

IDENTIFICATION

PRODUCT CODE: MAINDEC-08-DKVTB-A-D
PRODUCT NAME: VT78 CPU DIAGNOSTIC
PRODUCT DATE: JULY 1977
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: BARRY S. POLAND

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 DOCUMENT.

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) 1977, BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	HARDWARE
2.2	SOFTWARE
2.3	STORAGE
2.4	PREREQUISITE SOFTWARE
3.0	RELATED DOCUMENTS
4.0	RESTRICTIONS
5.0	OPERATING INSTRUCTIONS
5.1	LOADING THE PROGRAM
5.2	PPROGRAM OPTIONS
5.2.1	SWITCH REGISTER SETTINGS
5.2.2	HARDWARE CONFIGURATION SETTINGS
5.3	CONSOLE TERMINAL PACKAGE
5.4	EXECUTION TIMES
6.0	ERROR INFORMATION
7.0	SUB-TEST SUMMARIES
7.1.1	CPU TESTS
7.1.2	MEMORY EXTENSION TESTS
7.1.3	REAL TIME CLOCK TESTS
7.1.4	SERIAL LINE UNIT TESTS
7.1.5	PARALLEL INTERFACE TESTS
7.1.6	FLOPPY INTERFACE TESTS
8.0	PROGRAM LISTING

1.0 **ABSTRACT**

THIS DIAGNOSTIC WILL TEST ALL THE LOGIC ON THE VT78 PROCESSOR MODULE THAT CAN BE TESTED VIA PROGRAM INSTRUCTION. THE MODULE LOGIC TESTED INCLUDES THE PDP-8 CPU, MEMORY EXTENSION CONTROL, FLOPPY INTERFACE, THREE SERIAL LINE UNITS (SLU) AND THE PARALLEL I/O INTERFACE. THE DIAGNOSTIC IS ORGANIZED WITH A SOFTWARE MODULE FOR EACH OF THESE BASIC HARDWARE COMPONENTS. THE MAIN MEMORY FOR THIS PROCESSOR IS CONTAINED ON A SEPARATE "DAUGHTER" MODULE AND IS TESTED VIA THE VT78 MOS MEMORY DIAGNOSTIC (MAINDEC-08-DKVTA-A).

THE STANDARD PDP-8 CONSOLE PACKAGE WITH SOFTWARE CONTROLLED SWITCH REGISTER HAS BEEN INCORPORATED INTO THIS DIAGNOSTIC TO ALLOW THE PROGRAM TO RUN ON THE VT78 SYSTEM WHICH HAS NO HARDWARE SWITCH REGISTER. THIS PACKAGE PROVIDES AN INTERFACE BETWEEN THE USER AND THE DIAGNOSTIC VIA THE VIDEO DISPLAY/KEYBOARD TERMINAL.

THE VT78 PROCESSOR MODULE CONTAINS A SMALL AMOUNT OF LOGIC WHICH CAN NOT BE TESTED WITHOUT EXTERNAL DEVICES ATTACHED TO THE VT78 PERIPHERAL PORTS. FOR MANUFACTURING PURPOSES, A SPECIAL PERIPHERAL SIMULATOR PACKAGE HAS BEEN DESIGNED TO PLUG INTO THE VT78 PORTS AND PROVIDE DIAGNOSTIC ACCESS TO ALL MODULE LOGIC. THIS DIAGNOSTIC IS COMPATIBLE WITH THE HARDWARE SIMULATOR. ALL OTHER USERS OF THE DIAGNOSTIC WHICH DO NOT HAVE THE HARDWARE SIMULATOR, CAN FULLY TEST THE MODULE BY RUNNING BOTH THIS DIAGNOSTIC AND THE PERIPHERAL DIAGNOSTICS WITH RESPECTIVE PERIPHERALS ATTACHED TO VT78 SYSTEM.

THE PROGRAM IS COMPATIBLE WITH THE PDP-8 APT TEST SYSTEM.

THIS DIAGNOSTIC WILL RUN ONLY ON THE VT78 SYSTEM - IT WILL NOT RUN ON ANY OTHER PDP-8 SYSTEM.

2.0 **REQUIREMENTS**

2.1 **HARDWARE**

THE FOLLOWING HARDWARE IS REQUIRED FOR EXECUTION OF THIS PROGRAM:

VT78 VIDEO DATA PROCESSOR (INCLUDES PROCESSOR MODULE, 16K
MEMORY MODULE, SIGNAL DISTRIBUTION BOARD,
VIDEO DISPLAY AND KEYBOARD)

OPTIONAL EQUIPMENT INCLUDES THE PERIPHERAL SIMULATOR, THE RX01 DISKETTE SUBSYSTEM, AND THE LQP OR LA180 PRINTER.

NOTE: THE VIDEO DISPLAY AND KEYBOARD ARE NOT REQUIRED FOR TESTING UNDER APT CONTROL.

2.2 SOFTWARE -----

THE SOFTWARE ENVIRONMENTS IN WHICH THIS PROGRAM WILL EXECUTE INCLUDE:

- 1) STAND ALONE
- 2) UNDER OS/8 CONTROL
- 3) UNDER APT CONTROL

2.3 STORAGE -----

THE PROGRAM OCCUPIES 8K OF MEMORY(2 FIELDS) AND MUST BE LOADED INTO FIELDS 0 AND 1.

2.4 PREREQUISITE SOFTWARE -----

THE PANEL HANDLER AND BOOTSTRAP PORTIONS OF THE RESIDENT CONTROL PANEL PROGRAM MUST RUN SUCCESSFULLY TO PERMIT LOADING OF THIS DIAGNOSTIC. THE RESIDENT DIAGNOSTIC SHOULD ALSO BE RUN PRIOR TO ATTEMPTING TO EXECUTE ANY VT78 SYSTEM DIAGNOSTICS.

3.0 RELATED DOCUMENTS -----

VT78 HARDWARE SPECIFICATION
STANDARD APT SYSTEM TO PDP-8 DIAGNOSTIC INTERFACE
MR78BA USER INFORMATION DOCUMENT
MR78BB USER INFORMATION DOCUMENT

4.0 RESTRICTIONS -----

DURING MUCH OF THE SLU TEST SECTION OF THE DIAGNOSTIC, THE SLU WHICH INTERFACES TO THE VIDEO DISPLAY/KEYBOARD IS LOOPED UPON ITSELF. WHILE LOOPED, ANY ATTEMPT BY THE OPERATOR TO GAIN CONTROL VIA KEYBOARD INTERVENTION IS BLOCKED. THE OPERATOR CAN REGAIN SYSTEM CONTROL AT ANY TIME THROUGH USE OF THE START BUTTON ON THE SIDE OF THE VT78.

5.0 OPERATING INSTRUCTIONS

5.1 LOADING THE PROGRAM

THE DIAGNOSTICS CAN BE LOADED THROUGH ANY ONE OF THE AVAILABLE EXTERNAL PORTS; FLOPPY INTERFACE, PARALLEL I/O INTERFACE, SLU INTERFACE, OR PROGRAM INJECTOR INTERFACE. CURRENTLY LOADING IS SUPPORTED ONLY THROUGH THE FLOPPY INTERFACE AND SLU INTERFACE.

5.1.1 LOADING FROM FLOPPY DISKETTE

THIS DIAGNOSTIC IS PROVIDED ALONG WITH A SYSTEM MONITOR ON FLOPPY DISKETTE AND WILL NORMALLY BE LOADED FROM THIS MEDIUM. TO LOAD THE SYSTEM MONITOR SIMPLY INSERT THE DISKETTE CONTAINING THE DIAGNOSTIC INTO EITHER DRIVE RXA0 OR RXA1 AND PRESS THE VT78 START BUTTON. THE SYSTEM WILL RESPOND ON THE VIDEO DISPLAY WITH A START MESSAGE FOLLOWED BY A PROMPT CHARACTER. TO CALL AND START THE DIAGNOSTIC TYPE R DKVTBA FOLLOWED BY THE RETURN KEY. THE SYSTEM WILL RESPOND WITH A DISPLAY OF THE PROGRAM NAME, MAINDEC NUMBER, PSEUDO SWITCH REGISTER (PSR), AND HARDWARE CONFIGURATION WORD (HW3) AND WAIT FOR RUN MODE SELECTION BY THE OPERATOR. RUN MODE IS SELECTED BY SETTING THE PSR AND HW3 AS DESCRIBED IN SECTION 5.2.1.

5.1.2 LOADING FROM SLU INTERFACE

SLU #3 IS USED FOR LOADING THE DIAGNOSTIC WHEN THE VT78 SYSTEM IS RUNNING UNDER APT CONTROL. THIS TYPE OF LOAD REQUIRES USE OF THE MR78BA EXTERNAL PROGRAM INJECTOR MODULE. WITH VT78 POWER OFF ATTACH THE MR78BA TO THE PROGRAM INJECTOR CONNECTOR ON THE BACK OF THE VT78. ALSO ATTACH THE APT TEST LINE TO SLU #3. TURN POWER ON AND ALLOW THE RESIDENT DIAGNOSTIC TO RUN TO COMPLETION. THE VT78 IS NOW READY FOR DIAGNOSTIC PROGRAM LOADING FROM THE APT HOST PROCESSOR. THE VT78 KEYBOARD MUST NOT BE TOUCHED WHILE THE VT78 IS UNDER APT CONTROL. APT PROCEDURES FOR PROGRAM LOADING SHOULD NOW BE FOLLOWED.

SLU #3 IS ALSO USED FOR LOADING THE DIAGNOSTIC FROM PAPER TAPE BY FIELD SERVICE. THIS TYPE OF LOAD REQUIRES USE OF EITHER THE MR78BA OR MR78BB EXTERNAL PROGRAM INJECTOR MODULE. WITH VT78 POWER OFF, ATTACH THE MR78BA OR MR78BB TO THE PROGRAM INJECTOR CONNECTOR ON THE BACK OF THE VT78. ALSO ATTACH THE PRS01 PAPER TAPE READER TO SLU #3. TURN POWER ON AND ALLOW THE RESIDENT DIAGNOSTIC TO RUN TO COMPLETION. THE VT78 IS NOW IN ODT MODE READY TO ACCEPT ODT COMMANDS FROM THE VT78 KEYBOARD. PROCEDURES DETAILED IN THE RESPECTIVE MR78 DOCUMENT SHOULD BE FOLLOWED TO LOAD DIAGNOSTICS.

5.2 PROGRAM OPTIONS -----

SEVERAL OPTIONS RELATING TO THE RUN MODE OF THE DIAGNOSTIC ARE AVAILABLE TO THE OPERATOR. THE OPERATOR SELECTS BETWEEN THE VARIOUS OPTIONS BY CHANGING THE PSR AFTER DIAGNOSTIC STARTUP (SECTION 5.2.1).

SEVERAL OPTION SWITCHES RELATING TO THE HARDWARE CONFIGURATION OF THE VT78 SYSTEM UNDER TEST ARE AVAILABLE FOR OPERATOR SELECTION (SECTION 5.2.2).

5.2.1 SWITCH REGISTER SETTINGS -----

FOR NORMAL DIAGNOSTIC EXECUTION ALL SWITCH REGISTER BITS ARE SET = 0.

BIT 0 = 1	INHIBIT ERROR HALT
BIT 1 = 1	LOOP ON ERROR
BIT 2 = 1	LOOP ON TEST
BIT 3 = 1	HALT ON COMPLETION OF A PROGRAM PASS
BIT 4 = 1	INHIBIT ERROR TYPEOUT
BIT 5 = 1	DON'T RUN CPU TEST
BIT 6 = 1	DON'T RUN MEMORY EXTENSION TEST
BIT 7 = 1	DON'T RUN SLU TEST
BIT 8 = 1	DON'T RUN RX01 INTERFACE TEST
BIT 9 = 1	DON'T RUN PARALLEL I/O INTERFACE TEST
BIT 10 = 1	DON'T RUN REAL TIME CLOCK TEST
BIT 11 = 1	RUN BAUD RATE SWITCH TEST

SWITCH REGISTER BITS 0 THROUGH 4 RELATE TO THE ACTION TAKEN WHEN THE PROGRAM DETECTS AN ERROR.

SWITCH REGISTER BITS 5 THROUGH 10 PERMIT THE OPERATOR TO INHIBIT TESTING PARTS OF THE CPU MODULE. NORMAL TEST SEQUENCING WITH BITS 5-10 ALL ZERO IS: CPU TESTING -- MEMORY EXTENSION TESTING -- REAL TIME CLOCK TESTING -- SLU TESTING -- PARALLEL I/O TESTING -- FLOPPY INTERFACE TESTING.

SWITCH REGISTER BIT 11 PERMITS CHECKOUT OF THE SLU #2 BAUD RATE SWITCH LOCATED ON THE BACK OF THE VT78. THIS TEST REQUIRES MANUAL INTERVENTION AND IS THEREFORE NOT PART OF NORMAL DIAGNOSTIC EXECUTION. WITH BIT 11 OF THE PSR SET THE "BAUD RATE SWITCH TESTING" WILL OCCUR IMMEDIATELY FOLLOWING "REAL TIME CLOCK TESTING." DURING THIS TEST THE BAUD RATE SWITCH SETTING WILL BE DISPLAYED. THE OPERATOR MUST CHANGE THE SWITCH SETTING AND PRESS THE SPACE BAR TO DISPLAY EACH NEW SETTING. BAUD RATE SWITCH TESTING TERMINATES UPON TYPING OF EITHER A NORMAL CONSOLE PACKAGE CONTROL CHARACTER OR A RETURN. RETURN SIMPLY CONTINUES THE NORMAL DIAGNOSTIC SEQUENCE.

5.2.2 HARDWARE CONFIGURATION SETTINGS (HW3)

WARNING--THESE SETTINGS MUST BE SELECTED TO BE CONSISTANT WITH THE PARTICULAR HARDWARE CONFIGURATION UNDER TEST - OTHERWISE FALSE ERROR MESSAGES MAY RESULT.

BIT 0 = 1 FLOPPY DRIVE 0 NOT READY
BIT 1 = 1 NO RX01 CABLED TO VT78 SYSTEM
BIT 2 = 1 NO EXTERNAL DEVICE ATTACHED TO PARALLEL
INTERFACE
BIT 3 = 1 PARALLEL DEVICE ATTACHED AND USING DEVICE
CODE 50. (LA180=DEVICE CODE 66;
LOP=DEVICE CODE 50)
WARNING--POWER MUST BE TURNED OFF THE PARALLEL
I/O DEVICE BEFORE RUNNING THIS DIAGNOSTIC
BIT 4 = 1 PERIPHERAL SIMULATOR ATTACHED TO VT78
BIT 5 = 1 EXTERNAL WRAPS ATTACHED TO SLU #2 AND SLU #3

HW3 = 0000 FOR A VT78 SYSTEM CONFIGURED WITH AN LA180 PRINTER (POWERED DOWN)
AND A RX01 FLOPPY; FLOPPY DRIVE RXA0 CONTAINS A
DISKETTE.

HW3 = 7000 FOR A VT78 PROCESSOR WITH NO PERIPHERALS ATTACHED.

HW3 = 1000 FOR A VT78 SYSTEM CONFIGURED WITH A RX01
FLOPPY; FLOPPY DRIVE RXA0 CONTAINS A DISKETTE.

5.3 CONSOLE TERMINAL PACKAGE

THIS SOFTWARE PROVIDES A MEANS FOR THE OPERATOR TO COMMUNICATE WITH THE DIAGNOSTIC.

CONTROL G

THIS IS THE CONTROL CHARACTER TO OPEN THE PSEUDO SWITCH REGISTER WHEN CONTROL G IS TYPED THE PROGRAM IS INTERRUPTED AND SR=XXXX IS TYPED. XXXX IS THE PRESENT CONTENTS OF THE PSEUDO SWITCH REGISTER. THE OPERATOR CAN NOW CHANGE THE SETTING BY ENTERING A NEW SET OF NUMBERS, OR NOT CHANGE IT BY TYPING IN A TERMINATING CHARACTER. WHEN THE PROGRAM RECOGNIZES A CONTROL G IT WILL TYPE A UP ARROW THE A G TO SIGNAL OPERATOR IT IS RESPONDING TO A CONTROL G.

EXAMPLE:

```
TYPE CONTROL G
^G                               /ECHO CONTROL G
SR=XXXX                         /XXXX IS PRESENT SWITCH REGISTER
```

TERMINATING CHARACTERS

CARRIAGE RETURN

THIS WILL RESTORE THE PSEUDO SWITCH REGISTER WITH A NEW VALUE IF ONE WAS ENTERED OR KEEP THE OLD VALUE IF NO NUMBERS WERE TYPED IN. THE PROGRAM WILL THEN RETURN TO THE POINT AT WHICH IT WAS INTERRUPTED AND RESUME OPERATION.

EXAMPLE:

```
CONTROL G TYPED IN
^G                               /ECHO CONTROL G
SR=1000 (C.RET)                /DO NOT CHANGE SWITCH REG
                                /CONTINUE PROGRAM
```

EXAMPLE:

```
CONTROL G TYPED IN
^G                               /ECHO CONTROL G
SR=1000 2000 (C, RET)         /CHANGE SWITCH REG TO 2000
                                /CONTINUE PROGRAM
```

LINE FEED

A LINE FEED WILL RESTORE THE PSEUDO SWITCH REGISTER WITH THE NEW VALUE TYPED IN OR IF NO NUMBERS WERE ENTERED RESTORE THE OLD VALUE. THE PROGRAM WILL THEN RETURN TO THE BEGINNING OF THE PROGRAM.

EXAMPLE:

```
TYPE IN CONTROL G
^G                               /ECHO CONTROL G
SR=1000 (LINE FEED)           /KEEP SWITCH REGISTER
                               /RESTART PROGRAM
```

EXAMPLE:

```
TYPE IN CONTROL G
^G                               /ECHO CONTROL G
SR=1000 3000 (LINE FEED)       /CHANGE SWITCH REG
                               /TO 3000. RESTART PROG
```

ILLEGAL CHARACTERS

ANY CHARACTER THAT IS NOT A C, RET, LINE FEED OR A NUMBER FROM 0 TO 7 IS ILLEGAL. ALL ILLEGAL NUMBERS WILL BE TYPED FOLLOWED BY A "?" QUESTION MARK. THE SWITCH REGISTER WILL BE RETYPED WITH THE ORIGINAL CONTENTS DISPLAYED.

EXAMPLE:

```
TYPE IN CONTROL G
^G                               /ECHO CONTROL G
SR=1000 W?                       /W TYPED IN   RETYPE LINE
SR=1000
```

CONTROL S

THIS IS A CONTROL CHARACTER TO STOP SENDING DATA TO A TERMINAL. THE CONTROL S IS NOT ECHOED.

CONTROL Q

THIS CONTROL CHARACTER IS TO RESUME SENDING DATA TO THE TERMINAL. THIS CONTROL CHARACTER IS NOT ECHOED.

CONTROL C

THIS CONTROL CHARACTER IS USED TO RETURN CONTROL BACK TO A OPERATING SYSTEM KEYBOARD MONITOR. THE OPERATING SYSTEM SELECTED FOR THE PDP-8 IS THE OS-8 SYSTEM WITH ITS BOOTSTRAP IN LOCATION 07600. THIS CONTROL CHARACTER IS ECHOED WHEN IT IS RECOGNIZED AS AS UPARROW AND THEN C.

5.4 EXECUTION TIMES

THE TIME TO RUN ONE COMPLETE PASS IS APPROXIMATELY 15 SECONDS.

6.0 ERROR INFORMATION

THE FOLLOWING MESSAGE FORMATS ARE USED IN THIS DIAGNOSTIC.

A) NORMAL ERROR DISPLAY

DKVTB-A FAILED, FIELD X
TN:XXXX PC:XXXX AC:XXXX MQ:XXXX FL:XXXX

X = FIELD WHERE PROGRAM WAS EXECUTING AT TIME OF ERROR
TN = OCTAL TEST NUMBER OF TEST EXECUTING AT TIME OF ERROR
PC = PROGRAM COUNTER AT TIME OF ERROR
AC = ACCUMULATOR AT TIME OF ERROR
MQ = MEMORY QUOTIENT AT TIME OF ERROR
FL = FLAGS AT TIME OF ERROR

IN SOME SITUATIONS THE AC AND MQ DISPLAYS ARE LOADED WITH MORE MEANINGFUL DATA THAN WHAT THEY CONTAINED AT TIME OF ERROR. FOR EXAMPLE THE AC MIGHT BE LOADED WITH AN ACTUAL TEST RESULT AND THE MQ WITH THE EXPECTED TEST RESULT. IN THESE SITUATIONS THE PROGRAM LISTING CLEARLY IDENTIFIES WHAT INFORMATION THE VARIOUS REGISTER DISPLAYS CONTAIN.

EACH OF THE FOLLOWING MESSAGES ARE DISPLAYED AS EXECUTION PROCEEDS THROUGH THE COMPLETE MODULE DIAGNOSTIC:

CPU TESTING
MEMORY EXT. TESTING
REAL TIME CLOCK TESTING
SLU TESTING
PARALLEL I/O INTERFACE TESTING.
FLOPPY INTERFACE TESTING

THESE MESSAGES SERVE TO IDENTIFY WHICH PARTICULAR LOGIC FUNCTION OF THE CPU MODULE IS BEING DIAGNOSED AT ANY TIME. THE MESSAGE PRECEDES THE ACTUAL TESTING - I.E. THE SLU TESTING MESSAGE IS PRINTED JUST PRIOR TO THE ACTUAL SLU TEST.

THE TEST NUMBER DISPLAYED UPON ERROR REFERS TO A TEST WITHIN THE LOGIC FUNCTION BEING TESTED AND IS NOT UNIQUE TO THE ENTIRE MODULE DIAGNOSTIC.

THE FLAGS DISPLAY CONTAINS THE FOLLOWING DATA:

BIT 0 LINK
BIT 2 INTERRUPT REQUEST (1=ASSERTED)
BIT 4 INTERRUPT ENABLE (1=ENABLED)
BIT 7 ISF1
BIT 8 ISF2
BIT 10 DSF1
BIT 11 DSF2

ALL OTHER BITS ALWAYS CONTAIN 0.

B) COMBINED MICROINSTRUCTION SIMULATION ERROR DISPLAY

TEST 15 WITHIN CPU TESTING TESTS COMBINED MICROINSTRUCTIONS BY COMPARING THEIR EXECUTION RESULTS WITH SIMULATED RESULTS. THE NORMAL ERROR DISPLAY FORMAT IS INADEQUATE TO FULLY DESCRIBE THESE TYPE FAILURES. INSTEAD THE FOLLOWING FORMAT IS USED FOR DATA TYPE ERRORS.

DKVTB=A FAILED, FIELD 0
TN:XXXX PC:XXXX INST:XXXX
ACTUAL AC:XXXX MQ:XXXX LK:XXXX
EXPECTED AC:XXXX MQ:XXXX LK:XXXX

INST = OCTAL MICROINSTRUCTION WHICH FAILED
LK = LINK (BIT 11)

THIS FORMAT DISPLAYS BOTH THE ACTUAL CONTENTS OF THE AC, MQ AND LINK IMMEDIATELY AFTER MICROINSTRUCTION EXECUTION AND THE EXPECTED RESULTS.

A SOMEWHAT SIMPLIER DISPLAY FORMAT IS USED WHERE THE ERROR IS A SKIP TYPE FAILURE - I.E. THE MICROINSTRUCTION SKIPPED WHEN IT SHOULD NOT HAVE OR VICE VERSA.

DKVTB=A FAILED, FIELD 0
TN:XXXX PC:XXXX INST:XXXX
AC:XXXX MQ:XXXX LK:XXXX

THE AC, MQ AND LINK DISPLAY THE CONTENTS OF THE RESPECTIVE REGISTERS JUST PRIOR TO MICRO INSTRUCTION EXECUTION.

C) SLU ERROR DISPLAY

THE NORMAL ERROR DISPLAY IS INADEQUATE TO FULLY DESCRIBE SLU FAILURES, THE FOLLOWING FORMAT IS USED.

DKVTB=A FAILED, FIELD 1
TN:XXXX PC:XXXX AC:XXXX MQ:XXXX FL:XXXX SLU:XXXX BR:XXXX
XMIT DATA:XXXX REC DATA:XXXX

SLU = FAILING SLU IDENTIFIER (1,2 OR 3)
BR = BAUD RATE
XMIT DATA = DATA TRANSMITTED (WITH SLU LOOPED ON ITSELF)
REC DATA = CORRESPONDING DATA RECEIVED

NOTE THAT IN SOME SLU FAILURE SITUATIONS THE XMIT AND RECEIVE DATA IS IRRELEVANT AND THUS OMITTED FROM THE ERROR DISPLAY.

D) UNEXPECTED INTERRUPT

THE FORMAT IS USED TO IDENTIFY THAT AN INTERRUPT OCCURRED WHICH WAS NOT EXPECTED BY THE DIAGNOSTIC AND WAS NOT CAUSED BY A CONSOLE REQUEST.

UNEXPECTED INTERRUPT - FIELD X
TN:XXXX PC:XXXX AC:XXXX FL:XXXX
FLAGS SET: X1 X2 X3 R1 R2 R3 LA LQ

THE PC DISPLAY CONTAINS THE ADDRESS OF THE INSTRUCTION BEING EXECUTED WHEN THE INTERRUPT OCCURRED.

THE FLAGS SET MESSAGE LISTS THOSE DEVICE FLAGS SET AFTER THE INTERRUPT. THE LISTED FLAG MAY OR MAY NOT HAVE CAUSED THE INTERRUPT DEPENDING ON THE STATE OF THE DEVICE INTERRUPT ENABLE. IN THE SAMPLE FORMAT ABOVE THE X REPRESENTS THE XMIT FLAG AND R THE RECEIVE FLAG ASSOCIATED WITH THE NUMBERED SLU. LA AND LQ ARE THE PARALLEL I/O INTERFACE FLAGS. THE FLOPPY FLAG IS NOT SHOWN BECAUSE READING THIS FLAG CLEARS IT.

7.0 SUB-TEST SUMMARIES

7.1.1 CPU TESTS

TN 0001 -- FIRST OPERATE MICROINSTRUCTION TEST
TN 0002 -- FIRST TEST OF MRI
TN 0003 -- TEST ADDER FUNCTION
TN 0004 -- SECOND OPERATE MICROINSTRUCTION TEST
TN 0005 -- MQ MICROINSTRUCTION TEST
TN 0006 -- TEST DCA & ISZ DIRECT ADDRESSING TO PAGE ZERO
TN 0007 -- TEST AND, TAD, ISZ, & DCA DIRECT ADDRESSING TO SAME PAGE
TN 0010 -- TEST AND, TAD, ISZ, & DCA INDIRECT ADDRESSING THRU PAGE ZERO
TN 0011 -- TEST AND, TAD, ISZ, & DCA INDIRECT ADDRESSING THRU SAME PAGE
TN 0012 -- TEST AUTO-INDEX
TN 0013 -- TEST INTERNAL IOT INSTRUCTIONS
TN 0014 -- TEST JMP AND JMS
TN 0015 -- TEST COMBINED OPERATE MICROINST. OF FORM 7XX0,7XX1.

7.1.2 MEMORY EXTENSION TESTS

TN 0001 -- TEST CDF & PDF. USE CDF TO SET DATA FIELD AND
PDF TO READ THE DATA FIELD. DO ALL COMBINATIONS
0 TO 3 & 7
TN 0002 -- TEST SAVE FIELD BITS 9-11 WITH RIB. PROGRAM
INTERRUPT IS ENABLED. RECEIVE FLAG IS USED FOR INT.
DO ALL COMBINATIONS 0-3 & 7.
TN 0003 -- TEST DCA I AND TAD I TO ALL STACKS (1-3 & 7)
TN 0004 -- CIF TEST
TN 0005 -- TEST GTF FOR FLAG & SAVE FIELDS.
TN 0006 -- TEST ION & LINK FROM RTF - TEST INTERRUPT
INHIBIT BEFORE JMP
TN 0007 -- CONFIDENCE CHECK ON FIELDS 0,1,2, & 3
TN 0010 -- TEST DF + IF FROM SAVE FIELD AFTER INTERRUPT
TEST CDI TO CHANGE BOTH DF & IF.
TN 0011 -- TEST PROGRAM INTERRUPT IN ALL EXTENDED FIELDS
TN 0012 -- TEST SF WITH AN RMF IOT
TN 0013 -- TEST THAT RMF & RTF INST. ZERO MOST SIGN. BIT OF DF REG.
TN 0014 -- UNUSED IOT TEST - VERIFIES THAT ALL UNUSED
IOTS HAVE NO EFFECT ON SYSTEM.

7.1.3 REAL TIME CLOCK TESTS

TN 0001 -- TESTS THAT CLOCK FLAG WILL SET AND THAT CAF
WILL CLEAR IT.
TN 0002 -- TESTS THAT CLCL WILL CLEAR CLOCK FLAG
TN 0003 -- TESTS THAT CLLE ENABLES & DISABLES CLOCK INTERRUPTS
TN 0004 -- TESTS THAT CAF WILL CLEAR CLOCK INT. ENABLE
TN 0005 -- TESTS THAT ALL REAL TIME CLOCK IOTS LEAVE AC
UNDISTURBED
TN 0006 -- CLOCK TIMING TEST

7.1.4 SERIAL LINE UNIT TESTS

-
- TN 0001 -- TESTS THAT CAF WILL CLEAR RECEIVE FLAG AND
DISABLE TRANSMIT FLAG
 - TN 0002 -- TEST KIE TO CLEAR SLU INTERRUPT ENABLE
TEST SPF TO SET TRANSMIT FLAG ENABLE
TEST TCF TO CLEAR TRANSMIT FLAG ENABLE
 - TN 0003 -- TEST THAT CAF CLEARS TRANSMIT FLAG ENABLE
 - TN 0004 -- TEST THAT CAF SETS SLU INTERRUPT ENABLE
TEST TSK TO SKIP AND NOT TO SKIP
 - TN 0005 -- TEST KIE TO SET & CLEAR SLU INTERRUPT ENABLE
 - TN 0006 -- (LOOPAROUND ENABLED) TEST TLS TO CLEAR TRANSMIT
FLAG & THEN SET IT WHEN TRANSMISSION OF CHAR
IS COMPLETE. ALSO TEST RECEIVE FLAG TO
SET. TESTS KSF TO SKIP ON RECEIVE FLAG AND
KCF TO CLEAR RECEIVE FLAG.
 - TN 0007 -- (LOOPAROUND ENABLED) TESTS THAT TPC PRODUCES
IDENTICAL RESULTS AS TLS
 - TN 0010 -- (LOOPAROUND ENABLED) CHECKS THAT THE TLS-TCF SEQUENCE
CLEARS TRANSMIT FLAG ENABLE PREVENTING FOLLOWING INT.
 - TN 0011 -- (LOOPAROUND ENABLED) TESTS THAT KRB WILL CLEAR THE
RECEIVE FLAG
 - TN 0012 -- (LOOPAROUND ENABLED) TEST THAT CAF WILL CLEAR
RECEIVE FLAG
 - TN 0013 -- (LOOPAROUND ENABLED) TESTS THE EFFECT OF
SLU IOTS UPON THE AC
 - TN 0014 -- (LOOPAROUND ENABLED) CHECKS THAT ALL ZEROS CAN BE
TRANSMITTED AND READ BACK
 - TN 0015 -- (LOOPAROUND ENABLED) CHECKS THAT ALL ONES CAN
BE TRANSMITTED AND READ BACK
 - TN 0016 -- (LOOPAROUND ENABLED) CHECKS THAT A COMPLEMENTING
PATTERN (252-125) CAN BE TRANSMITTED AND READ BACK
 - TN 0017 -- (LOOPAROUND ENABLED) CHECKS THAT "ONE THRU A FIELD
OF ZEROS" AND "ZERO THRU A FIELD OF ONES" PATTERNS
CAN BE TRANSMITTED AND READ BACK.
 - TN 0020 -- CHECKS SLU TIMING FROM 50 BAUD TO 19200 BAUD
 - TN 0021 -- (PERIPHERAL SIMULATOR REQUIRED) VERIFIES OPERATION
OF EIA DRIVERS ON SIGNAL DISTRIBUTION BOARD
 - TN 0022 -- VERIFY PROGRAMMABLE MODES OF SLU #2

7.1.5 PARALLEL INTERFACE TESTS

(DEVICE CODE 66)

- TN 0001 -- CHECK PCLF TO CLEAR PRINT FLAG & PSSF TO SET PRINT FLAG
- TN 0002 -- CHECK THAT PCIE WILL SET & CLEAR INTERRUPT EN.
- TN 0003 -- TEST THAT CAF ENABLES PRINTER INT. & CLEARS FLAGS
- TN 0004 -- TESTS THAT PSTB LOADS & PRDB READS PRINTER INTERFACE BUFFER, CHECK THAT PSTB LEAVES FLAG ALONE, VERIFIES JAM INTO AC.
- TN 0005 -- CHECK THAT PCLP LOADS PRINTER INTERFACE BUFFER AND CLEARS FLAG
- TN 0006 -- CHECK THAT PSSF, PSKF, PCLF, PSTB, PCIE, & PCLP LEAVE AC UNDISTURBED, ALSO CHECKS THAT PSSF, PSKF, PCLF, PCIE, AND PRDB LEAVE INTERFACE BUFFER UNCHANGED
- TN 0007 -- CHECK PARALLEL I/O INTERFACE BUFFER

(DEVICE CODE 50)

- TN 0001 -- TEST THAT LQRE & LQRS CLEAR DONE FLAG AND THAT LQLS SETS DONE FLAG.
- TN 0002 -- TEST THAT LQLS WRITES STATUS AND LQRS READS STATUS ALSO VERIFIES INTERRUPT ENABLE FUNCTION.
- TN 0003 -- TEST THAT CAF DISABLES LQP INT. & CLEARS LIFT RIBBON
- TN 0004 -- TESTS THAT LQMP, LQMC, LQPC & LQRE LOAD INTERFACE BUFFER AND THAT LQRB READS INTERFACE BUFFER
- TN 0005 -- TESTS THAT LQMP, LQMC, LQLS, AND LQPC CLEAR AC
- TN 0006 -- TESTS THAT LQSK AND LQRE LEAVE AC UNDISTURBED, ALSO CHECKS THAT LQSK, LQRS, LQRB, & LQLS LEAVE INTERFACE BUFFER UNDISTURBED.
- TN 0007 -- TESTS PARALLEL INTERFACE BUFFER

(BOTH DEVICE CODE 50 & 66 --PERIPHERAL SIMULATOR REQUIRED)

- TN 0010 -- VERIFY IN/OUT AND DATA INTERFACE CONNECTOR
- TN 0011 -- TEST PAPER STROBE AND PAPER READY LOGIC
- TN 0012 -- TEST CARRIAGE STROBE AND CARRIAGE READY LOGIC
- TN 0013 -- TEST CHARACTER STROBE AND CHARACTER READY LOGIC
- TN 0014 -- TEST PRINTER READY LOGIC
- TN 0015 -- TEST CHECK LOGIC

7.1.6

FLOPPY INTERFACE TESTS

TN 0001 -- CAF PART I / FLAG DETECTION PART I
TN 0002 -- FLAG DETECTION PART II / "C" LINES VERIFICATION PART I
TN 0003 -- DIRECTION OF IOT XDR PART I / IOT DECODING PART I /
"C" LINES VERIFICATION PART II
TN 0004 -- FLAG DETECTION PART III / "C" LINES VERIFICATION PART III
TN 0005 -- TRANSFER REGISTER DIRECTION TEST PART II / "C" LINES
VERIFICATION PART IV
TN 0006 -- IOT DECODING VERIFICATION PART II
TN 0007 -- INTERRUPT TEST PART I / IOT DECODING VERIFICATION
PART III
TN 0010 -- INTERRUPT TEST PART II
TN 0011 -- INTERRUPT TEST PART III
TN 0012 -- INTERRUPT TEST PART IV
TN 0013 -- INIT TEST / INTERRUPT TEST PART V
TN 0014 -- (PERIPHERAL SIMULATOR REQUIRED) VERIFICATION OF
INTERFACE REGISTER AND SHIFT CONTROL OPERATION

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

/VT78 CPU DIAGNOSTIC

/VT78 CPU DIAGNOSTIC - PART 1 - FIELD 0

/PROGRAMMER: B. S. POLAND

/*****
/*****

6143 APTR#3=6143 /LOCATION OF HCW3 IN APT LOADER/MONITOR
7402 HLT=7402 /HALT
7002 BSW=7002 /BYTE SWAP
6203 CDI=6203
7421 MQL=7421 /AC TO MQ, 0 TO AC
7501 MGA=7501 /MQ + AC TO AC
7621 CAM=7621 /CLEAR AC AND MQ
7521 SWP=7521 /SWAP AC AND MQ
7701 ACL=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
6305 KIE1=6305 /LOAD CONTENT OF AC10 INTO STATUS EN FF,
/LOAD CONTENT OF AC11 INTO INT EN FF, SLU#2
6325 KIE2=6325 /LOAD CONTENT OF AC11 INTO INT EN FF, SLU#3
6234 RIB=6234 /READ(INCLUSIVE OR) THE ISF & DSP INTO BITS
/7-8 #10-11 OF THE AC RESPECTIVELY,
6040 SPF=6040 /SET TELEPRINTER FLAG
6007 CAF=6007 /CLEAR ALL FLAGS, AND CLEAR AC AND LINK
6301 KSF1=6301 /SLU #2 SKIP ON RECEIVE FLAG
6321 KSF2=6321 /SLU #3 SKIP ON RECEIVE FLAG
6311 TSF1=6311 /SLU #2 SKIP ON XMIT FLAG SET & EN
6331 TSF2=6331 /SLU #3 SKIP ON XMIT FLAG SET & EN
6661 PSKP=6661 /LA180 SKIP ON FLAG (CHAR READY)
6500 LQSK=6500 /LQP SKIP ON DONE FLAG
6137 CLSK=6137 /SKIP ON REAL TIME CLOCK FLAG
6136 CLCL=6136 /CLEAR REAL TIME CLOCK FLAG
6135 CLLE=6135 /LOAD CLOCK INT, ENABLE FROM AC11
/ AC11=0 SET INT, EN
/ AC11=0 CLR INT, EN

/*****

FIELD 0
*0
0000 0000 0000 /FIRST EDITION.

56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110

0001 6244 RMF
0002 5403 JMP I 3
0003 6200 SKPCHN
*4
0004 0000 ODT1, 0
0005 0000 ODT2, 0
0006 0000 ODT3, 0
0007 0000 0
0010 0000 0
0011 0000 0
0012 0000 0
0013 0000 0
0014 0000 0
0015 0000 0
0016 0000 0

/*****
/PDP-8 STANDARDIZED SWITCHES AND HARDWARE DESIGNATOR WORDS

0020 0000 PSR, 0000 /PSEUDO SWITCH REGISTER

/ BIT 0=1 INHIBIT ERROR HALT
/ BIT 1=1 LOOP ON ERROR
/ BIT 2=1 LOOP ON TEST

/ BIT 3=1 HALT ON COMPLETION OF A PROGRAM PASS
/ BIT 4=1 INHIBIT ERROR TYPEOUT
/ BIT 5=1 DON'T RUN CPU TEST

/ BIT 6=1 DON'T RUN EXT. MEMORY TEST
/ BIT 7=1 DON'T RUN SLU TEST
/ BIT 8=1 DON'T RUN RX01 INTERFACE TEST

/ BIT 9=1 DON'T RUN PARALLEL INTERFACE TEST
/ BIT 10=1 DON'T RUN REAL TIME CLOCK TEST
/ BIT 11=1 RUN BAUD RATE SWITCH TEST

/ HARDWARE WORD 1
/ BITS 7-11 MEMORY SIZE (16K)
/ HARDWARE WORD 2
/ BIT 0=1 APT CONTROL
/ HARDWARE WORD 3 (RETRIEVED FROM APT LOADER/MONITOR)

/ BIT 0=1 FLOPPY DRIVE 0 NOT READY
/ BIT 1=1 NO RX01 CABLED TO SYSTEM
/ BIT 2=1 NO EXTERNAL DEVICE ATTACHED TO PARALLEL INTERFACE

/ BIT 3=1 PARALLEL INTERFACE DEVICE ATTACHED
/ AND USING DEVICE CODE 50.
/ (LA180=DEVICE CODE 66,LQP=DEVICE CODE 50)

111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165

0040
0040 0000
0041 0001
0042 0010
0043 0077
0044 0100
0045 0200
0046 2525
0047 4000
0050 5252
0051 5253
0052 7700
0053 7777

*40
/CONSTANTS
KFLG, 0
K1, 0001
K10, 0010
K77, 0077
K100, 0100
K200, 0200
K2525, 2525
K4000, 4000
K5252, 5252
K5253, 5253
K7700, 7700
K7777, 7777

/SCRATCH LOCATIONS

0054 0000
0055 0000
0056 0000
0057 0000
0060 0000
0061 0000
0102
0097
0072
0062 7777
0063 0000
0064 0000
0065 0000
0066 7402
0067 0000
0070 0000
0071 0000
0072 0000
0073 0000
0074 0000
0075 0000
0076 0077
0077 2526
0100 0101
0101 0000
0102 0000
0103 1434
0104 1450
0105 1515

ACWAS, 0
MOWAS, 0
LKNAS, 0
SKPPED, 0
SOMSKP, 0
SAVFLD, 0
BIT6=POINTD
BIT7=0007
BIT11=TESLOC
BIT3, 7777
BIT4, 0
BIT5, 0
BIT8, 0
KSTOP, HLT
ACDATA, 0
MGDATA, 0
LKDATA, 0
TESLOC, 0
DAT, 0
NDF, 0
TSTNO, 0
POINTR, .+1
2526
POINTB, POINTC
POINTC, 0
POINTD, 0
POINTE, JMSLOC-1
POINTF, JMSLOC-2
INSTTR, INSTRT

/STORAGE FOR IF AND DF

166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216

0106 4537
0107 0000
0110 4533
0111 5512
0112 4447
0113 0000
0114 4635
0115 2023
4516
0116 7111
4517
0117 2237
4524
0120 5217
4521
0121 7200
4522
0122 7421
4523
0123 7464
4524
0124 7360
4525
0125 7225
4526
0126 7477
4527
0127 7321
4530
0130 7153
4531
0131 7217
4532
0132 7400
4533
0133 5000
4534
0134 6716
4535
0135 2217
4536
0136 2227
4537
0137 7012
4540
0140 7540
4541
0141 5427
0177
0177 7770

XTFLG, TFLG
STKS, 0
XSTKS, NSTKS
JMPTR, JMP I XRET
XRET, RET
RTCIF, 0
RTRN
TEST, TESTS
CHKRSF=JMS I
XCHKRSF
KBCHK=JMS I
CHKKB
FIXIL=JMS I
FIXLKG
C8CRLF=JMS I
XC8CRLF
C8PRNT=JMS I
MESAGX
C8PRT4=JMS I
XPRNT4
C8SWIT=JMS I
XC8SW
LISN=JMS I
XLISN
PRNT1=JMS I
XPRNT1
TYPE=JMS I
XTYPE
PRNT2=JMS I
XPRNT2
SPACE2=JMS I
SPACK2
C8APT=JMS I
XC8APT
C8ERR= JMS I
XC8ERR
C8START=JMS I
XC8START
INTST=JMS I
TSTIN
ENDTST=JMS I
TSTEND
C8RC=JMS I
XC8RC
C8H3=JMS I
XC8H3
CLS8WT=JMS I
WTCLSK
*177
DATPAT, 7770


```

217
218          0200 *200
219          /*****
220          /CPU TEST
221          /*****
222
223
224
225
226          /*****
227          /CPU TEST 1 - FIRST OPERATE MICROINSTRUCTION TEST
228          /*****
229          /TEST ALL BASIC SKIPS TO EITHER SKIP OR NOT SKIP WHEN AC=7777.
230          START, CBSTART          /CONSOLE CALL TO ASK SWITCH REG QUESTION.
231          CBREST, CLA CLL
232          DCA ISTNO          /CLEAR TEST NUMBER USED FOR ERROR DISPLAY
233          IOF
234          FIXIL          /FIX UP INTERRUPT LINKAGE
235          TAD PSR
236          AND (100          /EXECUTE CPU TEST????
237          SZA CLA
238          JMP BEGEXM          /NO
239          PCPUH, JMS PCPUMS          /YES - PRINT "CPU TESTING" MESSAGE
240
241          CPUT1, INTST          /IF NOT UNDER APT CONTROL.
242          CAF
243          NOP
244          NOP
245          NOP
246          SNA
247          SZL
248          CBERR          /SNA SKIPS WHEN AC CLEAR, OR SZL DOES NOT SKIP WHEN LINK = 0
249          ION          /ENABLE INTERRUPTS FOR CONSOLE PACKAGE
250          CMA          /AC TO 7777
251          SNA
252          CBERR          /CMA SKIPPED OR DID NOT COMPLEMENT, OR SNA DID NOT SKIP, OR ION SKIPPED
253          SZA
254          SKP
255          CBERR          /SNA CLEARED AC, OR SZA SKIPPED, OR SKP FAILED
256          SNA
257          CBERR          /SZA OR SKP CLEARED AC0, OR SNA FAILED TO SKIP
258          SPA
259          SKP
260          CBERR          /SNA CLEARED AC0, OR SPA SKIPPED WHEN AC0=1
261          /*****
262          /TEST CLA CLL TO CLEAR AC AND LINK
263          TSCACL, CLA CLL
264          SNA
265          SZL
266          CBERR          /CLA CLL DID NOT CLEAR AC OR LINK, OR SNA SKIPPED
267
268          CMA          /WHEN AC=0000, OR SZL DID NOT SKIP WHEN LINK=0
269          /AC TO 7777
270          /TEST BASIC SKIPS TO SKIP OR NOT SKIP WHEN AC=0000
271          TSSSSK, CMA          /AC TO 0000

```

```

272          0244 7440          SZA
273          0245 4533          CBERR          /CMA DID NOT COMPLEMENT AC OR SZA FAILED TO SKIP, OR CMA SKIPPED
274          0246 7510          SPA
275          0247 4533          CBERR          /SPA FAILED TO SKIP WHEN AC0=0
276          0250 7500          SNA
277          0251 7410          SKP
278          0252 4533          CBERR          /SNA SKIPPED WHEN AC0=0
279
280          /*****
281          /TEST IAC TO INCREMENT AC TO 0000 AND SET LINK
282          TSTIAC, CLL          /THIS INSTRUCTION NOT YET TESTED
283          CMA          /AC TO 7777
284          IAC          /AC TO 0000, LINK TO 1
285          SZA
286          CBERR          /SZA FAILED TO SKIP, OR IAC DID NOT INCREMENT AC TO 0000
287
288          /TEST TO SEE IF LINK COMPLEMENTED TO A 1 ON A CARRY OUT OF ADDER
289          TSTLCH, SNL
290          CBERR          /LINK DID NOT COMPLEMENT ON CARRY OUT, OR CLL FAILED
291
292          /OR SNL FAILED TO SKIP FOR LINK = 1
293          SZL
294          SKP
295          CBERR          /SZL SKIPPED ON LINK=1, OR SNL CLEARED LINK, OR SKP FAILS WHEN LINK =1
296          /TEST CLL TO CLEAR LINK
297          TSTCLL, CLL
298          SNL
299          SKP          /CLL FAILED, OR SNL SKIPPED WHEN LINK=0, OR SKP FAILED WHEN LINK=0
300          CBERR
301          SZL
302          CBERR          /SZL FAILED TO SKIP, OR SNL OR SKP SET LINK
303
304          /TEST ABILITY OF CML TO SET LINK
305          TSTCML, CML          /LINK TO 1
306          SNL
307          CBERR          /CML DID NOT SET LINK
308
309          /TEST ABILITY OF LINK TO COMPLEMENT FROM A 1 TO A 0 ON A CARRY OUT
310          CMA          /AC TO 7777, LINK=1
311          IAC          /AC TO 0000, CARRY TO LINK, LINK TO 0
312          SZL
313          CBERR          /CARRY OUT DID NOT COMPLEMENT LINK TO A 0
314
315          /TEST ABILITY OF CML TO COMPLEMENT LINK FROM A 0 TO A 1 AND BACK TO A 0
316          CLA
317          CML          /LINK TO 1
318          CML          /LINK TO 0
319          SZL
320          CBERR          /CML DID NOT COMPLEMENT LINK FROM A 1 TO A 0
321          SZA
322          CBERR          /CML CHANGED AC
323
324          /TEST CLA TO CLEAR AC AND NOT CLEAR LINK
325          CML          /MAKE LINK A 1
326          CMA          /AC TO 7777
327          CLA          /AC TO 0000

```

```

327 0314 7420 SNL
328 0315 4533 CBERR /CLA CLEARED LINK
329 0316 7440 SZA
330 0317 4533 CBERR /CLA DID NOT CLEAR AC
331 /*****
332 /TEST NOP TO NOT CHANGE AC OR LINK
333 0320 7000 NOP /AC=0000,LINK=1
334 0321 7420 SNL
335 0322 4533 CBERR /NOP SKIPPED OR CLEARED LINK
336 0323 7440 SZA
337 0324 4533 CBERR /NOP SET AC BIT
338 0325 7040 CMA /AC TO 7777
339 0326 7020 CML /LINK TO 0
340 0327 7000 NOP /AC=7777,LINK=0
341 0330 7430 SZL
342 0331 4533 CBERR /NOP SKIPPED OR SET LINK
343 0332 7450 SNA
344 0333 4533 CBERR /NOP CLEARED AC
345 /*****
346 /TEST RAL TO NOT PICK UP BITS BY SHIFTING ZEROES
347 0334 7200 CLA /AC TO 0000
348 0335 7004 RAL
349 0336 7450 SNA
350 0337 7430 SZL
351 0340 4533 CBERR /RAL OF ZEROES PICKED UP AC BIT OR LINK BIT
352 /*****
353 /TEST RTL TO NOT PICK UP BITS WHEN SHIFTING ALL ZEROES
354 0341 7006 RTL
355 0342 7450 SNA
356 0343 7430 SZL
357 0344 4533 CBERR /RTL PICKED UP BIT IN AC OR LINK WHEN SHIFTING ZEROES
358 /*****
359 /TEST RAR TO NOT PICK UP BITS WHEN SHIFTING ALL ZEROES
360 0345 7010 RAR
361 0346 7450 SNA
362 0347 7430 SZL
363 0350 4533 CBERR /RAR PICKED UP BIT IN AC OR LINK WHEN SHIFTING ZEROES
364 /*****
365 /TEST RTR TO NOT PICK UP BITS WHEN SHIFTING ALL ZEROES
366 0351 7012 RTR
367 0352 7450 SNA
368 0353 7430 SZL
369 0354 4533 CBERR /RTR PICKED UP BIT IN AC OR LINK WHEN SHIFTING ZEROES
370 /*****
371 /TEST BSW TO NOT PICK UP ANY BITS WHEN SWAPPING ZEROES,AND TO NOT AFFECT LINK
372 0355 7002 BSW
373 0356 7450 SNA
374 0357 7430 SZL
375 0360 4533 CBERR /BSW PICKED UP BIT IN AC WHEN SWAPPING ALL ZEROES,OR SET LINK
376 0361 4536 ENDTST
377 0362 5774 JMP CPUT2
378 0374 0400
379 0375 2200
380 0376 3000
381 0377 0100

```

```

382 0400 PAGE
383 /*****
384 /CPU TEST 2 - FIRST TEST OF MRI
385 /*****
386 0400 4535 /TEST TAD TO ADD 7777 TO A CLEAR AC
387 0401 1053 CPUT2, INTST
388 0402 7450 TAD K7777 /AC TO 7777, LINK=0
389 0403 4533 SNA
390 0404 7430 CBERR /TAD DID NOT LOAD AC, OR TAD SKIPPED,
391 0405 4533 SZL
392 CBERR /TAD SET LINK WHEN NO CARRY OUT EXPECTED
393 /*****
394 0406 1041 /TEST TAD TO ADD 1 TO AC=7777 TO PRODUCE AC=0000 AND LINK=1
395 0407 7440 TAD K1 /AC TO 0000, LINK TO 1
396 0410 4533 SZA
397 CBERR /TAD USED INCORRECT VALUE,OR ADDER CARRY CKT
398 0411 7420 /FAULTY OR TAD SKIPPED
399 0412 4533 SNL
400 CBERR /CARRY OUT OF ADDER DID NOT COMPLEMENT LINK
401 /*****
402 /TEST ADDER CARRY STRUCTURE TO GENERATE CARRY THROUGH ALL POSITIONS
403 0413 1041 /UNDER OPPOSITE CONDITIONS FROM PREVIOUS TEST
404 0414 1053 TAD K1 /AC TO 0001, LINK=1
405 0415 7450 TAD K7777 /AC TO 0000, LINK TO 0
406 0416 7430 SNA
407 0417 4533 SZL
408 CBERR /CARRY FAILED TO PROPAGATE ALL THROUGH ADDER TO LINK
409 /*****
410 0420 1377 /TEST ABILITY TO ADD 0000 TO A CLEAR AC TO PRODUCE A CLEAR AC
411 0421 7450 TAD 0 /AC=0000, LINK=0
412 0422 7430 SNA
413 0423 4533 SZL
414 CBERR /ADDING 0000 TO 0000 PRODUCED NON-ZERO RESULT
415 /OR COMPLEMENTED LINK
416 /*****
417 0424 1053 /TEST ADDER'S ABILITY TO PROPAGATE CARRY WHEN ACN=1, MDN=1, AND CARRY IN N=1
418 0425 1053 TAD K7777 /AC TO 7777, LINK=0
419 0426 7420 TAD K7777 /AC TO 7776, LINK TO 1
420 0427 4533 SNL
421 CBERR /CARRY DID NOT PROPAGATE TO LINK
422 0430 7001 IAC /MAKE AC=7777 FOR EASE OF CHECKING RESULT OF PREVIOUS TAD
423 0431 7040 CMA /AC TO 0000
424 0432 7440 SZA
425 0433 4533 CBERR /CARRY DID NOT PROPAGATE CORRECTLY
426 /OR TAD USED INCORRECT OPERAND
427 /*****
428 0434 1046 /TEST ADDER'S ABILITY TO GENERATE CARRY WHEN ACN=1, MDN=1, AND CARRY IN N=0 IN ODD NUMBERED BIT POSITIONS
429 0435 1046 TAD K2525 /AC TO 2525, LINK = 1
430 0436 1046 TAD K2525 /AC TO 5252
431 0437 7040 TAD K2525 /AC TO 7777, LINK = 1 (NO CARRY GENERATED TO LINK)
432 0440 7440 CMA /AC TO 0000, LINK = 1
433 0441 4533 SZA
434 0442 7420 CBERR /CARRY FAILED IN AN ODD BIT POSITION
435 0443 4533 SNL
CBERR /LINK COMPLEMENTED WHEN NO CARRY OUT EXPECTED

```

```

436 /*****
437 /TEST ADDER'S ABILITY TO GENERATE CARRY WHEN AC=1,MDN=1, AND CARRY IN=0 IN ALL EVEN BIT POSITIONS
438 TAD K5252 /AC TO 5252,LINK=1
439 K5252 /AC TO 2524,LINK=0
440 TAD K5252 /AC TO 7776,LINK=0
441 IAC /AC TO 7777,LINK=0
442 CMA /AC TO 0000,LINK=0
443 STA
444 CBERR /CARRY FAILED IN AN EVEN BIT POSITION
445 SZL
446 CBERR /CARRY OUT FAILED TO COMPLEMENT LINK
447 /*****
448 /TEST AND INSTRUCTION TO NOT SET ANY AC BITS WHEN AC=0000
449 AND K7777 /AC=0000,LINK=0
450 SNA
451 SZL
452 CBERR /AND SET BIT WHEN AC INITIALLY CLEAR,OR AND SET LINK
453 /*****
454 /TEST AND INSTRUCTION TO CLEAR ALL AC BITS WHEN USING AN OPERAND OF 0000
455 TAD K7777 /AC TO 7777,LINK=0
456 AND 0 /AC TO 0000,LINK=0
457 SNA
458 SZL
459 CBERR /AND FAILED TO CLEAR ALL AC BITS, OR SET LINK
460 /*****
461 /TEST AND INSTRUCTION TO NOT CLEAR ANY AC BITS WHEN AC=7777 AND MD=7777
462 TAD K7777 /AC TO 7777
463 AND K7777 /AC=7777,LINK=0
464 CMA /AC TO 0000,LINK=0
465 SNA
466 SZL
467 CBERR /AND OF 7777 CLEARED AC BIT OR SET LINK
468 /*****
469 /TEST FOR ADJACENT PIN SHORTS IN "AND" CIRCUITRY
470 TAD K2525 /AC TO 2525
471 AND K5252 /AC TO 0000
472 SEA
473 CBERR /*AND" DID NOT CLEAR AC,POSSIBLE ADJACENT PIN SHORTS IN AND CIRCUITRY
474 ENDIS
475 /*****
476 /CPU TEST 3 - TEST ADDER FUNCTION
477 /*****
478 /TEST BIT 11 INPUT TO ADDER=0
479 CPUT3, INTST
480 TAD K1 /AC TO 0001
481 SNA
482 CBERR /ADDER=0 OPEN ON BIT 11 INPUT
483 /*****
484 /TEST BIT 10 INPUT TO ADDER=0
485 CLA /AC TO 0000
486 TAD 0 /AC TO 0002
487 SNA
488 CBERR /ADDER=0 OPEN ON BIT 10 INPUT
489 /*****
490 /TEST BIT 9 INPUT TO ADDER=0

```

```

491 CLA /AC TO 0000
492 TAD 0 /AC TO 0004
493 SNA
494 CBERR /ADDER=0 OPEN ON BIT 9 INPUT
495 /*****
496 /TEST BIT 8 INPUT TO ADDER=0
497 CLA /AC TO 0000
498 TAD K10
499 SNA
500 CBERR /ADDER=0 OPEN ON BIT 8 INPUT
501 /*****
502 /TEST BIT 7 INPUT TO ADDER=0
503 CLA /AC TO 0000
504 TAD 0 /AC TO 0020
505 SNA
506 CBERR /ADDER=0 OPEN ON BIT 7 INPUT
507 /*****
508 /TEST BIT 6 INPUT TO ADDER=0
509 CLA /AC TO 0000
510 TAD 0 /AC TO 0000
511 SNA
512 CBERR /ADDER=0 OPEN ON BIT 6 INPUT
513 /*****
514 /TEST BIT 5 INPUT TO ADDER=0
515 CLA /AC TO 0000
516 TAD K100
517 SNA
518 CBERR /ADDER=0 OPEN ON BIT 5 INPUT
519 /*****
520 /TEST BIT 4 INPUT TO ADDER=0
521 CLA /AC TO 0000
522 TAD K200 /AC TO 0200
523 SNA
524 CBERR /ADDER=0 OPEN ON BIT 4 INPUT
525 /*****
526 /TEST BIT 3 INPUT TO ADDER=0
527 CLA /AC TO 0000
528 TAD 0 /AC TO 0400
529 SNA
530 CBERR /ADDER=0 OPEN ON BIT 3 INPUT
531 /*****
532 /TEST BIT 2 INPUT TO ADDER=0
533 CLA /AC TO 0000
534 TAD 0 /AC TO 1000
535 SNA
536 CBERR /ADDER=0 OPEN ON BIT 2 INPUT
537 /*****
538 /TEST BIT 1 INPUT TO ADDER=0
539 CLA /AC TO 0000
540 TAD 0 /AC TO 0000
541 SNA
542 CBERR /ADDER=0 OPEN ON BIT 1 INPUT
543 /*****
544 /TEST BIT 0 INPUT TO ADDER=0
545 CLA /AC TO 0000

```

```

546 0556 1047 TAD K4000 /AC TO 4000
547 0557 7450 SNA
548 0560 4533 CBERR /ADDER=0 OPEN ON BIT 0 INPUT
549 0561 4536 ENDTST
550 0562 5767 JMP CPUT4
551 0567 0600
552 0570 2000
553 0571 1300
554 0572 0400
555 0573 0740
556 0574 0020
557 0575 0004
558 0576 0002
559 0577 0000

```

PAGE

```

560 /*****
561 /CPU TEST 4 - SECOND OPERATE MICROINSTRUCTION TEST
562 /*****
563 /TEST RAR TO ROTATE AND NOT DROP ANY BITS
564 CPUT4, INTST
565 TAD K2525 /AC TO 2525
566 CLL /LINK TO 1
567 CML /LINK TO 1
568 RAR /AC TO 5252, LINK=1
569 SNL
570 CBERR /RAR DID NOT SHIFT AC11 TO LINK
571 CMA /AC TO 2525, LINK=1
572 AND K5252 /AC=0000, LINK=1
573 SZA
574 CBERR /RAR DROPPED BIT OR DID NOT SHIFT
575 /ANY BIT SET IN AC INDICATES POSITION OF DROPPED BIT
576 /*****
577 /TEST RAR TO NOT PICK UP ANY BITS
578 TAD K2525 /AC TO 2525, LINK=1
579 RAR /AC TO 5252, LINK=1
580 AND K2525 /AC TO 0000, LINK=1
581 SNA
582 SNL
583 CBERR /RAR PICKED UP BIT, POSITION OF BIT PICK UP IS
584 /INDICATED BY BIT(S) SET IN AC
585 /*****
586 /TEST RAL TO SHIFT AND NOT DROP ANY BITS
587 TAD K5252 /AC TO 5252 LINK=1
588 RAL /AC TO 2525 LINK=1
589 SNL
590 CBERR /RAL DROPPED LINK BIT
591 CMA /AC TO 5252 LINK=1
592 AND K2525 /AC TO 0000 LINK=1
593 SZA
594 CBERR /RAL DROPPED BIT OR DID NOT SHIFT
595 /FAILING BIT POSITIONS ARE SET IN AC
596 /*****
597 /TEST RAL TO NOT PICK UP ANY BITS
598 TAD K5252 /AC TO 5252 LINK=1
599 RAL /AC TO 2525 LINK=1

```

```

600 0633 0050 AND K5252 /AC TO 0000 LINK=1
601 0634 7440 SZA
602 0635 4533 CBERR /RAL PICKED UP BIT, BITS SET IN AC INDICATE FAILING POSITIONS
603 /*****
604 /TEST RTR TO SHIFT TWICE AND NOT DROP ANY BITS
605 CLL /CLEAR LINK
606 TAD (4444 /AC TO 4444 LINK=0
607 RTR /AC TO 1111 LINK=0
608 SZL
609 CBERR /RTR PICKED UP LINK BIT
610 CMA /AC TO 6666
611 AND (1111 /AC TO 0000 LINK=0
612 SZA
613 CBERR /RTR DROPPED BIT OR DID NOT SHIFT TWICE
614 /BIT SET IN AC INDICATES BIT DROPPED
615 /*****
616 /TEST RTR TO NOT PICK UP ANY BITS
617 TAD (4444 /AC TO 4444
618 RTR /AC TO 1111
619 AND (6666 /AC TO 0000
620 SZA
621 CBERR /RTR PICKED UP BIT, BIT SET IN AC INDICATES FAILING POSITION,
622 /*****
623 /TEST RTL TO SHIFT AND NOT DROP BITS
624 TAD (1111
625 RTL /AC TO 4444 LINK=0
626 SZL
627 CBERR /RTL PICKED UP LINK BIT
628 CMA /AC TO 3333 LINK=0
629 AND (4444 /AC TO 0000 LINK=0
630 SZA
631 CBERR /RTL DROPPED BIT OR DID NOT SHIFT
632 /BIT SET IN AC INDICATES BIT DROPPED
633 /*****
634 /TEST RTL TO NOT PICK UP ANY BITS
635 TAD (1111 /AC TO 1111
636 RTL /AC TO 4444
637 AND (3333 /AC TO 0000
638 SZA
639 CBERR /RTL PICKED UP BIT, BIT SET IN AC INDICATES FAILING POSITION
640 /*****
641 /TEST BSW TO SWAP AND NOT DROP BITS OR CHANGE LINK
642 CLL /INITIALIZE LINK TO ZERO
643 TAD K77 /AC TO 0077 LINK=0
644 BSW /AC TO 7700 LINK=0
645 SZL
646 CBERR /BSW SET LINK
647 TAD K100 /AC TO 0000 LINK TO 1
648 SZA
649 CBERR /BSW DID NOT SWAP OR PICKED UP BIT,
650 /*****
651 /TEST BSW FOR ADJACENT PIN SHORTS AND DROPPED BITS
652 TAD K7700 /AC TO 7700 LINK=1
653 BSW /AC TO 0077 LINK=1
654 SNL

```

655 0704 4533
 656 0705 7040
 657 0706 1044
 658 0707 7440
 659 0710 4533
 660 0711 4536
 661
 662
 663
 664
 665 0712 4535
 666 0713 1053
 667 0714 7421
 668 0715 7440
 669 0716 4533
 670
 671
 672 0717 7521
 673 0720 7040
 674 0721 7440
 675 0722 4533
 676
 677 0723 7040
 678
 679
 680 0724 7521
 681 0725 7440
 682 0726 4533
 683
 684
 685 0727 1046
 686 0730 7421
 687 0731 7521
 688 0732 7040
 689 0733 0046
 690 0734 7440
 691 0735 4533
 692
 693
 694
 695 0736 1046
 696 0737 7421
 697 0740 7521
 698 0741 0050
 699 0742 7440
 700 0743 4533
 701
 702 0744 7100
 703 0745 7040
 704
 705
 706 0746 7421
 707 0747 7501
 708 0750 7430
 709 0751 4533

CBERR /BSW CLEARED LINK
 CMA /AC TO 7700
 TAD K100 /AC TO 0000, LINK TO 0
 SEA
 CBERR /BSW PICKED UP OR DROPPED BIT(S).
 ENDTST
 /*****
 /CPU TEST 5 - MQ MICROINSTRUCTION TEST
 /*****
 /TEST MQL TO CLEAR AC
 CPUT5, INTST
 TAD K7777 /AC TO 7777
 MQL /AC TO MQ. 0 TO AC.
 SEA
 CBERR /MQL DID NOT CLEAR AC
 /*****
 /TEST SWP TO EXCHANGE MQ AND AC, TEST RESULTS OF PREVIOUS MQL.
 SWP /AC TO MQ. MQ TO AC. AC=7777 MQ=0000
 CMA /AC TO 0000
 SEA
 CBERR /SWP DID NOT LOAD AC WITH MQ, OR MQL DID NOT LOAD MQ, OR
 /SWP OR MQL DROPPED A BIT.
 CMA /AC TO 7777 MQ=0000
 /*****
 /TEST MQ REGISTER FOR OUTPUTS STUCK HIGH
 SWP /AC TO 0000 MQ TO 7777
 SEA
 CBERR /SWP PICKED UP BIT IN AC.
 /*****
 /TEST MQ FOR ADJACENT PIN SHORTS BY TESTING FOR DROPPED BITS
 TAD K2525 /AC TO 2525 MQ=7777
 MQL /AC TO 0000 MQ TO 2525
 SWP /AC TO 2525 MQ TO 0000
 CMA /AC TO 5252
 AND K2525 /AC TO 0000
 SEA
 CBERR /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
 TAD K2525 /AC TO 2525 MQ=0000
 MQL /AC TO 0000 MQ TO 2525
 SWP /AC TO 2525 MQ TO 0000
 AND K5252 /AC TO 0000
 SEA
 CBERR /MQL OR SWP PICKED UP BIT, POSSIBLE ADJACENT PIN SHORT IN MQ,
 /BIT SET IN AC INDICATES POSITION OF FAILURE.
 CLL
 CMA /AC TO 7777
 /*****
 /TEST MQA TO OR THE MQ WITH THE AC. ENTER WITH AC=7777 MQ=0000 LINK=0
 MQL /AC TO 0000 MQ TO 7777
 MQA /AC TO 7777 MQ=7777
 SEL
 CBERR /MQA SET LINK

710 0752 7040
 711 0753 7440
 712 0754 4533
 713 0755 7521
 714 0756 7450
 715 0757 4533
 716 0760 5773
 717 0773 1000
 718 0774 3333
 719 0775 6666
 720 0776 1111
 721 0777 4444
 722
 723
 724 1000 7521
 725 1001 7040
 726 1002 7020
 727 1003 7521
 728 1004 7440
 729 1005 4533
 730 1006 7420
 731 1007 4533
 732 1010 7521
 733 1011 7440
 734 1012 4533
 735
 736
 737 1013 7040
 738 1014 7701
 739 1015 7440
 740 1016 4533
 741 1017 7420
 742 1020 4533
 743 1021 7040
 744 1022 7521
 745 1023 7440
 746 1024 4533
 747 1025 4536
 748
 749
 750
 751
 752 1026 4535
 753 1027 7100
 754 1030 1053
 755 1031 3072
 756 1032 7440
 757 1033 4533
 758 1034 7430
 759 1035 4533
 760 1036 1072
 761 1037 7040
 762 1040 7440
 763 1041 4533

CMA /AC TO 0000 MQ=7777
 SEA
 CBERR /MQA DID NOT OR MQ WITH AC
 SWP /AC TO 7777 MQ TO 0000
 SNA
 CBERR /MQA CLEARED MQ
 JMP CANTST
 PAGE
 /*****
 /TEST CAM TO CLEAR AC AND MQ
 CANTST, SWP /AC TO 0000 MQ TO 7777
 CMA /AC TO 7777 MQ=7777
 CML /SET LINK
 CAM /CLEAR AC AND MQ
 SEA
 CBERR /CAM DID NOT CLEAR AC
 SNL
 CBERR /CAM CLEARED LINK
 SWP /AC=0000 MQ=0000
 SEA
 CBERR /CAM DID NOT CLEAR MQ
 /*****
 /TEST ACL TO LOAD ZEROS FROM MQ TO AC
 CMA /AC TO 7777 MQ=0000
 ACL /AC TO 0000 MQ=0000
 SEA
 CBERR /ACL DID NOT LOAD 0000 TO AC
 SNL
 CBERR /ACL CLEARED LINK
 CMA /AC TO 7777
 SWP /AC TO 0000 MQ TO 7777
 SEA
 CBERR /ACL CHANGED MQ
 ENDTST
 /*****
 /CPU TEST 6 - TEST DCA AND ISZ, DIRECT ADDRESSING TO PAGE ZERO
 /*****
 /TEST DCA TO STORE ALL 1'S, CLEAR AC, AND NOT AFFECT LINK
 CPUT6, INTST
 CLL /CLEAR LINK
 TAD K7777 /AC TO 7777 LINK=0
 DCA TESLOC /AC TO 0000 LINK=0
 SEA
 CBERR /DCA DID NOT CLEAR AC, OR DCA SKIPPED
 SEL
 CBERR /DCA SET LINK
 TAD TESLOC /AC TO 7777
 CMA
 SEA
 CBERR /DCA DID NOT STORE, OR DCA STORED TO WRONG ADDRESS

```

764
765
766      1042 3072
767      1043 1072
768      1044 7440
769      1045 4533
770
771
772
773      1046 2072
774      1047 7440
775      1050 4533
776      1051 7430
777      1052 4533
778      1053 1072
779      1054 1053
780      1055 7440
781      1056 4533
782
783
784      1057 1053
785      1060 3072
786      1061 2072
787      1062 4533
788      1063 7420
789      1064 4533
790      1065 1072
791      1066 7440
792      1067 4533
793
794      1070 4536
795
796
797
798
799      1071 4535
800      1072 1274
801      1073 7410
802      1074 2520
803      1075 1950
804      1076 7440
805      1077 4533
806
807
808      1100 1053
809      1101 3303
810      1102 7410
811      1103 0000
812      1104 1303
813      1105 7040
814      1106 7440
815      1107 4533
816      1110 3303
817
818

```

```

/*****
/TEST DCA TO STORE ALL ZEROS
DCA      TESLOC /0000 TO LOCATION "TESLOC"
TAD      TESLOC /0000 TO AC
SZA
CBERR    /DCA DID NOT STORE ALL 0'S, BIT SET IN AC
          /INDICATES FAILING BIT POSITION,
/*****
/TEST ISZ TO INCREMENT WITHOUT SKIPPING
ISZ      TESLOC /LOCATION "TESLOC" TO 0001
SZA
CBERR    /ISZ SKIPPED WHEN NO OVERFLOW, OR ISZ SET AC BIT
SZL
CBERR    /ISZ SET LINK
TAD      TESLOC /AC TO 0001 LINK=0
TAD      K7777 /AC TO 0000, LINK TO 1
SZA
CBERR    /ISZ DID NOT INCREMENT BY 1
/*****
/TEST ISZ TO SKIP ON OVERFLOW
TAD      K7777 /AC TO 7777 LINK=1
DCA      TESLOC /LOCATION "TESLOC" TO 7777, AC TO 0000
ISZ      TESLOC /SHOULD SKIP
CBERR    /ISZ DID NOT SKIP ON OVERFLOW
SNL
CBERR    /OVERFLOW ON ISZ SET LINK
TAD      TESLOC /0000 TO AC
SZA
CBERR    /ISZ DID NOT INCREMENT 7777 TO 0000, OR ISZ
          /AFFECTED AC ON OVERFLOW
ENDTST
/*****
/CPU TEST 7 - TEST "AND",TAD,ISZ,AND DCA DIRECT ADDRESSING TO SAME PAGE
/TEST TAD TO ADDRESS SAME PAGE DIRECT
CPUT7, INTST
TAD      .+2 /AC TO 2526 LINK=0
SKP
2526    /OPERAND FOR TAD SAME PAGE TEST
TAD      K5252 /AC TO 0000, LINK TO 1
SZA
CBERR    /TAD TO SAME PAGE DIRECT FAILED
/*****
/TEST DCA TO SAME PAGE DIRECT
TAD      K7777 /AC TO 7777
DCA      .+2 /AC TO 0000
SKP
0000    /TEST LOCATION FOR DCA TO SAME PAGE TEST
TAD      .-1 /AC TO 7777
CMA
CBERR    /DCA TO SAME PAGE FAILED
DCA      .-5 /CLEAR TEST LOCATION FOR POSSIBLE SECOND PASS
/*****
/TEST ISZ TO SAME PAGE DIRECT TO SKIP

```

```

819      1111 1053
820      1112 3314
821      1113 7410
822      1114 0000
823      1115 2314
824      1116 4533
825
826
827      1117 2314
828      1120 7410
829      1121 4533
830
831
832      1122 1050
833      1123 0325
834      1124 7410
835      1125 2525
836      1126 7440
837      1127 4533
838      1130 4536
839

```

```

TAD      K7777 /AC TO 7777
DCA      .+2 /AC TO 0000
SKP
0000    /ISZ TEST LOCATION
ISZ      .-1 /SHOULD SKIP
CBERR    /ISZ DID NOT SKIP
/*****
/TEST ISZ TO SAME PAGE DIRECT TO NOT SKIP
ISZ      .-3 /SHOULD NOT SKIP
SKP
CBERR    /ISZ SKIPPED WHEN NO SKIP EXPECTED
/*****
/TEST "AND" TO SAME PAGE DIRECT TO CLEAR ALL AC BITS
TAD      K5252 /AC TO 5252
AND      .+2 /AC TO 0000
SKP
2525    /OPERAND FOR AND SAME PAGE TEST
SZA
CBERR    /AND TO SAME PAGE DIRECT FAILED
ENDTST

```

```

840
841
842
843
844
845 1131 4535
846 1132 1476
847 1133 1050
848 1134 7440
849 1135 4533
850
851
852 1136 1053
853 1137 3500
854 1140 1101
855 1141 7040
856 1142 7440
857 1143 4533
858
859
860 1144 2500
861 1145 4533
862
863
864 1146 1046
865 1147 3101
866 1150 1053
867 1151 0500
868 1152 1051
869 1153 7440
870 1154 4533
871
872
873
874 1155 1210
875 1156 3072
876 1157 0610
877 1160 1210
878 1161 7041
879 1162 1072
880 1163 7440
881 1164 4533
882 1165 4536
883 1166 5777
884 1177 1200
885
886
887
888
889 1200 4535
890 1201 1603
891 1202 7410
892 1203 0077
893 1204 1050

```

/*****
/CPU TEST 10 - TEST TAD, "AND", DCA AND ISZ INDIRECT THRU PAGE ZERO
/*****
/TEST OF TAD THRU PAGE ZERO INDIRECT
CPUT10, INTST
TAD I POINTR /AC TO 2526
TAD K5252 /AC TO 0000
SZA
CBERR /TAD INDIRECT THRU PAGE ZERO FAILED, OR AUTO-INDEXED.
/*****
/TEST DCA INDIRECT THRU PAGE ZERO
TAD K7777 /AC TO 7777
DCA I POINTB /7777 TO LOCATION "POINTC", AC TO 0000
TAD POINTC /AC TO 7777
CMA /AC TO 0000
SZA
CBERR /DCA INDIRECT THRU PAGE ZERO FAILED
/*****
/TEST ISZ INDIRECT THRU PAGE ZERO
ISZ I POINTB /LOCATION "POINTC" TO 0000, SKIP
CBERR /ISZ INDIRECT THRU PAGE ZERO FAILED TO SKIP
/*****
/TEST "AND" INDIRECT THRU PAGE ZERO
TAD K2525 /AC TO 2525
DCA POINTC /AC TO 0000
TAD K7777 /AC TO 7777
AND I POINTB /AC TO 2525
TAD K5253 /AC TO 0000
SZA
CBERR /AND INDIRECT THRU PAGE ZERO FAILED
/*****
/TEST AUTO-INDEX DECODER FROM BITS 0 THRU 3 BY ADDRESSING ADDRESS 1010
/INDIRECTLY AND INSURING THAT AUTO-INDEXING DID NOT TAKE PLACE.
TAD 1010 /GET INITIAL CONTENTS OF 1010
DCA TESLOC /SAVE FOR LATER COMPARISON
AND I 1010 /REFERENCE 1010 INDIRECTLY
TAD 1010 /GET CONTENTS OF 1010
CIA /NEGATE IT
TAD TESLOC /COMPARE TO INITIAL CONTENTS
SZA /FINAL INITIAL?
CBERR /NO, AUTO INDEX OCCURRED FOR LOCATION 1010
ENDTST
JMP CPUT11
PAGE
/*****
/CPU TEST 11 - TEST TAD, DCA, ISZ, & "AND" INDIRECT THRU SAME PAGE
/*****
/TEST TAD INDIRECT THRU SAME PAGE
CPUT11, INTST
TAD I .+2 /AC TO 2526
SKP
POINTR+1 /POINTER FOR TAD INDIRECT THRU SAME PAGE
TAD K5252 /AC TO 0000

```

894 1205 7440
895 1206 4533
896
897
898 1207 1053
899 1210 3612
900 1211 7410
901 1212 0101
902 1213 1101
903 1214 7040
904 1215 7440
905 1216 4533
906
907
908 1217 2622
909 1220 4533
910 1221 7410
911 1222 0101
912
913
914 1223 1046
915 1224 3101
916 1225 1053
917 1226 0630
918 1227 7410
919 1230 0101
920 1231 1051
921 1232 7440
922 1233 4533
923 1234 4536
924
925
926
927
928 1235 4535
929 1236 1046
930 1237 3010
931 1240 1010
932 1241 1051
933 1242 7440
934 1243 4533
935
936
937 1244 3101
938 1245 1046
939 1246 3102
940 1247 1100
941 1250 3010
942 1251 1410
943 1252 1051
944 1253 7440
945 1254 4533
946
947
948

```

SZA
CBERR /TAD INDIRECT THRU SAME PAGE FAILED
/*****
/TEST DCA INDIRECT THRU SAME PAGE
TAD K7777 /AC TO 7777
DCA I .+2 /AC TO 0000
SKP
POINTC /POINTER FOR DCA INDIRECT THRU SAME PAGE
TAD POINTC /AC TO 7777
CMA
SZA
CBERR /DCA INDIRECT THRU SAME PAGE FAILED
/*****
/TEST ISZ INDIRECT THRU SAME PAGE
ISZ I .+3 /LOCATION "POINTC" TO 0000, SKIP
CBERR /ISZ INDIRECT THRU SAME PAGE FAILED TO SKIP
SKP
POINTC
/*****
/TEST "AND" INDIRECT THRU SAME PAGE
TAD K2525 /AC TO 2525
DCA POINTC /
TAD K7777 /AC TO 7777
AND I .+2 /AC TO 2525
SKP
POINTC /POINTER FOR "AND" INDIRECT THRU SAME PAGE
TAD K5253 /AC TO 0000
SZA
CBERR /"AND" INDIRECT THRU SAME PAGE FAILED
ENDTST
/*****
/CPU TEST 12 - TEST AUTO-INDEX
/*****
/TEST AUTO-INDEX TO NOT INCREMENT WHEN NOT INDIRECTLY ADDRESSED.
CPUT12, INTST
TAD K2525 /AC TO 2525
DCA 10 /ADDRESS 10 TO 2525
TAD 10 /AC TO 2525
TAD K5253 /AC TO 0000
SZA
CBERR /AUTO-INDEX INCREMENTED WHEN NOT INDIRECTLY ADDRESSED
/*****
/TEST AUTO-INDEX TO INCREMENT WHEN INDIRECTLY ADDRESSED
DCA POINTC /CLEAR LOCATION "POINTC"
TAD K2525 /AC TO 2525
DCA POINTD /LOCATION "POINTD" TO 2525
TAD POINTB
DCA 10 /SET LOCATION 10 TO THE ADDRESS OF "POINTC"
TAD I 10 /LOCATION "POINTD"'S CONTENTS TO AC, AC TO 2525
TAD K5253 /AC TO 0000
SZA
CBERR /AUTO-INDEX FAILED TO INCREMENT
/*****
/AUTO-INDEX DECODER OPEN ON INPUT FROM BITS 0-3
/TEST BIT 0 INPUT OF AUTO-INDEX DECODER BY ADDRESSING LOCATION 0007

```

949 /INDIRECTLY AND CHECKING THAT AUTO-INDEXXING DID NOT OCCUR
950 1255 3007 DCA 7 /CLEAR LOCATION 0007
951 1256 1407 TAD I 7 /REFERENCE LOCATION 0007 INDIRECTLY
952 1257 7700 CLA /CLEAR AC
953 1260 1007 TAD 7 /GET CONTENTS OF LOCATION 0007
954 1261 7440 SZA /DOES IT STILL CONTAIN 0000?
955 1262 4533 CBERR /NO, LOCATION 0007 AUTO-INDEXXED
956
957 /*****
958 /TEST AUTO-INDEXX DECODER INPUT FROM BITS 4 THRU 7, BY REFERENCING
959 1263 1030 /ADDRESS 0030 INDIRECTLY AND CHECKING THAT AUTO-INDEXXING DID NOT OCCUR
960 1264 3072 TAD 30 /GET INITIAL CONTENTS OF LOC 0030
961 1265 0430 DCA TESLOC /SAVE IT FOR COMPARISON
962 1266 1030 AND I 30 /REFERENCE 30 INDIRECTLY
963 1267 7040 TAD 30 /GET FINAL CONTENTS OF LOC 30
964 1270 7001 CMA /COMPLEMENT IT FOR COMPARE
965 1271 1072 IAC /COMPARE TO INITIAL CONTENTS
966 1272 7440 TAD TESLOC /WAS LOC 30 AUTO-INDEXXED?
967 1273 4533 SZA /YES, LOC 30 AUTO-INDEXXED?
968 1274 4536 CBERR /YES, LOC 30 AUTO-INDEXXED
969 ENDTST
970 /*****
971 /CPU TEST 13 - TEST INTERNAL IOT INSTRUCTIONS
972 /*****
973 /TEST GTF TO SAVE LINK
974 1275 4535 CPUT13, INTST
975 1276 7100 CLL /CLEAR LINK
976 1277 7020 CML /LINK TO 1
977 1300 6004 GTF
978 1301 7500 SMA
979 1302 4533 CBERR /GTF DID NOT SAVE A 1 FOR LINK
980 1303 7300 CLA CLL
981 1304 6214 RDF /GET DATA FIELD
982 1305 7012 RTR
983 1306 7010 RAR /MOVE DF TO AC 9-11
984 1307 6224 RIF /GET INSTRUCTION FIELD
985 1310 3061 DCA SAVFLD
986 /*****
987 /TEST RTF TO RESTORE LINK
988 1311 1047 TAD K4000 /AC TO 4000
989 1312 1061 TAD SAVFLD /GET CORRECT IF AND DF INFORMATION
990 1313 6005 RTF /RESTORE LINK TO 1
991 1314 5315 JMP ,+1 /ENABLE INTERRUPT FOR CONSOLE INTERACTION.
992 1315 7420 SNL
993 1316 4533 CBERR /RTF DID NOT RESTORE LINK TO A 1
994 /*****
995 /TEST GTF TO SAVE A LINK OF 0, AND INT REQUEST AND INT ENABLE OF 0
996 1317 6002 IOF
997 1320 7300 CLA CLL /CLEAR AC AND LINK
998 1321 6004 GTF /GET FLAGS, LINK TO AC0
999 1322 0777 AND K7600 /ELIMINATE SAVE FIELD REGISTER AND USER BIT
1000 1323 7440 SZA /LINK, INT REQUEST, AND INT ENABLE ALL ZERO?
1001 1324 4533 CBERR /NO, GTF DID NOT SAVE CORRECTLY.
1002 /IF BIT0=1 LINK WAS SAVED AS 1 INSTEAD OF 0
1003 /IF BIT2=1 INT REQUEST WAS SAVED AS 1 INSTEAD OF 0
1004 /IF BIT4=1 INT ENABLE WAS SAVED AS 1 INSTEAD OF 0

```

```

1004 /*****
1005 /TEST CAF TO CLEAR AC AND LINK
1006 1325 6001 ION
1007 1326 7040 CMA /AC TO 7777
1008 1327 7020 CML /AC=7777 LINK TO 1
1009 1330 6007 CAF /CLEAR ALL FLAGS, CLEARS AC AND LINK, AND INT ENABLE
1010 1331 6305 KIE1 /DISABLE SLU#2
1011 1332 6325 KIE2 /DISABLE SLU#3
1012 1333 7450 SWA
1013 1334 7430 SZL
1014 1335 4533 CBERR /CAF FAILED TO CLEAR AC OR LINK
1015 /*****
1016 /TEST SKON TO NOT SKIP WHEN INTERRUPT ENABLE IS CLEAR
1017 1336 6000 SKON /SHOULD NOT SKIP
1018 1337 7410 SKP
1019 1340 4533 CBERR /SKON SKIPPED WHEN INT ENABLE CLEAR, OR CAF FAILED
1020 /TO CLEAR INT ENABLE
1021 /*****
1022 /TEST SKON TO SKIP ON INTERRUPT ON AND TURN INTERRUPT OFF
1023 1341 6001 ION /INTERRUPT ON
1024 1342 6000 SKON /SKIP IF INTERRUPT ON, TURN INTERRUPT OFF
1025 1343 4533 CBERR /ION DID NOT ENABLE INTERRUPT, OR SKON FAILED TO SKIP
1026 /*****
1027 /TEST THAT SKON TURNED OFF INTERRUPT
1028 1344 6000 SKON /SHOULD NOT SKIP
1029 1345 7410 SKP
1030 1346 4533 CBERR /SKON DID NOT TURN OFF INT, OR SKON SKIPS WHEN INT OFF
1031 /*****
1032 /TEST IOF TO DISABLE INTERRUPTS
1033 1347 6001 ION /ENABLE INTERRUPTS
1034 1350 7000 NOP /ALLOWS TIME FOR INTERRUPT DELAY TO SET
1035 1351 6002 IOF /TURN OFF INTERRUPT
1036 1352 6000 SKON
1037 1353 7410 SKP
1038 1354 4533 CBERR /IOF DID NOT DISABLE INTERRUPT
1039 1355 5776 JMP INTEN1
1040 1376 1400
1041 1377 1511
1042 1400 PAGE
1043 /*****
1044 /TEST PROPER OPERATION OF INT ENABLE.
1045 1400 6001 INTEN1, ION /INTERRUPT ON
1046 1401 6004 GTF /SHOULD GET A 1 FOR INT ENABLE
1047 1402 0045 AND K200 /MASK OUT INT ENABLE BIT
1048 1403 7450 SWA /SKIP IF INT ENABLE BIT SET
1049 1404 4533 CBERR /ION DID NOT SET INT ENABLE OR GTF DID NOT GET A 1 FOR INT ENABLE
1050 /*****
1051 /TEST RTF TO SET INT ENABLE AND TO CLEAR LINK
1052 1405 6002 IOF /CLEAR INT ENABLE
1053 1406 7300 CLA CLL
1054 1407 7020 CML /SET LINK
1055 1410 1061 TAD SAVFLD /GET IF AND DF INFORMATION
1056 1411 6005 RTF /RESTORE FLAGS, LINK TO 0, SET INTERRUPT ENABLE
1057 1412 5213 JMP ,+1 /ENABLE INTERRUPT FOR CONSOLE INTERACTION.
1058 1413 7430 SZL

```



```

1050 1414 4533 CBERR /RTF DID NOT CLEAR LINK
1059 1415 6000 SKON /SKIP IF INTERRUPT ON
1060 1416 4533 CBERR /RTF FAILED TO ENABLE INTERRUPTS
1061 /*****
1062 /TEST SRQ TO NOT SKIP WHEN NO INTERRUPT REQUEST
1063 ION /INTERRUPT ON
1064 1420 7300 CLA CLL /GIVE POWER FAIL A CHANCE TO INTERRUPT
1065 1421 6003 SRQ /SKIP ON INTERRUPT REQUEST
1066 1422 7410 SKP
1067 1423 4533 CBERR /SRQ SKIPPED WHEN NO INTERRUPT, OR ILLEGAL INTERRUPT
1068 /CAME UP DURING SRQ INSTRUCTION.
1069
1070 1424 4536 ENDTST
1071 /*****
1072 /CPU TEST 14 - TEST JUMPS AND JMS'S
1073 /*****
1074 /TEST JUMP DIRECT
1075 1425 4535 CPUT14, INTST
1076 1426 7300 CLA CLL /CLEAR AC AND LINK
1077 TAD KSTOP
1078 DCA 0 /PUT CBERR IN LOC 0 IN CASE JUMP FAILS TO GATE MD TO PC
1079 JMP +2 /FIRST JUMP TESTED
1080 CBERR /JMP FAILED TO JUMP DIRECT
1081 /*****
1082 /TEST JMS DIRECT
1083 1433 4235 JMS +2
1084 1434 4533 CBERR /JMS FAILED TO JUMP
1085 1435 0000 JMSLOC, 0 /JMS ENTRY POINT, PC STORAGE.
1086 1436 1235 TAD -1 /GET STORED PC
1087 CMA
1088 IAC /COMPLEMENT IT
1089 TAD POINTE
1090 1441 1103 SZR
1091 1442 7400 CBERR /JMS DID NOT STORE PROPER PC
1092 1443 4533 ENDTST
1093 /*****
1094 /TEST JUMP INDIRECT TO JUMP CORRECTLY
1095 1444 5646 JMP I +2
1096 1445 4533 CBERR /JMP INDIRECT FAILED TO JUMP
1097 1446 1447 -1 /POINTER FOR JMP INDIRECT ABOVE
1098 /*****
1099 /TEST JMS INDIRECT TO JUMP AND STORE PC
1100 1447 4651 JMS I +2
1101 1448 4533 CBERR /JMS INDIRECT FAILED TO JUMP
1102 1449 1452 -1 /JMS INDIRECT PC STORAGE
1103 TAD
1104 CMA
1105 IAC /COMPLEMENT STORED PC
1106 TAD POINTF /COMPARE IT TO EXPECTED VALUE
1107 1455 1104 SZR /WAS IT EQUAL?
1108 1456 7400 CBERR /NO, JMS INDIRECT STORED INCORRECT PC.
1109 1457 4533 ENDTST
1110 /*****
1111 /CPU TEST 15 - TEST COMBINED OPERATE MICROINSTRUCTIONS OF FORM 7XX0,7XX1
1112 /*****
1113 /ALL INSTRUCTIONS IN RANGE 7XX0,7XX1 ARE TESTED.

```

```

1113 /FOR EACH OF THE INSTRUCTIONS
1114 /EXECUTED, A TOTAL OF 8 PATTERNS OF AC, MQ, AND LINK CONTENTS ARE USED,
1115 /TO TEST FOR ERRORS IN THE EXECUTED INSTRUCTION A SIMULATION OF THE
1116 /INSTRUCTION IS ALSO PERFORMED, AND THE RESULTS OF THE INSTRUCTION ARE
1117 /COMPARED TO THE SIMULATION.
1118 /*****
1119 /CREATE THE INSTRUCTION TO BE USED, INSTRUCTION RANGE IS 7XX0, 7XX1.
1120 /*****
1121 CPUT15, INTST
1122 DCA POINTC /INITIALIZE INSTRUCTION MAKER
1123 SKP
1124 1462 4535 NXTONE, CLA CLL
1125 1463 3101 ISZ POINTC
1126 1464 7410 CML
1127 1465 7300 TAD POINTC
1128 1466 2101 TAD K7700
1129 1467 7020 /ALL COMBINATIONS TRIED?
1130 1468 1101 /YES,CPU TEST OVER
1131 1469 1052 JMS CLA
1132 1470 7700 JMP CPUDN
1133 1471 5777 CLA CLL
1134 1472 7300 TAD POINTC
1135 1473 1101 RAL
1136 1474 7000 TAD (7000 /MAKE NEXT INSTRUCTION
1137 1475 1101 DCA INSTRT /SAVE IT.
1138 1476 7000 TAD (7770
1139 1477 7000 TAD DATPAT
1140 /*****
1141 /SET UP AC, MQ, AND LINK FOR EXECUTION OF TEST INSTRUCTION
1142 /*****
1143 /NEXTPAT, CAM
1144 DCA SKPPED
1145 1504 7621 DCA SOMSKP
1146 1505 3057 DCA LKDATA
1147 1506 3060 TAD
1148 1507 1071 RAR /LOAD LINK
1149 1508 7010 K7600, 7600/CLA
1150 1509 7600 TAD MQDATA
1151 1510 1070 SWP /LOAD MQ
1152 1511 7500 TAD ACDATA /LOAD AC
1153 /*****
1154 /EXECUTE TEST INSTRUCTION
1155 INSTRT, 0 /EXECUTE INSTRUCTION
1156 1515 0000 ISZ SKPPED /DIDN'T SKIP
1157 1516 2057 /*****
1158 /SAVE RESULTS OF TEST INSTRUCTION
1159 1517 3054 DCA ACWAS
1160 1518 7040 CMA
1161 1519 7420 SWL
1162 1520 7200 CLA
1163 1521 3056 DCA LKWAS
1164 1522 7521 SWP
1165 1523 3055 DCA MQWAS
1166 1524 5774 JMP GRFSIM
1167 1525 1600
1168 1526 7770
1169 1527 7000
1170 1528 2134
1171 1529 1600

```

```

1167 /*****
1168 /SET UP FOR SIMULATED EXECUTION TO TEST RESULT
1169 GRFSIM, CMA
1170 DCA BIT11
1171 CMA
1172 DCA BIT8
1173 CMA
1174 DCA BIT7
1175 CMA
1176 DCA BIT6
1177 CMA
1178 DCA BIT5
1179 CMA
1180 DCA BIT4
1181 CMA
1182 DCA BIT3
1183 SETSIM, TAD I INSTR /GET INSTRUCTION
1184 RTL
1185 RTL
1186 SZL /WAS BIT 3 SET?
1187 ISZ BIT3 /YES, CLEAR POINTER
1188 NOP
1189 RAL
1190 SZL /BIT 4 SET?
1191 ISZ BIT4 /YES, CLEAR POINTER
1192 NOP
1193 RAL
1194 SZL /BIT 5 SET?
1195 ISZ BIT5 /YES, CLEAR POINTER
1196 NOP
1197 RAL
1198 SZL /BIT 6 SET?
1199 ISZ BIT6 /YES, CLEAR POINTER
1200 NOP
1201 RAL
1202 SZL /BIT 7 SET?
1203 ISZ BIT7 /YES, CLEAR POINTER
1204 NOP
1205 PAL
1206 SZL /BIT 8 SET ?
1207 ISZ BIT8 /YES, CLEAR POINTER
1208 NOP
1209 RTL
1210 RAL
1211 SZL /WAS BIT 11 SET?
1212 ISZ BIT11 /YES, CLEAR POINTER
1213 NOP
1214 CLA
1215 /*****
1216 /SET UP AC, MQ, AND LINK FOR SIMULATED EXECUTION
1217 DOSIMU, TAD LKDATA
1218 RAR
1219 CLA
1220 TAD MQDATA
1221 SWP /LOAD LINK

```

```

1222 TAD ACDATA /LOAD AC
1223 /*****
1224 /SIMULATED EXECUTION BEGINS
1225 /*****
1226 /GROUP 1 SIMULATION
1227 GR1SIM, ISZ BIT3 /WAS BIT 3 SET IN THE INSTRUCTION?
1228 JMP GR2GR3 /YES, IT IS A GROUP 2 OR 3.
1229 ISZ BIT4 /WAS BIT 4 SET?(GROUP 1)
1230 CLA /YES, SO CLEAR THE AC
1231 ISZ BIT5 /WAS IT CLL?
1232 CLL /YES.
1233 ISZ BIT6 /WAS IT CMA?
1234 CMA /YES
1235 ISZ BIT7 /WAS IT CML?
1236 CML /YES
1237 ISZ BIT11 /WAS IT IAC?
1238 TAD K1 /YES, ADD 1
1239 ISZ BIT8 /WAS IT RAR?
1240 RAR /YES
1241 ISZ SKPPED /INDICATE NO SKIP
1242 JMP I TEST /GO CHECK RESULTS
1243
1244 /*****
1245 /GROUP 2 SIMULATION
1246 GR2GR3, ISZ BIT11 /GROUP 2?
1247 JMP GROUP3 /NO, GROUP 3.
1248 GROUP2, ISZ BIT8 /REVERSE SENSE SKIPS?
1249 JMP REVSEN /YES, GO DO REVERSE
1250 ISZ BIT5 /IS IT SMA?
1251 SMA /YES
1252 SKP
1253 ISZ SOMSKP /SMA SKIPPED,
1254 ISZ BIT6 /IS IT SZA?
1255 SZA /YES
1256 SKP
1257 ISZ SOMSKP /SZA SKIPPED
1258 ISZ BIT7 /IS IT SNL?
1259 SNL /YES
1260 SKP
1261 ISZ SOMSKP
1262 OUT, DCA BIT7 /SAVE AC
1263 TAD SOMSKP
1264 SNA /ANY SKIP?
1265 ISZ SKPPED /NO
1266 CLA
1267 TAD BIT7 /REPLACE AC
1268 ISZ BIT4 /WAS IT CLAT?
1269 CLA /YES
1270 JMP I TEST
1271 /*****
1272 /REVERSE SENSE SKIPPING FOR GROUP 2
1273 REVSEN, ISZ BIT5 /WAS IT SPA?
1274 SNA /YES.
1275 SKP /SPA WOULD HAVE SKIPPED.
1276 JMP OUT /SPA WOULD NOT HAVE SKIPPED

```

1277	1741	2102	ISZ	BIT6	/WAS IT SNA?
1278	1742	7440	SZA		/YES.
1279	1743	7410	SKP		/SNA WOULD HAVE SKIPPED
1280	1744	5324	JMP	OUT	/SNA WOULD NOT HAVE SKIPPED
1281	1745	2007	ISZ	BIT7	/SZL?
1282	1746	7420	SNL		/YES
1283	1747	5323	JMP	OUT-1	/QUALIFIED SKIP
1284	1750	5324	JMP	OUT	/SZL WOULD NOT HAVE SKIPPED.
1285	1777	2000			
		2000			

PAGE

```

/*****
/OPERATE SIMULATION
/INDICATE NO SKIP
/CLAS?
/IS IT MQA?
/IS IT MQL?
/COMPARE RESULTS
/IT IS MQA, IS IT ALSO MQL? (SWP)
/YES, DO A SWP.
/NO, JUST DO MQA.
/COMPARE RESULTS
/MQ TO AC
/SAVE PREVIOUS MQ
/GET PREVIOUS AC
/PUT IN MQ
/PUT PREVIOUS MQ IN AC.
/TEST RESULTS
/*****
/COMPARE RESULTS OF SIMULATION TO ACTUAL RESULTS
/*****
/TEST LINKS
/SAVE SIMULATED AC
/LINK TO AC 11
/SAVE SIMULATED LINK
/GET EXPECTED LINK
/ADD IN LINK OBTAINED
/SET ERROR INDICATOR IF LINKS DIFFERENT
/CLEAR ERROR INDICATOR IF LINKS SAME
/*****
/LINKS AGREE. TEST THE AC CONTENTS
/GET EXPECTED AC CONTENTS
/COMPLEMENT IT
/GET ACTUAL RESULTS
/SAME?
/NO, SET ERROR INDICATOR
/*****
/AC CONTENTS OK. TEST MQ CONTENTS
/MQ TO AC
/SAVE MQ

```

1331	2041	1065	TAD	BIT8	
1332	2042	7040	CMA		/COMPLEMENT IT
1333	2043	7001	IAC		/COMPARE TO ACTUAL RESULTS
1334	2044	1055	TAD	MQWAS	/SAME?
1335	2045	7440	SZA		/NO, SET ERROR INDICATOR
1336	2046	3062	DCA	BIT3	/NO, SET ERROR INDICATOR
1337					/*****
1338					/CHECK FOR SIMULATED INSTRUCTION AND ACTUAL INSTRUCTION TO BOTH HAVE
1339					/SKIPPED OR BOTH TO HAVE NOT SKIPPED
1340	2047	1057	SKPCHK, TAD	SKPPED	/GET SKIP INDICATOR
1341	2050	7010	RAR		/AC 11 TO LINK
1342	2051	7200	CLA		
1343	2052	7420	SNL		/BOTH SKIP OR BOTH NOT SKIP?
1344	2053	5255	JMP	SIMERR	/YES, BOTH SKIPPED OR BOTH DIDN'T
1345					
1346	2054	4533	SPE1, C8ERR		/THE INSTRUCTION SKIPPED WHEN EXECUTED, AND A
1347					/SIMULATION OF THE SAME INSTRUCTION DID NOT, OR VICE VERSA.
1348					/*****
1349					/TEST FOR ANY SIMULATION DATA ERRORS
1350	2055	1062	SIMERR, TAD	BIT3	/GET ERROR INDICATOR
1351	2056	7450	SNA		/ANY SIMULATION ERRORS?
1352	2057	5261	JMP	TESTPT	/NO
1353					
1354	2060	4533	SPE2, C8ERR		/OPERATE INSTRUCTION FAILED.
1355					/*****
1356					/SIMULATION AGREES WITH ACTUAL. SEE IF ALL DATA PATTERNS HAVE
1357					/BEEN TRIED WITH THIS INSTRUCTION.
1358					/ALL PATTERNS TRIED?
1359	2061	2177	TESTPT, ISZ	DATPAT	/NO, TRY NEXT PATTERN
1360	2062	5303	JMP	TRNXPT	
1361	2063	3067	DCA	ACDATA	
1362	2064	3070	DCA	MQDATA	
1363	2065	3071	DCA	LKDATA	
1364	2066	1505	TAD I	INSTTR	/GET INSTRUCTION
1365	2067	7010	RAR		/BIT11 TO LINK
1366	2070	7430	SZL		/HAS INSTRUCTION BEEN TRIED WITH BIT11=1?
1367	2071	5732	JMP I	NXTONN	/YES, DO NEXT INSTRUCTION (JMP TO NXTONE)
1368	2072	2505	ISZ I	INSTTR	/UPDATE INSTRUCTION
1369	2073	0045	AND	K200	/MASK OUT MQ TYPE BIT
1370	2074	7450	SNA		/MQ TYPE?
1371	2075	5733	JMP I	TESAGG	/NO, GO DO IT
1372	2076	7200	CLA		
1373	2077	1505	TAD I	INSTTR	
1374	2100	0377	AND	(7721	/MASK OUT BITS NOT ALLOWED(EAE)
1375	2101	3505	DCA I	INSTTR	
1376	2102	5733	JMP I	TESAGG	/JMP TO TESAGN
1377					/*****
1378					/CREATE NEXT DATA SET UP
1379	2103	7000	TRNXPT, NOP/JMS	POINT	
1380	2104	1177	TAD	DATPAT	
1381	2105	7010	RAR		/AC BIT TO LINK
1382	2106	7200	CLA		
1383	2107	1046	TAD	K2525	
1384	2110	7420	SNL		/AC TO BE SET?
1385	2111	7200	CLA		/NO

```

1386 2112 3067 DCA ACDATA
1387 2113 1177 TAD DATPAT
1388 2114 7012 RTR
1389 2115 7206 CLA
1390 2116 1535 TAD I INSTTR
1391 2117 7420 SNL /NO TO BE SET?
1392 2120 7200 CLA /NO
1393 2121 3070 DCA MODATA
1394 2122 1177 TAD DATPAT
1395 2123 7012 RTR
1396 2124 7010 RAR
1397 2125 7420 SNL /LINK TO BE SET?
1398 2126 7200 CLA /NO
1399 2127 3071 DCA LKDATA
1400 2130 5731 JHP I NXTPTT /JMP TO NXTPAT
1401 2131 1504 NXTPTT, NXTPAT
1402 2132 1465 NXTONN, NXTONE
1403 2133 1502 TESAGG, TESAGN
1404 2134 4536 CPUDN, ENDTST
1405 2135 3075 DCA TSTNC /CLEAR TEST NUMBER
1406 2136 5776 JMP BEGEXM
1407
1408
1409 2176 3000
1410 2177 7721
1411 PAGE
1412 /ROUTINE TO PRINT "CPU TESTING" MESSAGE
1413 2200 0000 PCPUMS, 0
1414 2201 4521 C0CRLF
1415 2202 4522 C0PRNT
1416 2203 2210 CPUMES
1417 2204 4522 C0PRNT
1418 2205 2213 TESTMS
1419 2206 4521 C0CRLF
1420 2207 5600 JMP I PCPUMS
1421 2210 0320 CPUMES, TEXT "CPU "
2211 2540
2212 0000
1422 2213 2405 TESTMS, TEXT "TESTING"
2214 2324
2215 1116
2216 0700
1423
1424 /ROUTINE TO INITIALIZE FOR TEST
1425
1426 2217 0000 TSTIN, 0
1427 2220 7300 CLA CLL
1428 2221 2075 ISZ TSTNO /INCREMENT TEST NUMBER
1429 2222 7000 NOP
1430 2223 1217 TAD TSTIN /GET TEST LOOP ADDRESS
1431 2224 3226 DCA TSLOP /SAVE TEST LOOP ADDRESS
1432 2225 5617 JMP I TSTIN
1433 2226 0000 TSLOP, 0
1434

```

```

1435 /ROUTINE TO HANDLE SCOOP LOOPING AT END OF TEST IF ERROR OCCURRED
1436
1437 2227 0000 TSTEND, 0
1438 2230 7300 CLA CLL
1439 2231 4517 KRCHK /CHECK FOR CONSOLE REQUEST
1440 2232 1020 TAD PSR /CHECK PSEUDO SWITCH REGISTER
1441 2233 7006 RTL /FOR BIT 2=1 TEST LOOP
1442 2234 7700 SMA CLA /SKIP IF SET
1443 2235 5627 JMP I TSTEND /CONTINUE TO NEXT TEST
1444 2236 5626 JMP I TSLOP /LOOP ON CURRENT TEST
1445
1446 /ROUTINE TO CHECK FOR KEYBOARD INTERVENTION
1447
1448 2237 0000 CHKKB, 0
1449 2240 7300 CLA CLL
1450 2241 6035 KIE /DISABLE KEYBOARD INTERRUPTS
1451 2242 4516 CHKKSF /CHECK FOR KEYBOARD FLAG
1452 2243 7000 NOP
1453 2244 7201 CLA IAC
1454 2245 6035 KIE /ENABLE KEYBOARD INTERRUPTS
1455 2246 7300 CLA CLL
1456 2247 5637 JMP I CHKKB
1457

```

```

1458
1459          3000 *3000
1460          /*****
1461          /MEMORY EXTENSION TEST
1462          /*****
1463
1464
1465
1466
1467
1468          /*****
1469          /MEMORY EXTENSION TEST 1
1470          /TEST CDF AND RDF, USE CDF TO SET THE DATA
1471          /FIELD AND RDF TO READ THE DATA FIELD.
1472          /DO ALL COMBINATIONS 0 TO 3 & 7.
1473          /
1474
1475          3000 7300 BEGEXM, CLA CLL
1476          3001 6002 IOF
1477          3002 1020 TAD PSR          /EXECUTE MEMORY EXTENSION TEST??????
1478          3003 0377 AND (0040
1479          3004 7640 SZA CLA
1480          3005 5776 JMP BEGRTC      /NO
1481          3006 4520 FIXL           /FIXUP INTERRUPT LINKAGE
1482          3007 4775 JMS PMEMES     /PRINT MEMORY EXTENSION TESTING MESSAGE
1483                                     /IF NOT UNDER APT CONTROL.
1484          3010 4535 EXMT1, INTST
1485          3011 6007 CAF
1486          3012 6001 ION
1487          /
1488          3013 6201 DF0, CDF 00      /DF 0
1489          3014 6214 RDF
1490          3015 7440 SZA
1491          3016 4533 CBERR          /ERROR, CDF OR RDF FAILED.
1492          /
1493          3017 6271 DF7, CDF 70      /DF 7
1494          3020 6214 RDF
1495          3021 1374 TAD (7747
1496          3022 7040 CMA           /AC = 0
1497          3023 7440 SZA
1498          3024 4533 CBERR          /ERROR, CDF OR RDF FAILED.
1499          /
1500          3025 6211 DF1, CDF 10      /DF 10
1501          3026 6214 RDF
1502          3027 1373 TAD (7767
1503          3030 7040 CMA           /AC=0
1504          3031 7440 SZA
1505          3032 4533 CBERR
1506          /
1507          3033 6221 DF2, CDF 20      /DF2
1508          3034 6214 RDF
1509          3035 1372 TAD (7757
1510          3036 7040 CMA           /AC=0
1511          3037 7440 SZA
1512          3040 4533 CBERR          /CDF2 OR RDF FAILED

```

```

1513          /
1514          3041 6231 DF3, CDF 30      /DF 3
1515          3042 6214 RDF
1516          3043 1374 TAD (7747
1517          3044 7040 CMA           /AC=0
1518          3045 7440 SZA
1519          3046 4533 CBERR          /CDF 3 OR RDF FAILED.
1520          /
1521          3047 6201 DF4, CDF 00      /
1522          3050 6214 RDF
1523          3051 7440 SZA
1524          3052 4533 CBERR          /CDF 0 OR RDF FAILED.
1525          3053 6000 SKON          /SKP IF ION, DISABLE INT SYSTEM.
1526          3054 4533 CBERR
1527          3055 4536 ENDTST
1528          /*****
1529          /CONSOLE PACKAGE REQUESTS ARE INHIBITED FROM HERE TO END OF EXT. MEMORY TESTS.
1530          /*****
1531          /MEMORY EXTENSION TEST 2
1532          /NOW TEST SAVE FIELD BITS 9-11 WITH
1533          /RIB, PI IS ENABLED, TELEPRINTER FLAG IS
1534          /USED FOR INTERRUPT, DO ALL COMBINATIONS 0 TO 3 & 7.
1535          /
1536          /THIS TEST OVERWRITES LOCATIONS 0 & 1 IN FIELD 0.
1537          /
1538          3056 4535 EXMT2, INTST
1539          3057 6201 CDF 00          /DF0
1540          3060 1376 TAD (JMP I 0
1541          3061 3001 DCA 1          /C(1)=JMP I 0
1542          3062 6041 TSF           /TEST TTY FLAG
1543          3063 4506 JMS I XTFLG     /SET FLAG
1544          /
1545          3064 6001 IB0, ION          /ENABLE PI
1546          3065 7200 CLA           /AN INTERRUPT SHOULD OCCUR AFTER THIS INST IS EXECUTED.
1547          3066 6234 RIB           /READ SF
1548          3067 7440 SZA
1549          3070 4533 CBERR          /RIB FAILED.
1550          /
1551          3071 6211 IB1, CDF 10      /DF 1
1552          3072 6001 ION
1553          3073 7200 CLA           /INTERRUPT HERE
1554          3074 6214 PDF          /DF SHOULD BE 0 AFTER A PI
1555          3075 7440 SZA
1556          3076 4533 CBERR          /DF NOT CLEARED, OR NO PI.
1557          /
1558          3077 6234 RIB           /READ SF
1559          3100 1371 TAD (7776
1560          3101 7040 CMA           /AC=0
1561          3102 7440 SZA
1562          3103 4533 CBERR          /RIB OR SF FAILED.
1563          /
1564          3104 6221 IB2, CDF 20      /DF 2
1565          3105 6001 ION
1566          3106 7200 CLA           /INTERRUPT HERE
1567          3107 6214 RDF          /SHOULD BE 0 AFTER PI

```

```

1568 3110 7440 SZA
1569 3111 4533 CBERR /DF NOT CLEARED, OR NO PI,
/
1570
1571 3112 1370 TAD (7775
1572 3113 6234 /AC=7777
1573 3114 7940 CMA /#0
1574 3115 7440 SZA
1575 3116 4533 CBERR /RIB OR SF FAILED,
/
1576
1577 3117 6231 IB3, CDF 30 /DF3
1578 3120 6001 ION
1579 3121 7200 CLA /INTERRUPT HERE
1580 3122 6214 RDF /DF SHOULD BE CLEARED
1581 3123 7440 SZA
1582 3124 4533 CBERR /DF NOT CLEARED,
/
1583
1584 3125 6234 RIB /AC=7777
1585 3126 1367 TAD (7774
1586 3127 7040 CMA /AC=0
1587 3130 7440 SZA
1588 3131 4533 CBERR /RIB OR SF FAILED,
/
1589
1590 3132 6201 IIB0, CDF 00 /DF0
1591 3133 6001 ION
1592 3134 7200 CLA /INTERRUPT HERE
1593 3135 6214 RDF /DF MUST=0 AFTER PI
1594 3136 7440 SZA
1595 3137 4533 CBERR /DF NOT 0 AFTER PI,
1596 3140 6234 RIB
1597 3141 7440 SZA
1598 3142 4533 CBERR /RIB OR SF FAILED,
/
1599
1600 3143 6271 IB7, CDF 70 /DF 7
1601 3144 6001 ION
1602 3145 7200 CLA /INTERRUPT HERE
1603 3146 6214 RDF /DF MUST = 0 AFTER PI
1604 3147 7440 SZA
1605 3150 4533 CBERR /DF NOT 0,
/
1606
1607 3151 6234 RIB
1608 3152 1367 TAD (7774
1609 3153 7940 CMA
1610 3154 7440 SZA
1611 3155 4533 CBERR /RIB OR SF FAILED,
/
1612
1613 3156 4536 ENDTST
1614 3157 5760 JMP I ,+1 /INTERRUPTS ARE NOW DISABLED DUE TO PREVIOUS INTERRUPT,
1615 3160 3200 EXMT3 /TTY OUTPUT FLAG IS SET (& ENABLED),
1616 3167 7774
1617 3170 7775
1618 3171 7776
1619 3172 7757
1620 3173 7767
1621 3174 7747
1622 3175 5200

```

```

1623 3176 5400
1624 3177 0040
/
1625
1626
1627
1628
1629
1630
1631
1632
1633 3200 4535
1634 3201 4510
1635 3202 7001
1636 3203 3074
1637 3204 1300
1638 3205 1042
1639 3206 3207
1640 3207 6201
1641 3210 1074
1642 3211 3777
1643 3212 2107
1644 3213 7410
1645 3214 5233
1646 3215 1107
1647 3216 7040
1648 3217 7640
1649 3220 5226
1650 3221 1376
1651 3222 3207
1652 3223 1377
1653 3224 3074
1654 3225 5207
1655 3226 1042
1656 3227 1207
1657 3230 3207
1658 3231 2074
1659 3232 5207
1660
1661 3233 4510
1662 3234 7001
1663 3235 3074
1664 3236 1300
1665 3237 1042
1666 3240 3241
1667 3241 6201
1668 3242 1777
1669 3243 3073
1670 3244 1073
1671 3245 7041
1672 3246 1074
1673 3247 7650
1674 3250 5255
1675 3251 1074
1676 3252 7421
/
PAGE
/*****
/MEMORY EXTENSION TEST 3
/NOW TEST DCA I AND TAD I TO ALL STACKS(1-3 & 7), EACH STACK WILL
/CONTAIN ITS DF# IN LOCATION 0007,
/
/THIS TEST OVERWRITES LOCATION 7 IN DF#S 1,2,3, & 7,
/
EXMT3, INTST
JMS I XSTKS /INITIALIZE STKS,
IAC
DCA NDF /DF NUMBER = 1 TO START
TAD CCDF /6201
TAD K10
DCA ,+1 /DF 001 TO START WITH
CDF 00 /WILL BE INCREMENTED
TAD NDF /DF#
DCA I (0007 /PUT IN 0007 OF STACK
ISZ STKS /ALL STACKS WHEN 0
SKP
JMP TADI /TEST TAD I
TAD STKS /IF STKS=-1 DO FIELD 7 (CPRAM),
CMA
SZA CLA
JMP DNF7 /NOT READY FOR FIELD 7 YET,
TAD (CDF 70 /DO FIELD 7,
DCA DFLD
TAD (0007
DCA NDF
JMP DFLD
DNF7, TAD K10
TAD DFLD /INCR, CDF IOT
DCA DFLD
ISZ NDF
JMP DFLD
/
TADI, JMS I XSTKS /RESET UP STKS,
IAC
DCA NDF /DF#1 AGAIN
TAD CCDF /6201
TAD K10
DCA ,+1
TFLD, CDF 00
TAD I (0007 /AC=DF CONTENTS NOW
DCA DAT /SAVE TEMP
TAD DAT
CIA
TAD NDF /2'S COMP
SNA CLA /BETTER BE EQUAL
JMP UPSTK /EQUAL
TAD NDF
MQL

```

```

1677 3253 1073 TAD DAT
1678 3254 4533 CBERR /AC,MQ=ACTUAL DATA READ FROM EXT. FIELD,EXPECTED DATA
1679 3255 2107 UPSTK, ISZ STKS /ALL WHEN 0
1680 3256 7410 SKP
1681 3257 5275 JMP IBSF /NEXT TEST
1682 3260 1107 TAD STKS /IF STKS=-1 CHECK FIELD 7.
1683 3261 7040 CMA
1684 3262 7640 SZA CLA
1685 3263 5271 JMP TNF7 /NOT READY FOR FIELD 7 YET.
1686 3264 1376 TAD (CDF 70 /CHECK FIELD 7.
1687 3265 3241 DCA TFLD
1688 3266 1377 TAD (0007
1689 3267 3074 DCA NDF
1690 3270 5241 JMP TFLD
1691 3271 1042 TNF7, TAD K10
1692 3272 1241 TAD TFLD /CDF IOT + 10
1693 3273 3241 DCA TFLD
1694 3274 2074 ISZ NDF
1695 3275 5241 JMP TFLD
1696 3276 4536 IBSF, ENDTST
1697 /
1698 /*****
1699 /MEMORY EXTENSION TEST 4
1700 /CIF TEST. CHECKS THE ABILITY OF A CIF-ION-NOP-JMP OR
1701 /CIF-ION-NOP-JMS SEQUENCE TO DO THE FOLLOWING:
1702 /1. CIF ENABLE MAR TO IB TRANSFER.
1703 /2. INHIBIT INTERRUPT TILL JMP OR JMS EXECUTED.
1704 /3. INTERRUPT AFTER JMP OR JMS EXECUTED.
1705 /4. JMP OR JMS ENABLES IB TO IF TRANSFER.
1706 /5. INTERRUPT ENABLES IF TO SF TRANSFER.
1707 /
1708 /THIS TEST OVERWRITES LOCATIONS 0,1,2, & 3 IN FIELD 0;
1709 / ALSO LOCATIONS 1,3347, 3422 & 3425 IN FIELDS 1,2, & 3.
1710 /
1711 /SET UP FOR CIF-ION-NOP-JMP CHECK.
1712
1713 3277 4535 EXMT4, INIST
1714 3300 6201 CCDF, CDF 00 /SET LOCS 1-2 TO ISZ 0.
1715 3301 1375 TAD (ISZ 0 /JMP I 0 RESPECTIVELY.
1716 3302 3001 DCA 1
1717 3303 1350 TAD KNOP
1718 3304 3002 DCA 2
1719 3305 1374 TAD (JMP I 0
1720 3306 3003 DCA 3
1721 /
1722 /NOW STORE HALTS IN LOC1, CIFJMP+1,
1723 /AND CIFJMS+1 OF ALL EXTENDED FIELDS.
1724 /
1725 /NOTE: IF THE HALT IS EXECUTED, THE SYSTEM WILL TRAP TO CONTROL
1726 / PANEL MEMORY AND PRINT OUT THE FACT THAT A HALT OCCURED.
1727 / THE FIELD AND LOCATION WHERE THE HALT WAS ENCOUNTERED WILL
1728 / ALSO BE IDENTIFIED, THE SYSTEM MUST BE RE-BOOTED TO
1729 / RESTART THE DIAGNOSTIC. SORRY ABOUT THAT....
1730 /
1731 3307 4510 JMS I XSTKS

```

```

1732 3310 2107 ISZ STKS /FIELD 0 NOT TESTED.
1733 3311 1330 TAD LCDF
1734 3312 1042 TAD K10
1735 3313 3314 HLTs, DCA ,+1
1736 3314 6211 CDF 10
1737 3315 1373 TAD (HLT
1738 3316 3441 DCA I K1
1739 3317 1373 TAD (HLT
1740 3320 3766 DCA I CAB
1741 3321 1373 TAD (HLT
1742 3322 3767 DCA I CAC
1743 3323 2107 ISZ STKS
1744 3324 7410 SKP
1745 3325 5330 JMP ,+3
1746 3326 1314 TAD HLTs
1747 3327 5312 JMP HLTs-2
1748 3330 6201 LCDF, CDF 00
1749 3331 6241 TSF /ENSURE T10 FLAG SET.
1750 3332 4506 JMS I XTFLG
1751 3333 1372 AGAIN1, TAD (CIF /INITIALIZE TO CIF 00.
1752 3334 3346 DCA CIFJMP
1753 3335 3365 DCA CIFCK /INITIALIZE I.F. CHECK TO 0.
1754 3336 4510 JMS I XSTKS /INITIALIZE STKS TO -4.
1755 3337 2107 ISZ STKS /FIELD 0 NOT TESTED.
1756 3340 1346 CIFJPL, TAD CIFJMP
1757 3341 1042 TAD K10
1758 3342 3346 DCA CIFJMP
1759 3343 1365 TAD CIFCK
1760 3344 1042 TAD K10
1761 3345 3365 DCA CIFCK
1762 3346 6202 CIFJMP, CIF 00 /MODIFIED TO CURRENT FIELD
1763 /UNDER TEST.
1764 3347 6001 ION
1765 3350 7000 NOP
1766 3351 5352 JMP ,+1 /INTERRUPT SHOULD OCCUR AFTER THIS JMP.
1767 3352 4533 CBERR /ERROR. NO PI OR INHIBIT PI.
1768 3353 6234 RIB
1769 3354 7041 CIA
1770 3355 1365 TAD CIFCK
1771 3356 7650 SNA CLA
1772 3357 5771 JMP CIFOK
1773 3360 1365 TAD CIFCK
1774 3361 7421 MQL /LOAD MQ
1775 3362 6234 RIB
1776 3363 4533 CAD, CBERR /ERROR. I.B. TO I.F. TRANSFER
1777 /FAILED AFTER CIF-JMP. BAD
1778 /I.F. IN AC, GOOD I.F. IN MQ.
1779 3364 5771 JMP CIFOK
1780 3365 0000 CIFCK, 0
1781 3366 3347 CAB, CIFJMP+1
1782 3367 3422 CAC, CIFJMS+1
1783
1784 3371 3400
1785 3372 6202
1786 3373 7402

```

1787 3374 5430
 1788 3375 2000
 1789 3376 6271
 1790 3377 0000
 1791 3400 2107
 1792 3401 5777
 1793 3402 7230
 1794 3403 6201
 1795 3404 6041
 1796 3405 4506
 1797 3406 1376
 1798 3407 3221
 1799 3410 3244
 1800 3411 4510
 1801 3412 2107
 1802 3413 1221
 1803 3414 1042
 1804 3415 3221
 1805 3416 1244
 1806 3417 1042
 1807 3420 3244
 1808 3421 6202
 1809
 1810 3422 6001
 1811 3423 7000
 1812 3424 4225
 1813 3425 0000
 1814 3426 4533
 1815 3427 6234
 1816 3430 7041
 1817 3431 1244
 1818 3432 7650
 1819 3433 5240
 1820 3434 1244
 1821 3435 7421
 1822 3436 6234
 1823 3437 4533
 1824
 1825
 1826 3440 2107
 1827 3441 5213
 1828 3442 4536
 1829 3443 5245
 1830 3444 0000
 1831
 1832
 1833
 1834
 1835
 1836
 1837
 1838
 1839 3445 4535
 1840 3446 4510

PAGE
 CIFOX, ISZ STKS /DONE?
 JMP CIFJPL /NO, DO NEXT FIELD
 IBSF1, CLA /TEST CIF-JMS
 CDF 00
 TSF /ENSURE TTO FLAG SET.
 JMS I XTFLG
 AGAIN2, TAD (CIF /INIT. TO CIF 00.
 DCA CIFJMS
 DCA CIFCK1 /INIT. I,F, CHECK TO 0.
 JMS I XSTKS /INITIALIZE STKS.
 ISZ STKS /FIELD 0 NOT TESTED.
 CIFJSL, TAD CIFJMS
 TAD K10
 DCA CIFJMS
 TAD CIFCK1
 TAD K10
 DCA CIFCK1
 CIFJMS, CIF 00 /MODIFIED TO CURRENT FIELD
 /UNDER TEST.
 ION
 NOP
 JMS ,+1 /INTERRUPT SHOULD OCCUR AFTER THIS JMS.
 0
 CSERR /ERROR, NO PI OR INHIBIT PI.
 RIB
 CIA
 TAD CIFCK1
 SNA CLA
 JMP CIFOK1
 TAD CIFCK1
 TAD CIFCK1
 MQL
 RIB
 CAE, CSERR /ERROR, I,B, TO I,F, TRANSFER
 /FAILED AFTER CIF-JMS, BAD
 /I,F, IN AC, GOOD I,F, IN MO.
 /DONE?
 CIFOX1, ISZ STKS /NO, DO NEXT FIELD.
 JMP CIFJSL
 ENDTST
 JMP EXMT5 /YES, GO ON TO NEXT TEST
 CIFCK1,0
 /
 /*****
 /MEMORY EXTENSION TEST 5
 /TEST GTF FOR FLAG AND SAVE FIELDS
 /GET SAVE FIELDS AFTER INTERRUPT
 /CHECK INTERRUPT INHIBIT, DO ALL
 /COMBINATIONS 0 TO 3.
 /
 EXMT5, INTST
 JMS I XSTKS

1841 3447 1375
 1842 3450 3001
 1843 3451 1374
 1844 3452 3261
 1845 3453 1261
 1846 3454 0373
 1847 3455 7120
 1848 3456 7010
 1849 3457 7012
 1850 3460 3310
 1851 3461 6201
 1852 3462 6041
 1853 3463 4506
 1854 3464 6001
 1855 3465 7340
 1856 3466 6024
 1857 3467 3311
 1858 3470 1311
 1859 3471 7041
 1860 3472 1310
 1861 3473 7650
 1862 3474 5301
 1863 3475 1310
 1864 3476 7421
 1865 3477 1311
 1866 3500 4533
 1867 3501 1042
 1868 3502 1261
 1869 3503 3261
 1870 3504 2107
 1871 3505 5253
 1872 3506 4536
 1873 3507 5772
 1874 3510 0000
 1875 3511 0000
 1876
 1877 3572 3600
 1878 3573 0070
 1879 3574 6201
 1880 3575 5400
 1881 3576 6202
 1882 3577 3340
 1883
 1884
 1885
 1886
 1887
 1888
 1889
 1890 3600 4535
 1891 3601 1377
 1892 3602 3001
 1893 3603 1376
 1894 3604 3002

TAD (JMP I 0 /SET FOR RETURN
 DCA 1
 TAD (CDF
 DCA XSDF
 MGTf, TAD XSDF /GET FIRST FIELD
 AND (0070 /INT REQUEST FLAG.
 STL
 RAR
 RTR
 DCA XSAV
 XSDf, CDF 00
 TSF /ENSURE TTY FLAG SET.
 JMS I XTFLG
 ION
 CLA CLL CMA /INTERRUPT HERE,CHECK FOR JAM ON GTF
 GTF /GET THE FLAGS
 DCA FSAV
 TAD FSAV
 CIA
 TAD XSAV /TTY + CURRENT FIELD
 SNA CLA
 JMP CKMF
 TAD XSAV
 MQL
 TAD FSAV
 CKMF, TAD K10 /AC,MQ=ACTUAL FLAGS,EXPECTED FLAGS
 TAD XSDF
 DCA XSDF
 ISZ STKS /MORE FIELDS TO CHECK
 JMP MGTf
 ENDTST
 JMP EXMT6 /YES, GO TO NEXT TEST
 XSAV, 0
 FSAV, 0
 /
 PAGE
 /*****
 /MEMORY EXTENSION TEST 6
 /TEST ION AND LINK FROM RTF
 /TEST INTERRUPT INHIBIT BEFORE JMP
 /GET THE FLAGS WITH GTF.
 /
 /THIS TEST OVERWRITES LOCATIONS 1 & 2 IN FIELD 0,
 EXMT6, INTST /TTY FLAG SET BEFORE ENTRY TO TEST.
 TAD (ISZ 0
 DCA 1
 TAD (JMP I 0
 DCA 2

1895 3605 6005
 1896 3606 5207
 1897 3607 4533
 1898 3610 6004
 1899 3611 3264
 1900 3612 1264
 1901 3613 7041
 1902 3614 1375
 1903 3615 7650
 1904 3616 5223
 1905 3617 1375
 1906 3620 7421
 1907 3621 1264
 1908 3622 4533
 1909 3623 1374
 1910 3624 6005
 1911 3625 7240
 1912 3626 6004
 1913 3627 3264
 1914 3630 1264
 1915 3631 7041
 1916 3632 1373
 1917 3633 7640
 1918 3634 5253
 1919 3635 6005
 1920 3636 7300
 1921 3637 6004
 1922 3640 3264
 1923 3641 1264
 1924 3642 7041
 1925 3643 1372
 1926 3644 7640
 1927 3645 5257
 1928 3646 5247
 1929 3647 4533
 1930 3650 7300
 1931 3651 4536
 1932 3652 5771
 1933
 1934 3653 1373
 1935 3654 7421
 1936 3655 1264
 1937 3656 4533
 1938 3657 1372
 1939 3660 7421
 1940 3661 1264
 1941 3662 4533
 1942 3663 5250
 1943 3664 0000
 1944 3771 0000
 1945 3772 0200
 1946 3773 4200
 1947 3774 5200
 1948 3775 1000
 1949 3776 5400

RTF
 JMP .+1
 CBERR
 GTF
 DCA SFLGS
 TAD SFLGS
 CIA
 TAD (1000
 SNA CLA
 JMP L61
 TAD (1000
 MQL
 TAD SFLGS
 CBERR
 TAD (5200
 RTF
 CLA CMA
 GTF
 DCA SFLGS
 TAD SFLGS
 CIA
 TAD (4200
 SZA CLA
 JMP L62
 RTF
 CLA CLL
 GTF
 DCA SFLGS
 TAD SFLGS
 CIA
 TAD (0200
 SZA CLA
 JMP RPERR
 JMP .+1
 CBERR
 CLA CLL
 ENDTST
 JMP EXMT7
 TAD (4200
 MQL
 TAD SFLGS
 CBERR
 TAD (0200
 MQL
 TAD SFLGS
 CBERR
 JMP RTFRP
 SFLGS, 0

/LOADS IN & DF,EN INT.,SETS INT INH.
 /INTERRUPT HERE;JMP LOADS IF FROM IN,CLRS INT INH.
 /INTERRUPT WAS INHIBITED.

/CHECK FOR LINK=0,INT REQ=1,INT EN=0
 /CHECK OK

/AC,NO=ACTUAL FLAGS,EXPECTED FLAGS

/CHECK FOR JAM ON GTF
 /EXPECTED BITS ARE LINK=1,INT EN=1

/ (NO INT REQ - TTY FLAG BECAUSE
 / INT INH=1 TILL NEXT JMS OR JMP).

/REPLACE ION,INT INH

/EXPECTED BITS ARE LINK=0,INT REQ=0 BECAUSE INT INH=1

/SOMETHING WRONG WITH FLAGS
 /INTERRUPT HERE
 /INTERRUPT WAS INHIBITED.

/AC,NO=ACTUAL FLAGS,EXPECTED FLAGS

/AC,NO=ACTUAL FLAGS,EXPECTED FLAGS

1950 3777 2000
 4000
 1951
 1952
 1953
 1954
 1955
 1956
 1957
 1958
 1959
 1960
 1961
 1962
 1963 4000 4535
 1964 4001 6002
 1965 4002 3270
 1966 4003 4510
 1967 4004 1377
 1968 4005 3206
 1969 4006 6201
 1970 4007 4252
 1971 4010 2107
 1972 4011 7410
 1973 4012 5220
 1974 4013 1042
 1975 4014 1206
 1976 4015 3206
 1977 4016 2273
 1978 4017 5206
 1979
 1980 4020 7300
 1981 4021 6201
 1982 4022 3270
 1983 4023 4510
 1984 4024 1376
 1985 4025 3226
 1986 4026 6202
 1987 4027 5775
 1988 4030 3271
 1989 4031 1271
 1990 4032 7041
 1991 4033 1270
 1992 4034 7650
 1993 4035 5242
 1994 4036 1270
 1995 4037 7421
 1996 4040 1271
 1997 4041 4533
 1998 4042 2107
 1999 4043 7410
 2000 4044 5266
 2001 4045 1226
 2002 4046 1042
 2003 4047 3226

PAGE
 /
 /*****
 /MEMORY EXTENSION TEST 7
 /CONFIDENCE CHECK ON ALL EXISTENT FIELDS.
 /MAKE SURE DCA I AND TAD I ARE TO CORRECT STACK.
 /MAKE SURE JUMP IS TO CORRECT STACK.
 /CHECK ALL COMBINATIONS.
 /FIELDS WILL CONTAIN THEIR DF NUMBER IN LOC.0
 /
 /THIS TEST OVERWRITES LOCATIONS 0,1,2,3, & 7 OF FIELDS 0,1,2, & 3.
 /
 EXMT7, INTST
 IOF
 DCA NUMX
 JMS I XSTKS
 TAD (CDF
 DCA .+1
 CDF
 JMS FILCOR
 ISZ STKS
 SKP
 JMP CONCHK
 TAD K10
 TAD FDWRD
 DCA FDWRD
 ISZ NUMX
 JMP FDWRD
 /
 CONCHK, CLA CLL
 CDF 00
 DCA NUMX
 JMS I XSTKS
 TAD (CIF
 DCA .+1
 CIF
 JMP I (0001
 DCA SNUMX
 TAD SNUMX
 CIA
 TAD NUMX
 SNA CLA
 JMP X1
 TAD NUMX
 MQL
 TAD SNUMX
 CBERR
 ISZ STKS
 SKP
 JMP XRTF1
 TAD CONCH
 TAD K10
 DCA CONCH

/MODIFIED UNDER TEST
 /ARE ALL STACKS DONE
 /CHECK RESULTS
 /UPDATE FIELD CHANGE
 /START WITH FIELD 0
 /MODIFIED UNDER TEST
 /RETURN HERE FROM FIELDS
 /GOOD FIELD
 /AC,NO=ACTUAL DATA , EXPECTED DATA
 /CHECK ALL AVAILABLE STACKS.
 /UPDATE FIELD CHANGE

```

2004 4050 2270 ISZ NUMX
2005 4051 5226 JMP CONCH
2006
2007 4052 0000 /FILCOR, 0000 /INSTRUCTIONS FOR FIELDS
2008 4053 1270 TAD NUMX /MODIFIED TO DF#
2009 4054 3774 DCA I (0000
2010 4055 1373 TAD (1000
2011 4056 3775 DCA I (0001
2012 4057 1376 TAD (CIF
2013 4060 3772 DCA I (0002
2014 4061 1273 TAD JMPRET
2015 4062 3771 DCA I (0003
2016 4063 1272 TAD XRETAD
2017 4064 3770 DCA I (0007
2018 4065 5652 JMP I FILCOR
2019
2020 4066 4536 /XRTF1, ENDTST
2021 4067 5767 JMP EXMT10
2022 4070 0000 NUMX, 0000
2023 4071 0000 SNUMX, 0
2024 4072 4030 XRETAD, RETADD
2025 4073 5407 JMPRET, JMP I 7
2026
2027 4167 4200
2028 4170 0007
2029 4171 0003
2030 4172 0002
2031 4173 1000
2032 4174 0000
2033 4175 0001
2034 4176 6202
2035 4177 6201
2036 4230
PAGE
/*****
/MEMORY EXTENSION TEST 10
/TEST DF + IF FROM SAVE FIELD AFTER PI
/USE RTF TO SET IB & DF REG AND GTF TO GET THE FLAGS
/DO FIELD COMBINATIONS 00,11,22, 433.
/TEST CDI TO CHANGE DATA & INST FIELDS
/
/THIS TEST OVERWRITES LOCATIONS 0,1, & 2 IN FIELD 0.
/
/
EXMT10, INTST
2047 4201 4510 JMS I XSTKS
2048 4202 4506 JMS I XTFLG /SET TTY FLAG
2049 4203 1377 TAD (ISZ 0
2050 4204 3001 DCA 1
2051 4205 1376 TAD (JMP I 0
2052 4206 3002 DCA 2
2053 4207 3275 DCA XTOR
2054 4210 1275 XSRTF, TAD XTOR
2055 4211 6005 RTF /MAKE DF XX + IF XX
2056 4212 5213 JMP ,+1 /INTERRUPT HERE
2057 4213 4533 CRERR /INTERRUPT WAS INHIBITED.

```

```

2058 4214 7300 CLA CLL
2059 4215 6004 GTF /GET THE FLAGS
2060 4216 0375 AND (0077
2061 4217 3276 DCA SFLDS
2062 4220 1276 TAD SFLDS
2063 4221 7041 CIA
2064 4222 1275 TAD XTOR /EXPECTED BITS
2065 4223 7650 SNA CLA
2066 4224 5231 JMP L101
2067 4225 1275 TAD XTOR
2068 4226 7421 MQL
2069 4227 1276 TAD SFLDS
2070 4230 4533 CRERR
2071 4231 1275 L101, TAD XTOR /AC,MQ=ACTUAL DF & IF,EXPECTED DF & IF.
2072 4232 1374 TAD (0011
2073 4233 3275 DCA XTOR
2074 4234 2107 ISZ STKS
2075 4235 5210 JMP XSRTF /DO THE REST DF XX + IF XX
2076 4236 4510 JMS I XSTKS
2077 4237 3275 DCA XTOR /CDI TEST
2078 4240 1275 XCDIL, TAD XTOR
2079 4241 0373 AND (0070
2080 4242 1372 TAD (CDI
2081 4243 3244 DCA ,+1
2082 4244 0000 ICDI, 0000 /CHANGE DATA AND INST FIELD TO XX
2083 4245 6001 ION
2084 4246 7000 NOP
2085 4247 5250 JMP ,+1 /INTERRUPT HERE
2086 4250 4533 CRERR /INTERRUPT WAS INHIBITED
2087 4251 7300 CLA CLL
2088 4252 6004 GTF /GET THE FLAGS
2089 4253 0375 AND (0077
2090 4254 3276 DCA SFLDS
2091 4255 1276 TAD SFLDS
2092 4256 7041 CIA
2093 4257 1275 TAD XTOR /EXPECTED BITS
2094 4260 7650 SNA CLA
2095 4261 5266 JMP L102
2096 4262 1275 TAD XTOR
2097 4263 7421 MQL
2098 4264 1276 TAD SFLDS
2099 4265 4533 CRERR /CDI FAILED TO SET DF & IF PROPERLY
2100 4266 1275 L102, TAD XTOR /AC,MQ=ACTUAL DF & IF,EXPECTED DF & IF
2101 4267 1374 TAD (0011
2102 4270 3275 DCA XTOR
2103 4271 2107 ISZ STKS
2104 4272 5240 JMP XCDIL /DO CDI TEST TO REMAINING FIELDS
2105 4273 4536 ENDTST
2106 4274 5771 JMP EXHT11
2107 4275 0000 XTOR, 0
2108 4276 0000 SFLDS, 0
2109 4371 4400
2110 4372 6203
2111 4373 0070
2112

```

2113 4374 0011
 2114 4375 0077
 2115 4376 5400
 2116 4377 2000
 4400

2117
 2118
 2119
 2120
 2121
 2122
 2123
 2124
 2125
 2126
 2127 4400 4535
 2128 4401 4510
 2129 4402 2107
 2130 4403 1377
 2131 4404 3211
 2132 4405 7001
 2133 4406 3324
 2134 4407 1376
 2135 4410 3010
 2136 4411 0000
 2137 4412 7300
 2138 4413 1211
 2139 4414 6005
 2140 4415 6002
 2141 4416 7300
 2142 4417 1324
 2143 4420 3410
 2144 4421 1375
 2145 4422 3410
 2146 4423 1374
 2147 4424 3410
 2148 4425 1373
 2149 4426 3410
 2150 4427 1372
 2151 4430 3410
 2152 4431 1371
 2153 4432 3410
 2154 4433 1370
 2155 4434 3410
 2156 4435 1367
 2157 4436 3410
 2158 4437 1366
 2159 4440 3410
 2160 4441 1365
 2161 4442 3246
 2162 4443 1111
 2163 4444 3001
 2164 4445 5646
 2165 4446 0000
 2166 4447 3325

PAGE
 /
 /*****
 /MEMORY EXTENSION TEST 11
 /TEST PROGRAM INTERRUPT IN ALL EXTENDED FIELDS
 /USE RTF, GTF, RDF AND RIF FOR CHECK
 /CHECK PC, AC, SF AND FLAGS AFTER PI
 /
 /THIS TEST OVERWRITES LOCATIONS 0 & 1 IN FIELD 0; ALSO
 / LOCATIONS 0 - 10 IN FIELDS 1,2, & 3.
 /
 EXMT11, INTST
 JMS I XSTKS
 ISZ STKS /FIELD # NOT TESTED
 TAD (0011
 DCA IFDF
 IAC
 DCA XDATA
 TAD (7777
 DCA 0010
 IPDF, 0000 /SET TO CURRENT FIELD UNDER TEST
 CLA CLL
 TAD IFDF
 RTF /SET FIELDS AND TURN ION
 IOF
 CLA CLL
 TAD XDATA
 DCA I 0010
 TAD (ISZ 00
 DCA I 0010
 TAD (JMP 5
 DCA I 0010
 TAD (ION
 /ION FOR THAT FIELD
 TAD (1000
 DCA I 0010 /TAD FOR THAT FIELD
 TAD (CDF 00
 DCA I 0010
 TAD (CIF 00
 DCA I 0010
 TAD (JMP I 10
 DCA I 0010
 TAD (FRET-1 /REMEMBER AUTO INDEX
 DCA I 0010
 TAD (0003
 DCA ADRS
 TAD JMPIR
 DCA 0001 /SET LOC 1 FOR RETURN AFTER PI
 JMP I ,+1 /GO TO THAT FIELD
 ADRS, 0000
 RET, DCA SDATA

2167 4450 1325
 2168 4451 7041
 2169 4452 1324
 2170 4453 7650
 2171 4454 5261
 2172 4455 1324
 2173 4456 7421
 2174 4457 1325
 2175 4460 4533
 2176
 2177 4461 1000
 2178 4462 7041
 2179 4463 1364
 2180 4464 7650
 2181 4465 5272
 2182 4466 1364
 2183 4467 7421
 2184 4470 1000
 2185 4471 4533
 2186
 2187 4472 6214
 2188 4473 6224
 2189 4474 7440
 2190 4475 4533
 2191
 2192 4476 6004
 2193 4477 0363
 2194 4500 3325
 2195 4501 1325
 2196 4502 7041
 2197 4503 1211
 2198 4504 7650
 2199 4505 5312
 2200 4506 1211
 2201 4507 7421
 2202 4510 1325
 2203 4511 4533
 2204
 2205 4512 2107
 2206 4513 7410
 2207 4514 5322
 2208 4515 7300
 2209 4516 1377
 2210 4517 1211
 2211 4520 3211
 2212 4521 5207
 2213 4522 4536
 2214 4523 5762
 2215 4524 0000
 2216 4525 0000
 2217 4526 5331
 2218 4527 4533
 2219 4530 7410
 2220 4531 4533
 2221 4532 5312

TAD SDATA
 CIA
 TAD XDATA
 SNA CLA
 JMP L111
 TAD XDATA
 MQL
 TAD SDATA
 CBERR /AC DATA FAILED DURING PI
 /AC,MQ=ACTUAL AC DATA,EXPECTED AC DATA
 L111, TAD 0000
 CIA
 TAD (0005
 SNA CLA
 JMP L112
 TAD (0005
 MQL
 TAD 0000
 CBERR /PC FAILED DURING PI
 /AC,MQ=ACTUAL PC,EXPECTED PC
 L112, RDF
 RIF
 SZA
 CBERR /I,F, SHOULD BE 0 AFTER A PI
 /AC=ACTUAL I,F.
 GTF
 AND (0077
 DCA SDATA
 TAD SDATA
 CIA
 TAD IFDF
 SNA CLA
 JMP CONTDG
 TAD IFDF
 MQL
 TAD SDATA
 CBERR /GTF,RTF, OR SF FAILED
 /AC,MQ=ACTUAL FIELDS,EXPECTED FIELDS
 CONTDG, ISZ STKS
 SKP
 JMP XSFIB
 CLA CLL
 TAD (0011
 TAD IFDF
 DCA IFDF /SET FOR NEXT FIELD
 JMP IFDF-2
 XSFIB, ENDTST
 JMP EXMT12
 XDATA, 0
 SDATA, 0
 FRET, JMP NOPI /ERROR RETURN FROM EXTENDED FIELD,
 /GOT PI BUT DIDN'T CHANGE FIELDS.
 NOPI, CBERR /NO PI
 JMP CONTDG

2222
2223 4533 0000
2224 4534 1361
2225 4535 3107
2226 4536 5733
2227
2228
2229
2230 4537 0000
2231 4540 7200
2232 4541 6940
2233 4542 6041
2234 4543 5342
2235 4544 7200
2236 4545 5737
2237 4561 7774
2238 4562 4600
2239 4563 0077
2240 4564 0005
2241 4565 0003
2242 4566 4525
2243 4567 5410
2244 4570 6202
2245 4571 6201
2246 4572 1000
2247 4573 6001
2248 4574 5005
2249 4575 2010
2250 4576 7777
2251 4577 0011
2252 4600

```

/
NSTKS, 0
      TAD (-4
      DCA STKS
      JMP I NSTKS
/
/SET TTY FLAG
/
TFLG, 0
      CLA          /SET OUTPUT FLAG ENABLE
      SPF
      TSF
      JMP ,=1
      CLA
      JMP I TFLG  /EXIT

```

PAGE

2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265 4600 4535
2266 4601 6041
2267 4602 4506
2268 4603 6211
2269 4604 1377
2270 4605 3776
2271 4606 1375
2272 4607 3774
2273 4610 1373
2274 4611 3772
2275 4612 1371

```

/
/*****
/MEMORY EXTENSION TEST 12
/TEST SF WITH AN RMF IOT, AN INTERRUPT IN FIELD 0 IS CREATED, AFTER
/WHICH, THE DF AND IB REGISTERS ARE SET TO FIELD 1.
/THE SF SHOULD CONTAIN FIELD 0, THE TEST
/THEN MAKES SURE THE IB IS RESET TO 0 BY ISSUING AN RMF,
/FOLLOWED BY A JMP I (0003, IF THE IB IS RESET TO 0 THE JMP GOES TO 0003 IN FIELD 0,
/IF THE IB FAILS TO BE RESET TO 0, THE JMP GOES TO 0003 IN FIELD 1.
/
/THIS TEST OVERWRITES LOCATIONS 0,1,2, & 3 IN FIELD 0; ALSO
/LOCATIONS 3,4,5, & 6 IN FIELD 1.
/
EXMT12, INTST
      TSF          /SEE IF FLAG IS SET.
      JMS I XTFLG /SET IT
      CDF 10      /DF=FIELD 1
      TAD (CDF 00 /SETUP RETURN FROM FIELD 1
      DCA I (0003 /IN CASE INST FAILS.
      TAD (CIF 00
      DCA I (0004
      TAD (JMP I 6
      DCA I (0005
      TAD (KKRTN

```

2276 4613 3770
2277 4614 6201
2278 4615 1367
2279 4616 3776
2280 4617 1366
2281 4620 3001
2282 4621 1365
2283 4622 3002
2284
2285
2286
2287 4623 6001
2288 4624 7000
2289 4625 4533
2290 4626 5235
2291
2292
2293
2294 4627 7200
2295 4630 6211
2296 4631 6212
2297 4632 6244
2298 4633 5776
2299
2300 4634 4533
2301 4635 4536
2302
2303
2304
2305
2306
2307
2308
2309 4636 4535
2310 4637 6042
2311 4640 6241
2312 4641 6244
2313 4642 1660
2314
2315 4643 6201
2316 4644 7440
2317 4645 4533
2318 4646 6241
2319 4647 7300
2320 4650 6005
2321 4651 6002
2322 4652 1660
2323 4653 6201
2324 4654 7440
2325 4655 4533
2326 4656 4536
2327 4657 5764
2328 4660 4661
2329 4661 0000
2330

```

      DCA I (0006
      CDF 00      /DF=0
      TAD (JMP I KFLD0
      DCA I (0003
      TAD (JMP I 2
      DCA 1
      TAD (KRTN
      DCA 2
/
/BEGIN TEST
/
      ION          /ENABLE PI
      NOP
      CAG,        CSERR      /ERROR NO PI
      JMP RTRN
/
/RETURN HERE AFTER PI
/
KRTN,  CLA
      CDF 10      /DF=FIELD1
      CIF 10      /IB=FIELD1
      RMF         /IB SHOULD=FIELD0
      JMP I (0003 /IF SHOULD=FIELD0
/
KKRTN, CSERR      /JMP WENT TO FIELD 1.
RTRN,  ENDTST
/
/*****
/MEMORY EXTENSION TEST 13
/CHECK THAT RMF AND RTF INSTRUCTIONS ZERO MOST SIGNIFICANT BIT
/OFF DATA FIELD REGISTER.
/
EXMT13, INTST
      TCF          /CLEAR TTY OUTPUT FLAG ENABLE.
      CDF 40
      RMF
      TAD I KTEST /IF MOST SIGN BIT OF DF REG IS NOT ZERO
                        /THEN THE INDIRECT WILL NOT OCCUR
      CDF 00
      SZA
      CSERR      /RMF FAILED TO ZERO MOST SIGN. BIT OF DF REG.
      CDF 40
      CLA CLL
      RTF
      IOF
      TAD I KTEST
      CDF 00
      SZA
      CSERR      /RTF FAILED TO ZERO MOST SIGN. BIT OF DF REG
      ENDTST
      JMP EXMT14
      KTEST, KTEST1
      KTEST1, 0

```

2331
2332 4764 5000
2333 4765 4627
2334 4766 5402
2335 4767 5514
2336 4770 0006
2337 4771 4634
2338 4772 0005
2339 4773 5406
2340 4774 0004
2341 4775 6202
2342 4776 0003
2343 4777 6201
5000

```

PAGE
/*****
/MEMORY EXTENSION TEST 14 (UNUSED IOT TEST)
/VERIFY THAT ALL UNUSED IOTS HAVE NO EFFECT ON SYSTEM
/
EXMT14, FIXIL          /FIXUP INTERRUPT LINKAGE
          INTST
          CAF
          ION          /UNUSED IOT'S SHOULD NOT CAUSE INTERRUPTS
          TAD KIOT
          DCA TSTIOT   /INITIALIZE TEST IOT
          TAD (USEDIOT
          DCA CUIPTR   /INITIALIZE POINTER INTO USED IOT TABLE
IOTLP,   TAD TSTIOT
          AND (0770
          CIA
          TAD I CUIPTR  /IS TEST IOT DEVICE CODE A SYSTEM DEVICE CODE ?
          SZA CLA
          JMP EXIOT    /NO-EXECUTE IT
          ISZ CUIPTR   /YES-SKIP THIS DEVICE CODE
          TAD TSTIOT
          TAD (0010    /UPDATE TO NEXT TEST IOT
          DCA TSTIOT
          TAD TSTIOT
          AND (1000
          SNA CLA
          JMP IOTLP
          JMP EXIOTLP  /ALL UNUSED IOTS TESTED
          TAD (5252    /LOAD AC WITH ARBITRARY DATA WORD
EXIOT,   TSTIOT, 0000 /EXECUTE UNUSED IOT
          SKP
          JMP SKPERR  /IOT CAUSED A SKIP
          DCA ACAIOT
          TAD ACAIOT
          TAD (-5252
          SZA CLA
          JMP ACCERR  /IOT DESTROYED AC CONTENTS
          ISZ TSTIOT  /UPDATE TO NEXT IOT WITHIN SAME DEVICE CODE
          TAD TSTIOT
          AND (0007
          SZA CLA
          JMP EXIOT
  
```

2344
2345
2346
2347
2348 5000 4520
2349 5001 4535
2350 5002 6007
2351 5003 6001
2352 5004 1317
2353 5005 3230
2354 5006 1377
2355 5007 3321
2356 5010 1230
2357 5011 0376
2358 5012 7041
2359 5013 1721
2360 5014 7640
2361 5015 5227
2362 5016 2321
2363 5017 1230
2364 5020 1375
2365 5021 3730
2366 5022 1230
2367 5023 0374
2368 5024 7650
2369 5025 5210
2370 5026 5261
2371 5027 1373
2372 5030 0000
2373 5031 7410
2374 5032 5246
2375 5033 3320
2376 5034 1320
2377 5035 1372
2378 5036 7640
2379 5037 5254
2380 5040 2230
2381 5041 1230
2382 5042 0371
2383 5043 7640
2384 5044 5227

2385 5045 5210
2386 5046 7300
2387 5047 1320
2388 5050 7421
2389 5051 1230
2390 5052 4533
2391
2392 5053 5240
2393 5054 1320
2394 5055 7421
2395 5056 1230
2396 5057 4533
2397
2398 5060 5240
2399 5061 4536
2400 5062 3075
2401 5063 5770
2402
2403
2404 5064 0010
2405 5065 0030
2406 5066 0040
2407 5067 0070
2408 5070 0130
2409 5071 0200
2410 5072 0210
2411 5073 0220
2412 5074 0230
2413 5075 0240
2414 5076 0250
2415 5077 0260
2416 5100 0270
2417 5101 0300
2418 5102 0310
2419 5103 0320
2420 5104 0330
2421 5105 0500
2422 5106 0660
2423 5107 0750
2424 5110 0000
2425 5111 0000
2426 5112 0000
2427 5113 0000
2428 5114 0000
2429 5115 0000
2430 5116 0000
2431
2432 5117 6010
2433 5120 0000
2434 5121 0000
2435
2436
2437 5170 5400
2438 5171 0007
2439 5172 2526

```

          JMP IOTLP    /DEVICE CODE FINISHED
          SKPERR, CLA CLL
          TAD ACAIOT
          MQL
          TAD TSTIOT
          CBERR
          /IOT CAUSED INST SKIP
          /AC,MQ=BAD IOT,AC DATA
          JMP NXTIOT
          ACCERR, TAD ACAIOT
          MQL
          TAD TSTIOT
          CBERR
          /AC,MQ=BAD IOT, AC DATA AFTER IOT EXECUTION.
          /AC DATA BEFORE EXECUTION=5252
          JMP NXTIOT
          EXIOTL, ENDTST
          DCA TSTNO
          JMP BEGRTC
          /CLEAR TEST NUMBER
          USEDIOT, 0010 /ORDERED LIST OF USED DEVICE CODES
          0030
          0040
          0070
          0130
          0200
          0210
          0220
          0230
          0240
          0250
          0260
          0270
          0300
          0310
          0320
          0330
          0500
          0660
          0750
          0000
          0000
          0000
          0000
          0000
          0000
          0000
          0000
          KIOT, 6010
          ACAIOT, 0
          CUIPTR, 0000
  
```

2440 5173 5252
2441 5174 1400
2442 5175 0810
2443 5176 0770
2444 5177 5064
5200

PAGE
/ROUTINE TO PRINT "MEMORY EXTENSION TESTING" MESSAGE

2445
2446
2447 5200 0000
2448 5201 4521
2449 5202 4522
2450 5203 5210
2451 5204 4522
2452 5205 2713
2453 5206 4521
2454 5207 5600
2455
2456 5210 1505
5211 1517
5212 2231
5213 4005
5214 3024
5215 5640
5216 0000

PMEMES, 0
C0CRLF
C0PRNT
MEMES
C0PRNT
TESTMS
C0CRLF
JMP I PMEMES
MEMES, TEXT "MEMORY EXT. "

2457
2458
2459

/ROUTINE TO RESTORE INTERRUPT LINKAGE TO NORMAL

2460 5217 0000
2461 5220 1377
2462 5221 3001
2463 5222 1227
2464 5223 3002
2465 5224 1376
2466 5225 3003
2467 5226 5617
2468 5227 5403
2469
2470 5230 0413
5231 2624
5232 0255
5233 0140
5234 0601
5235 1114
5236 0504
5237 5440
5240 0611
5241 0514
5242 0440
5243 6040
5244 4000

FIXLKG, 0
TAD (RMF
DCA 1
TAD KJMP3
DCA 2
TAD (SKPCHN
DCA 3
JMP I FIXLKG
KJMP3, JMP I 3
ERRMES, TEXT "DKVTS=A FAILED, FIELD 0 "

2471
2472
2473

2474
2475 5376 6200
2476 5377 6244
5400

*5400
/*****
/REAL TIME CLOCK TEST
/*****

2477
2478
2479
2480
2481
2482

/*****
/REAL TIME CLOCK TEST 1 - CHECKS THAT CLOCK FLAG WILL SET AND
/ THAT CAF WILL CLEAR IT, THE TEST IS
/ CHECKED NOT TO INTERRUPT.
/*****

2483
2484
2485
2486
2487
2488

BEGRTC, CLA CLL
IOF
TAD PSR /EXECUTE REAL TIME CLOCK TEST?
AND (0002
SZA CLA
JMP BEGBST /NO
FIXIL /FIXUP INTERRUPT LINKAGE
JMS PRTCMS /PRINT REAL TIME CLOCK TESTING MESSAGE
RTCT1, INTST
TAD (RTCI1 /PATCH INTERRUPT LINKAGE TO RETURN TO THIS
DCA 3 / TEST ON INTERRUPT.
CAF
ION
CLSKWT /WAIT FOR THE CLOCK FLAG TO SET
CBERR /CLOCK FLAG FAILED TO SET
CAF /CLEAR THE CLOCK FLAG
CLSK /SKIP ON THE CLOCK FLAG
SKP CLA
CBERR /CAF FAILED TO CLEAR CLOCK FLAG OR CLSK SKIPPED
RTCT1D, ENDTST
JMP RTCT2

2489 5400 7300
2490 5401 6002
2491 5402 1020
2492 5403 0377
2493 5404 7640
2494 5405 5776
2495 5406 4520
2496 5407 4242
2497 5410 4535
2498 5411 1375
2499 5412 3003
2500 5413 6007
2501 5414 6001
2502 5415 4541
2503 5416 4533
2504 5417 6007
2505 5420 6137
2506 5421 7610
2507 5422 4533
2508 5423 4536
2509 5424 5263
2510
2511 5425 4533
2512 5426 5223
2513
2514

/ROUTINE TO WAIT FOR THE REAL TIME CLOCK FLAG

2515
2516 5427 0000
2517 5430 3241
2518 5431 6137
2519 5432 7410
2520 5433 5237
2521 5434 2241
2522 5435 5231
2523 5436 7410
2524 5437 2227
2525 5440 5627
2526 5441 0000
2527

WTCLSK, 0
DCA RCNT
CLSK /WAS THE CLOCK FLAG SET
SKP /NO
JMP ,+4 /YES
ISZ RCNT
JMP ,=4
SKP
ISZ WTCLSK
JMP I WTCLSK
RCNT, 0

/ROUTINE TO PRINT "REAL TIME CLOCK TESTING" MESSAGE

2528
2529
2530 5442 0000
2531 5443 4521
2532 5444 4522
2533 5445 4522
2534 5446 4522
2535 5447 2213
2536 5450 4521
2537 5451 5642
2538
2539 5452 2205
5453 0114
5454 4024
5455 1115
5456 0540
5457 0314
5460 1703
5461 1340
5462 4000

PRICMS, 0
C8CRLF
C8PRNT
RTCMS
C8PRNT
TESTMS
C8CRLF
JMP I PRICMS

RTCMS, TEXT "REAL TIME CLOCK "

2540
2541
2542
2543
2544
2545
2546
2547
2548 5463 4535
2549 5464 1374
2550 5465 3003
2551 5466 6007
2552 5467 6001
2553 5470 4541
2554 5471 4533
2555 5472 6136
2556 5473 7610
2557 5474 4533
2558 5475 6137
2559 5476 7610
2560 5477 4533
2561 5500 4536
2562 5501 5304
2563
2564 5502 4533
2565 5503 5300
2566
2567
2568
2569
2570
2571
2572
2573 5504 4535
2574 5505 1373

/REAL TIME CLOCK TEST 2 - CHECKS THAT CLOCK FLAG WILL SET AND THAT
IT CAN BE CLEARED BY CLCL. THE TEST IS
CHECKED NOT TO INTERRUPT.

RTCT2, INTST
TAD (RTC12
DCA 3
CAF
ION
CLSKWT
C8ERR
CLCL
SKP CLA
C8ERR
CLSK
SKP CLA
C8ERR
RTCT2D, ENDTST
JMP RTCT3
RTC12, C8ERR
JMP RTCT2D

/REAL TIME CLOCK TEST 3 - CHECK THAT CLOCK INT. ENABLE CAN BE SET
AND CLEARED BY DATA BIT 11 AND CLLE USING
THE CLOCK FLAG TO INTERRUPT ON.

RTCT3, INTST
TAD (ISZ RTCIF /FIX UP INT. LINKAGE TO SET REAL TIME CLOCK

2575 5506 3002
2576 5507 1372
2577 5510 3003
2578 5511 3113
2579 5512 6007
2580 5513 6001
2581 5514 4541
2582 5515 4533
2583 5516 6136
2584 5517 6137
2585 5520 7610
2586 5521 4533
2587 5522 7301
2588 5523 6135
2589 5524 7610
2590 5525 4533
2591 5526 1113
2592 5527 7640
2593 5530 4533
2594 5531 4541
2595 5532 4533
2596 5533 1113
2597 5534 7650
2598 5535 4533
2599
2600 5536 3113
2601 5537 6135
2602 5540 6001
2603 5541 7300
2604 5542 1113
2605 5543 7640
2606 5544 4533
2607 5545 6136
2608 5546 6137
2609 5547 7610
2610 5550 4533
2611 5551 4536
2612 5552 5771
2613
2614
2615
2616 5571 5600
2617 5572 5400
2618 5573 2113
2619 5574 5502
2620 5575 5425
2621 5576 6000
2622 5577 0302
2623 5600

DCA 2
TAD (JMP I 0
DCA 3
DCA RTCIF
CAF
ION
CLSKWT
C8ERR
CLCL
CLSK
SKP CLA
C8ERR
CLA CLL IAC
CLLE
SKP CLA
C8ERR
TAD RTCIF
SZA CLA
C8ERR
CLSKWT
C8ERR
TAD RTCIF
SNA CLA
C8ERR
DCA RTCIF
CLLE
ION
CLA CLL
TAD RTCIF
SZA CLA
C8ERR
CLCL
CLSK
SKP CLA
C8ERR
ENDTST
JMP RTCT4

PAGE

/REAL TIME CLOCK TEST 4 - CHECK THAT CLOCK INT. ENABLE CAN BE SET
AND THAT CAF WILL CLEAR IT USING THE CLOCK
FLAG TO INTERRUPT ON.

2629 5600 4535
 2630 5601 3113
 2631 5602 6407
 2632 5603 6901
 2633 5604 4541
 2634 5605 4533
 2635 5606 7301
 2636 5607 6135
 2637 5610 7300
 2638 5611 1113
 2639 5612 7650
 2640 5613 4533
 2641
 2642 5614 3113
 2643 5615 6907
 2644 5616 6901
 2645 5617 4541
 2646 5620 4533
 2647 5621 1113
 2648 5622 7640
 2649 5623 4533
 2650 5624 6136
 2651 5625 4536
 2652 5626 5227
 2653
 2654
 2655
 2656
 2657
 2658
 2659 5627 4535
 2660 5630 6136
 2661 5631 6901
 2662 5632 7344
 2663 5633 6135
 2664 5634 1377
 2665 5635 7440
 2666 5636 4533
 2667 5637 7240
 2668 5640 6136
 2669 5641 7001
 2670 5642 7440
 2671 5643 4533
 2672 5644 7240
 2673 5645 6137
 2674 5646 7000
 2675 5647 7001
 2676 5650 7440
 2677 5651 4533
 2678 5652 7200
 2679 5653 4536
 2680 5654 5255
 2681
 2682
 2683

RTCT4, INTST
 DCA RTCIF
 CAF /CLEAR ALL FLAGS
 ION
 CLSKWT /WAIT FOR THE CLOCK FLAG TO SET
 CBERR /CLOCK FLAG FAILED TO SET
 CIA CLL IAC
 CLLE /SET INTERRUPT ENABLE TO A ONE
 CLA CLL /SHOULD INTERRUPT HERE
 TAD RTCIF
 SNA CLA
 CBERR /PROGRAM FAILED TO INT, WITH CLOCK FLAG SET
 / AND CLOCK INT, ENABLED,
 DCA RTCIF /CLEAR CLOCK FLAG
 CAF /CLEAR ALL FLAGS AND DISABLE CLK INT,
 ION
 CLSKWT /WAIT FOR FLAG
 CBERR /CLOCK FLAG FAILED TO SET
 TAD RTCIF /GET THE CLOCK INT, FLAG
 SZA CLA /DID IT INTERRUPT?
 CBERR /CAF FAILED TO CLEAR CLOCK INT, ENABLE
 CLCL /CLEAR THE CLOCK FLAG
 ENDTST
 JMP RTCT5

 /REAL TIME CLOCK TEST 5 - CHECK THAT THE THREE REAL TIME CLOCK IOTS
 / DON'T AFFECT THE AC.

RTCT5, INTST
 CLCL
 ION
 CLA CLL CMA RAL /-2
 CLLE /CLEAR CLOCK INT, ENABLE
 TAD (2
 SZA
 CBERR /CLLE CHANGED THE AC
 CLA CMA
 CLCL /CLEAR CLOCK FLAG
 IAC
 SZA
 CBERR /CLCL CHANGED THE AC
 CLA CMA
 CLSK /SKIP ON CLOCK FLAG
 NOP
 IAC
 SZA
 CBERR /CLSK CHANGED THE AC
 CLA
 ENDTST
 JMP RTCT6

2684
 2685
 2686
 2687 5655 4535
 2688 5656 4520
 2689 5657 1376
 2690 5660 3003
 2691 5661 6136
 2692 5662 3312
 2693 5663 6901
 2694 5664 4541
 2695 5665 4533
 2696 5666 6136
 2697 5667 7301
 2698 5670 6135
 2699 5671 2312
 2700 5672 5271
 2701 5673 4533
 2702 5674 1312
 2703 5675 7041
 2704 5676 1313
 2705 5677 7500
 2706 5700 5315
 2707 5701 1314
 2708 5702 7510
 2709 5703 5315
 2710 5704 7300
 2711 5705 6135
 2712 5706 4536
 2713 5707 3075
 2714 5710 4520
 2715 5711 5775
 2716
 2717 5712 0000
 2718 5713 1200
 2719 5714 0010
 2720
 2721 5715 7300
 2722 5716 1312
 2723 5717 4533
 2724
 2725 5720 7300
 2726 5721 1313
 2727 5722 1314
 2728 5723 7421
 2729 5724 1313
 2730 5725 4533
 2731
 2732 5726 5304
 2733
 2734
 2735

 /REAL TIME CLOCK TEST 6 - CHECK CLOCK TIMING

 RTCT6, INTST
 FIXIL
 TAD (CLKIR /FIX UP INT, LINKAGE TO RETURN TO THIS TEST ON INT,
 DCA 3
 CLCL /CLEAR CLOCK FLAG
 DCA CLKCT /CLEAR CLOCK COUNTER
 ION
 CLSKWT /WAIT FOR CLOCK FLAG TO SET
 CBERR /CLOCK FLAG FAILED TO SET
 CLCL /CLEAR FLAG
 CIA CLL IAC
 CLLE /ENABLE CLOCK INTERRUPT
 ISZ CLKCT
 JMP ,-1
 CBERR /CLOCK FAILED TO TRIGGER INTERRUPT
 CLKIR, TAD CLKCT
 CIA
 TAD CLCON
 SNA
 JMP CLKERR
 TAD CDELTA
 SPA
 JMP CLKERR
 CONTT, CLA CLL
 CLLE /DISABLE CLOCK INTERRUPT
 ENDTST
 DCA TSTNO /CLEAR TEST NUMBER
 FIXIL /FIX UP INTERRUPT LINKAGE
 JMP BEGBST
 CLKCT, 0
 CLCON, 1200 /LOWER LIMIT ON EXPECTED CLOCK COUNT
 CDELTA, 10 /ALLOWED RANGE FOR CLOCK COUNT
 CLKERR, CLA CLL
 TAD CLKCT
 CBERR /CLOCK TIMING ERROR - AC = ACTUAL COUNT
 / CONTINUE FOR EXPECTED COUNT,
 CLA CLL
 TAD CLCON
 TAD CDELTA
 MGL
 TAD CLCON
 CBERR /EXPECTED TIMING - AC, MG = LOWER LIMIT, UPPER LIMIT
 / FOR COUNT,
 JMP CONTT


```

2736
2737 5775 6000
2738 5776 5674
2739 5777 0002
      6000
2740
2741 /*****
2742 /BAUD RATE SWITCH TEST
2743 /*****
2744
2745 /THIS TEST READS BAUD RATE SWITCH AND DISPLAYS SETTING.
2746 /OPERATOR CAN CHANGE SWITCH SETTING &PRESS ANY KEY (EXCEPT RETURN)
2747 /      TO DISPLAY NEW SETTING.
2748 /OPERATOR EXITS TEST BY PRESSING THE RETURN KEY.
2749
2750 6000 7300 BEGBST, CLA CLL
2751 6001 6002 IOF
2752 6002 1022 TAD HCW2 /RUNNING UNDER APT CONTROL?
2753 6003 7719 SPA CLA /YES-DON'T EXECUTE THIS TEST-CONTINUE
2754 6004 5225 JMP DOSLU /WITH SLU TESTING.
2755 /NO-IS BAUD SWITCH TEST REQUESTED?
2756 6005 1020 TAD PSR /CHECK PSR BIT 11)
2757 6006 0377 AND (0001
2758 6007 7650 SNA CLA /NO-CONTINUE WITH SLU TESTING
2759 6010 5225 JMP DOSLU /YES-PRINT BAUD RATE SWITCH TESTING MESSAGE
2760 6011 4227 JMS PBRMS
2761 6012 7300 BSTLP, CLA CLL
2762 6013 7604 LAS /GET CURRENT SWITCH SETTING
2763 6014 0376 AND (0017 /MASK OFF UNWANTED BITS
2764 6015 1375 TAD (BASEP /INDEX INTO BAUD RATE TABLE
2765 6016 3275 DCA TBLPTR
2766 6017 4521 C8CRLF /PRINT A CARRIAGE RETURN & LINE FEED
2767 6020 4276 JMS PRTRS /PRINT DECIMAL SETTING
2768 6021 4247 JMS LISNR /LISTEN FOR A CHARACTER
2769 6022 1374 TAD (-015 /IS IT A CARRIAGE RETURN?
2770 6023 7640 SZA CLA /NO-DISPLAY NEW SETTING
2771 6024 5212 JMP BSTLP /YES-EXIT TO SLU TESTING IN FIELD 1
2772 6025 6213 DOSLU, CDI 10
2773 6026 5773 JMP CIREST
2774
2775 /ROUTINE TO PRINT "BAUD RATE SWITCH TESTING" MESSAGE
2776
2777 6027 0000 PBRMS, 0
2778 6030 4521 C8CRLF
2779 6031 4522 C8PRNT
2780 6032 6036 BRMS
2781 6033 4522 C8PRNT
2782 6034 2213 TESTMS
2783 6035 5027 JMP I PBRMS
2784
2785 BRMS, TEXT "BAUD RATE SWITCH "
      6037 2504
      6040 4022
      6041 0124
      6042 0540

```

```

6043 2327
6044 1124
6045 0310
6046 4000
2786
2787 /ROUTINE TO LISTEN FOR A KEYBOARD CHARACTER
2788
2789 6047 0000 LISNR, 0
2790 6050 6031 KSF /WAIT FOR KEYBOARD FLAG
2791 6051 5250 JMP =1
2792 6052 6036 KRB /READ THE CHARACTER TYPED
2793 6053 0372 AND (177 /MASK TO 7 BIT ASCII
2794 6054 5647 JMP I LISNR /RETURN WITH CHAR IN AC
2795
2796 /BAUD RATE SWITCH TABLE
2797
2798 6055 0050 BASEP, 0050
2799 6056 0075 0075
2800 6057 0110 0110
2801 6060 0134 0134
2802 6061 0150 0150
2803 6062 0300 0300
2804 6063 0600 0600
2805 6064 1200 1200
2806 6065 0000 SD1, 0000 /1000
2807 6066 2000 2000
2808 6067 2400 2400
2809 6070 3600 3600
2810 6071 0000 SD2, 0000 /4000
2811 6072 7200 7200
2812 6073 0000 SD3, 0000 /9600
2813 6074 0000 SD4, 0000 /19200
2814
2815 6075 0000 TBLPTR, 0000
2816
2817 /ROUTINE TO PRINT BAUD RATE SETTING=DECIMAL NUMBER
2818
2819 6076 0000 PRTRS, 0
2820 6077 1675 TAD I TBLPTR /GET PRINT SETTING FROM TABLE
2821 6100 7450 SNA
2822 6101 5304 JMP DECPRT /SETTING=0 - INDICATES DECIMAL #
2823 6102 4523 C8PRT4 /SETTING CONTAINS NO DECIMAL DIGITS
2824 / PRINT 4 OCTAL DIGITS
2825
2826 6103 5676 JMP I PRTRS
2827 6104 1275 DECPRT, TAD TBLPTR /SETTING CONTAINS A DECIMAL DIGIT
2828 6105 7041 CIA /DETERMINE WHICH SETTING SHOULD BE PRINTED
2829 6106 1371 TAD (SD1
2830 6107 7450 SNA
2831 6110 5342 JMP PSS1 /PRINT 1000
2832 6111 1378 TAD (4
2833 6112 7450 SNA
2834 6113 5334 JMP PSS2 /PRINT 4000
2835 6114 1367 TAD (2
2836 6115 7650 SNA CLA
2836 6116 5326 JMP PSS3 /PRINT 9600

```

```

/VT78 CPU DIAGNOSTIC PAL10 V142A 2-AUG-77 7:30 PAGE 6-2 SEQ 0071
2837 6117 4350 PSS4, JMS PRDD /PRINT 192
2838 6120 0261 #261
2839 6121 0271 #271
2840 6122 0262 #262
2841 6123 0000 #000
2842 6124 4530 PRNT2 /PRINT TRAILING 2 ZEROS
2843 6125 5676 JMP I PRTBRS
2844 6126 4350 PSS3, JMS PRDD /PRINT 96
2845 6127 0271 #271
2846 6130 0266 #266
2847 6131 0000 #000
2848 6132 4530 PRNT2 /PRINT TRAILING 2 ZEROS
2849 6133 5676 JMP I PRTBRS
2850 6134 4350 PSS2, JMS PRDD /PRINT 48
2851 6135 0264 #264
2852 6136 0270 #270
2853 6137 0000 #000
2854 6140 4530 PRNT2 /PRINT TRAILING 2 ZEROS
2855 6141 5676 JMP I PRTBRS
2856 6142 4350 PSS1, JMS PRDD /PRINT 18
2857 6143 0261 #261
2858 6144 0270 #270
2859 6145 0000 #000
2860 6146 4530 PRNT2 /PRINT TRAILING 2 ZEROS
2861 6147 5676 JMP I PRTBRS
2862
2863 6150 0000 PRDD, 0 /PRINT ASCII DIGITS FOLLOWING CALL
2864 6151 1750 TAD I PRDD
2865 6152 7450 SNA
2866 6153 5750 JMP I PRDD /DIGIT#0 = RETURN
2867 6154 4527 TYPE
2868 6155 2350 ISZ PRDD
2869 6156 5351 JMP PRDD+1
2870
2871
2872

```

```

/VT78 CPU DIAGNOSTIC PAL10 V142A 2-AUG-77 7:30 PAGE 7 SEQ 0072
2873
2874
2875 6167 0002
2876 6170 0004
2877 6171 6065
2878 6172 6177
2879 6173 0200
2880 6174 7763
2881 6175 6055
2882 6176 0017
2883 6177 0001
2884 6200
2885 /*****
2886 /FIELD 0 INTERRUPT SKIP CHAIN
2887 /*****
2888
2889 6200 3276 SKPCHN, DCA SAVAC /GET HERE VIA INTERRUPT
2890 /SAVE AC
2891 6201 4516 CHKKSF /CHECK FOR CONSOLE PACKAGE INT.
2892 6202 5206 JMP RFINT /SET-BUT IGNORE
2893 6203 7000 NOP /NOT SET
2894 6204 7000 NOP
2895 6205 4533 UEI, C9ERR /UNEXPECTED INTERRUPT
2896 6206 1276 RFINT, TAD SAVAC
2897 6207 5400 JMP I 0 /CONTINUE WITH DIAGNOSTIC
2898
2899
2900 /DISPLAY ROUTINE FOR UNEXPECTED INTERRUPT IN FIELD 0.
2901
2902 6210 4521 SERUEI, C9CRLF
2903 6211 4522 C9PRNT
2904 6212 6311 UEIMES
2905 6213 4521 C9CRLF
2906 6214 4777 JMS MTN /DISPLAY TEST NUMBER
2907 6215 1000 TAD 0
2908 6216 3776 DCA PCSAVE
2909 6217 4775 JMS MPC /DISPLAY PC
2910 6220 1276 TAD SAVAC
2911 6221 3774 DCA ACSAVE
2912 6222 4773 JMS MAC /DISPLAY AC
2913 6223 4772 JMS MFL /DISPLAY FLAGS
2914 6224 4521 C9CRLF
2915 6225 4522 C9PRNT
2916 6226 6331 F9MES
2917 6227 1371 TAD CDI
2918 6228 3274 DCA VCDI
2919 6231 4233 JMS CKFLG /DISPLAY FLAGS SET
2920 6232 5770 JMP IEH
2921
2922 6233 0000 CKFLG, 0
2923 6234 6041 TSF
2924 6235 5240 JMP UL1
2925 6236 4522 C9PRNT /SLU #1 XMIT FLAG SET
2926 6237 6337 MXHT1

```

/VT78 CPU DIAGNOSTIC			PAL10	V142A	2-AUG-77	7:30	PAGE 7-1	SEQ 0073
2927	6240	6311	UL1,	TSF1				
2928	6241	6244		JMP UL2				
2929	6242	4522		CSPRNT	/SLU #2 XMIT FLAG SET			
2930	6243	6342		MXMT2				
2931	6244	6331	UL2,	TSF2				
2932	6245	5250		JMP UL3				
2933	6246	4522		CSPRNT	/SLU #3 XMIT FLAG SET			
2934	6247	6345		MXMT3				
2935	6250	2040	UL3,	ISZ KSF2				
2936	6251	5254		JMP UL4				
2937	6252	4522		CSPRNT	/SLU #1 REC FLAG SET			
2938	6253	6350		MREC1				
2939	6254	6301	UL4,	KSF1				
2940	6255	5260		JMP UL5				
2941	6256	4522		CSPRNT	/SLU #2 REC FLAG SET			
2942	6257	6353		MREC2				
2943	6260	6321	UL5,	KSF2				
2944	6261	5264		JMP UL6				
2945	6262	4522		CSPRNT	/SLU #3 REC FLAG SET			
2946	6263	6356		MREC3				
2947	6264	6661	UL6,	PSKF				
2948	6265	5270		JMP UL7				
2949	6266	4522		CSPRNT	/LA180 PRINTER FLAG SET			
2950	6267	6361		MLAP				
2951	6270	6500	UL7,	LOGK				
2952	6271	5274		JMP VCDI				
2953	6272	4522		CSPRNT	/LQP PRINT FLAG SET			
2954	6273	6364		MLQP				
2955	6274	6203	VCDI,	CDI 00				
2956	6275	5633		JMP I CKFLG				
2957	6276	0000	SAVAC,	0				
2958	6277	1501	MMID,	TEXT	"MAINDEC=00-DKVTB-A"			
	6300	1116						
	6301	0405						
	6302	0355						
	6303	6070						
	6304	5504						
	6305	1326						
	6306	2402						
	6307	5501						
	6310	0000						
2959	6311	2516	UEIMES, TEXT		"UNEXPECTED INTERRUPT - FIELD 0"			
	6312	0530						
	6313	2005						
	6314	0324						
	6315	0504						
	6316	4011						
	6317	1624						
	6320	0522						
	6321	2225						
	6322	2024						
	6323	4055						
	6324	4006						
	6325	1105						
	6326	1404						

/VT78 CPU DIAGNOSTIC			PAL10	V142A	2-AUG-77	7:30	PAGE 7-2	SEQ 0074
	6327	4060						
	6330	0000						
2960	6331	0614	FSNES, TEXT		"FLAGE SET?"			
	6332	0107						
	6333	2340						
	6334	2305						
	6335	2472						
	6336	0000						
2961	6337	4040	MXMT1, TEXT		" X1"			
	6340	3061						
	6341	0000						
2962	6342	4040	MXMT2, TEXT		" X2"			
	6343	3062						
	6344	0000						
2963	6345	4040	MXMT3, TEXT		" X3"			
	6346	3063						
	6347	0000						
2964	6350	4040	MREC1, TEXT		" R1"			
	6351	2261						
	6352	0000						
2965	6353	4040	MREC2, TEXT		" R2"			
	6354	2262						
	6355	0000						
2966	6356	4040	MREC3, TEXT		" R3"			
	6357	2263						
	6360	0000						
2967	6361	4040	MLAP, TEXT		" LA"			
	6362	1401						
	6363	0000						
2968	6364	4040	MLQP, TEXT		" LQ"			
	6365	1421						
	6366	0000						

```

2969
2970 6370 6646
2971 6371 6203
2972 6372 6472
2973 6373 6456
2974 6374 6675
2975 6375 6532
2976 6376 6674
2977 6377 6524
        6400
2978 *6400
2979 /*****
2980 /ERROR DISPLAY FORMAT ROUTINES
2981 /*****
2982
2983 /NORMAL DISPLAY = TN,PC,AC,MQ,FLAGS
2984
2985 6400 4314 NORMDIS,JMS MESHDR /PRINT ERROR HEADER & PC
2986 6401 4256 JMS MAC /PRINT AC
2987 6402 4264 JMS MMQ /PRINT MQ
2988 6403 4272 JMS MFL /PRINT FLAGS
2989 6404 4521 C8CRLF
2990 6405 5777* JMP IEH
2991
2992 /COMBINED CPU MICROINST SKIP FAILURE DISPLAY
2993
2994 6406 4314 CPUSDIS,JMS MESHDR /PRINT ERROR HEADER & PC
2995 6407 4300 JMS MINS /PRINT INSTRUCTION
2996 6410 4521 C8CRLF
2997 6411 1067 TAD ACDATA /PRINT AC CONTENTS BEFORE INST EXECUTION
2998 6412 3776* DCA ACSAVE
2999 6413 4256 JMS MAC
3000 6414 1072 TAD MQDATA /PRINT MQ CONTENTS BEFORE INST EXECUTION
3001 6415 3775* DCA MQSAVE
3002 6416 4264 JMS MMQ
3003 6417 4306 JMS MLK /PRINT LINK CONTENTS BEFORE INST EXECUTION
3004 6420 4521 C8CRLF
3005 6421 5777* JMP IEH
3006
3007 /COMBINED CPU MICROINST DATA ERROR DISPLAY
3008
3009 6422 4314 CPUDIS, JMS MESHDR /PRINT ERROR HANDLER & PC
3010 6423 4300 JMS MINS /PRINT INSTRUCTION
3011 6424 4521 C8CRLF
3012 6425 4522 C8PRNT
3013 6426 6547 MESACT /PRINT ACTUAL AC,MQ,LINK
3014 6427 1054 TAD ACWAS
3015 6430 3776* DCA ACSAVE
3016 6431 4256 JMS MAC
3017 6432 1055 TAD MQWAS
3018 6433 3775* DCA MQSAVE
3019 6434 4264 JMS MMQ
3020 6435 1056 TAD LKWAS
3021 6436 3071 DCA LKDATA
3022 6437 4306 JMS MLK

```

```

3023 6440 4521 C8CRLF
3024 6441 4522 C8PRNT
3025 6442 6556 MESEXP
3026 6443 1072 TAD BIT11
3027 6444 3776* DCA ACSAVE
3028 6445 4256 JMS MAC
3029 6446 1065 TAD BIT8
3030 6447 3775* DCA MQSAVE
3031 6450 4264 JMS MMQ
3032 6451 1007 TAD BIT7
3033 6452 3071 DCA LKDATA
3034 6453 4306 JMS MLK
3035 6454 4521 C8CRLF
3036 6455 5777* JMP IEH
3037
3038 /SUBROUTINES FOR DISPLAY ROUTINES
3039
3040 6456 0000 MAC, 0 /DISPLAY AC MESSAGE
3041 6457 4522 C8PRNT
3042 6460 6663 MESAC
3043 6461 1776* TAD ACSAVE
3044 6462 4523 C8PRT4
3045 6463 5656 JMP I MAC
3046
3047 6464 0000 MMQ, 0 /DISPLAY MQ MESSAGE
3048 6465 4522 C8PRNT
3049 6466 6666 MESMQ
3050 6467 1775* TAD MQSAVE
3051 6470 4523 C8PRT4
3052 6471 5664 JMP I MMQ
3053
3054 6472 0000 MFL, 0 /DISPLAY FLAGS MESSAGE
3055 6473 4522 C8PRNT
3056 6474 6671 MESFL
3057 6475 1774* TAD FLSAVE
3058 6476 4523 C8PRT4
3059 6477 5672 JMP I MFL
3060
3061 6500 0000 MINS, 0 /DISPLAY OCTAL INSTRUCTION
3062 6501 4522 C8PRNT
3063 6502 6543 MESINS
3064 6503 1505 TAD I INSTTR
3065 6504 4523 C8PRT4
3066 6505 5700 JMP I MINS
3067
3068 6506 0000 MLK, 0 /DISPLAY LINK MESSAGE
3069 6507 4522 C8PRNT
3070 6510 6540 MESLK
3071 6511 1071 TAD LKDATA
3072 6512 4523 C8PRT4
3073 6513 5706 JMP I MLK
3074
3075 6514 0000 MESHDP, 0 /DISPLAY MESSAGE HEADER & PC
3076 6515 4521 C8CRLF
3077 6516 4522 C8PRNT

```

```

3070 6517 5230      ERRMES
3079 6520 4521      C0CPLF
3080 6521 4324      JMS MTN
3081 6522 4332      JMS MPC
3082 6523 5714      JMP I MESHDR
3083
3084 6524 0000      MTN, 0 /DISPLAY TEST NUMBER
3085 6525 4522      C0PRNT
3086 6526 6564      MESTN
3087 6527 1075      TAD TSTNO
3088 6530 4523      C0PRT4
3089 6531 5724      JMP I MTN
3090
3091 6532 0000      MPC, 0 /DISPLAY PC
3092 6533 4522      C0PRNT
3093 6534 6660      MESPC
3094 6535 1773      TAD PCSAVE
3095 6536 4523      C0PRT4
3096 6537 5732      JMP I MPC
3097
3098
3099 6540 4040      MESLK, TEXT " LK;"
3100 6541 1413
3100 6542 7200
3100 6543 4040      MESINS, TEXT " INST;"
3100 6544 1116
3100 6545 2324
3100 6546 7200
3101 6547 4040      MESACT, TEXT " ACTUAL "
3101 6550 0103
3101 6551 2425
3101 6552 0114
3101 6553 4040
3101 6554 4040
3101 6555 4000
3102 6556 4040      MESEXP, TEXT " EXPECTED "
3102 6557 0530
3102 6560 2005
3102 6561 0324
3102 6562 0504
3102 6563 4040
3102 6564 4000
3103 6565 4040      MESTN, TEXT " TN;"
3103 6566 2416
3103 6567 7200
3104
3105
3106
3107
    
```

```

3108 6573 6674
3109 6574 6677
3110 6575 6676
3111 6576 6675
3112 6577 6646
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126 6600 0000      XCBERR, 0
3127 6601 6002      IOF
3128 6602 7000      OWAPT, NOP /OVERWRITTEN WITH *JMP APTER* IF RUNNING UNDER APT CONTROL
3129 6603 3275      DCA ACSAVE /SAVE AC
3130 6604 6004      GTF
3131 6605 3277      DCA FLSAVE /SAVE THE FLAGS
3132 6606 7501      MQA
3133 6607 3276      DCA MQSAVE /SAVE THE MQ
3134 6610 7340      CLA CLL CMA /SUBTRACT A 1 FOR TRUE LOCATION
3135 6611 1200      TAD XCBERR /GET RETURN LOCATION
3136 6612 3274      DCA PCSAVE /SAVE ADD OF C0BERR CALL
3137 6613 6031      KSF /SAVE STATE OF SLU #1 REC FLAG FOR USE
3138 6614 7410      SKP / BY UNEXPECTED INT. ROUTINE
3139 6615 7040      CMA /KBCHK ROUTINE CLEARS FLAG UNCONDITIONALLY
3140 6616 3040      DCA KSFLG
3141 6617 4517      KBCHK /CHECK FOR KEYBOARD INTERVENTION
3142 6620 1777      TAD TSLUP
3143 6621 3300      DCA TLOOP
3144 6622 1020      TAD PSR /INHIBIT ERROR TYPEOUT?
3145 6623 0376      AND (0200 /MASK BIT 4 = INHIBIT BIT
3146 6624 7640      SZA CLA
3147 6625 5246      JMP IEH /NO ERROR PRINTOUT
3148 6626 1274      TAD PCSAVE
3149 6627 7041      CIA
3150 6630 1375      TAD (UEI
3151 6631 7650      SNA CLA
3152 6632 5774      JMP SERUEI /USE UNEXPECTED INT, DISPLAY
3153 6633 1274      TAD PCSAVE
3154 6634 7041      CIA
3155 6635 1373      TAD (SPE1
3156 6636 7650      SNA CLA
3157 6637 5772      JMP CPUSDIS /USE MICROINST SKIP FAILURE DISPLAY
3158 6640 1274      TAD PCSAVE
3159 6641 7041      CIA
3160 6642 1371      TAD (SPE2
3161 6643 7650      SNA CLA
    
```

```

3162 6644 5770* JMP CPUUIS /USE MICROINST DATA FAILURE DISPLAY
3163 6645 5707* JMP NORMDIS /USE NORMAL DISPLAY
3164 6646 1020 IEH, TAD PSR /INHIBIT ERPOP HALT?
3165 6647 7700 SVA CLA /TEST BIT 0
3166 6650 4524 CBRWIT /NO - GO TO THE INQUIRE ROUTINE
3167 6651 1020 TAD PSR /LOOP ON ERROR?
3168 6652 7904 RAL /TEST BIT 1
3169 6653 7710 SPA CLA
3170 6654 5703 JMP I TLOOP /LOOP
3171 6655 4303 JMS CBGET /CONTINUE TEST
3172 6656 7000 NOP /LEAV INTERRUPT SYSTEM DISABLED
3173 6657 5600 JMP I XCBERR
3174 6660 4040 MESPC, TEXT " PC:"
3175 6661 2003
3175 6662 7200 MESAC, TEXT " AC:"
3175 6663 4040
3175 6664 0103
3176 6665 7200 MESMQ, TEXT " MQ:"
3176 6666 4040
3176 6667 1521
3177 6670 7200 MESFL, TEXT " FL:"
3177 6671 4040
3177 6672 8014
3177 6673 7200
3178 6674 7777 PCSAVE, 7777
3179 6675 7777 ACSAVE, 7777
3180 6676 7777 MQSAVE, 7777
3181 6677 7777 FLSAVE, 7777
3182 6700 0000 TLOOP, 0
3183 6701 5702 APTOW1, JMP I .+1
3184 6702 7000 APTER
3185
3186
3187 6703 0000 CBGET, 0
3188 6704 1276 TAD MQSAVE
3189 6705 7421 MOL /RESTORE THE MQ
3190 6706 1277 TAD FLSAVE
3191 6707 7004 RAL /RESTORE THE LINK
3192 6710 7200 CLA
3193 6711 1200 TAD XCBERR /RESTORE AC IF IN CPU TEST, OTHERWISE CLEAR AC
3194 6712 1366 TAD (-2400
3195 6713 7710 SPA CLA
3196 6714 1275 TAD ACSAVE /RESTORE THE AC
3197 6715 5703 JMP I CBGET
3198
3199
3200 /*****
3201
3202 /XCSTART IS CALLED AT START OF PROGRAM TO PRINT THE MAINDEC NUMBER
3203 /AND THE SWITCH REGISTER QUESTION.
3204
3205 6716 0000 XCSTART, 0
3206 6717 4532 CBAPT /CHECK FOR APT CONTROL
3207 6720 4521 CRCRLF //PRINT A CR LF
3208 6721 4522 CBPRNT

```

```

3209 6722 6277 HWID /PRINT ID
3210 6723 4521 CRCRLF
3211 6724 4524 CBRWIT /ASK THE SWITCH REGISTER QUESTION
3212 6725 4540 CBH3 /ASK THE HCW3 QUESTION
3213 6726 6211 CDF 10
3214 6727 3765 DCA I (PASSNO /ZERO PASS COUNTER
3215 6730 6201 CDF 00
3216 6731 5716 LXCRST, JMP I XCSTART /EXIT CBSTART
3217
3218
3219
3220
3221
3222 /*****
3223
3224 /APT/ ROUTINE TO NOTIFY APT THAT THE PROGRAM IS RUNNING OK.
3225 6732 0000 APTOK, 0 /APT/
3226 6733 6002 IOF /APT/
3227 6734 7200 CLA /APT/
3228 6735 1354 TAD APTIMX /APT/DELAY 100MS.
3229 6736 3356 DCA APTCTX /APT/
3230 6737 1355 TAD APTIMY /APT/
3231 6740 3357 DCA APTCTY /APT/
3232 6741 2357 ISZ APTCTY /APT/
3233 6742 5341 JMP .-1 /APT/
3234 6743 2356 ISZ APTCTX /APT/
3235 6744 5337 JMP .-5 /APT/
3236 6745 6224 RIF /APT/AC=IF.
3237 6746 1364 TAD (6201 /APT/CREATE A CDF INST.
3238 6747 3350 DCA .+1 /APT/MODIFY NEXT CDF INST.
3239 6750 6201 CDF /APT/(MODIFIED CDF) DF=IF.
3240 6751 6272 CIF 70 /APT/IF=FIELD 7.
3241 6752 4763 JMS 6500 /APT/CALL APT = 'PROG OK'.
3242 6753 5732 JMP I APTOK /APT/RTN FROM APT = RTN TO CALL+1.
3243
3244 6754 7771 APTIMX, -7 /APT/
3245 6755 0000 APTIMY, 0 /APT/
3246 6756 0000 APTCTX, 0 /APT/
3247 6757 0000 APTCTY, 0 /APT/
3248
3249 6760 4310 HCW3MS, TEXT "HW3="
3249 6761 2763
3249 6762 7500
3250
3251 6763 6500
3252 6764 6201
3253 6765 0137
3254 6766 5400
3255 6767 6400
3256 6770 6422
3257 6771 2060
3258 6772 6406
3259 6773 2054
3260 6774 6213
3261 6775 6205

```

3262 6776 6200
3263 6777 2226
3264
3265
3266
3267
3268
3269
3270 7000 6002
3271 7001 7200
3272 7002 6224
3273 7003 1377
3274 7004 3227
3275 7005 7240
3276 7006 1770
3277 7007 6201
3278 7010 6272
3279 7011 5775

PAGE

/APR/ ROUTINE TO HANDLE ERRORS UNDER APR CONTROL.

APRTER, IOF /APR/
CLA /APR/
RIF /APR/AC=IF,
TAD (6201 /APR/CREATE A CDF INST.
DCA ,+3 /APR/MODIFY NEXT CDF INST.
CLA CMA /APR/
TAD XCERR /APR/AC=ERROR PC,
CDF /APR/(MODIFIED CDF) DF=IF.
CIF 70 /APR/IF=FIELD 7,
JMP 6520 /APR/CALL APR = 'ERROR'.

/ROUTINE USED FOR CONSOLE REGISTER CHANGES

3280
3281
3282
3283
3284
3285
3286 7012 0000 XCSR, 0 /GET MESSAGE PARAMETER
3287 7013 1612 TAD I XCSR /GET MESSAGE PARAMETER
3288 7014 3300 DCA PROPHS
3289 7015 2212 ISZ XCSR
3290 7016 1612 TAD I XCSR /GET THE REGISTER PARAMETER
3291 7017 3302 DCA PROPLC
3292 7020 2212 ISZ XCSR
3293 7021 4276 JMS PPRMS /PRINT REG QUESTION
3294 7022 1702 TAD I PROPLC /GET THE VALUE OF THE REGISTER
3295 7023 4523 C8PRT4 /PRINT THE 4 DIGITS
3296 7024 7346 CLA CLL CMA RTL /SETUP A COUNTER TO ACCEPT 4 DIGITS
3297 7025 3304 DCA TTYCNT /SAVE THE COUNTER
3298 7026 1374 TAD (CHARR0 /GET POINTER FOR FIRST CHARACTER
3299 7027 3232 DCA CHGCHR /SAVE THE POINTER FOR DIGITS
3300 7030 4525 LISN /WAIT FOR KEYBOARD INPUT
3301 7031 0001 I /CHECK FOR A OCTAL DIGIT
3302 7032 7050 CHGCHR, CHARR0 /THIS LOCATION WILL GET MODIFIED
3303 7033 7566 -212 /CHECK FOR LINE FEED
3304 7034 0200 START /LINE FEED TYPED- RETURN TO START
3305 7035 7563 -215 /CHECK FOR CARRIAGE RETURN
3306 7036 7064 RETYPE /RETYPE SP AND CONT IF DIGITS TYPED
3307 7037 7575 -203 /CHECK FOR A CONTROL C
3308 7040 7105 C8RM /CONTROL C TYPED -RETURN TO MONITOR
3309 7041 7555 -223 /CHECK FOR A CONTROL S
3310 7042 7145 CNTRS /WAS CONTROL S WAIT FOR 'Q OR 'C
3311 7043 0000 0 /NONE OF ABOVE CHARACTERS-ILLEGAL CHAR
3312 7044 7045 *.1 /GO TO NEXT ADDRESS TO PRINT ?
3313 7045 4522 C8PRNT /GO PRINT ?
3314 7046 7315 QESTMK /POINTER TO ? MESSAGE
3315 7047 5221 JMP RQEST /RETURN AND ASK QUESTION AGAIN

3316 7050 3702 CHARR0, DCA I PROPLC /SAVE THE LEAST SIGNIFICANT BIT
3317 7051 1373 TAD (CHARR1 /UPDATA POINTER FOR CHARACTERS 2 3 4
3318 7052 3232 DCA CHGCHR /SAVE THE POINTER ADDRESS
3319 7053 5230 JMP CHGCHR-2 /RETURN FOR NEXT CHARACTER INPUT
3320 7054 3303 CHARR1, DCA SAVCHR /SAVE THE CHARACTER TYPED
3321 7055 1702 TAD I PROPLC /GET THE VALUE OF REG
3322 7056 7106 CLL RTL /MOVE IT INTO NEXT POSITION
3323 7057 7004 RAL
3324 7060 1333 TAD SAVCHR /ADD NEW CHARACTER TO IT
3325 7061 3702 DCA I PROPLC /SAVE THE NEW VALUE
3326 7062 2304 ISZ TTYCNT /DONE ALL 4 CHARACTERS
3327 7063 5230 JMP CHGCHR-2 /NO GET NEXT INPUT FROM KEYBOARD
3328 7064 1374 RETYPE, TAD (CHARR0 /GET POINTER TO SEE IF REG ECHOED
3329 7065 7041 CIA /NEGATE THE POINTER
3330 7066 1232 TAD CHGCHR /GET THE POINTER STORED
3331 7067 7650 SNA CLA /ECHO VALUE OF REG?
3332 7070 5612 JMP I XCSR /NO-ONLY CR WAS TYPED-USE ORIGINAL VALUE
3333 7071 4276 JMS PPRMS /RE-ECHO VALUE TYPED
3334 7072 1702 TAD I PROPLC /GET VALUE OF REG
3335 7073 4523 C8PRT4 /PRINT THE 4 OCTAL DIGITS
3336 7074 4521 C8CRLF /ISSUE A CR AND LF
3337 7075 5612 JMP I XCSR /RETURN TO PROGRAM
3338 7076 0000 PPRMS, 0 /PRINT REGISTER MESSAGE
3339 7077 4522 C8PRNT
3340 7100 0000 PROPMS, 0
3341 7101 5676 JMP I PPRMS
3342
3343 7102 0000 PROPLC, 0
3344 7103 0000 SAVCHR, 0
3345 7104 0000 TTYCNT, 0
3346
3347
3348 7105 6007 C8RM, CAF /CHANGE INST AND DATA FIELD TO 0
3349 7106 6203 CDF CIF /GOTO 7600 OF THAT FIELD
3350 7107 5710 JMP I ,+1 /MONITOR STARTING ADDRESS
3351 7110 7600 7600
3352
3353
3354
3355
3356
3357
3358
3359

/ROUTINE TO CHECK FOR CONSOLE RECEIVE FLAG

3360 7111 0000 XCHKSF, 0 /SKIP ON CONSOLE RECEIVE FLAG
3361 7112 6031 KSF /RECEIVE FLAG NOT SET RETURN TO PROGRAM
3362 7113 5337 JMP NOCRF /CHECK TO SEE IF CONSOLE WAS ACTIVE
3363 7114 1022 TAD HCW2 / (VERSUS APR IN CONTROL),
3364 7115 0047 AND K4000
3365 7116 7650 SNA CLA /ACTIVE CONSOLE-CHECK FOR 'C OR 'G
3366 7117 5322 JMP ,+3 /APR ACTIVE,CLEAR CONSOLE RECEIVE FLAG
3367 7120 6032 KCC /RETURN TO PROGRAM
3368 7121 5711 JMP I XCHKSF /SAVE SUBROUTINE LINKAGE
3369 7122 4772 JMS SSUBLK /CHECK THE KEYBOARD CHARACTER
3370 7123 4525 LISN

```

3371 7124 7575      -203          /CODE FOR "C
3372 7125 7125      CRHM          /WAS A CONTROL C-EXIT TO MONITOR
3373 7126 7571      -207          /CODE FOR "G
3374 7127 7141      CNTRRG        /WAS "G ECHO CHAR-ENTER SR QUESTION
3375 7130 7555      -223          /CHECK FOR A CONTROL S
3376 7131 7147      CNTRSI1       /WAS A CONTROL S WAIT FOR "Q OR "C
3377 7132 6A9C      0              /CHAR WAS NOT "C OR "G
3378 7133 7134      ,+1          /ECHO CHAR AND QUESTION MARK
3379 7134 1371      IAD (277        /PRINT ?
3380 7135 4527      TYPE
3381 7136 5350      JMP RLK          /RETURN TO PROGRAM
3382 7137 2341      NOCRF, ISZ XCHKASF
3383 7140 5711      JMP I XCHKASF
3384
3385 7141 4522      CNTRLG, C8PRNT /PRINT "G AND CR LF
3386 7142 7317      UPARRG         /POINTER TO MESSAGE
3387 7143 4770      JMS XC8SW      /GO ASK THE SR QUESTION
3388 7144 5350      JMP RLK          /RETURN TO THE PROGRAM
3389
3390 7145 4767      CNTRS, JMS WAITQC /GO WAIT FOR A CONTROL Q OR C
3391 7146 5230      JMP CHGCHR=2   /GO WAIT FOR NEXT CHAR
3392
3393 7147 4767      CNTRS1, JMS WAITQC /WAIT FOR A CONTROL Q OR C
3394 7150 4766      RLK, JMS RSUBLK /RESTORE SUBROUTINE LINKAGE
3395 7151 5711      JMP I XCHKASF  /RETURN TO PROGRAM
3396 7152 0004      FILLER,0004    /SET TO NUMBER OF FILLERS REQUIRED.
3397
3398
3399
3400
3401
3402                /PRINT THE TWO OCTAL NUMBERS IN THE AC 6 THRU 11
3403
3404 7153 0000      XPRNT2, 0       /CALL BY "PRNT2"
3405 7154 3364      DCA TWOOCK
3406 7155 1364      TAD TWOOCK
3407 7156 7012      RTR
3408 7157 7010      RAR
3409 7160 4526      PRNT1
3410 7161 1364      TAD TWOOCK
3411 7162 4526      PRNT1
3412 7163 5753      JMP I XPRNT2
3413
3414 7164 0000      TWOOCK, 0
3415
3416
3417
3418 7166 7345
3419 7167 7521
3420 7170 7360
3421 7171 0277
3422 7172 7335
3423 7173 7054
3424 7174 7050
3425 7175 6520
    
```

```

3426 7176 6600
3427 7177 6201
3428                7200
3429                /PAGE
3430                /TYPE A CR AND LF WITH NUMBER OF FILLERS
3431                /AS DETERMINED BY LOCATION "FILLER"
3431 7200 0000      XC8CRLF,0        /CALL BY "CRLF"
3432 7201 7200      CLA
3433 7202 1215      TAD K215
3434 7203 4527      TYPE
3435 7204 1777      TAD FILLER
3436 7205 7040      CMA
3437 7206 3214      DCA XORS
3438 7207 1216      TAD K212
3439 7210 4527      TYPE
3440 7211 2214      ISZ XORS
3441 7212 5210      JMP ,=2
3442 7213 5600      JMP I XC8CRLF
3443
3444 7214 0000      XORS, 0
3445 7215 0215      K215,215
3446 7216 0212      K212,212
3447
3448                /PRINT 2 SPACES
3449
3450 7217 0000      SPACX2, 0       /CALL BY "SPACE2"
3451 7220 4522      C8PRNT
3452 7221 7223      ,+2
3453 7222 5617      JMP I SPACX2
3454 7223 4040      4040
3455 7224 0000      0000
3456
3457
3458                /COMPARE INPUT TO LIST FOLLOWING CALL
3459                /INPUT ONE CHARACTER IF AC=0
3460                /USE LAST INPUT IF AC NON ZERO
3461
3462 7225 0000      XLISN, 0       /CALL BY "LISN"
3463 7226 7640      SZA CLA
3464 7227 5255      JMP LISN1        /USE LAST INPUT SINCE AC NOT ZERO
3465 7230 6031      KSF
3466 7231 5230      JMP ,=1
3467 7232 6036      KRB
3468 7233 0376      AND (177
3469 7234 1045      TAD K200
3470 7235 3311      DCA CHIREC
3471 7236 1311      TAD CHIREC
3472 7237 1310      TAD M212
3473 7240 7450      SNA
3474 7241 5245      JMP ,+4          /IS IT A LFT
3475 7242 1307      TAD M3              /YES
3476 7243 7640      SZA CLA
3477 7244 5247      JMP ,+3          /IS IT A CRT
3478 7245 4521      CRCRLF        /NO
3479 7246 5255      JMP LISN1
    
```



```

3480 7247 1311 TAD CH1REC /GET THE CHAR
3481 7250 1375 TAD (-223 /CHECK FOR A CONTROL S
3482 7251 7650 SNA CLA /WAS IT A CONTROL S
3483 7252 5255 JMP LISN1 /YES=DO NOT ECHO CHARACTER
3484 7253 1311 TAD CH1REC
3485 7254 4527 TYPE /PRINT THE CHARACTER
3486 7255 1625 LISN1, TAD I XLISN /GET COMPARE VALUE
3487 7256 2225 ISZ XLISN
3488 7257 7450 SNA /EXIT?
3489 7260 5266 JMP LISN3 /YES
3490 7261 7500 SNA
3491 7262 5276 JMP LISNUM /LOOK FOR OCTAL NUMBER
3492 7263 1311 TAD CH1REC /COMPARE
3493 7264 7640 SZA CLA /EQUAL?
3494 7265 5273 JMP LISN2 /NO
3495 7266 3214 LISN3, DCA XORS
3496 7267 1625 TAD I XLISN
3497 7270 3225 DCA XLISN
3498 7271 1214 TAD XORS
3499 7272 5625 JMP I XLISN /AC IS ZERO UNLESS OCTAL NUMBER
3500 7273 7200 CLA
3501 7274 2225 ISZ XLISN
3502 7275 5255 JMP LISN1 /LOOK FOR OCTAL NUMBER
3503 7276 7200 LISNUM, CLA
3504 7277 1311 TAD CH1REC
3505 7300 1304 TAD M270
3506 7301 7500 SNA /IS IT LESS THAN 8?
3507 7302 5273 JMP LISN2 /NO, SO NOT AN OCTAL NUMBER
3508 7303 1042 TAD K10
3509 7304 7510 M270, SNA /IS IT GREATER THAN ZERO?
3510 7305 5273 JMP LISN2 /NO, SO NOT A NUMBER
3511 7306 5266 JMP LISN3
3512 7307 7775 M3, -3
3513 7310 7566 M212, 7566
3514 7311 0000 CH1REC, 0
3515
3516
3517 7312 4323 SRMSG, TEXT "#SR#"
3518 7313 2275
3519 7314 0000
3519 7315 7743 QESTMK, TEXT "??#"
3520 7316 0000
3519 7317 3607 UPARRG, TEXT ""G#"
3521 7320 4300
3520 /TYPE THE ASCII CHARACTER IN THE AC
3521
3522 7321 0000 XTYPE, 0 /CALL BY "TYPE"
3523 7322 3334 DCA CHAR /SAVE THE CHARACTER
3524 7323 7000 NOP/JMP I XTYPE /OVERWRITTEN IF RUNNING UNDER APT CONTROL.
3525 7324 4774 JMS CNTRLS /CONSOLF ACTIVE-CHECK FOR CONTROL S
3526 7325 1334 TAD CHAR /GET THE CHARACTER SAVED AND PRINT
3527 7326 6046 TDS
3528 7327 7200 CLA
3529 7330 6041 TSD
3530 7331 5330 JMP .-1

```

```

3531 7332 6042 TCF
3532 7333 5721 APTCON, JMP I XTYPE
3533
3534 7334 0000 CHAR, 0
3535
3536 /ROUTINE TO SAVE SUBROUTINE LINKAGES WHICH MAY GET DESTROYED
3537 / BY A KEYBOARD INTERVENTION CHECK.
3538
3539 7335 0000 SSUBLK, 0
3540 7336 1773 TAD XCHKKSF
3541 7337 3355 DCA LSAV1
3542 7340 1772 TAD MESAGX
3543 7341 3356 DCA LSAV2
3544 7342 1771 TAD XPRNT4
3545 7343 3357 DCA LSAV3
3546 7344 5735 JMP I SSUBLK
3547
3548 /ROUTINE TO RESTORE SUBROUTINE LINKAGES SAVED BY SSUBLK ROUTINE
3549
3550 7345 0000 RSUBLK, 0
3551 7346 1355 TAD LSAV1
3552 7347 3773 DCA XCHKKSF
3553 7350 1356 TAD LSAV2
3554 7351 3772 DCA MESAGX
3555 7352 1357 TAD LSAV3
3556 7353 3771 DCA XPRNT4
3557 7354 5745 JMP I RSUBLK
3558 7355 0000 LSAV1, 0
3559 7356 0000 LSAV2, 0
3560 7357 0000 LSAV3, 0
3561
3562 /ASK SWITCH REGISTER QUESTION
3563
3564 7360 0000 XC8SW, 0
3565 7361 4537 C08C
3566 7362 7312 SRMSG
3567 7363 0020 P8R
3568 7364 5760 JMP I XC8SW
3569
3570 7371 7464
3571 7372 7423
3572 7373 7111
3573 7374 7506
3574 7375 7555
3575 7376 0177
3576 7377 7152 PAGE
3577
3578
3579
3580
3581 /*****
3582
3583 7400 0000 XC8APT, 0
3584 7401 6002 IOF

```

```

3585 7402 1022 TAD HCW2 /RUN UNDER APT CONTROL$
3586 7403 7700 SNA CLA
3587 7404 5604 JMP I XC8APT
3588 7405 1777 TAD APT0W1 /YES--OVERWRITE C8EKR ROUTINE (FLD 2)
3589 7406 3776 DCA 0WAPT
3590 7407 1775 TAD APTCON /OVERWRITE 'TYPE' ROUTINE (FLD 0) TO INHIBIT
3591 / OUTPUT TO VT78 VIDEO.
3592 7410 3774 DCA TYOUT
3593 7411 6211 CDF 10
3594 7412 1773 TAD APT0W1 /OVERWRITE 'ERROR' ROUTINE (FLD 1)
3595 7413 3776 DCA 0WAPT
3596 7414 1372 TAD (APTICON /OVERWRITE 'TYPE' ROUTINE (FLD 1)
3597 7415 3771 DCA I (TYOUT1
3598 7416 6231 CDF 30 /GET HARDWARE CONFIG, WORD 3 FROM APT LOADER/MONITOR
3599 7417 1770 TAD I (APTHW3
3600 7420 6201 CDF 00
3601 7421 3023 DCA HCW3 /STORE IN LOC 23 IN FIELD 0
3602 7422 5767 JMP LXC8ST
3603
3604 /PRINT PACKED ASCII TEXT TERMINATED BY
3605 /SIX-BIT 00
3606
3607 7423 0000 MESAGX, 0
3608 7424 4517 KRCHK /CHECK FOR KEYBOARD INTERVENTION
3609 7425 1623 TAD I MESAGX
3610 7426 3263 DCA FOROCK
3611 7427 2223 ISZ MESAGX /SET UP RETURN
3612 7430 1663 TAD I FOROCK
3613 7431 7012 RTR
3614 7432 7012 RTR
3615 7433 7012 RTR
3616 7434 4241 JMS MESAGF
3617 7435 1663 TAD I FOROCK
3618 7436 4241 JMS MESAGF
3619 7437 2263 ISZ FOROCK
3620 7440 5230 JMP ,=-10
3621 7441 0000 MESAGF, 0
3622 7442 0043 AND K77
3623 7443 7450 SNA /TERMINATOR (00)?
3624 7444 5623 JMP I MESAGX /YES
3625 7445 1260 TAD M43
3626 7446 7450 SNA /CRLF?
3627 7447 5256 JMP ,+7 /YES
3628 7450 1261 TAD K3
3629 7451 7510 SPA /200 OR 300
3630 7452 1044 TAD K100 /300
3631 7453 1262 TAD K240 /200
3632 7454 4527 TYPE
3633 7455 5641 JMP I MESAGF
3634 7456 4521 C8CRLF
3635 7457 5641 JMP I MESAGF
3636 7460 7735 M43, 7735
3637 7461 0003 K3, 0003
3638 7462 0240 K240,240
3639 7463 0000 FOROCK, 0
    
```

```

3640
3641 /PRINT FOUR OCTAL NUMBERS IN AC 11 THRU 0 FOLLOWED
3642 /BY TWO SPACES
3643
3644 7464 0000 XPRNT4, 0 /CALL BY "PRNT4"
3645 7465 3263 DCA FOROCK
3646 7466 1263 TAD FOROCK
3647 7467 7012 RTR
3648 7470 7012 RTR
3649 7471 7012 RTR
3650 7472 4530 PRNT2
3651 7473 1263 TAD FOROCK
3652 7474 4530 PRNT2
3653 7475 4531 SPACE2
3654 7476 5664 JMP I XPRNT4
3655
3656 /PRINT THE OCTAL NUMBER IN AC 9 THRU 11
3657 7477 0000 XPRNT1, 0 /CALL BY "PRNT1"
3658 7500 0305 AND K7
3659 7501 1304 TAD K260
3660 7502 4527 TYPE
3661 7503 5677 JMP I XPRNT1
3662
3663 7504 0260 K260, 260
3664 7505 0007 K7, 7
3665
3666 /ROUTINE TO CHECK FOR A CONTROL S WHILE TYPING OUT MESSAGES
3667 /TO EXIT ROUTINE IF A CONTROL S WAS TYPED-A CONTROL Q OR C MUST BE
3668 /INPUTTED ON THE KEYBOARD
3669
3670 7506 0000 CNTRL$, 0
3671 7507 6031 KSF
3672 7510 5706 JMP I CNTRL$ /SKIP ON CONSOLE KEYBOARD FLAG
3673 7511 6034 KRS /RETURN TO TYPE ROUTINE-FLAG NOT SET
3674 7512 0366 AND (177 /READ THE CHARACTER STATICALLY
3675 7513 1365 TAD (-23 /MASK TO 7 BIT ASCII
3676 7514 7640 SZA /CHECK FOR A CONTROL S
3677 7515 5706 JMP I CNTRL$ /WAS IT A CONTROL S
3678 7516 6032 KCC /NO-RETURN WITH KEYBOARD FLAG STILL SET
3679 7517 4321 JMS WAITQC /CLEAR KEYBOARD FLAG FROM 'S
3680 7520 5726 JMP I CNTRL$ /WAIT FOR CONTROL Q OR C
3681 /RETURN TO PRINT MESSAGE BEING TYPED
3682 7521 0000 WAITQC, 0
3683 7522 6031 KSF /ROUTINE TO WAIT FOR CONTROL Q OR C
3684 7523 5322 JMP ,=-1 /WAIT FOR A CONTROL Q OR C TO EXIT
3685 7524 6036 KRB /
3686 7525 0366 AND (177 /READ THE CHARACTER TYPED
3687 7526 1364 TAD (-3 /MASK TO 7 BIT ASCII
3688 7527 7450 SNA /CHECK FOR A CONTROL C
3689 7530 5763 JMP C8RM /WAS IT A CONTROL C?
3690 7531 1362 TAD (-7 /YES-RESTORE MONITOR AND RETURN
3691 7532 7450 SNA /CHECK FOR A LINE FEED CHARACTER
3692 7533 5761 JMP START /WAS IT A LINE FEED
3693 7534 1362 TAD (-7 /YES GO RESTART THE PROGRAM
3694 7535 7640 SZA CLA /CHECK FOR A CONTROL Q 'Q
    /WAS IT A CONTROL Q
    
```

```

3695 7536 5322 JMP WAITQC+1 /NO-WAIT FOR APPROPRIATE KEY
3696 7537 5721 JMP I WAITQC /RETURN TO WHENCE IT CAME
3697
3698
3699
3700 7540 0J00 XCBH3, 0
3701 7541 4537 CRRC /PRINT QUESTION
3702 7542 6760 HCW3MS
3703 7543 0023 HCW3
3704 7544 5740 JMP I XCBH3
3705
3706
3707
3708 7561 0200
3709 7562 7771
3710 7563 7105
3711 7564 7775
3712 7565 7755
3713 7566 8177
3714 7567 6731
3715 7570 6143
3716 7571 7402
3717 7572 5600
3718 7573 6713
3719 7574 7323
3720 7575 7333
3721 7576 6602
3722 7577 6701

```

FIELD 1

```

0000 11111111 11111110 11110000 00000000 11111111 11111111 11111111 11111111
0100 11111111 11111111 11111111 11111111 11000000 00000000 00000000 00000001
0200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0300 11111111 11111111 11111111 11111111 11111111 11111111 11100000 00001111
0400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0500 11111111 11111111 11111111 11111111 11111111 11111111 11100001 11111111
0600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0700 11111111 11111111 11111111 11111111 11111111 11111111 10000000 00011111
1000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1100 11111111 11111111 11111111 11111111 11111111 11111111 11111110 00000001
1200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1300 11111111 11111111 11111111 11111111 11111111 11111100 00000000 00000011
1400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1500 11111111 11111111 11111110 00000000 00000000 00000000 00000000 00001111
1600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1700 11111111 11111111 11111111 11111111 11111111 10000000 00000000 00000001
2000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2100 11111111 11111111 11111111 11111110 00000000 00000000 00000000 00000011
2200 11111111 11111111 11111111 11111111 11111111 00000000 00000000 00000000
2300 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
2400
2500
2600
2700
3000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3100 11111111 11111111 11111111 11111111 11111111 11111111 10000001 11111111
3200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3300 11111111 11111111 11111111 11111111 11111111 11111111 11111111 01111111
3400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3500 11111111 11000000 00000000 00000000 00000000 00000000 00000000 00111111
3600 11111111 11111111 11111111 11111111 11111111 11111111 11110000 00000000
3700 00000000 00000000 00000000 00000000 00000000 00000000 00000000 01111111

```



```

3778 6300 KCF1=6300 /CLEAR RECEIVE FLAG
3779 6301 KSF1=6301 /SKIP ON RECEIVE FLAG
3780 6302 KCC1=6302 /CLEAR RECEIVE FLAG AND AC
3781 6304 KRS1=6304 /*OR* CONTENTS OF RECEIVE BUFFER INTO AC
3782 6305 KIE1=6305 /AC 11=1 SET INTERRUPT ENABLE
3783 /AC 11=0 CLEAR INTERRUPT ENABLE
3784 6306 KRB1=6306 /LOAD CONTENT OF RECEIVE BUFFER INTO AC; CLR RECV FLG
3785 6307 KMD1=6307 /SET THE OPERATING MODE OF SLU 2
3786
3787 /TRANSMIT IOTS
3788
3789 6310 SPF1=6310 /SET TRANSMIT FLAG ENABLE
3790 6311 TSF1=6311 /SKIP IF TRANSMIT FLAG IS SET AND ENABLED
3791 6312 TCF1=6312 /CLEAR TRANSMIT FLAG ENABLE
3792 6313 TSB1=6313 /SET BAUD RATE
3793 6314 TPC1=6314 /SAME AS TSB1
3794 6315 TSK1=6315 /SKIP IF INT EN SET AND IF RECV FLAG IS SET
3795 /OR IF XMIT FLAG AND XMIT EN ARE BOTH SET
3796 6316 TLS1=6316 /LOAD TRANSMIT BUFFER FROM AC4-11 AND SEND CHAR
3797 /OUT OVER SERIAL LINE, SET THE TRANSMIT FLAG ENABLE,
3798 /AS SOON AS A NEW CHARACTER CAN BE LOADED INTO
3799 /THE TRANSMITTER, SET TRANSMIT FLAG.
3800
3801 ////////////////////////////////////////////////////
3802 //SERIAL LINE UNIT #3
3803 ////////////////////////////////////////////////////
3804
3805 /RECEIVER IOTS
3806
3807 6320 KCF2=6320 /CLEAR RECEIVE FLAG
3808 6321 KSF2=6321 /SKIP ON RECEIVE FLAG
3809 6322 KCC2=6322 /CLEAR RECEIVE FLAG AND AC
3810 6324 KRS2=6324 /*OR* CONTENTS OF RECEIVE BUFFER INTO AC
3811 6325 KIE2=6325 /AC 11=1 SET INTERRUPT ENABLE
3812 /AC 11=0 CLEAR INTERRUPT ENABLE
3813 6326 KRB2=6326 /LOAD CONTENT OF RECEIVE BUFFER INTO AC; CLR RECV FLG
3814
3815 /TRANSMIT IOTS
3816
3817 6330 SPF2=6330 /SET TRANSMIT FLAG ENABLE
3818 6331 TSF2=6331 /SKIP IF TRANSMIT FLAG IS SET AND ENABLED
3819 6332 TCF2=6332 /CLEAR TRANSMIT FLAG ENABLE
3820 6333 TSB2=6333 /SET BAUD RATE
3821 6334 TPC2=6334 /SAME AS TSB1
3822 6335 TSK2=6335 /SKIP IF INT EN SET AND IF RECV FLAG IS SET
3823 /OR IF XMIT FLAG AND XMIT EN ARE BOTH SET
3824 6336 TLS2=6336 /LOAD TRANSMIT BUFFER FROM AC4-11 AND SEND CHAR
3825 /OUT OVER SERIAL LINE, SET THE TRANSMIT FLAG ENABLE,
3826 /AS SOON AS A NEW CHARACTER CAN BE LOADED INTO
3827 /THE TRANSMITTER, SET TRANSMIT FLAG.
3828
3829
3830
3831
3832

```

```

3833 /BAUD RATE TABLE
3834 ////////////////////////////////////////////////////
3835
3836 /AC 8-11      BAUD RATE      AC 8-11      BAUD RATE
3837 /-----
3838 /0              50              10             1800
3839 /1              75              11             2000
3840 /2             110             12             2400
3841 /3             134.5          13             3600
3842 /4             150             14             4800
3843 /5             300             15             7200
3844 /6             600             16             9600
3845 /7            1200             17            19200
3846
3847
3848 ////////////////////////////////////////////////////
3849 //SLU #2 MODE
3850 ////////////////////////////////////////////////////
3851
3852 0027 MODE1=0027 /8 BIT,NO PARITY,1 STOP
3853 0006 MODE2=0006 /8 BIT,ODD PARITY,1 STOP
3854 0007 MODE3=0007 /8 BIT,EVEN PARITY,1 STOP
3855
3856
3857 ////////////////////////////////////////////////////
3858 //FLOPPY IOTS
3859 ////////////////////////////////////////////////////
3860
3861 6750 SEL=6750 /IF AC11=0 SELECT UNIT A (SEL BY CAF & POWER ON)
3862 /IF AC11=1 SELECT UNIT B
3863 6751 LCD=6751 /LOAD COMMAND REG,CLEAR AC
3864 6752 XDR=6752 /TRANSFER DATA REGISTER
3865 6753 STR=6753 /SKIP ON XFER REG FLAG,CLEAR FLAG
3866 6754 SER=6754 /SKIP ON ERROR FLAG,CLEAR FLAG
3867 6755 SDN=6755 /SKIP ON DONE FLAG,CLEAR FLAG
3868 6756 INTR=6756 /INTERRUPT ENABLE/DISABLE
3869 6757 INIT=6757 /INIT CONTROLLER & RECALIBRATE DRIVES
3870
3871
3872 ////////////////////////////////////////////////////
3873 //PARALLEL I/O INTERFACE IOTS
3874 ////////////////////////////////////////////////////
3875
3876 6660 PSSF=6660 /SET PRINT FLAG
3877 6661 PSKP=6661 /SKIP ON FLAG
3878 6662 PCLF=6662 /CLEAR FLAG
3879 /6663=UNUSED
3880 6664 PSTB=6664 /LOAD PRINTER BUFFER;ISSUE CHAR STROBE
3881 6665 PCTE=6665 /IF AC11=0 CLEAR INTERRUPT ENABLE
3882 /IF AC11=1 SET INTERRUPT ENABLE
3883 6666 PCLP=6666 /LOAD PRINTER BUFFER;ISSUE CHAR STROBE;CLEAR FLAG
3884 6667 PRDB=6667 /IF OUT=0 READ EXTERNAL DEVICE(PRINTER)
3885 /IF OUT=1 READ INTERFACE BUFFER
3886
3887

```

```

3088
3089      6500  LOSK=6500      /SKIP ON DONE FLAG
3090      6501  LORR=6501      /IF OUT=0 READ EXTERNAL DEVICE(PRINTER)
3091
3092      6502  LQMP=6502      /MOVE PAPER
3093      6503  LOHC=6503      /MOVE CARRIAGE
3094      6504  LQPC=6504      /PRINT CHAP
3095      6505  LQRS=6505      /READ STATUS AND CLEAR DONE FLAG
3096      6506  LQLS=6506      /WRITE STATUS AND SET DONE FLAG
3097      6507  LQRE=6507      /RESTORE AND CLEAR DONE FLAG
3098
3099
3100
3101
3102      0000  0000  *0      ERROR
3103      0000  4450      /LOCATIONS 0-10 GET DESTROYED BY MEMORY EXTENSION
3104
3105
3106
3107      0004  0004  *4
3108      0004  7000      NOP
3109      0005  7000      NOP
3110      0006  7000      NOP
3111
3112
3113
3114
3115
3116      0024  0024  *24      NSQBDR, 16      /BAUD RATE USED FOR NORMAL TEST SEQUENCE
3117
3118      0025  0000      SDEV, 0      /DEVICE CODE BEING TESTED.
3119      0026  0016      BDR1, 16      /BAUD RATE INDICATOR FOR SLU#1
3120      0027  0016      BDR2, 16      /BAUD RATE INDICATOR FOR SLU#2
3121      0030  0016      BDR3, 16      /BAUD RATE INDICATOR FOR SLU#3
3122      0031  0000      LOOPA, 0      /STATUS OF LOOPAROUND (1=LOOPED,0=NOT LOOPED)
3123
3124
3125
3126      0032  7200      C1CRLF=JMS I .
3127      0033  4433      XC1CRLF
3128      0033  1660      SLU2MC=JMS I .
3129      0034  4434      XMODE2
3130      0034  7414      C1PRNT=JMS I .
3131      0035  4435      MFSAG
3132      0035  7453      C1PRT4=JMS I .
3133      0036  4436      X1PRN4
3134      0036  7013      C1SWIT=JMS I .
3135      0037  4437      XC1SW
3136      0037  7223      L1SNF1=JMS I .
3137      0040  4440      XL1SN1
3138      0041  7215      SPAC21=JMS I .
3139      0041  4441      PRN11=JMS I .
3140      0042  7400      X1PRN1
3141      0042  4442      TYPE1=JMS I .
3142      0042  7400      X1TYPE

```

```

3943
3944      0043  4443      C1EOP=JMS I .
3945      0044  7306      XC1EOP
3946      0044  4444      PRN21=JMS I .
3947      0045  7525      X1PRN2
3948      0045  4445      CHK1SF=JMS I .
3949      0045  7106      XCHK1SF
3950
3951
3952      0046  4446      LOOPPC=JMS I .
3953      0046  2200      PCLOOP
3954      0047  4447      DONLOP=JMS I .
3955      0047  2742      LOPDON
3956      0050  4450      EPROR=JMS I .
3957      0051  6020      XC1ERR
3958      0051  4451      TSWAT=JMS I .
3959      0051  2333      WATTSF
3960      0052  4452      KSWAT=JMS I .
3961      0052  2346      WATKSF
3962      0053  4453      SLUDAT=JMS I .
3963      0053  2421      DATSLU
3964      0054  4454      SLUDER=JMS I .
3965      0054  2462      DERSLU
3966      0055  4455      CLREIF=JMS I .
3967      0055  2234      EIFCLR
3968      0056  4456      CLREXI=JMS I .
3969      0056  2214      EXICLR
3970      0057  4457      CLRERI=JMS I .
3971      0057  2220      ERICLR
3972      0060  4460      SETEXI=JMS I .
3973      0061  2274      EXISET
3974      0061  4461      SETERI=JMS I .
3975      0061  2230      ERISER
3976      0062  4462      KBCHK=JMS I .
3977      0062  3320      CHKKB
3978      0063  4463      MIOT=JMS I .
3979      0063  3024      XMIOT
3980      0064  4464      WAIT=JMS I .
3981      0064  5553      XWAIT
3982      0065  4165      GETSR=JMS I .
3983      0065  3312      XGETSR
3984      0066  4466      GETHW=JMS I .
3985      0066  6346      XGHW
3986      0067  4467      LOOP=JMS I .
3987      0067  3233      SLOOPA
3988      0070  4470      DELAY=JMS I .
3989      0070  3245      XDELAY
3990      0071  4471      CLREPI=JMS I .
3991      0071  4475      XCLEPI
3992      0072  4472      SETEPI=JMS I .
3993      0072  4501      XSEPI
3994      0073  4473      VDELAY=JMS I .
3995      0073  1667      XVDLY
3996
3997

```

/MSLU IOTS

3998		4474	MKCF=JMS I ,
3999	0071	2620	XMKCF
4000		4475	MKSF=JMS I ,
4001	0075	2625	XMKSF
4002		4476	MKCC=JMS I ,
4003	0076	2612	XMKCC
4004		4477	MKPS=JMS I ,
4005	0077	2617	XMKPS
4006		4500	MKIE=JMS I ,
4007	0100	2624	XMKIE
4008		4501	MKRB=JMS I ,
4009	0101	2631	XMKRB
4010		4502	MSPF=JMS I ,
4011	0102	2636	XMSPF
4012		4503	MTSF=JMS I ,
4013	0103	2643	XMTSF
4014		4504	MTCF=JMS I ,
4015	0104	2650	XMTCF
4016		4505	MTSB=JMS I ,
4017	0105	2655	XMTSB
4018		4506	MTPC=JMS I ,
4019	0106	2707	XMTPC
4020		4507	MTSK=JMS I ,
4021	0107	2714	XMTSK
4022		4510	MTLS=JMS I ,
4023	0110	2721	XMTLS
4024		4511	SLUCAF=JMS I ,
4025	0111	3752	XSCAF
4026		4512	CPSLU=JMS I ,
4027	0112	3765	XCFSLU
4028		4513	KLSIM=JMS I ,
4029	0113	4511	XKLSIM
4030		4514	RLOOP=JMS I ,
4031	0114	3240	CLOOPA
4032		4515	APTREP=JMS I ,
4033	0115	2502	REPAPT
4034		4516	SPCIE=JMS I ,
4035	0116	3544	XSPCIE

/LOCATIONS USED BY THE PROGRAM

4041			
4042	0117	0000	INTFLG, 0
4043	0120	0000	EXMITI, 0
4044	0121	0000	ERECI, 0
4045	0122	0000	CNT, 0
4046	0123	0000	CNT1, 0
4047	0124	0000	TESTF1, 0
4048	0125	0000	TSTLOP, 0
4049	0126	0000	SLUXMT, 0
4050	0127	0000	SLUREC, 0
4051	0130	0000	BLANK, 0
4052	0131	0000	EAC, 0

4053	0132	0000	GOOD, 0
4054	0133	0000	RXHERE, 0
4055	0134	0000	COMP, 0
4056	0135	0000	EPRNTI, 0
4057	0136	0000	ECONSI, 0
4058	0137	0000	PASSNO, 0
4059	0140	0000	TSTNU, 0

/ROUTINE TO SETUP FIELD 0 TO HANDLE INTERRUPTS FROM ANOTHER FIELD

4061			
4062			
4063			
4064			
4065	0141	0000	PATCH, 0
4066	0142	1541	TAD I PATCH
4067	0143	3166	DCA ITOLOC
4068	0144	1177	TAD (LOC1=1
4069	0145	3010	DCA 10
4070	0146	3011	DCA 11
4071	0147	7346	CLA CLL CMA RTL
4072	0150	3123	DCA CNT1
4073	0151	6224	RIF
4074	0152	1155	TAD KCDF
4075	0153	3157	DCA .+4
4076	0154	1410	TAD I 10
4077	0155	6201	KCDF, CDF 00
4078	0156	3411	DCA I 11
4079	0157	6201	CDF
4080	0160	2123	ISE CNT1
4081	0161	5154	JMP .-5
4082	0162	2141	ISE PATCH
4083	0163	5541	JMP I PATCH
4084			
4085	0164	6244	LOC1, RMF
4086	0165	5403	JMP I 3
4087	0166	0000	ITOLOC, 0

/GET THE INTERRUPT SERVICE ADDRESS

/SETUP AUTO INDEX 10

/CLEAR AUTO INDEX 11

/SETUP MOVE COUNTER TO -3

/SAVE THE MOVE COUNTER

/READ THIS INSTRUCTION FIELD

/MAKE CDF INSTR TO THIS FIELD

/SAVE THE CDF INSTRUCTION

/GET FIRST CONTENTS TO TRANSFER

/CHANGE DATA FIELD TO FIELD 0

/SAVE THE CONTENTS IN ADDRESSES 1-3

/CHANGE DATA FIELD BACK TO PROGRAM FIELD

/DONE

/NO-GO MOVE NEXT CONTENTS FROM 1-3

/BUMP RETURN

/RETURN TO THE PROGRAM

```

4088
4089          0200 *200
4090
4091          0200 6002 CIREST, IOF
4092          0201 3140 DCA TSTNU /CLEAR TEST # FOR ERROR DISPLAY
4093          0202 7240 CLA CMA
4094          0203 3136 DCA ECONS1 /WHILE SLU'S ARE LOOPED OPERATOR CANNOT
4095 /GET CONTROL VIA CONSOLE PACKAGE,
4096 /SLU'S ARE LOOPED DURING SLU TESTS 6-17,
4097          0204 4141 JMS PATCH /SETUP INTERRUPT SERVICE LINKAGE
4098          0205 6200 SKPICHN
4099          0206 4465 GETSR /GET PSEUDO SWITCH REGISTER FROM FIELD 0.
4100          0207 0377 AND (0020
4101          0210 7040 SZA CLA /EXECUTE SLU TEST?????
4102          0211 5776 JMP PRTEST /NO
4103          0212 4775 JMS INIT1 /INITIALIZE FOR SLU TESTING STARTUP
4104          0213 4277 SMES, JMS PSMES /PRINT "SLU TESTING" MESSAGE IF NOT
4105 / NOT UNDER APT CONTROL,
4106          0214 4774 NXTDC, JMS BORNS /SETUP BAUD RATE FOR NORMAL TEST SEQUENCE
4107          0215 4773 JMS XIOT /SETUP DEVICE CODE FOR SLU TO TEST
4108          0216 4433 SLURMC /SETUP MODE CONTROL IN SLU #2
4109          0217 0027 MODE1
4110          0220 4515 APTREP /IF UNDER APT CONTROL REPORT OK STATUS TO APT
4111
4112
4113
4114
4115 /*****
4116 /INITIALIZATION TEST
4117 /TEST 1 - CHECKS THAT CAF (INITIALIZE) WILL CLEAR THE RECEIVE FLAG AND DISABLE XMIT FLAG,
4118 /NOTE: INITIALIZE SETS THE SERIAL LINE UNIT'S INTERRUPT ENABLE,
4119 /*****
4120
4121          0221 4446 TEST1, LOOPPC /SETUP LOOPING ADDRESS
4122          0222 1372 TAD (1 /SETUP TEST NUMBER FOR ERROR DISPLAY
4123          0223 3140 DCA TSTNU
4124          0224 6302 IOF
4125          0225 4455 CLRIF /CLEAR 'EXPECTING INTERRUPT' FLAGS,
4126          0226 4511 SLUCAF /INITIALIZE THE MODULE - CAF SETS INT ENA ON SLU
4127          0227 4475 MKSF /SKIP ON RECEIVE FLAG
4128          0230 7410 SKP
4129          0231 4450 ERROR /RECEIVE FLAG SET OR KSF SKIPPED
4130          0232 4533 MTSF /SKIP ON TRANSMIT FLAG SET AND ENABLED
4131          0233 7410 SKP
4132          0234 4450 ERROR /TRANSMIT FLAG SET AND ENABLED OR TSP SKIPPED
4133          0235 4537 MTSK /SKIP ON XMIT/RECEIVE + INT ENA
4134          0236 7410 SKP
4135          0237 4450 ERROR /TRANSMIT FLAG SET & ENABLED OR MTSK SKIPPED
4136          0240 4447 DONL0P /CHECK FOR LOOP ON TEST
4137
4138
4139 /*****
4140 /TEST 2 - TRY TO CLEAR SLU INT ENA BY ISSUING A KIE COMMAND. THEN TEST THE SLU XMIT
4141 /FLAG EN TO SET BY SPF AND CLEAR BY TCF. THE FLAG IS CHECKED WITH TSP AND TSK, IF AN
4142 /INTERRUPT OCCURRED, IT MAY BE DUE TO INT ENA NOT BEING CLEARED BY KIE AND DATA BIT 11 ON A 0,

```

```

4143 /*****
4144
4145          0241 4446 TEST2, LOOPPC /SETUP TEST NUMBER FOR ERROR DISPLAY
4146          0242 1371 TAD (2
4147          0243 3140 DCA TSTNU
4148          0244 4455 CLRIF /CLEAR EXPECTING INTERRUPT FLAGS
4149          0245 4511 SLUCAF /CLEAR ALL FLAGS + SET SLU INT ENA
4150          0246 6301 ION /TURN THE INTERRUPT ON
4151          0247 4475 MKSF /CHECK TO SEE IF RECEIVE FLAG IS A 0
4152          0250 7610 SKP CLA
4153          0251 4450 ERROR /RECEIVE FLAG SET OR KSF SKIPPED
4154          0252 4500 MKIE /CLEAR SLU INT ENA
4155          0253 4502 MSPF /SET XMIT FLAG ENABLE
4156          0254 4533 MTSF /SKIP ON XMIT FLAG SET AND ENABLED
4157          0255 4450 ERROR /SPF FAILED TO SET XMIT FLAG OR NO SKIP OCCURRED
4158          0256 4507 MTSK /SKIP ON XMIT/RECEIVE + INT ENA ON A 1
4159          0257 7410 SKP
4160          0260 4450 ERROR /TSK SKIPPED OR KIE AND DATA 11 L FAILED TO CLEAR INT ENA
4161          0261 4475 MKSF /SKIP ON RECEIVE FLAG
4162          0262 7410 SKP
4163          0263 4450 ERROR /RECEIVE FLAG SET BY ABOVE CODE
4164          0264 4504 MTSF /CLEAR TRANSMIT FLAG ENABLE
4165          0265 4503 MTSF /SKIP ON XMIT FLAG SET AND ENABLED
4166          0266 7410 SKP
4167          0267 4450 ERROR /TCF FAILED TO CLEAR XMIT FLAG ENABLE
4168          0270 4507 MTSK /SKIP ON XMIT/RECEIVE + INT ENA ON A 1
4169          0271 7610 SKP CLA
4170          0272 4450 ERROR /TSK SKIPPED WITH XMIT FLAG + INT ENA A 0
4171          0273 4475 MKSF /SKIP ON RECEIVE FLAG
4172          0274 7610 SKP CLA
4173          0275 4450 ERROR /RECEIVE FLAG GOT SET BY ABOVE CODE
4174          0276 4447 DONL0P
4175
4176 /PRINT "SLU TESTING MESSAGE"
4177
4178          0277 0000 PSMES, 0
4179          0300 4432 C1CRLF
4180          0301 4434 C1PRNT
4181          0302 3303 SLUMES
4182          0303 4434 C1PRNT
4183          0304 3306 TESMES
4184          0305 4432 C1CRLF
4185          0306 5677 JMP I PSMES
4186
4187
4188
4189          0371 0002
4190          0372 0001
4191          0373 3000
4192          0374 3200
4193          0375 3077
4194          0376 3400
4195          0377 0020
4196          0400 PAGE

```



```

4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229

```

```

/*****
/TEST 3 = CHECKS THAT CAF WILL CLEAR THE TRANSMIT FLAG EN, THE PROGRAM
/CHECKS THAT NO INTERRUPTS OCCURRED.
/*****
TEST3, LOOPPC          /SETUP TEST NUMBER FOR ERROR DISPLAY
      TAD (3
      DCA TSTNU
      CLREIF          /CLEAR EXPECTING INTERRUPT FLAGS,
      SLUCAF
      ION             /TURN THE INTERRUPT ON
      MKIE           /CLEAR SLU INT ENA
      MSPF          /SET THE TRANSMIT FLAG ENABLE
      MTSF          /SKIP ON THE XMIT FLAG SET AND ENABLED
      ERROR        /SPF FAILED TO SET THE XMIT FLAG
      MTSK         /SKIP ON XMIT/RECEIVE + INT ENA
      SKP
      ERROR        /TSK SKIPPED WITHOUT INT ENA SET OR KIE FAILED
      SLUCAF       /CLEAR ALL FLAGS
      MTSF          /SKIP ON THE TRANSMIT FLAG SET AND ENABLED
      SKP
      ERROR        /FAILED TO CLEAR XMIT FLAG
      MTSK         /SKIP ON RECEIVE FLAG
      SKP CLA
      ERROR        /RECEIVE FLAG SET BY ABOVE CODE
      DONLDP

```

```

/*****
/TEST 4 = CHECK THAT CAF WILL SET SLU INT ENABLE AND THAT KIE
/AND DATA 11 ON A 0 WILL CLEAR IT USING XMIT FLAG TO INTERRUPT ON,
/TSK IS CHECKED TO SKIP AND NOT TO SKIP.
/*****

```

```

4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251

```

```

TEST4, LOOPPC          /SETUP TEST NUMBER FOR ERROR DISPLAY
      TAD (4
      DCA TSTNU
      CLREIF          /CLEAR EXPECTING INTERRUPT FLAGS,
      SLUCAF          /CLEAR ALL FLAGS BUT SET SLU INTERRUPT ENABLE
      ION             /TURN THE INTERRUPT ON
      MTSF          /SKIP ON XMIT FLAG SET AND ENABLED
      SKP
      ERROR        /XMIT FLAG ENABLE SET AFTER A CAF
      MTSK         /SKIP ON XMIT/RECEIVE AND INT ENA ON A 1
      SKP
      ERROR        /TSK SKIPPED WITH INT ENA SET AND NO FLAG
      SETEXT        /SET EXPECTING XMIT INTERRUPT FLAG
      MSPF          /SET THE TRANSMIT FLAG ENABLE
      CLREXIF       /CLEAR EXPECTING XMIT INTERRUPT FLAG
      MTSF          /SKIP ON THE TRANSMIT FLAG SET AND ENABLED
      ERROR        /TFL FAILED TO SET THE XMIT FLAG
      MTSK         /SKIP ON XMIT FLAG AND INT ENA ON A 1
      ERROR        /CAF FAILED TO SET SLU INT ENA OR TSK DIDN'T SKIP
      ISZ INTFLG    /DID THE PROGRAM INTERRUPT WITH XMIT + INT ENA
      ERROR        /PROGRAM FAILED TO INTERRUPT WITH XMIT + INT ENA SET
      CLA           /CLEAR THE ACCUMULATOR

```

```

4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306

```

```

MKIE           /CLEAR INT ENA ON SLU
DCA INTFLG     /CLEAR PROGRAM INTERRUPT FLAG,
ION           /TURN THE INTERRUPT BACK ON
MTSF          /SKIP ON TRANSMIT FLAG SET AND ENABLED
ERROR        /XMIT FLAG EN GOT CLEARED
MTSK         /SKIP ON XMIT AND INT ENA ON A 1
SKP
ERROR        /KIE AND DATA 11 FAILED TO CLEAR INT ENA
MTCF         /CLEAR XMIT FLAG ENABLE
MTSF          /SKIP ON TRANSMIT FLAG SET AND ENABLED
SKP
ERROR        /TCF FAILED TO CLEAR XMIT FLAG
MKSF         /SKIP ON RECEIVE FLAG
SKP
ERROR        /RECEIVE FLAG GOT SET BY ABOVE CODE
DONLDP

```

```

/*****
/TEST 5 = CHECKS THAT SLU INT ENA CAN BE SET AND CLEARD BY KIE
/AND DATA BIT 11 USING THE XMIT FLAG TO INTERRUPT ON,
/*****

```

```

TEST5, LOOPPC          /SETUP TEST NUMBER FOR ERROR DISPLAY
      TAD (5
      DCA TSTNU
      CLREIF          /CLEAR EXPECTING INTERRUPT FLAGS,
      SLUCAF          /CLEAR ALL FLAGS
      MKIE           /CLEAR SLU INTERRUPT ENABLE
      ION           /TURN THE INTERRUPT ON
      MSPF          /SET THE TRANSMIT FLAG ENABLE
      MTSF          /SKIP ON TRANSMIT FLAG SET AND EN
      ERROR        /SPF FAILED TO SET TRANSMIT FLAG
      MTSK         /SKIP ON XMIT/RECEIVE + INT ENA ON A 1
      SKP CLA
      ERROR        /ERROR, INT ENA SET OR KIE FAILED TO CLEAR INT ENA
      SETEXT        /SET EXPECTING TRANSMIT INT FLAG
      CLA CLL IAC   /SET DATA 11 TO A 1
      MKIE           /SET INT ENA
      CLREXIF       /CLEAR EXP XMIT INT FLAG
      MTSF          /SKIP ON TRANSMIT FLAG SET AND ENABLED
      ERROR        /XMIT FLAG GOT CLEARED
      MTSK         /SKIP ON XMIT + INT ENA ON A 1
      ERROR        /KIE AND DATA 11 ON A 1 FAILED TO SET INT ENA
      CLA
      ISZ INTFLG    /PROGRAM FAILED TO INTERRUPT WITH INT ENA + XMIT FLAG
      ERROR
      DCA INTFLG
      MKIE           /CLEAR INTERRUPT ENABLE
      ION           /TURN THE INTERRUPT ON
      MTSF          /SKIP ON XMIT FLAG SET AND ENABLED
      ERROR        /XMIT FLAG CLEARED
      MTSK         /SKIP ON XMIT + INT ENA ON A 1
      SKP CLA
      ERROR        /KIE + DATA 11 ON A 0 FAILED TO CLEAR INT ENA
      MTCF         /CLEAR THE XMIT FLAG ENABLE

```

```

4307 0534 4523 MTSF /SKIP ON SLU XMIT FLAG SET AND ENABLED
4309 0535 7610 SKP CLA /TCF FAILED TO CLEAR XMIT FLAG
4309 0536 4450 ERROR /SKIP ON RECEIVE FLAG
4310 0537 4475 MTSF
4311 0540 7610 SKP CLA
4312 0541 4450 ERROR /RECEIVE FLAG SET BY ABOVE CODE
4313 0542 4447 DONLOP
4314
4315 0575 0005
4316 0576 0004
4317 0577 0003

```

PAGE

```

4318
4319
4320 /*****
4321 /TEST 6 = CHECKS THAT TLS WILL CLEAR THE XMIT FLAG AND THEN SET IT WHEN
4322 /XMISSION OF CHAR COMPLETE, THE PROGRAM THEN CLEARS THE XMIT FLAG AND WAITS FOR
4323 /RCV DATA TO SET RECEIVE FLAG, THE RECEIVE FLAG IS CHECKED TO CAUSE KSF TO
4324 /SKIP AND INTERRUPT AND THEN TO CLEAR BY KCF,
4325 /*****
4326
4327 0600 1377 PTEST6, TAD (MTLS
4328 0601 3252 DCA TS60W1
4329 0602 1377 TAD (MTLS
4330 0603 3263 DCA TS60W2
4331 0604 1376 TAD (6
4332 0605 3140 DCA TSTNU /SETUP TEST NUMBER FOR ERROR DISPLAY
4333 0606 4446 TEST6, LOUOPC
4334 0607 1775 TAD BORATE /IS BAUD RATE SET = 19200 BITS/SEC ?
4335 0610 0374 AND (0017
4336 0611 7640 SZA CLA
4337 0612 5224 JMP RES60W /NO-DO FULL INTERRUPT TEST
4338 0613 1373 TAD (NOP /YES=REPLACE SOME OF THE IONS WITH NOPS
4339 /OTHERWISE THE SECOND RCV FLAG WILL GET
4340 /LOST DUE TO PROGRAM TIMING
4341 0614 3264 DCA TS60W3
4342 0615 1373 TAD (NOP
4343 0616 3274 DCA TS60W4
4344 0617 1373 TAD (NOP
4345 0620 3341 DCA TS60W5
4346 0621 1373 TAD (NOP
4347 0622 3347 DCA TS60W6
4348 0623 5235 JMP OWSET
4349 0624 1372 RES60W, TAD (ION /RESTORE TEST6 TO NORMAL
4350 0625 3264 DCA TS60W3
4351 0626 1371 TAD (ERROR
4352 0627 3274 DCA TS60W4
4353 0630 1372 TAD (ION
4354 0631 3341 DCA TS60W5
4355 0632 1371 TAD (ERROR
4356 0633 3347 DCA TS60W6
4357 0634 5235 JMP OWSET
4358 0635 4455 OWSET, CLREIF /CLEAR EXPECTING INTERRUPT FLAGS,
4359 0636 4512 CFSLU /CLEAR ALL FLAGS BUT SET SLU INT ENA
4360 0637 4467 LOOP /SETUP LOOPAROUND ON ALL SLUS.

```

```

4361 0640 3136 DCA ECON61 /INT, CANNOT BE ATTRIBUTED TO CONSOLE PACK REQ
4362 / WHILE LOOPED,
4363 0641 6001 ION /TURN THE INTERRUPT ON
4364 0642 4464 SETEXI /SET EXP XMIT INT FLAG
4365 0643 4502 MSPF /SET THE TRANSMIT FLAG EN
4366 0644 4503 MTSF /SKIP ON XMIT FLAG SET AND EN
4367 0645 4450 ERROR /TRANSMIT FLAG FAILED TO SET BY SPF
4368 0646 4507 MTSK /SKIP ON XMIT FLAG AND INT ENA ON A 1
4369 0647 4450 ERROR /TSK FAILED TO SKIP WITH INT ENA + FLAG SET
4370 0650 2117 ISZ INTFLG /DID THE PROGRAM INTERRUPT
4371 0651 4450 EPROR /PROGRAM FAILED TO INTERRUPT WITH XMIT AND INT ENA SET
4372 0652 4510 TS60W1, MTL5/MTPC /LOAD TRANSMIT BUFFER, TRANSMIT, CLEAR XMIT FLAG, SET XMIT EN
4373 0653 6021 ION /TURN THE INTERRUPT ON
4374 0654 4503 MTSF /NOTE, XMIT FUNCTION IN UART IS BUFFERED,
4375 / THE FIRST TLS WILL BRING THE XMIT FLAG BACK
4376 / UP VERY QUICKLY, THE SECOND TLS MUST WAIT FOR THE
4377 / FIRST CHARACTER TO BE XMITTED BEFORE
4378 / BRINGING UP XMIT FLAG,
4379 0655 7610 SKP CLA
4380 0656 5261 JMP ,+3
4381 0657 4451 TSPWAT
4382 0660 4350 EPROR /NO XMIT FLAG
4383 0661 2117 ISZ INTFLG /ERROR, NO INTERRUPT OCCURRED
4384 0662 4450 EPROR
4385 0663 4510 TS60W2, MTL5/MTPC
4386 0664 6001 TS60W3, ION/NOP
4387 0665 4503 MTSF
4388 0666 7610 SKP CLA /SKIP ON THE TRANSMIT FLAG
4389 0667 4450 EPROR /TLS FAILED TO CLEAR XMIT FLAG
4390 0670 4461 SETERI /SET EXPECTING RECEIVE INTERRUPT FLAG
4391 /NOTE, FIRST REC FLAG MAY SET BEFORE SECOND XMIT FLAG
4392 0671 4451 TSPWAT /WAIT FOR TRANSMIT FLAG TO SET
4393 0672 4450 ERROR /TIMEOUT, UART FAILED TO SET XMIT FLAG
4394 0673 2117 ISZ INTFLG /DID THE PROGRAM INTERRUPT?
4395 0674 4450 ERROR/NOP /ERROR, NO INTERRUPT WITH XMIT AND INT ENA SET
4396 0675 4507 MTSK /SKIP ON XMIT FLAG SET & EN +INT ENA SET,
4397 0676 4450 ERROR /FAILED TO SKIP OR INT ENA OR FLAG GOT CLEARED
4398 0677 4504 WTCF /CLEAR TRANSMIT FLAG ENABLE
4399 0700 4456 CLREXIT /CLEAR EXPECTING TRANSMIT INTERRUPT FLAG
4400 0701 6001 TS60W5, ION/NOP /TURN THE INTERRUPT ON
4401 0702 4452 KSPWAT /WAIT FOR THE RECEIVE FLAG TO SET
4402 0703 4450 EPROR /NO SKIP, OR RECEIVE FLAG NOT SET BY RCD DATA AVAILABLE
4403 0704 4507 MTSK /SKIP ON RCV FLAG AND INT ENA
4404 0705 4456 ERROR
4405 0706 2117 ISZ INTFLG /DID RCV AND INT ENA CAUSE AN INTERRUPT?
4406 0707 4450 TS60W6, ERROR/NOP /NO, ERROR
4407 0710 4474 MKCF /CLEAR RECEIVE FLAG
4408 0711 6001 ION /TURN THE INTERRUPT ON
4409 0712 4452 KSPWAT /WAIT FOR SECOND CHAR TO BE RECEIVED
4410 0713 4450 ERROR
4411 0714 4474 MKCF /CLEAR RECEIVE FLAG
4412 0715 6001 ION
4413 0716 4475 MTSF /SKIP ON RECEIVE FLAG
4414 0717 7610 SKP CLA
4415 0720 4450 ERROR /KCF FAILED TO CLEAR RECEIVE FLAG

```

```

4416 0721 4503      MTSF          /SKIP ON XMIT FLAG
4417 0722 7610      SKP CLA
4418 0723 4450      ERROR        /TRANSMIT FLAG GOT RESET BY ABOVE CODE
4419 0724 4447      DONLOP

```

```

/*****
/TEST 7 = CHECKS THAT TPC PRODUCES SAME RESULTS AS TLS.
/REEXECUTE TEST 6 WITH TLS REPLACED WITH TPC.
*****/

```

```

4425          0725 1370  TEST7,  TAD (MTPC
4426          0726 3252      DCA TSGOW1
4427          0727 1370      TAD (MTPC
4428          0728 3263      DCA TSGOW2
4429          0730 3263      TAD (7
4430          0731 1367      DCA TSTNU
4431          0732 3140      JMP TEST6
4432          0733 5206

```

```

4433
4434
4435          0767 0007
4436          0770 4506
4437          0771 4450
4438          0772 0001
4439          0773 7000
4440          0774 0017
4441          0775 3124
4442          0776 0006
4443          0777 4512
4444          1000

```

PAGE

```

4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
/*****
/TEST 10 - CHECKS THAT THE TLS-TCF SEQUENCE WILL CLEAR XMIT FLAG ENABLE
/PREVENTING THE FOLLOWING INTERRUPT, ALSO CHECKS THAT THE RECEIVE FLAG
/WILL SET AND THAT IT CAN BE CLEARED BY KCC.
*****/

```

```

4451          1000 4446  TEST10, LOOPPC          /SETUP TEST NUMBER FOR ERROR DISPLAY
4452          1001 1377      TAD (10
4453          1002 3140      DCA TSTNU
4454          1003 4455      CLRIF
4455          1004 4511      SLUCAF
4456          1005 4467      LOOP
4457          1006 0001      ION
4458          1007 4460      SETEXI
4459          1010 4502      MSPF
4460          1011 4456      CLRFXI
4461          1012 4503      MTSF
4462          1013 4450      ERROR
4463          1014 2117      ISZ INTFLG
4464          1015 4450      ERROR
4465          1016 4510      MTL5
4466          1017 4504      MTCF
4467          1020 4461      SETERI
4468          1021 0001      ION
4469          1022 4523      MTSF

```

```

4470          1023 7410      SKP
4471          1024 4450      ERROR
4472          1025 4451      TSWFAT
4473          1026 7410      SKP
4474          1027 4450      ERROR
4475          1030 4452      KSWFAT
4476          1031 4450      ERROR
4477          1032 4457      CLRERI
4478          1033 4507      MTSK
4479          1034 4450      ERROR
4480          1035 2117      ISZ INTFLG
4481          1036 4450      EPROR
4482          1037 4476      MKCC
4483          1040 0001      ION
4484          1041 4475      MKSF
4485          1042 7610      SKP CLA
4486          1043 4450      ERROR
4487          1044 4447      DONLOP

```

```

/*****
/TEST 11 - CHECKS THAT KRB WILL CLEAR THE RCV FLAG, THE RCV FLAG
/IS SET BY ISSUING TLS COMMAND.
*****/

```

```

4496          1045 4446  TEST11, LOOPPC          /SETUP TEST NUMBER FOR ERROR DISPLAY
4497          1046 1376      TAD (11
4498          1047 3140      DCA TSTNU
4499          1050 4455      CLRIF
4500          1051 4511      SLUCAF
4501          1052 4467      LOOP
4502          1053 0001      ION
4503          1054 7000      NOP
4504          1055 4460      SETEXI
4505          1056 4510      MTL5
4506          1057 4451      TSWFAT
4507          1060 4450      ERROR
4508          1061 2117      ISZ INTFLG
4509          1062 4450      EPROR
4510          1063 4524      MTCF
4511          1064 4456      CLRFXI
4512          1065 4461      SETERI
4513          1066 0001      ION
4514          1067 4452      KSWFAT
4515          1070 4450      ERROR
4516          1071 4457      CLRERI
4517          1072 4477      MKRS
4518          1073 4475      MKSF
4519          1074 4450      ERROR
4520          1075 2117      ISZ INTFLG
4521          1076 4450      ERROR
4522          1077 4501      MKRB
4523          1100 0001      ION
4524          1101 4475      MTSF

```

```

4525 1102 7610 SKP CLA
4526 1103 4450 ERROR /KRB FAILED TO CLEAR RECEIVE FLAG
4527 1104 4447 DONLOP
4528
4529
4530
4531 /*****
4532 /TEST 12 - CHECKS THAT CAF WILL CLEAR RCV FLAG
4533 /*****
4533 1105 4446 TEST12, LOOPPC
4534 1106 1375 TAD (12 /SETUP TEST NUMBER FOR ERROR DISPLAY
4535 1107 3140 DCA TSTNU
4536 1110 4455 CLRWF /CLEAR EXPECTING INTERRUPT FLAGS
4537 1111 4511 SLUCAF /CLEAR ALL FLAGS
4538 1112 4467 LOOP /SETUP LOOPAROUND ON ALL SLU'S
4539 1113 6001 ION /TURN THE INTERRUPT ON
4540 1114 7000 NOP
4541 1115 4460 SETEXI /SET EXPECTING TRANSMIT INTERRUPT FLAG
4542 1116 4510 MTL5 /TRANSMIT,SET XMIT EN,CLEAR XMIT FLAG
4543 1117 4451 TSWAT /WAIT FOR XMIT FLAG
4544 1120 4450 ERROR /XMIT FLAG FAILED TO SET
4545 1121 4456 CLRWF /CLEAR EXPECTING TRANSMIT INTERRUPT FLAG
4546 1122 2117 ISZ INTFLG /DID THE PROGRAM INTERRUPT?
4547 1123 4450 ERROR /PROGRAM FAILED TO INTERRUPT
4548 1124 4504 MTCF /CLEAR TRANSMIT FLAG EN
4549 1125 4461 SETERI /SET EXPECTING RECEIVE INTERRUPT FLAG
4550 1126 6001 ION
4551 1127 4452 KSWAT /WAIT FOR RECEIVE FLAG
4552 1130 4450 ERROR /RECEIVE FLAG FAILED TO SET
4553 1131 4457 CLRERI /CLEAR EXPECTING RECEIVE INTERRUPT
4554 1132 2117 ISZ INTFLG /DID THE PROGRAM INTERRUPT
4555 1133 4450 ERROR /PROGRAM FAILED TO INTERRUPT
4556 1134 4511 SLUCAF /CLEAR ALL FLAGS
4557 1135 6001 ION /TURN THE INTERRUPT BACK ON
4558 1136 4475 MKSF /SKIP ON RECEIVE FLAG
4559 1137 7610 SKP CLA
4560 1140 4450 ERROR /INITIALIZE FAILED TO CLEAR RECEIVE FLAG
4561 1141 4447 DONLOP
4562
4563 1175 0012
4564 1176 0011
4565 1177 0010
4566 PAGE
4567
4568
4569 /*****
4570 /TEST 13 - CHECKS THE EFFECT OF THE SLU IOT'S UPON THE AC
4571 /*****
4571 1200 4446 TEST13, LOOPPC
4573 1201 1377 TAD (13 /SETUP TEST NUMBER FOR ERR DISPLAY
4574 1202 3140 DCA TSTNU
4575 1203 4455 CLRWF /CLEAR EXPECTING INTERRUPT FLAGS
4576 1204 4511 SLUCAF /CLEAR ALL FLAGS
4577 1205 4467 LOOP /SETUP LOOPAROUND ON SLU'S
4578 1206 6001 ION /TURN THE INTERRUPT ON

```

```

4579 1207 7144 CLA CLL CMA RAL /SET THE AC TO -2
4580 1210 4500 MKIE /CLEAR SLU INTERRUPT ENABLE
4581 1211 7050 CMA RAR
4582 1212 7420 SNL
4583 1213 4450 ERROR /KIE CHANGED THE AC
4584 1214 7240 CLA CMA
4585 1215 4476 MKCC /CLEAR RECEIVE FLAG AND AC
4586 1216 7440 SZA
4587 1217 4450 ERROR /KCC FAILED TO CLEAR THE AC
4588 1220 7240 CLA CMA
4589 1221 4501 MKRB /CLEAR AC AND READ RECEIVE BUFFER
4590 1222 7510 SPA
4591 1223 4450 ERROR /KRB FAILED TO CLEAR AC
4592 1224 7240 CLA CMA
4593 1225 4477 MKRS /READ RECEIVE BUFFER - INCLUSIVE OR WITH AC
4594 1226 7040 CMA /SET THE AC BACK TO 0
4595 1227 7440 SZA
4596 1230 4450 ERROR /KRS CHANGED THE AC
4597 1231 7340 CLA CLL CMA
4598 1232 4475 MKSF /SKIP ON RECEIVE FLAG
4599 1233 7040 CMA
4600 1234 7440 SZA
4601 1235 4450 ERROR /KSF CHANGED THE AC
4602 1236 7240 CLA CMA
4603 1237 4474 MKCF /CLEAR RECEIVE FLAG
4604 1240 7040 CMA
4605 1241 7440 SZA
4606 1242 4450 ERROR /KCF CHANGED THE AC
4607 1243 7240 CLA CMA
4608 1244 4502 MSPF /SET TRANSMIT FLAG ENABLE
4609 1245 7040 CMA
4610 1246 7440 SZA
4611 1247 4450 ERROR /TFL CHANGED THE AC
4612 1250 7240 CLA CMA
4613 1251 4504 MTCF /CLEAR THE TRANSMIT FLAG EN
4614 1252 7040 CMA
4615 1253 7440 SZA
4616 1254 4450 ERROR /TCF CHANGED THE AC
4617 1255 7240 CLA CMA
4618 1256 4503 MTSF /SKIP ON TRANSMIT FLAG SET AND ENABLED
4619 1257 7040 CMA
4620 1260 7440 SZA
4621 1261 4450 ERROR /TSF CHANGED THE AC
4622 1262 7240 CLA CMA
4623 1263 4506 MTPC /LOAD TRANSMIT BUFFER AND TRANSMIT
4624 1264 7040 CMA
4625 1265 7440 SZA
4626 1266 4450 ERROR /TPC CHANGED THE AC
4627 1267 4451 ISFWAT /WAIT FOR THE TRANSMIT FLAG
4628 1270 4450 ERROR /TRANSMIT FLAG FAILED TO SET
4629 1271 4452 KSWAT /WAIT FOR THE RECEIVE FLAG
4630 1272 4450 ERROR /RECEIVE FLAG FAILED TO SET
4631 1273 4504 MTCF /CLEAR THE XMIT FLAG EN
4632 1274 4474 MKCF /CLEAR THE RECEIVE FLAG
4633 1275 7240 CLA CMA

```

4634 1276 4537
 4635 1277 7040
 4636 1303 7440
 4637 1301 4450
 4638 1302 7240
 4639 1303 4510
 4640 1304 7040
 4641 1305 7440
 4642 1306 4450
 4643 1307 4451
 4644 1310 4450
 4645 1311 4452
 4646 1312 4450
 4647 1313 4504
 4648 1314 4476
 4649 1315 4447

MTRK
 CMA
 SZA
 ERKOR
 CLA CMA
 MTL5
 CMA
 SZA
 ERROR
 TSWAT
 ERROR
 KSWAT
 ERROR
 MTCF
 MKCC
 DONLOP

/SKIP IF XMIT/RCV FLAG SET AND INT ENA SET
 /TSK CHANGED THE AC
 /LOAD TRANSMIT BUFFER, TRANSMIT + CLEAR FLAG, SET XMIT EN
 /TLS CHANGED THE AC
 /WAIT FOR THE TRANSMIT FLAG
 /TRANSMIT FLAG FAILED TO SET
 /WAIT FOR THE RECEIVE FLAG TO SET
 /ERROR RECEIVE FLAG FAILED TO SET
 /CLEAR THE TRANSMIT FLAG
 /CLEAR AC AND RECEIVE FLAG

 /TEST 14 = CHECKS THAT ALL ZEROES CAN BE TRANSMITTED AND READ BACK IN

4654
 4655 1316 4446
 4656 1317 1376
 4657 1320 3140
 4658 1321 4455
 4659 1322 4512
 4660 1323 3126
 4661 1324 4453
 4662 1325 4454
 4663 1326 4447

TEST14, LOOPPC
 TAD (14
 DCA TSTNU
 CLRWLF
 CFSLU
 DCA SLUXMT
 SLUDAT
 SLUDER
 DONLOP

/SETUP TEST NUMBER FOR ERROR DISPLAY
 /CLEAR EXPECTING INT FLAGS
 /CLEAR ALL SLU FLAGS AND EN SLU INT.
 /CLEAR THE WORD TO BE TRANSMITTED
 /GO TRANSMIT, READ AND COMPARE THE WORD
 /DATA ERROR=WORD WAS NON ZERO BEING READ BACK

 /TEST 15 = CHECKS THAT ALL ONES CAN BE TRANSMITTED AND READ BACK

4664
 4665
 4666
 4667
 4668
 4669
 4670
 4671 1327 4446
 4672 1330 4775
 4673 1331 1374
 4674 1332 3140
 4675 1333 4455
 4676 1334 4512
 4677 1335 1373
 4678 1336 3126
 4679 1337 4453
 4680 1340 4454
 4681 1341 4447

TEST15, LOOPPC
 JMS APT
 TAD (15
 DCA TSTNU
 CLRWLF
 CFSLU
 TAD (1377
 DCA SLUXMT
 SLUDAT
 SLUDER
 DONLOP

/CHECK FOR APT CONTROL AND OK REPORT TO APT
 /SETUP TEST NUMBER FOR ERROR DISPLAY
 /CLEAR EXPECTING INT FLAGS
 /CLEAR ALL SLU FLAGS AND EN SLU INT.
 /SET THE WORD TO BE TRANSMITTED TO ALL ONE'S
 /GO TRANSMIT, READ AND COMPARE
 /DATA ERROR = WORDS DO NOT COMPARE

 /TEST 16 = CHECKS THAT A COMPLEMENTING PATTERN (252-125) CAN BE
 /TRANSMITTED AND READ BACK.

4689
 4690 1342 4446
 4691 1343 1372
 4692 1344 3140
 4693 1345 4455
 4694 1346 4512
 4695 1347 1371
 4696 1350 3126
 4697 1351 4453
 4698 1352 4454
 4699 1353 1370
 4700 1354 3126
 4701 1355 4453
 4702 1356 4454
 4703 1357 4447
 4704
 4705 1370 0125
 4706 1371 0252
 4707 1372 0510
 4708 1373 0377
 4709 1374 0015
 4710 1375 2513
 4711 1376 0014
 4712 1377 0013
 1400

TEST16, LOOPPC
 TAD (16
 DCA TSTNU
 CLRWLF
 CFSLU
 TAD (252
 DCA SLUXMT
 SLUDAT
 SLUDER
 TAD (125
 DCA SLUXMT
 SLUDAT
 SLUDER
 DONLOP

/SETUP TEST NUMBER FOR ERROR DISPLAY
 /CLEAR EXPECTING INT FLAGS
 /CLEAR ALL SLU FLAGS AND EN SLU INT.
 /SET THE TRANSMIT WORD TO 252
 /TRANSMIT, READ AND COMPARE THE WORD
 /DATA ERROR = TRANSMITTED A 252
 /SET TRANSMIT WORD TO 125
 /TRANSMIT, READ AND COMPARE THE WORD
 /DATA ERROR = TRANSMITTED A 125

PAGE

 /TEST 17 = CHECKS FOR LOADING AND READING A ONE THROUGH A FIELD
 / OF ZEROS AND A ZERO THROUGH A FIELD OF ONES PATTERNS.

4713
 4714
 4715
 4716
 4717
 4718
 4719 1400 4446
 4720 1401 1377
 4721 1402 3140
 4722 1403 4455
 4723 1404 4512
 4724 1405 7301
 4725 1406 3126
 4726 1407 4453
 4727 1410 4454
 4728 1411 1126
 4729 1412 0376
 4730 1413 7640
 4731 1414 5220
 4732 1415 1126
 4733 1416 7104
 4734 1417 5206
 4735 1420 7144
 4736 1421 0375
 4737 1422 3126
 4738 1423 4453
 4739 1424 4454
 4740 1425 1126
 4741 1426 0376
 4742 1427 7650

TEST17, LOOPPC
 TAD (17
 DCA TSTNU
 CLRWLF
 CFSLU
 OTZ,
 CLA CLL IAC
 DCA SLUXMT
 SLUDAT
 SLUDER
 TAD SLUXMT
 AND (0200
 SZA CLA
 JMP ZTO
 TAD SLUXMT
 CLL RAL
 JMP OTZ+1
 ZTO,
 CLL CMA RAL
 AND (0377
 DCA SLUXMT
 SLUDAT
 SLUDER
 TAD SLUXMT
 AND (0200
 SNA CLA

/SETUP TEST NUMBER FOR ERROR DISPLAY
 /CLEAR EXPECTING INTERRUPT FLAGS
 /CLEAR ALL SLU FLAGS AND EN SLU INT.

4743 1430 5235
4744 1431 1176
4745 1432 7120
4746 1433 7004
4747 1434 5221
4748 1435 4447
4749
4750
4751

JMP DONT17
TAD SLUXMT
STL
RAL
JMF ZTO+1
DONT17, DONLOP

/*
/TEST 20 - CHECKS THE TIMING OF THE SERIAL LINE UNIT FROM 50 BAUD TO
/ 19200 BAUD.
/*

4752
4753
4754
4755
4756
4757 1436 4446
4758 1437 1374
4759 1440 3140
4760 1441 6002
4761 1442 4455
4762 1443 4433
4763 1444 0027
4764 1445 4514
4765 1446 4512
4766 1447 4141
4767 1450 1514
4768 1451 3266
4769 1452 3267
4770 1453 4510
4771 1454 4503
4772 1455 5254
4773 1456 4510
4774 1457 6001
4775 1460 2266
4776 1461 5260
4777 1462 2267
4778 1463 5260
4779 1464 4450
4780 1465 5347
4781
4782 1466 0000
4783 1467 0000
4784 1470 0000
4785 1471 0000
4786
4787 1472 0000
4788
4789 1473 0000
4790
4791
4792 1474 1266
4793 1475 7421
4794 1476 1267
4795 1477 4450
4796 1500 7300
4797 1501 1672

TEST20, LOOPPC
TAD (20 /SETUP TEST NUMBER FOR ERROR DISPLAY
DCA TSTNU
IOF
CLREIF /CLEAR EXPECTING INTERRUPT FLAGS
SLU2MC /SET SLU # 2 MODE
MODE1 /8 BIT, NO PARITY, 1 STOP BIT
RLOOP /REMOVE SLU LOOPAROUND
CFSLU /CLEAR ALL SLU FLAGS AND EN SLU INT.
JMS PATCH /SETUP INTERRUPT LINKAGE TO RETURN TO LOCATION 'INTRTN'
INTRTN
DCA ATIMLS /CLEAR ACTUAL TIME COUNTERS
DCA ATIMMS
MTLS /LOAD TRANSMIT REGISTER
MTSF
JMP -1
MTLS /LOAD TRANSMIT BUFFER
ION /ENABLE XMIT FLAG TO TRIGGER INT
ISZ ATIMLS / WHEN TRANSMISSION OF CHAR IS COMPLETE
JMP -1
ISZ ATIMMS /COUNT TILL INTERRUPT
JMP -3 /TIMEOUT-EXPECTED INT DID NOT OCCUR
ERROR
JMP CLUPEX
ATIMLS, 0 /ACTUAL TIME COUNT, LEAST SIGN WORD
ATIMMS, 0 / MOST SIGN WORD
SVBDC, 0
ETIMMS, 0 /ADDRESS OF EXPECTED TIME COUNT
 / MOST SIGNIFICANT WORD
ETIMLL, 0 /ADDRESS OF EXPECTED TIME COUNT
 / LOWER LIMIT ON LEAST SIGN WORD
ETIMLU, 0 /ADDRESS OF EXPECTED TIME COUNT
 / UPPER LIMIT ON LEAST SIGN WORD
TIMERR, TAD ATIMLS /TIMING PROBLEM
MQL
TAD ATIMMS
ERROR /AC,MO=ACTUAL TIME COUNT - CONTINUE (RETURN)
CLA CLL / DIAGNOSTIC FOR EXPECTED TIME COUNT
TAD I ETIMLL

4798 1502 7421
4799 1503 1671
4800 1504 4450
4801 1505 7300
4802 1506 1673
4803 1507 7421
4804 1510 1671
4805 1511 4450
4806 1512 7300
4807 1513 5347
4808 1514 7248
4809 1515 1773
4810 1516 3270
4811 1517 1270
4812 1520 1372
4813 1521 3271
4814 1522 1270
4815 1523 1371
4816 1524 3272
4817 1525 1270
4818 1526 1370
4819 1527 3273
4820 1530 1671
4821 1531 7041
4822 1532 1267
4823 1533 7640
4824 1534 5274
4825 1535 1672
4826 1536 7041
4827 1537 1266
4828 1540 7710
4829 1541 5274
4830 1542 1673
4831 1543 7041
4832 1544 1266
4833 1545 7740
4834 1546 5274
4835 1547 4141
4836 1550 6200
4837 1551 4447
4838
4839
4840 1570 1640
4841 1571 1620
4842 1572 1600
4843 1573 3124
4844 1574 0020
4845 1575 0377
4846 1576 0000
4847 1577 0017
1600
4848
4849
4850 1600 0003
4851 1601 0002

MQL
TAD I ETIMMS
ERROR /AC,MO=EXPECTED TIME COUNT-LOWER LIMIT
CLA CLL / CONTINUE DIAGNOSTIC FOR EXPECTED
TAD I ETIMLU / TIME COUNT UPPER LIMIT.
MQL
TAD I ETIMMS
ERROR /AC,MO=EXPECTED TIME COUNT-UPPER LIMIT
CLA CLL
JMP CLUPEX
INTRTN, CLA CMA /RETURN FROM EXPECTED XMIT INTERRUPT
TAD BDRATE
DCA SVBDC /USE BAUD RATE CONST AS TABLE INDEX
TAD SVBDC
TAD (SLUMST
DCA ETIMMS /POINTER TO EXPECTED TIME
TAD SVBDC
TAD (SLULSL
UCA ETIMLL
TAD SVBDC
TAD (SLULSU
DCA ETIMLU
TAD I ETIMMS /CHECK MOST SIGN WORD OF TIME COUNT
CIA
TAD ATIMMS
SZA CLA
JMP TIMERR
TAD I ETIMLL /CHECK LEAST SIGN WORD OF TIME COUNT
CIA
TAD ATIMLS
SPA CLA
JMP TIMERR
TAD I ETIMLU
CIA
TAD ATIMLS
SMA SZA CLA
JMP TIMERR
CLUPEX, JMS PATCH /RESTORE INTERRUPT LINKAGE AND EXIT
SKP1CHN
DONLOP

PAGE
/SERIAL LINE UNIT TIMING VERIFICATION TABLES

SLUMST, 0003 /BAUD RATE = 50 MOST SIGN COUNT WORD
0002 / 75

```

4852 1602 0001 0001 / 110
4853 1603 0001 0021 / 134,5
4854 1604 0001 0001 / 150
4855 1605 0000 0000 / 300
4856 1606 0000 0000 / 600
4857 1607 0000 0000 / 1200
4858 1610 0000 0000 / 1800
4859 1611 0000 0000 / 2000
4860 1612 0000 0000 / 2400
4861 1613 0000 0000 / 3600
4862 1614 0000 0000 / 4800
4863 1615 0000 0000 / 7200
4864 1616 0000 0000 / 9600
4865 1617 0000 0000 / 19200
4866
4867
4868 1620 1275 SLULSL, 1275 /BAUD RATE = 50 LEAST SIGN COUNT WORD,LOWER LIMIT
4869 1621 0720 0720 / 75
4870 1622 3415 3415 / 110
4871 1623 1330 1330 / 134,5
4872 1624 0345 0345 / 150
4873 1625 4160 4160 / 300
4874 1626 2060 2060 / 600
4875 1627 1025 1025 / 1200
4876 1630 0540 0540 / 1800
4877 1631 0475 0475 / 2000
4878 1632 0410 0410 / 2400
4879 1633 0255 0255 / 3600
4880 1634 0200 0200 / 4800
4881 1635 0124 0124 / 7200
4882 1636 0076 0076 / 9600
4883 1637 0033 0033 / 19200
4884
4885
4886 1640 1302 SLULSU, 1302 /BAUD RATE = 50 LEAST SIGN COUNT WORD,UPPER LIMIT
4887 1641 0720 0720 / 75
4888 1642 3421 3421 / 110
4889 1643 1334 1334 / 134,5
4890 1644 0352 0352 / 150
4891 1645 4163 4163 / 300
4892 1646 2067 2067 / 600
4893 1647 1032 1032 / 1200
4894 1650 0546 0546 / 1800
4895 1651 0502 0502 / 2000
4896 1652 0413 0413 / 2400
4897 1653 0261 0261 / 3600
4898 1654 0203 0203 / 4800
4899 1655 0126 0126 / 7200
4900 1656 0100 0100 / 9600
4901 1657 0035 0035 / 19200
4902
4903
4904 /ROUTINE TO SET MODE FOR SLU #2
4905
4906 1660 0200 XMODE2, 0

```

```

4907 1661 7300 CLA CLL
4908 1662 1660 TAD I XMODE2
4909 1663 6307 KMD1 /GET MODE REQUEST
4910 1664 7300 CLA CLL /ISSUE MODE SETTING IOT
4911 1665 2260 ISZ XMODE2
4912 1666 5660 JMP I XMODE2
4913
4914 /VARIABLE DELAY ROUTINE - DELAY IS APPROX. 15,3 MICRO SEC PER REQUEST COUNT
4915
4916 1667 0000 XVDLY, 0
4917 1670 7300 CLA CLL
4918 1671 1667 TAD I XVDLY
4919 1672 7041 CIA
4920 1673 3300 DCA VDCTR
4921 1674 2267 ISZ XVDLY
4922 1675 2300 ISZ VDCTR
4923 1676 5275 JMP =-1
4924 1677 5667 JMP I XVDLY
4925 1700 0000 VDCTR, 0
4926
4927 /*****
4928 /TEST 21 - VERIFY EIA DRIVERS ON SIGNAL DISTRIBUTION BOARD
4929 / (EXCEPT FOR SLU #1 WHICH IS NOT AVAILABLE ON
4930 / ON AN EXTERNAL CONNECTOR FOR EXTERNAL LOOPAROUND)
4931 /*****
4932
4933 1701 4446 TEST21, LOOPPC
4934 1702 1377 TAD (21
4935 1703 3140 DCA TSTNU /SETUP TEST NUMBER FOR ERROR DISPLAY.
4936 1704 6002 IOF
4937 1705 4466 GETHW3 /GET HARDWARE WORD 3
4938 1706 0376 AND (0200
4939 1707 7650 SNA CLA /PERIPHERAL SIMULATOR ATTACHED?
4940 1710 5330 JMP T21DN /NO-SKIP THIS TEST
4941 1711 1775 TAD IOT0 /SLU #1 BEING TESTED
4942 1712 7041 CIA
4943 1713 1374 TAD (KCF
4944 1714 7650 SNA CLA
4945 1715 5330 JMP T21DN /YES--SKIP THIS TEST
4946 1716 6507 LORE /YES=LOOP SLU'S EXTERNALLY
4947 / VIA SIMULATOR (RESTORE)
4948 1717 4455 CLREIF /CLEAR EXPECTING INTERRUPT FLAGS
4949 1720 4512 CFSLU /CLEAR ALL SLU FLAGS & EN SLU INT.
4950 1721 1373 TAD (0252
4951 1722 3126 DCA SLUXMT /SET UP THE WORD TO BE TRANSMITTED - 252
4952 1723 1372 TAD (DLOOP
4953 1724 3771 DCA DLOP /MODIFY SLUDAT ROUTINE - INHIBIT THE
4954 / INTERNAL LOOPAROUND.
4955 1725 4453 SLUDAT /GO TRANSMIT, READ AND COMPARE
4956 1726 4454 SLUDER /DATA ERROR - FAULTY EIA DRIVER
4957 1727 6007 CAF /REMOVE EXTERNAL LOOPAROUND
4958 1730 4447 T21DN, DONLOP
4959
4960
4961 1771 2423

```

4962 1772 4514
4963 1773 6252
4964 1774 6030
4965 1775 2601
4966 1776 6200
4967 1777 8221
2060

PAGE

4968

4969

4970

4971

4972

/*
/TEST 22 - VERIFY PROGRAMMABLE MODES OF SLU #2 & AND STATUS ERROR BITS
*/

4973 2000 4446
4974 2001 1377
4975 2002 3140
4976 2003 4455
4977 2004 4512
4978 2005 1776
4979 2006 7041
4980 2007 1375
4981 2010 7640
4982 2011 5345
4983 2012 7040
4984 2013 3126
4985 2014 3216
4986 2015 4433
4987 2016 0000
4988 2017 4453
4989 2020 5223
4990 2021 1127
4991 2022 4450
4992
4993 2023 1216
4994 2024 6374
4995 2025 7110
4996 2026 1346
4997 2027 3362
4998 2030 1762
4999 2031 7041
5000 2032 1127
5001 2033 7640
5002 2034 4353
5003 2035 2216
5004 2036 1216
5005 2037 0373
5006 2040 7650
5007 2041 5215
5008 2042 4433
5009 2043 0027
5010 2044 6002
5011 2045 1372
5012 2046 3344
5013 2047 6316
5014 2050 6311
5015 2051 5250

TEST22, LOOPPC
TAD (22 /SETUP TEST NUMBER FOR ERROR DISPLAY
DCA TSTNU
CLKEIF /CLEAR EXPECTING INTERRUPT FLAGS
CFSLU /CLEAR ALL SLU FLAGS AND EN SLU INT,
TAD IOT0 /SLU #2 BEING TESTED?
CIA
TAD (KCF1
SZA CLA
JMP EXS17 /NO-SKIP TEST 17
CMA
DCA SLUXMT /SETUP XMIT WORD TO ALL ONES
DCA VMODE /CLEAR MODE CONTROL WORD
SLUZMC /SET MODE IN SLU#2
0000
SLUDAT /TRANSMIT CHAR AND READ REC'D CHAR
JMP L211 /REC'D CHAR NOT=XMIT WORD (AS EXPECTED)
TAD SLUREC
ERROR /REC'D CHAR CONTAINS 12 ONES - ALL FOUR ERROR
/ BITS ARE SET, AC=RECEIVED CHARACTER
L211, TAD VMODE
AND (0000 /ISOLATE CHAR LENGTH BITS IN VMODE
CLL RAR
TAD LENBAS
DCA PTED /SETUP POINTER TO EXPECTED RECV DATA
TAD I PTED
CIA
TAD SLUREC /COMPARE REC'D DATA WITH EXPECTED DATA
SZA CLA
JMS SLULER /NOT EQUAL
ISZ VMODE /EQUAL - WORK ON NEXT MODE
TAD VMODE
AND (0040
SNA CLA
JMP CHARLP
SLUZMC /ALL MODES TESTED
MODE1 /RESTORE SLU #2 TO NORMAL MODE
IOF
TAD (=4 /CHECK FOR SLU#2 OVERRUN ERROR
DCA XLCTR
XNC, TLS1 /XMIT FOUR CHARACTERS BEFORE UNLOADING RECEIVE BUFFER REGISTER
TSF1
JMP ,-1

5016 2052 2344
5017 2053 5247
5018 2054 4452
5019 2055 4450
5020 2056 6306
5021 2057 3127
5022 2060 1127
5023 2061 0371
5024 2062 1370
5025 2063 7640
5026 2064 4330
5027 2065 4470
5028 2066 4512
5029 2067 1427
5030 2070 1367
5031 2071 7640
5032 2072 5345
5033 2073 1374
5034 2074 3276
5035 2075 4433
5036 2076 0000
5037 2077 6316
5038 2100 6311
5039 2101 5300
5040 2102 4473
5041 2103 0010
5042 2104 6037
5043 2105 4473
5044 2106 0004
5045 2107 7201
5046 2110 6037
5047 2111 4452
5048 2112 4450
5049 2113 6306
5050 2114 3127
5051 2115 1127
5052 2116 0371
5053 2117 1366
5054 2120 7640
5055 2121 4335
5056 2122 1276
5057 2123 1365
5058 2124 7642
5059 2125 5345
5060 2126 2276
5061 2127 5275
5062
5063 2130 0000
5064 2131 1127
5065 2132 4450
5066 2133 7300
5067 2134 5730
5068
5069 2135 0000
5070 2136 1276

ISZ XLCTR
JMP XNC
KSPWAT /FOUR CHAR HAVE BEEN XMITTED-WAIT FOR REC FLAG
ERROR /NO RECEIVE FLAG
KRBI /CLEAR AC AND REC FLAG AND READ BUFFER
DCA SLUREC /SAVE REC'D STATUS
TAD SLUREC
AND (7400 /MASK OFF STATUS BITS
TAD (-4400 /EXPECTED OE=4400
SZA CLA
JMS OERR
DELAY /DELAY AND THEN CLEANUP RECEIVER
CFSLU /CLEAR ALL SLU FLAGS AND EN SLU INT.
TAD BOR2 /SKIP PARITY CHECK IF BAUD RATE IS NOT 9600.
TAD (-16 /{DELAYS ARE SET FOR 9600 BAUD RATE ONLY}
SZA CLA
JMP EXS17
TAD (MODE2 /CHECK FOR PARITY ERROR USING ODD PARITY MODE
DCA PMODE
SLUZMC
PMODE, 0000
TLS1 /XMIT AN ALL ZERO CHAR
TSF1
JMP ,-1
VDELAY /PRE-DELAY LET FIRST FEW BITS OF CHAR BE XMITTED
0010
KLB /REMOVE LOOPAROUND LONG ENOUGH TO MESS UP REC DATA AND
VDELAY / CAUSE A PARITY ERROR,
0004
CLA IAC /RESTORE LOOPAROUND
KLB
KSPWAT /WAIT FOR RECEIVE FLAG
ERROR /NO RECEIVE FLAG
KRBI /CLEAR AC AND REC FLAG AND READ BUFFER
DCA SLUREC /SAVE REC'D STATUS
TAD SLUREC
AND (7400 /MASK OFF STATUS BITS
TAD (-6000 /EXPECTED PE=6000
SZA CLA
JMS PERR
TAD PMODE
TAD (-MODE2
SZA CLA
JMP EXS17
ISZ PMODE /PERFORM SAME PARITY ERROR CHECK USING EVEN PARITY MODE
JMP PMODE=1
OERR, 0
TAD SLUREC
ERROR /AC=REC'D DATA AND STATUS-OVERRUN SHOULD HAVE OCCURRED
CLA CLL
JMP I OERR
PERR, 0
TAD PMODE


```

5071 2137 7421 MQL
5072 2140 1127 TAD SLURFC
5073 2141 4450 ERROR
5074 /AC,MQ=REC'D DATA AND STATUS,MODE
5075 2142 7300 CLA CLL /
5076 2143 5735 JMP I PERR PARITY ERROR SHOULD HAVE OCCURRED
5077 2144 0000 XLCTR, 0
5078 2145 4447 EXB17, DONLOP
5079
5080 2146 2147 LENBAS, ,+1
5081 2147 0037 /EXPECTED DATA - CHAR LENGTH= 5 BITS
5082 2150 0077 /
5083 2151 0177 / 6
5084 2152 0377 / 7
5085 / 8
5086
5087 2153 0000 SLULR, 0 /ERROR DISPLAY ROUTINE
5088 2154 1127 TAD SLUKEC
5089 2155 7421 MQL
5090 2156 1216 TAD VMODE
5091 2157 4450 ERROR
5092 /AC,MQ=MODE,REC'D DATA
5093 /EXPECTED DATA=ALL 1'S FOR CHAR LENGTH
5094 2160 7300 CLA CLL /DESIGNATED BY MODE
5095 2161 5753 JMP I SLULR /NOTE:AC BITS 0,1,2,3 ARE THE SLU2 ERROR BITS
5096 2162 0000 PTED, 0
5097

```

```

5098 2165 7772
5099 2166 2000
5100 2167 7762
5101 2170 3400
5102 2171 7400
5103 2172 7774
5104 2173 0000
5105 2174 0000
5106 2175 6300
5107 2176 2601
5108 2177 0022
5109 2200
5110 PAGE
5111 /ROUTINE TO SETUP # OF PASSES/TEST AND TO STORE THE RETURN ADDRESS FOR SCOPE LOOPING
5112 2200 0000 PCLOOP, 0
5113 2201 7300 CLA CLL
5114 2202 1377 TAD (SLUEND
5115 2203 7041 CIA
5116 2204 1200 TAD PCLOOP
5117 2205 7710 SPA CLA
5118 2206 5211 JMP NUOK
5119 2207 2140 ISZ TSTNU
5120 2210 7000 /UPDATE TEST NUMBER PROVIDE SLU TESTING IS NOT BEING DONE
5121 2211 1700 NOP
5122 2212 3120 NUOK, TAD PCLOOP
5123 2213 5600 DCA TESTF1
5124 JMP I PCLOOP
5125
5126 /EXPECTED INTERRUPT ROUTINES
5127 2214 0000 EXICLR, 0 /CLEAR EXPECTING TRANSMIT INT FLAG
5128 2215 7300 CLA CLL
5129 2216 3120 DCA EXMITI
5130 2217 5614 JMP I EXICLR
5131
5132 2220 0000 ERICLR, 0 /CLEAR EXPECTING RECEIVE INTERRUPT FLAG
5133 2221 7300 CLA CLL
5134 2222 3121 DCA ERECI
5135 2223 5626 JMP I ERICLR
5136
5137 2224 0000 EXISRT, 0 /SET EXPECTING XMIT INTERRUPT FLAG
5138 2225 7240 CLA CMA
5139 2226 3120 DCA EXMITI
5140 2227 5624 JMP I EXISRT
5141
5142 2230 0000 ERISRT, 0 /SET EXPECTING RECV INT FLAG
5143 2231 7240 CLA CMA
5144 2232 3121 DCA ERECI
5145 2233 5630 JMP I ERISRT
5146
5147 2234 0000 EIFCLR, 0 /CLEAR ALL EXPECTING INT FLAGS
5148 2235 7300 CLA CLL
5149 2236 3120 DCA EXMITI
5150 2237 3121 DCA ERECI
5151 2240 3117 DCA INTFLG /CLEAR PROGRAM INT FLAG

```

```

5152 2241 5634      JMP I EIFCLR
5153
5154
5155      /DONE TEST ROUTINE
5156
5157 2242 0000      LOPDON, 0
5158 2243 7300      CLA CLL
5159 2244 1757      TAD LLLD          /SLU TESTING IN PROGRESS?
5160 2245 7041      CIA
5161 2246 1267      TAD KJILD
5162 2247 7640      SZA CLA
5163 2250 5253      JMP GSR          /YES
5164 2251 4462      KBCHK          /CHECK FOR KEYBOARD INTERVENTION
5165 2252 5253      JMP GSR
5166 2253 4465      GSR, GETSR /CHECK FOR LOOP ON TEST
5167 2254 7006      RTL
5168 2255 7710      SPL CLA          /LOOP?
5169 2256 5524      JMP I TESTF1     /YES, LOOP ON THIS TEST
5170 2257 2265      LLLD, ISZ TESTAB/JMP I LOPDON
5171
5172
5173
5174 2260 1665      TAD I TESTAB     /LOCATION LLLD CONTAINS 'ISZ TESTAB' WHEN EXECUTING SLU TESTS
5175 2261 7450      SNA
5176 2262 5271      JMP CKBRSG      /AND 'JMP I LOPDON' WHEN EXECUTING PARALLEL I/O
5177 2263 3266      DCA TJLOC       /AND FLOPPY TESTS.
5178 2264 5666      JMP I TJLOC      /ANY TESTS REMAINING?
5179 2265 0000      TESTAB, 0
5180 2266 0000      TJLOC, 0
5181 2267 5642      KJILD, JMP I LOPDON
5182 2270 2265      KISZIT, ISZ TESTAB
5183
5184 2271 4776      CKBRSG, JMS XBAUD /SETUP FOR NEXT BAUD RATE
5185 2272 5775      JMP NXTDC       /ALL BAUD RATES TESTED, DO NEXT SLU
5186 2273 1374      TAD (BDRSQ-1)  /MORE BAUD RATES YET TO BE TESTED
5187
5188 2274 3265      DCA TESTAB     /SETUP TESTAB WITH BEGINNING OF BAUD RATE TEST SEQ TABLE
5189 2275 5257      JMP LLLD
5190
5191      /NORMAL TEST SEQUENCE
5192      /BAUD RATE FIXED=(NSQBDP)
5193
5194 2276 0221      NORMSQ, TEST1
5195 2277 0241      TEST2
5196 2300 0400      TEST3
5197 2301 0425      TEST4
5198 2302 0473      TEST5
5199 2303 0600      PTEST6
5200 2304 0725      TEST7
5201 2305 1000      TEST10
5202 2306 1045      TEST11
5203 2307 1105      TEST12
5204 2310 1200      TEST13
5205 2311 1316      TEST14
5206 2312 1327      TEST15
    
```

```

5207 2313 1342      TEST16
5208 2314 1400      TEST17
5209 2315 1701      TEST21
5210 2316 2000      TEST22
5211 2317 0000      0
5212 2320 0000      0
5213 2321 0000      0
5214
5215
5216      /VARIABLE BAUD RATE TEST SEQUENCE
5217      /TESTS LISTED HERE ARE REPEATED FOR EACH BAUD RATE POSSIBLE
5218
5219 2322 0600      BORSQ, PTEST6
5220 2323 1316      TEST14
5221 2324 1327      TEST15
5222 2325 1342      TEST16
5223 2326 1430      TEST20
5224 2327 0000      0
5225 2330 0000      0
5226 2331 0000      0
5227 2332 0000      0
5228
5229
5230      /ROUTINE TO WAIT FOR SERIAL LINE UNITS XMIT FLAG
5231
5232 2333 0000      WATTSF, 0
5233 2334 7300      CLA CLL
5234 2335 4500      MKIE          /CLEAR INTERRUPT ENABLE F/F
5235 2336 1373      TAD (7777)
5236 2337 3123      DCA CNT1
5237 2340 3122      DCA CNT
5238 2341 4772      JMS OFONKI     /CHECK TO SEE IF INT EN
5239 2342 4503      MTSF
5240 2343 4357      JMS ADDTIM     /GO ADD ONE TO THE COUNTER
5241 2344 2333      ISZ WATTSF
5242 2345 5733      JMP I WATTSF   /RETURN TO THE PROGRAM-GOT THE FLAG
5243
5244
5245      /ROUTINE TO WAIT FOR THE SERIAL LINE UNIT RECEIVE FLAG
5246
5247 2346 0000      WATKSF, 0
5248 2347 7300      CLA CLL
5249 2350 1373      TAD (7777)
5250 2351 3123      DCA CNT1
5251 2352 3122      DCA CNT
5252 2353 4475      MKSF
5253 2354 4357      JMS ADDTIM     /SKIP ON SLU RECEIVE FLAG
5254 2355 2346      ISZ WATKSF    /GO ADD A ONE TO THE COUNTER
5255 2356 5746      JMP I WATKSF   /RETURN TO THE PROGRAM-GOT THE FLAG
5256
5257
5258      /ROUTINE TO WAIT FOR THE FLAG
5259 2357 0000      ADDTIM, 0
5260 2360 2122      ISZ CNT       /EACH INCREMENT ON CNT = APP. 67,5 MICROSEC
5261 2361 7610      SKP CLA
    
```

5262 2362 2123
 5263 2363 7346
 5264 2364 7001
 5265 2365 1357
 5266 2366 3357
 5267 2367 5757
 5268
 5269
 5270 2372 2471
 5271 2373 7777
 5272 2374 2321
 5273 2375 0214
 5274 2376 3112
 5275 2377 3331
 2400

ISZ CNT1
 CLA CLL CHA RTL
 IAC
 TAD ADDTIM
 DCA ADUTIM
 JMP I ADDTIM

/EACH INCREMENT ON CNT1 = APP. 216 MILLISEC

PAGE

5276
 5277
 5278
 5279 2400 0000
 5280 2401 6201
 5281 2402 1616
 5282 2403 3217
 5283 2404 6724
 5284 2405 1155
 5285 2406 3207
 5286 2407 7402
 5287 2410 1600
 5288 2411 3220
 5289 2412 2200
 5290 2413 1217
 5291 2414 3620
 5292 2415 5600
 5293
 5294 2416 0000
 5295 2417 0000
 5296 2420 0000
 5297
 5298
 5299

/THIS ROUTINE SETS UP A RETURN ADDRESS FOR INTERRUPT RETURNS FROM ANOTHER FIELD

RETURN, 0
 CDF /CHANGE DATA FIELD TO FIELD 0
 TAD I K0 /GET THE INTERRUPT PC
 DCA RETIADD /SAVE IT
 RIF /READ THE PROGRAM INSTRUCTION FIELD
 TAD KCDF /ADD A CDF INSTRUCTION TO IT
 DCA ,+1 /SAVE IT IN THE NEXT LOCATION
 HLT/CDF /RETURN TO THE PROGRAM DATA FIELD
 TAD I RETURN /GET THE INTERRUPT RETURN LOCATION
 DCA SAVLOC /SAVE IT
 ISZ RETURN
 TAD RETIADD
 DCA I SAVLOC
 JMP I RETURN

K0, 0
 RETIADD, 0
 SAVLOC, 0

/ROUTINE FOR TRANSMITTING, READING AND COMPARING DATA FOR SLU

5300 2421 0000
 5301 2422 7300
 5302 2423 4467
 5303 2424 1377
 5304
 5305 2425 3223
 5306 2426 6001
 5307 2427 3117
 5308 2430 4460
 5309 2431 1126
 5310 2432 4510
 5311 2433 4451
 5312 2434 4450
 5313 2435 4450
 5314 2436 2117
 5315 2437 4450

DATSLU, 0
 CIA CLL
 DLOP, LOOP /SETUP LOOPAROUND IN SLU'S
 TAD (LOOP /SETUP LOOP INST. IN CASE EIA DRIVER TEST
 / (TEST 21) HAS CHANGED IT TO A NOP
 DCA DLOP
 ION /TURN THE INTERRUPT ON
 DCA INTFLG /CLEAR THE PROGRAM INTERRUPT FLAG
 SETEXI /SET EXPECTING TRANSMIT INT FLAG
 TAD SLUXMT /GET THE WORD TO BE TRANSMITTED
 MTL5 /LOAD AND TRANSMIT IT AND CLEAR THE FLAG
 ISFWAT /WAIT FOR THE TRANSMIT FLAG
 ERROR /XMIT FLAG FAILED TO SET
 CLREXI /CLEAR EXP XMIT INT FLAG
 ISZ INTFLG /DID THE PROGRAM INTERRUPT?
 ERROR /PROGRAM FAILED TO INTERRUPT

5316 2440 4504
 5317 2441 4461
 5318 2442 6001
 5319 2443 4452
 5320 2444 4450
 5321 2445 4457
 5322 2446 2117
 5323 2447 4450
 5324 2450 4501
 5325 2451 3127
 5326 2452 6001
 5327 2453 1126
 5328 2454 7041
 5329 2455 1127
 5330 2456 7640
 5331 2457 5621
 5332 2460 2221
 5333 2461 5621
 5334
 5335
 5336
 5337
 5338
 5339 2462 0000
 5340 2463 7240
 5341 2464 1262
 5342 2465 3270
 5343 2466 4450
 5344 2467 5662
 5345 2470 0000
 5346
 5347
 5348 2471 0000
 5349 2472 1124
 5350 2473 1376
 5351 2474 7640
 5352 2475 7301
 5353 2476 1375
 5354 2477 4500
 5355
 5356 2500 7200
 5357 2501 5671
 5358
 5359
 5360
 5361 2502 0000
 5362 2503 7300
 5363 2504 6201
 5364 2505 1774
 5365 2506 6211
 5366 2507 7700
 5367 2510 5702
 5368 2511 4773
 5369 2512 5702
 5370

MTCF /CLEAR THE XMIT FLAG EN
 SETERI /SET EXP RECV INT FLAG
 ION /TURN THE INTERRUPT BACK ON
 KSFWAT /WAIT FOR THE RECEIVE FLAG TO SET
 ERROR /RECEIVE FLAG FAILED TO SET
 CLRERI /CLEAR EXP RECV INT FLAG
 ISZ INTFLG /DID THE RECEIVE FLAG CAUSE A INTERRUPT
 ERROR /RECEIVE FLAG FAILED TO CAUSE A INTERRUPT
 MKRB /CLEAR THE AC AND RCV FLAG AND READ BUFFER
 DCA SLUREC /SAVE THE WORD READ BACK
 ION /TURN THE INTERRUPT BACK ON
 TAD SLUXMT /GET THE WORD TRANSMITTED
 CIA
 TAD SLUREC /GET THE WORD READ BACK
 SEA CLA
 JMP I DATSLU /DATA ERROR--RETURN TO REPORT THE ERROR
 ISZ DATSLU /BUMP RETURN ADDRESS POINTER BY ONE
 JMP I DATSLU /RETURN TO TEST

/DATA ERROR ROUTINE FOR SERIAL LINE UNIT

DERSLU, 0
 CLA CMA
 TAD DERSLU /
 DCA DDERSLU /SAVE ADDRESS WHERE ERROR WAS DETECTED
 SPE3, ERROR /WORD XMITTED NOT= WORD RECEIVED
 JMP I DERSLU /NO, RETURN TO TEST
 DDERSLU, 0

OFOVKI, 0
 TAD TESTF1 /ROUTINE TO TURN INT ENA OFF OR ON
 TAD (-TEST13=1 /GET THE TEST BEING EXECUTED
 SEA CLA
 CLA CLL IAC /NO, SET THE AC TO 0001
 TAD (2 /
 MKIE /ENABLE OR DISABLE SLU INT ENA
 /ENABLE SLU 2 STATUS BITS
 CLA /CLEAR THE AC BIT IF SET
 JMP I OFVKI /RETURN

/ROUTINE TO CHECK IF UNDER APT CONTROL--IF SO REPORT OK STATUS TO APT.

REPAPT, 0
 CLA CLL
 CDF 00
 TAD I (HCW2 /UNDER APT CONTROL?
 CDF 10
 SMA CLA
 JMP I REPAPT /NO
 JMS APTOK1 /YES--REPORT OK TO APT
 JMP I REPAPT

```

5371
5372 /REPORT TO APT DURING VARIABLE BAUD RATE SEQUENCE OF TESTING WHEN
5373 / BAUD RATE FOR TEST REACHES A PRESET VALUE.
5374
5375 2513 0000 APTP, 0
5376 2514 1772 TAD BDRATE /GET CUPRENT SETTING(+1)
5377 2515 0371 AND (0003) /LOOK AT ONLY 2 LEAST SIGNIF, BITS
5378 2516 7041 CIA
5379 2517 1323 TAD RPIHR /AGREE WITH PRESET VALUE?
5380 2520 7650 SNA CLA
5381 2521 4515 APTPEP /YES--REPORT TO APT
5382 2522 5713 JMF I APTR
5383 2523 0002 RPTRR, 0002
5384
5385
5386
5387 2571 0003
5388 2572 3124
5389 2573 6725
5390 2574 0022
5391 2575 0002
5392 2576 6577
5393 2577 4467
2600 PAGE
5394 /*****
5395 /SUBROUTINES TO ISSUE SLU IOTS
5396 /*****
5397
5398 2600 0000 XMKCF, 0
5399 2601 6030 IOT0, KCF /CLEAR RECEIVE FLAG
5400 2602 5600 JMP I XMKCF
5401 2603 4450 ERROR /KCF SKIPPED
5402 2604 5600 JMP I XMKCF
5403
5404 2605 0000 XMKSF, 0
5405 2606 6031 IOT1, KSF /SKIP ON RECEIVE FLAG
5406 2607 5605 JMP I XMKSF /RECV FLAG NOT SET
5407 2610 2205 ISZ XMKSF
5408 2611 5005 JMP I XMKSF /RECV FLAG SET - SKIP OCCURRED
5409
5410 2612 0000 XMKCC, 0
5411 2613 6032 IOT2, KCC /CLEAR REC V FLAG & AC
5412 2614 5612 JMP I XMKCC
5413 2615 4450 ERROR /KCC SKIPPED
5414 2616 5612 JMP I XMKCC
5415
5416 2617 0000 XMKRS, 0
5417 2620 6034 IOT3, KRS /'OR' CONTENTS OF REC V BUFFER INTO AC
5418 2621 5617 JMP I XMKRS
5419 2622 4450 ERROR /KRS SKIPPED
5420 2623 5617 JMP I XMKRS
5421
5422 2624 0000 XMKIE, 0
5423 2625 6035 IOT4, KIE /IF AC11=1 SET INT EN
5424 /IF AC11=0 CLR INT EN

```

```

5425 2626 5624 JMP I XMKIE
5426 2627 4450 ERROR /KIE SKIPPED
5427 2630 5624 JMP I XMKIE
5428
5429 2631 0000 XMKRB, 0
5430 2632 6036 IOT5, KRB /LOAD CONTENT OF REC BUFFER INTO AC,CLR REC FLAG
5431 2633 5631 JMP I XMKRB
5432 2634 4450 ERROR /KRB SKIPPED
5433 2635 5631 JMP I XMKRB
5434
5435 2636 0000 XMSPF, 0
5436 2637 6040 IOT7, SPF /SET XMIT FLAG ENABLE
5437 2640 5636 JMP I XMSPF
5438 2641 4450 ERROR /SPF SKIPPED
5439 2642 5636 JMP I XMSPF
5440
5441 2643 0000 XMTSF, 0
5442 2644 6041 IOT8, TSF /SKIP ON XMIT FLAG SET AND ENABLED
5443 2645 5643 JMP I XMTSF /XMIT FLAG NOT SET
5444 2646 2243 ISZ XMTSF
5445 2647 5643 JMP I XMTSF /XMIT FLAG SET - SKIP OCCURRED
5446
5447 2650 0000 XMTCF, 0
5448 2651 6042 IOT9, TCF /CLEAR XMIT FLAG ENABLE
5449 2652 5650 JMP I XMTCF
5450 2653 4450 ERROR /TCF SKIPPED
5451 2654 5650 JMP I XMTCF
5452
5453 2655 0000 XMTSB, 0
5454 2656 1302 TAD IOT10 /DETERMINE WHICH SLU IS BEING WORKED ON.
5455 2657 0377 AND (0030)
5456 2660 7450 SNA
5457 2661 5275 JMP SDR1
5458 2662 1376 TAD (-10)
5459 2663 7650 SNA CLA
5460 2664 5771 JMP SDR2
5461 2665 1775 SDR3, TAD BDRATE /SLU3 IS BEING WORKED ON.
5462 2666 0374 AND (0017)
5463 2667 3030 DCA BDR3 /UPDATE BAUD RATE INDICATOR FOR SLU#3
5464 2670 5300 JMP ISSTSB
5465 2671 1775 SDR2, TAD BDRATE /SLU#2 IS BEING WORKED ON.
5466 2672 0374 AND (0017)
5467 2673 3027 DCA BDR2 /UPDATE BAUD RATE INDICATOR FOR SLU#2
5468 2674 5300 JMP ISSTSB
5469 2675 1775 SDR1, TAD BDRATE /SLU#1 IS BEING WORKED ON.
5470 2676 0374 AND (0017)
5471 2677 3026 DCA BDR1 /UPDATE BAUD RATE INDICATOR FOR SLU#1
5472 2700 1775 ISSTSB, TAD BDRATE
5473 2701 0374 AND (0017)
5474 2702 6043 IOT10, TSB /SET BAUD RATE
5475 2703 7300 CLA CLL
5476 2704 5655 JMP I XMTSB
5477 2705 4450 ERROR /TSB SKIPPED
5478 2706 5655 JMP I XMTSB
5479

```

```

5480 2707 0000 XMTPC, 0
5481 2710 6044 IOT11, TPC /SAME AS TLS
5482 2711 5707 JMP I XMTPC
5483 2712 4454 ERROR /TPC SKIPPED
5484 2713 5707 JMP I XMTPC
5485
5486 2714 0000 XMTSK, 0
5487 2715 6045 IOT12, TSK /SKIP ON INT EN & REC FLAG OR INT EN & XMIT FLAG & XMIT EN
5488 2716 5714 JMP I XMTSK /NO SKIP
5489 2717 2314 ISZ XMTSK
5490 2720 5714 JMP I XMTSK /SKIP OCCURRED
5491
5492 2721 0000 XMTLS, 0
5493 2722 6046 IOT13, TLS /LOAD XMIT BUFFER,XMIT,EN XMIT FLAG
5494 2723 5721 JMP I XMTLS
5495 2724 4450 ERROR /TLS SKIPPED
5496 2725 5721 JMP I XMTLS
5497
5498 /DON'T TEST SLU #3 IF RUNNING UNDER APT CONTROL - CAN'T LOOP AROUND
5499 / SLU#3 WHEN RUNNING UNDER APT.
5500
5501 2726 0000 APT3CK, 0
5502 2727 1773 TAD TABADD /GET REC DEV CODE FROM TABLE
5503 2730 3346 DCA STAB
5504 2731 1746 TAD I STAB
5505 2732 0372 AND (0770
5506 2733 1371 TAD (-320
5507 2734 7640 SZA CLA /NEXT SLU TO BE TESTED = 3?
5508 2735 5726 JMP I APT3CK /NO
5509 2736 6201 CDF 00 /YES
5510 2737 1770 TAD I (MCW2 /RUNNING UNDER APT CONTROL?
5511 2740 6211 CDF 10
5512 2741 7700 SMA CLA
5513 2742 5726 JMP I APT3CK /NO
5514 2743 2773 ISZ TABADD /YES=SKIP TESTING OF THIS SLU
5515 2744 2773 ISZ TABADD
5516 2745 5726 JMP I APT3CK
5517
5518 2746 0000 STAB, 0
5519
5520
5521 2770 0022
5522 2771 7460
5523 2772 0770
5524 2773 3055
5525 2774 6017
5526 2775 3124
5527 2776 7770
5528 2777 0030
5529
5530 PAGE
5531 /SETUP IOTS FOR PARTICULAR SLU UNDER TEST
5532
5533 3000 0000 XIOT, 0
5534 3001 6002 IOF
5535 3002 4514 RLOOP /CLEAR LOOP AROUND.

```

```

5534 3003 4777 JMS APT3CK /CHECK FOR APT CONTROL AND SLU UNDER TEST=#3,
5535 3004 1655 TAD I TABADD /GET REC DEV CODE FROM TABLE - IF ANY
5536 3005 7450 SNA /WAS IT A DEVICE CODE
5537 3006 5331 JMP ENDSLX /ALL SLUS TESTED
5538 3007 0376 AND (770 /MASK IN CASE ANY UNWANTED BITS SET
5539 3010 4463 MIOT /GO MODIFY RECV IOTS
5540 3011 3056 IOTREC /REC IOT TABLE ADDRESS
5541 3012 2255 ISZ TABADD
5542 3013 1655 TAD I TABADD /GET THE XMIT DEVICE CODE
5543 3014 7450 SNA
5544 3015 5331 JMP ENDSLX /NO XMIT DEV CODE - IGNORE REC CODE
5545 3016 0376 AND (770 /MASK
5546 3017 4463 MIOT /MODIFY ODD IOTS
5547 3020 3066 IOTXMT /XMT IOT TABLE ADDRESS
5548 3021 2255 ISZ TABADD
5549 3022 7000 NOP
5550 3023 5600 JMP I XIOT
5551
5552 /MODIFY DEVICE CODE FOR A LIST OF IOT ADDRESSES
5553
5554 3024 0000 XMLOT, 0
5555 3025 3025 DCA SDEVC /SAVE DEVICE CODE
5556 3026 1624 TAD I XMLOT
5557 3027 2224 ISZ XMLOT
5558 3030 3243 DCA IOTTA /SAVE TABLE ADDRESS
5559 3031 1643 TAD I IOTTA /GET NEXT IOT ADDRESS
5560 3032 7450 SNA /END OF LIST? (ZERO)
5561 3033 5624 JMP I XMLOT /YES
5562 3034 3244 DCA IOTADD /SAVE NEXT ADDRESS
5563 3035 1644 TAD I IOTADD /GET IOT
5564 3036 0375 AND (7007 /REMOVE OLD DEVICE CODE
5565 3037 1025 TAD SDEVC /ADD NEW DEVICE CODE
5566 3040 3644 DCA I IOTADD /PUT BACK IOT
5567 3041 2243 ISZ IOTTA
5568 3042 5231 JMP ,-11
5569 3043 0000 IOTTA, 0
5570 3044 0000 IOTADD, 0
5571
5572
5573 3045 0030 DVCTAB, 0030
5574 3046 0040 0040
5575 3047 0300 0300
5576 3050 0310 0310
5577 3051 0320 0320
5578 3052 0330 0330
5579 3053 0000 0000
5580 3054 0000 0000
5581
5582 3055 0000 TABADD, 0
5583
5584 3056 2601 IOTREC, IOT0
5585 3057 2600 IOT1
5586 3060 2613 IOT2
5587 3061 2620 IOT3
5588 3062 2625 IOT4

```

```

5589 3063 2632 IOT5
5590 3064 2627 CKRFG
5591 3065 2620 0
5592
5593 3066 2637 IOTXMT, IOT7
5594 3067 2644 IOT8
5595 3070 2651 IOT9
5596 3071 2702 IOT10
5597 3072 2710 IOT11
5598 3073 2715 IOT12
5599 3074 2722 IOT13
5600 3075 2624 CXXMFG
5601 3076 2620 0
5602
5603
5604 /INITIALIZE FOR STARTUP
5605
5606 3077 2600 INIT1, 0
5607 3100 7300 CIA CLL
5608 3101 3324 DCA BDRATE /INIT TEST BAUD RATE TO 50 BITS/SEC
5609 3102 1374 TAD (DVCTAB) /INIT TABADD TO START OF DEVICE CODE TABLE
5610 3103 3255 DCA TABADD
5611 3104 1373 TAD (NORMSQ-1)
5612 3105 3772 DCA TESTAB /INITIALIZE TESTAB TO NORMAL TEST SEQUENCE,
5613 3106 1771 TAD KISZTT /SETUP DONLOP ROUTINE FOR SLU TESTING
5614 3107 3770 DCA LLLD
5615 3110 3100 DCA TSTNU /CLEAR TEST NUMBER USED FOR ERROR DISPLAY
5616 3111 5677 JMP I INIT1
5617
5618
5619
5620 /SET BAUD RATE IN SLU UNDER TEST
5621 3112 2600 XBAUD, 0
5622 3113 1324 TAD BDRATE /GET BAUD RATE CONSTANT
5623 3114 2607 AND (2020) /MASK
5624 3115 7640 SZA CIA /BAUD RATE SEQ 50-19200 COMPLETE?
5625 3116 5325 JMP LLXB /YES - WORK ON NEXT SLU.
5626 3117 4535 MTSB /NO-SET BAUD RATE IN SLU UNDER TEST
5627 3120 2324 ISZ BDRATE /GET SETUP FOR NEXT BAUD RATE LOOP
5628 3121 4470 DELAY /WAIT FOR BAUD RATE CHANGES TO TAKE EFFECT
5629 3122 2312 ISZ XBAUD
5630 3123 5712 JMP I XBAUD
5631 3124 2600 BDRATE, 0
5632
5633 3125 3324 LLXB, DCA BDRATE
5634 3126 1373 TAD (NORMSQ-1)
5635 3127 3772 DCA TESTAB
5636 3130 5712 JMP I XBAUD
5637
5638
5639 /SET BAUD RATES BEFORE LEAVING SLU DIAGNOSTIC; REMOVE LOOPAROUND
5640
5641 3131 4350 ENDSL, JMS SETBDR /SET BAUD RATES=SLU1 & 3 = 9600, SLU2 PER SW
5642 3132 1366 TAD (16)
5643 3133 3026 DCA BDR1

```

```

5644 3134 1366 TAD (16)
5645 3135 3030 DCA BDR3
5646 3136 7604 LAS
5647 3137 4365 AND (2017)
5648 3140 3027 DCA BDR2
5649 3141 6002 IOP
5650 3142 4514 RLOOP /BAUD RATE INDICATORS UPDATED
5651 3143 3025 DCA SDEVIC /DISABLE INT. SYSTEM WHILE REMOVING LOOPAROUND
5652 3144 1764 TAD KJILD /REMOVE LOOPAROUND
5653 3145 3770 DCA LLLD /CLEAR SLU DEVICE CODE
5654 3146 3140 DCA TSTNU /RESTORE DONLOP ROUTINE FOR NON SLU USE,
5655 3147 5763 JMP PRTEST /CLEAR TEST NUMBER
5656 /ENTER PARALLEL I/O DIAGNOSTIC
5657
5658 /SET BAUD RATES IN ALL SLUS; SLU1=9600, SLU2=FROM SWITCH, SLU3=9600.
5659 3150 2600 SETBDR, 0
5660 3151 1366 TAD (16)
5661 3152 6043 TSB /RESTORE SLU#1 BAUD RATE TO 9600
5662 3153 6333 TSB2 /RESTORE SLU#3 BAUD RATE TO 9600
5663 3154 7604 LAS
5664 3155 2607 AND (2017)
5665 3156 6313 TSB1 /RESTORE SLU#2 BAUD RATE FROM BAUD RATE SWITCH
5666 3157 7300 CLA CLL
5667 3160 4470 DELAY /WAIT FOR BAUD RATE CHANGES TO TAKE EFFECT
5668 3161 5750 JMP I SETBDR
5669
5670
5671
5672 3163 3400
5673 3164 2267
5674 3165 2017
5675 3166 3016
5676 3167 2020
5677 3170 2257
5678 3171 2270
5679 3172 2265
5680 3173 2275
5681 3174 3445
5682 3175 7007
5683 3176 2770
5684 3177 2720
5685 PAGE
5686 /SET BAUD RATE IN ALL 3 SLUS TO NORMAL TEST SEQUENCE
5687 / BAUD RATE (PRESET TO 9600)
5688 3200 2600 BDRNS, 0
5689 3201 1024 TAD NSQHDR
5690 3202 6043 TSB
5691 3203 6313 TSB1
5692 3204 6333 TSB2
5693 3205 3026 DCA BDR1
5694 3206 4470 DELAY
5695 3207 1024 TAD NSQHDR
5696 3210 3027 DCA BDR2
5697 3211 1024 TAD NSQHDR

```

```

5698 3212 3334          DCA HDR3
5699 3213 5000          JMF I HDRNS
5700                      /RESTORE BAUD RATES AND LOOPAROUND TO STATES EXPECTED BY DIAGNOSTIC
5701
5702
5703 3214 0000          RESTBR, 0
5704 3215 1026          TAD BDR1
5705 3216 6043          TSB
5706 3217 7300          CLA CLL
5707 3220 1027          TAD BDR2
5708 3221 6313          TSB1
5709 3222 7300          CLA CLL
5710 3223 1030          TAD BDR3
5711 3224 6333          TSB2
5712 3225 7300          CLA CLL
5713 3226 1031          TAD LOOPA
5714 3227 6037          KLB
5715 3230 7300          CLA CLL
5716 3231 4470          DELAY
5717 3232 5614          JMP I RESTBR
5718
5719
5720                      /SETUP LOOPAROUND ON ALL SLU'S
5721
5722 3233 0000          SLOOPA, 0
5723 3234 7301          CLA CLL IAC
5724 3235 6037          KLB
5725 3236 3031          DCA LOOPA
5726 3237 5633          JMP I SLOOPA
5727
5728                      /CLEAR LOOPAROUND ON ALL SLU'S
5729
5730 3240 0000          CLOOPA, 0
5731 3241 7300          CLA CLL
5732 3242 6037          KLB
5733 3243 3031          DCA LOOPA
5734 3244 5640          JMP I CLOOPA
5735
5736
5737 3245 0000          XDELAY, 0
5738 3246 2251          ISZ DELAY1
5739 3247 5246          JMP ,=1
5740 3250 5645          JMP I XDELAY
5741 3251 0000          DELAY1, 0
5742
5743
5744                      /CLEAR ALL FLAGS ROUTINE FOR SLU TESTING.
5745
5746
5747 3252 0000          XSCAF, 0
5748 3253 6007          CAF
5749 3254 7006          NOP/CLA CLL
5750 3255 7000          NOP/KIE
5751 3256 7000          NOP/KIE1
5752 3257 7000          NOP/KIE2

```

```

5753 3260 7000          NOP/IAC
5754 3261 7000          NOP/MKIE
5755                      /ABOVE 6 NOPS CAN BE REPLACED WITH CODE SHOWN TO DISABLE INTERRUPTS FROM THOSE SLUS
5756                      /
5757 3262 7300          CLA CLL
5758 3263 6665          PCIE
5759 3264 5652          JMP I XSCAF
5760
5761
5762                      /CLEAR SLU FLAGS AND ENABLE SLU INTERRUPTS
5763                      / (A SUBSTITUTE FOR CAF -AVOIDS RX01 CLICKING)
5764
5765 3265 0000          XCFSLU, 0
5766 3266 6030          KCF
5767 3267 6042          TCF
5768 3270 6300          KCF1
5769 3271 6312          TCF1
5770 3272 6320          KCF2
5771 3273 6332          TCF2
5772 3274 7300          CLA CLL
5773 3275 1377          TAD (0003
5774 3276 6035          KIE/MKIE
5775 3277 6305          KIE1/NOP
5776 3300 6325          KIE2/NOP
5777                      /THE ABOVE 3 KIE'S CAN BE REPLACED WITH CODE SHOWN TO DISABLE INTERRUPTS FROM
5778                      / THOSE SLUS NOT CURRENTLY UNDER TEST.
5779 3301 7300          CLA CLL
5780 3302 5665          JMP I XCFSLU
5781
5782
5783                      /SLU TESTING MESSAGE
5784 3303 2314          SLUMES, TEXT "SLU "
5785 3304 2540
5786 3305 0000
5787 3306 2405          TESMES, TEXT "TESTING"
5788 3307 2324
5789 3310 1116
5790 3311 0700
5791
5792                      /GET PSEUDO SWITCH REGISTER FROM LOCATION 20 IN FIELD 0
5793
5794 3312 0000          XGETSR, 0
5795 3313 7300          CLA CLL
5796 3314 6201          CDF 00
5797 3315 1776          TAD I (PSR
5798 3316 6211          CDF 10
5799 3317 5712          JMP I XGETSR
5800
5801                      /ROUTINE TO CHECK FOR KEYBOARD INTERVENTION
5802
5803 3320 0000          CHKKB, 0
5804 3321 7300          CLA CLL
5805 3322 6035          KIE
5806 3323 4445          CH1KSF
5807 3324 7000          NOP

```

```

5803 3325 7201          CLA IAC
5804 3326 6235          KIE          /ENABLE INTERRUPTS FROM KEYBOARD
5805 3327 7300          CUA CLL
5806 3330 5770          JMP I CH1KB
5807
5808
5809
5810
5811
5812 3331          /*****
SLUEND=          /END MARKER FOR SLU TESTS-THIS MARKER MUST ALWAYS
5813          /          BE LOCATED AFTER LAST SLU TEST LOOP
5814          /          USED BY LOOPPC & ERROR ROUTINE.
5815
5816
5817          /*****
5818          /WARNING
5819 3347          *3347          /LOCATION DESTROYED BY MEMORY EXTENSION TESTING
5820          /          FROM FIELD 0 (MEM EXT TEST 4)
5821
5822
5823
5824
5825

```

```

5826          /SLU TESTING COMPLETE - DIAGNOSE PARALLEL I/O INTERFACE
5827

```



```

5028
5029 3376 0020
5030 3377 0003
          3400 *3430
5031
5032 /*****
5033 /PARALLEL INTERFACE DIAGNOSTIC
5034 /*****
5035
5036 3400 4465 PRTEST, GETSR /GET PSEUDO SWITCH REGISTER FROM FIELD 0.
5037 3401 0377 AND (0004
5038 3402 7640 SZA CLA /EXECUTE PARALLEL INTERFACE TEST?????
5039 3403 5776 JMP PRETEST /NO
5040 3404 6002 IOF
5041 3405 7240 CLA CMA
5042 3406 3136 DCA ECONS /SET FLAG INDICATING THAT A CONSOLE INTERRUPT IS POSSIBLE NOW.
5043 3407 1775 TAD KJLD
5044 3410 3774 DCA LLLD
5045 3411 4773 PPIM, JMS PPIMES /PRINT PARALLEL I/O INTERF TESTING MESSAGE
5046 /IF NOT UNDER APT CONTROL,
5047 3412 6027 CAF /CLEAR ALL FLAGS & DISABLE PARALLEL I/O SIMULATION
5048 / STROBE DECODER,
5049 3413 4141 JMS PATCH /SETUP INTERRUPT SERVICE LINKAGE
5050 3414 6200 SKP1CHN
5051 3415 5772 JMP SIMCK0 /CHECK FOR SIMULATOR IN USE
5052
5053 /*****
5054 /WARNING
5055 /LOCATION 3422 & LOCATION 3425 GET DESTROYED BY MEM EXT TEST 4
5056 3426 *3426
5057
5058 3426 4466 NOSIM0, GETHW3
5059 3427 0371 AND (1000
5060 3430 7640 SZA CLA
5061 3431 5770 JMP TILQ /NO PARALLEL INTERFACE DEVICE CABLED TO SYSTEM
5062 3432 4466 GETHW3
5063 3433 0367 AND (400
5064 3434 7640 SZA CLA
5065 3435 5770 JMP TILQ
5066
5067 /*****
5068 /PARALLEL I/O DEVICE CODE = 60
5069 /*****
5070 /
5071 /
5072 /*****
5073 /TEST 1 - CHECK FOR PCLF TO CLR PRINT FLAG AND PSF TO SET
5074 / PRINT FLAG, USE PSKF TO VERIFY.
5075 /*****
5076
5077
5078 3436 4440 TILA, LOOPPC
5079 3437 6662 PCLF /CLEAR PRINT FLAG
5080 3440 7410 SKP
5081 3441 4450 ERROR /PCLF SKIPPED

```

```

5082 3442 6661 PSKF /SKIP ON FLAG
5083 3443 7410 SKP
5084 3444 4450 ERROR /EITHER PCLF FAILED TO CLEAR FLAG
5085 / OR PSKF SKIPPED ON NO FLAG,
5086 3445 6660 PSSF /SET PRINT FLAG
5087 3446 7410 SKP
5088 3447 4450 ERROR /PSSF SKIPPED
5089 3450 6661 PSKF /SKIP ON FLAG
5090 3451 4450 ERROR /EITHER PSSF FAILED TO SET FLAG OR
5091 / PSKF FAILED TO SKIP.
5092 3452 6662 PCLF /CLEAR THE KNOWN SET FLAG,
5093 3453 6661 PSKF /SKIP ON FLAG,
5094 3454 7410 SKP
5095 3455 4450 ERROR /PCLF FAILED TO CLEAR FLAG,
5096 3456 4447 DONLOP
5097
5098
5099 /*****
5100 /TEST 2 - CHECK THAT PCIE WILL SET AND CLEAR INTERRUPT ENABLE.
5101 /*****
5102
5103 3457 4446 T2LA, LOOPPC
5104 3460 4471 CLREPI /CLEAR EXPECTING PRINTER INTERRUPTS.
5105 3461 4516 SPCIE /CLEAR PRINTER INT, ENABLE
5106 3462 7410 SKP
5107 3463 4450 ERROR /PCIE SKIPPED.
5108 3464 6001 ION
5109 3465 6660 PSSF /SET PRINTER FLAG
5110 3466 6661 PSKF
5111 3467 4450 ERROR /FLAG FAILED TO SET
5112 3470 6662 PCLF /CLEAR FLAG
5113 3471 6661 PSKF
5114 3472 7410 SKP
5115 3473 4450 ERROR /FLAG FAILED TO CLEAR
5116 3474 7201 CLA IAC
5117 3475 4516 SPCIE /SET INT, ENABLE
5118 3476 4472 SETEPI /SET EXPECTING PRINTER INT.
5119 3477 6660 PSSF /SET PRINTER FLAG
5120 3500 7000 NOP
5121 3501 4471 CLREPI /CLEAR EXXPECTING PRINTER INT,
5122 3502 6661 PSKF
5123 3503 4450 ERROR /FLAG FAILED TO SET OR NO SKIP
5124 3504 2117 IS2 INTFLG
5125 3505 4450 ERROR /PROGRAM FAILED TO INT.
5126 3506 6662 PCLF
5127 3507 6001 ION
5128 3510 6661 PSKF
5129 3511 7410 SKP
5130 3512 4450 ERROR /FLAG FAILED TO CLEAR
5131 3513 4447 DONLOP
5132
5133
5134 /*****
5135 /TEST 3 - CHECK THAT CAF ENABLES PRINTER INTERRUPTS & CLEARS FLAGS
5136 /*****

```

```

5937
5938 3513 4446 T3LA, LOOPPC
5939 3515 4471 CLREPI /CLEAR EXPECTING PRINTER INT.
5940 3516 6662 PCLF /CLEAR PRINT FLAG
5941 3517 4510 SPCIE /DISABLE PRINTER INT.
5942 3522 6407 CAF /ENABLE PRINTER INTERRUPTS & CLEAR FLAGS
5943 /NOTE: CAF CLEARS LIFT RIBBON, IF THE SIMULATOR
5944 / IS IN USE CLEARING LIFT RIBBON SWITCHES THE
5945 / PARALLEL INTERFACE TO THE LQP MODE.
5946 3521 7326 CLA CLL CML RTL /IN CASE SIMULATOR IS IN USE SWITCH MODE BACK TO LA
5947 / & CLEAR FLAGS, THIS LQLS INST. HAS NO EFFECT
5948 / IF SIMULATOR IS NOT IN USE.
5949 3522 6536 LQLS
5950 3523 6662 PCLF
5951 3524 6001 ION
5952 3525 4472 SETEPI /SET EXPECTING PRINTER INT.
5953 3526 6060 PSSF
5954 3527 7000 NOP
5955 3530 4471 CLREPI
5956 3531 6661 PSKF
5957 3532 4450 ERROR /FLAG NOT SET OR NO SKIP
5958 3533 2117 ISZ INTFLG
5959 3534 4450 ERROR /PROGRAM FAILED TO INTERRUPT
5960 3535 6662 PCLF /CLEAR PRINT FLAG
5961 3536 6001 ION
5962 3537 6661 PSKF
5963 3540 7410 SKP
5964 3541 4450 ERROR /FLAG FAILED TO CLEAR
5965 3542 4447 DONLOP
5966 3543 5766 JMP T4LA
5967
5968
5969
5970 /ROUTINE TO ISSUE PCIE AND LEAVE LIFT RIBBON = 0.
5971 / NOTE: LIFT RIBBON = 0 DESIGNATES LA MODE WHEN SIMULATOR IS IN USE.
5972 3544 0000 XSPCIE, 0
5973 3545 1365 TAD (2 /AC BIT 10 CONTROLS LIFT RIBBON STATE UPON PCIE EXECUTION
5974 3546 6665 PCIE
5975 3547 7200 CLA
5976 3550 5744 JMP I XSPCIE
5977 3565 0002
5978 3566 3600
5979 3567 0400
5980 3570 4005
5981 3571 1000
5982 3572 4520
5983 3573 4342
5984 3574 2257
5985 3575 2267
5986 3576 5000
5987 3577 0004
5988 3600 PAGE
5989
5990

```

```

5991 /TEST 4 - CHECK THAT PSTB LOADS AND PRDB READS PRINTER
5992 / INTERFACE BUFFER, CHECK THAT PSTB LEAVES FLAG ALONE.
5993 / VERIFIES JAM INTO AC.
5994 /*****
5995
5996 3600 4446 T4LA, LOOPPC
5997 3601 4471 CLREPI
5998 3602 4516 SPCIE /DISABLE PRINTER INTERRUPTS
5999 3603 6600 PSSF /SET PRINTER FLAG
6000 3604 6664 PSTB /LOAD 0 INTO PRINTER INTERFACE BUFFER
6001 3605 7410 SKP
6002 3606 4450 ERROR /PSTB SKIPPED
6003 3607 6661 PSKF
6004 3610 4450 ERROR /PSTB CLEARED FLAG
6005 3611 7200 STA /READ WORD BACK
6006 3612 6667 PRDB
6007 3613 7410 SKP
6008 3614 4450 ERROR /PRDB SKIPPED
6009 3615 6601 PSKF
6010 3616 4450 ERROR /PRDB CLEARED FLAG
6011 3617 7440 SZA
6012 3620 4450 ERROR /DATA READ NOT = DATA WRITTEN
6013 /AC CONTAINS WORD READ
6014 /WORD WRITTEN=0
6015 3621 6662 PCLF
6016 3622 7200 STA
6017 3623 6664 PSTB /LOAD INTERFACE BUFFER WITH ALL ONES
6018 3624 6661 PSKF
6019 3625 7410 SKP
6020 3626 4450 ERROR /PSTB SET FLAG--NOTE: PARALLEL I/O DEVICE IS POWERED DOWN
6021 3627 7200 CLA /READ WORD BACK
6022 3630 6667 PRDB
6023 3631 6661 PSKF
6024 3632 7410 SKP
6025 3633 4450 ERROR /PRDB SET FLAG--NOTE: PARALLEL I/O DEVICE IS POWERED DOWN
6026 3634 6667 PRDB /READ WORD BACK AGAIN--IN CASE THE PREVIOUS 'ERROR' OCCURRED.
6027 3635 7000 CMA
6028 3636 7440 SZA
6029 3637 4450 ERROR /DATA READ NOT = DATA WRITTEN
6030 /AC CONTAINS COMPLEMENT OF WORD READ
6031 /WORD WRITTEN = 7777
6032 3640 4447 DONLOP
6033
6034
6035 /*****
6036 /TEST 5 - CHECK THAT PCLP LOADS PRINTER INTERFACE BUFFER AND CLEARS FLAG.
6037 /*****
6038
6039 3641 4446 T5LA, LOOPPC
6040 3642 4471 CLREPI
6041 3643 4516 SPCIE /DISABLE PRINTER INTERRUPTS
6042 3644 6001 ION
6043 3645 6660 PSSF /SET FLAG
6044 3646 6666 PCLP /LOAD PRINTER INTERFACE BUFFER
6045 3647 7410 SKP

```

```

6046 3650 4450 ERROR /PCLP SKIPPED
6047 3651 6661 PSKF
6048 3652 7410 SFP
6049 3653 4450 ERROR /PCLP FAILED TO CLEAR FLAG
6050 3654 7240 STA /READ WORD BACK
6051 3655 6667 PRDB
6052 3656 7440 SZA /DATA READ NOT = DATA WRITTEN
6053 3657 4450 ERROR /AC CONTAINS WORD READ
6054 /WORD WRITTEN=0
6055
6056 3660 7240 STA
6057 3661 6666 PCLP /LOAD ALL ONES INTO BUFFER
6058 3662 7200 CLA
6059 3663 6667 PRDB /READ WORD BACK
6060 3664 7040 CMA
6061 3665 7440 SZA
6062 3666 4450 ERROR /DATA READ NOT = DATA WRITTEN
6063 /AC CONTAINS COMPLEMENT OF WROD READ
6064 /WORD WRITTEN=7777
6065 3667 4447 DONLOP
6066
6067
6068 /*****
6069 /TEST 6 - CHECK THAT PSSF,PSKF,PCLF,PSFB,PCIE AND PCLP LEAVE ACC
6070 / UNCHANGED. ALSO CHECK THAT PSSF,PSKF,PCLF,PCIE PRDB
6071 / LEAVE INTERFACE REGISTER UNCHANGED.
6072 /*****
6073
6074 3670 4446 T6LA, LOOPPC
6075 3671 6002 IOF
6076 3672 1377 TAD (5252
6077 3673 6664 PSFB /LOAD INTERFACE BUFFER
6078 3674 1376 TAD (-5252
6079 3675 7440 SZA
6080 3676 4450 ERROR /PSFB CHANGED AC CONTENTS
6081 3677 1377 TAD (5252
6082 3700 6660 PSSF
6083 3701 1376 TAD (-5252
6084 3702 7440 SZA
6085 3703 4450 ERROR /PSSF CHANGED AC CONTENTS
6086 3704 1377 TAD (5252
6087 3705 6661 PSKF
6088 3706 7000 NOP
6089 3707 1376 TAD (-5252
6090 3710 7440 SZA
6091 3711 4450 ERROR /PSKF CHANGED AC CONTENTS
6092 3712 1377 TAD (5252
6093 3713 6665 PCIE
6094 3714 1376 TAD (-5252
6095 3715 7440 SZA
6096 3716 4450 ERROR /PCIE CHANGED AC CONTENTS
6097 3717 1377 TAD (5252
6098 3720 6662 PCLF
6099 3721 1376 TAD (-5252
6100 3722 7440 SZA

```

```

6101 3723 4450 ERROR /PCLF CHANGED AC CONTENTS
6102 3724 6667 PPDB /READ INTERFACE BUFFER
6103 3725 1376 TAD (-5252
6104 3726 7440 SZA
6105 3727 4450 ERROR /INTERFACE BUFFER CHANGED SINCE LOAD BY PSFB ABOVE
6106 3730 6667 PRDB
6107 3731 1376 TAD (-5252
6108 3732 7440 SZA
6109 3733 4450 ERROR /PRDB CHANGED INTEERFACE BUFFER CONTENTS
6110 3734 4447 DONLOP
6111 3735 5775 JMP T7LA
6112
6113 3775 4000
6114 3776 2526
6115 3777 5252
6116 PAGE
6117 /*****
6118 /TEST 7 - CHECK OUT PRINTER INTERFACE BUFFER REGISTER
6119 / (ONE THROUGH A FIELD OF ZEROS AND VICE VERSA)
6120 /*****
6121 4000 1377 T7LA, TAD (NOP
6122 4001 3776 DCA T70W1
6123 4002 1377 TAD (NOP
6124 4003 3775 DCA T70W2
6125 4004 5774 JMP T7LQ
6126
6127
6128
6129
6130
6131
6132 /*****
6133 /PARALLEL I/O DEVICE CODE = 50
6134 /*****
6135
6136
6137
6138 /*****
6139 /TEST 1 - CHECK THAT LORS AND LQRS CLEAR DONE FLAG AND THAT
6140 / LQLS SETS DONE FLAG. CHECK USING LQSK.
6141 /*****
6142
6143 4005 4446 T1LQ, LOOPPC
6144 4006 6002 IOF
6145 4007 6505 LQRS /READ STATUS AND CLEAR DONE FLAG
6146 4010 7410 SKP
6147 4011 4450 ERROR /LQRS SKIPPED
6148 4012 6500 LQSK /SKIP ON DONE FLAG
6149 4013 7410 SKP
6150 4014 4450 ERROR /LQRS FAILED TO CLEAR DONE FLAG OR
6151 / LQSK SKIPPED.
6152 4015 7300 CLA CLL
6153 4016 6500 LQLS /WRITE STATUS (CLEAR LIFT RIBBON & INT EN)
6154 / AND SET DONE FLAG

```

```

6155 4017 7410 SKP
6156 4020 4450 ERROR /LQLS SKIPPED
6157 4021 6520 LQSA /SKIP ON DONE FLAG
6158 4022 4450 ERROR /LQLS FAILED TO SET DONE FLAG
6159 4023 6520 LQFS /CLEAR KNOWN SET DONE FLAG
6160 4024 6520 LQSK /SKIP ON DONE FLAG
6161 4025 7410 SKP
6162 4026 4450 ERROR /LQFS FAILED TO CLEAR DONE FLAG
6163 4027 7300 CLA CLL
6164 4030 6506 LQLS /WRITE STATUS (CLEAR LIFT RIBBON & INT EN,)
6165 / AND SET DONE FLAG
6166 4031 6507 LQRE /RESTORE AND CLEAR DONE FLAG
6167 4032 7410 SKP
6168 4033 4450 ERROR /LQRE SKIPPED
6169 4034 6500 LQSK /SKIP ON DONE FLAG
6170 4035 7410 SKP
6171 4036 4450 ERROR /LQRE FAILED TO CLEAR DONE FLAG
6172 4037 4513 KLSIM /DISABLE PARALLEL I/O SIMULATOR IF IN USE.
6173 4040 4447 DONLOP
6174
6175
6176
6177
6178 /*****
6179 /TEST 2 - CHECK THAT LQLS WRITES STATUS AND LQRS READS STATUS.
6180 / ALSO CHECKS INTERRUPT ENABLE FUNCTION,
6181 /*****
6182 4041 4446 T2LQ, LOOPPC
6183 4042 4471 CLRPEI
6184 4043 6506 LQLS /WRITE STATUS (CLEARS LIFT RIBBON
6185 / AND INT EN) AND SET DONE FLAG.
6186 4044 7040 CMA
6187 4045 6505 LQRS /READ STATUS AND CLEAR DONE FLAG
6188 4046 6001 ION
6189 4047 0373 AND (0003 /MASK OFF ALL BUT LIFT RIBBON AND INT EN BITS
6190 4050 7440 SZA
6191 4051 4450 ERROR /LQLS FAILED TO WRITE STATUS OR LQRS
6192 4052 6506 LQLS / INCORRECTLY READ STATUS.
6193 /WRITE STATUS (CLEAR LIFT RIBBON AND INT EN)(
6194 / AND SET DONE FLAG. CHECK FOR NO INT.
6195 4053 4466 GETHW3 /IF PARALLEL I/O SIMULATOR IS IN USE, SETTING
6196 4054 0372 AND (0200 / LIFT RIBBON WOULD CAUSE A SWITCH TO LA MODE.
6197 4055 7040 SZA CLA
6198 4056 5267 JMP SKPLRC /SKIP THIS STATUS WRITE CHECK WHEN SIMULATOR IS IN USE.
6199 4057 1371 TAD (0002
6200 4060 6506 LQLS /WRITE STATUS (SET LIFT RIBBON, CLEAR INT EN),
6201 / AND SET DONE FLAG
6202 4061 7300 CLA CLL
6203 4062 6505 LQRS /READ STATUS AND CLEAR DONE FLAG
6204 4063 0373 AND (0003
6205 4064 1370 TAD (-0002
6206 4065 7440 SZA
6207 4066 4450 ERROR /STATUS WRITE FAILED
6208 4067 4472 SKPLRC, SETPEI /SET EXPECTING PRINTER INTERRUPT
6209 4070 1367 TAD (0001
6210 4071 6506 LQLS /WRITE STATUS (CLEAR LIFT RIBBON

```

```

6211 4072 7300 CLA CLL / SET INT. EN.) AND SET DONE FLAG
6212 4073 4471 CLRPEI
6213 4074 6505 LQRS /READ STATUS AND CLEAR DONE FLAG
6214 4075 0373 AND (0003
6215 4076 1366 TAD (-0001
6216 4077 7440 SZA
6217 4100 4450 ERROR /STATUS WRITE FAILED
6218 4101 2117 ISZ INTFLG
6219 4102 4450 ERROR /PROGRAM FAILED TO INT. WITH FLAG
6220 / SET AND ENABLED.
6221 4103 6506 LQLS /WRITE STATUS (CLEAR LIFT RIBBON AND INT. EN)
6222 / SET DONE FLAG
6223 4104 6001 ION
6224 4105 6500 LQSK
6225 4106 4450 ERROR /LQLS FAILED TO SET DONE FLAG
6226 4107 6507 LQRE /CLEAR DONE FLAG
6227 4110 4513 KLSIM /DISABLE PARALLEL I/O SIMULATOR IF IN USE.
6228 4111 4447 DONLOP
6229
6230
6231
6232 /*****
6233 /TEST 3 - CHECK THAT CAF DISABLES LQP INTERRUPTS & CLEAR LIFT RIBBON
6234 /*****
6235
6236 4112 4446 T3LQ, LOOPPC
6237 4113 4471 CLRPEI
6238 4114 6002 IOF
6239 4115 1366 TAD (7777
6240 4116 6506 LQLS /WRITE STATUS (SET INT EN & SET LIFT RIBBON)
6241 / SET DONE FLAG
6242 4117 6007 CAF /SHOULD CLEAR INT. EN & LIFT RIBBON
6243 4120 6001 ION
6244 4121 7200 CLA
6245 4122 6505 LQRS /READ STATUS AND CLEAR DONE FLAG
6246 4123 0365 AND (0337 /VERIFY ALSO THAT UNUSED STATUS BITS RETURN AS 0.
6247 4124 7440 SZA
6248 4125 4450 ERROR /CAF FAILED TO CLEAR INT. EN AND/OR LIFT RIBBON.
6249 4126 4447 DONLOP
6250 4127 5764 JMP T4LQ
6251
6252 4164 4200
6253 4165 0337
6254 4166 7777
6255 4167 0001
6256 4170 7776
6257 4171 0002
6258 4172 0200
6259 4173 0003
6260 4174 4400
6261 4175 4446
6262 4176 4421
6263 4177 7000
6264 4200
PAGE

```

```

6264
6265
6266
6267
6268
6269
6270
6271 4200 4446 T4LQ, LOOPPC
6272 4201 4471 CLREPI
6273 4202 1377 TAD (LQ4TAB
6274 4203 3251 DCA LQ4TAB
6275 4204 5232 JMP INITL
6276 4205 6500 INSL1, LQLS /DISABLE LQP INTERRUPTS AND SET DONE FLAG
6277 4206 6001 ION
6278 4207 6502 LQI1, LQMP/LQMC/LQPC/LQRE /AC0=11 TO INTERFACE BUFFER,CLEAR AC, CLEAR DONE FLAG
6279 4210 7410 SKP
6280 4211 4450 ERROR /LQXX SKIPPED-DETERMINE LQXX FROM LOCATION LQI1
6281 4212 6500 LQSK
6282 4213 7410 SKP
6283 4214 4450 ERROR /LQXX FAILED TO CLEAR DONE FLAG
6284 4215 6501 LQRB /READ INTERFACE BUFFER
6285 4216 7410 SKP
6286 4217 4450 ERROR /LQRB SKIPPED
6287 4220 7040 CMA
6288 4221 7440 SZA
6289 4222 4450 ERROR /DATA READ NOT = EXPECTED READ DATA
6290 /AC CONTAINS WORD READ BACK FROM INT BUFFER
6291 /WORD WRITTEN = 0; EXPECTED READ DATA=7777
6292 4223 7240 LQI2, STA
6293 4224 6502 LQMP/LQMC/LQPC/LQRE /AC0=11 TO INTERF. BUFFER,CLEAR AC,CLEAR DONE FLAG
6294 4225 7240 STA
6295 4226 6501 LQRB /READ INTERFACE BUFFER
6296 4227 7440 SZA
6297 4230 4450 ERROR /DATA READ NOT = EXPECTED READ DATA
6298 /AC CONTAINS WORD READ BACK FROM INTERF BUFFER
6299 /WORD WRITTEN=7777; EXPECTED READ DATA = 0000
6300 4231 2251 INITL, ISZ LQ4TAB
6301 4232 1651 TAD I LQ4TAB
6302 4233 7450 SNA
6303 4234 5241 JMP TABDON
6304 4235 3207 DCA LQI1
6305 4236 1651 TAD I LQ4TAB
6306 4237 3224 DCA LQI2
6307 4240 5245 JMP INSL1
6308 4241 7300 TABDON, CLA CLL /CHECK THAT LQRB DOES NOT CLEAR DONE FLAG
6309 4242 4513 KLSIM /DISABLE PARALLEL I/O SIMULATOR IF IN USE.
6310 4243 6500 LOLS /SET DONE FLAG
6311 4244 6501 LQRB
6312 4245 6500 LQSK
6313 4246 4450 ERROR /LQRB CLEARED DONE FLAG
6314 4247 4447 DONLOP
6315 4250 5257 JMP T5LQ
6316
6317 4251 0000 LQ4TAB, 0
6318 4252 6502 LQ4TAB, LQMP

```

```

6319 4253 6503 LQMC
6320 4254 6504 LQPC
6321 4255 6507 LQRE
6322 4256 0000 0
6323
6324
6325
6326
6327
6328
6329
6330
6331 4257 4446 T5LQ, LOOPPC
6332 4260 4471 CLREPI
6333 4261 6002 IOF
6334 4262 1376 TAD (7775 /IN CASE SIMULATOR IS IN USE DON'T SET LIFT
6335 / RIBBON = THIS WOULD SWITCH MODES TO LA.
6336 4263 6500 LOLS
6337 4264 7440 SZA
6338 4265 4450 ERROR /LQLS FAILED TO CLEAR AC
6339 4266 6500 LQLS /DISABLE LQP INTERRUPTS AND SET DONE FLAG
6340 4267 6001 ION
6341 4270 7040 CMA
6342 4271 6502 LQMP
6343 4272 7440 SZA
6344 4273 4450 ERROR /LQMP FAILED TO CLEAR AC
6345 4274 7040 CMA
6346 4275 6503 LQMC
6347 4276 7440 SZA
6348 4277 4450 ERROR /LQMC FAILED TO CLEAR AC
6349 4300 7040 CMA
6350 4301 6504 LQPC
6351 4302 7440 SZA
6352 4303 4450 ERROR /LQPC FAILED TO CLEAR AC
6353 4304 4447 DONLOP
6354
6355
6356
6357
6358
6359
6360
6361
6362 4305 4446 T6LQ, LOOPPC
6363 4306 6002 IOF
6364 4307 1375 TAD (5252 /LOAD INTERFACE BUFFER
6365 4310 6503 LQMC
6366 4311 1375 TAD (5252
6367 4312 6500 LQSK
6368 4313 7000 NOP
6369 4314 1374 TAD (=5252
6370 4315 7440 SZA
6371 4316 4450 ERROR /LQSK CHANGED AC
6372 4317 6505 LQRS
6373 4320 6501 LQRB

```

```

6374 4321 7200 CLA
6375 4322 6521 LQPR /READ INTERFACE BUFFER
6376 4323 1373 TAD (5253
6377 4324 7440 SZA
6378 4325 4450 EPROR /LQSK,LQRB,LQRS OR LQLS CHANGED INTERFACE BUFFER
6379 4326 1375 TAD (5252
6380 4327 6507 LQRE
6381 4330 1374 TAD (-5252
6382 4331 7440 SZA
6383 4332 4450 ERROR /LQRE CHANGED AC
6384 4333 4513 KLSIM /DISABLE PARALLEL I/O SIMULATOR IF IN USE.
6385 4334 4447 DONLOP
6386 4335 1372 TAD (CMA /RESTORE TEST 7 FOR LQ MODE
6387 4336 3771 DCA T70W1
6388 4337 1372 TAD (CMA
6389 4340 3770 DCA T70W2
6390 4341 5767 JMP T7LQ

```

/ROUTINE TO PRINT "PARALLEL I/O INTERFACE TESTING"

```

6391
6392
6393
6394 4342 0000 PPIMES, 0
6395 4343 4432 C1CRLF
6396 4344 4434 C1PRNT
6397 4345 4352 PIMES
6398 4346 4434 C1PRNT
6399 4347 3306 TESMES
6400 4350 4432 C1CRLF
6401 4351 5742 JMP I PPIMES

```

PIMES, TEXT "PARALLEL I/O INTERFACE "

```

6402
6403 4352 2001
6404 4353 2701
6405 4354 1414
6406 4355 0514
6407 4356 4011
6408 4357 5717
6409 4360 4011
6410 4361 1624
6411 4362 0522
6412 4363 0601
6413 4364 0305
6414 4365 4000

```

```

6404
6405
6406 4367 4400
6407 4370 4440
6408 4371 4421
6409 4372 7440
6410 4373 5253
6411 4374 2526
6412 4375 5252
6413 4376 7775
6414 4377 4252
6415 4400
6416

```

PAGE

/*****
/TEST 7 = CHECK OUT PRINTER INTERFACE BUFFER REGISTER (ONE THROUGH A

```

6417 / FIELD OF ZEROS AND VICE VERSA)
6418 /*****
6419
6420 4400 4446 T7LQ, LOOPPC
6421 4401 4471 CLREPI
6422 4402 4516 CLREPI
6423 4403 6506 SPCIE
6424 4404 6001 LQLS /DISABLE PRINTER INTERRUPTS
6425 4405 7301 ION
6426 4406 3273 LL1, CLA CLL IAC /TRY ONE THROUGH A FIELD OF ZEROS
6427 4407 1273 DCA SXMT
6428 4410 7421 TAD SXMT
6429 4411 1273 MQL /LOAD HQ WITH WORD WRITTEN
6430 4412 6666 TAD SXMT
6431 4413 6504 PCLP /LOAD WORD INTO BUFFER
6432 4414 7300 LQFC
6433 4415 7300 CLA CLL
6434 4416 6667 PRDB /READ BUFFER
6435 4417 3274 LQRB
6436 4420 1274 DCA SREC /SAVE ACTUAL WORD READ
6437 4421 7440 TAD SREC
6438 4422 7041 T70W1, CMA/NOP /CMA FOR LQ DEVICE CODE; NOP FOR LA DEV CODE
6439 4423 1273 CIA
6440 4424 7640 TAD SXMT
6441 4425 4305 SZA CLA /WORD READ NOT = EXPECTED READ DATA
6442 4426 1273 JMS BDTA
6443 4427 7104 TAD SXMT
6444 4430 7440 CLL RAL
6445 4431 5206 SZA
6446 4432 7144 JMP LL1
6447 4433 3273 LL2, CLL CMA RAL /TRY ZERO THROUGH A FIELD OF ONES
6448 4434 1273 DCA SXMT
6449 4435 7421 TAD SXMT
6450 4436 1273 MQL
6451 4437 6666 TAD SXMT
6452 4440 6504 PCLP
6453 4441 7300 LQFC
6454 4442 6667 CLA CLL /READ BUFFER
6455 4443 6501 PRDB
6456 4444 3274 LQRB
6457 4445 1274 DCA SREC /SAVE ACTUAL WORD READ
6458 4446 7040 T70W2, CMA/NOP /CMA FOR LQ DEVICE CODE; NOP FOR LA DEV CODE
6459 4447 7041 CIA
6460 4450 1273 TAD SXMT
6461 4451 7640 SZA CLA /WORD READ NOT = EXPECTED READ DATA
6462 4452 4305 JMS BDTA
6463 4453 1273 TAD SXMT
6464 4454 7120 STL
6465 4455 7004 RAL
6466 4456 7430 SZL
6467 4457 5233 JMP LL2
6468 4460 4447 DONLOP
6469
6470
6471 4461 6662 EXITLA, PCLF /CLEAR FLAG

```

```

6472 4462 7201          CLA IAC
6473 4463 6665          PCIE                /ENABLE LA PRINTER INTERRUPTS - SET LIFT RIBBON=1
6474                    /      NOTE:LIFT RIBBON=1 INDICATES LQ MODE WHEN SIMULATOR IS IN USE.
6475 4464 6002          IOF
6476 4465 6506          LOLS                /ENABLE LQ INTERRUPTS AND SET DONE FLAG
6477 4466 6505          LQRS                /CLEAR DONE FLAG
6478 4467 7300          CLA CLL
6479 4470 3140          DCA TSTNU           /CLEAR TEST NUMBER
6480 4471 5331          JMP SIMCK1
6481 4472 5777*        NOSIM1, JMP PRETEST
6482
6483 4473 0000          SXMT, 0
6484 4474 0000          SREC, 0
6485
6486
6487                    / SUBROUTINES USED BY PRINTER TESTS
6488
6489 4475 0000          XCLEPI, 0           /CLEAR EXPECTING PRINTER INT.
6490 4476 7300          CLA CLL
6491 4477 3135          DCA EPRNTI
6492 4500 5675          JMP I XCLEPI
6493
6494 4501 0000          XSEPI, 0
6495 4502 7240          CLA CMA
6496 4503 3135          DCA EPRNTI
6497 4504 5701          JMP I XSEPI
6498
6499
6500                    /TEST 7 ERROR ROUTINE
6501
6502 4505 0000          BDTA, 0
6503 4506 1274          TAD SREC
6504 4507 4450          ERROR              /WORD READ NOT = EXPECTED READ DATA
6505                    /AC,MC = ACTUAL WORD READ, WORD WRITTEN
6506                    /NOTE: WHEN TESTING WITH LQ DEVICE CODES DATA
6507                    /READ SHOULD BE THE COMPLEMENT OF DATA WORD WRITTEN
6508 4510 5705          JMP I BDTA
6509
6510
6511                    /DISABLE PARALLEL I/O SIMULATOR IF IN USE - NOTE: LQRE ENABLES SIMULATOR
6512
6513 4511 0000          XKLSIM, 0
6514 4512 7300          CLA CLL
6515 4513 4466          GETHW3              /IS PARALLEL I/O SIMULATOR IN USE?
6516 4514 0376          AND (0200
6517 4515 7040          SZA CLA
6518 4516 6007          CAF
6519 4517 5711          JMP I XKLSIM
6520
6521
6522
6523 4520 4466          SIMCK0, GETHW3      /USING SIMULATORS FOR PARALLEL I/O DEVI E?
6524 4521 0376          AND (0200
6525 4522 7650          SNA CLA
6526 4523 5775*        JMP NOSIM0          /NO

```

```

6527 4524 6505          LQRS                /YES-VERIFY LIFT RIBBON FALSE BY READ STATUS
6528 4525 0374          AND (0002
6529 4526 7440          SZA
6530 4527 4450          ERROR              /INITIALIZE(CAF) FAILED TO CLEAR LIFT RIBBON FLOP,
6531 4530 5773*        JMP ILO              /EXECUTE NORMAL LQ TYPE DIAGNOSTIC (DEVICE CODE=50)
6532
6533 4531 4466          SIMCK1, GETHW3      /USING SIMULATORS FOR PARALLEL I/O DEVICE?
6534 4532 0376          AND (0200
6535 4533 7650          SNA CLA
6536 4534 5272          JMP NOSIM1          /NO.
6537 4535 1221          TAD T70W1          /YES - BOTH DEVICE CODES TESTED?
6538 4536 1372          TAD (-CMA
6539 4537 7640          SZA CLA
6540 4540 5771*        JMP TI0LS           /YES - TEST LOGIC WHICH CAN ONLY BE CHECKED BY
6541                    /      USING AN EXTERNAL SIMULATOR.
6542 4541 7320          CLA CLL CML RTL    /NO - SETUP FOR DEVICE CODE 66 INTERFACE TESTING
6543                    /      SET LIFT RIBBON (AC#2)
6544
6545 4542 6506          LOLS
6546 4543 5776*        JMP TILA           /EXECUTE NORMAL LA TYPE DIAGNOSTIC (DEVICE CODE=66)
6547
6548
6549 4570 3436
6550 4571 4600
6551 4572 0740
6552 4573 4005
6553 4574 0002
6554 4575 3420
6555 4576 0200
6556 4577 5000
6557                    4600
6558
6559                    PAGE
6560                    /THE FOLLOWING TESTS CHECK OUT PARALLEL I/O INTERFACE LOGIC WHICH
6561                    /      CAN ONLY BE CHECKED USING A SPECIAL EXTERNAL HARDWARE SIMULATOR.
6562
6563                    /TEST 10 - SIMULATOR REQUIRED - VERIFY IN/OUT AND DATA INTERFACE
6564                    /      CONNECTOR.
6565
6566 4600 4446          TI0LS, LOOPPC
6567 4601 1377          TAD (10
6568 4602 3140          DCA TSTNU           /SETUP TEST NUMBER FOR ERROR DISPLAY,
6569 4603 6506          LOLS                /CLEAR LIFT RIBBON FLOP - SHOULD ALREADY BE CLEAR
6570 4604 6507          LQRE                /ENABLE PARALLEL I/O SIMULATOR
6571 4605 6504          LQPC                /ZERO INTERFACE BUFFER AND SWITCH IN/OUT
6572 4606 6503          LQMC                /      TO LOW BY SETTING ALL
6573 4607 7040          CMA                 /      SIMULATOR STROBE FLOPS.
6574 4610 6502          LQRB
6575 4611 6501          LQRB                /READ & VERIFY THAT IN DATA = 7777
6576 4612 7040          CMA
6577 4613 7440          SZA
6578 4614 4450          ERROR              /IN/OUT FAILED TO DISABLED INTERFACE BUFFER
6579                    /      OUTPUT, AC CONTAINS DATA READ.
6580
6581 4615 7040          CMA
6582 4616 6507          LQRE                /ISSUE 'RESTORE' TO SET IN/OUT = H
6583 4617 7300          CLA CLL
6584 4620 6501          LQRB                /READ & VERIFY THAT 'IN' DATA = 0000
6585 4621 7440          SZA

```

```

6561 4622 4450 ERROR /IN/OUT FAILED TO ENABLE INTERFACE BUFFER OUTPUT.
6562 4623 6505 LQRS /READ STATUS & VERIFY THAT ALL READY FLAGS=0.
6563 4624 7440 SZA /ISSUE CLEAR FLAG.
6564 4625 4450 ERROR /STATUS INCORRECT - AC = STATUS -SHOULD BE ZERO
6565 4626 6500 LQSK /VERIFY NO-SKIP
6566 4627 7410 SKP
6567 4630 4450 ERROR /LQSK SKIPPED WITH ALL FLAGS CLEAR
6568 4631 4447 DONLOP

```

/TEST 11 - SIMULATOR REQUIRED, TEST PAPER STROBE AND PAPER READY LOGIC

```

6590
6591
6592 4632 4446 T11LS, LOOPPC
6593 4633 6502 LQMP /ISSUE PAPER STROBE
6594 4634 6500 LQSK /VERIFY SKIP
6595 4635 4450 ERROR /LQSK FAILED TO SKIP WITH PAPER READY FLAG SET
6596 4636 6505 LQRS /READ STATUS AND VERIFY PAPER READY
6597 4637 1376 TAD (-400)
6598 4640 7640 SZA CLA
6599 4641 4450 ERROR /PAPER READY FLAG DID NOT SET
6600 4642 6507 LQRE /ISSUE RESTORE TO CLEAR FLAG
6601 4643 6525 LQRS /READ STATUS AND VERIFY ALL FLAGS=0
6602 4644 7440 SZA
6603 4645 4450 ERROR /PAPER READY FLAG FAILED TO CLEAR
6604 4646 6500 LQSK /VERIFY NO SKIP
6605 4647 7410 SKP
6606 4650 4450 ERROR /LQSK SKIPPED WITH ALL FLAGS CLEAR
6607 4651 4447 DONLOP

```

/TEST 12 - SIMULATOR REQUIRED, TEST CARRIAGE STROBE AND CARRIAGE READY LOGIC.

```

6608
6609
6610
6611 4652 4446 T12LS, LOOPPC
6612 4653 6503 LQMC /ISSUE CARRIAGE STROBE
6613 4654 6500 LQSK /VERIFY SKIP
6614 4655 4450 ERROR /LQSK FAILED TO SKIP WITH CARRIAGE RDY FLAG SET
6615 4656 6505 LQRS /READ STATUS AND VERIFY CARRIAGE READY
6616 4657 1375 TAD (-1000)
6617 4660 7640 SZA CLA
6618 4661 4450 ERROR /CARRIAGE READY FLAG DID NOT SET,
6619 4662 6507 LQRE /ISSUE RESTORE TO CLEAR FLAG
6620 4663 6505 LQRS /READ STATUS AND VERIFY ALL FLAGS = 0.
6621 4664 7440 SZA
6622 4665 4450 ERROR /CARRIAGE READY FLAG FAILED TO CLEAR
6623 4666 6500 LQSK /VERIFY NO SKIP
6624 4667 7410 SKP
6625 4670 4450 ERROR /LQSK SKIPPED WITH ALL FLAGS CLEAR
6626 4671 4447 DONLOP

```

/TEST 13 - SIMULATOR REQUIRED, TEST CHARACTER STROBE AND CHAR READY LOGIC.

```

6627
6628
6629
6630 4672 4446 T13LS, LOOPPC
6631 4673 6504 LQPC /ISSUE CHARACTER STROBE
6632 4674 6500 LQSK /VERIFY SKIP
6633 4675 4450 ERROR /LQSK FAILED TO SKIP WITH CHAR RDY FLAG SET
6634 4676 6505 LQRS /READ STATUS AND VERIFY CHARACTER READY
6635 4677 1374 TAD (-2000)

```

```

6636 4700 7640 SZA CLA
6637 4701 4450 ERROR /CHARACTER READY FLAG DID NOT SET.
6638 4702 6507 LQRE /ISSUE RESTORE TO CLEAR FLAG
6639 4703 6505 LQRS /READ STATUS AND VERIFY ALL FLAGS=0
6640 4704 7440 SZA
6641 4705 4450 ERROR /CHAR RDY FLAG FAILED TO CLEAR
6642 4706 6500 LQSK /VERIFY NO SKIP
6643 4707 7410 SKP
6644 4710 4450 ERROR /LQSK SKIPPED WITH ALL FLAGS CLEAR
6645 4711 4447 DONLOP

```

/TEST 14 - SIMULATOR REQUIRED, TEST PRINTER READY LOGIC.

```

6646
6647
6648
6649 4712 4446 T14LS, LOOPPC
6650 4713 6502 LQMP /ACTIVATE PRINTER READY INPUT - REQUIRES
6651 4714 6503 LQMC /ISSUEING CARRIAGE STROBE AND PAPER STROBE
6652 4715 6500 LQSK /VERIFY SKIP
6653 4716 4450 ERROR /LQSK FAILED TO SKIP WITH PTR RDY FLAG SET
6654 4717 6505 LQRS
6655 4720 1373 TAD (-4000)
6656 4721 7640 SZA CLA
6657 4722 4450 ERROR /PRINTER READY FLAG DID NOT SET
6658 4723 6507 LQRE /ISSUE RESTORE TO CLEAR FLAG
6659 4724 6505 LQRS /READ STATUS AND VERIFY ALL FLAGS = 0
6660 4725 7440 SZA
6661 4726 4450 ERROR /PTR RDY FLAG FAILED TO CLEAR
6662 4727 6500 LQSK /VERIFY NO SKIP
6663 4730 7410 SKP
6664 4731 4450 ERROR /LQSK SKIPPED WITH ALL FLAGS CLEAR.
6665 4732 4447 DONLOP

```

/TEST 15 - SIMULATOR REQUIRED, TEST CHECK LOGIC.

```

6666
6667
6668
6669 4733 4446 T15LS, LOOPPC
6670 4734 6502 LQMP /ACTIVATE CHECK INPUT - REQUIRES
6671 4735 6504 LQPC /ISSUEING PAPER STROBE AND CHAR STROBE
6672 4736 6500 LQSK /VERIFY SKIP
6673 4737 4450 ERROR /LQSK FAILED TO SKIP WITH CHECK SET
6674 4740 6505 LQRS
6675 4741 1372 TAD (-40)
6676 4742 7640 SZA CLA
6677 4743 4450 ERROR /CHECK FLAG DID NOT SET
6678 4744 6507 LQRE /ISSUE RESTORE TO CLEAR ALL FLAGS
6679 4745 6505 LQRS /READ AND VERIFY STATUS = 0
6680 4746 7440 SZA
6681 4747 4450 ERROR /RESTORE FAILED TO CLEAR CHECK FLAG
6682 4750 6500 LQSK /VERIFY NO SKIP
6683 4751 7410 SKP
6684 4752 4450 ERROR /LQSK SKIPPED WITH ALL FLAGS CLEAR.
6685 4753 4447 DONLOP
6686 4754 7300 CLA CLL
6687 4755 3140 DCA TSTNU
6688 4756 5771 JMP NOSIMI
6689
6690

```


6691
 6692
 6693 4771 4472
 6694 4772 7740
 6695 4773 4800
 6696 4774 6000
 6697 4775 7000
 6698 4776 7400
 6699 4777 8010
 5000

*5000

/TEST 1 = INITIALIZE [CAF] PART 1 / FLAG DETECTION PART 1

6700
 6701
 6702
 6703
 6704
 6705
 6706
 6707
 6708
 6709
 6710
 6711
 6712
 6713
 6714
 6715
 6716
 6717
 6718

/(A) IF AN RX01 MICROCONTROLLER IS NOT CABLED TO THE RX8 INTERFACE,
 THEN ALL FLAGS (DONE, TRANSFER REQUEST, AND ERROR)
 SHOULD HAVE BEEN CLEARED BY INITIALIZE (IF EVER SET).
 NOTE-FOR THE VT78 SYSTEM CAF DOES NOT CLEAR INTERFACE XFER REGISTER
 /
 /(B) IF AN RX01 MICROCONTROLLER IS CABLED TO THE RX8 INTERFACE,
 THEN INITIALIZE SHOULD HAVE SET THE DONE FLAG BECAUSE
 ANY INIT OF THE RX01 MICROCONTROLLER IS AN IMPLIED READ SECTOR
 OF TRACK 0 SECTOR 1 (FOR SYSTEMS PROGRAMMING BOOTSTRAP APPLICATIONS).
 /
 THEREFORE, ANY ERROR (EXCEPT PARITY) THAT MAY OCCUR FROM A NORMAL
 "READ SECTOR" COMMAND MAY OCCUR HERE CAUSEING THE ERROR FLAG TO SET, AND
 DISPLAYING THE ERROR STATUS WITHIN THE TRANSFER REGISTER AT "DONE",
 /
 THE TRANSFER REQUEST FLAG SHOULD BE CLEARED.

6719 5000 4465
 6720 5001 0377
 6721 5002 7640
 6722 5003 4443
 6723 5004 6002
 6724 5005 7240
 6725 5006 3130
 6726 5007 4141
 6727 5010 6200
 6728 5011 4345
 6729
 6730 5012 1776*
 6731 5013 3775*
 6732 5014 4446
 6733 5015 6007
 6734
 6735
 6736
 6737 5016 7300
 6738 5017 6502
 6739
 6740 5020 7300
 6741 5021 4466
 6742 5022 0374
 6743 5023 3133
 6744 5024 6753

PRETEST,GETSR /GET PSEUDO SWITCH REGISTER FROM FIELD 0
 AND (0010
 SZA CLA /EXECUTE FLOPPY INTERFACE TEST?????
 CIEOP /NO
 IOP
 CLA CMA
 DCA ECONS /SET FLAG INDICATING THAT A CONSOLE INTERRUPT IS POSSIBLE NOW
 JMS PATCH /SETUP INTERRUPT SERVICE LINKAGE
 SKPICHW
 PFW, JMS PFMES /PRINT "FLOPPY INTERF TESTING"
 /MESSAGE IF NOT UNDER APT CONTROL.
 TAD KJILD /MODIFY DONLOP ROUTINE.
 DCA LLLD
 TI, LOOPPC
 CAF /IN CASE FLOPPY SIMULATOR IS IN USE SETUP FLOPPY
 /TO "UNCABLED"(NO RX01) STATE,
 /SET PARALLEL I/O MODE TO LG AND CLEAR FLOPPY
 /FLAGS,DISABLE FLOPPY DATA LOOPBACK TO ERRORL,
 /ENABLE PARALLEL I/O SIMULATOR AND SETUP FLOPPY
 /TO NORMAL "UNCABLED" STATE(1>DONE,1>XFER ROSTL
 /1>OUTL,1>ERRORL,1>RXDATAL).
 CLA CLL
 LGMP
 CLA CLL
 GETHW3 /SETUP RXHERE
 AND (2200
 DCA RXHERE /RXHERE=0 IF RX01 IS CABLED TO INTERFACE.
 STF

6745 5025 7410
 6746 5026 4450
 6747 5027 6754
 6748 5030 7410
 6749 5031 4450
 6750
 6751
 6752
 6753
 6754
 6755
 6756
 6757 5032 1133
 6758 5033 7640
 6759 5034 5253
 6760 5035 4464
 6761 5036 6755
 6762 5037 5235
 6763
 6764
 6765
 6766
 6767
 6768
 6769
 6770
 6771
 6772
 6773 5040 4466
 6774 5041 6373
 6775 5042 7650
 6776 5043 5247
 6777 5044 1372
 6778 5045 3134
 6779 5046 5252
 6780 5047 1372
 6781 5050 3134
 6782
 6783
 6784 5051 1371
 6785
 6786
 6787
 6788
 6789
 6790
 6791
 6792
 6793
 6794
 6795
 6796
 6797
 6798 5052 7107
 6799 5053 3132

EOPRE, SKP /UNEXPECTED TRANSFER REQUEST FLAG
 EPPOR
 SER
 SKP
 E2PRE, ERROR /UNEXPECTED ERROR FLAG
 /
 /*****
 /
 /IF AN RX01 MICROCONTROLLER IS CABLED TO THE RX8 INTERFACE
 /THEN THE DONE FLAG SHOULD BE SET
 /
 TAD RXHERE
 SZA CLA
 JMP NORX01
 WAIT
 SDN
 JMP -2 /WAIT FOR DONE FLAG
 /
 /*****
 /
 /THE ENTIRE STATUS WORD IS DISPLAYED IN THE TRANSFER REGISTER AT ERROR/DONE TIME.
 /
 /IF AN RX01 CONTROLLER IS CABLED TO THE RX8 INTERFACE
 / (AND DRIVE 0 IS READY THEN THE STATUS SHOULD INDICATE "SEL DRV RDY"), ALSO
 /DELETED DATA MAY = 1 IF TRACK 0/SECTOR 1 WAS WRITTEN WITH DELETED DATA
 /AND "INIT DONE" SHOULD BE SET.
 /
 GETHW3 /IS DRIVE 0 READY?
 AND (6000
 SNA CLA
 JMP RDY /YES
 TAD (0277 /NO
 DCA COMP
 JMP NORX01=1
 RDY, TAD (0277
 DCA COMP
 /
 /
 TAD (40 /PROGRAM EXPECTS DRIVE 0 TO BE READY
 /
 /
 /
 4 5 - - 8 9 10 11 /
 /
 SEL WRITE INIT PAR /
 DRIVE DD PROTECT [DONE] CRC /
 RDY (N/A) /
 /
 /
 CLL IAC RTL / 4 [INIT] DONE OR 204
 NORX01, DCA GOOD

```

/VT78 CPU DIAGNOSTIC PAL10 V142A 2-AUG-77 7130 PAGE 15-2 SEQ 0151
6800 5054 1132 TAD GOOD /LOAD MQ FOR DISPLAY
6801 5055 7421 MQL
6802 5056 6752 XDR /TRANSFER DATA REGISTER FROM RX01 CONTROL
6803 /NOTE: IF NO RX01 IS CABLED, THEN
6804 /THIS XDR INST WILL CAUSE AC(=0) > INTERFACE REGISTER.
6805 5057 3130 DCA BLANK /ACTUAL STATUS FROM (INIT)
6806 5060 1130 TAD BLANK
6807 5061 0134 AND COMP
6808 5062 3131 DCA EAC /STATUS MINUS DELETED DATA (BIT 5)
6809 5063 1131 TAD EAC
6810 5064 7041 CTA
6811 5065 1132 TAD GOOD /EXPECTED
6812 5066 7650 SNA CLA
6813 5067 5272 JMP ,+3 /OK
6814 5070 1131 TAD EAC
6815 5071 4450 EIPRE, ERROR / [INIT] STATUS NOT = EXPECTED
6816 /AC, MQ=ACTUAL STATUS, EXPECTED STATUS
6817 /
6818 /IF AN RX01 MICROCONTROLLER IS CABLED TO THE RX8 INTERFACE
6819 /THEN THE PREVIOUS "SDN" SHOULD HAVE CLEARED THE DONE FLAG, BUT
6820 /
6821 /IF AN RX01 MICROCONTROLLER IS NOT CABLED TO THE RX8 INTERFACE
6822 /THEN INITIALIZE SHOULD HAVE CLEARED THE DONE FLAG
6823 /
6824 /TECHNICAL NOTE:
6825 /
6826 /IF THE DONE FLAG IS SET, AND IF THE INTERRUPT ENABLE FLIP-FLOP IS SET ILLEGALLY,
6827 /THEN AN "UNEXPECTED RX01 INTERRUPT" WILL OCCUR IN TEST 2 (IF AN RX01 CONTROLLER
6828 / IS CABLED TO THE RX8 INTERFACE) OR IN TEST 3 WHEN THE MAINTENANCE FLIP-
6829 /FLOP "SETS ALL FLAGS"
6830 /
6831 5072 6755 SDN
6832 5073 7410 SKP
6833 5074 4450 E4PRE, ERROR /UNEXPECTED DONE FLAG
6834 5075 4447 DONLOP
6835 /
6836 /
6837 /TEST 2 - FLAG DETECTION PART II / " C " LINES VERIFICATION PART I
6838 /
6839 /*****
6840 /*****
6841 /
6842 /IF AN RX01 MICROCONTROLLER IS CABLED TO THE RX8 INTERFACE
6843 /THEN DON'T EXECUTE THIS TEST
6844 /BECAUSE ISSUING THE IOT LCD WITH THE AC = 177
6845 /RESEMBLES A COMMAND TO THE RX01
6846 /
6847 5076 4446 T2, LOOPPC
6848 5077 1133 TAD RXHERE
6849 5100 7650 SNA CLA
6850 5101 5321 JMP T3
6851 5102 3132 DCA GOOD
6852 /*****
6853 /*****
6854 /

```

```

/VT78 CPU DIAGNOSTIC PAL10 V142A 2-AUG-77 7130 PAGE 15-3 SEQ 0152
6855 /THE PURPOSE OF THIS TEST IS TO VERIFY THAT THE LCD (LOAD COMMAND REGISTER)
6856 /IOT 6751 DOES NOT SET THE MAINTENANCE FLIP-FLOP WHEN THE CONTENTS
6857 /OF THE AC = 177 AT THE TIME THE LCD IOT IS ISSUED.
6858 /
6859 /
6860 /TECHNICAL NOTE:
6861 /
6862 /IF AN ERROR OCCURS, THEN IT IS ASSUMED (CAF) INIT FAILED TO CLEAR THE
6863 /MAINTENANCE FLIP-FLOP, OR, THAT THE ISSUING OF THE LCD IOT REALLY
6864 /SET THE MAINTENANCE FLIP-FLOP INSTEAD OF CLEARING .
6865 /
6866 /
6867 / " C " LINES VERIFICATION PART I
6868 /
6869 5103 1370 TAD (177)
6870 5104 6751 LCD /MAINTENANCE MODE <OFF>
6871 /THE (AC) SHOULD = 0 BECAUSE IOT LCD 6751 SHOULD CLEAR THE AC
6872 /
6873 5105 7440 SZA
6874 5106 4450 E0, ERROR / IOT 6751 DID NOT CLEAR THE AC
6875 /
6876 /FLAG DETECTION PART II
6877 /
6878 /THE PURPOSE OF THIS TEST IS TO VERIFY THAT ISSUING IOT LCD 6751 WITH
6879 /THE AC = 177 DOES NOT SET THE MAINTENANCE FLIP-FLOP
6880 /WHICH IN TURN WOULD SET ALL FLAGS
6881 /
6882 /THEREFORE ALL FLAGS SHOULD BE CLEARED
6883 /
6884 5107 6753 STR
6885 5110 7410 SKP
6886 5111 4450 E1, ERROR /UNEXPECTED TRANSFER REQUEST FLAG
6887 5112 6754 SER
6888 5113 7410 SKP
6889 5114 4450 E2, ERROR /UNEXPECTED ERROR FLAG
6890 5115 6755 SDN
6891 5116 7410 SKP
6892 5117 4450 E3, ERROR /UNEXPECTED DONE FLAG
6893 5120 4447 DONLOP
6894 /
6895 /TEST 3 - DIRECTION OF IOT XDR (6752) PART I / IOT DECODING PART I
6896 /
6897 / - " C " LINES VERIFICATION PART II
6898 /
6899 /THE PURPOSE OF THIS TEST IS TO VERIFY THAT
6900 /ISSUING THE IOT XDR (TRANSFER DATA REGISTER) 6752 DOES NOT CLEAR
6901 /THE MAINTENANCE FLIP-FLOP
6902 /
6903 /TECHNICAL NOTE:
6904 /
6905 /IF THE CONTENTS OF THE TRANSFER REGISTER IS NOT = 200, THEN IT MUST
6906 /BE ASSUMED THAT THE SECOND LCD IOT CLEARED THE MAINTENANCE FLIP-FLOP
6907 /OR THAT IOT XDR CLEARED THE MAINTENANCE FLIP-FLOP
6908 /
6909 5121 4446 T3, LOOPPC
6910 5122 3132 DCA GOOD

```

```

6910 5123 1367          TAD (200)
6911 5124 6751          LCD           / MAINTENANCE MODE <ON>
6912                    /THE (AC) SHOULD = 0 AFTER ISSUING IOT LCD 6751
6913                    /
6914 5125 7440          SZA
6915 5126 4450          E11, ERROR           / IOT LCD 6751 FAILED TO CLEAR AC
6916                    /
6917 5127 1367          TAD (200)
6918 5130 6751          LCD           / MAINTENANCE MODE <ON>, AGAIN
6919 5131 6752          XDR           /CONTENTS OF TRANSFER REGISTER
6920 5132 3131          DCA EAC          /SAVE
6921 5133 1131          TAD EAC
6922 5134 1366          TAD (-200)        /COMPARE WITH "EXPECTED"
6923 5135 7650          SNA CLA
6924 5136 5343          JMP ,+5           / OK
6925 5137 1367          TAD (200)
6926 5140 7421          MQL           / "EXPECTED" RESULT
6927 5141 1131          TAD EAC          / "ACTUAL" RESULT
6928 5142 4450          E10, ERROR        /TRANSFER REGISTER NOT =200
6929                    /AC,MQ=ACTUAL,EXPECTED RESULT
6930                    DONLOP
6931 5144 5765          JMF T4
6932
6933                    /ROUTINE TO PRINT "FLOPPY INTERFACE TESTING" MESSAGE
6934
6935                    PFMES, 0
6936 5145 0000          C1CRLF
6937 5146 4432          C1PRT
6938 5147 4434          FLMS
6939 5150 5242          C1PRT
6940 5151 4434          TESMES
6941 5152 3306          C1CRLF
6942 5153 4432          JMP I PFMES
6943 5154 5745
6944
6945
6946
6947 5165 5200
6948 5166 7600
6949 5167 0200
6950 5170 0177
6951 5171 0040
6952 5172 0277
6953 5173 0000
6954 5174 2200
6955 5175 2257
6956 5176 2267
6957 5177 0010
6958                    5200
6959
6960                    PAGE
6961                    /TEST 4 = FLAG DETECTION PART III / " C " LINES VERIFICATION PART III
6962                    /
6963 5200 4446          / THE SETTING OF THE MAINTENANCE FLIP-FLOP SHOULD "DIRECT SET" ALL
6964                    / FLAGS (DONE, TRANSFER REQUEST, AND ERROR).
6965                    /
6966                    T4, LOOPPC

```

```

6964 5201 1377          TAD (200)
6965 5202 6751          LCD           / MAINTENANCE <ON>
6966 5203 6755          SDN
6967 5204 4450          E20, ERROR           /MISSING DONE FLAG
6968 5205 6753          STR
6969 5206 4450          E21, ERROR           /MISSING TRANSFER REQUEST FLAG
6970 5207 6754          SFR
6971 5210 4450          E22, ERROR           /MISSING ERROR FLAG
6972                    /
6973                    /ALL FLAGS SHOULD REMAIN " DIRECT SET "
6974                    /BECAUSE THE MAINTENANCE FLIP-FLOP SHOULD STILL BE SET
6975                    /
6976                    /IF THE FLAGS ARE "MISSING" THEN IT IS ASSUMED THAT THE PREVIOUS
6977                    /FLAG TESTING ACTUALLY (CLEARED) THE FLAGS.
6978                    /
6979 5211 6755          SDN
6980 5212 4450          E23, ERROR           /MISSING DONE FLAG
6981 5213 6753          STR
6982 5214 4450          E24, EPROP        /MISSING TRANSFER REQUEST FLAG
6983 5215 6754          SER
6984 5216 4450          E25, ERROR           /MISSING ERROR FLAG
6985                    /
6986                    / " C " LINES VERIFICATION PART III
6987                    /
6988                    /THE FOLLOWING RX0 INTERFACE IOT'S SHOULD NOT CLEAR THE AC;
6989                    /IOT'S: SDN(6755), SER(6754), OR STR(6753)
6990                    /
6991 5217 7240          STA
6992 5220 3132          DCA GOOD
6993 5221 1132          TAD GOOD
6994 5222 6755          SDN
6995 5223 7000          NOP
6996 5224 7650          SNA CLA
6997 5225 4450          E26, ERROR           / IOT SDN (6755) CLEARED THE AC
6998 5226 1132          TAD GOOD
6999 5227 6753          STR
7000 5230 7000          NOP
7001 5231 7650          SNA CLA
7002 5232 4450          E27, ERROR           / IOT STR (6753) CLEARED THE AC
7003 5233 1132          TAD GOOD
7004 5234 6754          SFR
7005 5235 7000          NOP
7006 5236 7650          SNA CLA
7007 5237 4450          E28, ERROR           / IOT SER (6754) CLEARED THE AC
7008 5240 4447          DONLOP
7009 5241 5253          JMP TS
7010 5242 0014          FLMS, TEXT    "FLOPPY INTERFACE "
7011 5243 1720
7012 5244 2031
7013 5245 4011
7014 5246 1624
7015 5247 0522
7016 5250 0001
7017 5251 0305
7018 5252 4000

```



```

7079 5304 1130 TAD BLANK
7080 5305 7430 SZL
7081 5306 8375 AND (377) / 8-BIT BYTE "GOOD" MASK
7082 5307 3132 DCA GOOD
7083 5310 1132 TAD GOOD /LOAD MQ FOR DISPLAY ON ERROR
7084 5311 7421 MQL
7085 5312 1132 TAD GOOD /EXPECTED RESULT
7086 5313 7041 CIA
7087 5314 1131 TAD EAC /ACTUAL RESULT
7088 5315 7650 SNA CLA
7089 5316 5321 JMP ,+3 /COMPARED OK
7090 5317 1131 TAD EAC
7091 5320 4450 E40, ERROR / TRANSFER REGISTER NOT = "GOOD"
7092 / /AC,MQ=ACTUAL,EXPECTED RESULTS
7093 /
7094 5321 1130 TAD BLANK
7095 5322 2376 AND (100)
7096 5323 7640 SZA CLA
7097 5324 5260 JMP T5B /UNTIL (BLANK) = 7677
7098 5325 4447 DONLOP
7099
7100
7101 /TEST 6
7102 /
7103 /RX8 IOT DECODING VERIFICATION PART II
7104 /
7105 /*****
7106 /*****
7107 /
7108 /NOTE=CLEARING OF THE MAINTENANCE F/F ON THE VT78 DOES NOT
7109 /RESEMBLE A FILL BUFFER COMMAND
7110 /
7111 5326 4446 T6, LOOPPC
7112 /*****
7113 /*****
7114 /
7115 /THE MAINTENANCE FLIP-FLOP HAS PREVIOUSLY BEEN VERIFIED TO SET AND CLEAR,
7116 /THE IOT UNDER TEST SHOULD "SKIP AND CLEAR" (ONLY) ITS RESPECTIVE FLAG,
7117 /ALL OTHER FLAGS SHOULD REMAIN UNCHANGED
7118 /
7119 / (I.E. THE SDN IOT 6755 SHOULD SKIP AND CLEAR ONLY THE DONE FLAG, ALL
7120 /OTHER FLAGS SHOULD REMAIN SET)
7121 /
7122 5327 1377 TAD (200)
7123 5330 6751 LCD
7124 5331 6751 LCD / MAINTENANCE <ON> / <OFF>
7125 5332 6752 XDR
7126 5333 7440 SZA
7127 5334 4450 E56, ERROR /TRANSFER REGISTER NOT = 0
7128 5335 6755 SDN
7129 5336 4450 E50, ERROR /DONE FLAG WASN'T SET, OR
7130 /IOT LCD OR XDR CLEARED THE DONE FLAG
7131 5337 6755 SDN
7132 5340 7410 SKP
7133 5341 4450 E53, ERROR /IOT SDN DIDN'T SKIP AND CLEAR

```

```

7134 5342 6753 STR
7135 5343 4452 E51, EPROR /TRANSFER REQUEST FLAG WASN'T EVER SET, OR
7136 /IOT LCD, OR SDN OR XDR CLEARED THE TR FLAG
7137 5344 6753 STR
7138 5345 7410 SKP
7139 5346 4450 E54, EPROR /IOT STR DIDN'T "SKIP AND CLEAR"
7140 5347 6754 SER
7141 5350 4450 E52, EPROR /ERROR FLAG WASN'T EVER SET, OR
7142 /IOTS LCD OR SDN OR XDR OR STR CLEARED THE ERROR FLAG
7143 5351 6754 SER
7144 5352 7410 SKP
7145 5353 4450 E55, EPROR /IOT SER DIDN'T "SKIP AND CLEAR"
7146 5354 4447 DONLOP
7147 5355 5774 JMP T7
7148
7149
7150
7151 /
7152 5374 5430
7153 5375 0377
7154 5376 0100
7155 5377 0200
7156 /
7157 /PAGE
7158 /TEST 7 - INTERRUPT TEST PART I / IOT DECODING VERIFICATION PART III
7159 /
7160 /INTERRUPT TEST PART I (NO INTERRUPTS SHOULD OCCUR DURING THIS TEST)
7161 /
7162 /THE MAINTENANCE FLIP-FLOP HAS PREVIOUSLY BEEN VERIFIED TO DIRECT
7163 / SET ALL FLAGS AND THE INTERFACE IOT -SKIP ON DONE- "SDN" 6755 WAS
7164 /FOUND TO "SKIP AND CLEAR" SUCCESSFULLY,
7165 /
7166 /FIRST SET THE MAINTENANCE FLIP-FLOP WHICH IN TURN SETS ALL FLAGS,
7167 /
7168 /THEN ISSUE IOT INTR 6756 WITH THE AC = 0 CLEARING THE RX8 INTERRUPT ENABLE
7169 /NO INTERRUPTS SHOULD OCCUR
7170 /
7171 5400 4446 T7, LOOPPC
7172 5401 6002 IOF
7173 5402 4141 JMS PATCH /SETUP INTERRUPT SERVICE LINKAGE TO RETURN TO THIS TEST
7174 5403 5414 E00
7175 5404 6035 KIE /DISABLE CONSOLE PACKAGE INTERRUPTS
7176 5405 6001 ION
7177 5406 1377 TAD (200)
7178 5407 6751 LCD
7179 5410 6750 INTR /INTERRUPT ENABLE FLIP-FLOP <OFF>
7180 5411 7000 NOP /...WAIT
7181 5412 7000 NOP /...PLENTY
7182 5413 7410 SKP /...OF TIME
7183 5414 4450 E60, ERROR /UNEXPECTED INTERRUPT
7184 /
7185 /
7186 /*****
7187 /*****
7188 /
7189 /IOT DECODING PART III- IOT INTR 6756 DECODING VERIFICATION

```

```

7188 /
7189 /TECHNICAL NOTE:
7190 /
7191 /ALL FLAGS SHOULD REMAIN SET
7192 /IF ANY FLAG IS MISSING,
7193 /THEN IT IS ASSUMED THAT IOT "INTR" 6756 CLEARED THE FLAG(S)
7194 /
7195 5415 6751 LCD /MAINTENANCE MODE <OFF>
7196 5416 6756 INTR /DISABLE THE INTERRUPT ENABLE F/F
7197 5417 6755 SDN
7198 5420 4450 E61, ERKOR /MISSING DONE FLAG
7199 5421 6753 STR
7200 5422 4450 E62, EPROR /MISSING TRANSFER REQUEST FLAG
7201 5423 6754 SER
7202 5424 4450 E63, ERROR /MISSING ERROR FLAG
7203 5425 4447 DONLOP
7204 /
7205 /TEST 10 - INTERRUPT TEST PART II
7206 /
7207 /INTERRUPT TEST PART II (AN INTERRUPT SHOULD OCCUR IN THIS TEST).
7208 /
7209 /THE MAINTENANCE FLIP-FLOP HAS PREVIOUSLY BEEN VERIFIED TO DIRECT
7210 /SET ALL FLAGS AND THE INTERFACE IOT =SKIP ON DONE= "SDN" 6755 WAS
7211 /FOUND TO "SKIP AND CLEAR" SUCCESSFULLY,
7212 /
7213 /FIRST SET THE MAINTENANCE FLIP-FLOP
7214 /WHICH SHOULD DIRECT SET THE DONE FLAG,
7215 /THEN BY SETTING THE RX01 INTERRUPT ENABLE
7216 /BY ISSUING THE IOT "INTR" 6756 WITH THE AC = 1.
7217 /
7218 /AN INTERRUPT REQUEST SHOULD BE ASSERTED,
7219 /
7220 /THE PROGRAM IS EXPECTING AN INTERRUPT.
7221 /
7222 /TECHNICAL NOTE:
7223 /
7224 /IF AN INTERRUPT DOES NOT OCCUR, THEN IT IS ASSUMED THAT ISSUING THE IOT
7225 /"INTR" 6756 DID NOT SET THE RX8 INTERRUPT ENABLE, OR INTERRUPT REQUEST
7226 /
7227 5426 4446 T10, LOOPPC
7228 5427 6002 IOF
7229 5430 4141 JMS PATCH /SETUP INT, LINKAGE TO RETURN TO THIS TEST
7230 5431 5444 T70K
7231 5432 6035 KIE /DISABLE CONSOLE PACKAGE INT.
7232 5433 1377 TAD (200)
7233 5434 6751 LCD /MAINTENANCE <ON>
7234 5435 6001 ION
7235 5436 7201 CLA IAC
7236 5437 6756 INTR /RX01 INTERRUPT ENABLE <ON>
7237 5440 7000 NOP
7238 5441 7000 NOP
7239 5442 7000 NOP
7240 5443 4450 E70, ERROR /MISSING INTERRUPT
7241 5444 4447 T70K, DONLOP
7242 /

```

```

7243 /
7244 /TEST 11
7245 /
7246 /INTERRUPT TEST (PART III) (NO INT. SHOULD OCCUR DURING THIS TEST).
7247 /
7248 /IOT INTR 6756 SHOULD CLEAR THE INTERRUPT ENABLE FLIP-FLOP, THEN
7249 /WITH ALL FLAGS SET, NO INTERRUPTS SHOULD OCCUR
7250 /
7251 /TECHNICAL NOTE:
7252 /
7253 /IF AN UNEXPECTED PROGRAM INTERRUPT OCCURS FROM APPROXIMATELY THIS PC
7254 /THEN THE RX PROGRAM INTERRUPT REQUEST TOOK TOO LONG TO SET
7255 /FROM THE PREVIOUS TEST.
7256 /
7257 5445 4446 T11, LOOPPC
7258 5446 6002 IOF
7259 5447 4141 JMS PATCH /SETUP INT, LINKAGE TO RETURN TO THIS TEST
7260 5450 5460 E100
7261 5451 6035 KIE /DISABLE CONSOLE PACKAGE INT.
7262 5452 6756 INTR /DISABLE RX8 INTERRUPT ENABLE
7263 5453 6001 ION
7264 5454 1377 TAD (200)
7265 5455 6751 LCD
7266 5456 7000 NOP
7267 5457 7410 SKP
7268 5460 4450 E100, ERROR /UNEXPECTED INTERRUPT
7269 5461 4447 DONLOP
7270 /
7271 /TEST 12
7272 /
7273 /INTERRUPT TEST (PART IV) (NO INT. SHOULD OCCUR DURING THIS TEST).
7274 /
7275 /*****
7276 /*****
7277 /
7278 /TOGLING THE MAINTENANCE MODE <ON> / <OFF> SETS ALL FLAGS AND
7279 /
7280 /PERMITS IOT SDN TO CLEAR THE DONE FLAG
7281 /
7282 /THEREFORE NO INTERRUPTS SHOULD OCCUR (ONLY DONE FLAG RAISES AN INTERRUPT REQUEST)
7283 /
7284 5462 4446 T12, LOOPPC
7285 5463 6002 IOF
7286 5464 4141 JMS PATCH /SETUP INTERRUPT LINKAGE TO RETURN TO THIS TEST
7287 5465 5303 E110
7288 5466 6035 KIE /DISABLE CONSOLE PACKAGE INT.
7289 5467 1377 TAD (200)
7290 5470 6751 LCD

```

```

7298 5471 6751 LCD /MAINTENANCE <ON> / <OFF>
7299 5472 6755 SDN
7300 5473 7000 NOP /CLEAR THE DONE FLAG
7301 5474 7000 NOP
7302 5475 6001 ION
7303 5476 7201 CLA IAC /RX01 INTERRUPT ENABLE <ON>
7304 5477 6756 INTR
7305 5500 7000 NOP
7306 5501 7000 NOP
7307 5502 7410 SKP
7308 5503 4450 E110, ERROR /UNEXPECTED INTERRUPT
7309 5504 6756 INTR /RX01 INT. EN <OFF>
7310 5505 4447 DONLOP
7311 5506 5307 JMP T13

/TEST 13 - INITIALIZE TEST PART II INIT / INTERRUPT TEST PART V
/
/*****
/*****
/IF AN RX01 IS CABLED TO THE RX0 THEN DON'T EXECUTE THIS TEST
/
T13, LOOPPC
TAD RXHERE
SNA CLA
JMP FLDON
/*****
/*****
/ INTERRUPT TEST PART V / INITIALIZE TEST PART II
/ (NO INTERRUPTS SHOULD OCCUR DURING THIS TEST).
/
/ THE PURPOSE OF THIS TEST IS TO VERIFY THAT IOT INIT CLEARS THE INTERRUPT
/ ENABLE FLIP-FLOP WHEN SET
/
IOP /SETUP INT. LINKAGE TO RETURN TO THIS TEST
JMS PATCH
E124
KIE /DISABLE CONSOLE PACKAGE INTERRUPTS
ION
CLA IAC / SET THE RX0 INTERRUPT ENABLE F/F
INTR
INIT
/...BUT AN INTERRUPT SHOULD NOT OCCUR
/IF AN INTERRUPT OCCURS THEN IOT INIT FAILED TO CLEAR
/
/ THE RX0 INTERRUPT ENABLE FLIP-FLOP
/
TAD (200)
LCD / MAINTENANCE MODE <ON> / <OFF>
LCD
/
/ THE DONE FLAG SHOULD BE SET, BUT NO INTERUPTS SHOULD OCCUR
/

```

```

7353 5526 7410 SKP
7354 /
7355 /RETURN TO HERE IF AN INTERRUPT OCCURED
7356 /
7357 5527 4450 E124, ERROR /UNEXPECTED INT.
7358 /
7359 /IOT "INIT" 6757 SHOULD CLEAR THE
7360 /MAINTENANCE FLIP-FLOP, AND ALL FLAGS (DONE, TRANSFER REQUEST, AND ERROR).
7361 /
7362 5530 7240 STA
7363 5531 6751 LCD /ALL 1'S TO TRANSFER REGISTER
7364 5532 6757 INIT / IOT 6757
7365 5533 6755 SDN
7366 5534 7410 SKP
7367 5535 4450 E120, ERROR /UNEXPECTED DONE FLAG
7368 5536 6753 STR
7369 5537 7410 SKP
7370 5540 4450 E121, ERROR /UNEXPECTED TRANSFER REQUEST FLAG
7371 5541 6754 SER
7372 5542 7410 SKP
7373 5543 4450 E122, ERROR /UNEXPECTED ERROR FLAG
7374 5544 4447 DONLOP
7375 5545 4465 FLDON, GETHWJ /IS THE FLOPPY SIMULATOR IN USE?
7376 5546 0377 AND (0200)
7377 5547 7640 JMP T14S /YES-CONTINUE WITH FLOPPY INTERFACE TESTING.
7378 5550 5776 DCA TSTNU /CLEAR TEST NUMBER
7379 5551 3140 C1EOP
7380 5552 4443
7381 /WAIT FOR FLAG ROUTINE
7382 /
7383 /
7384 5553 0000 XWAIT, 0
7385 5554 2366 ISZ H1
7386 5555 5753 JMP I XWAIT
7387 5556 2367 ISZ HANGER
7388 5557 5753 JMP I XWAIT
7389 5560 1353 TAD XWAIT
7390 5561 4450 ERROR /FLAG NEVER SET
7391 5562 1353 TAD XWAIT
7392 5563 1375 TAD (2
7393 5564 3353 DCA XWAIT
7394 5565 5753 JMP I XWAIT
7395 5566 0000 H1, 0
7396 5567 0000 HANGER, 0
7397 /
7398 /
7399 5575 0002
7400 5576 5600
7401 5577 0200
7402 5600
PAGE
/ THE REMAINING FLOPPY INTERFACE TESTS REQUIRE A SPECIAL HARDWARE
/ SIMULATOR CONNECTED TO THE EXTERNAL FLOPPY PORT.
/
7404 /
7405 5600 4777 T14S, JMS INTUN /INITIALIZE PROG UNIT VARIABLES FOR UNIT A
7406 5601 4446 LOOPPC

```

```

7407 5602 4776* MODAGN, JMS INTMOD /INITIALIZE PROG MODE VARIABLES FOR 12 BIT MODE
7408 5603 6007 CAF /INITIALIZE FLOPPY INTERFACE=MAINT MODE=OFF,
7409 / MODE=12 BIT,UNIT=A,INI,EN=OFF,
7410 / RUNL=FALSE,FLAGS CLEARED,PARITY FLOP=CLEARED,
7411 5604 6507 LQRE /CLEAR PARALLEL I/O STROBE FLOPS AND ENABLE
7412 / THE PARALLEL I/O SIMULATOR & THE FLOPPY
7413 / RXDATA TO ERROR LOOPBACK.
7414 5605 1375 TAD (0017) /SETUP FLOPPY (VIA PARALLEL I/O INTERFACE BUFFER)
7415 / TO NORMAL IDLE STATE(0=DONEL,1=XFER RQSTL,
7416 / 1=OUTL). NOTE PARALLEL I/O INVERTS DATA
7417 / IN LQP MODE.
7418 5606 6502 LQMP /ALSO SETS PAPER RDY VIA PAPER STROBE
7419 5607 3774* DCA SHFTCT /INITIALIZE SHIFT COUNT TO ZERO
7420 5610 1773* TAD UNIT /AC=0 FOR UNIT A, AC=1 FOR UNIT B
7421 5611 6750 SEL /SETUP "UNIT" (A OR B )
7422 5612 6505 LQRS /VERIFY RUNL FALSE VIA PAPER READY
7423 5613 0372 AND (400) /FLAG = SET.
7424 5614 7650 SNA CLA
7425 5615 4450 ERROR /FITHER CAF FAILED TO ENABLE PARALLEL I/O
7426 / SIMULATOR OR FLOPPY RUNL IS TRUE.
7427 /NOTE: RUNL TRUE DISABLES PARALLEL I/O STROBE
7428 / DECODER.
7429 5616 1771* TAD LCDATA /AC=2452 FOR 12 BIT MODE,AC=0125 FOR 8 BIT MODE
7430 5617 6751 LCD / MAINT BIT = 0 IN EITHER CASE.
7431 5620 6505 LQRS /VERIFY THAT LCD CAUSES RUNL=TRUE
7432 5621 0372 AND (400)
7433 5622 7640 SZA CLA
7434 5623 4450 ERROR /LCD FAILED TO CAUSE RUNL TO GO TRUE.
7435 5624 1370 TAD (0017) /CHANGE DONEL TO 1 FOR UNIT UNDER TEST
7436 5625 1767* TAD UDONE /UDONE=2000 FOR UNIT A,4000 FOR UNIT B
7437 5626 6502 LQMP
7438 5627 6505 LQRS /VERIFY THAT DONEL GOING H CAUSES RUNL=FALSE
7439 / PAPER RDY FLAG SHOULD BE SET.
7440 5630 0372 AND (400)
7441 5631 7650 SNA CLA
7442 5632 4450 ERROR /DONEL FAILED TO CAUSE RUNL TO GO FALSE
7443 5633 0000 SHIFTO, 0/LQRE/LQPC /SHIFT OUT FLOPPY INTERFACE REGISTER
7444 / SHIFTL=RESTORE,SHIFTL=CHAR, STROBE
7445 5634 2774* ISZ SHFTCT /UPDATE SHIFT COUNTER
7446 5635 7300 CLA CLL
7447 5636 1774* TAD SHFTCT /NO OF SHIFTS ISSUED?
7448 5637 1366 TAD (-10)
7449 5640 7450 SNA
7450 5641 5250 JMP CKMODE /0 = CHECK MODE
7451 5642 7510 SPA
7452 5643 5255 JMP SMORE /<0 = TEST CURRENT RXDATA BIT & SHIFT.
7453 5644 1365 TAD (-4)
7454 5645 7640 SZA CLA
7455 5646 5255 JMP SMORE /0<SHIFT COUNT<12 = TEST CURRENT RXDAT & SHIFT
7456 5647 5257 JMP ISTPC /12 = TEST PARITY BIT,DONE SHIFTING
7457 5650 1771* CKMODE, TAD LCDATA /CHECK FOR 8 OR 12 BIT MODE
7458 5651 7041 CIA
7459 5652 1764* TAD DLCD8
7460 5653 7650 SNA CLA
7461 5654 5257 JMP ISTPC /8 BIT MODE = TEST PARITY BIT

```

```

7462 5655 4763* SMORE, JMS ISTDAB /TEST CURRENT RXDATA
7463 5656 5233 JMP SHIFTO /SHIFT OUT AGAIN
7464 5657 4763* TSTPC, JMS TSTDAB /TEST PARITY BIT-DONE WITH SHIFTING
7465 5660 3774* DCA SHFTCT /ZERO SHIFT COUNTER
7466 5661 4762* JMS SHINDA /SETUP SHIFT IN DATA
7467 5662 0000 SHIFTI, 0/LQRE/LQPC /SHIFT IN FLOPPY INTERFACE REGISTER
7468 5663 2774* ISZ SHFTCT /UPDATE SHIFT COUNTER
7469 /SHIFTL=RESTORE;SHIFTL=CHAR STROBE.
7470 5664 7300 CLA CLL
7471 5665 1774* TAD SHFTCT /NO OF SHIFTS ISSUED?
7472 5666 1366 TAD (-10)
7473 5667 7450 SNA
7474 5670 5277 JMP CKIMOD /0 = CHECK MODE
7475 5671 7510 SPA
7476 5672 5261 JMP SHIFTI-1 /<0 = SHIFT AGAIN
7477 5673 1365 TAD (-4)
7478 5674 7640 SZA CLA
7479 5675 5261 JMP SHIFTI-1 /0<SHIFT COUNT<12 = SHIFT AGAIN
7480 5676 5304 JMP CKINDA /DONE SHIFTING IN, CHECK DATA WORD THAT
7481 / GOT SHIFTED IN.
7482 5677 1761* CKIMOD, TAD XDRDATA /CHECK FOR 8 OR 12 BIT MODE.
7483 5700 7041 CIA
7484 5701 1760* TAD DXDR8
7485 5702 7640 SZA CLA
7486 5703 5261 JMP SHIFTI-1 /12 BIT MODE = CONTINUE SHIFTING
7487 /DONE SHIFTING IN = CHECK DATA WORD IN INTERFACE REG.
7488
7489 5704 6753 CKINDA, STR /TRANSFER REQUEST FLAG SHOULD BE CLEAR
7490 5705 7410 SKP
7491 5706 4450 ERROR /TRANSFER REQUEST FLAG FALSELY SET
7492 5707 1370 TAD (0017)
7493 5710 1767* TAD UDONE /CHANGE XFER RQSTL TO 0 FOR PROPER UNIT.
7494 5711 1757* TAD UOUT
7495 5712 1756* TAD UXFR
7496 5713 6502 LQMP
7497 5714 6753 STR
7498 5715 4450 ERROR
7499 5716 6752 XDR /CHANGING XFER RQSTL TO 0 FAILED TO SET FLAG
7500 5717 7421 MQL /READ INTERFACE REGISTER INTO AC.
7501 5720 7701 ACL /SAVE READ DATA IN MQ FOR ERROR DISPLAY
7502 5721 7041 CIA /RESTORE DATA TO AC FROM MQ
7503 5722 1761* TAD XDRDATA /COMPARE WITH EXPECTED DATA
7504 /EXPECTED DATA=0525 FOR 12 BIT MODE
7505 / " " =0242 FOR 8 BIT MODE
7506 5723 7440 SZA
7507 5724 4450 ERROR /BAD SHIFT IN = MQ CONTAINS ACTUAL DATA
7508 / SHIFTED IN.
7509 5725 6505 LQRS /VERIFY THAT XDR SETS RUNL TO TRUE
7510 / PAPER RDY FLAG=CLEAR
7511 5726 0372 AND (400)
7512 5727 7640 SZA CLA
7513 5730 4450 ERROR /XDR FAILED TO CHANGE RUNL TO TRUE.
7514 5731 1370 TAD (0017) /CHANGE XFR REQ L TO 1 & OUTL TO 1.
7515 5732 1767* TAD UDONE
7516 5733 6502 LQMP

```



```

7517 5734 6545      LQMS          /VERIFY THAT RUNL GOES FALSE
7518 5735 6355      AND (440)    /NOTE:FOR UNIT B TESTING CHAR STROBE GETS SET
7519              /          BY LQPC USED IN SHIFTING ABOVE.
7520 5736 7650      SNA CLA      /CHANGING XFR ROSTL TO 1 FAILED TO NEGATE RUNL
7521 5737 4450      ERROR      /CHANGE DONEL TO 0
7522 5740 1375      TAD (6017)
7523 5741 6502      LQMP
7524 5742 6755      SDN
7525 5743 4450      LPROR      /DONE FLAG FAILED TO SET
7526 5744 4754*     JMS CHGMOD  /CHANGE MODE & REPEAT TILL BOTH MODES HAVE BEEN TESTED.
7527 5745 5233      JMP MODAGN+1
7528 5746 4753*     JMS CHGUNIT /CHANGE UNIT & REPEAT TILL BOTH UNITS HAVE BEEN TESTED.
7529 5747 5202      JMP MODAGN
7530 5750 4447      DONLOP
7531 5751 3140      DCA TSTNU
7532 5752 4443      CLEOP
7533
7534 5753 6014
7535 5754 6090
7536 5755 6440
7537 5756 6054
7538 5757 6056
7539 5760 6104
7540 5761 6072
7541 5762 6131
7542 5763 6111
7543 5764 6103
7544 5765 7774
7545 5766 7770
7546 5767 6053
7547 5770 2017
7548 5771 6071
7549 5772 6400
7550 5773 6052
7551 5774 6110
7552 5775 6017
7553 5776 6034
7554 5777 6041
7555              6000
7556
7557 6000 0000      CHGMOD, 0
7558 6001 1271      TAD MODVAR  /WHICH MODE WAS JUST TESTED?
7559 6002 7041      CIA
7560 6003 1303      TAD MRCON
7561 6004 7650      SNA CLA
7562 6005 5212      JMP ALRBM  /8 BIT MODE
7563 6006 4344      JMS FILVAR /12 BIT MODE - SWITCH TO 8 BIT MODE
7564 6007 6071      MODVAR     /COPY 8 BIT MODE CONSTANTS INTO MODE VARIABLES
7565 6010 6103      MRCON
7566 6011 5600      JMP I CHGMOD
7567 6012 2200      ALRBM, ISZ CHGMOD /MODE TESTING COMPLETE FOR UNIT
7568 6013 5600      JMP I CHGMOD
7569
7570              /ROUTINE TO CHANGE UNIT FROM A TO B & THEN REPEAT TEST.
    
```

```

7571 6014 0000      CHGUNIT,0
7572 6015 1252      TAD UNVAR  /WHICH UNIT WAS JUST TESTED?
7573 6016 7041      CIA
7574 6017 1264      TAD UNBCON
7575 6020 7650      SNA CLA
7576 6021 5232      JMP ALRBU  /UNIT B
7577 6022 4344      JMS FILVAR /UNIT A - SWITCH TO UNIT B
7578 6023 6052      UNVAR     /COPY UNIT B CONSTANTS INTO UNIT VARIABLES
7579 6024 6064      UNBCON
7580 6025 1377      TAD (LQPC /SETUP LQPC FOR SHIFT INST.
7581 6026 3776*     DCA SHIFTO
7582 6027 1377      TAD (LQPC
7583 6030 3775*     DCA SHIFTI
7584 6031 5614      JMP I CHGUNIT
7585 6032 2214      ALRBU, ISZ CHGUNIT /UNIT TESTING COMPLETE
7586 6033 5614      JMP I CHGUNIT
7587
7588              /INITIALIZE MODE VARIABLES FOR 12 BIT MODE
7589
7590
7591 6034 0000      INTMOD, 0
7592 6035 4344      JMS FILVAR /COPY MODE 12 CONSTANTS INTO MODE VARIABLES
7593 6036 6071      MODVAR
7594 6037 6076      M12CON
7595 6040 5634      JMP I INTMOD
7596
7597              /INITIALIZE UNIT VARIABLES FOR UNIT A
7598
7599 6041 0000      INTUN, 0
7600 6042 4344      JMS FILVAR /COPY UNIT A CONSTANTS INTO UNIT VARIABLES
7601 6043 6052      UNVAR
7602 6044 6057      UNACON
7603 6045 1374      TAD (LQRE /SETUP LQRE FOR SHIFT INSTRUCTION.
7604 6046 3776*     DCA SHIFTO
7605 6047 1374      TAD (LQRE
7606 6050 3775*     DCA SHIFTI
7607 6051 5641      JMP I INTUN
7608
7609              /UNIT VARIABLES
7610
7611 6052 0000      UNVAR, 0
7612 6053 0000      UNIT, 0    /RX01 UNIT UNDER TEST 0=UNIT A; 1=UNIT B
7613 6054 0000      UDONE, 0
7614 6055 0000      UXFR, 0
7615 6056 0000      UDATA, 0
7616 6057 0000      UOUT, 0
7617
7618              /UNIT A CONSTANTS
7619
7620 6058 0000      UNACON, 0
7621 6059 2000      2000
7622 6060 1000      1000
7623 6061 0040      0040
7624 6062 0040      0040
7625 6063 0200      0200
    
```

```

7626
7627 /UNIT B CONSTANTS
7628
7629 UNBCON,
7630 6064 0001 0001
7631 6065 4000 4000
7632 6066 0400 0400
7633 6067 0020 0020
7634 6070 0100 0100
7635
7636 /MODE VARIABLES
7637
7638 MODVAR,
7639 6071 0000 LCDATA, 0
7640 6072 0000 XDRDATA, 0 /DATA EXPECTED TO READ FROM INTERFACE REG AFTER SHIFT IN.
7641 6073 0000 SLCDA, 0
7642 6074 0000 SXDRDA, 0 /DATA TO BE SHIFTED INTO INTERFACE REG
7643 6075 0000 0
7644
7645 /MODE 12 CONSTANTS
7646
7647 M12CON,
7648 6076 4452 4452
7649 6077 0525 0525
7650 6100 4452 4452
7651 6101 7252 7252
7652 6102 0000 0000
7653
7654 /MODE 8 CONSTANTS
7655
7656 M8CON,
7657 6103 0125 DLCD8, 0125
7658 6104 0242 DXDR8, 0242
7659 6105 6500 6500
7660 6106 2720 2720
7661 6107 0000 0000
7662
7663 6110 0000 SHFCT, 0 /SHIFT COUNTER
7664
7665 /ROUTINE TO CHECK FOR PROPER DATA AT RXDATA PRIOR TO EACH SHIFT
7666 / OUT PULSE, NOTE: RXDATA IS LOOPED TO ERROR VIA SIMULATOR
7667
7668 6111 0000 TSTDAB, 0 /INITIALLY (WHEN SHIFT COUNT=0) SLCDA CONTAINS
7669 / LCDATA BITS(EXCEPT BIT 0) CONCATENATED
7670 / WITH PARITY.
7671 6112 7300 CLA CLL /
7672 6113 1273 TAD SLCDA /AC(0) CONTAINS CURRENT RXDATA H BIT
7673 6114 7004 RAL /
7674 6115 3273 DCA SLCDA /LINK CONTAINS EXPECTED DATA BIT
7675 6116 1310 TAD SHFCT /LOAD AC WITH SHIFT POSITION FOR ERROR DISPLAY
7676 6117 7420 SNL /
7677 6120 5325 JMP ZEXP /
7678 6121 6754 SER /
7679 6122 4450 ERROR /EXPECTED RXDATA BIT=1,ACTUAL=0
7680 / AC CONTAINS SHIFT COUNT
/NOTE: DATA AT RXDATA IS INVERTED.

```

```

7661 6123 7300 CLA CLL
7662 6124 5711 JMP I TSTDAB
7663 6125 6754 ZEXP, SER
7664 6126 7610 SKP CLA
7665 6127 4450 ERROR /EXPECTED RXDATA BIT=0,ACTUAL = 1
7666 / AC CONTAINS SHIFT COUNT.
7667 /NOTE: DATA AT RXDATA IS INVERTED.
7668 6130 5711 JMP I TSTDAB
7669
7670 /ROUTINE TO SETUP RXDATA PRIOR TO EACH SHIFT IN PULSE.
7671
7672 SHINDA, 0
7673 6131 0000 CLA CLL /INITIALLY (BEFORE ANY IN SHIFTING) SXDRDA CONTAINS
7674 6132 7300 / XDRDATA.
7675 6133 1274 TAD SXDRDA /AC(0) CONTAINS NEXT DATA BIT TO BE
7676 / SHIFTED IN.
7677 RAL /
7678 6135 3274 DCA SXDRDA /SAVE SHIFTED DATA IN
7679 6136 7630 SZL CLA /
7680 6137 1255 TAD UDATA /FORM PARALLEL I/O WORD FOR PROPER SHIFT IN
7681 6140 1373 TAD (0017 / DATA ON RXDATA.
7682 6141 1256 TAD UOUT /AT SAME TIME CHANGE OUTL TO 0.
7683 6142 1253 TAD UDONE /
7684 6143 5731 JMP I SHINDA
7685
7686 /ROUTINE TO COPY A SET OF 5 CONSTANTS INTO 5 VARIABLES.
7687
7688 FILVAR, 0
7689 6144 0000 TAD I FILVAR /GET VARIABLE LOCATION
7690 6145 1744 DCA VAR
7691 6146 3364 ISZ FILVAR
7692 6147 2344 TAD I FILVAR /GET CONSTANT LOCATION
7693 6150 1744 DCA CONST
7694 6151 3365 ISZ FILVAR
7695 6152 2344 TAD (-5
7696 6153 1372 DCA FVLCY /SET A LOOP COUNTER
7697 6154 3366 TAD I CONST /COPY CONSTANT INTO VARIABLE
7698 6155 1765 DCA I VAR
7699 6156 3764 ISZ CONST
7700 6157 2365 ISZ VAR
7701 6160 2364 ISZ FVLCY
7702 6161 2366 JMP FVLOOP /REPEAT FOR EACH OF 5 CONSTANTS
7703 6162 5355 JMP I FILVAR
7704 6163 5744
7705
7706 VAR, 0
7707 6164 0000 CONST, 0
7708 6165 0000 FVLCY, 0
7709
7710
7711
7712
7713
7714
7715
7716
7717
7718
7719
7720
7721
7722
7723
7724
7725
7726
7727
7728
7729
7730

```

```

7731
7732
7733 6172 7773
7734 6173 8017
7735 6174 6507
7736 6175 5662
7737 6176 5633
7738 6177 6504
          6200 *6200
7739 /*****
7740 /FIELD 1 INTERRUPT SKIP CHAIN
7741 /*****
7742
7743
7744 6200 3255 SKPICHN,DCA AC /GET HERE VIA INTERRUPT
7745 /SAVE AC.
7746 6201 6201 CDF 00
7747 6202 1777 TAD I (0
7748 6203 3000 DCA 0
7749 6204 6244 /SAVE PC AT TIME OF INT.
7750 6205 1120 RMF /RESTORE DF
7751 6206 7640 XR, TAD EXMIII /EXPECTING A XMIT INT?
7752 6207 5224 SZA CLA
7753 6210 1121 CRA, JMP CKXMPG /YES=VERIFY XMIT FLAG SET
7754 6211 7640 TAD ERECI /EXPECTING A RECV INT?
7755 6212 5227 SZA CLA
7756 6213 1135 JMP CKRFG /YES=VERIFY RECV FLAG SET
7757 6214 7640 TAD EPRNTI /EXPECTING A PRINTER INT.
7758 6215 5232 SZA CLA
7759 6216 1136 JMP CKPTFG /YES=VERIFY PRINT FLAG
7760 6217 7640 TAD ECONS1 /CAN CONSOLE PACKAGE INT. IN?
7761 6220 5741 SZA CLA
7762 6221 1000 JMP CKCON /YES=CHECK FOR CONSOLE FLAG.
7763 6222 4450 UNEINT, TAD 0
7764 /UNEXPECTED INTERRUPT
7765 /AC= PC AT TIME OF INT.
7766 6223 5246 UNEI, ERROR /CONT TO DETERMINE WHICH FLAGS ARE SET
          JMP SETIF
7767
7768 6224 6041 CKXMPG, TSF /OVERWRITTEN WITH PROPER DEVICE CODE - CHECK XMIT FLAG
7769 6225 5210 JMP CRA /NOT SET-INT, DUE TO SOME OTHER FLAG
7770 6226 5246 JMP SETIF /XMIT FLAG SET AS EXPECTED
7771
7772 6227 6031 CKRFG, KSF /OVERWRITTEN WITH PROPER DEVICE CODE - CHECK RECV FLAG
7773 6230 5221 JMP UNEINT /NOT SET-INT, DUE TO SOME OTHER FLAG
7774 6231 5246 JMP SETIF /REC FLAG SET AS EXPECTED
7775
7776 6232 6500 CKPTFG, LQSK /CHECK PRINT FLAG
7777 6233 7410 SKP
7778 6234 5246 JMP SETIF /SET
7779 6235 6661 PSKF
7780 6236 7410 SKP
7781 6237 5246 JMP SETIF /SET
7782 6240 5221 JMP UNEINT /PRINT FLAG NOT SET
7783
7784 6241 4445 CKCON, CHIKSF /CHECK FOR CONSOLE REQ

```

```

7785 6242 7410 SKP
7786 6243 5221 JMP UNEINT /SET-BUT IGNORE
7787 6244 1255 TAD AC /NOT SET
7788 6245 5400 JMP I 0 /CONT DIAG
7789
7790 6246 7240 SETIF, CLA CMA
7791 6247 3117 DCA INTFLG /SET PROG INT FLAG
7792 6250 4776 JMS RETURN
7793 6251 6254 INTRET
7794 6252 1255 TAD AC
7795 6253 5654 JMP I INTRET
7796
7797 6254 0000 INTRET, 0
7798 6255 0000 AC, 0
7799
7800 /DISPLAY ROUTINE FOR UNEXPECTED INTERRUPT IN FIELD 1
7801
7802 6256 4432 SUNEI, CICRLF
7803 6257 4434 CIPRNT
7804 6260 6303 UNETMS
7805 6261 4432 CICRLF
7806 6262 4775 JMS MITM /DISPLAY TEST NUMBER
7807 6263 1000 TAD 0
7808 6264 3774 DCA PCISAVE
7809 6265 4773 JMS MIPC /DISPLAY PC
7810 6266 1255 TAD AC
7811 6267 3772 DCA ACISAVE
7812 6270 4771 JMS M1AC /DISPLAY AC
7813 6271 4770 JMS M1FL /DISPLAY FLAGS
7814 6272 4432 CICRLF
7815 6273 4434 CIPRNT
7816 6274 6323 FSIMES
7817 6275 1367 TAD (CDI 10
7818 6276 6201 CDF 00
7819 6277 3760 DCA I (VCDI
7820 6300 6203 CDI 00
7821 6301 4233 JMS CKFLG /DISPLAY FLAGS SET
7822 6302 5765 JMP IEH1
7823
7824
7825 6303 2516 UNETMS, TEXT "UNEXPECTED INTERRUPT = FIELD 1"
          6304 0530
          6305 2005
          6306 0324
          6307 0504
          6310 4011
          6311 1024
          6312 0522
          6313 2225
          6314 2024
          6315 4055
          6316 4006
          6317 1105
          6320 1404
          6321 4061

```

```

7826 6322 0000
      6323 0014
      6324 0107
      6325 2340
      6326 2305
      6327 2472
      6330 0000
7827 6331 0413
      6332 2624
      6333 0255
      6334 0140
      6335 0601
      6336 1114
      6337 0504
      6340 5440
      6341 0611
      6342 0514
      6343 0440
      6344 6140
      6345 4000

7828
7829
7830
7831 6346 0000
7832 6347 7300
7833 6350 6201
7834 6351 1764
7835 6352 6211
7836 6353 5746
7837
7838

```

```

FSIMES,TEXT "FLAGS SET:"

ERRIMES,TEXT "DKVTB=A FAILED, FIELD 1 "

```

/ROUTINE TO GET HARDWARE OPTION WORD 3 (LOC 23) FROM FIELD 0.

```

XGHW3, 0
      CIA CLL
      CDF 00
      TAD I (23)
      CDF 10
      JMP I XGHW3

```

```

7839
7840 6364 0023
7841 6365 6655
7842 6366 6274
7843 6367 6213
7844 6370 6440
7845 6371 6424
7846 6372 6710
7847 6373 6531
7848 6374 6707
7849 6375 6523
7850 6376 2400
7851 6377 0000
      6400

7852
7853
7854
7855
7856
7857
7858
7859 6403 4257
7860 6401 4224
7861 6402 4232
7862 6403 4240
7863 6404 4432
7864 6405 5777
7865
7866
7867
7868 6406 4246
7869 6407 5777
7870
7871
7872
7873 6410 1776
7874 6411 3775
7875 6412 4246
7876 6413 4434
7877 6414 6537
7878 6415 1126
7879 6416 4435
7880 6417 4434
7881 6420 6546
7882 6421 1127
7883 6422 4435
7884 6423 5777
7885
7886
7887
7888 6424 0000
7889 6425 4434
7890 6426 6676
7891 6427 1774
7892 6430 4435

```

```

*6400
/*****
/ERROR DISPLAY FORMAT ROUTINES
/*****

```

/NORMAL DISPLAY = TN,PC,AC,MQ,FLAGS

```

NORDIS, JMS MSHDR           /PRINT ERROR HEADER & PC
          JMS MIAC           /PRINT AC
          JMS MIMQ           /PRINT MQ
          JMS MIFL           /PRINT FLAGS
          CICRLF
          JMP IEH1

```

/SLU DISPLAY = TN,PC,AC,SLU,BR

```

SLUDIS, JMS NORSLU
          JMP IEH1

```

/SPECIAL SLU DISPLAY = TN,PC,AC,SLU,BR,XMIT DATA,REC DATA

```

SSLUDIS,TAD DDERSLU
          DCA PCISAVE
          JMS NORSLU
          CIPRNT
          MESXM
          TAD SLUXMT
          CIPRT4
          CIPRNT
          MESRC
          TAD SLUREC
          CIPRT4
          JMP IEH1

```

/SUBROUTINES FOR DISPLAY ROUTINES

```

MIAC, 0 /DISPLAY AC MESSAGE
      CIPRNT
      MESIAC
      TAD ACISAVE
      CIPRT4

```

```

7893 6431 5624 JMP I MIAC
7894
7895 6432 0000 MIMG, 0 /DISPLAY MQ MESSAGE
7896 6433 4434 CIPRNT
7897 6434 6701 MESIMQ
7898 6435 1773 TAD MQ1SAVE
7899 6436 4435 CIPRT4
7900 6437 5632 JMP I MIMG
7901
7902 6440 0000 MIFL, 0 /DISPLAY FLAGS MESSAGE
7903 6441 4434 CIPRNT
7904 6442 6701 MESIFL
7905 6443 1772 TAD FL1SAVE
7906 6444 4435 CIPRT4
7907 6445 5640 JMP I MIFL
7908
7909 6446 0000 NORSLU, 0
7910 6447 4257 JMS MSHDR /PRINT HEADER & PC MESSAGE
7911 6450 4224 JMS MIAC /PRINT AC
7912 6451 4232 JMS MIMG /PRINT MQ
7913 6452 4240 JMS MIFL /PRINT FLAGS
7914 6453 4267 JMS MSLU /PRINT SLU#
7915 6454 4312 JMS MBR /PRINT BAUD RATE
7916 6455 4432 CICRLF
7917 6456 5646 JMP I NORSLU
7918
7919 6457 0000 MSHDR, 0 /DISPLAY MESSAGE HEADER & PC
7920 6460 4432 CICRLF
7921 6461 4434 CIPRNT
7922 6462 6331 ERRIMES
7923 6463 4432 CICRLF
7924 6464 4323 JMS MITN
7925 6465 4331 JMS MIPC
7926 6466 5657 JMP I MSHDR
7927
7928 6467 0000 MSLU, 0 /DISPLAY SLU #
7929 6470 4434 CIPRNT
7930 6471 6557 MESSLU
7931 6472 1771 TAD IOT10 /DETERMINE WHICH SLU IS BEING WORKED ON,
7932 6473 0370 AND (0030
7933 6474 7450 SNA
7934 6475 5307 JMP UX
7935 6476 1367 TAD (-10
7936 6477 7640 SZA CLA
7937 6500 7001 IAC /SLU #3
7938 6501 7001 IAC /SLU #2
7939 6502 7001 U1, IAC /SLU #1
7940 6503 3311 DCA SSLUN
7941 6504 1311 TAD SSLUN
7942 6505 4435 CIPRT4
7943 6506 5667 JMP I MSLU
7944
7945 6507 7300 UX, CLA CLL
7946 6510 5302 JMP U1
7947 6511 0000 SSLUN, 0
    
```

```

7948
7949 6512 0000 MBR, 0 /DISPLAY SLU BAUD RATE CONSTANT
7950 6513 4434 CIPRNT
7951 6514 6563 MESBR
7952 6515 1360 TAD (BDR1-1
7953 6516 1311 TAD SSLUN
7954 6517 3311 DCA SSLUN
7955 6520 1711 TAD I SSLUN
7956 6521 4435 CIPRT4
7957 6522 5712 JMP I MBR
7958
7959 6523 0000 MITN, 0 /DISPLAY TEST NUMBER MESSAGE
7960 6524 4434 CIPRNT
7961 6525 6554 MESITN
7962 6526 1140 TAD TSTNU
7963 6527 4435 CIPRT4
7964 6530 5723 JMP I MITN
7965
7966 6531 0000 MIPC, 0 /DISPLAY PC MESSAGE
7967 6532 4434 CIPRNT
7968 6533 6672 MESIPC
7969 6534 1775 TAD PC1SAVE
7970 6535 4435 CIPRT4
7971 6536 5731 JMP I MIPC
7972
7973
7974 6537 4040 MESXM, TEXT " XMIT DATA:"
7975 6540 3015
7976 6541 1124
7977 6542 4004
7978 6543 0124
7979 6544 6172
7980 6545 0000
7981 6546 4040 MESRC, TEXT " REC DATA:"
7982 6547 2205
7983 6550 0340
7984 6551 0401
7985 6552 2401
7986 6553 7200
7987 6554 4040 MESITN, TEXT " IN:"
7988 6555 2416
7989 6556 7200
7990 6557 4040 MESSLU, TEXT " SLU:"
7991 6560 2314
7992 6561 2572
7993 6562 0000
7994 6563 4040 MESBR, TEXT " BR:"
7995 6564 0222
7996 6565 7200
    
```

7979
7980
7981 6566 2025
7982 6567 7770
7983 6570 6030
7984 6571 2702
7985 6572 6712
7986 6573 6711
7987 6574 6710
7988 6575 6707
7989 6576 2470
7990 6577 6655
6600

*6600

/ CONSOLE ROUTINES

/THIS ROUTINE WILL DETERMINE WHAT TO DO WHEN A ERROR IS ENCOUNTERED
/WILL CHECK IF UNDER APT CONTROL,

8000
8001 6600 8000
8002 6601 6002
8003 6602 7000
8004 6603 3310
8005 6604 6004
8006 6605 3312
8007 6606 7501
8008 6607 3311
8009 6610 7340
8010 6611 1200
8011 6612 3307
8012 6613 6031
8013 6614 7410
8014 6615 7040
8015 6616 6201
8016 6617 3777
8017 6620 6211
8018 6621 4462
8019 6622 1124
8020 6623 3275
8021 6624 4465
8022 6625 0376
8023 6626 7640
8024 6627 5255
8025 6630 6037
8026 6631 1375
8027 6632 6043
8028 6633 4470
8029 6634 7300
8030 6635 6032
8031 6636 1307
8032 6637 7041

XCIERR, 0
IOF
OWIAPT, NOP /OVERWRITTEN WITH *JMP APTIER* IF RUNNING UNDER APT CONTROL
DCA ACISAVE /SAVE AC
GTF
DCA FLISAVE /SAVE THE FLAGS
MQA
DCA MQISAVE /SAVE THE MQ
CLA CLL CMA /SUBTRACT A 1 FOR TRUE LOCATION
TAD XCIERR /GET RETURN LOCATION
DCA PCISAVE /SAVE ADD OF ERROR CALL
KSF /CHECK FOR SLU #1 REC FLAG
SKP /SAVE STATE OF FLAG IS KSFLG (FLD 0)
CMA /KB1CHK CLEARS FLAG
CDF 00 /KSFLG USED BY UNEXPECTED INT, DISPLAY
DCA I (KSFLG
CDF 10
KB1CHK
TAD TESTF1
DCA TLOOP1
GETSR /GET PSR FROM FIELD 0
AND (0200 /INHIBIT ERROR TYPEOUT?
SZA CLA
JMP IEH1 /NO ERROR PRINTOUT
KLB /REMOVE LOOP AROUND ON SLU
TAD (0016 /SET SLU#1 BAUD RATE TO 9600 BITS/SEC
TSB
DELAY /WAIT FOR BAUD RATE CHANGE TO TAKE EFFECT
CLA CLL
KCC /CLEAR OUT REC FLAG-MAY BE SET IF ENTERING FROM A SLU ERROR.
TAD PCISAVE
CIA

8033 6640 1374
8034 6641 7650
8035 6642 5773
8036 6643 1307
8037 6644 7041
8038 6645 1372
8039 6646 7650
8040 6647 5771
8041 6650 1307
8042 6651 1370
8043 6652 7710
8044 6653 5767
8045 6654 5766
8046 6655 4465
8047 6656 7710
8048 6657 5262
8049 6660 4765
8050 6661 4430
8051 6662 4764
8052
8053 6663 4465
8054 6664 7004
8055 6665 7710
8056 6666 5675
8057 6667 4315
8058 6670 7000
8059
8060 6671 5600
8061 6672 4042
6673 2003
6674 7200
8062 6675 9000
8063 6676 4040
6677 0103
6700 7200
8064 6701 4040
6702 1521
6703 7200
8065 6704 4040
6705 0614
6706 7200
8066 6707 7777
8067 6710 7777
8068 6711 7777
8069 6712 7777
8070 6713 5714
8071 6714 7000
8072
8073
8074 6715 8000
8075 6716 7200
8076 6717 1311
8077 6720 7421
8078 6721 1312
8079 6722 7004

TAD (UNEI
SNA CLA
JMP SUNEI /USE UNEXPECTED INTERRUPT DISPLAY
TAD PCISAVE
CIA
TAD (SPE3
SNA CLA
JMP SSLUDIS /USE SPECIAL SLU DISPLAY FORMAT
TAD PCISAVE
TAD (-SLUEND
SPA CLA
JMP SLUDIS /USE SLU DISPLAY FORMAT
JMP NORDIS /USE NORMAL DISPLAY FORMAT
IEH1, GETSR /INHIBIT ERROR HALT?
SPA CLA
JMP C1BY2 /LEAVE
JMS SETBDR /NO-SET BAUD RATES IN ALL SLUS IN CASE OPERATOR LEAVES DIAGNOSTIC
C1SMIT /GO TO THE INQUIRE ROUTINE
C1BY2, JMS RESTBR /RESTORE BAUD RATES IN ALL SLUS TO
/ STATES PRIOR TO ERROR,
GETSR /LOOP ON ERROR?
RAL /TEST BIT 1
SPA CLA
JMP I TLOOP1 /LOOP
JMS CIGET /RESTORE MQ AND LINK,CLEAR AC
NOP /LEAVE INT, SYS DISABLED
/NEXT TEST WILL EN IT
JMP I XCIERR
MES1PC, TEXT " PC,"
6673 2003
6674 7200
TLOOP1,0
MES1AC, TEXT " AC,"
6677 0103
6700 7200
MES1MQ, TEXT " MQ,"
6702 1521
6703 7200
MES1FL, TEXT " FL,"
6705 0614
6706 7200
PCISAVE,7777
ACISAVE,7777
MQISAVE,7777
FLISAVE,7777
APIOW1, JMP I ,+1
APTIER
CIGET, 0
CLA
TAD MQISAVE
MOL /RESTORE THE MQ
TAD FLISAVE
RAL /RESTORE THE LINK

```

8080 6723 7200 CLA
8081 6724 5715 JMP I C1GET
8082
8083
8084
8085
8086
8087
8088
8089
8090
8091 6725 0000 /APR/ ROUTINE TO NOTIFY APT THAT THE PROGRAM IS RUNNING OK.
8092 6726 6002 APTOK1, 0 /APR/
8093 6727 7200 IOF /APR/
8094 6730 1347 CLA /APR/
8095 6731 3351 TAD APTIM1 /APR/DELAY 100MS.
8096 6732 1350 DCA APTCT1 /APR/
8097 6733 3352 TAD APTIM2 /APR/
8098 6734 2352 DCA APTCT2 /APR/
8099 6735 5334 ISZ APTCT2 /APR/
8100 6736 2351 JMP ,+1 /APR/
8101 6737 5332 ISZ APTCT1 /APR/
8102 6740 6224 RIF ,=5 /APR/
8103 6741 1363 TAD (6201 /APR/AC=IF.
8104 6742 3343 DCA ,+1 /APR/CREATE A CDF INST.
8105 6743 6201 CDF /APR/MODIFY NEXT CDF INST.
8106 6744 6272 CIF 70 /APR/(MODIFIED CDF) DF=IF.
8107 6745 4762 JMS 6500 /APR/CALL APT - 'PROG OK'.
8108 6746 5725 JMP I APTOK1 /APR/RTN FROM APT - RTN TO CALL+1.
8109
8110 6747 7776 APTIM1, =2 /APR/
8111 6750 0000 APTIM2, 0 /APR/
8112 6751 0000 APTCT1, 0 /APR/
8113 6752 0000 APTCT2, 0 /APR/
8114
8115
8116 6762 6500
8117 6763 6201
8118 6764 3214
8119 6765 3150
8120 6766 6400
8121 6767 6406
8122 6770 4447
8123 6771 6410
8124 6772 2466
8125 6773 6256
8126 6774 6222
8127 6775 0016
8128 6776 6200
8129 6777 0040
8130
8131
8132
8133

```

PAGE

```

8134 /APR/ ROUTINE TO HANDLE ERRORS UNDER APT CONTROL.
8135
8136 7000 6002 APTIER, IOF /APR/
8137 7001 7200 CLA /APR/
8138 7002 6037 KLB /APR/ REMOVE LOOP AROUND ON SLU'S
8139 7003 6224 RIF /APR/AC=IF.
8140 7004 1377 TAD (6201 /APR/CREATE A CDF INST.
8141 7005 3216 DCA ,+3 /APR/MODIFY NEXT CDF INST.
8142 7006 7240 CLA CMA /APR/
8143 7007 1776 TAD XC1ERR /APR/AC=ERROR PC.
8144 7010 6201 CDF /APR/(MODIFIED CDF) DF=IF.
8145 7011 6272 CIF 70 /APR/IF=FIELD 7.
8146 7012 5775 JMP 6520 /APR/CALL APT - 'ERROR'.
8147
8148
8149
8150
8151
8152
8153 7013 0000 /ROUTINE USED FOR CONSOLE SWITCH REGISTER CHANGES
8154 7014 7200 XC1SW, 0
8155 7015 4434 CLA
8156 7016 7343 SRIGEST, C1PRNT /PRINT SR QUESTION
8157 7017 4465 SR1MSG /POINTER TO MESSAGE
8158 7020 4435 GETSR /GET PSR FORM FIELD 0
8159 7021 7346 C1PRT4 /PRINT THE 4 DIGITS
8160 7022 3301 CLA CLL CMA RTL /SETUP A COUNTER TO ACCEPT 4 DIGITS
8161 7023 1374 DCA TTYICNT /SAVE THE COUNTER
8162 7024 3227 TAD (CH1R0 /GET POINTER FOR FIRST CHARACTER
8163 7025 4437 DCA CHG1CHR /SAVE THE POINTER FOR DIGITS
8164 7026 0001 LISWF1 /WAIT FOR KEYBOARD INPUT
8165 7027 7045 CHG1CHR, CH1R0 /CHECK FOR A OCTAL DIGIT
8166 7030 7566 /THIS LOCATION WILL GET MODIFIED
8167 7031 7337 =212 /CHECK FOR LINE FEED
8168 7032 7563 COMPST /LINE FEED TYPED- RETURN TO START
8169 7033 7065 =215 /CHECK FOR CARRAIGE RETURN
8170 7034 7575 RE1TYPE /RETYPE SR AND COMT IF DIGITS TYPED
8171 7035 7102 =203 /CHECK FOR A CONTROL C
8172 7036 7555 C1RM /CONTROL C TYPED -RETURN TO MONITOR
8173 7037 7144 =223 /CHECK FOR A CONTROL S
8174 7040 0000 CTRS1 /WAS CONTROL S WAIT FOR "Q OR "C
8175 7041 7042 0 /NONE OF ABOVE CHARACTERS=ILLEGAL CHAR
8176 7042 4434 +1 /GO TO NEXT ADDRESS TO PRINT ?
8177 7043 7346 C1PRNT /GO PRINT ?
8178 7044 5215 QES1MK /POINTER TO ? MESSAGE
8179 7045 6201 JMP SRIGEST /RETURN AND ASK QUESTION AGAIN
8180 7046 3773 CH1R0, CDF 00 /SAVE THE LEAST SIGNIFICANT BIT
8181 7047 6211 DCA I (PSR
8182 7050 1372 CDF 10
8183 7051 3227 TAD (CH1R1 /UPDATA POINTER FOR CHARACTERS 2 3 4
8184 7052 5225 DCA CHG1CHR /SAVE THE POINTER ADDRESS
8185 7053 3300 JMP CHG1CHR-2 /RETURN FOR NEXT CHARACTER INPUT
8186 7054 4465 CH1P1, DCA SAV1CHR /SAVE THE CHARACTER TYPED
8187 7055 7106 GETSR /GET THE VALUE OF SR
8188 7056 7004 CLL RTL /MOVE IT INTO NEXT POSITION

```

```

8189 7057 1300 TAD SAVICHR /ADD NEW CHARACTER TO IT
8190 7060 6201 COF 00
8191 7061 3773 DCA I (PSR
8192 7062 6211 COF 10
8193 7063 2301 ISZ TTYICNT /DONE ALL 4 CHARACTERS
8194 7064 5225 JMP CHGICHR-2 /NO GET NEXT INPUT FROM KEYBOARD
8195 7065 1374 RE1TYPE,TAD (CH1R0 /GET POINTER TO SEE IF SR ECHOED
8196 7066 7041 CIA /NEGATE THE POINTER
8197 7067 1227 TAD CHGICHR /GET THE POINTER STORED
8198 7070 7650 SNA CLA /ECHO VALUE OF SR?
8199 7071 5613 JMP I XC1SW /NO-ONLY CR WAS TYPED-USE ORIGINAL VALUE
8200 7672 4434 C1PRNT /RE-ECHO VALUE TYPED
8201 7073 7343 SR1MSG /POINTER TO SR MESSAGE
8202 7074 4465 GETSR /GET PSR FORM FIELD 0
8203 7075 4435 C1PRT4 /PRINT THE 4 OCTAL DIGITS
8204 7076 4432 C1CRLF /ISSUE A CR AND LF
8205 7077 5613 JMP I XC1SW /RETURN TO PROGRAM
8206
8207 7100 0000 SAVICHR,0
8208 7101 0000 TTYICNT,0
8209
8210
8211 7102 6007 C1RM, CAF
8212 7103 6203 CDI 00 /CHANGE INST AND DATA FIELD TO 0
8213 7104 5705 JMP I ,+1 /GOTO 7600 OF THAT FIELD
8214 7105 7600 7600 /MONITOR STARTING ADDRESS
8215
8216
8217
8218
8219
8220
8221
8222
8223 7106 0000
8224 7107 6031 XCHIKSF,0
8225 7110 5336 KSF /SKIP ON CONSOLE RECEIVE FLAG
8226 7111 6201 JMP NORF /RECEIVE FLAG NOT SET RETURN TO PROGRAM
8227 7112 1771 CDF 00
8228 7113 6211 TAD I (MCW2 /CHECK TO SEE IF CONSOLE WAS ACTIVE
8229 7114 0370 CDF 10
8230 7115 7650 AND (4000 /(/VERSUS APT IN CONTROL),
8231 7116 5321 SNA CLA
8232 7117 6032 JMP ,+3 /ACTIVE CONSOLE-CHECK FOR "C OR "G
8233 7120 5706 KCC /APT ACTIVE,CLEAR CONSOLE RECEIVE FLAG
8234 7121 4767 JMP I XCHIKSF /RETURN TO PROGRAM
8235 7122 4437 JMS SISUBL /SAVE SUBROUTINE LINKAGE
8236 7123 7575 LISMF1 /CHECK THE KEYBOARD CHARACTER
8237 7124 7102 -203 /CODE FOR "C
8238 7125 7571 C1RM /WAS A CONTROL C-EXIT TO MONITOR
8239 7126 7140 -207 /CODE FOR "G
8240 7127 7555 CNTRIG /WAS "G ECHO CHAR-ENTER SR QUESTION
8241 7130 7146 -223 /CHECK FOR A CONTROL S
8242 7131 0000 CNT1S1 /WAS A CONTROL S WAIT FOR "Q OR "C
8243 7132 7133 0 /CHAR WAS NOT "C OR " G
,+1 /ECHO CHAR AND QUESTION MARK

```

/*****

/ROUTINE TO CHECK FOR CONSOLE RECEIVE FLAG

```

8244 7133 1366 TAD (277 /PRINT ?
8245 7134 4442 TYPE1
8246 7135 5347 JMP RILK /RETURN TO PROGRAM
8247 7136 2306 NORF, ISZ XCHIKSF
8248 7137 5706 JMP I XCHIKSF
8249
8250 7140 4434 CNTRIG, C1PRNT /PRINT "G AND CR LF
8251 7141 7350 UPARR1 /POINTER TO MESSAGE
8252 7142 4213 JMS XC1SW /GO ASK THE SR QUESTION
8253 7143 5347 JMP RILK /RETURN TO THE PROGRAM
8254
8255 7144 4765 CTRS1, JMS WAITIQC /GO WAIT FOR A CONTROL Q OR C
8256 7145 5225 JMP CHGICHR-2 /GO WAIT FOR NEXT CHAR
8257
8258 7146 4765 CNT1S1, JMS WAITIQC /WAIT FOR A CONTROL Q OR C
8259 7147 4764 RILK, JMS RISUBL /RESTORE SUBROUTINE LINKAGE
8260 7150 5706 JMP I XCHIKSF /RETURN TO PROGRAM
8261 7151 0004 FILLR1, 0004 /SET TO NUMBER OF FILLERS REQUIRED.
8262
8263
8264
8265
8266
8267
8268
8269
8270 7164 7547
8271 7165 7506
8272 7166 0277
8273 7167 7537
8274 7170 4000
8275 7171 0022
8276 7172 7053
8277 7173 0020
8278 7174 7045
8279 7175 6520
8280 7176 6600
8281 7177 6201
7200
8282 PAGE
8283 /TYPE A CR AND LF WITH NUMBER OF FILLERS
8284 /AS DETERMINED BY LOCATION "FILLR1"
8285 7200 0000 XC1CRLF,0 /CALL BY "CRLF"
8286 7201 7200 CLA
8287 7202 1377 TAD (215
8288 7203 4442 TYPE1
8289 7204 1776 TAD FILLR1
8290 7205 7040 CMA
8291 7206 3214 DCA XORS1
8292 7207 1375 TAD (212
8293 7210 4442 TYPE1
8294 7211 2214 ISZ XORS1
8295 7212 5210 JMP ,=2
8296 7213 5600 JMP I XC1CRLF
8297

```



```

0298 7214 0000 XORS1, 0
0299
0300 /PRINT 2 SPACES
0301
0302 7215 0000 SPAC2, 0 /CALL BY "SPAC21"
0303 7216 4434 CIPRNT
0304 7217 7221 .+2
0305 7220 5615 JMP I SPAC2
0306 7221 4P40 4040
0307 7222 0000 0000
0308
0309
0310 /COMPARE INPUT TO LIST FOLLOWING CALL
0311 /INPUT ONE CHARACTER IF AC=0
0312 /USE LAST INPUT IF AC NON ZERO
0313
0314 7223 0000 XLISN1, 0 /CALL BY "LISN1"
0315 7224 7640 SZA CLA /USE LAST INPUT SINCE AC NOT ZERO
0316 7225 5253 JMP LISN11
0317 7226 6031 KSF
0318 7227 5226 JMP .-1
0319 7230 6030 KRB
0320 7231 0374 AND (177
0321 7232 1373 TAD (200
0322 7233 3305 DCA CHAREC
0323 7234 1305 TAD CHAREC
0324 7235 1372 TAD (-212
0325 7236 7450 SNA /IS IT A LF?
0326 7237 5243 JMP .+4 /YES
0327 7240 1371 TAD (-3
0328 7241 7640 SZA CLA /IS IT A CR?
0329 7242 5245 JMP .+3 /NO
0330 7243 4432 C1CRLF
0331 7244 5253 JMP LISN11
0332 7245 1305 TAD CHAREC /GET THE CHAR
0333 7246 1370 TAD (-223 /CHECK FOR A CONTROL S
0334 7247 7650 SNA CLA /WAS IT A CONTROL S
0335 7250 5253 JMP LISN11 /YES-DO NOT ECHO CHARACTER
0336 7251 1305 TAD CHAREC
0337 7252 4442 TYPE1 /PRINT THE CHARACTER
0338 7253 1623 LISN11, TAD I XLISN1 /GET TEMPRE VALUE
0339 7254 2223 ISZ XLISN1
0340 7255 7450 SNA /EXIT?
0341 7256 5264 JMP LISN31 /YES
0342 7257 7500 SMA
0343 7260 5274 JMP LIS1NUM /LOOK FOR OCTAL NUMBER
0344 7261 1305 TAD CHAREC /COMPARE
0345 7262 7640 SZA CLA /EQUAL?
0346 7263 5271 JMP LISN21 /NO
0347 7264 3714 LISN31, DCA XORS1
0348 7265 1623 TAD I XLISN1
0349 7266 3223 DCA XLISN1
0350 7267 1214 TAD XORS1
0351 7270 5623 JMP I XLISN1 /AC IS ZERO UNLESS OCTAL NUMBER
0352 7271 7200 LISN21, CLA

```

```

0353 7272 2223 ISZ XLISN1
0354 7273 5253 JMP LISN11
0355 7274 7200 LIS1NUM,CLA /LOOK FOR OCTAL NUMBER
0356 7275 1305 TAD CHAREC
0357 7276 1367 TAD (-270
0358 7277 7500 SMA /IS IT LESS THAN 0?
0359 7300 5271 JMP LISN21 /NO, SO NOT AN OCTAL NUMBER
0360 7301 1360 TAD (10
0361 7302 7510 SPA /IS IT GREATER THAN ZERO?
0362 7303 5271 JMP LISN21 /NO, SO NOT A NUMBER
0363 7304 5264 JMP LISN31
0364
0365 7305 0000 CHAREC, 0
0366
0367
0368
0369
0370 /*****
0371
0372 /END OF PASS ROUTINE
0373
0374
0375 7306 0000 XC1EOP, 0
0376 7307 6002 IOF
0377 7310 4432 C1CRLF / CONTROL TO JMP APTJ01,
0378 7311 4434 CIPRNT
0379 7312 7327 EOP1MES
0380 7313 2137 ISZ PASSNO /UPDATE PASS COMPLETED NUMBER
0381 7314 7000 NOP
0382 7315 1137 TAD PASSNO
0383 7316 4435 C1PRT4
0384 7317 4432 C1CRLF
0385 7320 4465 GETSR /CONTINUE?
0386 7321 0365 AND (0400
0387 7322 7640 SZA CLA
0388 7323 5337 JMP COMPST /NO-ASK SWITCH REG QUESTION,
0389 7324 4515 APTREP /IF UNDER APT CONTROL REPORT OK STATUS TO APT
0390 7325 6203 CDI 00
0391 7326 5764 JMP COREST
0392 7327 0516 EOP1MES,TEXT "END OF PASS "
0393 7330 0440
0394 7331 1706
0395 7332 4020
0396 7333 0123
0397 7334 2340
0398 7335 4040
0399 7336 4000
0399 7337 6203 COMPST, CDI 00
0399 7340 5773 JMP START
0399
0399 7341 6203 C1RST, CDI 00
0399 7342 5764 JMP COREST
0399
0399 7343 4323 SR1MESG, TEXT "#SR#"
0399 7344 2275

```

8400 7345 8200
 8401 7346 7743
 8402 7347 8200
 8403 7350 3607
 8404 7351 4300
 8405 7364 8201
 8406 7365 8400
 8407 7366 8010
 8408 7367 7510
 8409 7370 7555
 8410 7371 7775
 8411 7372 7566
 8412 7373 8200
 8413 7374 8177
 8414 7375 8212
 8415 7376 7151
 8416 7377 8215
 8417 7400

QESIMK, TEXT "78"
 UPARR1, TEXT "G"
 PAGE

/TYPE THE ASCII CHARACTER IN THE AC

8422 7400 8800
 8423 7401 3213
 8424 7402 7800
 8425 7403 4273
 8426 7404 1213
 8427 7405 6040
 8428 7406 7200
 8429 7407 6841
 8430 7410 5207
 8431 7411 6042
 8432 7412 5600
 8433 7413 8000
 8434 7414 8000
 8435 7415 7200
 8436 7416 4462
 8437 7417 1614
 8438 7420 3252

X1TYPE, 0 /CALL BY "TYPE"
 /SAVE THE CHARACTER
 /OVERWRITTEN IF RUNNING UNDER APT CONTROL,
 /CONSOLE ACTIVE-CHECK FOR CONTROL S
 /GET THE CHARACTER SAVED AND PRINT
 DCA CHAR1
 NOP/JMP I X1TYPE
 JMS CNTRIS
 TAD CHAR1
 TIS
 CLA
 TSF
 JMP *-1
 TCF
 JMP I X1TYPE
 APT1CON=JMP I X1TYPE
 CHAR1, 0
 MESAG, 0 /CALL BY "MESSAGE"
 CLA
 KB1CHK
 TAD I MESAG
 DCA FOR10CK

/PRINT PACKED ASCII TEXT TERMINATED BY /SIX=BIT 00

8451 7421 2214
 8452 7422 1652
 8453 7423 7012
 8454 7424 7012
 8455 7425 7012
 8456 7426 4233
 8457 7427 1652
 8458 7430 4233
 8459 7431 2252
 8460 7432 5222
 8461 7433 8000
 8462 7434 8377
 8463 7435 7450
 8464 7436 5614
 8465 7437 1376
 8466 7440 7450
 8467 7441 5250
 8468 7442 1375
 8469 7443 7510
 8470 7444 1374
 8471 7445 1373
 8472 7446 4442
 8473 7447 5633
 8474 7450 4432
 8475 7451 5633
 8476
 8477
 8478
 8479 7452 8000
 8480
 8481
 8482
 8483
 8484 7453 8000
 8485 7454 3252
 8486 7455 1252
 8487 7456 7012
 8488 7457 7012
 8489 7460 7012
 8490 7461 4444
 8491 7462 1252
 8492 7463 4444
 8493 7464 4440
 8494 7465 5653
 8495
 8496
 8497 7466 8000
 8498 7467 8372
 8499 7470 1371
 8500 7471 4442
 8501 7472 5666

ISZ MESAG /SET UP RETURN
 TAD I FOR10CK
 RTR
 RTR
 RTR
 JMS MES1F
 TAD I FOR10CK
 JMS MES1F
 ISZ FOR10CK
 JMP *-10
 MES1F, 0
 AND (77 /TERMINATOR (00)?
 SNA /YES
 JMP I MESAG
 TAD (=43
 SNA /CRLF?
 JMP +7 /YES
 TAD (3
 SPA /200 OR 300
 TAD (100 /300
 TAD (240 /200
 TYPE1
 JMP I MES1F
 CICRLF
 JMP I MES1F
 FOR10CK,0
 /PRINT FOUR OCTAL NUMBERS IN AC 11 THRU 0 FOLLOWED /BY TWO SPACES
 X1PRN4, 0 /CALL BY "PRN4"
 DCA FOR10CK
 TAD FOR10CK
 RTR
 RTR
 RTR
 PRN21
 TAD FOR10CK
 PRN21
 SPAC21
 JMP I X1PRN4
 /
 X1PRN1, 0 /CALL BY "PRN1"
 AND (7
 TAD (260
 TYPE1
 JMP I X1PRN1

```

0506 /ROUTINE TO CHECK FOR A CONTROL S WHILE TYPING OUT MESSAGES
0507 /TO EXIT ROUTINE IF A CONTROL S WAS TYPED-A CONTROL Q OR C MUST BE
0508 /INPUTTED ON THE KEYBOARD
0509
0510 7473 0000 CNTRIS, 0
0511 7474 6031 KSF /SKIP ON CONSOLE KEYBOARD FLAG
0512 7475 5673 JMP I CNTRIS /RETURN TO TYPE ROUTINE-FLAG NOT SET
0513 7476 6034 KRS /READ THE CHARACTER STATICALLY
0514 7477 0370 AND (177 /MASK TO 7 BIT ASCII
0515 7500 1367 TAD (-23 /CHECK FOR A CONTROL S
0516 7501 7640 SZA CLA /WAS IT A CONTROL S
0517 7502 5673 JMP I CNTRIS /NO-RETURN WITH KEYBOARD FLAG STILL SET
0518 7503 6032 KCC /CLEAR KEYBOARD FLAG FROM *S
0519 7504 4306 JMS WAIT1QC /WAIT FOR CONTROL Q OR C
0520 7505 5673 JMP I CNTRIS /RETURN TO PRINT MESSAGE BEING TYPED
0521
0522 7506 0000 WAIT1QC,0 /ROUTINE TO WAIT FOR CONTROL Q OR C
0523 7507 6031 KSF /WAIT FOR A CONTROL Q OR C TO EXIT
0524 7510 5307 JMP *-1 /
0525 7511 6036 KRB /READ THE CHARACTER TYPED
0526 7512 0370 AND (177 /MASK TO 7 BIT ASCII
0527 7513 1366 TAD (-3 /CHECK FOR A CONTROL C
0528 7514 7450 SNA /WAS IT A CONTROL C?
0529 7515 5765 JMF CIRM /YES-RESTORE MONITOR AND RETURN
0530 7516 1364 TAD (-7 /CHECK FOR A LINE FEED CHARACTER
0531 7517 7450 SNA /WAS IT A LINE FEED
0532 7520 5763 JMP COMPST /YES GO RESTART THE PROGRAM
0533 7521 1364 TAD (-7 /CHECK FOR A CONTROL Q *Q
0534 7522 7640 SZA CLA /WAS IT A CONTROL Q
0535 7523 5307 JMP WAIT1QC+1 /NO-WAIT FOR APPROPRIATE KEY
0536 7524 5706 JMP I WAIT1QC /RETURN TO WHENCE IT CAME
0537
0538
0539 /PRINT THE TWO OCTAL NUMBERS IN THE AC 6 THRU 11
0540
0541 7525 0000 X1PRN2, 0 /CALL BY "PRN21"
0542 7526 3336 DCA TWO10CK
0543 7527 1336 TAD TWO10CK
0544 7530 7012 RTR
0545 7531 7010 RAR
0546 7532 4441 PRN11
0547 7533 1336 TAD TWO10CK
0548 7534 4441 PRN11
0549 7535 5725 JMP I X1PRN2
0550
0551 7536 0000 TWO10CK,0
0552
0553 /ROUTINE TO SAVE SUBROUTINE LINKAGES WHICH MAY GET DESTROYED BY KEYBOARD
0554 / INTERVENTION CHECK
0555
0556 7537 0000 SISUBL, 0
0557 7540 1762 TAD XCHKSF
0558 7541 3357 DCA LISAV1
0559 7542 1214 TAD MESAG
0560 7543 3360 DCA LISAV2

```

```

0561 7544 1753 TAD X1PRN4
0562 7545 3361 DCA LISAV3
0563 7546 5737 JMP I SISUBL
0564
0565 /ROUTINE TO RESTORE LINKAGES SAVED BY SISUBL ROUTINE
0566
0567 7547 0000 RISUBL, 0
0568 7550 1357 TAD LISAV1
0569 7551 3762 DCA XCHKSF
0570 7552 1360 TAD LISAV2
0571 7553 3214 DCA MESAG
0572 7554 1361 TAD LISAV3
0573 7555 3253 DCA X1PRN4
0574 7556 5747 JMP I RISUBL
0575
0576 7557 0000 LISAV1, 0
0577 7560 0000 LISAV2, 0
0578 7561 0000 LISAV3, 0
0579
0580 7562 7106
0581 7563 7337
0582 7564 7771
0583 7565 7102
0584 7566 7775
0585 7567 7755
0586 7570 0177
0587 7571 0260
0588 7572 6027
0589 7573 0240
0590 7574 0100
0591 7575 0003
0592 7576 7735
0593 7577 0077
0594 0177 0163
0595 0000

```

FIELD 0

0595 2700 *200
 0596
 0597 8888

AC	6255	BYT7	0007	CHGCHR	7032	CONTT	5704
ACISAV	6710	BITS	0065	CHGMOD	6000	CPUDIS	6422
ACAIOT	5120	BLANK	0130	CHGUNT	6014	CPUDN	2134
ACCERR	5054	BRMSB	0036	CHKKB	2237	CPUMES	2210
ACDATA	0067	BSTLP	0012	CHKKSF	4516	CPUSDI	6400
ACL	7701	BSW	7002	CIFCK	3365	CPUT1	0212
ACSAVE	6675	CIBY2	6662	CIFCK1	3444	CPUT10	1131
ACWAS	0054	CICRLF	4432	CIFJMP	3346	CPUT11	1200
ADDTIM	2357	CIEOP	4443	CIFJMS	3421	CPUT12	1235
ADRS	4446	CIGST	6715	CIFJPL	3340	CPUT13	1275
AGAIN1	3333	CIPRNT	4434	CIFJSL	3413	CPUT14	1425
AGAIN2	3406	CIPRT4	4435	CIFOK	3400	CPUT15	1462
ALRSM	6012	CIREST	0200	CIFOK1	3440	CPUT2	0400
ALRSU	6032	CIRM	7102	CKBRSG	2271	CPUT3	0501
APIOW1	6713	CIRST	7341	CKCON	6241	CPUT4	0600
APTICO	5600	CISWIT	4436	CKFLG	6233	CPUT5	0712
APTIER	7000	CSAPT	4532	CKIMOD	5677	CPUT6	1026
APT3CK	2726	CSBRLF	4521	CKINDA	5704	CPUT7	1071
APTCON	7333	CSERR	4533	CKMF	3001	CRA	6210
APTCT1	6751	CSGST	6703	CKMODE	5600	CRS1	7144
APTCT2	6752	CSH3	4540	CKPTFG	6232	CUIPTR	5121
APTCTX	6756	CSPRNT	4522	CKRFG	6227	DAT	0073
APTCTY	6757	CSPT4	4523	CKXMPG	6224	DATPAT	0177
APTER	7000	CSRC	4537	CLCL	6136	DATSLU	2421
APTHW3	6143	CSREST	0201	CLCON	5713	DDERSL	2470
APTIM1	6747	CSRH	7105	CLKCT	5712	DECPRT	6104
APTIM2	6750	CSSTAR	4534	CLKERR	5715	DELAY	4470
APTIMX	6754	CSWIT	4524	CLKIR	5674	DELAY1	3251
APTIMY	6755	CAB	3366	CLLE	6135	DESLU	2462
APTOK	6732	CAC	3367	CLOOPA	3240	DF0	3013
APTOK1	6725	CAD	3363	CLREIF	4455	DF1	3025
APTOH1	6701	CAE	3437	CLREPI	4471	DF2	3033
APTR	2513	CAF	6007	CLRERI	4457	DF3	3041
APTREP	4515	CAG	4625	CLREXI	4456	DF4	3047
ATIMLS	1466	CAM	7621	CLSK	6137	DF7	3017
ATIMMS	1467	CAMTST	1000	CLSKWT	4541	DPLD	3207
BASEP	6055	CCDF	3300	CLUPEX	1547	DLCD0	6103
BDR1	0026	CDELTA	5714	CNT	0122	DLOP	2423
BDR2	0027	CDI	6203	CNT1	0123	DNF7	3226
BDR3	0030	CFSLU	4512	CNT1B1	7146	DONLOP	4447
BDRATE	3124	CH1KB	3320	CNTR1G	7140	DONT17	1435
BDNRN	3200	CH1KSF	4445	CNTR1S	7473	DOSIMU	1056
BDRSQ	2322	CH1R0	7045	CNTRLG	7141	DOSLU	6025
BDTA	4505	CH1R1	7053	CNTRLS	7506	DOSWAP	2014
BEGBST	6000	CH1REC	7311	CNTRS	7145	DVCTAB	3045
BEGEXM	3000	CHAR	7334	CNTRS1	7147	DXDR0	6104
BEGRTC	5400	CHAR1	7413	COMP	0134	E0	5100
BIT11	0072	CHAREC	7305	COMPST	7337	E0PRE	5026
BIT3	0062	CHARLP	2015	CONCH	4026	E1	5111
BIT4	0063	CHARR0	7050	CONCHK	4020	E10	5142
BIT5	0064	CHARR1	7054	CONST	6165	E100	5460
BIT6	0102	CHG1CH	7027	CONTDG	4512	E11	5126

E110	5503	EXISET	2224	HLTS	3314	JMSLOC	1435
E120	5535	EXI10A	4481	IR0	3064	K0	2416
E121	5540	EXMLTI	0120	IB1	3071	K1	0041
E122	5543	EXMT1	3010	IB2	3104	K10	0042
E124	5527	EXMT10	4200	IB3	3117	K100	0044
E1PRE	5071	EXMT11	4400	IB7	3143	K200	0045
E2	5114	EXMT12	4600	IBSF	3276	K212	7216
E20	5204	EXMT13	4636	IBSF1	3402	K215	7215
E21	5206	EXMT14	5000	ICDI	4244	K240	7462
E22	5210	EXMT2	3056	IEH	6646	K2525	0046
E23	5212	EXMT3	3200	IEH1	6655	K260	7504
E24	5214	EXMT4	3277	IDF	4411	K3	7461
E25	5216	EXMT5	3445	IE0	3132	K4000	0047
E26	5225	EXMT6	3600	INIT	6757	K5252	0050
E27	5232	EXMT7	4000	INIT1	3077	K5253	0051
E28	5237	EXS17	2145	INITL	4232	K7	7505
E2PRE	5031	FDWRD	4006	INGL1	4205	K7600	1511
E3	5117	FILCOR	4052	INSTR	1515	K77	0043
E40	5320	FILDER	7152	INSTR	0105	K7700	0052
E42	5274	FILDR1	7151	INTEN1	1400	K7777	0053
E4PRE	5074	FILVAR	6144	INTPLG	0117	KB1CHK	4462
E50	5336	FIXL	4520	INTMOD	6034	KBCHK	4517
E51	5343	FIXLKG	5217	INTR	6756	KCC	6032
E52	5350	FL1SAV	6712	INTRET	6254	KCC1	6302
E53	5341	FLDN	5545	INTRTN	1514	KCC2	6322
E54	5346	FLMES	5242	INTST	4535	KCDF	0155
E55	5353	FLSAYC	6677	INTUN	6041	KCF	6030
E56	5334	FOR10C	7452	IOF	6002	KCF1	6300
E60	5414	FOROCK	7463	ION	6001	KCF2	6320
E61	5420	FRET	4526	IOT0	2601	KFLD0	0114
E62	5422	FSIMES	6323	IOT1	2606	KIE	6035
E63	5424	FSAV	3511	IOT10	2702	KIE1	6305
E70	5443	FSMES	6331	IOT11	2710	KIE2	6325
EAC	0131	FVLCT	6166	IOT12	2715	KIOT	5117
ECONSI	0136	FVLOOP	6155	IOT13	2722	KISZTT	2270
EIFCLR	2234	GETHW3	4466	IOT2	2613	KJILD	2267
ENDSLU	3131	GETSR	4465	IOT3	2620	KJMP3	5227
ENDTST	4530	GOOD	0132	IOT4	2625	KKRTN	4634
EOP1ME	7327	GR1SIM	1664	IOT5	2632	KLB	6037
EPRNTI	0135	GR2GR3	1704	IOT7	2637	KL5IM	4513
ERECI	0121	GRFSIM	1600	IOT8	2644	KMD1	6307
ERICLR	2220	GROUP2	1706	IOT9	2651	KNOP	3350
ERISSET	2230	GROUP3	2000	IOTADD	3044	KRB	6036
ERR1ME	6331	GSR	2253	IOTLP	5010	KRB1	6306
ERRMES	5230	GTF	6004	IOTREC	3056	KRB2	6326
ERROR	4450	H1	5566	IOTTA	3043	KRS	6034
ETIMLL	1472	HANGER	5567	IOTXMT	3066	KRS1	6304
ETIMLU	1473	HCW1	0021	ISSTBB	2700	KRS2	6324
ETIMMS	1471	HCW2	0022	ITOLOC	0166	KRTN	4627
EXICLR	2214	HCW3	0023	JMP1R	0111	KSF	6031
EXIOT	5027	HCW3MS	6760	JMPRET	4073	KSF1	6301
EXIOTL	5061	HLT	7402	JMSLO2	1452	KSF2	6321

KSFLG	0040	LQTAB	4251	MKSF	4475	NXTONN	2132
KSFHAT	4452	LSAV1	7355	MLAP	6361	NXPAT	1504
KSTOP	0066	LSAV2	7356	MLA	6506	NXPTT	2131
KTEST	4660	LSAV3	7357	MLQP	6364	ODT1	0004
KTEST1	4661	LXCRST	6731	MMID	6277	ODT2	0005
L101	4231	M12CON	6076	MMO	6464	ODT3	0006
L102	4266	M1AC	6424	MODAGN	5602	OERR	2130
L111	4461	M1FL	6440	MODE1	0027	OPONKI	2471
L112	4472	M1MG	6432	MODE2	0006	OTZ	1405
L1SAV1	7557	M1PC	6531	MODE3	0007	OUT	1724
L1SAV2	7560	M1TN	6523	MODVAR	6071	OW1APT	6602
L1SAV3	7561	M212	7310	MPC	6532	OWAPT	6602
L211	2023	M270	7304	MQ1SAV	6711	OWSET	0635
L61	3623	M3	7307	MQA	7501	PASSNO	0137
L62	3653	M43	7460	MQDATA	0070	PATCH	0141
LCD	6751	M8CON	6103	MQL	7421	PBRMS	6027
LCDATA	6071	MAC	6456	MQSAVE	6676	PC1SAV	6707
LCDF	3330	MBR	6512	MQWAS	0055	PCIE	6665
LENBAS	2146	MEMES	5210	MREC1	6350	PCLF	6662
LIS1NU	7274	MEO	3007	MREC2	6353	PCLLOOP	2200
LISN	4525	MES1AC	6676	MREC3	6356	PCLP	6666
LISN1	7255	MES1F	7433	MSHDR	6457	PCPUM	0211
LISN11	7253	MES1FL	6704	MSLU	6467	PCPUMS	2200
LISN2	7273	MES1MQ	6701	MSPF	4502	PCSAVE	6674
LISN21	7271	MES1PC	6672	MTCF	4504	PERR	2135
LISN3	7266	MES1TN	6554	MTLS	4510	PFM	5011
LISN31	7264	MESAC	6663	MTN	6524	PFMES	5145
LISNF1	4437	MESACT	6547	MTPC	4506	PIMES	4352
LISNSR	6047	MESAG	7414	MTSB	4505	PMENES	5200
LISNUM	7276	MESAGF	7441	MTSF	4503	PMODE	2076
LKDATA	0071	MESAGX	7423	MTSK	4507	POINTB	0100
LKWAS	0056	MESBR	6563	MXMT1	6337	POINTC	0101
LL1	4406	MESEXP	6556	MXMT2	6342	POINTD	0102
LL2	4433	MESFL	6671	MXMT3	6345	POINTE	0103
LLLD	2257	MESHDR	6514	NDF	0074	POINTE	0104
LLXB	3125	MESINS	6543	NOCRF	7137	POINTR	0076
LOC1	0164	MESLK	6540	NOPI	4531	PPIM	3411
LOOP	4467	MESMQ	6666	NORDIS	6400	PPIMES	4342
LOOPA	0031	MESPC	6660	NORF	7136	PPRMS	7076
LOOPFC	4446	MESRC	6546	NORMDI	6400	PRDB	6667
LOPOON	2242	MESSLU	6557	NORMSQ	2276	PRDD	6150
LQ4TAB	4252	MESTN	6565	NORSLU	6446	PRETES	5000
LQ11	4207	MESXM	6537	NORX01	5053	PRN11	4441
LQ12	4224	MFL	6472	NOSIM0	3426	PRN21	4444
LQLS	6506	MGTF	3453	NOSIM1	4472	PRNT1	4526
LQMC	6503	MINS	6500	NSQBRD	0024	PRNT2	4530
LQMP	6502	MIOT	4463	NSTKS	4533	PROPLC	7102
LQPC	6504	MKCC	4476	NUMX	4070	PROPM3	7100
LQRB	6501	MKCF	4474	NUOK	2211	PRTBR3	6076
LQRE	6507	MKIE	4500	NXTDC	0214	PRTCMS	5442
LQRS	6505	MKRB	4501	NXTIOT	5040	PRTST	3400
LQSK	6500	MKRS	4477	NXTONE	1465	PSKF	6661

PSMES	0277	RXHERE	0133	SLULSU	1640	T21DN	1730
PSR	0020	S1SURL	7547	SLUMES	3303	T2LA	3457
PS81	0142	SAVICH	7100	SLUMST	1600	T2LQ	4001
PS82	0134	SAVAC	6276	SLUREC	0127	T3	5121
PS83	0126	SAVCHR	7103	SLUXMT	0120	T3LA	3514
PS84	0117	SAVFLD	0061	SMES	0213	T3LQ	4112
PS8F	0660	SAVLOC	2420	SMOPE	5655	T4	5200
PSTB	0664	SBDP1	2675	SNUMX	4071	T4LA	3000
PTEB	2162	SBDP2	2671	SOMSKP	0060	T4LQ	4200
PTEST6	0000	SBDP3	2665	SPAC2	7215	T5	5253
QES1MK	7346	SD1	6065	SPAC21	4440	T5B	5200
QES2MK	7315	SD2	6071	SPACE2	4531	T5LA	3641
RLK	7147	SD3	6073	SPACE2	7217	T5LQ	4257
R1SUBL	7547	SD4	6074	SPCIE	4516	T6	5326
RCNT	5441	SDATA	4525	SPE1	2054	T6LA	3670
RDY	6214	SDEVC	0025	SPE2	2060	T6LQ	4305
RE1TYP	7065	SDN	6755	SPE3	2466	T7	5400
RE2TYP	2502	SEL	6750	SPE4	6040	T70K	5444
RE860W	0624	SER	6754	SPE5	6310	T7LA	4000
RE8TBR	3214	SERUEI	6210	SPE6	6330	T7LO	4400
RET	4447	SETBDR	3150	SPE7	7343	T70W1	0421
RETIAD	2017	SETBPI	4472	SPE8	7015	T70W2	0440
RETIADD	4030	SETERI	4461	SREC	4474	TABADD	3055
RETRN	2400	SETEXI	4460	SRESG	7312	TABDON	4241
RETYPE	7064	SETIF	6246	SRO	6003	TADI	3233
REYSEN	1735	SETSIM	1010	SSLUDI	6410	TBLPTR	6075
RFINT	0200	SFLDS	4276	SSLUN	6511	TCF	6042
RIB	6234	SFLGS	3664	SSUBLK	7335	TCF1	6332
RIF	6224	SHFCT	0110	STAB	2746	TCF2	6332
RLK	7150	SHIFT	5662	START	0200	TESAGG	2130
RLOOP	4514	SHIFTO	5633	STKS	0107	TESAGN	1502
RMP	6244	SHINDA	6131	STR	6753	TESLOC	0072
RPERR	3657	SIMCK0	4520	SUNEI	6256	TESMCS	3300
RPTBR	2523	SIMCK1	4531	SVBDC	1470	TEST	0119
RQEST	7021	SIMERR	2055	SWP	7521	TEST1	0221
RSUBLK	7345	SKON	6000	SXDRDA	6074	TEST10	1000
RTC1	5425	SKP1CH	6200	SXMT	4473	TEST11	1045
RTC2	5502	SKPCHK	2047	T1	5014	TEST12	1105
RTCIF	0113	SKPCHN	6200	T10	5426	TEST13	1200
RTCMS	5452	SKPERR	5046	T10LS	4600	TEST14	1310
RTCT1	5410	SKPLRC	4067	T11	5445	TEST15	1327
RTCT10	5423	SKPPED	0057	T11LS	4632	TEST16	1342
RTCT2	5463	SLCOTA	6073	T12	5462	TEST17	1400
RTCT2D	5500	SLOOPA	3233	T12LS	4652	TEST2	0241
RTCT3	5504	SLU2MC	4433	T13	5507	TEST20	1430
RTCT4	5600	SLUCAP	4511	T13LS	4672	TEST21	1701
RTCT5	5627	SLUDAT	4453	T14LS	4712	TEST22	2000
RTCT6	5655	SLUDEF	4454	T14S	5600	TEST3	0400
RTF	6005	SLUDIS	6406	T15LS	4733	TEST4	0425
RTRP	3650	SLUEND	3331	T1LA	3430	TEST5	0473
RTRN	4635	SLULER	2153	T1LQ	4005	TEST6	0606
		SLULSL	1620	T2	5076	TEST7	0725

TESTAB	2265	TWO10C	7536	XC1SW	7013	XSEPI	4501
TESTF1	0124	TWOOCK	7164	XC8APT	7400	XSFBI	4522
TESTMS	2213	TYOUT	7323	XC8CRR	7200	XSPCIE	3544
TESTPPT	2061	TYOUT1	7402	XC8ERR	6600	XSRFF	4210
TESTS	2023	TYPE	4527	XC8HJ	7540	XSTKS	0110
TFLD	3241	TYPE1	4442	XC8RC	7012	XTPLG	0100
TFLG	4537	U1	6502	XC8STA	6716	XTOR	4275
TIMERR	1474	UDATA	6055	XC8SW	7360	XTYPE	7321
TJLOC	2266	UDONE	6053	XCDIL	4240	XVDLY	1667
TLOOP	6700	UEI	6205	XCFBLU	3265	XWAIT	5553
TLOOP1	6675	UEIMES	6311	XCHKMS	7100	XZAP	6125
TLS	6046	UL1	6240	XCHKRS	7111	ZTO	1420
TLS1	6310	UL2	6244	XCLEPT	4475		
TLS2	6330	UL3	6250	XDATA	4524		
TNF7	3271	UL4	6254	XDELAY	3245		
TPC	6044	UL5	6260	XDR	0752		
TPC1	6314	UL6	6264	XDRDAT	6072		
TPC2	6334	UL7	6270	XGETSR	3312		
TRMXP	2133	UNACON	6057	XGHW3	6346		
TS60W1	0652	UNBCON	6064	XIOT	3000		
TS60W2	0663	UNEI	6222	XKLSIM	4511		
TS60W3	0664	UNEINT	6221	XLCTR	2144		
TS60W4	0674	UNETMS	6303	XLISN	7225		
TS60W5	0701	UNIT	6052	XLISN1	7223		
TS60W6	0707	UNVAR	6052	XMIOT	3024		
TSB	6043	UOUT	6056	XMKCC	2612		
TSB1	6313	UPARR1	7350	XMKCF	2600		
TSB2	6333	UPARRG	7317	XMKIE	2624		
TSBSSK	0243	UPSTK	3255	XMKRB	2631		
TSCACL	0230	USEDIO	5064	XMKRS	2617		
TSF	6041	UX	6507	XMKSF	2605		
TSF1	6311	UXFR	6054	XMODE2	1660		
TSF2	6331	VAR	6164	XMSPF	2636		
TSFNAT	4451	VCDI	6274	XMTCF	2650		
TSK	6045	VDCTR	1700	XMTLS	2721		
TSK1	6315	VDELAY	4473	XMTPC	2707		
TSK2	6335	VMODE	2016	XMTSB	2655		
TSL0P	2220	WAIT	4464	XMTSF	2643		
TSLCLL	0265	WAIT10	7506	XMTSK	2714		
TSLCML	0273	WAITQC	7521	XNC	2047		
TSTDAB	6111	WATKSF	2346	XORS	7214		
TSTEND	2227	WATTSF	2333	XORS1	7214		
TSTIAC	0253	WTCLSK	5427	XPRNT1	7477		
TSTIN	2217	XI	4042	XPRNT2	7153		
TSTIOT	5030	XIPRN1	7466	XPRNT4	7464		
TSTLCH	0260	XIPRN2	7525	XR	6205		
TSTL0P	0125	XIPRN4	7453	XRET	0112		
TSTNO	0075	XITYPE	7400	XRETAD	4072		
TSTNU	0140	XBAUD	3112	XRTF1	4066		
TSTPC	5657	XCICRL	7200	XSAV	3510		
TTYICN	7101	XC10OP	7306	XSCAF	3252		
TTYCNT	7104	XC1ERR	6600	XSDP	3461		

SLUDIS	7868#	8044																	
SLUEND	5114	5912#	8042																
SLULER	5002	5007#	5095																
SLULSL	4815	4868#																	
SLULSU	4818	4886#																	
SLUMES	4181	5784#																	
SLUMST	4812	4850#																	
SLURCC	4054#	4990	5000	5021	5022	5050	5051	5064	5072	5088	5325	5329	7082						
SLUXMT	4049#	4660	4670	4696	4700	4725	4728	4732	4737	4740	4744	4951	4984	5309					
	5327	787#																	
SMES	4104#																		
SMORE	7452	7455	7462#																
SNUMX	1980	1989	1996	2023#															
SOMSKP	141#	1142	1253	1257	1261	1263													
SPAC2	3939	8302#	8305																
SPAC21	3937#	8493																	
SPAC22	196#	3653																	
SPACX2	197	3450#	3453																
SPC1E	4034#	5905	5917	5941	5998	6041	6422												
SPE1	1346#	3155																	
SPE2	1354#	3160																	
SPE3	5343#	8038																	
SPF	37#	2232	3759#	5436															
SPF1	3709#																		
SPF2	3817#																		
SR1MES	8156	8201	8399#																
SR1QES	8155#	8178																	
SREC	6435	6436	6456	6457	6484#	6503													
SRMESG	3517#	3566																	
SRQ	29#	1065																	
SRLUDI	7873#	8040																	
SRLUN	7940	7941	7947#	7953	7954	7955													
SSUBLK	3369	3539#	3546																
STAB	5503	5504	5518#																
START	230#	3304	3692	8394															
STKS	167#	1643	1646	1679	1682	1732	1743	1755	1791	1801	1826	1870	1971	1998					
	2074	2104	2129	2205	2225														
STR	3865#	6744	6883	6968	6981	6999	7134	7137	7199	7368	7489	7497							
SUNFI	7802#	8035																	
SVBDC	4784#	4810	4811	4814	4817														
SWP	22#	672	680	687	697	713	724	732	744	1147	1160	1221	1329						
SXDRDA	7642#	7695	7698																
SXMT	6426	6427	6429	6439	6442	6447	6448	6450	6460	6463	6483#								
T1	6732#																		
T10	7227#																		
T10LS	6540	6562#																	
T11	7261#																		
T11LS	6592#																		
T12	7291#																		
T12LS	6611#																		
T13	7311	7324#																	
T13LS	6630#																		
T14LS	6643#																		

SEQ 0211

T14S	737#	7405#																	
T15LS	6669#																		
T1LA	5878#	6545																	
T1LO	5861	5865	6143#	6531															
T2	6847#																		
T21DN	4940	4945	4958#																
T2LA	5903#																		
T2LO	6181#																		
T3	6854	6900#																	
T3LA	5938#																		
T3LO	6236#																		
T4	6931	6963#																	
T4LA	5966	5996#																	
T4LO	6250	6271#																	
T5	7009	7045#																	
T5B	7050#	7097																	
T5LA	6039#																		
T5LO	6315	6331#																	
T6	7111#																		
T6LA	6074#																		
T6LO	6362#																		
T7	7147	7169#																	
T7OK	7230	7241#																	
T7LA	6111	6121#																	
T7LO	6125	6390	6420#																
T7OW1	6122	6387	6437#	6537															
T7OW2	6124	6389	6458#																
TABADD	5502	5514	5515	5535	5541	5542	5548	5582#	5610										
TABDON	6303	6300#																	
TADI	1645	1661#																	
TBLPTR	2765	2815#	2820	2826															
TCF	2310	3531	3761#	5448	5767	8432													
TCF1	3791#	5769																	
TCF2	3819#	5771																	
TESAGG	1371	1376	1403#																
TESAGN	1136#	1403																	
TESLOC	145	154#	755	760	766	767	773	778	785	786	790	875	879	960					
	965																		
TESMES	4183	5785#	6399	6941															
TEST	173#	1242	1270	1295	1299	1306													
TEST1	4121#	5194																	
TEST10	4451#	5201																	
TEST11	4496#	5202																	
TEST12	4533#	5203																	
TEST13	4572#	5204	5350																
TEST14	4655#	5205	5220																
TEST15	4671#	5206	5221																
TEST16	4690#	5207	5222																
TEST17	4719#	5208																	
TEST2	4145#	5195																	
TEST20	4757#	5223																	
TEST21	4933#	5209																	
TEST22	4973#	5210																	

SEQ 0212

L1570	4816	4840#			
L1571	4815	4841#			
L1572	4812	4842#			
L1573	4809	4843#			
L1574	1162	1163#	4758	4844#	
L1575	1136	1164#	4736	4845#	
L1576	1134	1165#	4729	4741	4846#
L1577	1129	1166#	4720	4847#	
L1771	4953	4961#			
L1772	4952	4962#			
L1773	4950	4963#			
L1774	4943	4964#			
L1775	4941	4965#			
L1776	4938	4966#			
L1777	1247	1285#	4934	4967#	
L2165	5057	5099#			
L2166	5053	5099#			
L2167	5030	5100#			
L2170	5024	5101#			
L2171	5023	5052	5102#		
L2172	5011	5103#			
L2173	5005	5104#			
L2174	4994	5033	5105#		
L2175	4980	5106#			
L2176	1496	1499#	4978	5107#	
L2177	1374	1410#	4974	5108#	
L2372	5238	5270#			
L2373	5235	5240	5271#		
L2374	5186	5272#			
L2375	5185	5273#			
L2376	5184	5274#			
L2377	5114	5275#			
L2571	5377	5307#			
L2572	5376	5388#			
L2573	5368	5389#			
L2574	5364	5390#			
L2575	5353	5391#			
L2576	5350	5392#			
L2577	5303	5393#			
L2770	5510	5521#			
L2771	5506	5522#			
L2772	5505	5523#			
L2773	5502	5514	5515	5524#	
L2774	5462	5466	5470	5473	5525#
L2775	5461	5465	5469	5472	5526#
L2776	5458	5527#			
L2777	5455	5528#			
L3163	5655	5672#			
L3164	5652	5673#			
L3165	5647	5664	5674#		
L3166	5642	5644	5660	5675#	
L3167	1585	1608	1616#	5623	5676#
L3170	1571	1617#	5614	5653	5677#

L3171	1559	1618#	5613	5678#	
L3172	1509	1619#	5612	5635	5679#
L3173	1502	1620#	5611	5634	5680#
L3174	1495	1516	1621#	5609	5681#
L3175	1482	1622#	5564	5682#	
L3176	1480	1540	1623#	5538	5545 5683#
L3177	1478	1624#	5534	5684#	
L3371	1772	1779	1784#		
L3372	1751	1785#			
L3373	1737	1739	1741	1786#	
L3374	1719	1787#			
L3375	1715	1788#			
L3376	1650	1686	1789#	5792	5829#
L3377	1642	1652	1668	1688	5773 5830#
L3565	5973	5977#			
L3566	5966	5978#			
L3567	5803	5979#			
L3570	5861	5865	5980#		
L3571	5859	5981#			
L3572	1873	1877#	5851	5982#	
L3573	1846	1878#	5845	5983#	
L3574	1843	1879#	5844	5984#	
L3575	1841	1880#	5843	5985#	
L3576	1797	1881#	5839	5986#	
L3577	1792	1882#	5837	5987#	
L3771	1932	1944#			
L3772	1925	1938	1945#		
L3773	1916	1934	1946#		
L3774	1909	1947#			
L3775	1902	1905	1948#	6111	6113#
L3776	1893	1949#	6078	6083	6089
L3777	1891	1950#	6076	6081	6086
L4164	6254	6252#			
L4165	6246	6253#			
L4166	6215	6239	6254#		
L4167	2021	2027#	6208	6255#	
L4170	2017	2028#	6204	6256#	
L4171	2015	2029#	6198	6257#	
L4172	2013	2030#	6195	6258#	
L4173	2010	2031#	6188	6203	6214 6259#
L4174	2009	2032#	6125	6260#	
L4175	1987	2011	2033#	6124	6261#
L4176	1984	2012	2034#	6122	6262#
L4177	1967	2035#	6121	6123	6263#
L4367	6398	6406#			
L4370	6389	6407#			
L4371	2107	2110#	6387	6408#	
L4372	2080	2111#	6386	6388	6409#
L4373	2079	2112#	6376	6410#	
L4374	2072	2102	2113#	6369	6381 6411#
L4375	2060	2089	2114#	6364	6366 6379 6412#
L4376	2051	2115#	6334	6413#	
L4377	2049	2116#	6273	6414#	

