

IDENTIFICATION

PRODUCT CODE:           MAINDEC-08-DJKKA-B-D  
-----  
PRODUCT NAME:           PDP-8/A CPU TEST  
-----  
DATE CREATED:           FEBRUARY 15, 1975  
-----  
MAINTAINER:             DIAGNOSTIC GROUP  
-----  
AUTHOR:                 MICHAEL J. HARE  
-----

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

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

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

COPYRIGHT (C) 1974,1975 BY DIGITAL EQUIPMENT CORPORATION



## TABLE OF CONTENTS

1. ABSTRACT
2. REQUIREMENTS
  - 2.1 HARDWARE
  - 2.2 STORAGE
  - 2.3 PREREQUISITE SOFTWARE
3. RESTRICTIONS
4. STANDARD TEST PROCEDURE
  - 4.1 PROGRAM LOADING PROCEDURE
    - 4.1.1 4K TO 32K MACHINES
    - 4.1.2 1K TO 3K MACHINES
  - 4.2 PROGRAM RUN PROCEDURE
    - 4.2.1 INITIALIZATION
    - 4.2.2 PROGRAM START
    - 4.2.3 XOR VERSION
    - 4.2.4 ACT-8/A VERSION
    - 4.2.5 ACT-8/E VERSION
5. ERRORS
  - 5.1 ERROR HALTS
    - 5.1.1 LOOPING ON ERROR
  - 5.2 ERROR PRINTOUTS
6. SWITCH REGISTER SETTINGS
  - 6.1 NORMAL OPERATION SWITCHES
  - 6.2 ERROR SWITCHES
7. REVISIONS
8. PROGRAM DESCRIPTION
  - 8.1 BASIC INSTRUCTION TESTS
  - 8.2 TESTS OF ROMS "D" AND "F"
  - 8.3 DATA BREAK/INTERRUPT TESTING
  - 8.4 UNTESTED LOGIC
9. FLOW CHARTS
10. LISTING

-----  
HOW TO USE THIS DOCUMENT FOR:  
-----

LOADING AND RUNNING THE DIAGNOSTIC  
-----

READ SECTION 3, THEN SEE SECTIONS 4.1 AND 4.2 FOR LOADING AND STARTING INFORMATION. THERE ARE SEVEN (7) PAPER TAPES ASSOCIATED WITH THIS PROGRAM:

MAINDEC-08-DJKKA-B-PB1 FIELD SERVICE/XOR 4K VERSION  
MAINDEC-08-DJKKA-B-PM1 FIELD SERVICE 1K SEGMENT PART 1  
MAINDEC-08-DJKKA-B-PM2 FIELD SERVICE 1K SEGMENT PART 2(I/O SIMULATOR REQUIRED)  
MAINDEC-08-DJKKA-B-PB2 ACT-8/A 4K VERSION \*  
MAINDEC-08-DJKKA-B-PB3 ACT-8/A 1K SEGMENT PART 1 \*  
MAINDEC-08-DJKKA-B-PB4 ACT-8/A 1K SEGMENT PART 2 \*  
MAINDEC-08-DJKKA-B-PB5 ACT-8/E VERSION \*\*

- \* WILL RUN ONLY IN CONJUNCTION WITH THE ACT-8/A INTERFACING PROGRAM AND THE ACT-8/A OPERATING SYSTEM.
- \*\* WILL RUN ONLY UNDER CONTROL OF THE ACT-8/E SYSTEM.

FINDING ERROR INFORMATION  
-----

SEE SECTION 5 FOR ERROR INFORMATION. CHECK SECTION 3 FOR POSSIBLE VIOLATION OF PROGRAM RESTRICTIONS.

XOR TROUBLESHOOTING  
-----

SEE SECTION 4.2.3 FOR XOR INITIALIZATION INSTRUCTIONS.

1. ABSTRACT  
-----

THE PDP-8/A INSTRUCTION TEST IS DESIGNED TO TEST ALL LOGIC ON THE 8/A CPU BOARD THAT IS TESTABLE BY THE USE OF PROGRAMMED INSTRUCTIONS. IN ORDER TO TEST LOGIC ON THE CPU BOARD THAT IS DIRECTLY CONCERNED WITH DATA BREAKS AND/OR INTERRUPTS, A SPECIAL SIMULATOR IS REQUIRED. THIS TEST WILL RUN IN ANY FIELD (SEE 2.2 BELOW), AND WILL POWER FAIL IF A POWER FAIL OPTION IS INSTALLED IN THE CPU.

2. REQUIREMENTS  
-----

2.1 HARDWARE  
-----

THE FOLLOWING HARDWARE IS REQUIRED FOR THE EXECUTION OF THE 8/A INSTRUCTION TEST:

PROCESSOR-PDP-8/A HEX CPU BOARD (SEE SECTION 3-B)

MEMORY-MINIMUM 1K REQUIRED

OPTIONS- NONE REQUIRED

SPECIAL- PDP-8/A I/O SIMULATOR TO TEST I/O FUNCTIONS.

2.2 STORAGE  
-----

THE 8/A INSTRUCTION TEST USES LOCATIONS 0000 THRU 3777. SPECIAL RIM FORMAT BINARY TAPES ARE AVAILABLE TO ALLOW RUNNING THE TEST IN 1K OF MEMORY, BY EXECUTING THE TEST IN TWO CONSECUTIVE 1K SEGMENTS. IN THIS CASE ADDRESSES 0000 THRU 1777 ARE USED, WHEN RUNNING THE XOR VERSION OF THE TEST, LOCATIONS 0000 THRU 4177 ARE USED. THE 4K VERSION OF THIS TEST WILL RUN IN ANY FIELD, SO LONG AS THE LOCATIONS 00000 THRU 00177, AND LOCATION 01777 EXIST AND ARE R/W MEMORY.

2.3 PREREQUISITE SOFTWARE  
-----

NO OTHER SOFTWARE IS NECESSARY TO TEST THE 8/A CPU BOARD.

3. RESTRICTIONS  
-----

THE FOLLOWING RESTRICTIONS APPLY TO THE 8/A INSTRUCTION TEST:

- A. 1K TO 3K MACHINES-SPECIAL RIM FORMAT PAPER TAPES ARE REQUIRED TO RUN THIS TEST IN MACHINES WITH LESS THAN 4K OF MEMORY. SEE SECTION 4.1.2.
- B. TRADITIONAL PDP-8 COMPUTERS - THIS TEST IS NON-OPERATIONAL ON PDP-8, PDP-8/I, PDP-8/L, PDP-12, AND PDP-8/S COMPUTERS. THIS TEST IS OPERATIONAL ON PDP-8/E, PDP-8/F, AND PDP-8/M COMPUTERS. (SEE SECTION 4.2.1 FOR INITIALIZATION)
- C. INTERRUPTS - NO INTERRUPTS EXCEPT POWER FAIL AND THOSE FROM THE DATA BREAK/INTERRUPT SIMULATOR ARE PERMITTED.

- D. MACHINES WITH PROGRAMMER'S CONSOLE - TO RUN PART 2 OF THIS TEST (USING THE I/O SIMULATOR), IN A MACHINE WITH A PROGRAMMER'S CONSOLE INSTALLED, THE DISPLAY MUST BE SET FOR THE "MD", "STATE", OR "STATUS" OR AN ERROR HALT WILL OCCUR.
- E. DEVICE CODE "77" - IN ORDER TO COMPLETELY TEST ROM H, IT IS NECESSARY TO EXECUTE AN IOT INSTRUCTION WITH BIT 3 OF THE IOT A "1". E.G. 64XX, 65XX, 66XX, 67XX, THE IOT INSTRUCTION USED BY THE TEST IS 6770. IF THIS IOT CONFLICTS WITH A DEVICE ON THE SYSTEM UNDER TEST, DISCONNECT THE DEVICE FROM THE MACHINE WHILE RUNNING PART 2 OF THIS TEST.
- F. IT IS RECOMMENDED THAT THE PROGRAM BE RELOADED AFTER AN ERROR HAS BEEN DETECTED.
- G. NO OTHER DATA BREAK OR INTERRUPT DEVICE MAY BE OPERATING ON THE CPU WHILE RUNNING PART TWO OF THIS TEST WITH THE PDP-8/A I/O SIMULATOR.
- H. IN ORDER TO RUN PART 2 OF THIS TEST IN A CPU THAT CONTAINS A TIME SHARE OPTION(MB37 ONLY), THE TIME SHARE OPTION MUST BE DISABLED AS IT MAY CAUSE UNEXPECTED INTERRUPTS.

4. STANDARD TEST PROCEDURE

4.1 PROGRAM LOADING PROCEDURE

4.1.1 4K TO 32K MACHINES

FOR MACHINES WITH 4K TO 32K OF MEMORY THE STANDARD BINARY LOADER TECHNIQUE IS USED. USE PAPER TAPE 08-DJKKA-PB1.

4.1.2 1K TO 3K MACHINES

THE INSTRUCTION TEST IS SEGMENTED INTO TWO (2) 1K SEGMENTS THAT MAY BE RUN CONSECUTIVELY IN A 1K MACHINE. THE 1K SEGMENTS ARE PUNCHED ON TWO RIM FORMAT PAPER TAPES LABELED 08-DJKKA-PM1 AND PM2. SEGMENT 1 SHOULD ALWAYS BE RUN PRIOR TO SEGMENT 2. TO LOAD EITHER SEGMENT PERFORM THE FOLLOWING STEPS:

NOTE: THE SECOND 1K SEGMENT IS ONLY FOR USE WITH AN 8/A I/O SIMULATOR. IF NO SIMULATOR IS AVAILABLE, THE SECOND 1K SEGMENT SHOULD NOT BE RUN.

A. DEPOSIT THE FOLLOWING INSTRUCTIONS INTO PAGE ZERO:

ADDRESS	CONTENTS	
	(HIGH SPEED)	(LOW SPEED)
0156	6014	6032
0157	6011	6031
0160	5357	5357
0161	6016	6036
0162	7106	7106
0163	7006	7006
0164	7510	7510
0165	5374	5374

0166	7006	7006
0167	6011	6031
0170	5367	5367
0171	6016	6034
0172	7420	7420
0173	3776	3776
0174	3376	3376
0175	5357	5356

B. PLACE RIM TAPE IN READER ON LEADER PORTION OF TAPE.

C. LOAD ADDRESS 0156, INIT AND CONTINUE.

D. HALT COMPUTER WHEN ENTIRE TAPE HAS BEEN READ IN.

NOTE: WHEN RUNNING THE SECOND 1K SEGMENT, ADD 2000 TO THE ADDRESS OF ANY ERROR HALTS BEFORE CONSULTING THE LISTING.

#### 4.2 PROGRAM RUN PROCEDURE

##### 4.2.1 INITIALIZATION

IF NO INITIALIZATION IS PERFORMED, THE PROGRAM WILL ASSUME THAT NO PROGRAMMER'S CONSOLE OR I/O SIMULATOR IS AVAILABLE.

PERFORM THE FOLLOWING INITIALIZATION FOR OTHER CONFIGURATIONS:

##### A. MACHINES WITH PROGRAMMER'S CONSOLE

- 1) LOAD ADDRESS 0021.
- 2) SET SR0=1 TO INDICATE A PROGRAMMER'S CONSOLE IS AVAILABLE.
- 3) SET SR3=1 IF A CPU I/O SIMULATOR IS AVAILABLE.
- 4) SET SR5=1 IF RUNNING THE XOR VERSION.
- 5) SET SR6=1 IF THE PROCESSOR IS A PDP-8/E FAMILY COMPUTER.  
(PDP8=E,F, OR M)
- 6) DEPOSIT
- 7) PROGRAM IS NOW INITIALIZED

##### B. MACHINES WITHOUT PROGRAMMER'S CONSOLE

NOTE: ADDRESS 0021 IS ALREADY INITIALIZED TO INDICATE NO PROGRAMMER'S CONSOLE.

- 1) IF NO PROGRAMMER'S CONSOLE IS AVAILABLE, USE ANY MEANS AVAILABLE TO INITIALIZE LOCATION 21 AS DESCRIBED IN (A.) ABOVE.
- 2) IF RUNNING THE 2K VERSION, OR THE FIRST 1K SEGMENT, INITIALIZE LOCATION 0221 TO A 7000, TO PREVENT THE INITIAL HALT.
- 3) START PROGRAM AT LOCATION 0200.

##### 4.2.2 PROGRAM START

TO START THE TEST, DO THE FOLLOWING:

- A. PERFORM INITIALIZATION IF NECESSARY. SEE SECTION 4.2.1.
- B. LOAD ADDRESS 0200
- C. MAKE SR SETTINGS IF DESIRED. SEE SECTION 6.
- D. DEPRESS INIT, THEN CONTINUE
- E. COMPUTER SHOULD HALT WITH MA=0222, AC=7777, AND LINK=1.

- F. VERIFY THAT THE PC, AC, AND LINK ARE CORRECT. IF THEY ARE CORRECT, PROCEED TO STEP G. IF THEY ARE INCORRECT, REFER TO "ERRORS" SECTION 5.1.
- G. SET FRONT PANEL INDICATE SWITCH TO "MD", "STATE", OR "STATUS" POSITION.
- H. DEPRESS CONTINUE. !!!! DO NOT DEPRESS INIT, !!!!
- I. THE COMPUTER SHOULD RUN CONTINUOUSLY, UNLESS AN ERROR IS DETECTED, OR UNLESS SR3=1, (SEE SECTION 6),

#### 4.2.3 XOR VERSION

-----

THE 2K (STANDARD) VERSION OF THIS TEST CONTAINS ALL NECESSARY CODE TO ALLOW THIS PROGRAM TO RUN ON THE PDP-8/A XOR TESTER. TO RUN THE XOR VERSION, INITIALIZE LOCATION 0021 AS DESCRIBED IN SECTION 4.2.1, THEN START AT LOCATION 0200, AFTER THE FIRST PASS THRU THE DIAGNOSTIC, THE PROGRAM WILL BE MODIFIED FOR XOR TESTING. THE DEVICE CODE SWITCHES ON THE XOR SHOULD BE SET TO 00 WHEN RUNNING THIS TEST.

NOTE: 1K SEGMENTS OF THIS PROGRAM DO NOT CONTAIN XOR CODE.

#### 4.2.4 ACT-8/A VERSION

-----

THE ACT-8/A VERSION OF THIS PROGRAM IS A SPECIALLY MODIFIED VERSION OF THE BASIC PROGRAM. THE DIFFERENCES BETWEEN THE STANDARD VERSION AND THE ACT-8/A VERSION ARE AS FOLLOWS:

- A. ALL ERROR HALTS ARE REPLACED WITH A "JMS SUBROUTINE" IN THE ACT-8/A VERSION, THIS SUBROUTINE GETS THE PC OF THE ERROR AND DOES A JMP TO LOCATION 76520 TO NOTIFY THE ACT-8/A MONITOR OF THE PROGRAM DETECTED ERROR,
- B. AT THE END OF EACH SUCCESSFUL PASS OF THE TEST, THE PROGRAM DOES A JMS TO LOCATION 76500 TO NOTIFY THE ACT-8/A MONITOR OF THE SUCCESSFUL PASS. RETURN IS MADE TO THE PROGRAM FROM THE MONITOR AND TESTING RESUMES. TESTING WILL CONTINUE WITH THE MONITOR BEING NOTIFIED OF EACH SUCCESSFUL PASS UNTIL EITHER AN ERROR OCCURS OR THE MONITOR LOADS ANOTHER PROGRAM.
- C. XOR CODE IS NOT IMPLEMENTED IN THE ACT-8/A VERSION.
- D. THE ACT-8/A VERSION HAS THREE BINARY TAPES:

MD-08-DJKKA-PB2 4K VERSION  
 MD-08-DJKKA-PB3 1K SEGMENT PART 1  
 MD-08-DJKKA-PB4 1K SEGMENT PART 2

THESE THREE PROGRAMS ARE THE EQUIVALENT OF THE THREE FIELD SERVICE VERSIONS, WITH THE EXCEPTION OF THE DIFFERENCES NOTED ABOVE.

#### 4.2.5 ACT-8/E VERSION

-----

THE ACT-8/E VERSION OF THIS TEST IS A SPECIALLY MODIFIED VERSION OF THE BASIC 2K PROGRAM, USE PAPER TAPE DJKKA-PB5, THE DIFFERENCES BETWEEN THE ACT-8/E VERSION AND THE BASIC 2K VERSION ARE AS FOLLOWS:

- A. ALL ERROR HALTS ARE REPLACED WITH A "JMS SUBROUTINE" IN THE ACT-8/E VERSION, THE SUBROUTINE NOTIFIES THE ACT-8, MONITOR



OF THE ERROR.

- B. IF BIT2 OF LOCATION 0022 = 1, THEN AFTER ONE PASS THROUGH THE TEST THE ACT-8/E MONITOR IS NOTIFIED OF THE GOOD PASS AND PROGRAM EXECUTION CEASES.
- C. IF BIT2 OF LOCATION 0022 = 0, THE PROGRAM RUNS FOR 10 MINUTES (AS TIMED IN A CORE MEMORY MACHINE), AT THE END OF TEN MINUTES THE ACT-8/E MONITOR IS NOTIFIED OF A GOOD PASS.
- D. XOR CODE IS NOT IMPLEMENTED IN THE ACT-8/E VERSION.
- E. POWER FAIL IS NOT IMPLEMENTED IN THE ACT-8/E VERSION.
- F. THE ACT-8/E VERSION RUNS ONLY IN FIELD 0.

5. ERRORS  
-----

5.1 ERROR HALTS  
-----

ALL PROGRAM ERRORS ARE INDICATED BY MEANS OF ERROR HALTS. THE PROGRAM LISTING CONTAINS A BRIEF EXPLANATION OF THE ERROR TO THE RIGHT OF EACH HALT IN THE LISTING. USE THE PC CONTENTS AFTER THE ERROR HALT TO FIND THE ERROR INFORMATION IN THE LISTING.

NOTE: IF RUNNING THE 2ND 1K SEGMENT IN A 1K CPU, ADD 2000 TO THE ADDRESS OF ANY ERROR HALT BEFORE CONSULTING THE LISTING.

THE FOLLOWING ERRORS HAVE BEEN INCLUDED HERE, BECAUSE THEY REQUIRE FURTHER EXPLANATION:

PC=0036 THIS INDICATES THAT AN UNEXPECTED INTERRUPT WAS RECEIVED BY THE CPU, LOCATION 0000 WILL CONTAIN THE ADDRESS+1 OF WHERE THE PROGRAM WAS INTERRUPTED. E.G. IF LOCATION 0000 CONTAINS 2635, THEN THE COMPUTER WAS INTERRUPTED AFTER THE INSTRUCTION AT LOCATION 2634.

PC=0221 THIS IS NOT AN ERROR HALT UNLESS THE AC IS NOT EQUAL TO 7777 OR THE LINK IS NOT EQUAL TO 1. IF THE AC OR THE LINK IS INCORRECT, LOAD ADDRESS 0200, DEPRESS THE HALT KEY, AND DEPRESS CLEAR MOMENTARILY. THE PROGRAM MAY NOW BE EXECUTED ONE INSTRUCTION AT A TIME BY DEPRESSING THE CONTINUE KEY. THE OPERATOR SHOULD CHECK THE CONTENTS OF THE AC AND LINK AGAINST THE EXPECTED CONTENTS GIVEN IN THE PROGRAM LISTING AFTER EXECUTING EACH INSTRUCTION, WHEN AN INSTRUCTION IS EXECUTED AND THE COMPUTER REGISTERS NO LONGER AGREE, THE FAILING INSTRUCTION HAS BEEN FOUND.

PC=1631 A SKIP ERROR OCCURED DURING THE TEST OF ROMS "D" AND "F". THE INSTRUCTION IN THE AC WAS EXECUTED, AND RESULTED IN A SKIP WHEN NONE WAS EXPECTED, OR DID NOT SKIP WHEN IT WAS EXPECTED TO SKIP. MAKE A NOTE OF THE INSTRUCTION, THEN DEPRESS CONTINUE TO GET THE CONTENTS OF THE AC, MQ AND LINK AT THE TIME OF THE FAILURE. TO EXECUTE THE FAILING INSTRUCTION AGAIN, DEPRESS CONTINUE. TO EXECUTE THE NEXT INSTRUCTION (OR NEXT DATA PATTERN WITH SAME INSTRUCTION), MAKE A NOTE OF THE AC, LINK, AND MQ CONTENTS FOR REFERENCE, THEN LOAD ADDRESS 1673, DEPRESS

CLEAR , THEN CONTINUE. IF FURTHER ERROR HALTS OCCUR, THE ERROR INFORMATION SHOULD BE RECORDED FOR USE IN DETERMINING A PATTERN FOR THE FAILURE, (E.G. CLL CML COMBINATION SKIPS, SPA SNA WON'T SKIP IF LINK IS SET, ETC.)

PC=1650 A DATA ERROR OCCURRED DURING THE TEST OF ROMS "D" AND "F", THE INSTRUCTION IN THE AC WAS EXECUTED, AND RESULTED IN INCORRECT CONTENTS OF THE AC, MQ, OR LINK. MAKE A NOTE OF THE INSTRUCTION, THEN DEPRESS CONTINUE TO GET THE EXPECTED CONTENTS OF THE AC, LINK, AND MQ. MAKE A NOTE OF THE EXPECTED CONTENTS, THEN DEPRESS CONTINUE TO GET THE CONTENTS OF THE AC, MQ, AND LINK AS THEY WERE FOUND AFTER THE INSTRUCTION WAS EXECUTED. TO EXECUTE SAME INSTRUCTION AGAIN DEPRESS CONTINUE. TO EXECUTE NEXT INSTRUCTION, OR NEXT DATA PATTERN FOR SAME INSTRUCTION, LOAD ADDRESS 1673, CLEAR AND CONTINUE. IF FURTHER ERRORS OCCUR, THE ERROR INFORMATION SHOULD BE RECORDED FOR USE IN FINDING A POSSIBLE PATTERN IN THE ERROR, E.G. CLL CMA CML COMBINATION DOES NOT WORK CORRECTLY.

PC=ALL OTHER HALTS REFER TO PROGRAM LISTING UNDER PROPER PC.

### 5.1.2 LOOPING ON ERROR

NOTE: DISREGARD THIS INFORMATION WHEN RUNNING THE XOR VERSION, XOR LOOPING IS AUTOMATIC,

TO LOOP ON A FAILING INSTRUCTION, (OTHER THAN ROM "D" & "F" TEST), IT IS NECESSARY TO DEPOSIT A JUMP INSTRUCTION IN PLACE OF THE ERROR HALT THAT IS OCCURRING. THE JUMP INSTRUCTION SHOULD CAUSE THE PROGRAM TO JUMP BACK TO THE POINT WHERE THE FAILING INSTRUCTION IS EXECUTED. NOTE: IF SPECIAL CONDITIONS ARE REQUIRED (E.G. AC & LINK MUST BE CLEAR, AC MUST BE EQUAL TO 7777, ETC) THE OPERATOR WILL HAVE TO DEPOSIT THE PROPER INSTRUCTIONS TO CAUSE THESE CONDITIONS PREVIOUS TO THE FAILING INSTRUCTION, AND MAKE THE JMP AFTER THE FAILING INSTRUCTION JUMP ACCORDINGLY.

EXAMPLE:

ADDRESS	CONTENTS	MNEMONIC	
0361	1024	TAD K1	/AC TO 0001 LINK=1
0362	1054	TAD K7777	/AC TO 0000 LINK TO 0
0363	7450	SNA	/SHOULD NOT SKIP IF AC=0000
0364	7430	SZL	/SHOULD SKIP IF LINK=0
0365	7402	HLT	/CARRY FAILED TO PROPAGATE THRU ADDER

IF THE PROGRAM IS HALTING AT ADDRESS 0365 WITH THE AC NON-ZERO, ONE OR BOTH OF THE TAD INSTRUCTIONS ARE FAILING. IN ORDER TO LOOP ON THE FAILING INSTRUCTIONS, A JUMP TO THE FIRST TAD INSTRUCTION (LOCATION 0361) COULD BE PLACED AT LOCATION 0365, BUT THE AC WOULD NOT BE CLEAR, AND THE LINK WOULD NOT BE SET WHEN THE FIRST TAD IS EXECUTED. IN ORDER TO LOOP CORRECTLY, THE FOLLOWING PATCH IS REQUIRED TO SET THE AC AND LINK TO THE PROPER VALUES. "\*" INDICATES INSTRUCTIONS THAT WERE DEPOSITED FOR LOOPING PURPOSES,

ADDRESS	CONTENTS	MNEMONIC	
0360*	7320	CLA CLL CML	(CLEAR AC, SET LINK)
0361	1024	TAD K1	
0362	1054	TAD K7777	
0363	7450	SNA	
0364	7430	SZL	
0365*	5360	JMP ,-5	(FAILED, DO AGAIN)
0366*	7402	HLT	(DID NOT FAIL)

THE LOOP SHOULD BE EXECUTED THE FIRST TIME DOING ONE INSTRUCTION AT A TIME WITH HALT/SS SELECTED, TO INSURE THAT THE INSTRUCTION IS STILL FAILING, AND THAT ANY INSTRUCTIONS INSERTED (IN THE ABOVE EXAMPLE THE JMP ,-5, AND THE CLA CLL CML) ARE NOT ALSO FAILING.

IN ORDER TO RUN THE PROGRAM AGAIN AFTER REPAIRS HAVE BEEN MADE, A RELOAD OF THE PROGRAM IS REQUIRED.

## 5.2 ERROR PRINTOUTS

NO ERROR PRINTOUTS ARE USED BY THIS TEST, ALL ERRORS ARE INDICATED BY PROGRAM HALTS.

## 6. SWITCH REGISTER SETTINGS

### 6.1 NORMAL OPERATING SWITCHES

MACHINES WITH PROGRAMMER'S CONSOLE SHOULD INITIALIZE THE PROGRAM TO USE THE PROGRAMMER'S CONSOLE SWITCH REGISTER, (SEE SECTION 4.2.1). MACHINES WITHOUT PROGRAMMER'S CONSOLE USE LOCATION 0020 IN LIEU OF A SWITCH REGISTER. REGARDLESS OF WHETHER THE PROGRAMMER'S CONSOLE SWITCH REGISTER OR ADDRESS 0020 IS USED, THE BITS HAVE THE FOLLOWING PURPOSES:

BIT	FUNCTION WHEN 0	FUNCTION WHEN 1
0-2	NOT USED	NOT USED
3	LOOP ON COMPLETE TEST	HALT AT END OF TEST
4-11	NOT USED	NOT USED

## 6.2 ERROR SWITCHES

NO ERROR SWITCHES ARE PROVIDED, SEE SECTION 5.1.2 FOR ERROR LOOPING PROCEDURE

## 7. REVISIONS

THE "B" REVISION WAS RELEASED TO DELETE THE "LOAD ADDRESS" TEST WHICH COULD CAUSE MARGINAL (BUT LEGAL) FAILURES, CERTAIN TESTS WERE ALSO ADDED TO CATCH FAILURES THAT WERE NOT CAUGHT IN THE FAULT INSERTION OF THE "A" REVISION. THE "B" VERSION OF THE SOURCE ALSO SUPPORTS AN ACT-B/E VERSION.

8. PROGRAM DESCRIPTION  
-----

8.1 BASIC INSTRUCTION TESTS  
-----

DURING THE FIRST PORTION OF THE TEST, ALL BASIC GROUP 1 AND GROUP 2 OPERATE INSTRUCTIONS ARE TESTED. NO COMBINED OPERATES ARE TESTED EXCEPT CLA CLL. THEN ALL MRI INSTRUCTIONS ARE TESTED, USING BOTH DIRECT AND INDIRECT ADDRESSING. DURING THIS SECTION THE ADDER IS TESTED BY THE USE OF TAD INSTRUCTIONS AND IAC. FINALLY BASIC GROUP 3 OPERATE INSTRUCTIONS ARE TESTED.

8.2 TEST OF ROMS "D" AND "F"  
-----

ROMS "D" AND "F" ON THE CPU MODULE ARE TESTED BY EXECUTING ALL GROUP 1,2, AND 3 OPERATE INSTRUCTIONS OF THE FORM: 7XX0, 7XX1. FOR EACH INSTRUCTION THUS TESTED, 8 DIFFERENT DATA COMBINATIONS OF AC, MQ, AND LINK ARE USED. THE SEQUENCE OF ROM TESTING IS AS FOLLOWS:

- A. THE AC, LINK, AND MQ ARE SET TO SPECIFIED VALUES.
- B. THE INSTRUCTION TO BE TESTED IS EXECUTED.
- C. THE AC, LINK, AND MQ ARE SAVED, AND WHETHER THE INSTRUCTION SKIPPED IS NOTED.
- D. THE AC, LINK, AND MQ ARE SET TO THE SAME VALUES AS IN STEP A.
- E. THE INSTRUCTION IS SIMULATED USING ONLY THOSE INSTRUCTIONS THAT WERE TESTED DURING THE FIRST PART OF THE TEST. (BASIC OPERATE AND MRI INSTRUCTIONS)
- F. THE RESULTS OF THE SIMULATION ARE COMPARED TO THE RESULTS OF THE ACTUAL INSTRUCTION. ANY DIFFERENCES RESULT IN AN ERROR HALT.

8.3 DATA BREAK/INTERRUPT TESTING  
-----

THE LOGIC ON THE CPU BOARD CONCERNED WITH DATA BREAKS AND INTERRUPTS IS TESTED BY THE USE OF A SPECIAL SIMULATOR. I/O SKIPS, AC TRANSFERS, INTERRUPTS, INDICATE LOGIC, AND DATA BREAKS ARE TESTED IF THE SIMULATOR IS AVAILABLE. ALL OMNIBUS LINES ARE TESTED EITHER DIRECTLY OR INDIRECTLY WITH THE EXCEPTION OF "NEXT TIME STATE STALL" (BR2) AND THE UNUSED OMNIBUS LINE (BS2).

8.4 UNTESTED LOGIC  
-----

DUE TO CERTAIN HARDWARE RESTRICTIONS, SOME LOGIC ON THE PDP-8/A CPU BOARD IS NOT TESTED BY THIS PROGRAM. BELOW IS A LIST OF LOGIC THAT IS KNOWN TO BE UNTESTED.

ROM A - ADDRESSES 00 THRU 03 (EXTENDED LOAD ADDRESS 7)  
ADDRESSES 04 THRU 07 (EXTENDED LOAD ADDRESS)  
ADDRESSES 14 THRU 17 (LOA)

/BA CPU TEST 08-DJKKA-B  
 /PROGRAMMER: MIKE HARE  
 /COPYRIGHT © 1974,1975 DIGITAL EQUIPMENT CORP.  
 /\*\*\*\*\*

/ASSEMBLY INSTRUCTIONS:  
 /1, IF ASSEMBLING 2K VERSION, INSERT THE FOLLOWING: TWOK#0  
 /2, IF ASSEMBLING THE FIRST 1K SEGMENT, INSERT THE FOLLOWING: ONEK#1#0  
 /3, IF ASSEMBLING THE SECOND 1K SEGMENT, INSERT THE FOLLOWING: ONEK#2#0  
 / ONE AND ONLY ONE OF THE ABOVE THREE DEFINITIONS MUST BE INSERTED IN  
 / THIS SOURCE BEFORE ANY OTHER ASSEMBLY CODE,  
 /4, IF ASSEMBLING ANY OF THE ABOVE 3 VERSIONS FOR USE ON THE ACT-8/A LINE,  
 / INSERT ONE OF THE ABOVE DEFINITIONS AND THE FOLLOWING: ACT8A#0  
 /5, IF ASSEMBLING THE ACT-8/E VERSION, DEFINE ACT8E#0 AND ACT8A#0,  
 / THE ACT-8/E VERSION MUST BE A 2K VERSION, 1K VERSIONS ARE NOT  
 / SUPPORTED ON THE ACT-8/E LINE.  
 /\*\*\*\*\*

0000 TWOK#0  
 /INSTRUCTION EQUALITIES  
 7402 HLT=7402 /HALT  
 7002 BSW=7002 /BYTE SWAP  
 7421 MQL=7421 /AC TO MQ, 0 TO AC  
 7501 MQA=7501 /MQ + AC TO AQ  
 7621 CAM=7621 /CLEAR AC AND MQ  
 7521 SHP=7521 /SWAP AC AND MQ  
 7701 AQL=7701 /MQ TO AC  
 6214 RDF=6214 /READ DATA FIELD  
 6224 RIF=6224 /READ INSTRUCTION FIELD  
 6000 SKON=6000 /SKIP IF INTERRUPT ON, TURN INTERRUPT OFF  
 6001 ION=6001 /TURN INTERRUPT ON  
 6002 IOF=6002 /TURN INTERRUPT OFF  
 6003 SRQ=6003 /SKIP ON INTERRUPT REQUEST  
 6004 GTF=6004 /GET FLAGS  
 6005 RTF=6005 /RESTORE FLAGS  
 6006 SGT=6006 /SKIP ON "GREATER THAN" FLAG, NOTE: THE "GT" FLAG  
 /IS NOT IMPLEMENTED IN THE PDP-8/A  
 6007 CAF=6007 /CLEAR ALL FLAGS, AND CLEAR AC AND LINK  
 6102 SPL=6102 /SKIP ON POWER LOW  
 /\*\*\*\*\*

/XOR INSTRUCTIONS  
 6170 XRON=6170 /TURN XOR ON  
 6171 SKXR=6171 /SKIP ON XOR ERROR 1  
 6172 XRCI=6172 /TURN OFF XOR INTERRUPT  
 6173 STIP=6173 /SKIP IF MUT ON AND FIRST XRON DONE  
 6174 XRSI=6174 /SET XOR INTERRUPT ENABLE  
 6175 SXRC=6175 /SKIP ON XOR ERROR 2  
 6176 XRTO=6176 /SET TIME OUT FLOP  
 /\*\*\*\*\*

/B-A I/O SIMULATOR INSTRUCTIONS  
 6140 CNTENA=6140 /START TP1 COUNTER, ENABLE STATUS READING VIA 6141 IF BIT 11=1  
 6141 READA=6141 /READ REGISTER A, OR DATA BREAK DATA, OR STATUS.  
 6142 STROB=6142 /GATE C LINES AS DEFINED BY REGISTER B TO CPU  
 6143 SETBK=6143 /SET BREAK REQUEST, (DELAY MUST ALSO TIME OUT BEFORE BREAK OCCURS)  
 6144 SKPOFV=6144 /SKIP IF OVERFLOW F/F SET IN SIMULATOR, CLEARS SIMULATOR INTERRUPT.  
 6145 CLRAL=6145 /CLEAR SIMULATOR LOGIC.  
 6146 LOADA=6146 /LOAD REGISTER A WITH AC.  
 6147 LOADB=6147 /LOAD REGISTER B WITH AC.  
 /\*\*\*\*\*

0000 \*0  
 0001 7402 HLT/HLT(7402) /A HALT HERE INDICATES A JUMP'S FAILURE TO GATE MD TO PC.  
 0002 7402 HLT/RMF(6244) /CHANGE IF AND DF BACK  
 0003 7402 HLT/JMP SKPCHN /GET BACK TO CORRECT FIELD  
 0004 7777 BIT3,  
 0005 0000 BIT4, 0  
 0006 0000 BIT5, 0  
 0007 0000 BIT6, 0  
 0008 0000 0  
 0009 0000 0  
 0010 0000 0  
 0011 0000 0  
 0013 \*13  
 0014 0000 0  
 0015 0000 0  
 0016 0000 0  
 0017 0000 0  
 /\*\*\*\*\*

/PDP-8 STANDARDIZED SWITCHES AND HARDWARE DESIGNATOR WORDS

0020 \*20  
 0021 0000 SWITCH, 0 /PSEUDO SWITCH REGISTER  
 0022 0000 HWRDES, 0 /BIT3=1 FOR I/O TESTER, BIT5=1 FOR XOR  
 /BIT6=1 FOR B/E TYPE CPU  
 /\*\*\*\*\*  
 /SKIP CHAIN  
 0023 3074 SKPCHN, DCA SAVAC /SAVE THE AC FOR A MOMENT  
 0024 6201 CDF, 00 /DATA FIELD TO 0 FOR GETTING INTERRUPT PC  
 0025 1440 TAJ I K0 /GET CONTENTS OF 0000,  
 0026 3000 DCA 0 /PUT INTERRUPT PC IN THIS FIELD  
 0027 6102 SPL /POWER FAILURE?  
 0030 7410 SKP /NO, CHECK FOR XOR INTERRUPT  
 0031 5125 JMP, POWRDN /YES, PREPARE FOR BLACKOUT  
 0032 6244 RMF /RESTORE DF  
 0033 6144 SKPOFV /CLEAR SIMULATOR INTERRUPT  
 0034 7000 K7000, NOP  
 0035 6175 XRC /XOR ERROR? (TYPE 2)  
 0036 7402 CHNCON, HLT /NO, UNEXPECTED INTERRUPT, REPLACED BY A JMP I 7  
 /IF SOME OTHER INTERRUPT EXPECTED,  
 0037 5166 JMP, XORBAK /YES, PROCESS XOR INTERRUPT.  
 /\*\*\*\*\*

```

/CONSTANTS
0040 0000 K0, 0000
0041 0001 K1, 0001
0042 0002 K2, 0002
0043 0004 K4, 0004
0044 0010 K10, 0010
0045 0020 K20, 0020
0046 0040 K40, 0040
0047 0077 K77, 0077
0050 0100 K100, 0100
0051 0200 K200, 0200
0052 0400 K400, 0400
0053 1111 K1111, 1111
0054 1777 K1777, 1777
0055 2000 K2000, 2000
0056 2525 K2525, 2525
0057 2552 K2552, 2552
0060 3333 K3333, 3333
0061 4000 K4000, 4000
0062 4444 K4444, 4444
0063 5225 K5225, 5225
0064 5252 K5252, 5252
0065 5253 K5253, 5253
0066 6666 K6666, 6666
0067 7700 K7700, 7700
0070 7721 K7721, 7721
0071 7770 K7770, 7770
0072 7777 K7777, 7777

```

IFNDEF ACT0E <

.....  
/SCRATCH LOCATIONS

```

0074 0000 SAVAC, 0 /POWERFAIL AC STORAGE
0075 0000 SAVFLG, 0 /POWERFAIL FLAG STORAGE
0076 0000 SAVHQ, 0 /POWERFAIL HQ STORAGE
0077 0000 SAVRET, 0 /POWERFAIL RETURN STORAGE
0100 0000 PARTHO, 0 /CONTAINS 7777 WHEN IN I/O SIM PORTION OF CPU TEST
0101 0000 ACWAS, 0
0102 0000 HQWAS, 0
0103 0000 LKWAS, 0
0104 0000 SKPPED, 0
0105 0000 SOMSKP, 0
0106 0000 SAVFLD, 0 /STORAGE FOR IF AND DF
0117 BIT0=POINTD
0007 BIT7=0007
0112 BIT11=TESLOC
0107 0000 AQDATA, 0
0110 0000 HQDATA, 0
0111 0000 LKDATA, 0
0112 0000 TESLOC, 0
0113 0114 POINTR, +1
0114 2526 2526
0115 0116 POINTB, POINTC

```

```

0116 0000 POINTG, 0
0117 0000 POINTD, 0
0120 1310 POINTI, JMSLOC=1
0121 1324 POINTF, JMSLOC=2
0122 1372 INBTTR, INSTRT
0123 1577 TEST, TESTS
0124 1745 CKSWIT, XOR29> /CONTAINS THE CONSTANT 0200 FOR PART 2 OF THE 1K VERSION,

```

.....

```

/POWER FAIL ROUTINE
POWRDN, GTF /GET LINK AND FLAGS
/SAVE FLAGS
DCA I PAVFLG /HQ TO AC
SNP /SAVE HQ CONTENTS
DCA I PSAVHQ /GET RETURN ADDRESS FOR POWER-UP
K1000, TAD 0 /SAVE RETURN ADDRESS IN FLD 0,
DCA I PAVRET /INSTRUCTION FOR POWER-UP EXECUTION
TAD RETINS /PUT IT IN ADDRESS 00000
DCA I KB /WAIT OUT POWER FAILURE
KSTOP, HLT
PAVFLG, SAVFLG
PSAVHQ, SAVHQ
PAVRET, SAVRET
THOBAK, JMP I CKSWIT
PHRAD0,
+, +13
/POWER UP ROUTINE
/PHRUP, TAD SAVHQ
/RESTORE HQ
/ TAD SAVFLG /RESTORE FLAGS
/ RTF
/ CLA
/ TAD I PPRTWO
/ SET CLA /IN THE SECOND PART OF THE TEST?
/ JMP THOBAK /YES, BEGIN SECOND PART OVER,
/ TAD SAVAC /RESTORE AC
/ JMP I SAVRET /RETURN TO PROGRAM
/PPRTWO, PARTWO

```

.....

```

/XOR CODE
0156 0156
/THIS ROUTINE MUST ALWAYS REMAIN AT LOCATION 0156. IT IS REPLACED
/BY THE RIM LOADER PROGRAMS IN 1K MACHINES, AND BY ACT=8/A CODE ON THE ACT LINE.
/ROUTINE FOR SAVING XOR RETURN FOR ERROR LOOPING
0156 0000 POINT, 0
0157 7300 CLA CLL
0160 6171 SKXR /TYPE 1 ERROR EXISTING?
0161 7410 SKP /NO SWAP POINTER AND GO NEXT TEST
0162 5166 JMP XORBAK /ERROR EXISTING, LOOP ON LAST TEST.
0163 1156 TAD POINT /SWAP POINTER
0164 3173 DCA POINTX
0165 5556 JMP I POINT

```

```

/XOR RETURN AND INITIALIZE
0166 6007 XORBAK, CAF /CLEAR AC, LINK, AND WORLD

```

```

0167 7621 CAM /CLEAR MO
0170 6145 CLRAL /CLEAR I/O SIMULATOR
0171 6001 ION
0172 5573 JMP I POINTX /RETURN TO TEST THAT ERRORED,
0173 0174 POINTX, POINTX
0174 7402 POINTX, HLT /XOR INTERRUPT WHEN NOT RUNNING XOR VERSION
0177 8177 *177
0177 7770 DATPAT, 7770

0200 *200
/*****
/INITIAL CONDITIONS: AC AND LINK CLEAR FROM INITIALIZE
/*****
/TEST ALL BASIC SKIPS TO EITHER SKIP OR NOT SKIP WHEN AC=7777,
START, CLA/JMS POINT FOR XOR
0200 7200 SNA
0201 7450 HLT /SNA SKIPS WHEN AC CLEAR, OR SEL DOES NOT SKIP WHEN LINK = 0
0202 7430 SEL /AC TO 7777
0203 7402 HLT /ENABLE INTERRUPTS FOR POSSIBLE POWER FAIL
0204 7040 CMA /CMA SKIPPED OR DID NOT COMPLEMENT, OR SNA DID NOT SKIP, OR ION SKIPPED
0205 6001 ION
0206 7450 SNA
0207 7402 HLT
0210 7440 SEA
0211 7410 SKP /SNA CLEARED AC, OR SEA SKIPPED, OR SKP FAILED
0212 7402 HLT
0213 7500 SMA /SEA OR SKP CLEARED ACB, OR SMA FAILED TO SKIP
0214 7402 HLT
0215 7510 SPA
0216 7410 SKP
0217 7402 HLT /SMA CLEARED ACB, OR SPA SKIPPED WHEN AC=1
0220 7020 CML /SET LINK TO 1
0221 7402 EXHLT1, HLT/JMS POINT FOR XOR /EXPECTED HALT #1, AC SHOULD EQUAL 7777, LINK=1
/*****
/VERIFY THAT AC=7777 AND LINK=1, THEN CONTINUE
/*****
/TEST CLA CLL TO CLEAR AC AND LINK
/*****
0222 7300 TSCACL, CLA CLL
0223 7450 SNA
0224 7430 SEL
0225 7402 HLT /CLA CLL DID NOT CLEAR AC OR LINK, OR SNA SKIPPED
/WHEN AC=0000, OR SEL DID NOT SKIP WHEN LINK=0
0226 7040 CMA /AC TO 7777
/*****
/TEST BASIC SKIPS TO SKIP OR NOT SKIP WHEN AC=0000
TSSSK, CMA /AC TO 0000
0227 7040 SEA
0230 7440 HLT /CMA DID NOT COMPLEMENT AC OR SEA FAILED TO SKIP, OR CMA SKIPPED
0231 7402 SPA
0232 7510 HLT /SPA FAILED TO SKIP WHEN AC=0
0233 7402 HLT
0234 7500 SMA
0235 7410 SKP /SMA SKIPPED WHEN AC=0
0236 7402 HLT
/*****
/TEST IAC TO INCREMENT AC TO 0000 AND SET LINK

```

```

0237 7100 TSTIAC, CLL /THIS INSTRUCTION NOT YET TESTED
0240 7040 CMA /AC TO 7777
0241 7001 IAC /AC TO 0000, LINK TO 1
0242 7440 SEA
0243 7402 HLT /SEA FAILED TO SKIP, OR IAC DID NOT INCREMENT AC TO 0000
/*****
/TEST TO SEE IF LINK COMPLEMENTED TO A 1 ON A CARRY OUT OF ADDER
0244 7420 TSTLCM, SNL
0245 7402 HLT /LINK DID NOT COMPLEMENT ON CARRY OUT, OR CLL FAILED
/OR SNL FAILED TO SKIP FOR LINK = 1
0246 7430 SEL
0247 7410 SKP
0250 7402 HLT /SEL SKIPPED ON LINK=1, OR SNL CLEARED LINK, OR SKP FAILS WHEN LINK = 1
/*****
/TEST CLL TO CLEAR LINK
TSTCLL, CLL
0251 7100 SNL
0252 7420 SKP /CLL FAILED, OR SNL SKIPPED WHEN LINK=0, OR SKP FAILED WHEN LINK=0
0253 7410 HLT
0254 7402 HLT /SEL FAILED TO SKIP, OR SNL OR SKP SET LINK
0255 7430 SEL
0256 7402 HLT
/*****
/TEST ABILITY OF CML TO SET LINK
TSTCML, CML /LINK TO 1
0257 7020 SNL
0260 7420 HLT /CML DID NOT SET LINK
/*****
/TEST ABILITY OF LINK TO COMPLEMENT FROM A 1 TO A 0 ON A CARRY OUT
0262 7040 CMA /AC TO 7777, LINK=1
0263 7001 IAC /AC TO 0000, CARRY TO LINK, LINK TO 0
0264 7430 SEL
0265 7402 HLT /CARRY OUT DID NOT COMPLEMENT LINK TO A 0
/*****
/TEST ABILITY OF CML TO COMPLEMENT LINK FROM A 0 TO A 1 AND BACK TO A 0
XOR03, CLA/JMS POINT FOR XOR
0266 7200 CML /LINK TO 1
0267 7020 CML /LINK TO 0
0270 7020 SEL
0271 7430 HLT
0272 7402 HLT /CML DID NOT COMPLEMENT LINK FROM A 1 TO A 0
0273 7440 SEA
0274 7402 HLT /CML CHANGED AC
/*****
/TEST CLA TO CLEAR AC AND NOT CLEAR LINK
0275 7020 CML /MAKE LINK A 1
0276 7040 CMA /AC TO 7777
0277 7200 CLA /AC TO 0000
0300 7420 SNL
0301 7402 HLT /CLA CLEARED LINK
0302 7440 SEA
0303 7402 HLT /CLA DID NOT CLEAR AC
/*****
/TEST NOP TO NOT CHANGE AC OR LINK
0304 7000 NOP /AC=0000 LINK=1
0305 7420 SNL
0306 7402 HLT /NOP SKIPPED OR CLEARED LINK

```

```

0307 7440 SZA
0310 7402 HLT /NOP SET AC BIT
0311 7040 CMA /AC TO 7777
0312 7020 CML /LINK TO 0
0313 7000 NOP /AC=7777,LINK=0
0314 7430 SEL
0315 7402 HLT /NOP SKIPPED OR SET LINK
0316 7450 SNA
0317 7402 HLT /NOP CLEARED AC
/.....
/TEST RAL TO NOT PICK UP BITS BY SHIFTING ZEROES
XOR04, NOP/JMS POINT FOR XOR
0320 7000 CLA /AC TO 0000
0321 7200
0322 7004 RAL
0323 7450 SNA
0324 7430 SEL
0325 7402 HLT /RAL OF ZEROES PICKED UP AC BIT OR LINK BIT
/.....
/TEST RTL TO NOT PICK UP BITS WHEN SHIFTING ALL ZEROES
0326 7006 RTL
0327 7450 SNA
0330 7430 SEL
0331 7402 HLT /RTL PICKED UP BIT IN AC OR LINK WHEN SHIFTING ZEROES
/.....
/TEST RAR TO NOT PICK UP BITS WHEN SHIFTING ALL ZEROES
0332 7010 RAR
0333 7450 SNA
0334 7430 SEL
0335 7402 HLT /RAR PICKED UP BIT IN AC OR LINK WHEN SHIFTING ZEROES
/.....
/TEST RTR TO NOT PICK UP BITS WHEN SHIFTING ALL ZEROES
0336 7012 RTR
0337 7450 SNA
0340 7430 SEL
0341 7402 HLT /RTR PICKED UP BIT IN AC OR LINK WHEN SHIFTING ZEROES
/.....
/TEST BSW TO NOT PICK UP ANY BITS WHEN SWAPPING ZEROES,AND TO NOT AFFECT LINK
0342 7002 BSW
0343 7450 SNA
0344 7430 SEL
0345 7402 HLT /BSW PICKED UP BIT IN AC WHEN SWAPPING ALL ZEROES,OR SET LINK
/.....
/FIRST TEST OF MRI
/.....
/TEST TAD TO ADD 7777 TO A CLEAR AC
XOR05, NOP/JMS POINT FOR XOR
0346 7000 TAD K7777 /AC TO 7777, LINK=0
0347 1072
0350 7450 SNA
0351 7402 HLT /TAD DID NOT LOAD AC, OR TAD SKIPPED,
0352 7430 SEL
0353 7402 HLT /TAD SET LINK WHEN NO CARRY OUT EXPECTED
/.....
/TEST TAD TO ADD 1 TO AC=7777 TO PRODUCE AC=0000 AND LINK=1
0354 1041 TAD K1 /AC TO 0000,LINK TO 1
0355 7440 SZA

```

```

0356 7402 HLT /TAD USED INCORRECT VALUE,OR ADDER CARRY CKT
/FAULTY OR TAD SKIPPED
0357 7420 SNLI
0360 7402 HLT /CARRY OUT OF ADDER DID NOT COMPLEMENT LINK
/.....
/TEST ADDER CARRY STRUCTURE TO GENERATE CARRY THROUGH ALL POSITIONS
/UNDER OPPOSITE CONDITIONS FROM PREVIOUS TEST
0361 1041 TAD K1 /AC TO 0001,LINK=1
0362 1072 TAD K7777 /AC TO 0000,LINK TO 0
0363 7450 SNA
0364 7430 SEL
0365 7402 HLT /CARRY FAILED TO PROPAGATE ALL THROUGH ADDER TO LINK
/.....
/TEST ABILITY TO ADD 0000 TO A CLEAR AC TO PRODUCE A CLEAR AC
0366 1040 TAD K0 /AC=0000,LINK=0
0367 7450 SNA
0370 7430 SEL
0371 7402 HLT /ADDING 0000 TO 0000 PRODUCED NON-ZERO RESULT
/OR COMPLEMENTED LINK
/.....
/TEST ADDER'S ABILITY TO PROPAGATE CARRY WHEN ACN=1,MDN=1,AND CARRY IN N=1
0372 1072 TAD K7777 /AC TO 7777,LINK=0
0373 1072 TAD K7777 /AC TO 7776,LINK TO 1
0374 7420 SNLI
0375 7402 HLT /CARRY DID NOT PROPAGATE TO LINK
0376 7001 IAC /MAKE AC=7777 FOR EASE OF CHECKING RESULT OF PREVIOUS TAD
0377 7040 CMA /AC TO 0000
0400 7440 SZA
0401 7402 HLT /CARRY DID NOT PROPAGATE CORRECTLY
/OR TAD USED INCORRECT OPERAND
/.....
/TEST ADDER'S ABILITY TO GENERATE CARRY WHEN ACN=1, MDN=1, AND CARRY IN N=0 IN ODD NUMBERED BIT POSITIONS
0402 1056 TAD K2525 /AC TO 2525,LINK = 1
0403 1056 TAD K2525 /AC TO 5252
0404 1056 TAD K2525 /AC TO 7777,LINK = 1(NO CARRY GENERATED TO LINK)
0405 7040 CMA /AC TO 0000,LINK = 1
0406 7440 SZA
0407 7402 HLT /CARRY FAILED IN AN ODD BIT POSITION
0410 7420 SNLI
0411 7402 HLT /LINK COMPLEMENTED WHEN NO CARRY OUT EXPECTED
/.....
/TEST ADDER'S ABILITY TO GENERATE CARRY WHEN ACN=1,MDN=1, AND CARRY IN N=0 IN ALL EVEN BIT POSITIONS
0412 1064 TAD K5252 /AC TO 5252,LINK=1
0413 1064 TAD K5252 /AC TO 2924,LINK TO 0
0414 1064 TAD K5252 /AC TO 7776,LINK=0
0415 7001 IAC /AC TO 7777,LINK=0
0416 7040 CMA /AC TO 0000,LINK=0
0417 7440 SZA
0420 7402 HLT /CARRY FAILED IN AN EVEN BIT POSITION
0421 7430 SEL
0422 7402 HLT /CARRY OUT FAILED TO COMPLEMENT LINK
0423 7000 XOR06, NOP/JMS POINT FOR XOR
/.....
/TEST AND INSTRUCTION TO NOT SET ANY AC BITS WHEN AC =0000
0424 0072 AND K7777 /AC=0000,LINK=0

```



```

0425 7450 SNA
0426 7430 SEL
0427 7402 HLT /AND SET BIT WHEN AC INITIALLY CLEAR,OR AND SET LINK
/.....
/TEST AND INSTRUCTION TO CLEAR ALL AC BITS WHEN USING AN OPERAND OF 0000
0430 1072 TAD K7777 /AC TO 7777,LINK = 0
0431 0040 AND K0 /AC TO 0000,LINK = 0
0432 7450 SNA
0433 7430 SEL
0434 7402 HLT /AND FAILED TO CLEAR ALL AC BITS, OR SET LINK
/.....
/TEST AND INSTRUCTION TO NOT CLEAR ANY AC BITS WHEN AC=7777 AND MD=7777
0435 1072 TAD K7777 /AC TO 7777
0436 0072 AND K7777 /AC=7777,LINK=0
0437 7040 CMA /AC TO 0000,LINK = 0
0440 7450 SNA
0441 7430 SEL
0442 7402 HLT /AND OF 7777 CLEARED AC BIT OR SET LINK
/.....
/TEST FOR ADJACENT PIN SHORTS IN "AND" CIRCUITRY
0443 1056 TAD K2525 /AC TO 2525
0444 0064 AND K2525 /AC TO 0000
0445 7440 SZA
0446 7402 HLT /"AND" DID NOT CLEAR AC,POSSIBLE ADJACENT PIN SHORTS IN AND CIRCUITRY
0447 7000 XOR07, NOP/JMS POINT FOR XOR
/.....
/TEST ADDER=0 CIRCUITRY
/.....
/TEST BIT 11 INPUT TO ADDER=0
0450 1041 ADDER, TAD K1 /AC TO 0001
0451 7450 SNA
0452 7402 HLT /ADDER=0 OPEN ON BIT 11 INPUT
/.....
/TEST BIT 10 INPUT TO ADDER=0
0453 7200 CLA /AC TO 0000
0454 1042 TAD K2 /AC TO 0002
0455 7450 SNA
0456 7402 HLT /ADDER=0 OPEN ON BIT 10 INPUT
/.....
/TEST BIT 9 INPUT TO ADDER=0
0457 7200 CLA /AC TO 0000
0460 1043 TAD K4 /AC TO 0004
0461 7450 SNA
0462 7402 HLT /ADDER=0 OPEN ON BIT 9 INPUT
/.....
/TEST BIT 8 INPUT TO ADDER=0
0463 7200 CLA /AC TO 0000
0464 1044 TAD K10 /AC TO 0000
0465 7450 SNA
0466 7402 HLT /ADDER=0 OPEN ON BIT 8 INPUT
/.....
/TEST BIT 7 INPUT TO ADDER=0
0467 7200 CLA /AC TO 0000
0470 1045 TAD K20 /AC TO 0020
0471 7450 SNA

```

```

0472 7402 HLT /ADDER=0 OPEN ON BIT 7 INPUT
/.....
/TEST BIT 6 INPUT TO ADDER=0
0473 7200 CLA /AC TO 0000
0474 1046 TAD K40 /AC TO 0000
0475 7450 SNA
0476 7402 HLT /ADDER=0 OPEN ON BIT 6 INPUT
/.....
/TEST BIT 5 INPUT TO ADDER=0
0477 7200 CLA /AC TO 0000
0500 1050 TAD K100 /AC TO 0000
0501 7450 SNA
0502 7402 HLT /ADDER=0 OPEN ON BIT 5 INPUT
0503 7000 XOR08, NOP/JMS POINT FOR XOR
/.....
/TEST BIT 4 INPUT TO ADDER=0
0504 7200 CLA /AC TO 0000
0505 1051 TAD K200 /AC TO 0200
0506 7450 SNA
0507 7402 HLT /ADDER=0 OPEN ON BIT 4 INPUT
/.....
/TEST BIT 3 INPUT TO ADDER=0
0510 7200 CLA /AC TO 0000
0511 1052 TAD K400 /AC TO 0400
0512 7450 SNA
0513 7402 HLT /ADDER=0 OPEN ON BIT 3 INPUT
/.....
/TEST BIT 2 INPUT TO ADDER=0
0514 7200 CLA /AC TO 0000
0515 1131 TAD K1000 /AC TO 1000
0516 7450 SNA
0517 7402 HLT /ADDER=0 OPEN ON BIT 2 INPUT
/.....
/TEST BIT 1 INPUT TO ADDER=0
0520 7200 CLA /AC TO 0000
0521 1055 TAD K2000 /AC TO 0000
0522 7450 SNA
0523 7402 HLT /ADDER=0 OPEN ON BIT 1 INPUT
/.....
/TEST BIT 0 INPUT TO ADDER=0
0524 7200 CLA /AC TO 0000
0525 1061 TAD K4000 /AC TO 4000
0526 7450 SNA
0527 7402 HLT /ADDER=0 OPEN ON BIT 0 INPUT
0530 7000 XOR09, NOP/JMS POINT FOR XOR
/.....
/TEST RAR TO ROTATE AND NOT DROP ANY BITS
0531 7200 RARTST, CLA /AC TO 0000
0532 1056 TAD K2525 /AC TO 2525
0533 7100 CLL
0534 7020 CML /LINK TO 1
0535 7010 RAR /AC TO 5252,LINK=1
0536 7420 SNL
0537 7402 HLT /RAR DID NOT SHIFT AC11 TO LINK
0540 7040 CMA /AC TO 2525,LINK=1

```

```

0541 0064 AND K5252 /AC=0000, LINK=1
0542 7440 SZA
0543 7402 HLT /RAR DROPPED BIT OR DID NOT SHIFT
/ANY BIT SET IN AC INDICATES POSITION OF DROPPED BIT
/.....
/TEST RAR TO NOT PICK UP ANY BITS
0544 1056 TAD K2525 /AC TO 2525, LINK=1
0545 7010 RAR /AC TO 5252, LINK=1
0546 0056 AND K2525 /AC TO 0000, LINK=1
0547 7450 SNA
0550 7420 SNL
0551 7402 HLT /RAR PICKED UP BIT, POSITION OF BIT PICK UP IS
/INDICATED BY BIT(S) SET IN AC
/.....
/TEST RAL TO SHIFT AND NOT DROP ANY BITS
0552 1064 TAD K5252 /AC TO 5252 LINK=1
0553 7004 RAL /AC TO 2525 LINK=1
0554 7420 SNL
0555 7402 HLT /RAL DROPPED LINK BIT
0556 7040 CMA /AC TO 5252 LINK=1
0557 0056 AND K2525 /AC TO 0000 LINK=1
0560 7440 SZA
0561 7402 HLT /RAL DROPPED BIT OR DID NOT SHIFT
/FAILING BIT POSITIONS ARE SET IN AC
/.....
/TEST RAL TO NOT PICK UP ANY BITS
0562 1064 TAD K5252 /AC TO 5252 LINK=1
0563 7004 RAL /AC TO 2525 LINK=1
0564 0064 AND K5252 /AC TO 0000 LINK=1
0565 7440 SZA
0566 7402 HLT /RAL PICKED UP BIT, BITS SET IN AC INDICATE FAILING POSITIONS
/.....
/TEST RTR TO SHIFT TWICE AND NOT DROP ANY BITS
0567 7100 CLLI /CLEAR LINK
0570 1062 TAD K4444 /AC TO 4444 LINK=0
0571 7012 RTR /AC TO 1111 LINK=0
0572 7430 SEL
0573 7402 HLT /RTR PICKED UP LINK BIT
0574 7040 CMA /AC TO 0606
0575 0053 AND K1111 /AC TO 0000 LINK=0
0576 7440 SZA
0577 7402 HLT /RTR DROPPED BIT OR DID NOT SHIFT TWICE
/BIT SET IN AC INDICATES BIT DROPPED
0600 7000 XOR10, NOP/JMS POINT FOR XOR
/.....
/TEST RTR TO NOT PICK UP ANY BITS
0601 1062 RTRPCK, TAD K4444 /AC TO 4444
0602 7012 RTR /AC TO 1111
0603 0066 AND K6666 /AC TO 0000
0604 7440 SZA
0605 7402 HLT /RTR PICKED UP BIT, BIT SET IN AC INDICATES FAILING POSITION.
/.....
/TEST RTL TO SHIFT AND NOT DROP BITS
0606 1053 RTLDRP, TAD K1111
0607 7006 RTL /AC TO 4444 LINK=0

```

```

0610 7430 SEL
0611 7402 HLT /RTL PICKED UP LINK BIT
0612 7040 CMA /AC TO 3333 LINK=0
0613 0062 AND K4444 /AC TO 0000 LINK=0
0614 7440 SZA
0615 7402 HLT /RTL DROPPED BIT OR DID NOT SHIFT
/BIT SET IN AC INDICATES BIT DROPPED
/.....
/TEST RTL TO NOT PICK UP ANY BITS
0616 1053 RTLDRP, TAD K1111 /AC TO 1111
0617 7006 RTL /AC TO 4444
0620 0060 AND K3333 /AC TO 0000
0621 7440 SZA
0622 7402 HLT /RTL PICKED UP BIT, BIT SET IN AC INDICATES FAILING POSITION
/.....
/TEST BSW TO SWAP AND NOT DROP BITS ON CHANGE LINK
0623 7100 BSWDRP, CLLI /INITIALIZE LINK TO ZERO
0624 1047 TAD K77 /AC TO 0077 LINK=0
0625 7002 BSW /AC TO 7700 LINK=0
0626 7430 SEL
0627 7402 HLT /BSW SET LINK
0630 1050 TAD K100 /AC TO 0000 LINK TO 1
0631 7440 SZA
0632 7402 HLT /BSW DID NOT SWAP OR PICKED UP BIT.
/.....
/TEST BSW FOR ADJACENT PIN SHORTS AND DROPPED BITS
0633 1067 BSWPCK, TAD K7700 /AC TO 7700 LINK=1
0634 7002 BSW /AC TO 0077 LINK=1
0635 7420 SNL
0636 7402 HLT /BSW CLEARED LINK
0637 7040 CMA /AC TO 7700
0640 1050 TAD K100 /AC TO 0000, LINK TO 0
0641 7440 SZA
0642 7402 HLT /BSW PICKED UP OR DROPPED BIT(S),
0643 7000 XOR11, NOP/JMS POINT FOR XOR
/.....
/TEST OF GROUP 3 OPERATES AND MQ REGISTER FOLLOWS
/.....
/TEST MQL TO CLEAR AC
0644 1072 MQLTST, TAD K7777 /AC TO 7777
0645 7421 MQL /AC TO MQ, 0 TO AC.
0646 7440 SZA
0647 7402 HLT /MQL DID NOT CLEAR AC
/.....
/TEST SWP TO EXCHANGE MQ AND AC, TEST RESULTS OF PREVIOUS MQL,
0650 7521 SWPST, SWP /AC TO MQ, MQ TO AC, AC=7777 MQ=0000
0651 7040 CMA /AC TO 0000
0652 7440 SZA
0653 7402 HLT /SWP DID NOT LOAD AC WITH MQ, OR MQL DID NOT LOAD MQ, OR
0654 7040 CMA /SWP OR MQL DROPPED A BIT,
/AC TO 7777 MQ=0000
/.....
/TEST MQ REGISTER FOR OUTPUTS STUCK HIGH
0655 7521 SWP /AC TO 0000 MQ TO 7777
0656 7440 SZA

```

```

0657 7402      HLT          /SWP PICKED UP BIT IN AC,
/*****
/TEST MQ FOR ADJACENT PIN SHORTS BY TESTING FOR DROPPED BITS
0660 1056      TAD          K2525   /AC TO 2525 MQ=7777
0661 7421      MQL          /AC TO 0000 MQ TO 2525
0662 7521      SWP          /AC TO 2525 MQ TO 0000
0663 7040      CMA          /AC TO 9252
0664 0056      AND          K2525   /AC TO 0000
0665 7440      SEA
0666 7402      HLT          /MQL OR SWP DROPPED BIT, POSSIBLE ADJACENT PIN SHORTS
/IN MQ, BIT SET IN AC INDICATES POSITION OF FAILURE
/*****
/TEST FOR ADJACENT PIN SHORTS IN MQ BY TESTING FOR BITS PICKED UP
XOR13, NOP/JMS POINT FOR XOR
0667 7000      TAD          K2525   /AC TO 2525 MQ=0000
0670 1056      MQL          /AC TO 0000 MQ TO 2525
0671 7421      SWP          /AC TO 2525 MQ TO 0000
0672 7521      AND          K2525   /AC TO 0000
0673 0064      SEA
0674 7440      HLT          /MQL OR SWP PICKED UP BIT, POSSIBLE ADJACENT PIN SHORT IN MQ,
/BIT SET IN AC INDICATES POSITION OF FAILURE,
0676 7100      CLL
0677 7040      CMA          /AC TO 7777
/*****
/TEST MQA TO OR THE MQ WITH THE AC. ENTER WITH AC=7777 MQ=0000 LINK=0
0700 7421      MQL          /AC TO 0000 MQ TO 7777
0701 7501      MQA          /AC TO 7777 MQ=7777
0702 7430      SEL
0703 7402      HLT          /MQA SET LINK
0704 7040      CMA          /AC TO 0000 MQ=7777
0705 7440      SEA
0706 7402      HLT          /MQA DID NOT OR MQ WITH AC
0707 7521      SWP          /AC TO 7777 MQ TO 0000
0710 7450      SNA
0711 7402      HLT          /MQA CLEARED MQ
/*****
/TEST CAM TO CLEAR AC AND MQ
CAHTST, SWP          /AC TO 0000 MQ TO 7777
0713 7040      CMA          /AC TO 7777 MQ=7777
0714 7020      CML          /SET LINK
0715 7621      CAM          /CLEAR AC AND MQ
0716 7440      SEA
0717 7402      HLT          /CAM DID NOT CLEAR AC
0720 7420      SNL
0721 7402      HLT          /CAM CLEARED LINK
0722 7521      SWP          /AC=0000 MQ=0000
0723 7440      SEA
0724 7402      HLT          /CAM DID NOT CLEAR MQ
/*****
/TEST ACL TO LOAD ZEROES FROM MQ TO AC
ACLTST, CMA          /AC TO 7777 MQ=0000
0726 7701      ACL          /AC TO 0000 MQ=0000
0727 7440      SEA
0730 7402      HLT          /ACL DID NOT LOAD 0000 TO AC
0731 7420      SNL

```

```

0732 7402      HLT          /ACL CLEARED LINK
0733 7040      CMA          /AC TO 7777
0734 7521      SWP          /AC TO 0000 MQ TO 7777
0735 7440      SEA
0736 7402      HLT          /ACL CHANGED MQ
/*****
/TEST OF DCA AND ISZ, DIRECT ADDRESSING TO PAGE ZERO FOLLOWS
/*****
/TEST DCA TO STORE ALL 1'S, CLEAR AC, AND NOT AFFECT LINK
DCAZTS, CML          /SET LINK
0740 1072      TAD          K7777   /AC TO 7777 LINK=0
0741 3112      DCA          TESLOC /AC TO 0000 LINK=0
0742 7440      SEA
0743 7402      HLT          /DCA DID NOT CLEAR AC, OR DCA SKIPPED
0744 7430      SEL
0745 7402      HLT          /DCA SET LINK
0746 1112      TAD          TESLOC /AC TO 7777
0747 7040      CMA
0750 7440      SEA
0751 7402      HLT          /DCA DID NOT STORE, OR DCA STORED TO WRONG ADDRESS
/*****
/TEST DCA TO STORE ALL ZEROES
XOR15, NOP/JMS POINT
0752 7000      DCA          TESLOC /0000 TO LOCATION "TESLOC"
0753 3112      TAD          TESLOC /0000 TO AC
0754 1112
0755 7440      SEA
0756 7402      HLT          /DCA DID NOT STORE ALL 0'S, BIT SET IN AC
/INDICATES FAILING BIT POSITION,
/*****
/TEST ISZ TO INCREMENT WITHOUT SKIPPING
ISZT9, ISZ          TESLOC /LOCATION "TESLOC" TO 0001
0760 7440      SEA
0761 7402      HLT          /ISZ SKIPPED WHEN NO OVERFLOW, OR ISZ SET AC BIT
0762 7430      SEL
0763 7402      HLT          /ISZ SET LINK
0764 1112      TAD          TESLOC /AC TO 0001 LINK=0
0765 1072      TAD          K7777   /AC TO 0000, LINK TO 1
0766 7440      SEA
0767 7402      HLT          /ISZ DID NOT INCREMENT BY 1
/*****
/TEST ISZ TO SKIP ON OVERFLOW
0770 1072      TAD          K7777   /AC TO 7777 LINK=1
0771 3112      DCA          TESLOC /LOCATION "TESLOC" TO 7777, AC TO 0000
0772 2112      ISZ          TESLOC /SHOULD SKIP
0773 7402      HLT          /ISZ DID NOT SKIP ON OVERFLOW
0774 7420      SNL
0775 7402      HLT          /OVERFLOW ON ISZ SET LINK
0776 1112      TAD          TESLOC /0000 TO AC
0777 7440      SEA
1000 7402      HLT          /ISZ DID NOT INCREMENT 7777 TO 0000, OR ISZ
/AFFECTED AC ON OVERFLOW
/*****
/TEST OF "AND", TAD, ISZ, AND DCA, DIRECT ADDRESSING TO SAME PAGE
/*****
/TEST TAD TO ADDRESS SAME PAGE DIRECT

```

```

1001 1203 TODSTS, TAD ,+2 /AC TO 2526 LINK=0
1002 7410 SKP
1003 2526 2526 /OPERAND FOR TAD SAME PAGE TEST
1004 1064 TAD K5252 /AC TO 0000, LINK TO 1
1005 7440 SZA
1006 7402 HLT /TAD TO SAME PAGE DIRECT FAILED
/.....
/TEST DCA TO SAME PAGE DIRECT
1007 1072 DODSTS, TAD K7777 /AC TO 7777
1010 3212 DCA ,+2 /AC TO 0000
1011 7410 SKP
1012 0000 0000 /TEST LOCATION FOR DCA TO SAME PAGE TEST
1013 1212 TAD ,+1 /AC TO 7777
1014 7040 CMA /AC TO 0000
1015 7440 SZA
1016 7402 HLT /DCA TO SAME PAGE FAILED
1017 3212 DCA ,+5 /CLEAR TEST LOCATION FOR POSSIBLE SECOND PASS
/.....
/TEST ISZ TO SAME PAGE DIRECT TO SKIP
1020 7000 XOR16, NOP/JMS POINT FOR XOR
1021 1072 ISDSTS, TAD K7777 /AC TO 7777
1022 3224 DCA ,+2 /AC TO 0000
1023 7410 SKP
1024 0000 0000 /ISZ TEST LOCATION
1025 2224 ISZ ,+1 /SHOULD SKIP
1026 7402 HLT /ISZ DID NOT SKIP
/.....
/TEST ISZ TO SAME PAGE DIRECT TO NOT SKIP
1027 2224 ISZ ,+3 /SHOULD NOT SKIP
1030 7410 SKP
1031 7402 HLT /ISZ SKIPPED WHEN NO SKIP EXPECTED
/.....
/TEST "AND" TO SAME PAGE DIRECT TO CLEAR ALL AC BITS
1032 1064 ANDSTS, TAD K5252 /AC TO 2526
1033 0235 AND ,+2 /AC TO 0000
1034 7410 SKP
1035 2525 2525 /OPERAND FOR AND SAME PAGE TEST
1036 7440 SZA
1037 7402 HLT /AND TO SAME PAGE DIRECT FAILED
/.....
/TESTS OF TAD, "AND", DCA AND ISZ INDIRECT THRU PAGE ZERO FOLLOW
/.....
/TEST OF TAD THRU PAGE ZERO INDIRECT
1040 1513 TDISTS, TAD I POINTR /AC TO 2526
1041 1064 TAD K5252 /AC TO 0000
1042 7440 SZA
1043 7402 HLT /TAD INDIRECT THRU PAGE ZERO FAILED, OR AUTO-INDEXED,
/.....
/TEST DCA INDIRECT THRU PAGE ZERO
1044 1072 DCISTS, TAD K7777 /AC TO 7777
1045 3515 DCA I POINTB /7777 TO LOCATION "POINTC", AC TO 0000
1046 1116 TAD POINTC /AC TO 7777
1047 7040 CMA /AC TO 0000
1050 7440 SZA
1051 7402 HLT /DCA INDIRECT THRU PAGE ZERO FAILED

```

```

/.....
/TEST ISZ INDIRECT THRU PAGE ZERO
1052 2515 ISISTS, ISZ I POINTB /LOCATION "POINTC" TO 0000, SKIP
1053 7402 HLT /ISZ INDIRECT THRU PAGE ZERO FAILED TO SKIP
/.....
/TEST "AND" INDIRECT THRU PAGE ZERO
1054 1056 ANISTS, TAD K2525 /AC TO 2525
1055 3116 DCA POINTC /AC TO 0000
1056 7000 XOR17, NOP/JMS POINT
1057 1072 TAD K7777 /AC TO 7777
1060 0515 AND I POINTB /AC TO 2525
1061 1065 TAD K5253 /AC TO 0000
1062 7440 SZA
1063 7402 HLT /AND INDIRECT THRU PAGE ZERO FAILED
/.....
/TESTS OF TAD, DCA, ISZ, & "AND" INDIRECT THRU SAME PAGE FOLLOW
/.....
/TEST TAD INDIRECT THRU SAME PAGE
1064 1666 TDISTS, TAD I ,+2 /AC TO 2526
1065 7410 SKP
1066 0114 POINTR+1 /POINTER FOR TAD INDIRECT THRU SAME PAGE
1067 1064 TAD K5252 /AC TO 0000
1070 7440 SZA
1071 7402 HLT /TAD INDIRECT THRU SAME PAGE FAILED
/.....
/TEST DCA INDIRECT THRU SAME PAGE
1072 1072 DCISTS, TAD K7777 /AC TO 7777
1073 3675 DCA I ,+2 /AC TO 0000
1074 7410 SKP
1075 0116 POINTC /POINTER FOR DCA INDIRECT THRU SAME PAGE
1076 1116 TAD POINTC /AC TO 7777
1077 7040 CMA
1100 7440 SZA
1101 7402 HLT /DCA INDIRECT THRU SAME PAGE FAILED
/.....
/TEST ISZ INDIRECT THRU SAME PAGE
1102 2705 ISISTS, ISZ I ,+3 /LOCATION "POINTC" TO 0000, SKIP
1103 7402 HLT /ISZ INDIRECT THRU SAME PAGE FAILED TO SKIP
1104 7410 SKP
1105 0116 POINTC
/.....
/TEST "AND" INDIRECT THRU SAME PAGE
1106 1056 ANISTS, TAD K2525 /AC TO 2525
1107 3116 DCA POINTC /AC TO 0000
1110 1072 TAD K7777 /AC TO 7777
1111 0713 AND I ,+2 /AC TO 2525
1112 7410 SKP
1113 0116 POINTC /POINTER FOR "AND" INDIRECT THRU SAME PAGE
1114 1065 TAD K5253 /AC TO 0000
1115 7440 SZA
1116 7402 HLT /"AND" INDIRECT THRU SAME PAGE FAILED
1117 7000 XOR18, NOP/JMS POINT
/.....
/TESTS OF AUTO-INDEX FOLLOW
/.....

```

```

1120 1056 /TEST AUTO-INDEX TO NOT INCREMENT WHEN NOT INDIRECTLY ADDRESSED.
1121 3010 AINITS, TAD K2525 /AC TO 2525
1122 1010 DCA 10 /ADDRESS 10 TO 2525
1123 1065 TAD 10 /AC TO 2525
1124 7440 TAD K2523 /AC TO 0000
1125 7402 SEA
HLT /AUTO-INDEX INCREMENTED WHEN NOT INDIRECTLY ADDRESSED
/.....
1126 3116 /TEST AUTO-INDEX TO INCREMENT WHEN INDIRECTLY ADDRESSED
1127 1056 AIIOTS, DCA POINTC /CLEAR LOCATION "POINTC"
1130 3117 TAD K2525 /AC TO 2525
1131 1115 DCA POINTD /LOCATION "POINTD" TO 2525
1132 3010 TAD POINTB
1133 1410 DCA 10 /SET LOCATION 10 TO THE ADDRESS OF "POINTC"
1134 1065 TAD I 10 /LOCATION "POINTD"'S CONTENTS TO AC, AC TO 2525
1135 7440 TAD K2523 /AC TO 0000
1136 7402 SEA
HLT /AUTO-INDEX FAILED TO INCREMENT
/.....
1137 7000 /TEST AUTO-INDEX DECODER FROM BITS 0 THRU 3 BY ADDRESSING ADDRESS 1010
1140 1210 /INDIRECTLY AND INSURING THAT AUTO-INDEXING DID NOT TAKE PLACE.
1141 3112 XOR20, NOP/JMS POINT
1142 0610 AIBIT0, TAD 1010 /GET INITIAL CONTENTS OF 1010
1143 1210 DCA TESLOC /SAVE FOR LATER COMPARISON
1144 7041 AND I 1010 /REFERENCE 1010 INDIRECTLY
1145 1112 TAD 1010 /GET CONTENTS OF 1010
1146 7440 CIA /NEGATE IT
1147 7402 TAD TESLOC /COMPARE TO INITIAL CONTENTS
SEA /FINAL=INITIAL?
HLT /NO, AUTO INDEX OCCURRED FOR LOCATION 1010
/AUTO-INDEX DECODER OPEN ON INPUT FROM BITS 0-3
/.....
1150 3007 /TEST BIT 0 INPUT OF AUTO-INDEX DECODER BY ADDRESSING LOCATION 0007
1151 1407 /INDIRECTLY AND CHECKING THAT AUTO-INDEXING DID NOT OCCUR
1152 7200 AIBIT0, DCA 7 /CLEAR LOCATION 0007
1153 1007 TAD I 7 /REFERENCE LOCATION 0007 INDIRECTLY
1154 7440 CLA /CLEAR AC
1155 7402 TAD 7 /GET CONTENTS OF LOCATION 0007
SEA /DOES IT STILL CONTAIN 0007?
HLT /NO, LOCATION 0007 AUTO-INDEXED
/.....
1156 1030 /TEST AUTO-INDEX DECODER INPUT FROM BITS 4 THRU 7, BY REFERENCING
1157 3112 /ADDRESS 0030 INDIRECTLY AND CHECKING THAT AUTO-INDEXING DID NOT OCCUR
1160 0430 AIBIT4, TAD 30 /GET INITIAL CONTENTS OF LOC 0030
1161 1030 DCA TESLOC /SAVE IT FOR COMPARISON
1162 7040 AND I 30 /REFERENCE 30 INDIRECTLY
1163 7001 TAD 30 /GET FINAL CONTENTS OF LOC 30
1164 1112 CMA /COMPLEMENT IT FOR COMPARE
1165 7440 JAC TESLOC /COMPARE TO INITIAL CONTENTS
1166 7402 TAD TESLOC /WAS LOC 30 AUTO-INDEXED?
1167 7000 SEA /YES, LOC 30 AUTO-INDEXED
HLT
XOR19, NOP/JMS POINT FOR XOR
/.....
/TESTS OF INTERNAL IOT INSTRUCTIONS FOLLOW

```

```

1170 7100 /.....
1171 7020 /TEST GTF TO SAVE LINK, AND READ ZERO FOR THE GT FLAG
1172 6004 GTFTS1, CLL /CLEAR LINK
1173 7500 CML /LINK TO 1
1174 7402 GTF
1175 7004 SMA /GTF DID NOT SAVE A 1 FOR LINK
1176 7510 HLT
1177 7402 RAL
1178 7004 SPA
1179 7402 HLT /GTF READ A 1 FOR GT FLAG, NO GT FLAG EXISTS IN B/A
1200 7300 CLA CLL
1201 6214 RDF /GET DATA FIELD
1202 7012 RTR
1203 7010 RAR /MOVE DF TO AC 9-11
1204 6224 RIF /GET INSTRUCTION FIELD
1205 3106 DCA SAVFLD /SAVE IF AND DF FOR EXTENDED FIELD INTERRUPT PROCESSING
1206 7000 XOR20, NOP/JMS POINT
/.....
1207 1061 /TEST RTF TO RESTORE LINK
1210 1106 RTFTS1, TAD K4000 /AC TO 4000
1211 6005 TAD SAVFLD /GET CORRECT IF AND DF INFORMATION
1212 5213 RTF /RESTORE LINK TO 1
1213 7420 JMP ,+1 /ENABLE INTERRUPT FOR POWER FAIL
1214 7402 SNL
HLT /RTF DID NOT RESTORE LINK TO A 1
/.....
1215 6002 /TEST GTF TO SAVE A LINK OF 0, AND INT REQUEST AND INT ENABLE OF 0
1216 7300 GTFTS2, IOF
1217 6004 CLA CLL /CLEAR AC AND LINK
1220 0366 GTF /GET FLAGS, LINK TO AC0
1221 7440 AND K7600 /ELIMINATE SAVE FIELD REGISTER AND USER BIT
1222 7402 SEA /LINK, INT REQUEST, AND INT ENABLE ALL ZERO?
HLT /NO, GTF DID NOT SAVE CORRECTLY,
/IF BIT0=1 LINK WAS SAVED AS 1 INSTEAD OF 0
/IF BIT2=1 INT REQUEST WAS SAVED AS 1 INSTEAD OF 0
/IF BIT4=1 INT ENABLE WAS SAVED AS 1 INSTEAD OF 0
/.....
1223 6006 /TEST SKIP ON GT FLAG TO NOT SKIP
1224 7410 SGTST, SGT /SKIP ON GT FLAG(DOESN'T EXIST IN B/A)
1225 7402 SKP
HLT /SGT SKIPPED WHEN NO GT FLAG EXISTS
/.....
1226 6001 /TEST CAF TO CLEAR AC AND LINK
1227 7040 ION
1230 7020 CMA /AC TO 7777
1231 6007 CML /AC=7777 LINK TO 1
1232 7450 CAF /CLEAR ALL FLAGS, CLEARS AC AND LINK, AND INT ENABLE
1233 7430 SNA
1234 7402 SZL
HLT /CAF FAILED TO CLEAR AC OR LINK
/.....
1235 6000 /TEST SKON TO NOT SKIP WHEN INTERRUPT ENABLE IS CLEAR
1236 7410 SKON
1237 7402 SKP /SHOULD NOT SKIP
HLT /SKON SKIPPED WHEN INT ENABLE CLEAR, OR CAF FAILED

```

```

.....
/TO CLEAR INT ENABLE
/TEST SKON TO SKIP ON INTERRUPT ON AND TURN INTERRUPT OFF
1240 6001 IOFTS1, ION /INTERRUPT ON.
1241 7000 XOR21, NOP/JMS POINT FOR XOR
1242 6000 SKON /SKIP IF INTERRUPT ON,TURN INTERRUPT OFF
1243 7402 HLT /ION DID NOT ENABLE INTERRUPT, OR SKON FAILED TO SKIP
.....
/TEST THAT SKON TURNED OFF INTERRUPT
1244 6000 SKONT2, SKON /SHOULD NOT SKIP
1245 7410 SKP
1246 7402 HLT /SKON DID NOT TURN OFF INT, OR SKON SKIPS WHEN INT OFF
.....
/TEST IOF TO DISABLE INTERRUPTS
1247 6001 IOFTS1, ION /ENABLE INTERRUPTS
1250 7000 NOP /ALLOWS TIME FOR INTERRUPT DELAY TO SET
1251 6002 IOF /TURN OFF INTERRUPT
1252 6000 SKON
1253 7410 SKP
1254 7402 HLT /IOF DID NOT DISABLE INTERRUPT
.....
/TEST PROPER OPERATION OF INT ENABLE.
1255 6001 INTEN1, ION /INTERRUPT ON
1256 6004 GTF /SHOULD GET A 1 FOR INT ENABLE, AND A 1 FOR INT DELAY
1257 0001 AND K200 /MASK OUT INT ENABLE BIT
1260 7450 SNA /SKIP IF INT ENABLE BIT SET
1261 7402 HLT /ION DID NOT SET INT ENABLE OR GTF DID NOT GET A 1 FOR INT ENABLE
.....
/TEST RTF TO SET INT ENABLE AND TO CLEAR LINK
1262 6002 RTFTS2, IOF /CLEAR INT ENABLE
1263 7300 CLA CLL
1264 7020 CML /SET LINK
1265 1106 TAD SAVFLD /GET IF AND OF INFORMATION
1266 6005 RTF /RESTORE FLAGS, LINK TO 0, SET INTERRUPT ENABLE
1267 5270 JMP ,+1 /ENABLE INTERRUPT FOR POWER FAIL USE
1270 7430 SZA
1271 7402 HLT /RTF DID NOT CLEAR LINK
1272 6000 SKON /SKIP IF INTERRUPT ON
1273 7402 HLT /RTF FAILED TO ENABLE INTERRUPTS
.....
/TEST SRQ TO NOT SKIP WHEN NO INTERRUPT REQUEST
1274 6001 SRQTS1, ION /INTERRUPT ON
1275 7300 CLA CLL /GIVE POWER FAIL A CHANCE TO INTERRUPT
1276 6003 SRQ /SKIP ON INTERRUPT REQUEST
1277 7410 SKP
1300 7402 HLT /SRQ SKIPPED WHEN NO INTERRUPT, OR ILLEGAL INTERRUPT
/CAME UP DURING SRQ INSTRUCTION.
.....
/TESTS OF JUMPS AND JMS'S FOLLOW
/TEST JUMP DIRECT
1301 7000 JMPTS1, NOP
1302 7300 CLA CLL /CLEAR AC AND LINK
1303 1135 TAD KSTOP
1304 3000 DCA 0 /PUT HLT IN LOC 0 IN CASE JUMP FAILS TO GATE MD TO PC

```

```

1305 5307 JMP ,+2 /FIRST JUMP TESTED
1306 7402 HLT /JMP FAILED TO JUMP DIRECT
.....
/TEST JMS DIRECT
1307 4311 JMSTS1, JMS ,+2
1310 7402 HLT /JMS FAILED TO JUMP
1311 0000 JMSLOC, 0 /JMS ENTRY POINT, PC STORAGE.
1312 1311 TAD ,.-1 /GET STORED PC
1313 7040 CMA
1314 7001 IAC /COMPLEMENT IT
1315 1120 TAD POINTE
1316 7440 SZA
1317 7402 HLT /JMS DID NOT STORE PROPER PC
.....
/TEST JUMP INDIRECT TO JUMP CORRECTLY
1320 5722 JMPTS2, JMP I ,+2
1321 7402 HLT /JMP INDIRECT FAILED TO JUMP
1322 1323 ,+1 /POINTER FOR JMP INDIRECT ABOVE
.....
/TEST JMS INDIRECT TO JUMP AND STORE PC
1323 4725 JMSTS2, JMS I ,+2
1324 7402 HLT /JMS INDIRECT FAILED TO JUMP
1325 1326 ,+1
1326 0000 JMSLO2, 0 /JMS INDIRECT PC STORAGE
1327 1326 TAD ,.-1
1330 7040 CMA
1331 7001 IAC /COMPLEMENT STORED PC
1332 1121 TAD POINTE /COMPARE IT TO EXPECTED VALUE
1333 7440 SZA /WAS IT EQUAL?
1334 7402 HLT /NO, JMS INDIRECT STORED INCORRECT PC.
1335 7000 XOR22, NOP/JMS POINT FOR XOR
.....
/TESTS OF ROMS D AND F FOLLOW
/.....
/THE ROMS ARE TESTED BY EXECUTING EVERY INSTRUCTION THAT CAUSES A UNIQUE
ADDRESS INPUT TO THESE ROMS, FOR EACH OF THE INSTRUCTIONS
EXECUTED, A TOTAL OF 8 PATTERNS OF AC, MD, AND LINK CONTENTS ARE USED,
/TO TEST FOR ERRORS IN THE EXECUTED INSTRUCTION A SIMULATION OF THE
INSTRUCTION IS ALSO PERFORMED, AND THE RESULTS OF THE INSTRUCTION ARE
/COMPARED TO THE SIMULATION, BARRING INTERMITTENT ERRORS, ALL FAILURES
/WILL BE IN THE EXECUTED INSTRUCTION, AND WILL BE DUE TO A ROM D OR F FAILURE.
/.....
/CREATE THE INSTRUCTION TO BE USED. INSTRUCTION RANGE IS 7XX0, 7XX1.
.....
1336 3116 ROMTST, DCA POINTC /INITIALIZE INSTRUCTION MAKER
1337 7410 SKP
1340 7300 NXTONE, CLA CLL
1341 2116 ISZ POINTC
1342 7020 CML
1343 7000 XOR12, NOP/JMS POINT
1344 1116 TAD POINTC
1345 1067 TAD K7700
1346 7500 SNA
1347 5524 JMP I CKSWIT /ALL COMBINATIONS TRIED?
1350 7300 CLA CLL /YES, TEST OVER, CHECK FOR I/O SIMULATOR,

```

```

1351 1116 TAD POINTC
1352 7000 RTL
1353 7004 RAL
1354 1034 TAD K7000 /MAKE NEXT INSTRUCTION
1355 3372 DCA INSTRT /SAVE IT.
1356 1071 TESAGN, TAD K7770
1357 3177 DCA DATPAT

```

```

/.....
/SET UP AC, MQ, AND LINK FOR EXECUTION OF TEST INSTRUCTION
NXTPAT, CAM
XOR23, NOP/JMS POINT FOR XOR
1360 7621 DCA SKPPED
1361 7000 DCA SOMSKP
1362 3104 TAD LKDATA
1363 3105 RAR /LOAD LINK
1364 1111 K7000, 7000/CLA
1365 7010 TAD MQDATA
1366 7600 SWP /LOAD MQ
1367 1110 TAD ACDATA /LOAD AC
1370 7521
1371 1107

```

```

/.....
/EXECUTE TEST INSTRUCTION
INSTRT, 0 /EXECUTE INSTRUCTION
1372 0000 /DONT SKIP
1373 2104 /SE SKPPED

```

```

/.....
/SAVE RESULTS OF TEST INSTRUCTION
DCA ACWAS
1374 3101 CMA
1375 7040 SNL
1376 7420 CLA
1377 7200 DCA LKWAS
1400 3103 SWP
1401 7521 DCA MQWAS
1402 3102

```

```

/.....
/SET UP FOR SIMULATED EXECUTION TO TEST RESULT
1403 7040 CMA
1404 3112 DCA BIT11
1405 7040 CMA
1406 3006 DCA BIT0
1407 7040 CMA
1410 3007 DCA BIT7
1411 7040 CMA
1412 3117 DCA BIT6
1413 7040 CMA
1414 3005 DCA BIT5
1415 7040 CMA
1416 3004 DCA BIT4
1417 7040 CMA
1420 3003 DCA BIT3
1421 1522 SETSIM, TAD I INSTRT /GET INSTRUCTION
1422 7006 RTL
1423 7006 RTL
1424 7430 SEL /WAS BIT 3 SET?
1425 2003 /YES, CLEAR POINTER
1426 7000 NOP
1427 7004 RAL

```

```

1430 7430 SEL /BIT 4 SET?
1431 2004 /YES, CLEAR POINTER
1432 7000 NOP
1433 7004 RAL
1434 7430 SEL /BIT 5 SET?
1435 2005 /YES, CLEAR POINTER
1436 7000 NOP
1437 7004 RAL
1440 7430 SEL /BIT 6 SET?
1441 2117 /YES, CLEAR POINTER
1442 7000 NOP
1443 7004 RAL
1444 7430 SEL /BIT 7 SET?
1445 2007 /YES, CLEAR POINTER
1446 7000 NOP
1447 7004 RAL
1450 7430 SEL /BIT 8 SET ?
1451 2006 /YES, CLEAR POINTER
1452 7000 NOP
1453 7006 RTL
1454 7004 RAL
1455 7430 SEL /WAS BIT 11 SET?
1456 2112 /YES, CLEAR POINTER
1457 7000 NOP
1460 7200 CLA

```

```

/.....
/SET UP AC, MQ, AND LINK FOR SIMULATED EXECUTION
DOSIMU, TAD LKDATA
1461 1111 RAR
1462 7010 CLA
1463 7200 TAD MQDATA
1464 1110 SWP /LOAD LINK
1465 7521 TAD ACDATA /LOAD AC
1466 1107

```

```

/.....
/SIMULATED EXECUTION BEGINS
/.....
/GROUP 1 SIMULATION
GR1SIM, /SE BIT3 /WAS BIT 3 SET IN THE INSTRUCTION?
1467 2003 /YES, IT IS A GROUP 2 OR 3.
1470 5307 JMP GR2GR3 /WAS BIT 4 SET?(GROUP 1)
1471 2004 /YES, SO CLEAR THE AC
1472 7200 CLA /WAS IT CLL?
1473 2005 /YES.
1474 7100 CLL /YES
1475 2117 /WAS IT CMA?
1476 7040 CMA /YES
1477 2007 /WAS IT CML?
1500 7020 CML /YES
1501 2112 /WAS IT IAC?
1502 1041 TAD K1 /YES, ADD 1
1503 2006 /WAS IT RAR?
1504 7010 RAR /YES
1505 2104 /SE SKPPED /INDICATE NO SKIP
1506 5523 JMP I TEST /GO CHECK RESULTS

```





```

1634 7521 SWP /PUT IT IN MQ
1635 1111 TAD LKDATA /GET LINK DATA
1636 7010 RAR /LOAD LINK
1637 7200 CLA
1640 1107 TAD ACDATA /LOAD AC DATA
1641 7402 HLT /AC, LINK, AND MQ ARE AS THEY WERE WHEN INSTRUCTION WAS EXECUTED.
/DEPRESS CONTINUE TO EXECUTE FAILING INSTRUCTION OVER.
1642 5522 JMP I INSTTR /GO EXECUTE INSTRUCTION AGAIN
/TEST FOR ANY SIMULATION DATA ERRORS
1643 1003 SIMERR, TAD BIT3 /GET ERROR INDICATOR
1644 7450 SNA /ANY SIMULATION ERRORS?
1645 5271 JMP TESTPT /NO
/.....
/DISPLAY THE ERROR INFORMATION
1646 7200 DISERR, CLA /ERRORS EXIST
1647 1522 TAD I INSTTR /GET FAILING INSTRUCTION
1650 7402 HLT /OPERATE INSTRUCTION FAILED, FAILING INSTRUCTION
/IS IN THE AC. DEPRESS CONTINUE FOR EXPECTED DATA,
/BARRING INTERMITTENTS, ROM 0 OR F AT FAULT.
/.....
/GET EXPECTED DATA FOR AC, MQ, AND LINK
1651 7300 GETEXP, CLA CLL
1652 1007 TAD BIT7 /GET EXPECTED LINK
1653 7010 RAR
1654 1006 TAD BIT8
1655 7421 MQL /LOAD EXPECTED MQ
1656 1112 TAD BIT11 /LOAD EXPECTED AC
1657 7402 HLT /AC, MQ, AND LINK CONTAIN EXPECTED DATA
/DEPRESS CONTINUE FOR DATA FOUND
/.....
/GET DATA FOUND
1660 7300 GETFND, CLA CLL
1661 1103 TAD LKMAS
1662 7010 RAR
1663 7200 CLA
1664 1102 TAD MQMAS
1665 7421 MQL /LOAD MQ AS IT WAS FOUND
1666 1101 TAD ACWAS /LOAD AC AS IT WAS FOUND
1667 7402 HLT /AC, MQ, AND LINK ARE AS THEY WERE FOUND
/DEPRESS CONTINUE TO EXECUTE SAME INSTRUCTION OVER
/DO SAME INSTRUCTION OVER FOR ERROR
1670 5742 DOAGAN, JMP I NXTPTT /JMP TO NXTPAT
/.....
/SIMULATION AGREES WITH ACTUAL, SEE IF ALL DATA PATTERNS HAVE
/BEEN TRIED WITH THIS INSTRUCTION,
1671 2177 TESTPT, ISE DATPAT /ALL PATTERNS TRIED?
1672 5314 JMP TRNXPT /NO, TRY NEXT PATTERN

```

```

1673 3107 DCA ACDATA
1674 7000 XOR14, NOP/JMS POINT
1675 3110 DCA MQDATA
1676 3111 DCA LKDATA
1677 1522 TAD I INSTTR /GET INSTRUCTION
1700 7010 RAR /BIT11 TO LINK
1701 7430 SEL /HAS INSTRUCTION BEEN TRIED WITH BIT11=1?
1702 5743 JMP I NXTONN /YES, DO NEXT INSTRUCTION (JMP TO NXTONE)
1703 2522 ISE I INSTTR /UPDATE INSTRUCTION
1704 0051 AND K200 /MASK OUT MQ TYPE BIT
1705 7450 SNA /MQ TYPE?
1706 5744 JMP I TESAGG /NO, GO DO IT
1707 7200 CLA
1710 1522 TAD I INSTTR
1711 0070 AND K7721 /MASK OUT BITS NOT ALLOWED(EAE)
1712 3522 DCA I INSTTR
1713 5744 JMP I TESAGG /JMP TO TESAGN
/.....
/CREATE NEXT DATA SET UP
1714 7000 TRNXPT, NOP/JMS POINT
1715 1177 TAD DATPAT
1716 7010 RAR /AC BIT TO LINK
1717 7200 CLA
1720 1056 TAD K2525
1721 7420 SNL /AC TO BE SET?
1722 7200 CLA /NO
1723 3107 DCA ACDATA
1724 1177 TAD DATPAT
1725 7012 RTR
1726 7200 CLA
1727 1522 TAD I INSTTR
1730 7420 SNL /MQ TO BE SET?
1731 7200 CLA /NO
1732 3110 DCA MQDATA
1733 1177 TAD DATPAT
1734 7012 RTR
1735 7010 RAR
1736 7420 SNL /LINK TO BE SET?
1737 7200 CLA /NO
1740 3111 DCA LKDATA
1741 5742 JMP I NXTPTT /JMP TO NXTPAT
1742 1360 NXTPTT, NXTPAT
1743 1340 NXTONN, NXTONE
1744 1356 TESAGG, TESAGN
/.....
/TEST FOR I/O SIMULATOR
1745 7000 XOR29, NOP/JMS POINT
1746 1021 TAD 21 /GET HARDWARE DESIGNATOR
1747 0052 AND K400 /MASK OUT SIMULATOR BIT,
1750 7440 SEA /DO WE HAVE A SIMULATOR?
1751 5772 JMP I TSTCHX/CLA CLL FOR 1K /YES, DO PART 2
/TEST FOR HALT AT END OF PASS
1752 1021 LOPBAK, TAD 21 /NO
1753 7700 SMA CLA /FRONT PANEL AVAILABLE?

```

```

1754 5357 JMP ,+3 /NO
1755 7404 QSR /YES, "OR" THE SWITCH REGISTER WITH AC
1756 7410 SKP
1757 1020 TAD 20 /NO, FRONT PANEL, USE 20 FOR SR
1760 0052 AND K400
1761 7640 SEA CLA /HALT ON PASS COMPLETE?
1762 7402 7402/HLT /YES, HALT.
1763 1021 TAD 21 /GET HARDWARE DESIGNATOR
1764 0050 AND K100
1765 7440 SEA /RUNNING ON XOR?
1766 5773 JMP I XORPNT/NOP FOR 1K /YES
1767 7000 XOR30, NOP/JMS POINT
1770 5771 STOVER, JMP I STOVER /STAR! OVER BUT SKIP INITIAL HALT
1771 0222 STOVER, TSCACL
1772 2200 TSTCHX, 2200
1773 4000 XORPNT, XORSTT
1777 1777 *1777
1777 0000 0 /USED FOR DATA BREAK
2200 7000 *2200
/*****
/B/A INSTRUCTION TEST, PART 2, TO BE RUN ONLY WITH THE B/A I/O SIMULATOR
/MODULE, THIS SECTION COMPLETES TESTING OF THE B/A CPU MODULE.
/*****
/NOTE: DURING ALL I/O TRANSFERS WITH THE B/A I/O SIMULATOR, THE "0"
/LINES ARE ASSERTED FROM TP3 TIME OF THE IOT INSTRUCTION UNTIL TP3 TIME
/OF THE FOLLOWING INSTRUCTION, IN ORDER TO TEST THE ADDRESSING OF ROM E.
/(I.E. "0" LINES ASSERTED WITH NO I/O PAUSE, THIS CONDITION SHOULD CAUSE
/NO ENABLES TO BE OUTPUTTED FROM ROM E) AN IMPROPERLY BLASTED ROM E COULD
/CAUSE ANY OF THE ERROR HALTS IN THE SUCCEEDING SECTION, OR IN THE ROM E
/TESTS THAT FOLLOW LATER IN THE PROGRAM.
/*****
/CHECK FOR B/A OR B/E TYPE CPU
TSTFBE, NOP/JMS POINT
2201 7200 CLA
2202 6001 ION
2203 1021 TAD 21 /GET HARDWARE DESIGNATOR
2204 0046 AND K40 /MASK OUT B/E BIT
2205 7650 SNA CLA /B/A?
2206 1353 TAD K410 /YES, MAKE A SKIP
2207 1034 TAD K7000 /NO, B/E, MAKE A NOP
2210 3777 DCA TSOPNS /THIS ACTION PREVENTS TESTING "DEPOSIT
/ NON-STOP" ON PDP-B/E'S
2211 1021 TAD 21
2212 0046 AND K40
2213 7640 SEA CLA /B/E?
2214 1034 TAD K7000 /YES
2215 3777 DCA DFBEBA /THIS ACTION MAKES UP FOR A SMALL TIMING
/ DIFFERENCE IN THE HANDLING OF "USER MODE"
/ BETWEEN B/E'S AND B/A'S.
/*****
/CHANGE POWER FAIL TO RETURN TO BEGINNING OF PART TWO IN THE EVENT
/OF A POWER FAILURE OCCURRING DURING PART TWO
2216 7240 CLA CMA
2217 3100 DCA PARTWO

```

```

/*****
/SET UP INTERRUPT SKIP CHAIN IF ON ACT-B/E LINE
2220 1022 ACTINT, TAD 22 /GET HARDWARE WORD 2
2221 7004 RAL /ACT-B/E BIT TO LINK
2222 7700 SMA CLA /ON ACT-B/E LINE?
2223 5236 JMP XOR27 /NO, SKIP THE RESTORE
2224 1375 TAD (6244 /YES, GET A RMP INSTRUCTION FOR SKIP CHAIN.
2225 3001 DCA 1
2226 1374 TAD (JMP SKPOH
2227 3002 DCA 2
/*****
/TEST THAT A JMS IS NOT DECODED AS AN IOT, CAUSING "I/O PAUSE".
JHSTS3, TAD KJMP14
2230 1354 DCA 0 /SET UP A RETURN FROM LOCATION 4
2231 3005 CLE CML /SET LINK
2232 7120 JMS 4 /THIS WILL CAUSE A RTF OPERATION
2233 4004 /IF "I/O PAUSE" IS ERRORRONEOUSLY GENERATED.
/RETURN HERE IMMEDIATELY FROM LOCATION 4
2234 7440 SEA /AC STILL CLEAR?
2235 7402 HLT /NO, JMS TO LOCATION 4 SET THE AC.
/*****
/TEST FOR AC OUTPUTTED WHEN ALL C LINES=H (ROM E, ADDRESS 27)
XOR27, NOP/JMS POINT
TSTCHI, CLRAL
2236 7000 CLE CLA CMA /AC TO 7777
2237 6145 STROB /IOT WITH C0,C1,C2=H
2240 7340 SNA CLA /ENSURE AC IS STILL SET
2241 6142 HLT /AC WAS CLEARED WHEN C0,C1,C2=H
2242 7650 READA /READ BACK THE OUTPUT DATA, AC TO 7777
2243 7402 CMA /AC TO 0000, DROPPED BITS NOW APPEAR AS 1'S
2244 6141 SEA /ANY BITS DROP?
2245 7040
2246 7440
/A HALT AT THE FOLLOWING LOCATION CAN BE CAUSED BY SETTING THE I/O SIMULATOR
/BIT IN LOCATION 21, WHEN NO I/O SIMULATOR IS ON THE OMNIBUS. SEE SECTION
/4.2.1 FOR PROPER INITIALIZATION PROCEDURE
2247 7402 HLT /YES, BITS NOT OUTPUTTED/INPUTTED ARE
/ NOW SET IN THE AC.
/*****
/TEST FOR OUTPUT AC AND CLEAR AC WHEN C0=L, C1=H, & C2=H (ROM E ADD 26)
XOR32, NOP/JMS POINT
TSTCLB, CLRAL /CLEAR SIMULATOR
2250 7000 CLA CMA
2251 6145 LOADA
2252 7240 SEA /WAS AC CLEARED?
2253 6146 HLT /NO, C0=L C1=C2=H DID NOT CLEAR AC
2254 7440 HLT
2255 7402 READA /GET OUTPUTTED DATA
2256 6141 CMA /AC TO 0000, ANY BITS DROPPED ARE NOW SET
2257 7040 SEA /ANY BITS DROP?
2260 7440 HLT /YES, BITS NOT INPUTTED/OUTPUTTED IN AC
2261 7402
/*****
/TEST FOR "OR" INPUT TO AC WHEN C0&C2=H AND C1=L (ROM E ADDRESS 25)
XOR46, NOP/JMS POINT
TSTCLC, CLRAL
2262 7000 TAD K2000
2263 6145 LOADB /C1=L
2264 1055
2265 6147

```

```

2266 1056 TAD K2525
2267 6146 LOADA /REGISTER A TO 2525
2270 1064 TAD K5252 /AC TO 2525
2271 6142 STROB /AC TO 7777
2272 7040 CMA /AC TO 0000,BITS NOT "OR'ED" SET IN AC
2273 7440 SEA /ANY BITS NOT OR'ED?
2274 7402 HLT /YES,BITS SET IN AC DID NOT "OR" INPUT
/.....
2275 7000 /TEST FOR JAM INPUT WHEN C0&C1=L AND C2=H (FROM E ADDRESS 24)
XOR47, NOP/JMS POINT
2276 6145 TSTCLD, CLRAL /CLEAR SIMULATOR
2277 1056 TAD K2525 /AC TO 2525
2300 6146 LOADA
2301 1355 TAD K6000 /C0&C1=LOW
2302 6147 LOADB /AC TO 2525
2303 1064 TAD K9252 /AC TO 2525
2304 6142 STROB /AC TO 0000
2305 1065 TAD K9253 /JAM INPUT DONE CORRECTLY?
2306 7450 SNA /YES,
2307 5312 JMP XOR31 /NO, JAM INPUT FAILED.
2310 7402 HLT
/.....
2311 7402 /IOFHLT, HLT /IOF FAILED TO DISABLE INTERRUPT, SEE BELOW.
/TEST REMAINDER OF ROM J AND ASSOCIATED CIRCUITRY
XOR31, NOP/JMS POINT
2312 7000 CLRAL /CLEAR SIMULATOR
2313 6145 TAD (IOFHLT) /SET UP INTERRUPT RETURN
2314 1373 DCA 7 /TO IOFHLT
2315 3007 TAD (JMP I 7) /SET UP SKIP CHAIN FOR RETURNS
2316 1372 IOF /TURN OFF INTERRUPT
2317 6002 DCA CHNCON
2320 3036 TAD K200 /CAUSE AN INTERRUPT
2321 1051 LOADA CNTENA
2322 6146 /.....
/TEST SRQ TO SKIP ON INTERRUPT REQUEST
2324 6003 TSTSRQ, SRQ /SKIP ON INTERRUPT REQUEST
2325 7402 HLT /SRQ DID NOT SKIP, OR SIMULATOR DID NOT
/CAUSE AN INTERRUPT
/.....
2326 6004 /TEST GTF TO SAVE INTERRUPT LINE
GTFSS, GTF /GET FLAGS
2327 0131 AND K1000 /MASK INTERRUPT REQUEST
2330 7650 SNA CLA /GTF DID NOT SAVE INTERRUPT LINE
2331 7402 HLT
2332 7000 XOR33, NOP/JMS POINT
/.....
/TEST FOR INTERRUPT TO OCCUR ONLY WHEN "FSET LOW"
2333 6145 TSFSET, CLRAL
2334 6001 ION
2335 1371 TAD (FSETER
2336 3007 DCA 7
2337 1051 TAD K200

```

```

2340 6146 LOADA /CAUSE AN INTERRUPT
2341 6140 CNTENA
2342 1513 TAD I POINTR /INTERRUPT AFTER EXECUTE, AC TO 2526
2343 7402 HLT /WE SHOULD HAVE INTERRUPTED
2344 1343 FSETER, TAD ,=1
2345 3036 DCA CHNCON
2346 1074 TAD SAVAC /GET INTERRUPTED AC
2347 1064 TAD K9252
2350 7440 SEA
2351 7402 HLT /TAD INDIRECT DID NOT WORK CORRECTLY WHEN
/INTERRUPT WAS UP DURING DEFER AND
/EXECUTE, MAY HAVE INTERRUPTED AT END OF
/FETCH OR DEFER
/CONTINUE ON NEXT PAGE
2352 5770 JMP XOR34
2353 0410 K410, 0410
2354 5404 KJMP14, JMP I 4
2355 6000 K6000, 6000
2370 2400
2371 2344
2372 5407
2373 2311
2374 5023
2375 6244
2376 2451
2377 3350
2400 2400 PAGE
/.....
/TEST OPERATION OF INTERRUPT DELAY
2400 7000 XOR34, NOP/JMS POINT
2401 1377 TSTINT, TAD (OLYHLT
2402 3007 DCA 7 /SET UP INTERRUPT RETURN
2403 1376 TAD (JMP I 7
2404 3036 DCA CHNCON
2405 1051 TAD K200
2406 6146 LOADA /CAUSE INTERRUPT PULSE
2407 6140 CNTENA
2410 6001 ION /WILL NOT INTERRUPT IF INT DELAY WORKING
2411 6002 IOF
2412 7410 SKP
2413 7402 OLYHLT, HLT /INTERRUPT DELAY NOT WORKING
/.....
/TEST OPERATION OF INTERRUPT
2414 7000 XOR35, NOP/JMS POINT
2415 6145 TSTINT, CLRAL
2416 6001 ION
2417 1375 TAD (TSTPCI
2420 3007 DCA 7 /SET UP INTERRUPT RETURN
2421 1051 TAD K200
2422 6146 LOADA /CAUSE INTERRUPT
2423 6140 CNTENA
2424 0101 K101, AND 101 /SHOULD INTERRUPT AT THE END OF THIS INSTRUCTION
2425 7402 INTADD, HLT /INTERRUPT FAILED
/.....

```

```

/TEST THAT INTERRUPT STORED CORRECT PC
2426 1225 TSTPCI, TAD      -1
2427 3036 DCA      CHNCON      /SET UP FOR UNEXPECTED INTERPUPT
2430 6001 ION
2431 1374 TAD      (-INTADD
2432 1000 TAD      0
2433 7440 SZA
2434 7402 HLT      /INTERRUPT DID NOT STORE CORRECT PC
/.....
/TEST THAT USER MODE DISABLES A HLT INSTRUCTION
2435 7000 XOR36, NOP/JMS POINT
2436 6145 CLRAL
2437 1050 TAD      K100
2440 6146 LOADA      /CAUSE A PULSE ON USER MODE
2441 6140 CNTENA
2442 7602 7602/HLT CLA      /ONLY HALTS IF USER MODE FAILS TO DISABLE
/.....
/TEST THAT USER MODE DISABLES AN IOT INSTRUCTION
/THIS TEST DIFFERS SLIGHTLY IN TIMING BETWEEN 8/A AND 8/E TYPE PROCESSORS
2443 7000 XOR37, NOP/JMS POINT
2444 6145 CLRAL
2445 1050 TAD      K100      /USER MODE PULSE BIT
2446 7001 IAC      /3 CYCLE DELAY
2447 6146 LOADA
2450 6140 CNTENA
2451 0000 DFBEBA, AND      0/NOP      /((NOP FOR PDP-8/E), THE INSTRUCTION DIFFERENCES EXIST
/FOR THE FOLLOWING REASON: THE SIMULATOR PULLS THE USER MODE LINE
/ON THE THIRD TP1 AFTER THE CNTENA INSTRUCTION, THE USER MODE
/LINE WILL REMAIN LOW UNTIL THE FOLLOWING TP1 PULSE, FOR PDP-8/A'S
/THE STATE OF THE USER MODE LINE IS GATED TO A F/F AT TP1 TIME
/AND THE F/F IS USED TO DISABLE I/O PAUSE AT TP3 TIME, PDP-8/E'S
/USE THE USER MODE DIRECTLY AT TP3 TIME TO DISABLE I/O PAUSE.
/THEREFORE IN THE PROGRAM, ALTHOUGH THE USER MODE LINE GOES LOW
/AT TP1 TIME OF THE NOP AT DFBEBA+1, IT DOES NOT ARRIVE AT THE
/8/A CPU IN TIME TO SET THE F/F WHICH CHECKS THE USER MODE LINE
/AT TP1 TIME, SO I/O PAUSE IS NOT DISABLED DURING THE NOP,
/AT THE NEXT TP1 TIME, USER MODE REMAINS LOW LONG ENOUGH TO
/SET THE F/F, THEREFORE I/O PAUSE IS DISABLED FOR THE READA INSTRUCT
/FOR PDP-8/E'S, THE "AND 0"(USES FETCH & EXECUTE), IS CHANGED
/TO A "NOP"(USES FEICH ONLY), THIS ELIMINATES ONE MAJOR STATE BETWEEN
/THE "CNTENA" AND THE "READA", SINCE IN EFFECT THE PDP-8/E SEES
/THE USER MODE LINE ONE MAJOR STATE EARLIER THAN THE 8/A IN THIS
/PARTICULAR INSTANCE.

2452 7000 NOP
2453 6141 READA      /WON'T READ IF USER MODE DISABLES I/O PAUSE
2454 7640 SZA CLA

/.....
/TEST SKIP LINE TO CPU BOARD AND OVERFLOW LINE FROM CPU BOARD, AND "LINK LOAD" LINE
/.....
/THE FOLLOWING HALT CAN BE CAUSED BY SETTING BIT 6 IN LOCATION 21 TO
/SPECIFY AN 8/E TYPE CPU WHEN THE TEST IS ACTUALLY RUNNING ON A PDP-8/A.
/OR VICE VERSA,
2455 7402 HLT      /USER MODE DID NOT DISABLE I/O PAUSE
/.....

```

```

2456 7000 XOR38, NOP/JMS POINT
2457 7360 CLA CMA CLL CML      /AC TO 7777 LINK TO 1
2460 3112 DCA      TESLOC      /SET UP FOR ISE SKIP
2461 6145 CLRAL      /CLEAR SIMULATOR
2462 2112 ISE      TESLOC      /ISE SKIPS AND PUTS OVERFLOW LOW ON BUS
2463 7402 HLT      /ISE UID NOT SKIP
2464 6144 SKPOFV      /SKIP ON OVERFLOW
2465 7402 HLT      /SKIP LINE FROM OMNIBUS STUCK HIGH
/OR OVERFLOW LINE TO BUS STUCK HIGH
2466 7430 SELI      /DID LINK LOAD FROM "CLRAL" INSTRUCTION CLEAR LINK?
2467 7402 HLT      /NO, LINK LOAD FAILED TO CLEAR LINK
/.....
/TEST ROM H ADDRESS 04, BY EXECUTING AN IOT WITH M03 SET, CHECK TO INSURE
/OP1, OP2, OR OP3 NOT BEING DECODED, AND THAT NEITHER EXECUTE NOR DEFER
/IS BEING DONE,
2470 7000 XOR48, NOP/JMS POINT
2471 7300 ROMH04, CLA CLL
2472 1224 TAD      K101
2473 6146 LOADA      /PULSE TO USER MODE AFTER 3 CYCLE DELAY
2474 6140 CNTENA
2475 7240 CLA CMA
2476 6770 6770
2477 7402 7402/HLT

2500 7650 SNA CLA
2501 7402 HLT      /AC STILL SET?
/NO, ROM H ADDRESS 04 PROBABLY TRANSLATED
/AN IOT WITH BIT 3 SET AS AN OPERATE,
/OR DEVICE ON SYSTEM USES 6770 IOT,
/.....
/TEST THAT OVERFLOW F/F DOES NOT SET SKIP WHEN NOT DOING AN ISE.
2502 7000 XOR39, NOP/JMS POINT
2503 7340 I2ITTS, CLA CLL CMA      /AC TO 7777
2504 3010 DCA      10      /SET AUTO-INDEX LOCATION TO 7777
2505 1376 TAD      (JMP I 7      /SET UP INTERRUPT RETURN
2506 3036 DCA      CHNCON
2507 1373 TAD      (PCTST2
2510 3007 DCA      7      /SET RETURN FROM INTERRUPT
2511 1051 TAD      K200      /INTERRUPT BIT
2512 6146 LOADA      /INTERRUPT
2513 6140 CNTENA
2514 5410 JMP I 10      /0010=7777 SO WE AUTO-INDEX AND JUMP TO 0000, BUT
/INTERRUPT COMES UP AT TP1 OF FETCH
/AND IS HONORED AFTER DEFER MAJOR STATE
/DO NOT INTERRUPT
2515 7402 HLT
2516 1000 PCTST2, TAD      0      /GET INTERRUPTED PC(SHOULD BE 0000)
2517 7440 SZA      /DID WE END UP AT LOCATION 0000 AFTER JUMP?
2520 7402 HLT      /NO, IF AC=0001 THE OVERFLOW F/F PROBABLY
/SET THE SKIP F/F DURING DEFER OF JMP I (AUTO-INDEX)
/.....
/TEST INTERRUPT TO NOT INTERFERE WITH A NON-SKIPPING ISE
2521 1372 TAD      (JMP I 0

```

```

2522 3036 DCA CHNCON
2523 3003 DCA BIT3
2524 6001 IOV
2525 1051 TAD K200
2526 6146 LOADA
2527 6140 CNTENA
2530 2003 ISZ BIT3 /INTERUPT NEXT CYCLE
2531 7410 RETURN, SKP /SHOULD NOT SKIP
2532 7402 HLTGOD, HLT /INTERUPTED, RETURNED, AND DID NOT SKIP,
2533 1332 TAD HLTGOD /INTERUPT CAUSED ISZ TO SKIP
2534 3036 DCA CHNCON /MAKE HLT FOR UNEXPECTED INTERRUPT
2535 6001 IOV

/*****
/TEST INDICATE LOGIC, FIRST MAKE IND1+2 LOW AND TEST FOR AC TO BUS
XOR40, NOP/JMS POINT
TSINDL, CLRAL /CLEAR SIMULATOR
TAD K2525 /AC TO 2525
MQL /MO TO 2525, AC TO 0000
TAD K446 /IND1+2 LOW AC TO BUS
LOADB
CNTENA /ENABLE IND 1&2 TO BUS
TAD K5253 /AC TO 5253, AC ON BUS DURING TS1
READA /READ REGISTER A, AC REMAINS 5253
TAD K2525 /AC TO 0000
SEA
HLT /IND1+2 LOW DID NOT GATE AC TO BUS DURING TS1

/*****
/TEST THAT MQ GATES TO BUS DURING TS1 WHEN IND1=HI, IND2=LO
XOR41, NOP/JMS POINT
TSINDB, CLRAL
TAD K2525
MQL /MO TO 2525
TAD K444 /IND1=HI IND2=L
LOADB
TAD K5252 /ENABLE IND 1&2 (MO WILL BE ON BUS DURING TS1)
CNTENA /AC TO 2525
READA /AC TO 0000
TAD K5253 /HAS MO ON BUS AT TS1?
SEA

/A HALT AT THE NEXT LOCATION CAN BE CAUSED BY HAVING THE FRONT PANEL "INDICATE" SWITCH
/IN SOME POSITION OTHER THAN THE "MO", "STATUS", OR "STATE" POSITIONS. SEE
/SECTION 3, PART D.
HLT /NO, IND1=HI, IND2=LOW DID NOT PUT MO TO BUS
JMP TSINDC
K444, 0444
K446, 0446
2565 7402 HLT
2566 5771 JMP TSINDC
2567 0444 K444, 0444
2570 0446 K446, 0446
2571 2600
2572 5400
2573 2516
2574 5353
2575 2426
2576 5407
2577 2413
2600 PAGE
    
```

```

/*****
/TEST THAT IND1=HI, IND2=HI PUTS STATUS ON BUS DURING TS1
/SINCE STATUS TO BUS IS A FUNCTION OF ROM J, THIS TEST MUST BE DONE
/8 TIMES WITH A DIFFERENT OCTAL DIGIT IN MD BITS 9 THRU 11 EACH TIME
/TO COVER ALL THE ADDRESSES USED IN ROM J TO PERFORM THIS FUNCTION
2600 3112 TSINDC, DCA TESLOC
2601 1071 TAD K7770 /#8
2602 3003 DCA BIT3
2603 7000 XOR42, NOP/JMS POINT
2604 6145 CLRAL /CLEAR SIMULATOR
2605 7120 CLL CML /SET LINK FOR STATUS INDICATION
2606 1304 TAD K440
2607 6147 LOADB
2610 6140 CNTENA /ENABLE IND OUTPUTS TO BUS
2611 1112 TAD TESLOC /TO GET DIFFERENT BITS IN MD9 THRU 11
2612 6141 READA /AC SHOULD BE 42XX
2613 0067 ANC K7700 /MASK OUT IF + DF
2614 1305 TAD M4200
2615 7440 SEA

/A HALT AT THE FOLLOWING LOCATION CAN BE CAUSED BY HAVING THE FRONT PANEL
/"INDICATE" SWITCH IN SOME POSITION OTHER THAN THE "MO", "STATUS", OR
/"STATE" POSITION, SEE SECTION 3, PART D.
2616 7402 HLT /DIDN'T GET STATUS TO BUS WHEN IND1+2 HI
/IF (TESLOC) = 0000 THEN SUSPECTED FAILURE
/IN INDICATE LOGIC, IF (TESLOC) = NON-ZERO
/THEN SUSPECTED ROM J FAILURE

2617 7000 XOR43, NOP/JMS POINT
2620 2112 ISZ TESLOC /FOR NEXT COMBO OF MD9 THRU 11
2621 2003 ISZ BIT3 /ALL COMBINATIONS TRIED? (0)
2622 5203 JMP XOR42 /NO

/*****
/TEST THAT AC, MQ, OR STATUS IS NOT ON DATA BUS DURING TS1 WHEN IND1=LO, IND2=HI
2623 7000 XOR44, NOP/JMS POINT
2624 1302 TSINDD, TAD K442
2625 6147 LOADB /IND1=LOW IND2=HI
2626 1056 TAD K2525
2627 7421 MQL /MO TO 2525
2630 7240 CLA CMA /AC TO 7777
2631 6140 CNTENA
2632 6141 READA /AC TO 0000
2633 7440 SEA
2634 7402 HLT /AC, MQ, OR STATUS ON BUS WHEN IND1=LOW IND2=HI
2635 7000 XOR45, NOP/JMS POINT

/*****
/CONTINUED TEST OF ROM E
/*****
/TEST FOR INPUT, DATA ADDED TO PC WHEN C0=H, C1=H, C2=L (ROM E ADDRESS 23)
2636 6172 TSTCLE, XRC1 /DISABLE XOR INTERRUPTS
2637 6145 CLRAL /CLEAR SIMULATOR
2640 1377 TAD (BACKAD
2641 3007 DCA 7 /SET UP INTERRUPT RETURN
2642 1376 TAD (JMP I 7
2643 3036 DCA CHNCON
    
```

```

2644 1131      TAD      K1000
2645 6147      LOADB
2646 1051      TAD      K200
2647 6146      LOADA
2650 6140      CNTENA
2651 6142      STROB
2652 7402      HLT
2653 1074      BACKAD, TAD      SAVAC
2654 7440      SEA
2655 7402      HLT
2656 1255      TAD      ,=1
2657 3036      DCA      CHNCON
2660 1000      TAD      0
2661 6001      IOY
2662 1375      TAD      (-BACKAD=200+1
2663 7440      SEA
2664 7402      HLT
/NO, C0&C1=L AND C2=H DID NOT ADD TO PC
/*****
/A FAILURE IN THE TEST ABOVE OR THE TEST BELOW COULD BE DUE TO A PROBLEM
/IN THE "NOT LAST XFER" OR "I/O STALL" LOGIC. THESE TWO ROUTINES ARE THE
/ONLY ONES IN THE PROGRAM THAT CAUSE THE SIMULATOR TO ASSERT "NOT LAST XFER",
/WHICH IN TURN ASSERTS I/O STALL IN THE CPU.
/*****
/TEST FOR INPUT, DATA ADDED TO PC WHEN C0&2=L AND C1=H (ROM E ADDRESS 22)
XOR49, NOP/JMS POINT
TSTCLF, CLRAL
/BAKADB
DCA 7
TAD K5000
LOADB
(JMP I 7
DCA CHNCON
TAD K200
LOADA
CNTENA
STROB
HLT
K442, 0442
K5000, 5000
K440, 0440
M4200, 3600
BAKADB, TAD SAVAC
SEA
HLT
TAD ,=1
DCA CHNCON
TAD 0
IOY
TAD (-K442=200+1
SEA
HLT
/NO, C0&2=L C1=H DID NOT ADD TO PC
/*****
/TEST FOR INPUT TO PC WHEN C0=H C1&2=L (ROM E ADDRESS 21)
XOR50, NOP/JMS POINT
TSTCLG, XRSI
/BAKADB
DCA 7
TAD K3000
LOADB
(JMP I 7
DCA CHNCON
TAD K200
LOADA
CNTENA
STROB
HLT
K3000, 3000
BAKADB, TAD SAVAC
SEA
HLT
TAD ,=1
DCA CHNCON
TAD 0
IOY
TAD M200
SEA
HLT
/NO, C0=H C1&2=L DID NOT LOAD PC
/*****
/TEST FOR INPUT TO PC WHEN C0, C1, & C2 = L (ROM E ADDRESS 20)
XOR51, NOP/JMS POINT
TSTCLH, CLRAL
/BAKADD
DCA 7
TAD K7000
LOADB
(JMP I 7
DCA CHNCON
TAD K200
LOADA
CNTENA
STROB
HLT
M200, 7600
PAGE
BAKADD, TAD SAVAC
SEA
HLT
TAD ,=1
DCA CHNCON
TAD 0

```

```

2722 6145      CLRAL
2723 1372      TAD      (BAKADC
2724 3007      DCA      7
2725 1336      TAD      K3000
2726 6147      LOADB
2727 1376      TAD      (JMP I 7
2730 3036      DCA      CHNCON
2731 1051      TAD      K200
2732 6146      LOADA
2733 6140      CNTENA
2734 6142      STROB
2735 7402      HLT
2736 3000      K3000, 3000
2737 1074      BAKADC, TAD SAVAC
2740 7440      SEA
2741 7402      HLT
2742 1341      TAD      ,=1
2743 3036      DCA      CHNCON
2744 1000      TAD      0
2745 6001      IOY
2746 1366      TAD      M200
2747 7440      SEA
2750 7402      HLT
/NO, C0=H C1&2=L DID NOT LOAD PC
/*****
/TEST FOR INPUT TO PC WHEN C0, C1, & C2 = L (ROM E ADDRESS 20)
XOR51, NOP/JMS POINT
TSTCLH, CLRAL
/BAKADD
DCA 7
TAD K7000
LOADB
(JMP I 7
DCA CHNCON
TAD K200
LOADA
CNTENA
STROB
HLT
M200, 7600
PAGE
BAKADD, TAD SAVAC
SEA
HLT
TAD ,=1
DCA CHNCON
TAD 0

```

```

3006 6001      ION
3007 1777      TAD      H200      / 7600
3010 7440      SEA              /WAS PC CORRECT?
3011 7402      HLT              /NO, 001.4 2 LOW DID NOT INPUT TO PC
/.....
3012 7000      /TEST ROM B, ADDRESS 15 BY INTERRUPTING AT THE END OF A DEFER
XOR52, NOP/JMS POINT
ROMB15, CLRAL      /CLEAR SIMULATOR
3013 6145      TAD      (ROBRET
3014 1376      DCA      7      /SET UP INTERRUPT RETURN
3015 3007      TAD      (JMP I 7
3016 1375      DCA      CHNCON      /SET UP FOR EXPECTED INTERRUPT
3017 3036      TAD      K201
3020 1227      LOADA      /INTERUPT, 3 CYCLE DELAY
3021 6146      CNTENA
3022 6140      CLA GLL
3023 7300      JMP I 1,+1
3024 5774      ROBHLT, HLT      /INTERRUPT AFTER DEFER
3025 7402      HLT      /DID NOT INTERRUPT
3026 7402      HLT      /DID NOT INTERRUPT, BUT CARRY IN ENABLED
3027 0201      K201, 0201
3030 1226      ROBRET, TAD      ,=2
3031 3036      DCA      CHNCON      /SET UP FOR UNEXPECTED INTERRUPT
3032 1000      TAD      0      /GET INTERRUPTED PC
3033 6001      ION
3034 1373      TAD      (=ROBHLT
3035 7440      SEA              /WAS PC CORRECT?
3036 7402      HLT              /NO, INTERRUPT AFTER DEFER DID NOT STORE CORRECT PC
/.....
3037 7000      /TEST "ROM ADD L" TO DISABLE STORING PC FOR A JMS AND DISABLE MA+1 TO PC.
XOR53, NOP/JMS POINT
TSROMA, CLRAL      /CLEAR SIMULATOR
3041 1372      TAD      (GDRETU
3042 3007      DCA      7      /SET UP INTERRUPT RETURN
3043 1375      TAD      (JMP I 7
3044 3036      DCA      CHNCON      /SET UP FOR EXPECTED INTERRUPT
3045 1225      TAD      ROBHLT
3046 3254      DCA      ROMCLR      /SET UP INDICATOR
3047 1357      TAD      K221
3050 6146      LOADA      /CAUSE PULSE ON ROM ADD L, 3 CYCLE DELAY, & INT
3051 6140      CNTENA
3052 7000      NOP
3053 4254      JMS      ,+1      /PULSE ON ROM ADD L SHOULD ARRIVE DURING
/EXECUTE AND DISABLE STORING PC AND DISABLE
/MA+1 TO PC. IN CORE MACHINES NEXT LOCATION IS CLEARED SINCE
/ROM ADD L ARRIVES AND PREVENTS RESTORING MEMORY
3054 7402      ROMCLR, HLT
3055 7402      HLT      /DID NOT INTERRUPT AFTER EXECUTE OF JMS
3056 1225      GDRETU, TAD      ROBHLT
3057 3036      DCA      CHNCON      /SET UP FOR UNEXPECTED INTERRUPT
3060 1000      TAD      0      /GET INTERRUPTED PC
3061 6001      ION
3062 1371      TAD      (=ROMCLR
3063 7440      SEA              /DID ROM ADD L DISABLE MA+1 TO PC?
3064 7402      HLT              /NO, ROM ADD L DID NOT DISABLE MA+1 TO PC
/.....

```

```

/TEST BREAK FROM MEMORY AND BREAK BACK INTO MEMORY - (USES ROMA ADDRESSES 30 THRU 37
/AND ROM B ADDRESS 3)
3065 7000      XOR54, NOP/JMS POINT
3066 6145      DBOUT, CLRAL
3067 1050      TAD      K100
3070 6147      LOADB      /BREAK OUT
3071 7240      CLA CMA      /AC TO 7777
3072 6201      CDF 00      /IN CASE PROGRAM IS RUNNING IN EXT MEMORY
3073 3494      DCA I K1777      /1777 TO 7777, AC TO 0000
3074 6143      SETBK      /SET BREAK REQUEST
3075 6140      CNTENA
3076 7000      NOP
3077 6141      READA      /AC TO 7777
3100 7040      CMA      /AC TO 0000 ANY BITS DROPPED FROM DB ARE SET
3101 7440      SEA      /ANY BITS DROP DURING DB?
3102 7402      HLT      /YES, DB OUT DID NOT WORK, ANY BIT SET IN
/AC IS A BIT DROPPED ON A DB OUT OF MEMORY.
/.....
3103 3494      /TEST BREAK INTO MEMORY (ADDRESS 1777)
TSBKTO, DCA I K1777      /CLEAR 1777
3104 6147      LOADB      /BREAK IN
3105 6143      SETBK
3106 6140      CNTENA      /BREAK IN OCCURS END OF NEXT CYCLE
3107 7000      NOP
3110 1494      TAD I K1777      /AC TO 7777 IF BREAK IN AND OUT BOTH WORKED
3111 7040      CMA      /AC TO 0000, DROPPED BITS APPEAR AS 1'S
3112 7440      SEA      /ANY BITS DROPPED DURING DATA BREAK?
3113 7402      HLT      /YES, BREAK OUT OR BREAK IN FAILED COMPLETELY
/OR DROPPED BITS, BITS NOT TRANSFERRED ARE
/NOT SET IN AC
/.....
/TEST DATA BREAK ADD TO MEMORY
3114 1051      TSBKAD, TAD      K200
3115 6147      LOADB
3116 1064      TAD      K5252
3117 3494      DCA I K1777      /SET 1777 TO 5252
3120 6143      SETBK      /SET BREAK REQUEST
3121 6140      CNTENA      /BREAK IN AND ADD TO 1777 NEXT CYCLE
3122 7000      NOP      /1777 TO 5251 DURING DATA BREAK
3123 1494      TAD I K1777      /AC TO 5251
3124 1360      TAD      K2527      /AC TO 0000
3125 7440      SEA      /DID ADD TO MEMORY BREAK WORK?
3126 7402      HLT      /NO, DATA BREAK ADD TO MEMORY DID NOT FUNCTION.
/ROM A ADDRESSES 30 THRU 33 ARE SUSPECT
/.....
/TEST BREAK OUT OF ADDRESS 0017
3127 7000      XOR55, NOP/JMS POINT
3130 6145      BKFLOW, CLRAL
3131 1056      TAD      K2525      /AC TO 2525
3132 3761      DCA I K17      /0017 TO 2525(DF=0)
3133 7307      CLA GLL IAC RTL      /+4
3134 6146      LOADA      /BREAK FORM LOW ADDRESS (0017)
3135 1050      TAD      K100
3136 6147      LOADB      /BREAK OUT OF MEMORY
3137 6143      SETBK

```

```

3140 6140 CNTENA
3141 7000 /BREAK AFTER NOP, 2525 TO SIMULATOR
3142 6141 READA /AC TO 2525
3143 1065 TAD K5253 /AC TO 0000
3144 7440 SZA /BREAK FROM LOW ADDRESS WORK?
3145 7402 HLT /NO, BREAK OUT OF ADDRESS 0017 FAILED, CPMA
/BIT MAY NOT BE GOING TO ITS HIGH IMPEDENCE STATE,
/*****
/TEST BREAK INTO 0017
3146 6147 BKIN17, LOADB /BREAK IN
3147 3761 DCA I K17 /CLEAN 00017
3150 6143 SETBK
3151 6140 CNTENA
3152 7240 CLA CMA /BREAK AFTER THIS INSTRUCTION
3153 7040 CMA /AC TO 0000
3154 7448 SZA /DATA BREAK AFFECT AC?
3155 7402 HLT /YES, DATA BREAK AFFECTED AC,
3156 5770 JMP BKINTO
3157 0221 K221, 0221
3160 2527 K2527, 2527
3161 0017 K17, 0017
3170 3200
3171 4724
3172 3056
3173 4753
3174 3025
3175 5407
3176 3030
3177 2766
3200 3200 PAGE
BKINTO, TAD I K178 /AC TO 2525
3201 1773 TAD K5253 /AC TO 0000
3202 1065 SZA /BREAK INTO 0017 CORRECTLY
3203 7402 HLT /NO, BREAK INTO 0017 FAILED.
/*****
/CAUSE AN ASYNCHRONOUS DATA BREAK
3204 7000 XOR56, NOP/JMS POINT
3205 6145 ASYNCS, CLRAL
3206 1052 TAD K400
3207 6146 LOADA /ASYNCS DATA BREAK
3210 7240 CLA CMA /AC TO 7777
3211 3454 DCA I K1777 /1777 TO 7777
3212 6143 SETBK
3213 6140 CNTENA /BREAK WITHIN 14 USEC,
3214 7346 CLA CMA CLL RTL /-3
3215 3112 DCA TESLOC
3216 2112 ISZ TESLOC
3217 5216 JMP ,=1 /TIME WASTER, DB OCCURS DURING THIS WAIT,
3220 1454 TAD I K1777 /AC TO 0000
3221 7440 SZA /BREAK OCCUR CORRECTLY?
3222 7402 HLT /NO, ASYNCS DATA BREAK DID NOT CLEAR 1777
/*****
/TEST FOR PROPER OPERATION OF 5 BREAKS IN A ROW
3223 7000 XOR57, NOP/JMS POINT
3224 6145 TS5BRK, CLRAL

```

```

3225 1050 TAD K100
3226 6147 LOADB
3227 1056 TAD K2525
3230 3454 DCA I K1777 /((1777) TO 2525
3231 6143 SETBK
3232 6140 CNTENA
3233 7000 NOP /((1777) TO SIMULATOR
3234 1043 TAD K4
3235 6146 LOADA /LOW ADDRESS FOR DB
3236 1041 TAD K1
3237 6147 LOADB /5 BREAKS
3240 1371 TAD M58
3241 3112 DCA TESLOC /SET UP LOOP OF 5
3242 1271 TAD K12
3243 3010 DCA 10 /POINTER FOR ADDRESS 13
3244 3410 DCA I 10 /CLEAR 13 THRU 17
3245 2112 ISZ TESLOC /ALL 5 CLEARED?
3246 5244 JMP ,=2 /NO
3247 6143 SETBK
3250 6140 CNTENA
3251 7000 NOP /((13) THRU (17) TO 2525
3252 1271 TAD K12
3253 3010 DCA 10
3254 1371 TAD M58
3255 3112 DCA TESLOC
3256 1410 SOMMOR, TAD I 10 /GET A LOCATION BETWEEN 13 AND 17
3257 1065 TAD K5253 /COMPARE TO 2525
3260 7440 SZA /EQUAL 2525?
3261 7402 HLT /NO, 5 BREAKS DID NOT WORK CORRECTLY
3262 2112 ISZ TESLOC /ALL LOCATIONS CHECKED?
3263 5256 JMP SOMMOR /NO,
/*****
/TEST THAT INITIALIZE LINE(CR1) IS NOT OPEN TO BUS
3264 7000 XOR58, NOP/JMS POINT
3265 6145 TSINLN, CLRAL /CLEAN SIMULATOR
3266 7240 CLA CMA /AC TO 7777
3267 3112 DCA TESLOC
3270 2112 ISZ TESLOC /SHOULD SKIP AND SET OVERFLOW F/F IN SIMULATOR
3271 0012 K12, 0012
3272 6144 SKPOFV /SKIP ON OVERFLOW IN SIMULATOR, SHOULD SKIP
3273 7402 HLT /OVERFLOW DID NOT SET OVERFLOW F/F IN SIMULATOR,
3274 6007 CAF, /CLEAR OVERFLOW F/F IN SIMULATOR
3275 6144 SKPOFV /SKIP ON OVERFLOW IN SIMULATOR, SHOULD NOT SKIP NOW,
3276 7410 SKP /DID NOT SKIP, OK,
3277 7402 HLT /CAF DID NOT CLEAR OVERFLOW F/F IN SIMULATOR,
/*****
/TEST THAT DMA DISABLES CLOCKING INTERRUPT CIRCUITRY
3300 7000 XOR59, NOP/JMS POINT
3301 6145 TSDMIT, CLRAL
3302 6001 IOY
3303 1056 TAD K2525
3304 3454 DCA I K1777
3305 1050 TAD K100
3306 6147 LOADB /BREAK OUT OF MEMORY (1777)
3307 7001 IAC

```



```

3310 6146          LOADA          /DELAY OF THREE CYCLES
3311 6143          SETBK           /SET BREAK REQUEST
3312 6140          CNTENA          /BREAK IN 3 CYCLES
3313 7000          NOP
3314 3454          DCA I K1777      /DB SHOULD OCCUR AT BEGINNING OF EXECUTE, BEFORE
                                   /1777 IS CLEARED, THEREFORE DB DATA SHOULD BE 2525.
3315 6141          READA          /AC TO 0000
3316 1065          TAD            K5253 /DB DATA CORRECT?
3317 7440          SZA            /NO, IF AC = 5253, THEN DB OCCURRED AFTER THE EXECUTE
3320 7402          HLT            /OF THE DCA, OR DID NOT OCCUR, IF AC NOT EQUAL
                                   /5253 THEN DATA BREAK INTERFERED WITH THE DCA

3321 1051          TAD            K200
3322 7001          IAC
3323 6146          LOADA          /INTERUPT DELAY OF THREE CYCLES
3324 6147          LOADB
3325 1377          TAD            (DMARET
3326 3007          DCA            7      /SET UP INTERRUPT RETURN
3327 1376          TAD            (JMP I 7
3330 3036          DCA            CHNCON /SET UP FOR EXPECTED INTERRUPT
3331 6143          SETBK           /SET BREAK REQUEST
3332 6140          CNTENA          /ENABLE INT AND DB ON THIRD TP1
3333 7000          NOP            /CYCLE MASTER
3334 1454          TAD I K1777      /DB OCCURS JUST BEFORE EXECUTE OF TAD AND LOADS
                                   /ADDRESS 1777 WITH 2525, THEN THE EXECUTE OF THE TAD
                                   /LOADS THE AC WITH 2525, INTERRUPT OCCURS AFTER
                                   /THE EXECUTE OF THE TAD,
                                   /DID NOT INTERRUPT CORRECTLY, DATA BREAK AND INTERRUPT INTERFERED.
3335 7402          HLT            /GET INTERRUPTED AC, AC TO 2525.
3336 1074          DMARET, TAD    SAVAC /HSA AC CLEAR BECAUSE THE INTERRUPT CIRCUITRY WAS
3337 7450          SNA            /CLOCKED DURING DB, PREVENTING THE EXECUTE OF THE TAD?
                                   /YES, INTERRUPT WAS CLOCKED DURING DB.
3340 7402          HLT            /AC TO 0000.
3341 1065          TAD            K5253 /HSA AC CORRECT?
3342 7440          SZA            /NO, DMA AND INTERRUPT INTERFERED,
3343 7402          HLT            /SET UP FOR UNEXPECTED INTERRUPTS
3344 1343          TAD            ,=1
3345 3036          DCA            CHNCON
3346 6001          ION

/*****
/TEST ADDRESSES 10 THRU 13 OF ROM A.(DEPOSIT NON-STOP FUNCTION)
3347 7000          XOR60, NOP/JMS POINT
3350 7410          TSDPNS, SKP
3351 5775          JMP            XOR62 /NOP FOR PDP-8/E TYPE COMPUTERS
3352 6145          CLRAL          /USED TO SKIP DEPOSIT NON-STOP TEST FOR PDP-8/E
3353 7240          CLA CMA        /CLEAR OUT SIMULATOR
3354 3494          DCA I K1777
3355 1374          TAD            (ROMRTN
3356 3007          DCA            7      /SET UP INTERRUPT RETURN
3357 1376          TAD            (JMP I 7
3360 3036          DCA            CHNCON /SET UP FOR EXPECTED INTERRUPT
3361 1372          TAD            K220
3362 6147          LOADB          /DMA WITH LA ENABLE LOW, KEY CONTROL HIGH,
                                   /AND BREAK DATA CONTROL LOW.
3363 1051          TAD            K200
3364 6146          LOADA          /INTERUPT ENABLE

```

```

3365 6143          SETBK           /DMA WILL DEPOSIT 0 IN LOCATION 1777 BEFORE THE DEFER
3366 6140          CNTENA          /OF THE TAD, THEREFORE THE AC WILL BE LOADED WITH
3367 1454          TAD I K1777      /0000 DURING THE EXECUTE OF THE TAD, THE DMA WILL LOAD
                                   /THE PC WITH 2000, THEREFORE THE MA=2000
                                   /AFTER THE TAD, AN INTERRUPT OCCURS IMMEDIATELY
                                   /AFTER THE COMPLETION OF THE TAD.
                                   /FAILED TO LOAD PC OR INTERRUPT.

3370 7402          HLT            /MA,MS LOAD CONTROL LOW INHIBITS PC TO CPMA
3371 7773          MSB,          7773
3372 0220          K220,        0220
3373 0017          K17B,        0017
3374 3400
3375 3415
3376 5407
3377 3336          PAGE
                                   3400
3400 1074          ROMRTN, TAD    SAVAC /GET INTERRUPTED AC, AC TO 0000 IF DMA WORKED
3401 7440          SZA            /DMA WORK CORRECTLY?
3402 7402          HLT            /NO, DEPOSIT NON-STOP FAILED, ROM A ADDRESSES 10-13.
3403 1202          TAD            ,=1
3404 3036          DCA            CHNCON
3405 1106          TAD            SAVFLO /SET UP BACK TO WHATEVER IT WAS.
3406 6005          RTF            /RESTORE
3407 7300          CLA CLL
3410 1000          TAD            0      /GET INTERRUPTED PC, AC TO 2000
3411 5212          JMP            ,+1    /ENABLES INTERRUPTS
3412 1377          TAD            (=2000 /AC TO 0000
3413 7440          SZA            /HSA PC LOADED BY DMA?
3414 7402          HLT            /NO, DEPOSIT NON-STOP DID NOT LOAD PC
                                   /CORRECTLY, ROM A ADDRESSES 10 THRU 13 AT FAULT.

/*****
/TEST THAT MA,MS LOAD CONTROL LOW INHIBITS PC TO CPMA
3415 7000          XOR62, NOP/JMS POINT
3416 6145          TSMAMS, CLRAL
3417 1055          TAD            K2000 /PULSE MA,MS LOAD CONTROL LINE
3420 6146          LOADA          /PULSE WILL ARRIVE DURING NEXT INSTRUCTION
3421 6140          CNTENA          /THIS INSTRUCTION WILL BE EXECUTED TWICE, CAUSING A FINAL
3422 7001          IAC            /AC OF 0002, THE MA,MS LOAD CONTROL PULSE DURING
                                   /THE FIRST EXECUTION, PREVENTS PC TO CPMA AND
                                   /THEREBY CAUSES THE SAME INSTRUCTION TO BE EXECUTED OVER,
3423 1776          TAD            M2      /AC TO 0000
3424 7440          SZA            /HSA AC = 0002?
3425 7402          HLT            /NO, MA,MS LOAD CONTROL DID NOT INHIBIT PC TO CPMA

/*****
/TEST REMAINDER OF OMNIBUS LINES THAT SO FAR HAVE NOT BEEN USED IN BOTH A
/HIGH AND LOW STATE: MA0,RUN,INT IN PROG,LINK L, LINK DATA L,F SET L,
/FETCH,EXECUTE,DEFER,IR0,IR1,IR2.
3426 7000          XOR63, NOP/JMS POINT
3427 6145          TSLINE, CLRAL
3430 7005          IAC RAL          /AC TO 0002
3431 6146          LOADA          /5 CYCLE DELAY
3432 7001          IAC
3433 6140          CNTENA          /ENABLE STATUS READING

```

```

3434 7200 CLA /1ST CYCLE OF DELAY
3435 7000 NOP /2ND CYCLE
3436 0461 AND I K4000 /3TH CYCLE IS EXECUTE OF THIS INSTRUCTION
3437 6141 READA /AC TO 6120
3440 7500 SMA /WAS MA0 LOW?
3441 7402 HLT /NO. MA0 STUCK HIGH ON BUS
3442 7006 RTL /RUN BIT TO LINK.
3443 7420 SNL /RUN LOW?
3444 7402 HLT /NO. RUN LINE STUCK HIGH ON BUS.
3445 7500 SMA /INT IN PROG LOW?
3446 7402 HLT /NO. INT IN PROG STUCK HIGH ON BUS
3447 7006 RTL
3450 7430 SEL /LINK LINE HIGH?
3451 7402 HLT /NO. LINK LINE(AV2) STUCK LOW ON BUS.
3452 7510 SPA /LINK DATA LINE HIGH?
3453 7402 HLT /NO. LINK DATA LINE(CR2) STUCK LOW ON BUS.
3454 7006 RTL /
3455 7420 SNL /F SET LOW?
3456 7402 HLT /NO. F SET(DP2) STUCK HIGH ON BUS.
3457 7510 SPA /FETCH LINE HIGH?
3460 7402 HLT /NO. FETCH LINE(DJ2) STUCK LOW ON BUS.
3461 7006 RTL /
3462 7420 SNL /EXECUTE LOW?
3463 7402 HLT /NO. EXECUTE(DL2) STUCK HIGH ON BUS.
3464 7510 SPA /DEFER HIGH?
3465 7402 HLT /NO. DEFER(DK2) STUCK LOW ON BUS.
3466 7006 RTL
3467 7430 SEL /IR0 HIGH?
3470 7402 HLT /NO. IR0 STUCK LOW ON BUS.
3471 7510 SPA /IR1 HIGH?
3472 7402 HLT /NO. IR1 STUCK LOW ON BUS.
3473 7004 RAL
3474 7710 SPA CLA /IR2 HIGH?
3475 7402 HLT /NO. IR2 STUCK LOW ON BUS.
.....
/TEST FOR IR0,IR1,AND IR2 LOW, FEICH LOW, MA0 HIGH, EXECUTE HIGH AND LINK LOW
XOR64, NOP/JMS POINT
3476 7000 TSLINB, CLRAL
3477 6145 IAC OHL /AC TO 0001, LINK TO 1
3500 7021 CNTENA /ENABLE STATUS READING
3501 6140 CLA /STATUS LOADED AT TP3 OF THIS INSTRUCTION
3502 7200 READA /READ SIMULATOR STATUS REGISTER
3503 6141 SPA /MA0 HIGH?
3504 7510 HLT /NO. MA0 STUCK LOW ON BUS.
3505 7402 RTL
3506 7006 RTL
3507 7006 RTL
3510 7420 SNL /LINK LOW?
3511 7402 HLT /NO. LINK LINE(AV2) STUCK HIGH ON BUS.
3512 7006 RTL
3513 7500 SMA /FETCH LOW?
3514 7402 HLT /NO. FETCH (DJ2) STUCK HIGH ON BUS.
3515 7006 RTL
3516 7430 SEL /EXECUTE HIGH?
3517 7402 HLT /NO. EXECUTE(DL2) STUCK LOW ON BUS.
3520 7006 RTL

```

```

3521 7420 SNL /IR0 LOW?
3522 7402 HLT /NO. IR0(DD2) STUCK HIGH ON BUS.
3523 5775 JMP TSLN0N
3575 3600 PAGE
3576 3741 TSLN0N, SMA /IR1 LOW?
3577 6000 HLT /NO. IR1(DE2) STUCK HIGH ON BUS.
3600 7500 RAL
3601 7402 SMA CLA /IR2 LOW?
3602 7004 HLT /NO. IR2(DH2) STUCK HIGH ON BUS.
3603 7700
3604 7402
.....
/TEST FOR LINK DATA LOW ON BUS,
XOR65, NOP/JMS POINT
3605 7000 TSLING, CLRAL
3606 6145 CLA OLL IAC RAL /+2
3607 7305 LOADA /5 CYCLE DELAY
3610 6146 IAC
3611 7001 CNTENA /ENABLE STATUS READING
3612 6140 RTR /AC TO 4000
3613 7012 TAD SAVFLD /GET IF AND DF
3614 1106 NOP
3615 7000 /STATUS READ BY SIMULATOR AT TP3 THIS INSTRUCTION
3616 6005 RTR
3617 7200 CLA
3620 6141 READA /READ STATUS
3621 0051 AND K200
3622 7650 SNA CLA /LINK DATA LINE LOW?
3623 7402 HLT /NO. LINK DATA LINE(CR2) STUCK HIGH ON BUS
.....
/TEST FOR DEFER LOW ON BUS AND F SET HIGH
XOR66, NOP/JMS POINT
3624 7000 TSLIND, CLRAL
3625 6145 IAC
3626 7001 LOADA /DELAY OF 3 CYCLES
3627 6146 IAC
3630 7001 CNTENA
3631 6140 CLA
3632 7200 AND I K200 /READ STATUS AT TP3 OF DEFER
3633 0451 READA
3634 6141 AND K10
3635 0044 SNA CLA /WAS DEFER LOW?
3636 7650 HLT /NO. DEFER LINE(DK2) STUCK LOW ON BUS.
3637 7402 READA /READ STATUS AGAIN.
3640 6141 AND K100 /MASK OUT F SET.
3641 0050 SZA /F SET HIGH?
3642 7440 HLT /NO. F SET STUCK LOW ON BUS (DP2)
3643 7402
.....
/TEST FOR INT IN PROG HIGH
XOR67, NOP/JMS POINT
3644 7000 TSLINF, CLRAL
3645 6145 TAD K200
3646 1051 IAC
3647 7001 LOADA /DELAY OF THREE CYCLES
3650 6146 TAD KJMPI0
3651 1334

```

```

3652 3036 DCA CHNCON
3653 7001 IAC
3654 6140 CNTENA /ENABLE STATUS CHECKING
3655 7200 CLA
3656 7000 NOP /INTERRUPT AT END OF THIS INSTRUCTION
3657 1265 TAD KHALT /RETURN HERE FROM INTERRUPT
3660 3036 DCA CHNCON /SET UP FOR UNEXPECTED INTERRUPT
3661 6001 IOV
3662 6141 READA /READ STATUS FROM SIMULATOR
3663 0131 AND K1000 /MASK OUT INT IN PROG
3664 7650 SNA CLA /INT IN PROG HIGH?
3665 7402 KHALT, HLT /NO. INT IN PROG STUCK LOW ON BUS. (BPZ)
/.....
3666 7000 /TEST THAT CPMA DISABLE LOW DISABLES MA+1 TO PC
XOR68, NOP/JMS POINT
TSCPDS, CLRAL
3670 1336 TAD KCPHRE
3671 3007 DCA 7 /SET UP INTERRUPT RETURN
3672 1342 TAD K4201
3673 6146 LOADA /CPMA DISABLE PULSE, INTERRUPT, 3 CYCLE DELAY
3674 1335 TAD KJMP17
3675 3036 DCA CHNCON /SET UP FOR EXPECTED INTERRUPT
3676 6140 CNTENA
3677 7000 NOP
3700 4301 JMS ,+1 /CYCLE MASTER
/CPMA DISABLE PULSE OCCURS DURING EXECUTE
/PREVENTING MA+1 TO PC
3701 0000 CPSTOR, 0
3702 7402 HLT /DID NOT INTERRUPT
3703 1302 CPHRET, TAD ,+1
3704 3036 DCA CHNCON /SET UP FOR UNEXPECTED INTERRUPT
3705 1000 TAD 0
3706 6001 IOV
3707 1333 TAD KCPSTO
3710 7440 SEA /HAS PC AS EXPECTED?
3711 7402 HLT /NO. CPMA DISABLE DID NOT PREVENT MA+1 TO PC
/.....
3712 7000 /TEST FOR HALT AT END OF PASS
XOR69, NOP/JMS POINT
FINALE, DCA PARTND /GO BACK TO REGULAR TYPE POWER FAIL RECOVERY
3713 3100 TAD 21 /GET HARDWARE DESIGNATOR
3714 1021 TAD 21 /USE FRONT PANEL?
3715 7700 SNA CLA /NO. USE PSEUDO SR,
3716 5321 JMP ,+3 /YES, USE FRONT PANEL SR,
3717 7404 OSR
3720 7410 SKP
3721 1020 TAD 20 /GET PSEUDO SR,
3722 0052 AND K400 /MASK OUT STOP BIT,
3723 7640 SEA CLA /HALT ON PASS COMPLETE?
3724 7402 7402/HLT /YES.
3725 7000 XOR70, NOP/JMS POINT
3726 1021 TAD 21 /GET HARDWARE DESIGNATOR
3727 0050 AND K100 /MASK OUT XOR BIT
3730 7640 SEA CLA /RUNNING ON XOR?
3731 5740 JMP I XRPONT/CLA /YES. (CLA IN 1K VERSION)
3732 5737 LEAVE, JMP I KSTOVR /NO. DO TEST AGAIN,
3733 4077 KCPSTO, -CPSTOR

```

```

3734 5400 KJMP10, JMP I 0
3735 5407 KJMP17, JMP I 7
3736 3703 KCPHRE, CPHRET
3737 0222 KSTOVR, TSCACL/0200 FOR 1K VERSION
3740 4000 XRPONT, XORSTT/(LEAVE) FOR 1K VERSION
3741 7776 H2, 7776
3742 4201 K4201, 4201
3743 7773 H3, 7773
IFDEF ACT0E <
/THIS CODE IS USED ON THE ACT-0/E LINE ONLY
ACTCHK, TAD 22 /GET HARDWARE WORD #2
RTL /QUICK VERIFY BIT TO AC0
SPA CLA /QUICK VERIFY?
JMP ACTBAK /YES, REPORT SUCCESSFUL PASS.
/NOT QUICK VERIFY, MUST TIME OUT 10 MINUTES
CLL
ISE ACTNTR /BUMP TIMER
JMP I K200 /NO OVERFLOW, DO ANOTHER PASS.
JMP ACTBAK /OVERFLOW, REPORT GOOD RUN.
/10 MINUTES WORTH OF CPU TEST>
ACTNTR, 3240
IFDEF ACT8A <
IFDEF ONEKP2 <
ACTSLO, 7777
/THIS CODE IS USED ON THE ACT-8/A LINE ONLY
ACT0KK, CLL:
TAD KSTOVR
AND K77
SEA CLA /RUNNING 1K?
JMP ACTBAK /NO, REPORT IMMEDIATELY.
ISE ACTSLO
JMP I KSTOVR /SLOW DOWN ACT REPORTING
TAD K7700
DCA ACTSLO
JMP ACTBAK>>>
/.....
4000 /XOR INITIALIZATION CODE
*4000
4000 0000 XORSTT, 0
4001 6173 STIP /MUT POWER ON?
4002 7410 SKP /NO. DO CORE SWAP AND INITIALIZATION PASS
4003 5217 JMP XORNIT /YES, SKIP INITIALIZING AND CORE CHANGING
/.....
/START INITIALIZING THE MUT AND THE KQM
/AND CHANGE PROGRAM FOR XOR
4004 6170 XORN DCA /POWERS MUT AND BEGINS INITIALIZATION
4005 1377 TAD (XORTAB
4006 3112 DCA TESLOC
4007 1912 DOMORE, TAD I TESLOC
4010 7400 SNA /ALL DONE?
4011 5451 JMP I K200 /YES, DO ONE PASS OF TEST FOR SYNC PURPOSES
4012 3116 DCA POINTC /NO. SAVE POINTER.
4013 1376 TAD (JMS POINT /GET XOR HOOK INSTRUCTION

```

```

4014 3516          DCA I POINTC
4015 2112          ISE  TESLOC
4016 5207          JMP  OOMORE
          /ENABLE "XOR"ING
4017 6007          XORNIT, CAF
4020 7621          CAM
4021 6171          SKXR
4022 6170          XRON          /START "XOR"ING PROCESS
4023 6007          CAF
4024 7621          CAM
4025 1375          TAD          (XORNIT+4
4026 3173          DCA  POINTX
4027 4156          JMS  POINT          /CHECK FOR STATIC ERROR
          /IF A STATIC ERROR EXISTS, RETURN IS MADE
          /TO (XORNIT+4)
          /TURN ON XOR INTERRUPT, NO STATIC ERROR,
          /BEGIN TEST,
          .....
4030 6174          XRSI
4031 5451          JMP I K200
          /*****
/XOR HOOK TABLE
XORTAB, EXHLT1          /THESE LOCATIONS ARE CHANGED TO
          /JMS POINT FOR XOR USE
4032 0221          XORT03
4033 0200          XORT04
4034 0266          XORT05
4035 0320          XORT06
4036 0346          XORT07
4037 0423          XORT08
4040 0447          XORT09
4041 0503          XORT10
4042 0530          XORT11
4043 0600          XORT12
4044 0643          XORT13
4045 1343          XORT14
4046 0667          XORT15
4047 1674          XORT16
4050 0752          XORT17
4051 1020          XORT18
4052 1056          XORT19
4053 1117          XORT20
4054 1167          XORT21
4055 1206          XORT22
4056 1241          XORT23
4057 1335          TRNXPT
4060 1361          TS1FBE
4061 1714          XORT27
4062 2200          XORT28
4063 2236          XORT29
4064 1137          XORT30
4065 1745          XORT31
4066 1767          XORT32
4067 2312          XORT33
4070 2332          XORT34
4071 2400          XORT35
4072 2414          XORT36
4073 2435          XORT37
4074 2443          XORT38
4075 2456          XORT38

```

```

4076 2502          XOR39
4077 2536          XOR40
4100 2552          XOR41
4101 2603          XOR42
4102 2617          XOR43
4103 2623          XOR44
4104 2635          XOR45
4105 2262          XOR46
4106 2275          XOR47
4107 2470          XOR48
4110 2665          XOR49
4111 2720          XOR50
4112 2751          XOR51
4113 3012          XOR52
4114 3037          XOR53
4115 3065          XOR54
4116 3127          XOR55
4117 3204          XOR56
4120 3223          XOR57
4121 3264          XOR58
4122 3300          XOR59
4123 3347          XOR60
4124 3415          XOR62
4125 3426          XOR63
4126 3476          XOR64
4127 3605          XOR65
4130 3624          XOR66
4131 3644          XOR67
4132 3666          XOR68
4133 3712          XOR69
4134 3725          XOR70
4135 0000          0          /END OF XOR TABLE
4179 4023
4176 4156
4177 4032          FIELD 0
0000

```

```

0000 11111111 11011111 11111111 11111111 11111111 11111111 11111111 11111111
0100 11111111 11111111 11111111 11111111 11000000 00000011 11111111 11111001

0200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0300 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

0400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0500 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

0600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0700 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 11110001

2000
2100

2200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2300 11111111 11111111 11111111 11111111 11111111 11111100 00000000 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 11111110 01111111

3000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3100 11111111 11111111 11111111 11111111 11111111 11111111 11000000 11111111

3200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3300 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

3400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3500 11111111 11111111 11110000 00000000 00000000 00000000 00000000 00000111

3600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3700 11111111 11111111 11111111 11111111 11110000 00000000 00000000 00000000

```

```

4000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
4100 11111111 11111111 11111111 11111100 00000000 00000000 00000000 00000111

4200
4300

4400
4500

4600
4700

5000
5100

5200
5300

5400
5500

5600
5700

6000
6100

6200
6300

6400
6500

6600
6700

7000
7100

7200
7300

7400
7500

7600
7700

```

.....  
/THIS CODE WILL ALWAYS BE LOADED TO FIELD 8 FOR INTERRUPT PROCESSING  
/PURPOSES, REGARDLESS OF WHAT FIELD THE PROGRAM IS LOADED INTO.

```

0000 0000 *0
0000 7402 HLT
0001 6244 RMF
0002 5023 JMP SKPCHN

0142 0142 *PWRAOD
0142 1076 PWRUP, TAD SAVHQ
0143 7421 HQL
0144 1075 TAD SAVFLG
0145 6005 RTP
0146 7200 CLA
0147 1194 TAD PPRTWO
0150 7640 SEA CLA
0151 5141 JMP THOBK
0152 1074 TAD SAVAC
0153 5477 JMP I SAVRET
0154 0100 PPRTWO, PARTWO
S

```

```

0000 11100000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0100 00000000 00000000 00000000 00000000 00111111 11111000 00000000 00000000

```

- 0200
- 0300
- 0400
- 0500
- 0600
- 0700
- 1000
- 1100
- 1200
- 1300
- 1400
- 1500
- 1600
- 1700
- 2000
- 2100
- 2200
- 2300
- 2400
- 2500
- 2600
- 2700
- 3000
- 3100
- 3200
- 3300
- 3400
- 3500
- 3600
- 3700

4000  
4100  
  
4200  
4300  
  
4400  
4500  
  
4600  
4700  
  
5000  
5100  
  
5200  
5300  
  
5400  
5500  
  
5600  
5700  
  
6000  
6100  
  
6200  
6300  
  
6400  
6500  
  
6600  
6700  
  
7000  
7100  
  
7200  
7300  
  
7400  
7500  
  
7600  
7700

ACDATA	0107	DOMORE	4007	K2	0042	M4200	2705
ACL	7701	DOSIMU	1461	K20	0045	M5	3743
ACLTST	0725	DOSMAP	1570	K200	0051	M5B	3371
ACTINT	2220	EXMLT1	0221	K2000	0055	HQA	7501
ACWAS	0101	FINALE	3713	K201	3027	MQDATA	0110
ADDRER	0450	FSETER	2344	K220	3372	NQL	7421
AIBIT0	1140	GDRETEU	3056	K221	3157	NQLTST	0044
AIBIT4	1156	GETEXP	1651	K2525	0056	MQWAS	0102
AIBIT8	1150	GETFND	1660	K2527	3100	NXTONE	1340
AIIDTS	1126	GR1SIM	1467	K2552	0057	NXTONN	1743
AINIT5	1120	GR2GR3	1507	K3000	2736	NXTPAT	1360
ANDSTS	1032	GROUP2	1511	K3333	0060	NXTPTT	1742
ANISTS	1106	GROUP3	1554	K4	0043	OUT	1527
ANIZTS	1054	GTF	6004	K40	0046	PARTWO	0100
ASYNCB	3205	GTFTS1	1170	K400	0052	PAVFLG	0136
BACKAD	2653	GTFTS2	1215	K4000	0061	PAVRET	0140
BAKADB	2706	GTFTS3	2326	K410	2353	PCTST2	2516
BAKADC	2737	HLT	7402	K4201	3742	POINT	0156
BAKADD	3000	HLTCDD	2532	K440	2704	POINTB	0115
BIT11	0112	HWRDES	0021	K442	2702	POINTC	0116
BIT3	0003	INSTRT	1372	K444	2567	POINTD	0117
BIT4	0004	INSTRT	0122	K4444	0062	POINTE	0120
BIT5	0005	INTADD	2425	K446	2570	POINTF	0121
BIT6	0117	INTEN1	1255	K5000	2703	POINTG	0113
BIT7	0007	IOF	6002	K5225	0063	POINTX	0173
BIT8	0006	IOFHLT	2311	K5252	0064	PONTXA	0174
BKFLOW	3130	IOFTS1	1247	K5253	0065	POHRDN	0125
BKIN17	3146	ION	0001	K6000	2355	PPRTWO	0154
BKINTO	3200	IONTS1	1240	K6666	0066	PSAVMO	0137
BSW	7002	ISDSTS	1021	K7000	0034	PWRADD	0142
BSWDRP	0623	ISISTS	1102	K7600	1366	PWRUP	0142
BSWPCK	0633	ISZTS	1052	K77	0047	RARTST	0531
CAF	6007	ISZETS	0757	K7700	0067	RDF	6214
CAM	7621	IZITS	2503	K7721	0070	READA	6141
CAMTST	0712	JMPTS1	1301	K7770	0071	RETINS	0073
CHNCON	0036	JMPTS2	1320	K7777	0072	RETURN	2531
CKSHIT	0124	JMSL02	1320	KOPHRE	3736	REYSEN	1540
CLRAL	6145	JMSL0C	1311	KOPST0	3733	RIF	6224
CNTENA	6140	JMST51	1307	KHALT	3685	ROBHLT	3025
CPHRET	3703	JMST52	1323	KJMP10	3734	ROBRET	3030
CPSTOR	3701	JMST53	2230	KJMP14	2354	ROMB15	3013
DATPAT	0177	K0	0040	KJMP17	3735	ROMCLR	3054
DBOUT	3066	K1	0041	KSTOP	0135	ROMH04	2471
DCAZTS	0737	K10	0044	KSTOVR	3737	ROMRTN	3400
DDDSTS	1007	K100	0050	LEAVE	3732	ROMTST	1336
DCISTS	1072	K1000	0131	LKDATA	0111	RTF	6005
DCIZTS	1044	K101	2424	LKWA5	0103	RTFTS1	1207
DFEBEA	2451	K1111	0053	LOADA	0146	RTFTS2	1262
DISERR	1646	K12	3271	LOADB	0147	RTLDRP	0006
DLVHLT	2413	K17	3161	LOPBK	1792	RTLPCK	0016
DMARET	3336	K1777	0054	M2	3741	RTRPCK	0001
DOAGAN	1670	K17B	3373	M200	2706	SAVAC	0074

SAVFLQ	0106	TSINLN	3265	XOR29	1745
SAVFLG	0075	TSINTO	2401	XOR30	1767
SAVMD	0076	TS_LNB	3477	XOR31	2312
SAVRET	0077	TS_LNC	3606	XOR32	2250
SETBK	6143	TS_LND	3625	XOR33	2332
SETSIH	1421	TS_LNE	3427	XOR34	2400
SGT	0006	TS_LNF	3645	XOR35	2414
SGTTST	1223	TS_LNCN	3600	XOR36	2435
SIMERR	1643	TS_MMS	3416	XOR37	2443
SKON	0000	TSROMA	3040	XOR38	2496
SKONT2	1244	TSTCHI	2237	XOR39	2502
SKPCHK	1623	TSTCHX	1772	XOR40	2536
SKPCHN	0023	TSTCLB	2251	XOR41	2552
SKPQFV	6144	TSTCLC	2263	XOR42	2603
SKPPED	0104	TSTCLD	2276	XOR43	2617
SKXR	6171	TSTCLE	2636	XOR44	2623
SOMHOR	3256	TSTCLF	2666	XOR45	2635
SOMSKP	0105	TSTCLG	2721	XOR46	2262
SPL	6102	TSTCLH	2721	XOR47	2275
SRQ	0003	TSTCLL	0251	XOR48	2470
SRQTS1	1274	TSTCLM	0257	XOR49	2665
START	0200	TSTFBE	2200	XOR50	2720
STIP	6173	TSTIAC	0237	XOR51	2751
STOVER	1771	TSTINT	2415	XOR52	3012
STROB	6142	TSTLCH	0244	XOR53	3037
SWITCH	0020	TSTFCI	2426	XOR54	3085
SWP	7521	TSTSRQ	2324	XOR55	3127
SWPTST	0650	TW0BAK	0141	XOR56	3204
SXRC	6175	TW0K	0000	XOR57	3223
TDDSTS	1001	XOR03	0266	XOR58	3264
TDISTS	1064	XOR04	0320	XOR59	3300
TDIZTS	1040	XOR05	0346	XOR60	3347
TESAGG	1744	XOR06	0423	XOR62	3415
TESAGN	1356	XOR07	0447	XOR63	3426
TESLOC	0112	XOR08	0503	XOR64	3476
TEST	0123	XOR09	0530	XOR65	3605
TESTPT	1671	XOR10	0600	XOR66	3624
TESTS	1577	XOR11	0643	XOR67	3644
TRNXPT	1714	XOR12	1343	XOR68	3666
TSYBRK	3224	XOR13	0667	XOR69	3712
TSBKAD	3114	XOR14	1674	XOR70	3725
TSBKTO	3103	XOR15	0752	XORBAK	0166
TSBSSK	0227	XOR16	1020	XORNI1	4017
TSCACL	0222	XOR17	1056	XORPN1	1773
TSCPD5	3667	XOR18	1117	XORST1	4000
TSDMIT	3301	XOR19	1167	XORTAB	4032
TSDPNS	3350	XOR20	1206	XRC1	6172
TSFSET	2333	XOR21	1241	XRON	6170
TSINDB	2553	XOR22	1335	XRPON1	3740
TSINDC	2600	XOR23	1361	XRS1	6174
TSINDD	2624	XOR27	2236	XRT0	6176
TSINDL	2537	XOR28	1137		

ERRORS DETECTED: 0  
 LINKS GENERATED: 9  
 RUN-TIME: 19 SECONDS  
 3K CORE USED





