

IDENTIFICATION

PRODUCT CODE: MAINDEC-12-D1AC-D  
PRODUCT NAME: EXTENDED MEMORY CONTROL  
(EXTMC12)  
DATE CREATED: JUNE 19, 1970  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: HAROLD LONG

RSW = 7001 in BK

S 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.

8K: 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  
LMODE 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 DMF IN EXTENDED BANK FAILED  
BANK GF

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 POP-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 0KNK10 OCTAL

/PROGRAM CONTROL IS HANDLED BY A MONITOR RESIDENT IN BANK 0, TIMES  
/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 P-MODE, START 22

/PDP-12 INSTRUCTION DEFINITIONS

164	LDF=0640	/LOAD DATA FIELD 0-37
190	LIF=0604	/LOAD INSTRUCTION FIELD 0-37
2006	DJR=0006	/DISABLE JUMP RETURN
	/MODE CHANGE	
1002	POP=0002	/SWITCH TO P MODE
6141	LINC=6141	/SWITCH TO L MODE
	/L MODE PROGRAMMING INSTRUCTIONS	
6200	LJMP=6000	/JMP
1311	CLR=0011	
1450	AZE=0450	
2000	ADD=2000	
1500	IOB=0500	
10016	LNOP=0016	/NOP
1300	ROR=0300	
1456	LSKP=0456	
1240	ROL=0240	
1600	BSE=1600	
1540	BCL=1540	
1360	SET=0360	/(REALLY SET I)
4000	STC=4000	
1500	SRO=1500	/USED AS A SWITCH CHECK
1000	LDA=1000	
1240	STA=1040	
1220	XSK=0220	/(REALLY XSK I)
	/DATA MATRIX SWITCHES	
7777	EXITA=7777	
4444	EXITB=4444	/SPECIAL RESTART SWITCH
1000	EXIT=0000	

```

/ P MODE INTERRUPT HANDLER
* 020
PINTR, 0220 /INTERRUPT RETURN STORAGE (ALSO LING JUMP SAVE)
CLA CLL CML /SET LINK, CLEAR AC
RIS /READ SF
DCA /SAVE IT
TAD PPOINT /GET SWITCH
SNA CLA /SET?
JMP I RETURN /NO, RETURN THROUGH PRESET LINKUP
DCA /CLEAR SWITCH
RNF /RESTORE MEMORY
ISZ R /ENABLE RETURN
JMP I PINTR /BACK TO MAINLINE VIA INTERRUPT RETURN LINKUP
/AUTO-INDEX REGISTERS
LREG1, 0020 /DATA POINTER
PINT, 0220 /MESSAGE POINTER
AUTO11, 0020
AUTO12, 0220
COUNT, 0220
/ CROSS-PAGE REFERENCE TAGS AND CONSTANTS
* 020
0020 JNP 176 /MINOR START
0021 K 023, 0023
0022 K 027, 0227
0023 K 010, 0212
0024 K 017, 0217
0025 K 023, 0020
0026 K 040, 0240
0027 K 070, 0270
0030 K 077, 0277
0031 K 100, 2100
0032 K 177, 0177
0033 K 207, 0207
0034 K 400, 2400
0035 K 026, 1026
0036 K 177, 1777
0037 K 200, 2777

```

/LMODE INTERRUPT HANDLER

```

0240      0240
0241      0241
0242      0242
0243      0243
0244      0244
0245      0245
0246      0246
0247      0247
0250      0250
0251      0251
0252      0252
0253      0253
0254      0254
0255      0255
0256      0256
0257      0257
0260      0260
0261      0261
0262      0262
0263      0263
0264      0264
0265      0265

LINTR, 0000
LHAN,   CLR
        IOB
        RIR
        STC
        SRC
        LREG
        LPOINT
LSET,   LSKP
        LJMP
        IOB
        RMF
        XSK
        ADD
        BSE
        6000
        STC
        STC
        IOB
        IOB
        LJMP
        0000
        LPOINT, 0000

/INTERRUPT RETURN STORAGE
/CLEAR LINK, CLEAR AC
/
/READ SAVE FIELD REG
/SAVE IT
/SWITCH SET?
/
/TO HERE IF BIT 0=1
/NO, RETURN TO BANK 0 THROUGH PRESET LINKUP
/
/YES, RESTORE MEMORY FIELDS
/INCREMENT
/GET RETURN
/MAKE IT A LINC JUMP (BSE I)
/
/STORE FOR EXECUTION
/CLEAR SWITCH
/
/ENABLE INTERRUPTS
/BACK TO BANK 0 VIA INTERRUPT RETURN LINKUP

```



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	REGCV,	REGC	
0151	0000	REGD,	0000	/DATA
0152	0000	REGE,	0000	/DATA
0153	5201	RELOCR,	RELOC	/CROSS PAGE TO RELOCATOR SUBR
0154	0000	RELPT,	PIVTR	/CROSS BANK TO INTERRUPT RETURN STORAGE
0155	0000	RETURN,	0000	/PMODE INTERRUPT RETURN IF SWITCH=?
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

```

176
0176 7412 SKP /DON'T RING ON STARTUP, INITIALIZE TEST
0177 4467 /GO RING BELL, RETURN TO TST01

/TO HERE FROM MINOR START
/
*176
/START, JMS I BELL
/MAJOR START P MODE: INITIALIZATION ROUTINE
/
*200
START, LAS /READ SWITCHES
AVO K0007 /SAVE BANK BITS
DCA BANK /AMOUNT OF EXTENDED MEMORY
CCA REGA /CLEAR LOOP COUNTER
DCA LSTERR /CLEAR OLD ERROR
CCA PBANK /CLEAR PASS COUNTER
DCA COUNT /SET LBANK TO UPPER MEMORY
TAD K2003 /RESTORE DATA FIELD
CCA LBANK /TEST FOR NO INTERRUPT
COF 30
JMS I TSTINT
/PMODE
/CAN THE DATA FIELD REGISTER BE LOADED WITH ALL NUMBERS (BINARY COUNT)
/
TST01, TAD REGA /FETCH TEST NUMBER
AVO PMASK /SAVE BITS 06-08
DCA REGB /SAVE FOR OBSERVATION
TAD REGB /FETCH IT
TAD KCDF /ADD CDF
DCA *+1 /PLACE IT IN ROUTINE
AVO PMASK /EXECUTE CDF N
DCA REGC /GET DATA FIELD
TAD REGC /SAVE BITS 06-08
DCA REGC /SAVE FOR TYPING
DCA REGC /FETCH IT
DCA CLA /2'S COMPLEMENT
TAD CLA /COMPARE WITH DATA SENT
DCA CLA /RESTORE DATA FIELD
JMS I ERROR /INCORRECT IF NOT ZERO
JMS I ERROR /CHECK WITH MONITOR
TST01M /CDF OR RDF FAILED
HLT /MESSAGE POINTER
SKIP CLA /ERROR HALT
TST01 /GO TO NEXT TEST
TST01 /SCOPE LOOP, ISX LOOP

```

```

/PMODE
/CAN THE DATA FIELD BE LOADED WITH RANDOM NUMBERS
/
TST02, JMS I RANDOM /GET A RANDOM NUMBER
AND AND PMASK /SAVE BITS 06-08
DCA DCA REGB /SAVE FOR OBSERVATION
TAD TAD REGR /FETCH IT
TAD TAD KCDF /ADD CDF
DCA DCA *+1 /PLACE IT IN ROUTINE
0000 0000 /EXECUTE CDF R
RDF RDF /GET DATA FIELD
AND AND PMASK /SAVE BITS 06-08
DCA DCA REGC /SAVE FOR TYPING
TAD TAD REGC /FETCH IT
DCA DCA /2'S COMPLEMENT
TAD TAD REGB /COMPARE
CDF CDF 00 /RESTORE DATA FIELD
SNA CLA /INCORRECT IF NOT ZERO
JMS I MERROR /CHECK WITH MONITOR
JMS I ERROR /CDF FAILED
TST02M TST02M /MESSAGE POINTER
HLT HLT /ERROR HLT
SKP CLA /NEXT TEST
TST02 /SCOPE LOOP; ISZ LOOP
0264 0264

```

```

/LMODE
/CAN THE DATA FIELD REGISTER BE LOADED WITH BINARY COUNT
/
TST03, TAD REGA /FETCH TEST NUMBER
AND AND LMASK /SAVE BITS 07-11
DCA DCA REGB /SAVE FOR OBSERVATION
TAD TAD REGR /FETCH IT
TAD TAD KCDF /ADD LDF
DCA DCA *+2 /PLACE IN ROUTINE
LINC LINC /GO TO LINC MODE
0000 0000 /EXECUTE LDF
ICR ICR /PREPARE TO GET DATA FIELD
PDF PDF /GET DATA FIELD
PDP PDP /BACK TO PROMPT
RAR CLL /JUSTIFY RIGHT TO AGREE WITH REGR
DCA DCA REGC /SAVE FOR TYPING
TAD TAD REGC /FETCH IT
DCA DCA /2'S COMPLEMENT
CDF CDF 00 /COMPARE
SNA CLA /RESTORE DATA FIELD
JMS I MERROR /INCORRECT IF NOT ZERO
JMS I ERROR /CHECK WITH MONITOR
TST03M TST03M /CDF FAILED
HLT HLT /MESSAGE POINTER
SKP CLA /ERROR HLT
TST03 /GO TO NEXT TEST
0265 0265 /SCOPE LOOP; ISZ LOOP

```



```

0315 4544
0316 114
0317 3146
0320 1146
0321 1144
0322 3324
0323 6141
0324 0000
0325 0520
0326 6214
0327 0002
0330 7110
0331 3147
0332 1147
0333 7041
0334 1146
0335 6221
0336 765
0337 4516
0340 4471
0341 5613
0342 7402
0343 7610
0344 0315

/LSMODE
/CAN THE DATA FIELD REGISTER BE LOADED WITH RANDOM NUMBERS
/
TST04, JMS I RANDCY
AND LMASK
DCA REGR
TAD REGB
TAD KLDF
DCA ,+2
LINC
0000
IOR
RDF
PDP
RAR CLL
DCA REGC
TAD REGC
CIA
TAD REGB
CDF 0?
SNA CLA
JMS I NERROR
JMS I ERROR
TST04M
HLT
SKP CLA
TST04

```

```

/GET RANDOM NUMBER
/SAVE BITS 07-11
/SAVE FOR OBSERVATION
/FEICH IT
/ADD LF
/PLACE IN ROUTINE
/GO TO LINC MODE
/EXECUTE LDF
/PREPARE TO GET DATA FIELD
/GET DATA FIELD
/BACK TO PMODE
/JUSTIFY RIGHT TO AGREE WITH REGB
/SAVE FOR TYPING
/FEICH IT
/2'S COMPLEMENT
/COMPARE
/RESTORE DATA FIELD
/INCORRECT IF NOT ZERO
/CHECK WITH MONITOR
/LDF FAILED
/MESSAGE POINTER
/ERROR HALT
/GO TO NEXT TEST
/SCOPE LOOP; ISZ LOOP

```

```

/PMODE
/GATE SHAKER TEST
/
TST05, JMS I RANDOM
AND PMASK
DCA REG8
TAD REG8
TAD KCOF
DCA NOW1
CDF W0
CDF 40
CDF 20
CDF 10
CDF 70
CDF 60
CDF 50
CDF 40
CDF 30
CDF 20
CDF 12
CDF 20
CDF 30
CDF 40
CDF 50
CDF 60
CDF 70
NOW1,
CDF 000
RDF
AND
DCA
TAD
CIA
TAD
CDF
SVA CLA
JMS I NERROR
JMS I ERROR
TST05M
HLT
SKP CLA
TST05
4345 4544
4346 1121
4347 3146
4350 1146
4351 1101
4352 3374
4353 6201
4354 6241
4355 6221
4356 6211
4357 6271
4360 6261
4361 6251
4362 6241
4363 6231
4364 6221
4365 6211
4366 6221
4367 6231
4370 6241
4371 6251
4372 6261
4373 6271
4374 000
4375 6214
4376 6121
4377 3147
4400 1147
4401 7041
4402 1146
4403 6201
4404 765
4405 4516
4406 4471
4407 5645
4410 7412
4411 761
4412 1112

```

```

/GET A RANDOM NUMBER
/SAVE BITS 06-08
/SAVE FOR OBSERVATION
/FETCH IT
/ADD CDF
/STORE FOR EXECUTION
/FOLLOWING IS A SERIES OF CDF
/NOISE MAKERS.

```

```

/EXECUTE ACTUAL CDF
/GET DATA FIELD
/SAVE BITS 06-08
/SAVE FOR TYPING
/FETCH IT
/2'S COMPLEMENT
/COMPARE
/RESTORE DATA FIELD
/INCCORRECT IF NOT ZERO
/CHECK WITH MONITOR
/PROBLEMS WITH NOISY DATA FIELD
/MESSAGE POINTER
/ERROR HALT
/GO TO NEXT TEST
/SCOPE LOOP; ISZ LOOP

```

```

PMASK
REG8
REGC
REG8
W0
NERROR
ERROR

```

```

0413 4544 /LMODE
0414 0114 /GATE SHAKER TEST
0415 3146 /
0416 1146 TST06, JMS I RANDOM
0417 1104 AND LMASK
0420 3252 DCA REGB
0421 6141 TAD REGB
0422 0640 TAD KLDF
0423 0677 DCA NOW2
0424 0660 LINC
0425 0650 LDF 00
0426 0644 LDF 37
0427 0642 LDF 20
0430 0641 LDF 10
0431 0665 LDF 04
0432 0652 LDF 02
0433 0647 LDF 01
0434 0670 LDF 25
0435 0640 LDF 12
0436 0641 LDF 07
0437 0642 LDF 30
0440 0643 LDF 00
0441 0644 LDF 01
0442 0645 LDF 02
0443 0646 LDF 03
0444 0644 LDF 04
0445 0647 LDF 05
0446 0650 LDF 06
0447 0657 LDF 07
0450 0667 LDF 10
0451 0677 LDF 17
0452 0000 LDF 27
0453 0500 LDF 37
0454 6214 NOW2,
0455 0002 W000
0456 7110 IOB
0457 3147 RAR CLL
0460 1147 DCA REGC
0461 7041 TAD REGC
0462 1146 CIA
0463 6211 TAD REGH
0464 765 CDF
0465 4516 SMA CLA
0466 4471 JMS I MERROR
0467 5677 JMS I ERROR
0470 7402 TSTM6M
0471 7610 HLT
0472 0413 SKP CLA
TST06

```

```

/GET A RANDOM NUMBER
/SAVE BITS 07-11
/SAVE FOR OBSERVATION
/FETCH IT
/ADD LDF
/STORE FOR EXECUTION
/GO TO LINC MODE
/TRY SOME DATA FIELD
/NOISEMAKERS

```

```

/EXECUTE ACTUAL LDF
/PREPARE TO GET DATA FIELD
/GET DATA FIELD
/GO TO PMODE
/JUSTIFY WITH REGB
/SAVE FOR TYPING
/FETCH IT
/2'S COMPLEMENT
/COMPARE
/RESTORE DATA FIELD
/INCORRECT IF NOT ZERO
/CHECK WITH MONITOR
/PROBLEMS WITH NOISY DATA FIELD
/MESSAGE POINTER
/ERROR HALT
/GO TO NEXT TEST
/SCOPE LOOP: ISZ LOOP

```

/THE DATA FIELD IS NOW CONSIDERED TO BE TESTED.

/NOW CHECK RIB

/PMODE

/CHECK INTERRUPT FACILITY.

```

TST07,      0473 6041      /CHECK FOR FLAG
             0474 4556      /NOT UP; GO SET IT
             0475 1122      /GET ADDRESS RETURN
             0476 3155      /STORE IT
             0477 3142      /ZERO THE PMODE SWITCH
             0500 6001      /ENABLE INTERRUPT
             0501 7000      /WAIT
             0502 6002      /DISABLE INTERRUPT
             0503 7430      /CHECK LINK; INCORRECT IF ZERO
             0504 4516      /CHECK WITH MONITOR
             0505 4471      /INTERRUPT FAILED
             0506 5731      /MESSAGE POINTER
             0507 7402      /ERROR HALT
             0510 7610      /GO TO NEXT TEST
             0511 0473      /SCOPE LOOP; ISZ LOOP

```

/PMODE

/NOW CHECK RIB

```

TST08,      0513 4556      /CHECK FOR FLAG
             0514 1123      /NOT UP; GO SET IT
             0515 3155      /GET RETURN ADDRESS
             0516 4544      /STORE IT
             0517 0121      /GET RANDOM NUMBER
             0520 3146      /SAVE BITS 06-08
             0521 1146      /FETCH IT
             0522 1101      /ADD CDF
             0523 3324      /STORE FOR EXECUTION
             0524 0000      /EXECUTE CDF
             0525 6001      /ENABLE INTERRUPT
             0526 7000      /WAIT
             0527 6002      /DISABLE INTERRUPT
             0530 6234      /READ INTERRUPT BUFFER
             0531 7006      /JUSTIFY WITH REGS
             0532 7104      /SOME MORE
             0533 0121      /SAVE BITS 06-08
             0534 3147      /SAVE FOR TYPING
             0535 1147      /FETCH IT
             0536 7041      /2'S COMPLEMENT
             0537 1146      /COMPARE
             0540 6201      /RESTORE DATA FIELD
             0541 7650      /INCORRECT IF NOT ZERO
             0542 4516      /CHECK WITH MONITOR
             0543 4471      /LOAD SF OR RIB FAILED
             0544 5752      /MESSAGE POINTER
             0545 7402      /ERROR HALT
             0546 7610      /GO TO NEXT TEST
             0512 0512      /SCOPE LOOP; ISZ LOOP

```

LOC8,

```

0550 6041 /LMODE
0551 4556 /CHECK INTERRUPT FACILITY
0552 1125 /
0553 0036 TST9A, TSF
0554 1112 JMS I SETFLG
0555 3050 TAD PNTCA
0556 3065 AND K1777
0557 7120 TAD KLJMP
0560 6141 DCA LSET
0561 0500 DCA LPOINT
0562 6001 CLL CML
0563 0016 LINC
0564 0500 IOB
0565 6002 IOF
0566 0002 PDP
0567 7420 SNL
0570 4516 JMS I NERROR
0571 4471 JMS I ERROR
0572 6004 TST9AM
0573 7402 HLT
0574 7610 SKP CLA
0575 55 TST9A

100CA,
LNOP
LOCCA,

```

```

/CHECK FOR FLAG
/NOT UP; GO SET IT
/GET RETURN ADDRESS
/10 BIT ADDRESS
/ADD LINC JUMP
/STORE FOR EXECUTION
/ZERO THE LMODE SWITCH
/SET LINK
/GO TO LINC MODE
/PREPARE TO EXECUTE IOT
/ENABLE INTERRUPTS
/WAIT
/PREPARE TO EXECUTE IOT
/DISABLE INTERRUPTS
/BACK TO PMODE
/CHECK LINK, INCORRECT IF SET
/CHECK WITH MONITOR
/INTERRUPT FAILED
/MESSAGE POINTER
/ERROR HALT
/GO TO NEXT TEST
/ISZ LOOP; SCOPE LOOP

```

```

0575 6041 /CHECK FOR FLAG
0577 4556 /NOT UP; GO SET IT
0600 1124 /GET RETURN ADDRESS
0601 0036 /10 BIT ADDRESS
0602 1112 /ADD LINC JUMP
0603 3050 /STORE IN RETURN ADDRESS
0604 4544 /GET RANDOM NUMBER
0605 0114 /SAVE BITS 07-11
0606 3146 /SAVE FOR COMPARISON
0607 1146 /FETCH IT
0610 1124 /ADD LDF
0611 3214 /STORE FOR EXECUTION
0612 7120 /SET LINK
0613 6141 /GO TO LINC MODE
0614 0000 /EXECUTE LDF
0615 0500 /PREPARE FOR IOT
0616 6001 /ENABLE INTERRUPT
0617 0016 /WAIT
0620 0500 /PREPARE FOR IOT
0621 6002 /DISABLE INTERRUPT
0622 0500 /PREPARE FOR IOT
0623 6234 /READ INTERRUPT BUFFER
0624 0242 /JUSTIFY WITH REGB
0625 0002 /BACK TO PMODE
0626 0114 /SAVE BITS 07-11
0627 3147 /SAVE FOR TYPING
0630 1147 /FETCH IT
0631 7041 /2'S COMPLEMENT
0632 1146 /COMPARE
0633 6201 /RESTORE DATA FIELD
0634 7657 /INCORRECT IF NOT ZERO
0635 4516 /CHECK WITH MONITOR
0636 4471 /LMODE RIB FAILED
0637 6025 /MESSAGE POINTER
0640 7402 /ERROR HALT
0641 7611 /GO TO NEXT TEST
0642 1576 /SCOPE LOOP: ISZ LOOP

```

/LMODE  
/CHECK RIB

/TST09, TSF  
 JMS I SETFLG  
 TAD PNTC  
 AND K1777  
 TAD KLJMP  
 DCA LSET  
 JMS I RANDOM  
 AND LMASK  
 DCA REGB  
 TAD REGB  
 TAD KLDF  
 DCA ,+3  
 CLL CML  
 LINC  
 0000  
 IOB  
 ION  
 LNOP  
 IOB  
 IOF  
 IOB  
 RIB  
 ROL  
 POP  
 AND  
 DCA  
 TAD  
 CIA  
 TAD  
 CDF  
 SNA CLA  
 JMS I  
 JMS I  
 TST09M  
 HLT  
 SKP CLA  
 TST09

LOCC,

```

/PMODE
/DOES THE DATA FIELD SET TO ZERO FOR AN INTERRUPT
TST10,
0643 6041 TSF /CHECK FLAG
0644 4556 JMS I /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 /STORE FOR EXECUTION
0655 0000 /EXECUTE CDF
0656 6001 ION /ENABLE INTERRUPT
0657 7001 NOP /WAIT
0660 6002 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 6211 CDF /RESTORE DATA FIELD
0672 7653 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

```

```

0701 6041 /LMODE
0702 4556 /DOES THE DATA FIELD SET TO ZERO FOR AN INTERRUPT
0703 1127 /
0704 0036 TST11,
0705 1112 TSF
0706 3057 JMS I
0707 4544 AND
0710 0114 SETFLG
0711 3146 PNIE
0712 1146 AND
0713 1104 TAD
0714 3316 DCA
0715 6141 LINC
0716 0000 0000
0717 0500 IOB
0720 6001 ION
0721 0016 LNOP
0722 0500 IOB
0723 6002 IOF
0724 0500 IOB
0725 6234 RIB
0726 0242 ROL
0727 4147 STC
0730 0500 IOB
0731 6214 RDF
0732 0002 PDP
0733 7110 RAR
0734 3151 DCA
0735 1151 TAD
0736 6201 CDF
0737 7050 SNA
0740 4516 CLA
0741 4471 JMS I
0742 6123 JMS I
0743 7402 TST11M
0744 761 HLT
0745 0701 SKP CLA
TST11,
L0CE,
REGC
2
REGD
REGD
A0
NERRR
ERROR
NERRR
ERROR
TST11M
HLT
SKP CLA
TST11
/SCOPE LOOP; ISZ LOOP
/ERROR HALT
/GO TO NEXT TEST
/MESSAGE POINTER
/BACK TO PMODE
/RESTORE DATA FIELD
/INCCORRECT IF NOT ZERO
/SAVE FOR TYPING
/SAVE FOR TYPING
/SAVE FOR TYPING
/BACK TO PMODE
/READ DATA FIELD
/READ DATA FIELD
/JUSTIFY WITH REGB
/JUSTIFY WITH REGB
/ENABLE INTERRUPT
/EXECUTE LDF
/GO TO LINC MODE
/STORE FOR EXECUTION
/ADD LDF
/FETCH IT
/STORE FOR TYPING
/SAVE BITS 07-11
/GET RANDOM NUMBER
/STORE IT
/ADD LINC MODE JMP
/10 BIT ADDRESS
/GET RETURN ADDRESS
/CHECK FLAG
/NOT UP; GO SET IT

```



```

/PMODE
/00ES DCA I--TAD I WORK FOR ALL DATA FIELDS
/
TST12, CLA CLL          /CLEAR AC
      JMS I   GETBNK     /GET NEXT BANK
      SNA     /DONE?
      JMP I   TST13N     /YES, NEXT TEST VIA PAGE 0
      DCA    REG8        /SAVE BANK
      TAD    K0020       /GET CONSTANT
      DCA    REGA        /SET REGA = 20
      TAD    REGB        /GET CURRENT BANK
      RTL    /JUSTIFY
      RAL    /JUSTIFY
      TAD    KCDF        /GET CDF
      DCA    EXC12       /STORE FOR EXECUTION
      TAD    K5252       /GET CONSTANT
      DCA    0000        /EXECUTE CDF
      DCA    REGA        /STORE IN TEST BANK
      TAD    REGA        /GET IT
      DCA    00         /RESTORE DATA FIELD
      CDF    REGC        /SAVE DATA
      DCA    REGC        /FETCH IT
      TAD    REGC        /2'S COMPLEMENT
      TAD    K5252       /COMPARE
      CDF    00         /RESTORE DATA FIELD
      SNA CLA /INCORRECT IF NOT ZERO
      JMS I   ERROR      /CHECK WITH MONITOR
      JMS I   ERROR      /DCA I OR TAD I FAILED
      TST12M /MESSAGE POINTER
      HLT     /ERROR HALT
      SKP CLA /TO NEXT BANK
      TST12A /SCOPE LOOP; ISZ LOOP
      JMP I   TST12N     /NEXT BANK VIA PAGE 0
EXC12, 0000
      DCA I   REGA
      TAD I   REGA
      CDF    00
      DCA    REGC
      TAD    REGC
      CIA    /FETCH IT
      TAD    K5252
      CDF    00
      SNA CLA /RESTORE DATA FIELD
      JMS I   ERROR
      JMS I   ERROR
      TST12M
      HLT
      SKP CLA
      TST12A
      JMP I   TST12N
7300
3746 7300
3747 4474
3750 7450
3751 5562
3752 3146
3753 1025
3754 3145
3755 1146
3756 7006
3757 7004
3760 1101
3761 3363
3762 1076
3763 0000
3764 3545
3765 1545
3766 6201
3767 3147
3770 1147
3771 7041
3772 1076
3773 6201
3774 7659
3775 4516
3776 4471
3777 6167
1000 7402
1001 7610
1002 1755
1003 5561

```

```

1004 7300 /LMODE
1005 4475 /DOES STA-LDA WORK FOR ALL DATA FIELDS
1006 7450 /
1007 5244 TST13, CLA CLL GETSNL /CLEAR AC
1010 3146 JMP SNA JMS I /FIND NEXT BANK
1011 1077 DCA SNA /DONE
1012 3145 DCA JMP TST14 /YES, GO TO NEXT TEST
1013 1146 DCA REG8 /SAVE BANK
1014 1104 DCA K6020 /GET CONSTANT
1015 3222 DCA REGA /SET REGA TO 6020
1016 1145 TAD REG8 /GET CURRENT BANK
1017 3013 TAD KLDF /ADD LDF
1020 1076 DCA EXC13 /STORE FOR EXECUTION
1021 6141 DCA REGA /GET ADDRESS
1022 0000 DCA LREG1 /STORE FOR INDIRECT ACCESS
1023 1053 TAD K5252 /GET CONSTANT
1024 1013 LINC /GO TO LMODE
1025 0640 0000 /EXECUTE LDF
1026 0002 STA LREG1 /STORE INDIRECT TO DF
1027 3147 LDA LREG1 /FETCH NUMBER
1030 1147 LDF /RESTORE DATA FIELD
1031 7041 PDP /TO PMODE
1032 1076 DCA REGC /SAVE FOR TYPING
1033 6201 TAD REGC /FETCH IT
1034 7650 CIA /2'S COMPLEMENT
1035 4516 TAD K5252 /COMPARE
1036 4471 CDF 00 /RESTORE DATA FIELD
1037 6225 SNA CLA JMS I /INCORRECT IF NOT ZERO
1040 7402 JMS I ERROR /CHECK WITH MONITOR
1041 7610 TST13M /STA OR LDA FAILED
1042 1016 HLT /MESSAGE POINTER
1043 5204 SKP CLA /ERROR HALT
1044 1016 TST13A /NEXT TEST
1045 5204 JMP TST13 /SCOPE LOOP; ISZ LOOP
1046 1016 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	POP	0	/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	0026	DJR		/DISABLE JUMP SAVE RETURN
1071	7072	LJMP	,+1	/DO A LINC JUMP
1072	0002	PDP	0	/BACK TO PMODE
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

```

1105 7302 /LMODE
1106 1076 /DOES A LINC JUMP CLEAR DJR?
1107 3000 /
1110 6141 TST16, CLA CLL /CLEAR AC
1111 0026 TAD K5252 /GET CONSTANT
1112 7113 DCA 0 /SET 0
1113 7114 LINC /TO LMODE
1114 0022 DJR ,+1 /DISABLE JUMP SAVE RETURN
1115 1000 LJMP ,+1 /DO A LINC JUMP
1116 7041 PDP 0 /DO ANOTHER LINC JUMP
1117 1076 TAD 0 /BACK TO PMODE
1118 7640 CIA K5252 /GET 0
1119 4516 TAD SZA CLA /2'S COMPLEMENT
1120 4471 JMS I NERROR /COMPARE WITH CONSTANT
1121 6344 JMS I ERROR /DID DJR CLEAR?
1122 7472 TST16M /CHECK MONITOR
1123 7610 HLT /DJR FAILED TO CLEAR
1124 1105 SKP CLA /MESSAGE POINTER
1125 1105 TST16 /ERROR HALT
1126 1105 /TO NEXT TEST
/SCOPE LOOP; ISZ LOOP

```

```

1127 7302 /PMODE
1128 1076 /DOES JUMP SAVE RETURN WORK FOR 8 MODE JUMPS?
1129 3000 /
1130 5333 TST17, CLA CLL /CLEAR AC
1131 1000 TAD K5252 /GET CONSTANT
1132 5333 DCA 0 /SET 0
1133 1000 JMP ,+1 /DO AN 8 MODE JUMP
1134 7041 TAD 0 /GET 0
1135 1076 CIA K5252 /2'S COMPLEMENT
1136 7650 TAD SNA CLA /COMPARE WITH CONSTANT
1137 4516 JMS I NERROR /DID WE SAVE IN ERROR?
1140 4471 JMS I ERROR /CHECK MONITOR
1141 6371 TST17M /JUMP SAVE RETURN OPERATED IN ERROR
1142 7402 HLT /MESSAGE POINTER
1143 7610 SKP CLA /ERROR HALT
1144 1127 TST17 /TO NEXT TEST
/ISZ LOOP; SCOPE LOOP

```

```

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

```

```

1163 7300 /LMODE
1164 1076 /DOES JUMP SAVE RETURN WORK FOR NON-JUMP COMMANDS?
1165 3000 /
1166 6141 TST19, CLA CLL /CLEAR AC
1167 0500 TAD TAD K5252 /GET CONSTANT
1170 6002 LINC LINC 0 /SET 0
1171 0002 IOB IOB /GO TO LMODE
1172 1000 IOF IOF /PREPARE FOR IOT
1173 7041 PDP PDP 0 /DISABLE INTERRUPTS
1174 1076 TAD TAD /BACK TO PMODE
1175 7650 CIA CIA 0 /FETCH 0
1176 4516 SNA CLA SNA CLA K5252 /2'S COMPLEMENT
1177 4471 JMS I JMS I NERROR /CHECK MONITOR
1178 6441 TST19M TST19M ERROR /IOF/IOF CAUSED LOC 0000 TO ALTER
1201 7402 HLT HLT /MESSAGE POINTER
1202 7610 SKP CLA /ERROR HALT
1203 1163 TST19 SKP CLA /TO NEXT TEST
/ISZ LOOP; SCOPE LOOP

```

```

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

```

```

1230 733
1231 1130
1232 3155
1233 1376
1234 3000
1235 6041
1236 4556
1237 6141
1240 1006
1241 3002
1242 6001
1243 7000
1244 6002
1245 7410
1246 1000
1247 7041
1250 1076
1251 7647
1252 4516
1253 4471
1254 6506
1255 7402
1256 7610
1257 1230
1260 7340
1261 3145

/PMODE
/DOES DJR INHIBIT 8 MODE INTERRUPT SAVE?
/
TST21, CLA CLL /CLEAR AC
TAD PN1F /GET RETURN POINTER TO LOCF
DCA RETURN /SET UP INTERRUPT HANDLER
TAD K5252 /GET CONSTANT
DCA 0 /STORE IN 0
TSF /FLAG SET?
JMS I SETFLG /NO, GO SET IT
LINC /TO LMODE
DJR /SET DJR
PDP /TO PMODE
ION /ENABLE INTERRUPTS
NOP /WAIT
IOF /DISABLE INTERRUPTS
SKP 0 /IF NO INTERRUPT, THIS CAUSES ERROR
TAD 0 /GET 0
CIA /2'S COMPLEMENT
TAD K5252 /ADD CONSTANT
SZA CLA /EQUAL?
JMS I NERROR /CHECK MONITOR
JMS I ERROR /DJR INHIBITED 8 MODE INTERRUPT
TST21M /MESSAGE POINTER
HLT /ERROR HALT
SKP CLA /TO NEXT TEST
TSI21 /ISZ LOOP; SCOPE LOOP
CLA CLL CMA /SET AC=7777
DCA REGA /PRESET REGA FOR NEXT TEST
LOCF,

```

/PMODE

/WILL NON-EXISTANT MEMORY DETECT WORK FOR ALL BANKS?

```

1262 7300  /CLEAR AC
1263 1066  /GET AVAILABLE MEMORY
1264 3151  /SAVE IT
1265 1151  /FETCH IT
1266 7041  /2'S COMPLEMENT
1267 1022  /ADD MAXIMUM MEMORY
1270 3146  /SAVE IT
1271 1146  /FETCH IT
1272 7450  /HOW MUCH WAS LEFT?
1273 5370  /NONE; 32K MACHINE
1274 7010  /CHECK BIT 11
1275 7620  /IS MEMORY ODD OR EVEN?
1276 5342  /NEXT BANK IS EVEN
1277 5316  /NEXT BANK IS ODD
1300 7300  /CLEAR AC
1301 6201  /RESTORE DATA FIELD
1302 1151  /GET LAST BANK TESTED
1303 7041  /2'S COMPLEMENT
1304 1022  /COMPARE WITH MAXIMUM
1305 7640  /DONE?
1306 5342  /NO, TEST NEXT BANK
1307 4516  /CHECK MONITOR
1310 4471  /NON-EXIST DETECT FAILED
1311 6535  /MESSAGE POINTER
1312 7402  /ERROR HALT
1313 7410  /TO NEXT TEST
1314 1262  /ISZ LOOP; SCOPE LOOP
1315 5370  /JUMP OVER READ ROUTINES

```

```

TST22,  CLA CLL
        BANK
        REGD
        REGD
        K0007
        REGB
        REGB
        TST23
        RAR
        SNL CLA
        JMP READ1
        JMP READ0
        BAK22,  CLA CLL
        CDF
        TAD
        CIA
        TAD
        SZA CLA
        JMP READ1
        JMS I
        JMS I
        TST22M
        HLT
        SKP
        TST22
        JMP

```

```

/CLEAR AC
/GET AVAILABLE MEMORY
/SAVE IT
/FETCH IT
/2'S COMPLEMENT
/ADD MAXIMUM MEMORY
/SAVE IT
/FETCH IT
/HOW MUCH WAS LEFT?
/NONE; 32K MACHINE
/CHECK BIT 11
/IS MEMORY ODD OR EVEN?
/NEXT BANK IS EVEN
/NEXT BANK IS ODD
/CLEAR AC
/RESTORE DATA FIELD
/GET LAST BANK TESTED
/2'S COMPLEMENT
/COMPARE WITH MAXIMUM
/DONE?
/NO, TEST NEXT BANK
/CHECK MONITOR
/NON-EXIST DETECT FAILED
/MESSAGE POINTER
/ERROR HALT
/TO NEXT TEST
/ISZ LOOP; SCOPE LOOP
/JUMP OVER READ ROUTINES

```



```

1316 7300 /PMODE
1317 2151 /READ 0 ROUTINE FOR TST22 (USED ONLY ONCE PER PASS)
1320 1151 /
1321 7006 /READ0: CLA CLL
1322 7004 ISZ REGD
1323 1101 RTL TAD
1324 3325 RAL
1325 0000 RAL
1326 7040 KCDF
1327 3015 DCA
1330 3000 DCA
1331 1415 TAD I
1332 7440 SZA
1333 5337 JMP
1334 2000 ISZ
1335 5331 JMP
1336 5300 JMP
1337 6201 CDF
1340 3147 DCA
1341 5310 JMP

/CLEAR AC
/INCREMENT NON-EXIST BANK
/FETCH BANK NUMBER
/JUSTIFY
/JUSTIFY
/FETCH CDF 00
/STORE FOR EXECUTION
/EXECUTE CDF
/SET AC=7777
/SETUP POINTER REGISTER
/SETUP COUNTER
/FETCH NON-EXISTANT WORD
/ZERO?
/NO, ENTER ERROR ROUTINE
/INCREMENT COUNTER
/TRY NEXT LOCATION
/BANK FINISHED; RETURN
/RESTORE DATA FIELD
/SAVE AC FOR TYPEOUT
/TO ERROR MONITOR
    
```

```

1342 7300 /PMODE
1343 2151 /READ 1 ROUTINE FOR TEST 22
1344 1151 /
1345 7006 /READ1: CLA CLL
1346 7004 ISZ REGD
1347 1101 RTL TAD
1350 3351 RAL
1351 2000 RAL
1352 7040 DCA
1353 3015 DCA
1354 3000 DCA
1355 1415 TAD I
1356 7040 CMA
1357 7440 SZA
1360 5364 JMP
1361 2000 ISZ
1362 5355 JMP
1363 5300 JMP
1364 7040 CMA
1365 3147 DCA
1366 6201 CDF
1367 5310 JMP

/CLEAR AC
/INCREMENT NON-EXIST BANK
/FETCH IT
/JUSTIFY
/JUSTIFY
/FETCH CDF 00
/STORE FOR EXECUTION
/EXECUTE CDF
/SET AC=7777
/SETUP POINTER REGISTER
/SETUP COUNTER
/FETCH NON-EXISTANT *WORD
/COMPLEMENT
/ZERO?
/NO, ENTER ERROR ROUTINE
/INCREMENT COUNTER
/TRY NEXT LOCATION
/BANK FINISHED; RETURN;
/RESTORE AC
/SAVE FOR TYPEOUT
/RESTORE DATA FIELD
/TO ERROR MONITOR
    
```

/PMODE

/NOW SET UP EXTENDED MEMORY FOR FURTHER TESTING

```

1370 7300      /CLEAR AC
1371 4474      /GO FIND NEXT BANK
1372 7450      /DONE?
1373 5564      /YES, EXIT
1374 3376      /NO SAVE BANK FOR EXECUTION
1375 4553      /GO RELOCATE ALL OF MEMORY
1376 0000      /TARGET BANK
1377 7777      /ORG
1420 7777      /DEST
1421 7777      /LENGTH
1422 5563      /DO IT AGAIN
TST23:  CLA CLL
        JMS I  GETBNK
        SNA
        JMP I  TST24N
        DCA  :*2
        JMS I  RELOCR
        0000
        7777
        7777
        7777
        JMP I  TST23N

```

/TRY A CIF-ION-JMP TO ALL BANKS

```

1403 7324 CLA CLL /CLEAR AC
1424 3142 DCA PPOINT /ZERO THE PMODE SWITCH
1405 4474 JMS I GETBANK /GO GET THE NEXT BANK
1406 7457 SNA /DONE?
1407 5246 JMP TST25 /EXIT
1410 7006 RTL /JUSTIFY
1411 7024 RAL /JUSTIFY
1412 3146 DCA REGS /SAVE IT
1413 1146 TAD REGB /FETCH IT
1414 1102 TAD KCIF /ADD CIF
1415 3222 DCA I+S /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 0020 0000 /EXECUTE CIF
1423 6001 ION /ENABLE INTERRUPTS
1424 5224 JMP . /WAIT
1425 6002 IOF /DISABLE INTERRUPTS
1426 6202 CIF /BACK TO BANK 0
1427 5555 JMP I /JUMP DOWN
1430 1143 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 I /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
    
```

```

1446 7300 /LMODE
1447 3065 /TRY A LIF-ION-NOP TO ALL BANKS
1450 4474 /
1451 7450 TST25, CLA CLL /CLEAR AC
1452 5314 JMS I LPOINT /ZERO THE LMODE SWITCH
1453 3146 SNA GETBNK /GET NEXT BANK
1454 1146 JMP TST26 /DONE?
1455 7006 DCA REGB /SAVE FIELD
1456 1111 TAD REGB /FETCH IT
1457 3266 RTL KLIF /JUSTIFY
1460 1132 TAD *+7 /MAKE IT A LIF
1461 1112 TAD PNTH /STORE FOR EXECUTION
1462 3050 TAD KLJMP /GET RETURN ADDRESS
1463 6041 TSF LSET /MAKE IT A LINC JUMP
1464 4556 JMS I SETFLG /STORE FOR RETURN
1465 6141 LINC /FLAG SET?
1466 0000 0000 /NO, GO SET IT
1467 0500 IOB /GO TO LMODE
1470 6001 ION /EXECUTE LIF
1471 0016 LNOP
1472 7472 LJMP
1473 2064 LOCH, LREG
1474 1560 BCL 20
1475 6037 6037
1476 0305 ROR 5
1477 4147 STC REGC
1500 0002 PDP REGC
1501 1146 TAD REGC
1502 7041 CIA REGC
1503 1147 TAD REGC
1504 765 SNA CLA
1505 4516 JMS I NERROR
1506 4471 JMS I ERROR
1507 6631 TST25M
1510 7402 HLT
1511 741 SKP
1512 1454 TST25+6
1513 5246 JMP TST25

```

ENABLE INTERRUPTS

```

/WAIT
/GET SAVE FIELD
/CLEAR OUT ALL BUT IF
/JUSTIFY
/SAVE IT
/BACK TO PMODE
/GET TARGET IF
/2'S COMPLEMENT
/GET CURRENT IF
/EQUAL?
/CHECK MONITOR
/IF FAILED TO LOAD
/MESSAGE POINTER
/ERROR HALT
/TO NEXT TEST
/ISZ LOOP; SCOPE LOOP

```

/PMODE

/NOW GO TO EXTENDED MEMORY AND TEST RMF

1514	7300		CLA CLL	/CLEAR AC
1515	4474	JMS I	GETBNK	/GET NEXT BANK
1516	7450	SNA		/DONE?
1517	5265	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	ION		/EXECUTE CIF
1533	6001	JMP		/ENABLE INTERRUPTS
1534	5334	IOF		/GO TO EXTENDED MEMORY AND WAIT
1535	6002	RIF		/DISABLE INTERRUPTS
1536	6224	DCA I	REGCN	/GET INSTRUCTION FIELD
1537	3550	CIF	00	/SAVE IT
1540	6202	JMP	,+1	/BACK TO FIELD 0
1541	5342	CLA CLL	REGB	/CHANGE FIELDS
1542	7300	TAD		/GET TARGET FIELD
1543	1146	CIA		/2'S COMPLEMENT
1544	7041	TAD	REGC	/COMPARE WITH ACTUAL FIELD
1545	1147	SNA CLA		/EQUAL?
1546	7650	JMS I	NERROR	/CHECK MONITOR
1547	4516	JMS I	ERROR	/CIF FAILED TO FIND PROPER IF
1550	4471	TST26M		/MESSAGE POINTER
1551	6664	HLT		/ERROR HALT
1552	7402	SKP		/TO NEXT TEST
1553	7410	TST26+7		/ISZ LOOP; SCOPE LOOP
1554	1523	JMP	TST26	/DO NEXT BANK
1555	5314			

```

1556 7300 /PMODE
1557 1133 /INTERRUPT INHIBIT TEST BANK 0 - BANK N - BANK 0
1560 3155 /
1561 4474 TST27, CLA CLL
1562 7451 TAD PNTI
1563 5566 RETURN
1564 3146 OCA
1565 1146 JMS I GETBANK
1566 7006 SNA
1567 7004 JMP I TST26N
1570 1102 DCA REGB
1571 3374 TAD REGB
1572 6241 RTL
1573 4556 RAL
1574 7000 KCIF
1575 7000 ,+3
1576 5377 TSF SETFLG
1577 7000 JMS I
1600 7000 0000
1601 6002 NOP
1602 6202 JMP ,+1
1603 5204 JMS I ERROR
1604 4516 JMS I ERROR
1605 4471 TST27M
1606 6721 HLT
1607 7402 SKP
1610 7410 TST27+7
1611 1565 JMP I TST27N
1612 5565

```

```

/CLEAR AC
/GET RETURN
/SET UP HANDLER
/GET NEXT BANK
/DONE
/YES, GO TO NEXT TEST
/SAVE BANK
/FETCH IT
/JUSTIFY FOR CIF
/JUSTIFY
/MAKE IT A CIF
/STORE FOR EXECUTION
/FLAG SET?
/NO, GO SET IT
/EXECUTE CIF
/SPACER
/GO TO UPPER MEMORY
/WAIT FOR INTERRUPT
/WAIT FOR INTERRUPT
/TO HERE IF NO INTERRUPT
/BACK TO BANK 0
/JUMP INTO MONITOR
/INTERRUPT OK; CHECK MONITOR
/PMODE INTERRUPT INHIBIT FAILED
/MESSAGE POINTER
/ERROR HALT
/TO NEXT TEST
/ISZ LOOP; SCOPE LOOP
/DO NEXT BANK

```

```

1613 7300 /LMODE
1614 3065 /INTERRUPT INHIBIT TEST BANK 0 -BANK N- BANK 2
1615 1134 /
1616 1112 TST28, CLA CLL
1617 3050 DCA LPOINT
1620 4474 PNTJ
1621 7450 KIJMP
1622 5263 TAD KLJMP
1623 3146 TAD LSET
1624 1146 DCA GETBNK
1625 7006 JMS I
1626 1111 SNA
1627 3233 JMP
1630 6041 DCA REGB
1631 4556 TAD REGB
1632 6141 RTL
1633 0600 TAD KLIF
1634 0500 DCA ,+4
1635 6001 TSF
1636 7637 JMS I SETFLG
1637 0016 LINC
1640 0600 LIF
1641 0500 IOB
1642 6001 IOB
1643 7644 LJM ,+1
1644 0016 LNOP
1645 0500 IOB
1646 6002 IOF
1647 0002 PDP
1650 5254 JMP ,+4
1651 0002 PDP
1652 6002 IOF
1653 7410 SKP I NERROR
1654 4516 JMS I ERROR
1655 4471 JMS I ERROR
1656 6756 TST28M
1657 7402 HLT
1660 7610 SKP CLA
1661 1624 TST28+11
1662 5213 JMP TST28

```

```

/CLEAR AC
/CLEAR HANDLER SWITCH
/GET ERROR RETURN
/MAKE IT A LINC JUMP
/PLACE IT IN HANDLER
/GET NEXT 4K BANK
/DONE?
/YES, NEXT TEST
/SAVE BANK
/FETCH IT
/JUSTIFY FOR LMODE LIF
/MAKE IT A LIF N
/STORE FOR EXECUTION
/FLAG SET?
/NO, GO SET IT
/TO LINC MODE
/EXECUTE LIF N
/
/ENABLE INTERRUPTS (SHOULD INHIBIT)
/TO EXTENDED MEMORY
/WAIT FOR INTERRUPT
/LOAD IB
/
/ENABLE INTERRUPT AGAIN
/BACK TO BANK 0
/WAIT FOR INTERRUPT
/
/DISABLE INTERRUPT
/BACK TO PMODE
/TO NON-ERROR
/BACK HERE IF INTERRUPT OCCURS
/DISABLE INTERRUPT
/SKIP INTO ERROR
/CHECK MONITOR
/LIF FAILED TO INHIBIT INTERRUPT
/MESSAGE POINTER
/ERRCR HALT
/TO NEXT TEST
/TSZ LOOP; SCOPE LOOP
/NEXT BANK

```

/LMODE

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

```

TST29, 1663 7300 /CLEAR AC
          1664 3065 /SET L SWITCH TO OFF
          1665 1135 /GET RETURN
          1666 1112 /MAKE IT A LINC JUMP
          1667 3057 /PUT IT IN HANDLER
          1670 4474 /GET NEXT BANK
          1671 7454 /DONE?
          1672 5347 /YES, NEXT TEST
          1673 3146 /SAVE TARGET
          1674 1146 /FETCH IT
          1675 7006 /JUSTIFY FOR LIF
          1676 1111 /MAKE IT A LIF N
          1677 3325 /STORE FOR EXECUTION
          1700 6041 /FLAG SET?
          1701 4556 /NO, GO SET IT
          1702 3014 /SET UP AUTO-INDEX
          1703 1146 /GET TARGET
          1704 7006 /JUSTIFY FOR CDF
          1705 7004 /JUSTIFY
          1706 1101 /MAKE IT A CDF N
          1707 3310 /STORE FOR EXECUTION
          1710 4000 /EXECUTE CDF
          1711 7020 /SET LINC
          1712 1105 /GET IOB
          1713 3414 /CELL 0001 BANK N
          1714 1106 /GET IOF
          1715 3414 /CELL 0002, BANK N
          1716 1325 /GET LIF N
          1717 3414 /CELL 0003, BANK N
          1720 1050 /GET LUMP LOCK
          1721 3414 /CELL 0004, BANK N
          1722 6201 /RESTORE OF
          1723 1107 /SET NOP
          1724 6141 /TO L MODE
          1725 0000 /EXECUTE LIF
          1726 1500
          1727 6001
          1730 7731
          1731 1016
          1732 4000
          1733 6000
EX29,
          1734 6141 /TO UPPER MEMORY
          1735 0000 /WAIT FOR INTERRUPT
          1736 1500 /SET UP 0
          1737 6001 /JMP 0
          1738 7731
          1739 1016
          1740 4000
          1741 6000

```



```

1734 0002 LOCK, PDP
1735 6202 CIF
1736 5337 JMP
1737 7430 SZL
1740 4516 JMS I
1741 4471 JMS I
1742 7013 TST29M
1743 7402 HLT
1744 7410 SKP
1745 1674 TST29+11
1746 5263 JMP

```

```

/TO HERE AFTER INTERRUPT OR JMP 0
/BACK TO BANK 0 IF NOT THERE
/TO LOWER MEMORY
/SKIP MEANS INTERRUPT NOT INHIBITED
/CHECK MONITOR
/JMP 0 INT INH IN ERROR
/MESSAGE POINTER
/ERROR HALT
/TO NEXT TEST
/ISZ LOOP; SCOPE LOOP
/NEXT BANK

```

TST29

```

1747 7340 /LMODE
1750 3065 /WILL DJR-JMP 0 LOAD THE IF?
1751 1136
1752 2036
1753 1112
1754 3057
1755 4474
1756 7450
1757 5577
1760 7006
1761 7001
1762 3146
1763 1146
1764 1111
1765 3541
1766 1036
1767 3014
1770 1146
1771 7004
1772 121
1773 1101
1774 3375
1775 0000
1776 1105
1777 7412
2000 0000
2001 3414
2002 1110
2003 3414
2004 1050
2005 3414
2006 6141
2007 0000
2010 0006
2011 6000

1747 7340 /CLEAR AC
1750 3065 /SWITCH=1
1751 1136 /GET RETURN
1752 2036 /CLEAR BITS 0,1
1753 1112 /MAKE IT A LINC JUMP
1754 3057 /PUT IT IN HANDLER (WE WON'T USE INTERRUPTS)
1755 4474 /GET NEXT BANK
1756 7450 /DONE?
1757 5577 /YES, NEXT TEST
1760 7006 /JUSTIFY FOR LIF
1761 7001 /ADD CURRENT IF
1762 3146
1763 1146
1764 1111
1765 3541
1766 1036
1767 3014
1770 1146
1771 7004
1772 121
1773 1101
1774 3375
1775 0000
1776 1105
1777 7412
2000 0000
2001 3414
2002 1110
2003 3414
2004 1050
2005 3414
2006 6141
2007 0000
2010 0006
2011 6000

TST30, CLA CLL CMA /CLEAR AC
DCA LPOINT /SWITCH=1
TAD PNTL /GET RETURN
AND K1777 /CLEAR BITS 0,1
TAD KLJMP /MAKE IT A LINC JUMP
DCA LSET /PUT IT IN HANDLER (WE WON'T USE INTERRUPTS)
JMS I GETBANK /GET NEXT BANK
SNA /DONE?
JMP I TSTJ2N /YES, NEXT TEST
RTL /JUSTIFY FOR LIF
IAC /ADD CURRENT IF
DCA REGB
TAD REGB
TAD KLIF
DCA I PNT32N
TAD K1777
DCA PINT
TAD REGB
RAL PMASK
AND KCDF
TAD .+1
DCA K10B
0000
TAD SKP
0000
DCA I PINT
TAD KRIF
DCA I PINT
TAD LSET
DCA I PINT
LINC
0000
DJR LUMP
LJMP 0

TST30, DCA DCA /WASTE A SPACE
DCA DCA /LINC JUMP SAVE
DCA DCA /CELL 2000, BANK N
DCA DCA /RIF LINC MODE (5 BITS)
DCA DCA /CELL 2001, BANK N
DCA DCA /LJMP LOCL
DCA DCA /CELL 2002, BANK N
LINC /TO LMODE
0000 /EXECUTE LIF N
DJR /DISABLE JUMP RETURN SAVE
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	6002		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

```

2033 734 /LMODE
2034 3065 /WILL ION-LIF INHIBIT INTERRUPTS?
2035 4474 /TIMING RACE IF EP12-0018 IS NOT INSTALLED)
2036 7452 /SET AC = 7777
2037 5272 /SET SWITCH
2040 3146 /GET NEXT BANK
2041 6041 /DONE?
2042 4556 /YES, NEXT TEST
2043 1146 /SAVE BANK
2044 7006
2045 7021 /FETCH IT
2046 1111 /JUSTIFY FOR LIF
2047 3253 /GET CURRENT IF, SET LINK
2050 6141 /MAKE IT A LIF N
2051 0500 /STORE FOR EXECUTION
2052 6001 /TO LMODE
2053 0000 /
2054 0016 /ENABLE INTERRUPTS
2055 0016 /EXECUTE LIF
2056 1500 /WAIT FOR INTERRUPT
2057 6002 /WAIT
2060 1601 /
2061 1002 /DISABLE INTERRUPTS
2062 743 /TO PMODE
2063 4516 /ERROR?
2064 4471 /CHECK MONITOR
2065 7106 /LIF FAILED TO INHIBIT INTERRUPT
2066 7402 /MESSAGE POINTER
2067 741 /ERROR HALT
2070 2043 /TO NEXT TEST
2071 5233 /ISZ LOOP; SCOPE LOOP
      /NEXT BANK
      TST32X, TAD SETFLG
      RTL REGB
      IAC CML KLIF
      TAD ,+4
      DCA
      LINC
      IOB
      ION
      0000
      LNOP
      LNOP
      IOB
      IOF
      LIF
      POP
      SZL
      JMS I NERROR
      JMS I ERROR
      TST32M
      HLT
      SKP
      TST32X
      JMP TST32

```

```

2072 7301 /LMODE
2073 4474 /DOES LIF CAUSE THE IF/DF TO TRANSFER TO THE SF?
2074 7450 TST33, CLA CLL
2075 5572 JMS I GETBANK
2076 3146 SNA I
2077 6041 JMP I TST34N
2100 4556 DCA REGC
2101 1146 TSF SETFLG
2102 7006 JMS I REGC
2103 7001 TAD REGC
2104 3147 DCA REGC
2105 1147 TAD REGC
2106 7040 CMA LMASK
2107 1114 AND REGE
2110 3152 DCA REGE
2111 1152 TAD REGE
2112 1104 TAD KLDF
2113 3472 DCA I EXDF33
2114 1147 TAD REGC
2115 1111 TAD KLIF
2116 3473 DCA I EXIF33
2117 6141 LINC
2120 1070 XDF33, 3000
2121 1000 XIF33, 2000
2122 6123 LUMP ,+1
2123 1691 LIF 1
2124 1500 IOB 2
2125 6234 RIB ,+1
2126 3640 LDF 2
2127 6131 LUMP
2130 1242 ROL
2131 1002 PDP
2132 3151 DCA REGC
2133 1147 TAD REGC
2134 7006 RTL
2135 7006 RTL
2136 7004 RAL
2137 1152 TAD REGE
2140 7041 CIA REGD
2141 1151 TAD REGD
2142 7651 SNA CLA
2143 4516 JMS I NERROR
2144 4471 JMS I ERROR
2145 714 TST33M
2146 7402 HLT
2147 7411 SKP
2150 1077 TST33+5
2151 1571 JMP I TST33N

```

```

/CLEAR AC
/GET NEXT BANK
/IDONE?
/YES, NEXT TEST
/SAVE BANK
/FLAG SET?
/NO, GO SET IT
/GET BANK
/JUSTIFY FOR LIF/LDF
/GET CURRENT IF
/SAVE IT
/FETCH IT
/COMPLEMENT
/SAVE DF BITS
/SAVE IT
/FETCH IT
/MAKE IT A LDF -N
/STORE FOR EXECUTION
/FETCH CONSTANT
/MAKE IT A LIF N
/STORE FOR EXECUTION
/TO LMODE
/EXECUTE LDF
/EXECUTE LIF
/TO UPPER MEMORY
/RESTORE IF
/
/READ SF
/RESTORE DF
/BACK TO BANK 0
/JUSTIFY
/TO PMODE
/SAVE RIB DATA
/GET IF
/JUSTIFY
/
/GET DF
/2'S COMPLEMENT
/ADD RECEIVED
/EQUAL?
/CHECK MONITOR
/LIF FAILED TO LOAD SF
/MESSAGE POINTER
/ERROR HALT
/TO NEXT TEST
/ISZ LOOP; SCOPE LOOP
/NEXT BANK

```

/LMODE  
 /WILL RMF WORK IN EXTENDED MEMORY?  
 /

2152	7320	CLA CLL	/CLEAR AC
2153	4474	JMS I GETBNK	/GET NEXT BANK
2154	7451	SNA	/DONE?
2155	5573	JMP I TST35N	/YES, NEXT TEST
2156	3146	DCA REG8	/SAVE TARGET
2157	1146	TAD REG8	/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	0000	/EXECUTE LIF N
2167	6170	LJMP ,+1	/TO UPPER MEMORY
2170	0500	IOB	/
2171	6244	RMF	/RESTORE MEMORY
2172	0500	IOB	/
2173	6234	RI8	/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	6202	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

2450

\*2400

/PMODE

/AUTO INDEX TEST (FIRST SET UP REGISTERS)

```

2450 7300 /CLEAR AC
2451 4474 /GET NEXT BANK
2452 7457 /DONE?
2453 5330 /NEXT TEST
2454 3146 /SAVE IT
2455 1146 /FETCH IT
2456 7006 /JUSTIFY
2457 7004 /JUSTIFY
2458 1102 /MAKE IT A CIF N
2459 3212 /STORE FOR EXECUTION
2460 0007 /EXECUTE CIF
2461 1146 /GET BANK
2462 7006 /JUSTIFY
2463 7004 /JUSTIFY
2464 1101 /MAKE IT A CDF N
2465 3220 /STORE FOR EXECUTION
2466 0000 /EXECUTE CDF
2467 5222 /TO UPPER MEMORY
2468 3000 /CLEAR 0
2469 7040 /COMPLEMENT AC
2470 3010 /SET END (END=7777)
2471 7040 /NOW SET AUTO 10-17 TO 7777
2472 3734
2473 7040
2474 3010
2475 7040
2476 3010
2477 7040
2478 3011
2479 7040
2480 3012
2481 7040
2482 3013
2483 7040
2484 3014
2485 7040
2486 3014
2487 7040
2488 3015
2489 7040
2490 3016
2491 7040
2492 3017
2493 7040
2494 3017

```

2445 1410  
2446 7640  
2447 5305  
2450 1411  
2451 7640  
2452 5304  
2453 1412  
2454 7640  
2455 5303  
2456 1413  
2457 7640  
2460 5302  
2461 1414  
2462 7640  
2463 5301  
2464 1415  
2465 7640  
2466 5300  
2467 1416  
2470 7640  
2471 5277  
2472 1417  
2473 7640  
2474 5276  
2475 5316

/NOW TEST REGISTERS

TAD I 10  
SZA CLA ERR10  
JMP TAD I 11  
SZA CLA ERR11  
JMP TAD I 12  
SZA CLA ERR12  
JMP TAD I 13  
SZA CLA ERR13  
JMP TAD I 14  
SZA CLA ERR14  
JMP TAD I 15  
SZA CLA ERR15  
JMP TAD I 16  
SZA CLA ERR16  
JMP TAD I 17  
SZA CLA ERR17  
JMP OK35

/FETCH INDIRECT INDEXING TO 0  
/ZERO?  
/TO ERROR LOOP.

/THIS BANK IS OK



```

2476 7001 /NOW HANDLE THE RETURN
2477 7001 /
2500 7001 ERR17, IAC /INCREMENT AC TO FAILING CELL
2501 7001 ERR16, IAC
2502 7001 ERR15, IAC
2503 7001 ERR14, IAC
2504 7001 ERR13, IAC
2505 7001 ERR12, IAC
2506 7001 ERR11, IAC
2507 6200 ERR10, IAC
2510 1020 CIF
2511 3147 JAD *+1
2512 1547 DCA K0010
2513 3151 TAD I REGC
2514 6211 DCA REGC
2515 5322 JMP FAL35
2516 6221 CDF
2517 6202 CIF
2520 5321 JMP *+1
2521 4516 JMS I NERRR
2522 4471 JMS I ERROR
2523 7236 TST35M
2524 7412 HLT
2525 7611 SKP CLA
2526 2405 TST35+5
2527 5200 JMP TST35
2530 7340 CLA CLL CMA
2531 3145 DCA REGA
2532 5733 JMP I *+1
2533 2600 TST36
2534 7777 TST36
7777 END,

```

```

/BACK TO BANK 0
/TO LOWER MEMORY
/ADD JUSTIFICATION
/SAVE FAILING CELL
/GET CONTENTS
/SAVE IT
/RESTORE DATA FIELD
/TO ERROR MONITOR
/RESTORE DATA FIELD
/RESTORE INSTRUCTION FIELD
/TO LOWER MEMORY
/CHECK MONITOR
/AUTO INDEX FAILED
/MESSAGE POINTER
/ERROR HALT
/TO NEXT BANK
/SCOPE LOOP
/NEXT BANK
/SET AC=7777
/PRESET REGA
/TO NEXT TEST
/(DONE THIS WAY TO AVOID PAGING ERRORS)

```

```

2600 *2000
/LMODE
/AUTO INDEX TEST
/
TST36, CLA CLL GETSNK /CLEAR AC
2601 4474 SNÄ JMS I /FIND NEXT BANK
2602 7450 JMP ,+6 /DONE?
2603 5211 DCA ,+3 /YES, RELOCATE
2604 3207 DCA REGB /SAVE BANK
2605 3146 DCA REGB /ZERO REGB
2606 4471 JMS I BNKSET /ZERO BANK
2607 1001 0001 /TARGET BANK TO BE SET TO ZERO
2610 5200 JMP TST36 /NEXT BANK
2611 4475 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 MA 0,1
2617 7011 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 REGR /GET BANK
2625 7012 RTR /JUSTIFY
2626 1031 AND K0077 /CLOSE ENOUGH
2627 3231 DCA ,+2 /STORE
2630 4553 JMS I RELOCR /GO RELOCATE PROGRAM
2631 3001 0000 /BANK
2632 4017 LAUTO-1 /ORG
2633 0001 0000 /DEST.
2634 1165 JMP LEND-LAUTO /LENGTH
2635 5211 JMS I BNK /NEXT FIELD
2636 4475 JMS I GETBNL /FIND TEST
2637 7451 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 REGR /FETCH TARGET
2646 1104 TAD KLOF /MAKE IT A LOF N
2647 3255 DCA ,+6 /STORE FOR EXECUTION
2648 1025 TAD K0020 /GET 20
2649 1112 TAD KLJMP /MAKE A LUMP 20
2650 3256 DCA ,+4 /STORE FOR EXECUTION
2653 0141 EXAUT, LINC /TO LMODE
2654 0000 /LIF N
2655 0000 /LOF N
2656 1000 /JMP 20

```

```
2657 6201 /  
2660 4516 /TO HERE IN PMODE IF INDEX OK  
2661 4471 LOK, CDF JMS I NERROR /CHECK MONITOR  
2662 7272 ERL36, JMS I ERROR /AUTO-INDEX FAILED (DIRECT TO HERE FROM ERROR)  
2663 7422 TST36M /MESSAGE POINTER  
2664 7410 HLT /ERROR HALT  
2665 2653 SKP /TO NEXT FIELD  
2666 5236 JMP EXAUT /SCOPE LOOP  
GOAUTO /NEXT FIELD.
```

4020

\*4020

/LMODE (THIS PORTION IS RELOCATED FOR EACH FIELD)  
/AUTO INDEX TEST (IF=OF)

LAUTO, SET 1 /SET UP REGISTERS

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

/NOW TEST THE REGISTERS

4064	1011	CLR		
4065	1021	LDA	21	/GET INDIRECT INDEX 0
4066	1050	AZE		/ZERO?
4067	1072	LUMP	ERL1	
4070	1022	LDA	22	/AUTO INDEX FAILED
4071	1050	AZE		
4072	1064	LUMP	ERL2	
4073	1023	LDA	23	
4074	1050	AZE		
4075	1063	LUMP	ERL3	
4076	1024	LDA	24	
4077	1050	AZE		
4100	1062	LUMP	ERL4	
4101	1025	LDA	25	
4102	1050	AZE		
4103	1061	LUMP	ERL5	
4104	1026	LDA	26	
4105	1050	AZE		
4106	1060	LUMP	ERL6	
4107	1027	LDA	27	
4110	1050	AZE		
4111	1057	LUMP	ERL7	
4112	1030	LDA	30	
4113	1050	AZE		
4114	1056	LUMP	ERL10	
4115	1031	LDA	31	
4116	1050	AZE		
4117	1055	LUMP	ERL11	
4120	1032	LDA	32	
4121	1050	AZE		
4122	1054	LUMP	ERL12	
4123	1033	LDA	33	
4124	1050	AZE		
4125	1053	LUMP	ERL13	
4126	1034	LDA	34	
4127	1050	AZE		
4130	1052	LUMP	ERL14	
4131	1035	LDA	35	
4132	1050	AZE		
4133	1051	LUMP	ERL15	
4134	1036	LDA	36	
4135	1050	AZE		
4136	1050	LUMP	ERL16	
4137	1037	LDA	37	
4140	1050	AZE		
4141	1047	LUMP	ERL17	

```

4142 0002 PDP /AUTO OK
4143 6201 GDF
4144 6202 CIF
4145 5746 JMP I ,+1
4146 2657 LOK
4147 0221 ERL17, XSK
4150 0221 ERL16, XSK
4151 0221 ERL15, XSK
4152 0221 ERL14, XSK
4153 0221 ERL13, XSK
4154 0221 ERL12, XSK
4155 0221 ERL11, XSK
4156 0221 ERL10, XSK
4157 0221 ERL7, XSK
4160 0221 ERL6, XSK
4161 0221 ERL5, XSK
4162 0221 ERL4, XSK
4163 0221 ERL3, XSK
4164 0221 ERL2, XSK
4165 0221 XSK
4166 0016 LNOP
4167 0016 LNOP
4170 6174 LNOP
4171 0061 LUMP ,+3
4173 0021 SET 1
4174 0001 ERL1,
4175 0021 LDA
4176 0002 21
4177 6201 PDP
4200 3604 GDF
4201 6202 DCA I
4202 5603 CIF
4203 2661 JMP I ,+1
4204 0147 ERL36
4205 0001 REGC
4206 0001 LEND,

```

```

/RETURN FOR NEXT BANK
/INCREMENT ERROR POINTER
/NONE OF THESE WILL SKIP,

```

/WASTE SOME SPACE FOR PAGING REASONS

```

/MUST BE CELL 1 THAT FAILED
/FETCH IT

```

```

/TO PMODE
/RESTORE DF
/SAVE ERROR
/RESTORE IF
/TO BANK 0
/RETURN
/ERROR POINTER
/END POINTER

```

/ALERT OPERATOR OF PASS COMPLETION (INHIBIT IF RSW 06#F1)

4226	730	PASS,	CLA CLL	REGA	/CLEAR REGA
4227	3145	DCA	00	COUNT	/INCREMENT COUNT
4210	6201	ISZ	NOP		/DON'T SKIP
4211	2017	NOP	LAS		/GET SWITCHES
4212	7000	LAS	AND	K304Z	/PICK OUT BIT 06
4213	7634	AND	SZA CLA		/SET ?
4214	0226	SZA CLA	JMP	177	/YES, INHIBIT AND RESTART
4215	764Z	JMP	TAD	PNT0	/GET POINTER TO TEXT
4216	5177	TAD	DCA I	ERROR	/CHEAT MONITOR
4217	1137	DCA I	JMP I	PASPNT	/GO TYPE MESSAGE
4220	3471	JMP I	TST37M		/MESSAGE POINTER
4221	5623	TST37M	PASPNT, ASCII		/LINKUP POINTER
4222	7323	PASPNT, ASCII			
4223	5051				

```

5000 *5000
5001 /NON ERROR MONITOR DETERMINES IF OPERATOR WANTS TO LOOP ON NON FAILING TEST
5002 /RETURN ADDRESS
5003 /SET AC = 4
5004 /GET RETURN ADDRESS
5005 /RETURN ADDRESS +4
5006 /GET SCOPE LOOP ADDRESS
5007 /STORE IT
5008 /UPDATE DATA
5009 /LOOP BACK TO TEST
5010 /READ SWITCHES
5011 /SAVE SR3
5012 /TEST AND CLEAR
5013 / LOOPING
5014 /SET AC--1
5015 /ADD ERRORS
5016 /STORE IN ERRORS
5017 /JUMP INDIRECT LOOP

/ERROR PROCESSOR, SCOPE LOOP, HALT, PRINT
ERRORS, 0 /RETURN ADDRESS STORAGE
LAS /READ SWITCHES
RAL 7004 /MOVE SR1 INTO AC00
SMA CLA /IS IT SET
JMP ASCII /NO TYPE A MESSAGE
IAD ERRORS /GET CURRENT ERROR ADDRESS
CIA /INVERT IT
DCA LSTERR /STORE IN LAST ERROR
ISZ ERRORS /YES INDEX ESCAPE
ISZ ERRORS /YES INDEX ESCAPE TO JUMP OUT
IAD ERRORS /INDEX ERRORS TO SCOPE MODE
DCA ERRORS /GET SCOPE ADDRESS
LAS ERRORS /STORE IN TYPE
SMA CLA /READ SWITCHES
HLT /MOVE SW02 TO AC0
ISZ ERRORS /IS SCOPE MODE SELECTED
ISZ ERRORS /YES CONTINUE IN SCOPE LOOP
IAD ERRORS /NO SET AC=7777
DCA ERRORS /SUBTRACT ONE FROM ERRORS
LAS ERRORS /STORE SELECTED ADDRESS
RTL ERRORS /EXIT TO NEXT TEST
SPA CLA
JMP I ERRORS
CMA
IAD ERRORS
DCA ERRORS
JMP I ERRORS

```



5050	7240	ASCII:	CLA CMA		/SET C(AC)=-1
5051	1620	TAD I	ERRORS		/GET MESSAGE ADDRESS STORAGE
5052	3014	DCA	PINT		/STORE IT IN AUTO INDEX REGISTER
5053	1220	TAD	ERRORS		/GET RETURN ADDRESS
5054	1115	TAD	LSTERR		/SUBTRACT LAST ERROR ADDRESS
5055	7650	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	ASCRXT		/TEST IT
5063	5225	JMP			/NUMBER=EXIT
5064	7040	CMA			/INVERT IT
5065	7450	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		/TYPEIT
5076	5257	JMP	ASCII+7		/MUST BE MORE WORDS THAT NEED TYPING
5077	0000	TYPECH, 0			
5100	0030	AND	K0077		/SAVE SIGNIFICANT PART
5101	3157	DCA	SPACE		/STORE WORD
5102	1157	TAD	SPACE		/FETCH IT
5103	7650	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	K0100		/TEST POLARITY
5110	1031	TAD	K240		/ADD 340
5111	1376	TAD			/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,	0		/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	TYPECH	
5142	3277		DCA	CRLF	
5143	1353		TAD		
5144	7004		RAL		
5145	7420		SNL	REDO	
5146	5337		JMP	TYPE	
5147	4574		JMS I	SPACE	
5150	2157		ISZ	HERE	
5151	5336		JMP	OCTYP	
5152	5732		JMP I		
5153	0000		0		/EXIT
5154	1374	CRLF,	TAD	K0215	/RETURN ADDRESS STORAGE
5155	4574		JMS I	TYPE	/GET CR
5156	1375		TAD	K0212	/TYPE IT
5157	4574		JMS I	TYPE	/GET LF
5160	1032		TAD	K0177	/TYPE IT
5161	5753		JMP I	CRLF	/SET TO RUBOUT
5162	1414	DATYP,	TAD I	PINT	/EXIT
5163	7450		SNA	ASCRXT	/GET A TERM OFF OF TYPE LIST
5164	5225		JMP		/END OF LIST?
5165	7047		CMA		/YES EXIT
5166	7640		SZA CLA		/INVERT
5167	5362		JMP	DATYP	/BEGINNING OF DATA
5170	4353		JMS	CRLF	/NO
5171	7300		CLA CLL		/YES OK RETURN THE TTY CARRIAGE AND LINE FEED
5172	5314		JMP	DATUM	/CLEAR AC AND LINK
5173	3334	M4444,	-4444		/GO TYPE THE DATA
5174	0215	K 215,	0215		/SWITCH CHECK
5175	0212	K 212,	0212		
5176	0240	K 240,	0240		
5177	7740	M40,	-40		

```

5200  *5200
5201  RANDY, Ø
5202  TAD RNA
5203  TAD RNB
5204  TAD K5252
5205  DCA RND
5206  TAD RND
5207  TAD RNC
5208  DCA RNA
5209  RAL
5210  TAD RNA
5211  TAD RNB
5212  TAD K5252
5213  TAD RND
5214  DCA RNC
5215  TAD RNC
5216  TAD RNB
5217  DCA RNA
5218  RAL
5219  TAD RNA
5220  TAD RNB
5221  TAD K5252
5222  TAD RND
5223  DCA RNC
5224  TAD RNC
5225  TAD RNB
5226  TAD RNA
5227  DCA RND
5228  TAD RND
5229  RAL
5230  TAD RNA
5231  TAD RNB
5232  DCA K5252
5233  TAD RND
5234  TAD RNC
5235  DCA RNB
5236  TAD RNA
5237  TAD RND
5238  JMP I RANDY
5239  7621
5240  RNA,
5241  RNB, 3542
5242  RNC, 3755
5243  RND, Ø016
5244  TYPDUT, Ø
5245  TLF
5246  TSF
5247  JMP .-1
5248  TCF
5249  CLA
5250  JMP I TYPDUT
5251  /
5252  /TELEPRINTER FLAG SET ROUTINE
5253  /
5254  FLAG, ØØØØ
5255  CLA
5256  TLF
5257  TSF
5258  JMP .-1
5259  JMP I FLAG
5260  /INDIRECT RETURN,

```

```

/NEW PAGE
/RANDOM NUMBER GENERATOR

```

```

/AC TO PRINTER
/FLAG SET?
/NOT UP; WAIT
/NOV CLEAR IT
/CLEAR AC
/INDIRECT RETURN

```

```

/CLEAR AC
/BUMP PRINTER
/WAIT 100 MS

```

```

5261 0000 /PROGRAM RELOCATOR
5262 7300 /CALL: RELOC; BANK, ORG-1, DEST-1, END-ORG.
5263 1261 /RELOC, 2000 CLA CLL /CONTAINS CALLING LOCATION +1
5264 6201 TAD RELOC /CLEAR AC
5265 3146 DCA REGB /GET BANK ADDRESS
5266 1546 TAD I REGC /RESET DATA FIELD
5267 3147 DCA REGB /SAVE ADDRESS
5268 1546 DCA REGC /BANK
5269 2146 ISZ REGB /SAVE IT
5270 1546 TAD I REGC /INCREMENT
5271 3015 DCA AUT011 /ORG-1
5272 2146 ISZ REGB /SAVE IT
5273 1546 TAD I REGC /INCREMENT
5274 3016 DCA AUT012 /DEST-1
5275 2146 ISZ REGB /SAVE IT
5276 1546 TAD I REGC /INCREMENT
5277 704 CMA /LENGTH
5278 3151 DCA REGD /COMPLEMENT
5279 2146 ISZ REGB /SAVE IT
5280 1146 TAD REGB /INCREMENT
5281 3261 DCA RELOC /GET RETURN
5282 704 RTL /SAVE RETURN
5283 704 RAL /GET BANK
5284 121 AND PMAK /JUSTIFY
5285 1181 TAD KCDF /SOME MORE
5286 3323 DCA EXCREL /SAVE BITS 06-08
5287 2151 ISZ REGD /CHECK IF DONE
5288 5321 JMP PICKUP /SAVE INSTRUCTION FOR EXECUTION
5289 3146 DCA REGB /NOT DONE; MOVE A WORD
5290 3147 DCA REGC /RESET REGISTER
5291 6201 CDF 00 /RESET REGISTER
5292 5661 JMP I RELOC /RESET DATA FIELD
5293 6201 CDF 00 /RETURN
5294 1415 TAD I AUT011 /RESET DATA FIELD
5295 3416 DCA I AUT012 /GET WORD
5296 5313 JMP I INCREL /CHANGE DATA FIELD
5297 5313 JMP I INCREL /DEPOSIT WORD
5298 5313 JMP I INCREL /CHECK BACK

```

```

5326 0000
5327 7300
5330 1726
5331 7006
5332 7004
5333 0121
5334 1101
5335 3342
5336 2326
5337 3147
5340 6201
5341 1146
5342 0000
5343 3547
5344 2147
5345 5340
5346 6201
5347 5726

/
/BANK SET
/CALL: LOCSET; BANK; REGB HAS CONSTANT
/
LOCSET, 0000
    CLA CLL
    TAD I LOCSET
    RTL
    RAL
    AND PMASK
    TAD KCDF
    DCA EXCSET
    ISZ LOCSET
    DCA REGC
    CDF 00
    TAD REGB
    TAD 0000
    DCA I REGC
    ISZ REGC
    JMP PICSET
    CDF 00
    JMP I LOCSET

/CLEAR AC
/GET BANK
/JUSTIFY
/SOME MORE
/BITS 06-08
/ADD CDF
/STORE FOR EXECUTION
/INCREMENT RETURN
/ZERO REGC
/RESET DATA FIELD
/GET CONSTANT
/EXECUTE CDF
/DEPOSIT C(REGB) IN BANK (N)
/DONE?
/NO, NEXT WORD
/RESET DATA FIELD
/RETURN
    
```

```

5400      *5400
5401      /PMODE-LMODE
5402      /INTERRUPT TEST; DO WE HAVE A SPURIOUS INTERRUPT ON-LINE?
5403      /
5404      INTTST, 0000
5405      CLA CLL CMA
5406      DCA RECA
5407      TSF
5408      JMS I SETFLG
5409      LINC
5410      LDA 20
5411      0020
5412      0004
5413      PDP
5414      JMP ,+1
5415      JMP ,+1
5416      CLA CLL
5417      TAD PNTP
5418      DCA RETURN
5419      ION
5420      NOP
5421      IOF
5422      JMS I NERRR
5423      JMS I ERROR
5424      INTSTM
5425      HLT
5426      SKP
5427      INTTST+1
5428      JMP I INTTST
5429
5430      /FLAG SET?
5431      /NOT UP; GO SET IT
5432      /TO LMODE
5433      /GET BIT 07
5434      /I/O PRESET
5435      /ESF
5436      /TO PMODE
5437      /CLEAR INHIBIT
5438      /CLEAR INHIBIT
5439      /ZERO AC, LINK
5440      /GET POINTER
5441      /SET UP RETURN
5442      /ENABLE INTERRUPTS
5443      /WAIT
5444      /DISABLE INTERRUPTS
5445      /NO INTERRUPT ON-LINE
5446      /SPURIOUS INTERRUPT?
5447      /MESSAGE POINTER
5448      /ERROR HALT
5449      /RETURN
5450      /ISZ LOOP; SCOPE LOOP
5451      /RETURN

```

```

5431 1000 /PMODE FIND BANK
5432 7300 /
5433 1066 /GETNXT, 0000
5434 7041 CLA CLL
5435 1120 TAD BANK
5436 7650 CIA
5437 5243 TAD PBANK
5440 2120 SNA CLA
5441 1120 JMP +4
5442 5631 ISZ PBANK
5443 3120 TAD PBANK
5444 5631 JMP I GETNXT
5445 3000 /LMODE FIND BANK
5446 7300 /
5447 1066 /GETNXL, 0000
5450 7006 CLA CLL
5451 1021 TAD BANK
5452 7041 RTL
5453 1113 TAD K0003
5454 7650 CIA
5455 5261 TAD LBRANK
5456 2113 JMP +4
5457 1113 ISZ LBANK
5460 5645 TAD LBANK
5461 1021 JMP I GETNXL
5462 3113 TAD K0003
5463 5645 DCA LBANK
        JMP I GETNXL

```

```

5464 1000 /RING THE BELL
5465 7404 /
5466 0031 BELLS, 0000
5467 764 AND K0100
5470 5273 SZA CLA
5471 1033 JMP +3
5472 4574 TAD K0207
5473 5674 JMS I TYPE
5474 1213 JMP I +1
        TST01

```

```

5475 1000 /CLEAR AC
5476 7404 /GET BANK
5477 0031 /2'S COMPLEMENT
5478 764 /CHECK
5479 5273 /EQUAL?
5480 1033 /YES, RESET
5481 4574 /INCREMENT
5482 4574 /FETCH IT
5483 5674 /RETURN
5484 1213 /CLEAR BANK

```

```

5485 1000 /CLEAR AC
5486 7404 /FETCH AVAILABLE BANK
5487 0031 /JUSTIFY
5488 764 /INCREASE TO MAXIMUM
5489 5273 /2'S COMPLEMENT
5490 1033 /COMPARE
5491 4574 /EQUAL?
5492 4574 /YES, RESET
5493 5674 /INCREMENT
5494 1213 /FETCH IT
5495 1000 /RETURN
5496 7404 /DON'T USE FIELDS 0-3
5497 0031 /SAVE IT
5498 764 /RETURN

```

```

5499 1000 /READ SWITCHES
5500 7404 /SAVE SR05
5501 0031 /IS IT SET?
5502 764 /YES, INHIBIT BELL
5503 5273 /GET BELL
5504 1033 /GO RING IT
5505 4574 /RETURN
5506 4574 /AVOID CLORBERING PASS COUNTER
5507 1213

```



/ERROR MESSAGES

```

5475 0024 /TST01, 0024
5476 2324 /TST01
5477 6061 /CDF OR RDF FAILED (PMODE)
5500 0003 /SENT RCVD
5501 0406
5502 4017
5503 2240
5504 2204
5505 0640
5506 0601
5507 1114
5510 0504
5511 4050
5512 2015
5513 1704
5514 0551
5515 0023
5516 0516
5517 2440
5520 2203
5521 2604
5522 4000
5523 7777
5524 0146
5525 0147
5526 0000

```

```

5527 0024 /TST02
5530 2324 /TST02
5531 6062 /CDF OR RDF FAILED (PMODE)
5532 0003 /SENT RCVD
5533 0406
5534 4017
5535 2240
5536 2204
5537 0640
5540 0601
5541 1114
5542 0504
5543 4050
5544 2015
5545 1704
5546 0551
5547 0023
5550 0516
5551 2440
5552 2203
5553 2604
5554 4000
5555 7777
5556 0146
5557 0147

```

5560 0000  
EXIT

```

5561 0024 TST03M, 0024 /TST03
5562 2324
5563 6063
5564 0014 /LDF OR RDF FAILED (LMODE)
5565 0406 /SENT RCVD
5566 4017
5567 2240
5570 2204
5571 0640
5572 0601
5573 1114
5574 0504
5575 4050
5576 1415
5577 1704
5600 0551
5601 4000
5602 2305
5603 1624
5604 4022
5605 0326
5606 0400
5607 7777
5610 0146
5611 0147
5612 0000

```

/TST04

/LDF OR RDF FAILED (LMODE)  
/SENT RCVD

```

5613 0024 TST04M, 0024
5614 2324
5615 6064
5616 0014
5617 0406
5620 4017
5621 2240
5622 2204
5623 0640
5624 0601
5625 1114
5626 0504
5627 4050
5630 1415
5631 1704
5632 0551
5633 0023
5634 0516
5635 2440
5636 2203
5637 2604
5640 4000
5641 7777
5642 146
5643 0147
5644 0000
EXIT

```

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

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

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

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

EXITA  
REGB  
REGC  
EXIT

EXITA  
REGB  
REGC  
EXIT

5731 0024  
 5732 2324  
 5733 6067  
 5734 0020  
 5735 1517  
 5736 0405  
 5737 4011  
 5740 1624  
 5741 0522  
 5742 2225  
 5743 2024  
 5744 4006  
 5745 0111  
 5746 1405  
 5747 0400  
 5750 7777  
 5751 0000

/TST07

/PMODE INTERRUPT FAILED

5752 0024  
 5753 2324  
 5754 6070  
 5755 0020  
 5756 1517  
 5757 0405  
 5760 4014  
 5761 1701  
 5762 0440  
 5763 2306  
 5764 4017  
 5765 2240  
 5766 2211  
 5767 0240  
 5770 0601  
 5771 1114  
 5772 0504  
 5773 0040  
 5774 0406  
 5775 4040  
 5776 4023  
 5777 0600  
 6000 7777  
 6001 146  
 6002 0147  
 6003 0000

/TST08

/PMODE LOAD SF OR RIB FAILED  
/ DF SF

6004 0024  
 6005 2324  
 6006 7101  
 6007 0014  
 6010 1517  
 6011 0405  
 6012 4011  
 6013 1624  
 6014 0522  
 6015 2225

/TST9A

/LMODE INTERRUPT FAILED

6016 2024  
6017 4006  
6020 1111  
6021 1405  
6022 0400  
6023 7777  
6024 0000  
EXITA  
EXIT

TST09M, 0024 /TST09

6025 0024  
6026 2324  
6027 6071  
6030 0014  
6031 1517  
6032 0405  
6033 4014  
6034 1701  
6035 0440  
6036 2306  
6037 4017  
6040 2240  
6041 2211  
6042 0240  
6043 0601  
6044 1114  
6045 0504  
6046 0040  
6047 0406  
6050 4040  
6051 4023  
6052 0600  
6053 7777  
6054 0146  
6055 0147  
6056 0000  
EXITA  
REGB  
REGC  
EXIT

/LMODE LOAD SF OK RIB FAILED  
/ DF SF

TST10M, 0024 /TST10

6057 0024  
6060 2324  
6061 6160  
6062 0020  
6063 1517  
6064 0405  
6065 4004  
6066 0640  
6067 0601  
6070 1114  
6071 0504  
6072 4024  
6073 1740  
6074 3205  
6075 2217  
6076 4017  
6077 1640  
6078 0116  
6101 4011  
6102 1624  
EXITA  
REGB  
REGC  
EXIT

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

```

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

```

/TST11

TST11M, 0024

```

6123 0024
6124 2324
6125 6161
6126 0014
6127 1517
6130 0405
6131 4004
6132 0640
6133 0601
6134 1114
6135 0504
6136 4024
6137 1740
6140 3205
6141 2217
6142 4017
6143 1640
6144 0116
6145 4011
6146 1624
6147 0522
6150 2225
6151 2024
6152 0023
6153 0516
6154 2440
6155 4023
6156 1640
6157 4022
6160 326
6161 0400
6160 7777
6163 146
6164 147
6165 0151
6166 0000

```

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

/TST12

TST12M, 0024

```

6124 0024

```

6170 2324  
6171 6162  
6172 0004  
6173 0301  
6174 4011  
6175 4055  
6176 4024  
6177 0104  
6200 4011  
6201 4006  
6202 0111  
6203 1405  
6204 0400  
6205 0611  
6206 1404  
6207 4014  
6210 1703  
6211 1640  
6212 2305  
6213 1624  
6214 4022  
6215 0326  
6216 0400  
6217 7777  
6220 0146  
6221 0145  
6222 0076  
6223 0147  
6224 0000

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

6225 0024  
6226 2324  
6227 6163  
6230 0023  
6231 2401  
6232 4055  
6233 4014  
6234 0401  
6235 4006  
6236 0111  
6237 1405  
6240 0400  
6241 0201  
6242 1613  
6243 4014  
6244 1703  
6245 1640  
6246 2305  
6247 1624  
6250 4022  
6251 326  
6252 0400  
6253 7777  
6254 0146  
6255 0145

/TST13

/STA - LDA FAILED  
/BANK LOCN SENT RCVD

TST13M, 0024

6256 3076 K5252  
6257 0147 REGC  
6260 0000 EXIT

YST14M, 0024 /YST14

/LMODE JUMP SAVE RETURN FAILED FOR NORMAL JUMP

6261 0024  
6262 2324  
6263 6164  
6264 0014  
6265 1517  
6266 0405  
6267 4012  
6270 2515  
6271 2040  
6272 2301  
6273 2605  
6274 4022  
6275 0524  
6276 2522  
6277 1640  
6300 0601  
6301 1114  
6302 0504  
6303 4006  
6304 1722  
6325 4016  
6326 1722  
6307 1501  
6310 1440  
6311 1225  
6312 1520  
6313 4000  
6314 7777  
6315 0000  
EXITA  
EXIT

TST15M, 0024 /TST15

/DJR FAILED TO INHIBIT JUMP SAVE

6316 0024  
6317 2324  
6320 6165  
6321 0004  
6322 1222  
6323 4006  
6324 0111  
6325 1405  
6326 0440  
6327 2417  
6330 4011  
6331 1610  
6332 1102  
6333 1124  
6334 4012  
6335 2515  
6336 2040  
6337 2301  
6340 2605  
6341 4000  
EXITA  
EXIT



6343 0000 EXIT

/TST16

/LMODE JUMP FAILED TO CLEAR DJR

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

/TST17

/PMODE JUMP ALTERED CELL 0000

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

/TST18

/PMODE IOP ALTERED CELL 0000

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

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

/TST19

TST19M, 0024

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

/LMODE IOF ALTERED CELL 0000

/TST20

TST20M, 0024

6465 0024  
6466 2324  
6467 6260  
6470 0020  
6471 1517  
6472 0405  
6473 4012  
6474 2515  
6475 2040  
6476 0314  
6477 0501  
6500 2205  
6501 0440  
6502 0412  
6503 2200  
6504 7777  
6515 0000  
EXITA  
EXIT

/PMODE JUMP CLEARED DJR

/TST21

TST21M, 0024

6506 0024  
6507 2324  
6510 6261  
6511 0024  
1222

/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

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

/TST22  
 /NON-EXISTANT MEMORY READ-BACK FAILED  
 /BANK DATA  
 /REGD REGC

TST23M, 0000  
 6574 0000  
 6575 0024  
 6576 2324  
 6577 6264

/RESERVED  
 /TST24

```

6600 0003 0003
6601 1106 1106
6602 4006 4006
6603 0111 0111
6604 1405 1405
6605 0440 0440
6606 2417 2417
6607 4040 4040
6608 1417 1417
6609 0104 0104
6610 4020 4020
6611 2217 2217
6612 2005 2005
6613 2240 2240
6614 1106 1106
6615 0023 0023
6616 0516 0516
6617 2440 2440
6618 2203 2203
6619 2604 2604
6620 4000 4000
6621 0000 EXITA
6622 0146 REGB
6623 0147 REGC
6624 0000 EXIT

                        TST25M, 0024
6631 0024 /TST25
6632 2324 2324
6633 0265 0265
6634 0014 0014
6635 1106 1106
6636 4006 4006
6637 0111 0111
6638 1405 1405
6639 0440 0440
6640 2417 2417
6641 4014 4014
6642 1701 1701
6643 0440 0440
6644 2022 2022
6645 1720 1720
6646 0522 0522
6647 4011 4011
6648 0600 0600
6649 2305 2305
6650 1624 1624
6651 4022 4022
6652 0326 0326
6653 0420 0420
6654 0000 EXITA
6655 0146 REGB
6656 0147 REGC
6657 0000 EXIT

                        TST26M, 0024
6661 0024 /TST26
6662 2324 2324
6663 0265 0265
6664 0014 0014
6665 1106 1106
6666 4006 4006
6667 0111 0111
6668 1405 1405
6669 0440 0440
6670 2417 2417
6671 4014 4014
6672 1701 1701
6673 0440 0440
6674 2022 2022
6675 1720 1720
6676 0522 0522
6677 4011 4011
6678 0600 0600
6679 2305 2305
6680 1624 1624
6681 4022 4022
6682 0326 0326
6683 0420 0420
6684 0000 EXITA
6685 0146 REGB
6686 0147 REGC
6687 0000 EXIT

                        TST26M, 0024
6691 0024 /TST26
6692 2324 2324
6693 0265 0265
6694 0014 0014
6695 1106 1106
6696 4006 4006
6697 0111 0111
6698 1405 1405
6699 0440 0440
6700 2417 2417
6701 4014 4014
6702 1701 1701
6703 0440 0440
6704 2022 2022
6705 1720 1720
6706 0522 0522
6707 4011 4011
6708 0600 0600
6709 2305 2305
6710 1624 1624
6711 4022 4022
6712 0326 0326
6713 0420 0420
6714 0000 EXITA
6715 0146 REGB
6716 0147 REGC
6717 0000 EXIT

```

6665 2324  
6266  
0003  
1106  
4006  
0111  
1405  
0440  
2417  
4006  
1116  
0440  
2022  
1720  
0522  
4015  
0515  
1722  
3100  
2305  
1624  
4022  
0326  
0400  
EXITA  
REG8  
REGC  
EXIT

6665 2324  
6266  
0003  
1106  
4006  
0111  
1405  
0440  
2417  
4006  
1116  
0440  
2022  
1720  
0522  
4015  
0515  
1722  
3100  
2305  
1624  
4022  
0326  
0400  
EXITA  
REG8  
REGC  
EXIT

/TST27  
/PMODE INTERRUPTS NOT INHIBITED BY CIF  
/BANK

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

TST27M,

0024 2324 6267 0020 1517 0405 4011 1624 0522 2225 2024 2340 1617 2440 1116 1011 0211 2405 0440 0231 4003 1106 4000 0201 1613 4000

6753 7777 EXITA

6754 1146 REG8

6755 0000 EXIT

6756 0024 TST28M, 0024

6757 2324 2324

6760 6270 6270

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

/TST28

/LMODE JMP 0 FAILED TO CLEAR

/INTERRUPT INHIBIT

/LMODE LIF FAILED TO INHIBIT INTERRUPTS

/BANK

TST29M, 0024

7013 0024 0024

7014 2324 2324

7015 6271 6271

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

7040 2024  
 7041 4011  
 7042 1610  
 7043 1102  
 7044 1124  
 7045 4000  
 7046 0201  
 7047 1613  
 7050 4000  
 7051 7777  
 7052 1146  
 7053 0000  
 EXITA  
 REGB  
 EXIT

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

TST30M, 0024  
 7054 0024  
 7055 2324  
 7056 6360  
 7057 0014  
 7060 1517  
 7061 0405  
 7062 4004  
 7063 1222  
 7064 5512  
 7065 1520  
 7066 4060  
 7067 4006  
 7070 0111  
 7071 1405  
 7072 0440  
 7073 2417  
 7074 4014  
 7075 1701  
 7076 0440  
 7077 1106  
 7080 0002  
 7081 0116  
 7082 1300  
 7083 7777  
 7084 1146  
 7085 0000  
 EXITA  
 REGB  
 EXIT

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

TST32M, 0024  
 7086 0024  
 7087 2324  
 7088 6362  
 7089 0014  
 7092 1517  
 7093 0405  
 7094 4011  
 7095 1716  
 7096 5514  
 7097 1106  
 7098 4006  
 7099 0111  
 7102 1405  
 7103 0440

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

TST33M, 0024  
 7140 0024  
 7141 2324  
 7142 6363  
 7143 0014  
 7144 1517  
 7145 0405  
 7146 4014  
 7147 1106  
 7150 5512  
 7151 1520  
 7152 4016  
 7153 4006  
 7154 0111  
 7155 1405  
 7156 0440  
 7157 2417  
 7160 4014  
 7161 1701  
 7162 0440  
 7163 2306  
 7164 0040  
 7165 1106  
 7166 4040  
 7167 4004  
 7170 0640  
 7171 4040  
 7172 2306  
 7173 4000  
 7174 7777  
 7175 1147  
 7176 1152  
 7177 1151  
 7180 0000  
 EXIT

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

TST34M, 0024  
 7201 1024  
 7202 2324  
 7203 6364  
 7204 0014  
 7205 1517  
 7206 0405  
 7207 4022  
 7208 1526

/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 7777  
 7233 0146 0146  
 7234 0147 0147  
 7235 0007 0007

TST35M, 0024

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

7236 0024 0024  
 7237 2324 2324  
 7240 6365 6365  
 7241 0020 0020  
 7242 1517 1517  
 7243 2405 2405  
 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 7777  
 7266 0146 0146  
 7267 0147 0147  
 7270 0151 0151  
 7271 0007 0007

TST36M, 0024

/TST36  
 /LMODE AUTO-INDEX FAILED  
 /FIELD LOCN

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

```

7276 1517
7277 0405
7300 4001
7301 2524
7302 1755
7303 1116
7304 0405
7305 3040
7306 0601
7307 1114
7310 0504
7311 0006
7312 1105
7313 1404
7314 4014
7315 1703
7316 1600
7317 7777
7320 0146
7321 0147
7322 0000
7323 0005
7324 3024
7325 4015
7326 0515
7327 4024
7330 2324
7331 4020
7332 0123
7333 2355
7334 5555
7335 7777
7336 0017
7337 4444

7340 0023
7341 2025
7342 2211
7343 1725
7344 2342
7345 1116
7346 2405
7347 2222
7350 2520
7351 2441
7352 0005
7353 0101
7354 0503
7355 134
7356 1117
7357 034
7360 1157
7361 174
7362 2022
7363 523

TST37M, 0005
3024
4015
0515
4024
2324
4020
0123
2355
5555
EXITA
COUNT
EXITB

INTSTM, 2023
2025
2211
1725
2342
1116
2405
2222
2520
2441
2005
0101
0503
134
1117
034
1157
174
2022
523

/EXT MEM TST PASS--(PASS)

/SPECIAL RESTART: EVENTUALLY GETS TO TST01
/SPURIOUS INTERRUPT:
/(CHECK IOC I/O PRESET)

```

/PDP-1 UNATTENDED MEMORY TEST, VERSION 2;

7364 W524  
7365 5100  
7366 5000

0524  
5100  
EXIT

MAINDEC 12-D1AC-L

V141

8-OCT-70

10:32 PAGE 57-10

5





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
BKNSSET	0070	GETNXL	5445	LOCB	0527	PNTH	0132
BSE	1000	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	INTSIM	7340	LOCF	1246	PNTD	0137
DATYP	5162	INTTST	5400	LOGG	1430	PNTP	0140
DEST36	2633	I08	0500	LOCH	1473	PONT	4204
DJR	0006	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	REG8	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	REGI	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	2512	KCOF	0101	NOW2	0452	ROL	0240
ERR14	2501	KCIF	0102	OCTYP	5132	ROR	0300
ERR15	2500	KHLT	0103	OK35	2516	SET	0060
ERR16	2477	KI08	0105	PASPNT	4223	SETFLG	0156
ERR17	2476	KI0F	0106	PASS	4206	SPACE	0157
ERRORS	0071	KLOF	0104	PASSN	0117	SHO	1500
EX29	5020	KLIF	0111	PBANK	0120	STA	1040
EXAUT	1725	KLJMP	0112	PDP	0002	START	0200
EXC12	2653	KNOP	0107	PICKUP	5321	STC	4000
EXC13	1763	KRIF	0110	PICSET	5340	TST01	0213
EXC14	1022	LAUTO	0222	PINT	0014	TST01M	5475
EXCREL	5323	LBANK	0113	PINTR	0000	TST02	2240

TST02M	5527	TST25M	6631
TST03	0265	TST26	1514
TST03M	5961	TST26M	6664
TST04	0315	TST27	1556
TST04M	5613	TST27M	6721
TST05	0345	TST27N	0169
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	0804	TST34M	7201
TST13A	0816	TST34N	0172
TST13M	6225	TST35	2400
TST13N	0162	TST35M	7236
TST14	0844	TST35N	0173
TST14M	6261	TST36	2600
TST15	0864	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