414		DCA	I PINT	/CELL 0003, BANK N
1720	1050		TAD	LSET	/GET LUMP LOCK
1721	3414		DCA	I PINT	/CELL 0004, BANK N
1722	6201		CDF	00	/RESTORE DF
1723	1107		TAD	KLNOP	/GET NOP
1724	6141		LINC		/TO L MODE
1725	0000	EX29,	0000		/EXECUTE LIF
1726	1500		IOB		
1727	6001		ION		
1730	7731		LJMP	,+1	/TO UPPER MEMORY
1731	0016		LNOP		/WAIT FOR INTERRUPT
1732	4000		STC	0	/SET UP 0
1733	6000		LJMP	0	/JMP 0

1734	0002	LOCK,	PDP	/TO HERE AFTER INTERRUPT OR JMP 0
1735	6202		CIF 00	/BACK TO BANK 0 IF NOT THERE
1736	5337		JMP ,+1	/TO LOWER MEMORY
1737	7430		SZL	/SKIP MEANS INTERRUPT NOT INHIBITED
1740	4516		JMS I NERROR	/CHECK MONITOR
1741	4471		JMS I ERROR	/JMP 0 INT INH IN ERROR
1742	7013		TST29M	/MESSAGE POINTER
1743	7402		HLT	/ERROR HALT
1744	7410		SKP	/TO NEXT TEST
1745	1674		TST29+11	/ISZ LOOP; SCOPE LOOP
1746	5263		JMP TST29	/NEXT BANK

```

/LMODE
/WILL DJR-JMP 0 LOAD THE IF?
/
1747 7340 TST30, CLA CLL CMA /CLEAR AC
1750 3065 DCA LPOINT /SWITCH=1
1751 1136 TAD PNTL /GET RETURN
1752 2036 AND K1777 /CLEAR BITS 0,1
1753 1112 TAD KLJMP /MAKE IT A LINC JUMP
1754 3057 DCA LSET /PUT IT IN HANDLER (WE WON'T USE INTERRUPTS)
1755 4474 JMS I GETBNK /GET NEXT BANK
1756 7450 SNA /DONE?
1757 5577 JMP I TST32N /YES, NEXT TEST
1760 7006 RTL /JUSTIFY FOR LIF
1761 7001 IAC /ADD CURRENT IF
1762 3146 DCA REGB
1763 1146 TST30X, TAD REGB
1764 1111 TAD KLIF /MAKE IT A LIF N
1765 3541 DCA I PNT30N /STORE FOR EXECUTION
1766 1036 TAD K1777 /SET AC = 1777 FOR IF 01
1767 3014 DCA PINT /SET UP AUTO-INDEX
1770 1146 TAD REGB /GET TARGET
1771 7004 RAL /JUSTIFY FOR CDF
1772 121 AND PMASK /JUSTIFY
1773 1101 TAD KCDF /MAKE IT A CDF N
1774 3375 DCA ,+1 /STORE FOR EXECUTION
1775 0000 0000 /EXECUTE CDF N
1776 1105 TAD KIOB /IOB
1777 7410 SKP /WASTE A SPACE
2000 0000 0000 /LINC JUMP SAVE
2001 3414 DCA I PINT /CELL 2000, BANK N
2002 1117 TAD KRIF /RIF LINC MODE (5 BITS)
2003 3414 DCA I PINT /CELL 2001, BANK N
2004 1057 TAD LSET /LJMP LOCL
2005 3414 DCA I PINT /CELL 2002, BANK N
2006 6141 LINC /TO LMODE
2007 0000 PNT30, 0000 /EXECUTE LIF N
2010 0006 DJR /DISABLE JUMP RETURN SAVE
2011 6000 LJMP 0 /JMP 0

```

2012	0601	LOCL,	LIF	1	/WE WILL ALWAYS BE IN UPPER MEM
2013	6014		LJMP	,+1	/BACK TO LOWER MEMORY
2014	0002		PDP		/BACK TO PMODE
2015	7010		RAR		/JUSTIFY
2016	3147		DCA	REGC	/SAVE FIRST RIF
2017	6201		CDF	0	/RESTORE DF
2020	1146		TAD	REGB	/GET TARGET
2021	7041		CIA		/2'S COMPLEMENT
2022	1147		TAD	REGC	/FETCH IF
2023	7650		SNA CLA		/DID WE LOAD THE IF?
2024	4516		JMS I	NERROR	/CHECK MONITOR
2025	4471		JMS I	ERROR	/DJR-JMP 0 FAILED TO LOAD IF
2026	7054		TST30M		/MESSAGE POINTER
2027	7402		HLT		/ERROR HALT
2030	7410		SKP		/TO NEXT TEST
2031	1763		TST30X		/ISZ LOOP; SCOPE LOOP
2032	5567		JMP I	TST30N	/NEXT BANK

```

/LMODE
/WILL ION-LIF INHIBIT INTERRUPTS?
/(TIMING RACE IF EP12-00018 IS NOT INSTALLED)
2033 7347 TST32, CLA CLL CMA /SET AC = 7777
2034 3065 DCA LPOINT /SET SWITCH
2035 4474 JMS I GETBNK /GET NEXT BANK
2036 7457 SNA /DONE?
2037 5272 JMP TST33 /YES, NEXT TEST
2040 3146 DCA REGB /SAVE BANK
2041 6041 TSF
2042 4556 JMS I SETFLG
2043 1146 TST32X, TAD REGB /FETCH IT
2044 7006 RTL /JUSTIFY FOR LIF
2045 7021 IAC CML /GET CURRENT IF, SET LINK
2046 1111 TAD KLIF /MAKE IT A LIF N
2047 3253 DCA ,+4 /STORE FOR EXECUTION
2050 6141 LINC /TO LMODE
2051 0500 IOB /
2052 6001 ION /ENABLE INTERRUPTS
2053 0000 0000 /EXECUTE LIF
2054 0016 LNOP /WAIT FOR INTERRUPT
2055 0016 LNOP /WAIT
2056 0500 IOB /
2057 6002 IOF /DISABLE INTERRUPTS
2060 0601 LIF 1
2061 0002 PDP /TO PMODE
2062 7437 SZL /ERROR?
2063 4516 JMS I NERROR /CHECK MONITOR
2064 4471 JMS I ERROR /LIF FAILED TO INHIBIT INTERRUPT
2065 7106 TST32M /MESSAGE POINTER
2066 7402 HLT /ERROR HALT
2067 741 SKP /TO NEXT TEST
2070 2043 TST32X /ISZ LOOP; SCOPE LOOP
2071 5233 JMP TST32 /NEXT BANK

```

```

/LMODE
/DOES LIF CAUSE THE IF/DF TO TRANSFER TO THE SF?
/
TST33:  CLA CLL           /CLEAR AC
        JMS I  GETBNK     /GET NEXT BANK
        SNA                /DONE?
        JMP I  TST34N     /YES, NEXT TEST
        DCA   REGB       /SAVE BANK
        TSF                /FLAG SET?
        JMS I  SETFLG     /NO, GO SET IT
        TAD   REGB       /GET BANK
        RTL                /JUSTIFY FOR LIF/LDF
        IAC                /GET CURRENT IF
        DCA   REGC       /SAVE IT
        TAD   REGC       /FETCH IT
        CMA                /COMPLEMENT
        AND   LMASK      /SAVE DF BITS
        DCA   REGE       /SAVE IT
        TAD   REGE       /FETCH IT
        TAD   KLDF       /MAKE IT A LDF -N
        DCA I  EXDF33     /STORE FOR EXECUTION
        TAD   REGC       /FETCH CONSTANT
        TAD   KLIF       /MAKE IT A LIF N
        DCA I  EXIF33     /STORE FOR EXECUTION
        LINC                /TO LMODE
XDF33:  0000             /EXECUTE LDF
XIF33:  0000             /EXECUTE LIF
        LUMP   ,+1       /TO UPPER MEMORY
        LIF   1           /RESTORE IF
        IOB                /
        RIB                /READ SF
        LDF   0           /RESTORE DF
        LUMP   ,+1       /BACK TO BANK 0
        ROL   2           /JUSTIFY
        PDP                /TO PMODE
        DCA   REGD       /SAVE RIB DATA
        TAD   REGD       /GET IF
        RTL                /JUSTIFY
        RTL                /
        RAL                /
        TAD   REGE       /GET OF
        CIA                /2'S COMPLEMENT
        TAD   REGD       /ADD RECEIVED
        SNA CLA          /EQUAL?
        JMS I  NERROR     /CHECK MONITOR
        JMS I  ERROR     /LIF FAILED TO LOAD SF
        TST33M           /MESSAGE POINTER
        HLT                /ERROR HALT
        SKP                /TO NEXT TEST
        TST33+5          /ISE LOOP; SCOPE LOOP
        JMP I  TST33N     /NEXT BANK

```

/LMODE
 /WILL RMF WORK IN EXTENDED MEMORY?
 /

2152	7320	TST34,	CLA	CLL	/CLEAR AC
2153	4474	JMS	I	GETBNK	/GET NEXT BANK
2154	7450	SNA			/DONE?
2155	5573	JMP	I	TST35N	/YES, NEXT TEST
2156	3146	DCA		REGB	/SAVE TARGET
2157	1146	TAD		REGB	/FETCH IT
2160	7006	RTL			/JUSTIFY FOR LIF
2161	7001	IAC			/INCREMENT FOR FIELD 2
2162	1111	TAD		KLIF	/MAKE IT A LIF N
2163	3366	DCA		,+3	/STORE FOR EXECUTION
2164	6002	IOF			/DISABLE INTERRUPTS
2165	6141	LINC			/TO LMODE
2166	0000				/EXECUTE LIF N
2167	6170	LJMP		,+1	/TO UPPER MEMORY
2170	0500	IOB			/
2171	6244	RMF			/RESTORE MEMORY
2172	0500	IOB			/
2173	6234	RIB			/FIND OUT WHERE WE ARE
2174	6175	LJMP		,+1	/TO LOWER MEMORY
2175	0002	PDP			/TO PMODE
2176	3550	DCA	I	REGCN	/SAVE TARGET - DATA FIELD IS ZERO
2177	6224	RIF			/NOW WHERE ARE WE?
2200	7640	SZA	CLA		/RMF FAILED IF NOT ZERO
2201	7040	CMA			/SET AC=7777 TO CAUSE ERROR
2202	6242	CIF		00	/JUST TO BE SURE
2203	5204	JMP		,+1	/BACK TO BANK 0
2204	7650	SNA	CLA		/AC=7777 IF ERROR
2205	4516	JMS	I	NERROR	/CHECK MONITOR
2206	4471	JMS	I	ERROR	/RMF FILED
2207	7201	TST34M			/MESSAGE POINTER
2210	7402	HLT			/ERROR HALT
2211	7410	SKP			/TO NEXT TEST
2212	2157	TST34+5			/ISZ LOOP; SCOPE LOOP
2213	5572	JMP	I	TST34N	/NEXT BANK

```

      2400      *2400
      /PMODE
      /AUTO INDEX TEST (FIRST SET UP REGISTERS)
      /
2400  7300  TST35:  CLA  CLL          /CLEAR AC
2401  4474      JMS  I   GETBNK    /GET NEXT BANK
2402  7450      SNA                    /DONE?
2403  5330      JMP          TST36X  /NEXT TEST
2404  3146      DCA  REGB    /SAVE IT
2405  1146      TAD  REGB    /FETCH IT
2406  7006      RTL                    /JUSTIFY
2407  7004      RAL                    /JUSTIFY
2408  1102      TAD  KCIF    /MAKE IT A CIF N
2409  3212      DCA  ,+1      /STORE FOR EXECUTION
2410  0000      0000        /EXECUTE CIF
2411  1146      TAD  REGB    /GET BANK
2412  7006      RTL                    /JUSTIFY
2413  7004      RAL                    /JUSTIFY
2414  1101      TAD  KCDF    /MAKE IT A CDF N
2415  3220      DCA  ,+1      /STORE FOR EXECUTION
2416  0000      0000        /EXECUTE CDF
2417  5222      JMP  ,+1      /TO UPPER MEMORY
2418  3000      DCA  0        /CLEAR 0
2419  7040      CMA                    /COMPLEMENT AC
2420  3734      DCA  I   END    /SET END (END=7777)
2421  7040      CMA                    /NOW SET AUTO 10-17 TO 7777
2422  3010      DCA  10
2423  7040      CMA
2424  3011      DCA  11
2425  7040      CMA
2426  3012      DCA  12
2427  7040      CMA
2428  3013      DCA  13
2429  7040      CMA
2430  3014      DCA  14
2431  7040      CMA
2432  3015      DCA  15
2433  7040      CMA
2434  3016      DCA  16
2435  7040      CMA
2436  3017      DCA  17

```

/
/NOW TEST REGISTERS
/

2445	1410	TAD I	10	/FETCH INDIRECT INDEXING TO 0
2446	7640	SZA CLA		/ZERO?
2447	5305	JMP	ERR10	/TO ERROR LOOP.
2450	1411	TAD I	11	
2451	7640	SZA CLA		
2452	5304	JMP	ERR11	
2453	1412	TAD I	12	
2454	7640	SZA CLA		
2455	5303	JMP	ERR12	
2456	1413	TAD I	13	
2457	7640	SZA CLA		
2460	5302	JMP	ERR13	
2461	1414	TAD I	14	
2462	7640	SZA CLA		
2463	5301	JMP	ERR14	
2464	1415	TAD I	15	
2465	7640	SZA CLA		
2466	5300	JMP	ERR15	
2467	1416	TAD I	16	
2470	7640	SZA CLA		
2471	5277	JMP	ERR16	
2472	1417	TAD I	17	
2473	7640	SZA CLA		
2474	5276	JMP	ERR17	
2475	5316	JMP	OK35	/THIS BANK IS OK


```

/
/NOW HANDLE THE RETURN
/
2476 7001 ERR17, IAC /INCREMENT AC TO FAILING CELL
2477 7001 ERR16, IAC
2500 7001 ERR15, IAC
2501 7001 ERR14, IAC
2502 7001 ERR13, IAC
2503 7001 ERR12, IAC
2504 7001 ERR11, IAC
2505 7001 ERR10, IAC
2506 6202 CIF 00 /BACK TO BANK 0
2507 5317 JMP ,+1 /TO LOWER MEMORY
2510 1023 TAD K0010 /ADD JUSTIFICATION
2511 3147 DCA REGC /SAVE FAILING CELL
2512 1547 TAD I REGC /GET CONTENTS
2513 3151 DCA REGD /SAVE IT
2514 6201 CDF 00 /RESTORE DATA FIELD
2515 5322 JMP FAL35 /TO ERROR MONITOR
2516 6201 0435, CDF 00 /RESTORE DATA FIELD
2517 6202 CIF 00 /RESTORE INSTRUCTION FIELD
2520 5321 JMP ,+1 /TO LOWER MEMORY
2521 4516 JMS I NERROR /CHECK MONITOR
2522 4471 FAL35, JMS I ERROR /AUTO INDEX FAILED
2523 7236 TST35M /MESSAGE POINTER
2524 7402 HLT /ERROR HALT
2525 7617 SKP CLA /TO NEXT BANK
2526 2405 TST35+5 /SCOPE LOOP
2527 5200 JMP TST35 /NEXT BANK
2530 7340 TST36X, CLA CLL CMA /SET AC=7777
2531 3145 DCA REGA /PRESET REGA
2532 5733 JMP I ,+1 /TO NEXT TEST
2533 2600 TST36 /
2534 7777 END, 7777 /
(DONE THIS WAY TO AVOID PAGING ERRORS)

```

```

2600 *2600
      /LMODE
      /AUTO INDEX TEST
      /
2601 7300 TST36, CLA CLL /CLEAR AC
2601 4474 JMS I GETBNK /FIND NEXT BANK
2602 7457 SNA /DONE?
2603 5211 JMP ,+6 /YES, RELOCATE
2604 3257 DCA ,+3 /SAVE BANK
2605 3146 DCA REGB /ZERO REGB
2606 4474 JMS I BNKSET /ZERO BANK
2607 1001 0001 /TARGET BANK TO BE SET TO ZERO
2610 5200 JMP TST36 /NEXT BANK
2611 4475 NBNK, JMS I GETBNL /GET NEXT LINC FIELD
2612 7550 SNA CLL /DONE?
2613 5236 JMP GOAUTO /YES, START TESTING
2614 3146 DCA REGB /SAVE IT
2615 1146 TAD REGR /FETCH IT
2616 7012 RTR /JUSTIFY FOR IF BITS 3 & 4 TO MAKE 0,1
2617 701 RAR /JUSTIFY
2620 7100 CLL /CLEAR LINK
2621 0112 AND KLJMP /CLEAR BITS 2-11
2622 1024 TAD K0017 /ADD 17; THIS WILL BE THE TARGET ADDRESS=1
2623 3233 DCA DEST36 /STORE
2624 1146 TAD REGB /GET BANK
2625 7012 RTR /JUSTIFY
2626 1037 AND K0077 /CLOSE ENOUGH
2627 3231 DCA ,+2 /STORE
2630 4553 JMS I RELOCR /GO RELOCATE PROGRAM
2631 1000 0000 /BANK
2632 4017 LAUTO-1 /ORG
2633 1000 DEST36, 0000 /DEST.
2634 1165 LEND-LAUTO /LENGTH
2635 5211 JMP NBNK /NEXT FIELD
2636 4475 GOAUTO, JMS I GETBNL /FIND TEST
2637 7457 SNA /DONE?
2640 5517 JMP I PASSN /GO TYPE PASS ALARM
2641 3146 DCA REGB /SAVE TARGET
2642 1146 TAD REGR /FETCH IT
2643 1111 TAD KLIF /MAKE IT A LIF N
2644 3254 DCA ,+10 /STORE FOR EXECUTION
2645 1146 TAD REGB /FETCH TARGET
2646 1104 TAD KLDF /MAKE IT A LDF N
2647 3255 DCA ,+6 /STORE FOR EXECUTION
2650 1025 TAD K0020 /GET 20
2651 1112 TAD KLJMP /MAKE A LJMP 20
2652 3256 DCA ,+4 /STORE FOR EXECUTION
2653 0141 EXAUT, LINC /TO LMODE
2654 1000 0000 /LIF N
2655 1000 0000 /LDF N
2656 1000 0000 /JMP 20

```

```

2657 0201 /
2660 4516 /TO HERE IN PMODE IF INDEX OK
2661 4471 /
2662 7272 LOK, CDF JMS I NERROR /CHECK MONITOR
2663 7422 ERL36, JMS I ERROR /AUTO-INDEX FAILED (DIRECT TO HERE FROM ERROR)
2664 7410 TST36M JMS I ERROR /MESSAGE POINTER
2665 2653 HLT /ERROR HALT
2666 5236 SKP EXAUT /TO NEXT FIELD
JMP GOAUTO /SCOPE LOOP
/NEXT FIELD,

```

```

4020      4020
          /LMODE (THIS PORTION IS RELOCATED FOR EACH FIELD)
          /AUTO INDEX TEST (IF=DF)
          /
4020 0061 LAUTO, SET 1 /SET UP REGISTERS
4021 3777          SET 3777
4022 0062          SET 2
4023 3777          SET 3777
4024 0063          SET 3
4025 3777          SET 3777
4026 0064          SET 4
4027 3777          SET 3777
4030 0065          SET 5
4031 3777          SET 3777
4032 0066          SET 6
4033 3777          SET 3777
4034 0067          SET 7
4035 3777          SET 3777
4036 0070          SET 10
4037 3777          SET 3777
4040 0071          SET 11
4041 3777          SET 3777
4042 0072          SET 12
4043 3777          SET 3777
4044 0073          SET 13
4045 3777          SET 3777
4046 0074          SET 14
4047 3777          SET 3777
4050 0075          SET 15
4051 3777          SET 3777
4052 0076          SET 16
4053 3777          SET 3777
4054 0077          SET 17
4055 3777          SET 3777
4056 0011          CLR
4057 4000          STC 0 /SET 0=0000
4060 1020          LDA 20 /PICK UP CONSTANT
4061 5252          STA 5252
4062 1040          STA /SET 1777=5252
4063 1777          STA 1777

```

/NOW TEST THE REGISTERS

4064	1011	CLR		
4065	1021	LDA	21	/GET INDIRECT INDEX 0
4066	0450	AZE		/ZERO?
4067	6172	LJMP	ERL1	/AUTO INDEX FAILED
4070	1022	LDA	22	
4071	0450	AZE		
4072	6164	LJMP	ERL2	
4073	1023	LDA	23	
4074	0450	AZE		
4075	6163	LJMP	ERL3	
4076	1024	LDA	24	
4077	0450	AZE		
4100	6162	LJMP	ERL4	
4101	1025	LDA	25	
4102	0450	AZE		
4103	6161	LJMP	ERL5	
4104	1026	LDA	26	
4105	0450	AZE		
4106	6160	LJMP	ERL6	
4107	1027	LDA	27	
4110	0450	AZE		
4111	6157	LJMP	ERL7	
4112	1030	LDA	30	
4113	0450	AZE		
4114	6156	LJMP	ERL10	
4115	1031	LDA	31	
4116	0450	AZE		
4117	6155	LJMP	ERL11	
4120	1032	LDA	32	
4121	0450	AZE		
4122	6154	LJMP	ERL12	
4123	1033	LDA	33	
4124	0450	AZE		
4125	6153	LJMP	ERL13	
4126	1034	LDA	34	
4127	0450	AZE		
4130	6152	LJMP	ERL14	
4131	1035	LDA	35	
4132	0450	AZE		
4133	6151	LJMP	ERL15	
4134	1036	LDA	36	
4135	0450	AZE		
4136	6150	LJMP	ERL16	
4137	1037	LDA	37	
4140	0450	AZE		
4141	6147	LJMP	ERL17	

4142	0002	PDP		/AUTO OK
4143	6201	ODF	02	
4144	6202	CIF	02	
4145	5746	JMP I	,+1	
4146	2657	LOK		/RETURN FOR NEXT BANK
4147	0221	ERL17,	XSK 1	/INCREMENT ERROR POINTER
4150	0221	ERL16,	XSK 1	/NONE OF THESE WILL SKIP.
4151	0221	ERL15,	XSK 1	
4152	0221	ERL14,	XSK 1	
4153	0221	ERL13,	XSK 1	
4154	0221	ERL12,	XSK 1	
4155	0221	ERL11,	XSK 1	
4156	0221	ERL10,	XSK 1	
4157	0221	ERL7,	XSK 1	
4160	0221	ERL6,	XSK 1	
4161	0221	ERL5,	XSK 1	
4162	0221	ERL4,	XSK 1	
4163	0221	ERL3,	XSK 1	
4164	0221	ERL2,	XSK 1	
4165	0221	XSK	1	
4166	0016	LNOP		/WASTE SOME SPACE FOR PAGING REASONS
4167	0016	LNOP		
4170	0016	LNOP		
4171	6174	LJMP	,+3	
4172	0061	ERL1,	SET 1	/MUST BE CELL 1 THAT FAILED
4173	0001	0001		
4174	0001	LDA		/FETCH IT
4175	0001	21		
4176	0002	PDP		/TO PMODE
4177	6201	ODF	02	/RESTORE DF
4220	3604	DCA I	PONT	/SAVE ERROR
4201	6202	CIF	02	/RESTORE IF
4202	5603	JMP I	,+1	/TO BANK 0
4203	2661	ERL36		/RETURN
4204	0147	PONT,	REGC	/ERROR POINTER
4225	0000	LEND,	0000	/END POINTER

```

/
/ALERT OPERATOR OF PASS COMPLETION (INHIBIT IF RSW 06=R1)
/
4226 7300 PASS, CLA CLL
4227 3145 DCA REGA /CLEAR REGA
4210 6201 CDF 00
4211 2017 ISZ COUNT /INCREMENT COUNT
4212 7000 NOP /DON'T SKIP
4213 7604 LAS /GET SWITCHES
4214 1026 AND K0040 /PICK OUT BIT 06
4215 7640 SZA CLA /SET ?
4216 5177 JMP 177 /YES, INHIBIT AND RESTART
4217 1137 TAD PNT0 /GET POINTER TO TEXT
4220 3471 DCA I ERROR /CHEAT MONITOR
4221 5623 JMP I PASPNT /GO TYPE MESSAGE
4222 7323 LOCO, TST37M /MESSAGE POINTER
4223 5051 PASPNT, ASCII /LINKUP POINTER

```

```

5000      *5000
/
/ NON ERROR MONITOR DETERMINES IF OPERATOR WANTS TO LOOP ON NON FAILING TEST
NERR0S, 0
5000      1000      /RETURN ADDRESS
5001      7307      CLA CLL IAC RTL      /SET AC = 4
5002      1200      TAD      NERR0S      /GET RETURN ADDRESS
5003      3200      DCA      NERR0S      /RETURN ADDRESS +4
5004      1600      TAD I      NERR0S      /GET SCOPE LOOP ADDRESS
5005      3220      DCA      ERR0RS      /STORE IT
5006      2145      ISZ      REGA      /UPDATE DATA
5007      5620      JMP I      ERR0RS      /LOOP BACK TO TEST
5010      7604      LAS      /READ SWITCHES
5011      0034      AND      K0400      /SAVE SR3
5012      7640      SZA CLA      /TEST AND CLEAR
5013      5620      JMP I      ERR0RS      / LOOPING
5014      7040      CMA      /SET AC=-1
5015      1200      TAD      NERR0S      /ADD NERR0RS
5016      3200      DCA      NERR0S      /STORE IN NERR0RS
5017      5600      JMP I      NERR0S      /JUMP INDIRECT LOOP

/
/ ERROR PROCESSOR, SCOPE LOOP, HALT, PRINT
ERR0RS, 0
5020      0000      /RETURN ADDRESS STORAGE
5021      7604      LAS      /READ SWITCHES
5022      7004      RAL      /MOVE SR1 INTO AC00
5023      7700      SMA CLA      /IS IT SET
5024      5250      JMP      ASCII      /NO TYPE A MESSAGE
5025      1220      ASCRXT, TAD      ERR0RS      /GET CURRENT ERROR ADDRESS
5026      7041      CIA      /INVERT IT
5027      3115      DCA      LSTERR      /STORE IN LAST ERROR
5030      2220      ISZ      ERR0RS      /YES INDEX ESCAPE
5031      7604      LAS      /READ SWITCHES
5032      7704      SMA CLA      /IS SR0 SET?
5033      7422      HLT      /NO, ERROR HALT
5034      2220      ISZ      ERR0RS      /YES INDEX ESCAPE TO JUMP OUT
5035      2220      ISZ      ERR0RS      /INDEX ERR0RS TO SCOPE MODE
5036      1620      TAD I      ERR0RS      /GET SCOPE ADDRESS
5037      3200      DCA      NERR0S      /STORE IN TYPE
5040      7604      LAS      /READ SWITCHES
5041      7006      RTL      /MOVE SR02 TO AC0
5042      771      SPA CLA      /IS SCOPE MODE SELECTED
5043      560      JMP I      NERR0S      /YES CONTINUE IN SCOPE LOOP
5044      704      CMA      /NO SET AC=7777
5045      122      TAD      ERR0RS      /SUBTRACT ONE FROM ERR0RS
5046      322      DCA      ERR0RS      /STORE SELECTED ADDRESS
5047      562      JMP I      ERR0RS      /EXIT TO NEXT TEST

```


5050	7240	ASCII,	CLA	CMA	/SET C(AC)=-1
5051	1627		TAD	I ERRORS	/GET MESSAGE ADDRESS STORAGE
5052	3014		DCA	PINT	/STORE IT IN AUTO INDEX REGISTER
5053	1227		TAD	ERRORS	/GET RETURN ADDRESS
5054	1115		TAD	LSTERR	/SUBTRACT LAST ERROR ADDRESS
5055	7657		SNA	CLA	/TEST
5056	5362		JMP	DATYP	/SAME GO TYPE DATA
5057	1414		TAD	I PINT	/GET FIRST CHARACTER
5060	3200		DCA	NERROS	/SAVE IT
5061	1200		TAD	NERROS	/GET IT
5062	7450		SNA		/TEST IT
5063	5225		JMP	ASCRXT	/NUMBER=EXIT
5064	7040		CMA		/INVERT IT
5065	7457		SNA		/NUMBER=EXITA
5066	5314		JMP	DATUM	/TYPE OUT DATA ROUTINE
5067	7040		CMA		/CHANGE IT BACK
5070	7112		RTR	CLL	/SWAP AC TO THE RIGHT
5071	7012		RTR		/MOVE
5072	7012		RTR		/MOVE
5073	4277		JMS	TYPECH	/TYPE IT
5074	1200		TAD	NERROS	/GET IT AGAIN
5075	4277		JMS	TYPECH	/TYPE IT
5076	5257		JMP	ASCII+7	/MUST BE MORE WORDS THAT NEED TYPING
5077	0000	TYPECH, 0			
5100	0037		AND	K0077	/SAVE SIGNIFICANT PART
5101	3157		DCA	SPACE	/STORE WORD
5102	1157		TAD	SPACE	/FETCH IT
5103	765		SNA	CLA	/TEST FOR 00 CRLF CODE
5104	4353		JMS	CRLF	/YES IT WAS
5105	1157		TAD	SPACE	/NO TYPE IT
5106	1377		TAD	M40	/SUBTRACT 40
5107	7510		SPA		/TEST POLARITY
5110	1031		TAD	K0100	/ADD 340
5111	1376		TAD	K240	/ADD 240
5112	4574		JMS	I TYPE	/TYPE
5113	5677		JMP	I TYPECH	/EXIT

5114	1414	DATUM,	TAD I	PINT	/GET ADDRESS OF REGISTER
5115	3200		DCA	NERROS	/STORE IN TEMP
5116	1200		TAD	NERROS	/GET TEMP
5117	7650		SNA CLA		/TEST FOR EXIT
5120	5225		JMP	ASCRXT	/EQUALS 0000 EXIT
5121	1200		TAD	NERROS	/GET TEMP
5122	1373		TAD	M4444	/ADD CONSTANT
5123	7650		SNA CLA		/TEST FOR RESTART
5124	4467		JMS I	BELL	/IT'S THERE; RESTART
5125	1600		TAD I	NERROS	/GET DATA
5126	4332		JMS	OCTYP	/TYPE IT
5127	1376		TAD	K240	/SPACE
5130	4574		JMS I	TYPE	/TYPE IT
5131	5314		JMP	DATUM	/TYPE NUMERIC DATA
5132	0000	OCTYP,	Ø		/RETURN ADDRESS STORAGE
5133	3277		DCA	TYPECH	/STORE DATA TO BE PRINTED
5134	1100		TAD	K7774	/SET UP TALLY
5135	3157		DCA	SPACE	/SET IT

5136	1035	HERE,	TAD	K1026	/GET FLAG NUMBER
5137	3353	REDO,	DCA	CRLF	/STORE
5140	1277		TAD	TYPECH	
5141	7004		RAL		
5142	3277		DCA	TYPECH	
5143	1353		TAD	CRLF	
5144	7004		RAL		
5145	7420		SNL		
5146	5337		JMP	REDO	
5147	4574		JMS I	TYPE	
5150	2157		ISZ	SPACE	
5151	5336		JMP	HERE	
5152	5732		JMP I	OCTYP	/EXIT
5153	0202	CRLF,	0		/RETURN ADDRESS STORAGE
5154	1374		TAD	K0215	/GET CR
5155	4574		JMS I	TYPE	/TYPE IT
5156	1375		TAD	K0212	/GET LF
5157	4574		JMS I	TYPE	/TYPE IT
5160	1032		TAD	K0177	/SET TO RUBOUT
5161	5753		JMP I	CRLF	/EXIT
5162	1414	DATYP,	TAD I	PINT	/GET A TERM OFF OF TYPE LIST
5163	7450		SNA		/END OF LIST?
5164	5225		JMP	ASCRXT	/YES EXIT
5165	7040		CMA		/INVERT
5166	7640		SZA CLA		/BEGINNING OF DATA
5167	5362		JMP	DATYP	/NO
5170	4353		JMS	CRLF	/YES OK RETURN THE TTY CARRIAGE AND LINE FEED
5171	7300		CLA CLL		/CLEAR AC AND LINK
5172	5314		JMP	DATUM	/GO TYPE THE DATA
5173	3334	M4444,	-4444		/SWITCH CHECK
5174	0215	K0215,	0215		
5175	0212	K0212,	0212		
5176	0240	K0240,	0240		
5177	7740	M40,	-40		

```

5200 5200 *5200
5201 1240 RANDY, 0 /NEW PAGE
5202 1240 TAD RNA /RANDOM NUMBER GENERATOR
5203 1241 TAD RNB
5204 1076 TAD K5252
5205 3243 DCA RND
5206 1243 TAD RND
5207 1242 TAD RNC
5208 3240 DCA RNA
5209 7004 RAL
5210 1240 TAD RNA
5211 1241 TAD RNB
5212 1076 TAD K5252
5213 3243 DCA RND
5214 1243 TAD RND
5215 1242 TAD RNC
5216 3241 DCA RNB
5217 7004 RAL
5218 1240 TAD RNA
5219 1076 TAD K5252
5220 3243 DCA RND
5221 1243 TAD RND
5222 1241 TAD RNB
5223 1242 TAD RNC
5224 3242 DCA RNC
5225 7004 RAL
5226 1240 TAD RNA
5227 3240 DCA RNA
5228 1241 TAD RNB
5229 1076 TAD K5252
5230 3243 DCA RND
5231 1243 TAD RND
5232 500 JMP I RANDY
5233 7601 RNA, 7601
5234 3542 RNB, 3542
5235 3755 RNC, 3755
5236 0016 RND, 0016
5237 5000 TYP0UT, 0
5238 6046 TLS /AC TO PRINTER
5239 6041 TSF /FLAG SET?
5240 5246 JMP , -1 /NOT UP; WAIT
5241 6042 TCF /NOW CLEAR IT
5242 7200 CLA /CLEAR AC
5243 5644 JMP I TYP0UT /INDIRECT RETURN
/
/ TELEPRINTER FLAG SET ROUTINE
/
5253 5000 FLAG, 0000
5254 7200 CLA /CLEAR AC
5255 6046 TLS /BUMP PRINTER
5256 6041 TSF /WAIT 100 MS
5257 5256 JMP , -1
5258 5653 JMP I FLAG /INDIRECT RETURN,

```

```

/
/PROGRAM RELOCATOR
/CALL: RELOC; BANK, ORG-1, DEST-1, END-ORG,
/
RELOC, 0000 /CONTAINS CALLING LOCATION +1
5261 0000 CLA CLL /CLEAR AC
5262 7300 TAD RELOC /GET BANK ADDRESS
5263 1261 CDF 00 /RESET DATA FIELD
5264 6201 DCA REGB /SAVE ADDRESS
5265 3146 TAD I REGB /BANK
5266 1546 DCA REGC /SAVE IT
5267 3147 ISZ REGB /INCREMENT
5270 2146 TAD I REGB /ORG-1
5271 1546 DCA AUTO11 /SAVE IT
5272 3015 ISZ REGB /INCREMENT
5273 2146 TAD I REGB /DEST-1
5274 1546 DCA AUTO12 /SAVE IT
5275 3016 ISZ REGB /INCREMENT
5276 2146 TAD I REGB /LENGTH
5277 1546 CMA /COMPLEMENT
5300 7040 DCA REGD /SAVE IT
5301 3151 ISZ REGB /INCREMENT
5302 2146 TAD REGB /GET RETURN
5303 1146 DCA RELOC /SAVE RETURN
5304 3261 TAD REGC /GET BANK
5305 1147 RTL /JUSTIFY
5306 7006 RAL /SOME MORE
5307 7004 AND PMASK /SAVE BITS 06-08
5310 121 TAD KCDF /GET CDF
5311 1101 DCA EXCREL /SAVE INSTRUCTION FOR EXECUTION
5312 3323 INCREL, ISZ REGD /CHECK IF DONE
5313 2151 JMP PICKUP /NOT DONE; MOVE A WORD
5314 5321 DCA REGB /RESET REGISTER
5315 3146 DCA REGC /RESET REGISTER
5316 3147 CDF 00 /RESET DATA FIELD
5317 6201 JMP I RELOC /RETURN
5320 5661 PICKUP, CDF 00 /RESET DATA FIELD
5321 6201 TAD I AUTO11 /GET WORD
5322 1415 EXCREL, 0000 /CHANGE DATA FIELD
5323 1000 DCA I AUTO12 /DEPOSIT WORD
5324 3416 JMP INCREL /CHECK BACK
5325 5313

```

```

/
/BANK SET
/CALL: LOCSET; BANK; REGB HAS CONSTANT
/
LOCSET, 0000
5326 0000
5327 7300      CLA CLL      /CLEAR AC
5330 1726      TAD I LOCSET  /GET BANK
5331 7006      RTL          /JUSTIFY
5332 7004      RAL          /SOME MORE
5333 0121      AND PMASK    /BITS 06-08
5334 1101      TAD KCDF     /ADD CDF
5335 3342      DCA EXCSET   /STORE FOR EXECUTION
5336 2326      ISZ LOCSET   /INCREMENT RETURN
5337 3147      DCA REGC     /ZERO REGC
5340 6201      PICSET, CDF 00 /RESET DATA FIELD
5341 1146      TAD REGB     /GET CONSTANT
5342 0000      EXCSET, 0000  /EXECUTE CDF
5343 3547      DCA I REGC   /DEPOSIT C(REGB) IN BANK (N)
5344 2147      ISZ REGC     /DONE?
5345 5340      JMP PICSET   /NO, NEXT WORD
5346 6201      CDF 00       /RESET DATA FIELD
5347 5726      JMP I LOCSET /RETURN

```

```

5400 *5400
      /PMODE-LMODE
      /INTERRUPT TEST: DO WE HAVE A SPURIOUS INTERRUPT ON-LINE?
      /
5400 INTTST, 0000
5401 7340 CLA CLL CMA
5402 3145 DCA REGA
5403 6041 TSF /FLAG SET?
5404 4556 JMS I SETFLG /NOT UP; GO SET IT
5405 6141 LINC /TO LMODE
5406 1020 LDA 20 /GET BIT 07
5407 0020 0020 /I/O PRESET
5410 0004 0004 /ESF
5411 0002 PDP /TO PMODE
5412 5213 JMP ,+1 /CLEAR INHIBIT
5413 5214 JMP ,+1 /CLEAR INHIBIT
5414 7300 CLA CLL /ZERO AC, LINK
5415 1140 TAD PNTP /GET POINTER
5416 3155 DCA RETURN /SET UP RETURN
5417 6001 ION /ENABLE INTERRUPTS
5420 7000 NOP /WAIT
5421 6002 IOF /DISABLE INTERRUPTS
5422 4516 JMS I NERROR /NO INTERRUPT ON-LINE
5423 4471 JMS I ERROR /SPURIOUS INTERRUPT?
5424 7340 INTSTM /MESSAGE POINTER
5425 7402 HLT /ERROR HALT
5426 7410 SKP /RETURN
5427 5401 INTTST+1 /ISZ LOOP; SCOPE LOOP
5430 5600 JMP I INTTST /RETURN

```

```

/
/PMODE FIND BANK
/
5431 1000 GETNXT, 0000
5432 7300 CLA CLL /CLEAR AC
5433 1066 TAD BANK /GET BANK
5434 7041 CIA /2'S COMPLEMENT
5435 1120 TAD PBANK /CHECK
5436 7650 SNA CLA /EQUAL?
5437 5243 JMP ,+4 /YES, RESET
5440 2120 ISZ PBANK /INCREMENT
5441 1120 TAD PBANK /FETCH IT
5442 5631 JMP I GETNXT /RETURN
5443 3120 DCA PBANK /CLEAR BANK
5444 5631 JMP I GETNXT

/
/LMODE FIND BANK
/
5445 0000 GETNXL, 0000
5446 7300 CLA CLL /CLEAR AC
5447 1066 TAD BANK /FETCH AVAILABLE BANK
5450 7006 RTL /JUSTIFY
5451 1021 TAD K0003 /INCREASE TO MAXIMUM
5452 7041 CIA /2'S COMPLEMENT
5453 1113 TAD LBANK /COMPARE
5454 7650 SNA CLA /EQUAL?
5455 5261 JMP ,+4 /YES, RESET
5456 2113 ISZ LBANK /INCREMENT
5457 1113 TAD LBANK /FETCH IT
5460 5645 JMP I GETNXL /RETURN
5461 1021 TAD K0003 /DON'T USE FIELDS 0-3
5462 3113 DCA LBANK /SAVE IT
5463 5645 JMP I GETNXL /RETURN

/
/RING THE BELL
/
5464 1000 BELLS, 0000
5465 7404 OSR /READ SWITCHES
5466 0031 AND K0100 /SAVE SR05
5467 764 SZA CLA /IS IT SET?
5470 5273 JMP ,+3 /YES, INHIBIT BELL
5471 1033 TAD K0207 /GET BELL
5472 4574 JMS I TYPE /GO RING IT
5473 5674 JMP I ,+1 /RETURN
5474 213 TST01 /AVOID CLOBBERING PASS COUNTER

```


/
/ERROR MESSAGES
/

5475	0024	TST01M, 0024
5476	2324	2324
5477	6061	6061
5500	0003	0003
5501	0406	0406
5502	4017	4017
5503	2240	2240
5504	2204	2204
5525	0640	0640
5526	0601	0601
5507	1114	1114
5510	0504	0504
5511	4050	4050
5512	2015	2015
5513	1704	1704
5514	0551	0551
5515	0023	0023
5516	0516	0516
5517	2440	2440
5520	2203	2203
5521	2604	2604
5522	4000	4000
5523	7777	EXITA
5524	0146	REGB
5525	0147	REGC
5526	0000	EXIT

/TST01

/CDF OR RDF FAILED (PMODE)
/SENT RCVD

5527	0024	TST02M, 0024
5530	2324	2324
5531	6062	6062
5532	0003	0003
5533	0406	0406
5534	4017	4017
5535	2240	2240
5536	2204	2204
5537	0640	0640
5540	0601	0601
5541	1114	1114
5542	0504	0504
5543	4050	4050
5544	2015	2015
5545	1704	1704
5546	0551	0551
5547	0023	0023
5550	0516	0516
5551	2440	2440
5552	2203	2203
5553	2604	2604
5554	4000	4000
5555	7777	EXITA
5556	0146	REGB
5557	0147	REGC

/TST02

/CDF OR RDF FAILED (PMODE)
/SENT RCVD

5560	0000	EXIT	
5561	0024	TST03M, 0024	/TST03
5562	2324	2324	
5563	6063	6063	
5564	0014	0014	/LDF OR RDF FAILED (LMODE)
5565	0406	0406	/SENT RCVD
5566	4017	4017	
5567	2240	2240	
5570	2204	2204	
5571	0640	0640	
5572	0601	0601	
5573	1114	1114	
5574	0504	0504	
5575	4050	4050	
5576	1415	1415	
5577	1704	1704	
5600	0551	0551	
5601	4000	4000	
5602	2305	2305	
5603	1624	1624	
5604	4022	4022	
5605	0326	0326	
5606	0400	0400	
5607	7777	EXITA	
5610	0146	REGB	
5611	0147	REGC	
5612	0000	EXIT	
5613	0024	TST04M, 0024	/TST04
5614	2324	2324	/LDF OR RDF FAILED (LMODE)
5615	6064	6064	/SENT RCVD
5616	0014	0014	
5617	0406	0406	
5620	4017	4017	
5621	2240	2240	
5622	2204	2204	
5623	0640	0640	
5624	0601	0601	
5625	1114	1114	
5626	0504	0504	
5627	4050	4050	
5630	1415	1415	
5631	1704	1704	
5632	0551	0551	
5633	0023	0023	
5634	0516	0516	
5635	2440	2440	
5636	2203	2203	
5637	2604	2604	
5640	4000	4000	
5641	7777	EXITA	
5642	0146	REGB	
5643	0147	REGC	
5644	0000	EXIT	

```

5645 0024 TST05M, 0024
5646 2324 2324
5647 6065 6065
5650 0003 0003
5651 0406 0406
5652 4017 4017
5653 2240 2240
5654 2204 2204
5655 0640 0640
5656 0601 0601
5657 1114 1114
5660 0504 0504
5661 4050 4050
5662 2015 2015
5663 1704 1704
5664 0551 0551
5665 0023 0023
5666 0516 0516
5667 2440 2440
5670 2203 2203
5671 2604 2604
5672 4000 4000
5673 7777 EXITA
5674 0146 REGB
5675 0147 REGC
5676 0000 EXIT

```

```

/TST05
/ODF OR RDF FAILED (PMODE)
/SENT RCVD

```

```

5677 0024 TST06M, 0024
5700 2324 2324
5701 6066 6066
5702 0014 0014
5703 0406 0406
5704 4017 4017
5705 2240 2240
5706 2204 2204
5707 0640 0640
5710 0601 0601
5711 1114 1114
5712 0504 0504
5713 4050 4050
5714 1415 1415
5715 1704 1704
5716 0551 0551
5717 0023 0023
5720 0516 0516
5721 2440 2440
5722 2203 2203
5723 2604 2604
5724 4000 4000
5725 7777 EXITA
5726 0146 REGB
5727 0147 REGC
5730 0000 EXIT

```

```

/TST06
/LDF OR RDF FAILED (LMODE)
/SENT RCVD

```

```

5731 0024 TST07M, 0024 /TST07
5732 2324 2324
5733 6067 6067
5734 0020 /PMODE INTERRUPT FAILED
5735 1517 1517
5736 0405 0405
5737 4011 4011
5740 1624 1624
5741 0522 0522
5742 2225 2225
5743 2024 2024
5744 4006 4006
5745 0111 0111
5746 1405 1405
5747 0400 0400
5750 7777 EXITA
5751 0000 EXIT

5752 0024 TST08M, 0024 /TST08
5753 2324 2324
5754 6070 6070
5755 0020 /PMODE LOAD SF OR RIB FAILED
5756 1517 1517 / DF SF
5757 0405 0405
5760 4014 4014
5761 1701 1701
5762 0440 0440
5763 2306 2306
5764 4017 4017
5765 2240 2240
5766 2211 2211
5767 0240 0240
5770 0601 0601
5771 1114 1114
5772 0504 0504
5773 0040 0040
5774 0406 0406
5775 4040 4040
5776 4023 4023
5777 0600 0600
6000 7777 EXITA
6001 146 REGB
6002 147 REGC
6003 0000 EXIT

6004 0024 TST9AM, 0024 /TST9A
6005 2324 2324
6006 7101 7101
6007 0314 /LMODE INTERRUPT FAILED
6010 1517 1517
6011 0405 0405
6012 4011 4011
6013 1624 1624
6014 0522 0522
2225 2225

```

6016 2024 2024
6017 4006 4006
6020 0111 0111
6021 1405 1405
6022 0400 0400
6023 7777 EXITA
6024 0000 EXIT

6025 0024 TST09M, 0024 /TST09

6026 2324 2324
6027 6071 6071
6030 0014 /LMODE LOAD SF OR RIB FAILED
6031 1517 / DF SF
6032 0405 0405
6033 4014 4014
6034 1701 1701
6035 0440 0440
6036 2306 2306
6037 4017 4017
6040 2240 2240
6041 2211 2211
6042 0240 0240
6043 0601 0601
6044 1114 1114
6045 0504 0504
6046 0040 0040
6047 0406 0406
6050 4040 4040
6051 4023 4023
6052 0600 0600
6053 7777 EXITA
6054 0146 REGB
6055 0147 REGC
6056 0000 EXIT

6057 0024 TST10M, 0024 /TST10

6060 2324 2324
6061 6160 6160
6062 0020 /PMODE DF FAILED TO ZERO ON AN INTERRUPT
6063 1517 /SENT SF RCVD
6064 0405 0405
6065 4004 4004
6066 0640 0640
6067 0601 0601
6070 1114 1114
6071 0504 0504
6072 4024 4024
6073 1740 1740
6074 3205 3205
6075 2217 2217
6076 4017 4017
6077 1640 1640
6100 0116 0116
6101 4011 4011
6102 1624 1624

6103	0522	0522
6104	2225	2225
6105	2024	2024
6106	0023	0023
6107	0516	0516
6110	2440	2440
6111	4023	4023
6112	0640	0640
6113	4022	4022
6114	0326	0326
6115	0400	0400
6116	7777	EXITA
6117	0146	REGB
6120	0147	REGC
6121	0151	REGD
6122	0000	EXIT

6123	0024	TST11M, 0024
6124	2324	2324
6125	6161	6161
6126	0014	0014
6127	1517	1517
6130	0405	0405
6131	4004	4004
6132	0640	0640
6133	0601	0601
6134	1114	1114
6135	0504	0504
6136	4024	4024
6137	1740	1740
6140	3205	3205
6141	2217	2217
6142	4017	4017
6143	1640	1640
6144	0116	0116
6145	4011	4011
6146	1624	1624
6147	0522	0522
6150	2225	2225
6151	2024	2024
6152	0023	0023
6153	0516	0516
6154	2440	2440
6155	4023	4023
6156	0640	0640
6157	4022	4022
6160	0326	0326
6161	0400	0400
6162	7777	EXITA
6163	0146	REGB
6164	0147	REGC
6165	0151	REGD
6166	0000	EXIT

/TST11

/LMODE DF FAILED TO ZERO ON AN INTERRUPT
/SENT SF RCVD

6167	0024	TST12M, 0024
------	------	--------------

/TST12

6170	2324	2324
6171	6162	6162
6172	0004	0004
6173	0301	0301
6174	4011	4011
6175	4055	4055
6176	4024	4024
6177	0104	0104
6200	4011	4011
6201	4006	4006
6202	0111	0111
6203	1405	1405
6204	0400	0400
6205	0611	0611
6206	1404	1404
6207	4014	4014
6210	1703	1703
6211	1640	1640
6212	2305	2305
6213	1624	1624
6214	4022	4022
6215	0326	0326
6216	0400	0400
6217	7777	EXITA
6220	0146	REGB
6221	0145	REGA
6222	0076	K5252
6223	0147	REGC
6224	0000	EXIT
6225	0024	TST13M, 0024
6226	2324	2324
6227	6163	6163
6230	0023	0023
6231	2401	2401
6232	4055	4055
6233	4014	4014
6234	0401	0401
6235	4006	4006
6236	0111	0111
6237	1405	1405
6240	0400	0400
6241	0201	0201
6242	1613	1613
6243	4014	4014
6244	1703	1703
6245	1640	1640
6246	2305	2305
6247	1624	1624
6250	4022	4022
6251	0326	0326
6252	0400	0400
6253	7777	EXITA
6254	0146	REGB
6255	0145	REGA

/DCA I - TAD I FAILED
/FIELD LOCN SENT RCVD

/TST13

/STA - LDA FAILED
/BANK LOCN SENT RCVD

6256 0076 K5252
6257 0147 REGC
6260 0001 EXIT

6261 0024 TST14M, 0024 /TST14

6262 2324 2324

6263 6164 6164

6264 0014 0014

/LMODE JUMP SAVE RETURN FAILED FOR NORMAL JUMP

6265 1517 1517

6266 0405 0405

6267 4012 4012

6270 2515 2515

6271 2040 2040

6272 2301 2301

6273 2605 2605

6274 4022 4022

6275 0524 0524

6276 2522 2522

6277 1640 1640

6300 0601 0601

6301 1114 1114

6302 0504 0504

6303 4006 4006

6304 1722 1722

6305 4016 4016

6306 1722 1722

6307 1501 1501

6310 1440 1440

6311 1225 1225

6312 1520 1520

6313 4000 4000

6314 7777 EXITA

6315 0000 EXIT

6316 0024 TST15M, 0024 /TST15

6317 2324 2324

6320 6165 6165

6321 0004 0004

/DJR FAILED TO INHIBIT JUMP SAVE

6322 1222 1222

6323 4006 4006

6324 0111 0111

6325 1405 1405

6326 0440 0440

6327 2417 2417

6330 4011 4011

6331 1610 1610

6332 1102 1102

6333 1124 1124

6334 4012 4012

6335 2515 2515

6336 2040 2040

6337 2301 2301

6340 2605 2605

6341 4000 4000

6342 7777 EXITA

6343	0000	EXIT
6344	0024	TST16M, 0024
6345	2324	2324
6346	6166	6166
6347	0014	0014
6350	1517	1517
6351	0405	0405
6352	4012	4012
6353	1520	1520
6354	4006	4006
6355	0111	0111
6356	1405	1405
6357	0440	0440
6360	2417	2417
6361	4003	4003
6362	1405	1405
6363	0122	0122
6364	4004	4004
6365	1222	1222
6366	4000	4000
6367	7777	EXITA
6370	0000	EXIT

/TST16

/LMODE JUMP FAILED TO CLEAR DJR

6371	0024	TST17M, 0024
6372	2324	2324
6373	6167	6167
6374	0020	0020
6375	1517	1517
6376	0405	0405
6377	4012	4012
6400	2515	2515
6401	2040	2040
6402	0114	0114
6403	2405	2405
6404	2205	2205
6405	0440	0440
6406	0305	0305
6407	1414	1414
6410	4060	4060
6411	6060	6060
6412	6000	6000
6413	7777	EXITA
6414	0000	EXIT

/TST17

/PMODE JUMP ALTERED CELL 0000

6415	0024	TST18M, 0024
6416	2324	2324
6417	6170	6170
6420	0020	0020
6421	1517	1517
6422	0405	0405
6423	4011	4011
6424	1706	1706
6425	4001	4001
6426	1424	1424

/TST18

/PMODE IOF ALTERED CELL 0000

6427	0522	0522
6430	0504	0504
6431	4003	4003
6432	0514	0514
6433	1440	1440
6434	6060	6060
6435	6060	6060
6436	4000	4000
6437	7777	EXITA
6440	0000	EXIT

6441	0024	TST19M, 0024
6442	2324	2324
6443	6171	6171
6444	0014	0014
6445	1517	1517
6446	0405	0405
6447	4011	4011
6450	1706	1706
6451	4001	4001
6452	1424	1424
6453	0522	0522
6454	0504	0504
6455	4003	4003
6456	0514	0514
6457	1440	1440
6460	6060	6060
6461	6060	6060
6462	4000	4000
6463	7777	EXITA
6464	0000	EXIT

/TST19

/LMODE IOF ALTERED CELL 0000

6465	0024	TST20M, 0024
6466	2324	2324
6467	6260	6260
6470	0020	0020
6471	1517	1517
6472	0405	0405
6473	4012	4012
6474	2515	2515
6475	2040	2040
6476	0314	0314
6477	0501	0501
6500	2205	2205
6501	0440	0440
6502	0412	0412
6503	2200	2200
6504	7777	EXITA
6505	0000	EXIT

/TST20

/PMODE JUMP CLEARED DJR

6506	0024	TST21M, 0024
6507	2324	2324
6510	6261	6261
6511	0024	0024
6512	1222	1222

/TST21

/DJR INHIBITED PMODE INTERRUPT SAVE

```

6513 4011      4011
6514 1610      1610
6515 1102      1102
6516 1124      1124
6517 0504      0504
6520 4020      4020
6521 1517      1517
6522 0405      0405
6523 4011      4011
6524 1624      1624
6525 0522      0522
6526 2225      2225
6527 2024      2024
6530 4023      4023
6531 0126      0126
6532 0500      0500
6533 7777      EXITA
6534 0000      EXIT
    
```

```

6535 0024      TST22M, 0024      /TST22
6536 2324      /NON-EXISTANT MEMORY READ-BACK FAILED
6537 6262      /BANK DATA
6540 0016      /REGD REGC
6541 1716
6542 5505
6543 3011
6544 2324
6545 0116
6546 2440
6547 1505
6550 1517
6551 2231
6552 4022
6553 0501
6554 0455
6555 0201
6556 0313
6557 4006
6560 0111
6561 1405
6562 0400
6563 0201
6564 1613
6565 4004
6566 0124
6567 0100
6570 7777      EXITA
6571 0151      REGD
6572 0147      REGC
6573 0000      EXIT
    
```

```

6574 0000      TST23M, 0000      /RESERVED
6575 0024      TST24M, 0024      /TST24
6576 2324
6577 6264
    
```

6600	0003	0003
6601	1106	1106
6602	4006	4006
6603	0111	0111
6604	1405	1405
6605	0440	0440
6606	2417	2417
6607	4040	4040
6610	1417	1417
6611	0104	0104
6612	4020	4020
6613	2217	2217
6614	2005	2005
6615	2240	2240
6616	1106	1106
6617	0023	0023
6620	0516	0516
6621	2440	2440
6622	2203	2203
6623	2604	2604
6624	4000	4000
6625	7777	EXITA
6626	0146	REGB
6627	0147	REGC
6630	0000	EXIT

/CIF FAILED TO LOAD PROPER IF
/SENT RCVD

6631	0024	TST25M, 0024
6632	2324	2324
6633	6265	6265
6634	0014	0014
6635	1106	1106
6636	4006	4006
6637	0111	0111
6640	1405	1405
6641	0440	0440
6642	2417	2417
6643	4014	4014
6644	1701	1701
6645	0440	0440
6646	2022	2022
6647	1720	1720
6650	0522	0522
6651	4011	4011
6652	0600	0600
6653	2305	2305
6654	1624	1624
6655	4022	4022
6656	0326	0326
6657	0400	0400
6660	7777	EXITA
6661	0146	REGB
6662	0147	REGC
6663	0000	EXIT

/TST25

/LIF FAILED TO LOAD PROPER IF
/SENT RCVD

6664	0024	TST26M, 0024
------	------	--------------

/TST26

6665	2324	2324
6666	6266	6266
6667	0003	0003
6670	1106	1106
6671	4006	4006
6672	0111	0111
6673	1405	1405
6674	0440	0440
6675	2417	2417
6676	4006	4006
6677	1116	1116
6700	0440	0440
6701	2022	2022
6702	1720	1720
6703	0522	0522
6704	4015	4015
6705	0515	0515
6706	1722	1722
6707	3100	3100
6710	2305	2305
6711	1624	1624
6712	4022	4022
6713	0326	0326
6714	0400	0400
6715	7777	EXITA
6716	1146	REGB
6717	1147	REGC
6720	0000	EXIT

/CIF FAILED TO FIND PROPER MEMORY
/SENT RCVD

6721	0024	TST27M, 0024
6722	2324	2324
6723	6267	6267
6724	0020	0020
6725	1517	1517
6726	1405	0405
6727	4011	4011
6730	1624	1624
6731	0522	0522
6732	2225	2225
6733	2024	2024
6734	2340	2340
6735	1617	1617
6736	2440	2440
6737	1116	1116
6740	1011	1011
6741	0211	0211
6742	2405	2405
6743	0440	0440
6744	0231	0231
6745	4003	4003
6746	1106	1106
6747	4000	4000
6750	0201	0201
6751	1613	1613
6752	4000	4000

/TST27
/PMODE INTERRUPTS NOT INHIBITED BY CIF
/BANK

6753	7777	EXITA
6754	1146	REG8
6755	0000	EXIT

6756	0024	TST28M, 0024	/TST28
6757	2324	2324	/LMODE LIF FAILED TO INHIBIT INTERRUPTS
6760	6270	6270	/BANK
6761	0014	0014	
6762	1517	1517	
6763	0405	0405	
6764	4014	4014	
6765	1106	1106	
6766	4006	4006	
6767	0111	0111	
6770	1405	1405	
6771	0440	0440	
6772	2417	2417	
6773	4011	4011	
6774	1610	1610	
6775	1102	1102	
6776	1124	1124	
6777	4011	4011	
7000	1624	1624	
7001	0522	0522	
7002	2225	2225	
7003	2024	2024	
7004	2300	2300	
7005	0201	0201	
7006	1613	1613	
7007	4000	4000	
7010	7777	EXITA	
7011	1146	REG8	
7012	0000	EXIT	

7013	0024	TST29M, 0024	/TST29
7014	2324	2324	/LMODE JMP 0 FAILED TO CLEAR
7015	6271	6271	/INTERRUPT INHIBIT
7016	0014	0014	
7017	1517	1517	
7020	0405	0405	
7021	4012	4012	
7022	1520	1520	
7023	4060	4060	
7024	4006	4006	
7025	0111	0111	
7026	1405	1405	
7027	0440	0440	
7030	2417	2417	
7031	4003	4003	
7032	1405	1405	
7033	0122	0122	
7034	4011	4011	
7035	1624	1624	
7036	0522	0522	
	2225	2225	

```

7040 2024      2024
7041 4011      4011
7042 1610      1610
7043 1102      1102
7044 1124      1124
7045 4000      4000
7046 0201      0201
7047 1613      1613
7050 4000      4000
7051 7777      EXITA
7052 0146      REGB
7053 0000      EXIT

```

```

7054 0024      TST30M, 0024
7055 2324      2324
7056 6360      6360
7057 0014      0014
7060 1517      1517
7061 1405      0405
7062 4004      4004
7063 1222      1222
7064 5512      5512
7065 1520      1520
7066 4060      4060
7067 4006      4006
7070 0111      0111
7071 1405      1405
7072 0440      0440
7073 2417      2417
7074 4014      4014
7075 1701      1701
7076 0440      0440
7077 1106      1106
7080 0002      0002
7081 0116      0116
7082 1300      1300
7083 7777      EXITA
7084 0146      REGB
7085 0000      EXIT

```

```

/TST30
/LMODE DJR-JMP 0 FAILED
/TO LOAD IF
/BANK

```

```

7106 0024      TST32M, 0024
7107 2324      2324
7110 6362      6362
7111 0014      0014
7112 1517      1517
7113 1405      0405
7114 4011      4011
7115 1716      1716
7116 5514      5514
7117 1106      1106
7120 4006      4006
7121 0111      0111
7122 1405      1405
7123 0440      0440

```

```

/TST32
/LMODE ION-LIF FAILED TO
/INHIBIT INTERRUPTS

```

7124	2417	2417
7125	4011	4011
7126	1610	1610
7127	1102	1102
7130	1124	1124
7131	4011	4011
7132	1624	1624
7133	0522	0522
7134	2225	2225
7135	2024	2024
7136	2300	2300
7137	0000	EXIT

7140	0024	TST33M, 0024
7141	2324	2324
7142	6363	6363
7143	0014	0014
7144	1517	1517
7145	0405	0405
7146	4014	4014
7147	1106	1106
7150	5512	5512
7151	1520	1520
7152	4016	4016
7153	4006	4006
7154	0111	0111
7155	1405	1405
7156	0440	0440
7157	2417	2417
7160	4014	4014
7161	1701	1701
7162	0440	0440
7163	2306	2306
7164	0040	0040
7165	1106	1106
7166	4040	4040
7167	4004	4004
7170	0640	0640
7171	4040	4040
7172	2306	2306
7173	4000	4000
7174	7777	EXITA
7175	1147	REGC
7176	1152	REGD
7177	1151	REGD
7200	0000	EXIT

```

/TST33
/LMODE LIF-JMP N FAILED TO LOAD SF
/ IF DF SF

```

7201	0024	TST34M, 0024
7202	2324	2324
7203	6364	6364
7204	0014	0014
7205	1517	1517
7206	0405	0405
7207	4022	4022
	1506	1506

```

/TST34
/LMODE RMF IN EXTENDED
/BANK FAILED

```


7211	4011	4011
7212	1640	1640
7213	0530	0530
7214	2405	2405
7215	1604	1604
7216	0504	0504
7217	4002	4002
7220	0116	0116
7221	1340	1340
7222	0601	0601
7223	1114	1114
7224	0504	0504
7225	0002	0002
7226	0116	0116
7227	1340	1340
7230	4023	4023
7231	0600	0600
7232	7777	EXITA
7233	0146	REGB
7234	0147	REGC
7235	0000	EXIT

7236	0024	TST35M, 0024
7237	2324	2324
7240	6365	6365
7241	0020	0020
7242	1517	1517
7243	0405	0405
7244	4001	4001
7245	2524	2524
7246	1755	1755
7247	1116	1116
7250	0405	0405
7251	3040	3040
7252	0601	0601
7253	1114	1114
7254	0504	0504
7255	0002	0002
7256	0116	0116
7257	1340	1340
7260	0305	0305
7261	1414	1414
7262	4001	4001
7263	0404	0404
7264	2200	2200
7265	7777	EXITA
7266	0146	REGB
7267	0147	REGC
7270	0151	REGD
7271	0000	EXIT

/TST35
 /PMODE AUTO-INDEX FAILED
 /BANK CELL ADDR

7272	0024	TST36M, 0024
7273	2324	2324
7274	6366	6366
7275	4014	4014

/TST36
 /LMODE AUTO-INDEX FAILED
 /FIELD LOCN

7276	1517	1517
7277	0405	0405
7300	4001	4001
7301	2524	2524
7302	1755	1755
7303	1116	1116
7304	0405	0405
7305	3040	3040
7306	0601	0601
7307	1114	1114
7310	0504	0504
7311	0006	0006
7312	1105	1105
7313	1404	1404
7314	4014	4014
7315	1703	1703
7316	1600	1600
7317	7777	EXITA
7320	0146	REGB
7321	0147	REGC
7322	0000	EXIT
7323	0005	TST37M, 0005
7324	3024	3024
7325	4015	4015
7326	0515	0515
7327	4024	4024
7330	2324	2324
7331	4020	4020
7332	0123	0123
7333	2355	2355
7334	5555	5555
7335	7777	EXITA
7336	0017	COUNT
7337	4444	EXITB
7340	0023	INTSTM, 0023
7341	2025	2025
7342	2211	2211
7343	1725	1725
7344	2340	2340
7345	1116	1116
7346	2405	2405
7347	2222	2222
7350	2520	2520
7351	2441	2441
7352	0050	0050
7353	0310	0310
7354	0503	0503
7355	1340	1340
7356	1117	1117
7357	0340	0340
7360	1157	1157
7361	1740	1740
7362	2022	2022
7363	0523	0523

/EXT MEM TST PASS--(PASS)

/SPECIAL RESTART: EVENTUALLY GETS TO TST01

/SPURIOUS INTERRUPT!
/(CHECK IOC I/O PRESET)

7364	0524	0524
7365	5100	5100
7366	0000	EXIT

\$

2000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2300 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2500 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2700 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

1000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1300 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1500 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1700 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

2000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2200 11111111 11110000 00000000 00000000 00000000 00000000 00000000 00000000
2300 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
2400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2500 11111111 11111111 11111111 11111000 00000000 00000000 00000000 00000000
2600 11111111 11111111 11111111 11111111 11111111 11111111 11111110 00000000
2700 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

3000
3100

3200
3300

3400
3500

3600
3700

4000	00000000	00000000	11111111	11111111	11111111	11111111	11111111	11111111	11111111
4100	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
4200	11111111	11111111	11110000	00000000	00000000	00000000	00000000	00000000	00000000
4300	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

4400
4500

4600
4700

5000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
5100	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
5200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
5300	11111111	11111111	11111111	11111111	11111111	00000000	00000000	00000000	00000000

5400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
5500	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
5600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
5700	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111

6000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
6100	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111

6200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
6300	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111

6400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
6500	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111

6600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
6700	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111

7000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
7100	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111

7200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
7300	11111111	11111111	11111111	11111111	11111111	11111111	11111110	00000000	00000000

7400
7500

7600
7700

ADD	2000	EXCSET	5342	LDA	1000	PMASK	0121
ASCII	5050	EXDF33	0072	LDF	0640	PNT30	2007
ASCRXT	5025	EXIF33	0073	LEND	4205	PNT30N	0141
AUTO11	0015	EXIT	0000	LHAN	0042	PNTA	0122
AUTO12	0016	EXITA	7777	LIF	0600	PNTB	0123
AZE	0450	EXITB	4444	LINC	6141	PNTC	0124
BAK22	1300	FAL22	1310	LINTR	0040	PNTCA	0125
BANK	0066	FAL35	2522	LJMP	6000	PNTD	0126
BCL	1540	FLAG	5253	LMASK	0114	PNTE	0127
BELL	0067	GETBNK	0074	LNOP	0016	PNTF	0130
BELLS	5464	GETBNL	0075	LOCA	0502	PNTG	0131
BNKSET	0070	GETNXL	5445	LOCB	0527	PNTH	0132
BSE	1600	GETNXT	5431	LOCC	0620	PNTI	0133
CLR	0011	GOAUTO	2636	LOCCA	0564	PNTJ	0134
COUNT	0017	HERE	5136	LOCD	0660	PNTK	0135
CRLF	5153	INCREL	5313	LOCE	0722	PNTL	0136
DATUM	5114	INTSTM	7340	LOCF	1246	PNTO	0137
DATYP	5162	INTTST	5400	LOCG	1430	PNTP	0140
DEST36	2633	IOB	0500	LOCH	1473	PONT	4204
DJR	0026	K0003	0021	LOCI	1605	PPOINT	0142
END	2534	K0007	0022	LOCJ	1651	PREG	0143
ERL1	4172	K0010	0023	LOCK	1734	RANDOM	0144
ERL10	4156	K0017	0024	LOCL	2012	RANDY	5200
ERL11	4155	K0020	0025	LOCO	4222	READ0	1316
ERL12	4154	K0040	0026	LOCP	5423	READ1	1342
ERL13	4153	K0070	0027	LOCSET	5326	REDO	5137
ERL14	4152	K0077	0030	LOK	2657	REGA	0145
ERL15	4151	K0100	0031	LPOINT	0065	REGB	0146
ERL16	4150	K0177	0032	LREG	0064	REGC	0147
ERL17	4147	K0207	0033	LREG1	0013	REGCN	0150
ERL2	4164	K0212	5175	LSET	0050	REGD	0151
ERL3	4163	K0215	5174	LSKP	0456	REGE	0152
ERL36	2661	K0400	0034	LSTERR	0115	RELOC	5261
ERL4	4162	K1026	0035	M40	5177	RELOCR	0153
ERL5	4161	K1777	0036	M4444	5173	RELNT	0154
ERL6	4160	K2000	0037	MSTART	0177	RETURN	0155
ERL7	4157	K240	5176	NBNK	2611	RNA	5240
ERR10	2505	K5252	0076	NERROR	0116	RNB	5241
ERR11	2504	K6020	0077	NERRCS	5000	RNC	5242
ERR12	2503	K7774	0100	NOW1	0374	RND	5243
ERR13	2502	KCOF	0101	NOW2	0452	ROL	0240
ERR14	2501	KCIF	0102	OCTYP	5132	ROR	0300
ERR15	2500	KHLT	0103	OK35	2516	SET	0060
ERR16	2477	KIOB	0105	PASPNT	4223	SETFLG	0156
ERR17	2476	KIUF	0106	PASS	4206	SPACE	0157
ERROR	0071	KLOF	0104	PASSN	0117	SHO	1500
ERRORS	5020	KLIF	0111	PBANK	0120	STA	1040
EX29	1725	KLJMP	0112	PDF	0002	START	0200
EXAUT	2653	KLNOP	0107	PICKUP	5321	STC	4000
EXC12	0763	KRIF	0110	PICSET	5340	TST01	0213
EXC13	0222	LAUTO	4222	PINT	0014	TST01M	5475
EXCREL	5323	LBANK	0113	PINTR	0000	TST02	0240

TST02M	5527	TST25M	6631
TST03	0265	TST26	1514
TST03M	5561	TST26M	6664
TST04	0315	TST27	1556
TST04M	5613	TST27M	6721
TST05	0345	TST27N	0165
TST05M	5645	TST28	1613
TST06	0413	TST28M	6756
TST06M	5677	TST28N	0166
TST07	0473	TST29	1663
TST07M	5731	TST29M	7013
TST08	0512	TST30	1747
TST08M	5752	TST30M	7054
TST09	0576	TST30N	0167
TST09M	6025	TST30X	1763
TST10	0643	TST32	2033
TST10M	6057	TST32M	7106
TST11	0701	TST32N	0170
TST11M	6123	TST32X	2043
TST12	0746	TST33	2072
TST12A	0755	TST33M	7140
TST12M	6167	TST33N	0171
TST12N	0161	TST34	2152
TST13	1004	TST34M	7201
TST13A	1016	TST34N	0172
TST13M	6225	TST35	2400
TST13N	0162	TST35M	7236
TST14	1044	TST35N	0173
TST14M	6261	TST36	2600
TST15	1064	TST36M	7272
TST15M	6316	TST36X	2530
TST16	1105	TST37M	7323
TST16M	6344	TST9A	0550
TST17	1127	TST9AM	6004
TST17M	6371	TSTINT	0160
TST18	1145	TYPE	0174
TST18M	6415	TYPECH	5077
TST19	1163	TYPOUT	5244
TST19M	6441	XDF33	2120
TST20	1204	XIF33	2121
TST20M	6465	XSK	0220
TST21	1231		
TST21M	6506		
TST22	1262		
TST22M	6535		
TST23	137		
TST23M	6574		
TST23N	0163		
TST24	1403		
TST24M	6575		
TST24N	0164		
TST25	1446		

ERRORS DETECTED: 0

LINKS GENERATED: 0

RUN-TIME: 25 SECONDS

3K CORE USED