ICR11

CONTROLLER DIAGNOSTIC

MD-11-DZIRA-A

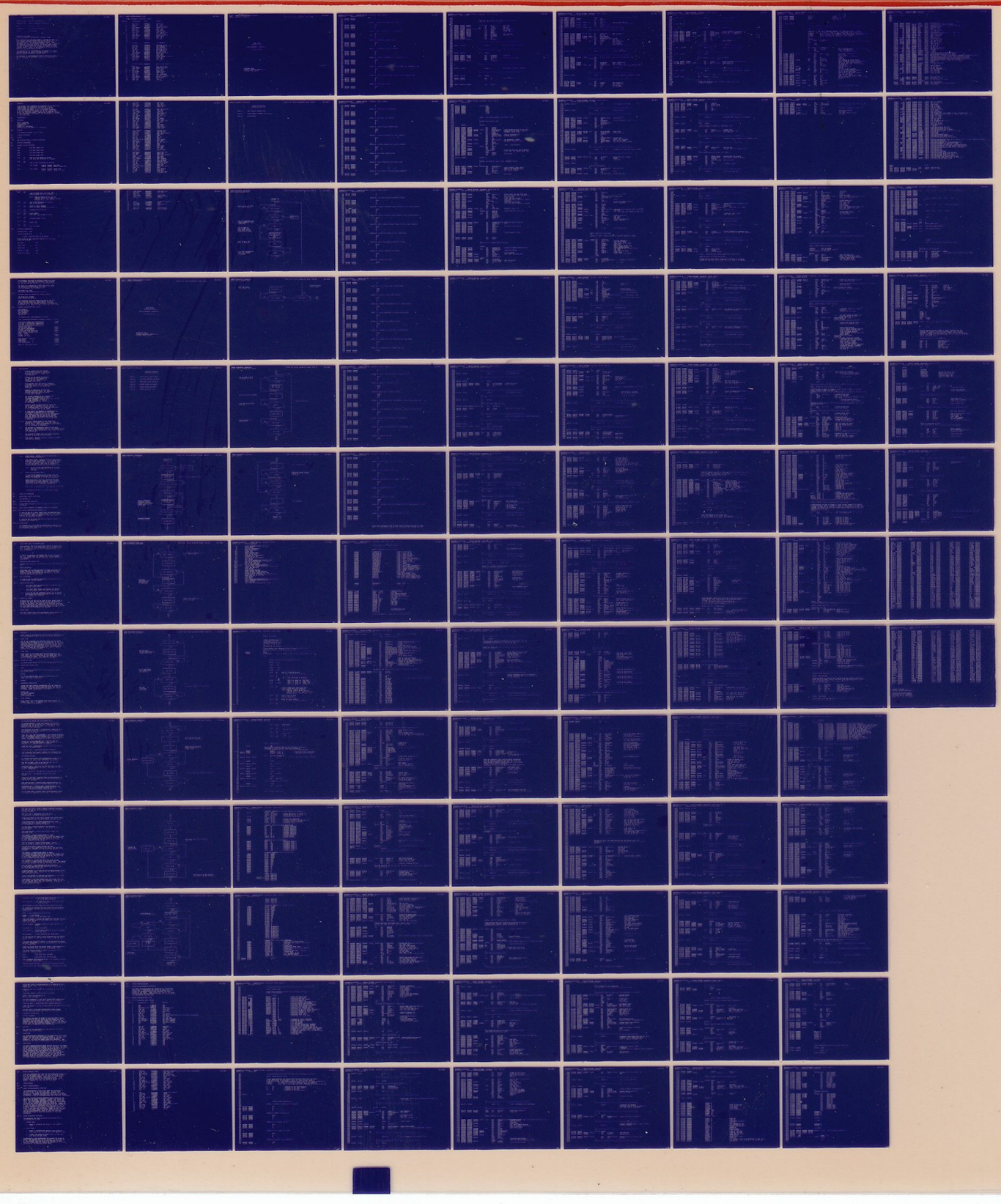
EP-DZIRA-A-DL

FICHE 1 OF 1

JUN 1978

digital

MADE IN USA



PRODUCT CODE: MAINDEC-11-DZIRA-A-D

PRODUCT NAME: ICR11 CONTROLLER DIAGNOSTIC

DATE: FEBRUARY 1976

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: DAN DEKNIS

1. ABSTRACT

THIS PROGRAM WILL EXERCISE AND DIAGNOSE ONE OR MORE ICR11 CONTROLLERS. THE CONTROLLER CONSISTS OF A M8094, M8098, M8098 AND M8096. IT WILL NOT EXERCISE ANY LOGIC BEYOND THE M8096. (I.E. I/O BOARDS) THEREFORE IT IS NOT NECESSARY TO REMOVE ANY OF THE MODULES FROM WITHIN THE FILE BOX EXCEPT UNSOLICITED INTERRUPT MODULES OR THEIR SOURCES. THIS PROGRAM ALSO EXERCISES THE TERMINAL AT THE ICR11 END.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 COMPUTER
ICR11 CONTROLLER
CONSOLE TTY
REMOTE TTY (OPTIONAL)
SHORTING PLUG FOR REMOTE TTY (OPTIONAL)

2.2 STORAGE

THIS PROGRAM RESIDES IN 8K OF CORE

3. LOADING PROCEDURE

3.1 METHOD

USE STANDARD PDP-11 LOADING PROCEDURE

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SW00		FILE BOX UNDER TEST			
SW01		FILE BOX UNDER TEST			
SW02		FILE BOX UNDER TEST			
SW03		FILE BOX UNDER TEST			
SW04	-0-	TEST ALL FILE BOXES ON SYSTEM			
	-1-	TEST FILE BOX SPECIFIED BY SW00-03			
SW05	-0-				
	-1-	LOOP ON TEST SPECIFIED BY SW07,06			
SW06		TEST NUMBER	SW07=0	SW06=0	CSR TEST
			SW07=0	SW06=1	M8094, M8098 TEST
SW07		TEST NUMBER	SW07=1	SW06=0	M8096 TEST
			SW07=1	SW06=1	REMOTE TTY TEST

```
        SW08    -0-
                -1-     LOOP ON M8098 DATA LOOP BACK TEST
                        (USES SW00-07 FOR DATA PATTERN)

                        NOTE:  SW06,07 SHOULD BE 0,1 OR 1,0
                               SW06,07 AS 0,0 WILL DEFAULT TO 0,1
                               SW06,07 AS 1,1 WILL DEFAULT TO 1,0

        SW09    -0-     HALT AFTER PRINTING RUN SUMMARY
                -1-     LOOP ON DIAGNOSTIC

        SW10    -0-     PRINT ON LOCAL CONSOLE
                -1-     PRINT ON LOCAL CONSOLE AND REMOTE TTY

        SW11    -0-     RESERVED FOR FUTURE USE
                -1-

        SW12    -0-     QUICK VERIFY
                -1-     LONG TEST (100 PASSES OF TEST)

        SW13    -0-
                -1-     SUPPRESS ERROR TYPEOUT

        SW14    -0-
                -1-     SCOPE LOOP

        SW15    -0-
                -1-     HALT ON ERROR

4,2     STARTING ADDRESS(ES)

        PROGRAM START   200

        PROGRAM RESTART 210

                NOTE:  SAME FILE BOX TO BE TESTED

        AFTER START OF 200, SEPARATE RESTARTS OF INDIVIDUAL
        TESTS MAY BE DONE.

        CSR TEST                210

        M8094 TEST              204

        M8096 TEST              214

        REMOTE TTY TEST         220

        POWER FAIL OPTION       224
```

C1

THIS PROGRAM INCLUDES AN OPTIONAL POWER FAIL TEST
IF THE POWER FAIL TEST IS REQUIRED, LOAD ADDRESS
224 AND SET SWR PER SECTION 4.1, AND PRESS START.

THE TEST WILL EXECUTE AS A START 200 UNTIL AFTER
THE REMOTE TTY TEST(SECTION 4.3 *18*), AT WHICH
TIME THE PROGRAM WILL RESPOND WITH:

ICR POWER FAIL TEST
POWER DOWN THE REMOTE END OF THE ICR

POWERING DOWN THE REMOTE END SHOULD PRODUCE:

ICR POWER FAIL SENSED
POWER REMOTE END BACK UP

THIS VERIFIES THAT THE POWER FAIL BIT OF THE ICR
CSR FUNCTIONS CORRECTLY, POWER REMOTE END BACK
UP AND THE TEST WILL PROCEED NORMALLY. THIS TEST IS
PERFORMED ON ALL ICR'S AND ONLY WITHIN THE FIRST PASS.

4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM
SET SWR TO 200
LOAD ADDRESS
SET SWR PER 4.1
PRESS START

THE PROGRAM WILL THEN RESPOND BY TYPING:

```
ICR11 CONTROLLER DIGNOSTIC MAINDEC-11-DZIRA-A    (*1*)

********** SYSTEM MAP **********                 (*2*)
FILE BOX X ICAR=XXXXXX ICSR=XXXXXX               (*3*)
FILE BOX Y ICAR=YYYYYY ICSR=YYYYYY
************************************
TESTING FILE BOX X                               (*4*)
ICR VECTOR ADDRESS=NNN                           (*5*)
PRIORITY LEVEL=ZZZZZZ                            (*6*)
MAINTENANCE MODE ACTIVATED                       (*7*)
INPUT MODULE ADDRESS RANGE                       (*8*)
START   XXXXX
FINISH  YYYYYY                                   (*9*)
REMOTE  TTY TEST                                 (*10*)

*RUN SUMMARY - FILE X*                           (*11*)
PASS COUNT          X                            (*12*)
ERROR COUNT         Y                            (*13*)
LINE ERROR COUNT    Z                            (*14*)

RERUN OR LOAD FIELD TEST?                        (*15*)
```

1 HEADING IDENTIFYING THE PROGRAM
A OF MAINDEC-11-DZIRA-A INDICATES
REVISION, BE SURE IT AGREES WITH
THIS DOCUMENT.

2 PROGRAM IS NOW GOING TO INDICATE
ALL ICR'S THAT ARE PRESENT ON
SYSTEM AND WILL RESPOND WHEN
ADDRESSED WITH SSYNC.

3 FILE BOX(ES) THAT ARE ACTUALLY PRESENT
ON SYSTEM. CHECK THAT ALL ICR(S) REPORTED
EXIST AND THAT ALL THAT EXIST ARE
REPORTED.

4 PROGRAM IS BEGINNING TO TEST ONE
SPECIFIC FILE BOX (ICR). ALL MESSAGES
PRINTED FROM THIS POINT ON REFER
TO THIS ONE FILE BOX.

5 ICR VECTOR ADDRESS IS THE ADDRESS THAT
THE FILE BOX INTERRUPTED TO WHEN A
INTERRUPT OCCURRED. COMPARE THIS VALUE WITH
THAT ONE EXPECTED, IF THEY DO
NOT ARGEE INVESTIGATE JUMPER VX-VY
ON THE M8094.

6 PRIORITY LEVEL IS THAT LEVEL AT WHICH THE
ICR IS SET TO INTERRUPT AT. SIX (6) IS
THE STANDARD LEVEL. THIS LEVEL IS DETERMINED
BY THE PRIORITY CHIP ON THE M8094.

7 AT THIS POINT THE PROGRAM IS EXERCISING THE
MAINTENANCE MICROCODE WITHIN THE M8094
AND M8096. AT THIS TIME IT IS NOT RECOMMENDED
THAT THE REMOTE END OF THE ICR BE POWERED
DOWN NOR SHOULD THE PROGRAM BE HALTED. FOR
EXPLANATION OF PRINTOUTS AT THIS TIME REFER
TO SECTION 7.

8,*9* THE INPUT MODULE RANGE IS THE RANGE THAT
JUMPERS OF THE M8096 WILL ALLOW THE M8096 TO
SCAN OR POLL. THIS IS DETERMINED BY THE PROGRAM
AND THE STANDARD IS 0000-0016.

10 THE PROGRAM IS PERFORMING TESTS TO THE ICR11
REMOTE TTY. THE SHORTING PLUG MUST BE PLUGGED IN FOR
THIS TEST TO RUN, OTHERWISE IT WILL ABORT THE TEST
AND INDICATE SO.

11 RUN SUMMARY OF TESTS THAT HAVE BEEN PERFORMED.
(SUMMARY IS PRINTED FOR EACH FILE)

12 PASS COUNT. DECIMAL VALUE OF NUMBER OF PASSES
COMPLETED BY PROGRAM

13 ERROR COUNT. DECIMAL VALUE OF ERRORS SEEN BY PROGRAM DURING RUNNING.

14 LINE ERROR COUNT. DECIMAL VALUE OF LINE ERRORS DETECTED DURING DIAGNOSTIC. (LINE ERRORS INDICATE ONLY THE NOISE LEVEL OF THE CABLE BETWEEN THE ICR AND PDP-11, THEY HAVE NO BEARING ON ERRORS DETECTED UNLESS THEY ARE EXTREMELY HIGH.)

NOTE: IF ANY VALUES ARE PRECEEDED BY A MINUS SIGN (-), AN OVERFLOW OF 2**14 (16,384) WAS SEEN.

15 RERUN OR LOAD FIELD TEST?

AT THIS POINT RESPONDING ON EITHER LOCAL OR REMOTE (IF SELECTED) TERMINALS WITH A "R" WILL RESTART THE PROGRAM AT *6* OF THIS SECTION.

RESPONDING WITH A "L" WILL INITIATE THE HIGH SPEED READER TO START LOADING. THIS LOADING CAPABILITY IS TO FACILITATE THE LOADING OF THE ICR-11 FIELD TEST (MAINDEC-11-DZIRB) ONLY.

D O N O T A T T E M P T T O L O A D O T H E R
P R O G R A M S B Y T H I S M E T H O D!

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

SEE SECTION 4.1

5.2 OPERATING FROM LOCAL END

5.2.1 TO RUN THIS DIAGNOSTIC FROM THE LOCAL END FOLLOW THIS PROCEDURE.

5.2.1.1 LOAD SWITCH REGISTER TO 000200, PRESS LOAD ADDRESS.

5.2.1.2 SET SWITCH REGISTER PER SECTION 4.1

IF SW10 IS SET TO A ZERO, CHECK THAT THE SHORTING PLUG IS IN THE REMOTE END. IF SW10 IS ZERO, AND THE SHORTING PLUG IS NOT IN, THEN THE REMOTE TTY TEST WILL NOT RUN AND WILL BE INDICATED AS SUCH.

IF SW10 IS SET TO A ONE, BE SURE THE REMOTE TERMINAL IS CONNECTED AND POWERED ON.

5.2.1.3 PRESS START

THE PROGRAM WILL NOW START TO RESPOND AS IN SECTION 4.3. ANY DEVIATION FROM THIS TYPEOUT IS EXPLAINED IN SECTION UNDER ERROR EXPLANATIONS.

5.3 GOING FROM LOCAL TO REMOTE USAGE

WHEN QUESTION "RERUN OR LOAD FIELD TEST?" IS ASKED, SET SW10 TO A ONE (1), THEN POWER DOWN ICR AND ATTACH REMOTE TTY. POWERING UP ICR WILL REASK QUESTION REMOTELY (AS WELL AS LOCALLY).

OR

IF ERROR IS OCCURING, SET SCOPE LOOP SWITCH AND SW10 TO ONE (1). POWER DOWN ICR (REMOTE END), ATTACH REMOTE TTY. POWERING UP ICR (REMOTE END) WILL PRINT "RESTARTING DIAGNOSTIC".

5.4 OPERATING FROM REMOTE END

PROGRAM MUST INITIALLY BE STARTED AT LOCAL END PER SECTION 5.2.

5.4.1 ERROR REPORTING

SINCE ONCE THE MAINTENANCE MODE HAS BEEN ACTIVATED THE REMOTE TERMINAL IS INOPERATIVE, ALL ERRORS THAT ARE DETECTED DURING THIS PERIOD ARE BUFFERED AND REPORTED PRIOR TO THE END OF EACH PASS.

5.4.2 RUN INDICATION

IF SW09 IS SET TO LOOP ON DIAGNOSTIC, TESTING WILL FLASH (ON RT02 ONLY) DURING PASSES.

5.4.3 ABORTING FROM TEST

^C WILL EXIT TEST AND RETURN YOU TO QUESTION "RERUN OR LOAD FIELD TEST?"

^P WILL EXIT TEST, PRINT RUN SUMMARY AND RETURN YOU TO QUESTION "RERUN OR LOAD FIELD TEST?"

NOTE: ^C AND ^P ARE NOT INTERRUPT DRIVEN DUE TO MAINTENANCE MICRO CODE BEING EXECUTED SO HAVE PATIENCE, THE PROGRAM WILL RESPOND.

5.4.4 POWER FAIL ABORT

POWERING DOWN THE ICR MAY BE USED TO EXIT SCOPE LOOPS OR TO ABORT THE TEST ONCE THE POWER FAILURE HAS BEEN DETECTED BY THE PDP11, THE LOCAL END IS NOTIFIED, AND WILL BE KEPT INFORMED OF THE SITUATION EVERY MINUTE OR SO UNTIL THE LINK GOES ERROR FREE INDICATING POWER HAS BEEN RESTORED. AT THIS TIME THE DIAGNOSTIC WILL RESTART.

6. SCOPE LOOPS

SW14 AS A ONE(1) WILL LOCK THE PROGRAM INTO A SCOPE LOOP. THE SCOPE LOOP OPERATION VARIES WITH EACH TEST.

6.1.1 CSR TEST

SCOPE LOOPS IN THIS SECTION LOOP ON THE ONE ERROR INVOLVED AND ANY NECESSARY CODE TO SET UP THE ERROR CONDITION.

6.1.2 M8094 TEST

DUE TO THE MICROCODE AND HARDWARE STRUCTURE THE SCOPE LOOP IS FROM THE ERROR BACK TO THE BEGINNING OF THE TEST. THEREFORE FOR AN ERROR IN THE LATTER PART OF THE TEST ALL PREVIOUS SUBTESTS ARE PERFORMED WITHIN THE SCOPE LOOP. THE EXCEPTION IS THE M8098 TEST WHICH IS CONTROLLED BY SW08. WITH SW08 SET THE PROGRAM WILL LOOP ON SENDING SW07-00 UNTIL SW08 IS CLEARED.

6.1.3 M8096 TEST

SCOPE LOOPS IN THE M8096 TEST INVOLVE LOOPING ON THE WHOLE TEST, WITH THE EXCEPTION OF THE M8098 TEST WHICH IS CONTROLLED BY SW08. WITH SW08 SET THE PROGRAM WILL LOOP ON SENDING SW07-00 UNTIL SW08 IS CLEARED.

6.1.4 REMOTE TTY TEST

ALL SCOPE LOOPS RETURN TO THE LAST TRANSMISSION ATTEMPT.

6.1.5 ESCAPE FROM SCOPE LOOPS

LOCAL

^C, ^P OR SW14 TO ZERO WITH ALLOW THE USER TO ESCAPE FROM SCOPE LOOPS.

REMOTE

^C, ^P OR POWERING DOWN ICR WILL ALLOW ESCAPE FROM SCOPE LOOP BACK TO BEGINNING.

7.0 ERRORS

7.1 ERRONEOUS INFORMATION

DURING THE TEST CERTAIN INFORMATION WILL BE TYPED OUT INDICATING CERTAIN JUMPER CONFIGURATIONS, FILE BOX ADDRESSES. THESE MAY BE IN ERROR AND SHOULD BE CHECKED TO VERIFY THEIR CORRECTNESS. THEY ARE:

SYSTEM MAP
ICR VECTOR ADDRESS
PRIORITY LEVEL
INPUT MODULE RANGE

7.2 FATAL ERRORS

FATAL ERRORS ARE THOSE ERRORS WHICH UPON FINDING THE PROGRAM WILL HALT. TO RECOVER FROM THESE ERRORS A START OF THE PROGRAM IS NECESSARY.

THE FOLLOWING ERRORS ARE CONSIDERED FATAL.

"NO ICR'S-ABORTED" - WHEN THE PROGRAM WAS TRYING TO DETERMINE THE ICR CONFIGURATION, THERE WAS NO SSYNC RECEIVED FROM ANY OF THE LEGAL ICR ADDRESSES. SEE SECTION 9.3 FOR ADDRESSES.

"NON-EXISTANT FILE BOX" - IN SELECTING A SPECIFIC FILE BOX TO TEST VIA SW4 AND SW03-00, THE USER SELECTED A NON-EXISTANT FILE BOX.

"MOD. INT. WILL NOT INT-ABORTED" - WHEN TRYING TO DETERMINE THE INTERRUPT VECTOR ADDRESS AND INTERRUPT PRIORITY LEVEL, THE FORCED MODULE INTERRUPT WOULD NOT TAKE PLACE. (NOTE: DEPRESSING CONTINUE WILL FORCE A SCOPE LOOP).

"REMOTE TTY HUNG--DESELECT IT" - TEST OF TBMT OF THE REMOTE TTY WAS UNSUCCESSFUL, CHECK REMOTE TTY OR DESELECT IT BY SETTING SW10 TO A ZERO(0).

"SPUR ICR INT" - EXTRANEOUS INTERRUPT FROM ICR OCCURED DURING TTY INTERRUPT TEST.

"UNEXP MOD INT" - UNEXPECTED MODULE INTERRUPT.

"TTY INTERRUPT TEST HUNG" - REMOTE TTY INTERRUPT TEST TIMED OUT WHILE WAITING FOR TBMT OR DA TO INTERRUPT.

7.3 RECOVERABLE ERRORS

ALL ERRORS ARE PRINTED WITH CORRESPONDING ERROR PC OF THE LISTING SO ACTUAL TEST PERFORMED MAY BE SEEN.

"CSR BIT 15 SET" - BIT 15 OF THE ICR CSR WAS SET WHEN NO TRANSMISSION WAS OCCURRING.

"RESET BIT SET" - BIT 6 OF THE ICR CSR WHICH IS USED TO RESET MODULES WITHIN FILE BOX. READ BACK AS A ONE(1) WHICH IT SHOULD NOT.

"TTY ENA NOT SET" - ICR CSR BIT 5 WOULD NOT SET (ONE).

"TTY ENA SET" - ICR CSR BIT 5 WOULD NOT CLEAR (ZERO).

"TMOUT FLP NOT SET" - DURING M8894 MICRODIAGNOSTIC THE TIMEOUT FLOP ON THE M8894 DID NOT SET WITH SUFFICIENT TIME FOR IT TO OCCUR.

"SEQ COMPARE ERR" - DURING M8894 MICRODIAGNOSTIC THE SEQUENCE COMPARE FLOP ON THE M8894 WOULD NOT TEST TRUE.

"ERRONEOUS INT" - DURING M8894 MICRODIAGNOSTIC, INTERRUPT OTHER THAN INTERRUPT EXPECTED OCCURRED. CHECK LISTING AT ERRPC FOR FURTHER EXPLANATION.

"NO INT FROM ERR" - DURING M8894 MICRODIAGNOSTIC ERROR WAS FORCED BUT ERROR BIT DID NOT SET IN CSR (BIT 12).

"INT SET AFT RIF" - AFTER A MODULE INTERRUPT OCCURRED, THE RIF BIT WAS SET AND CLEARED BUT THE MODULE INTERRUPT WAS STILL POSTED.

"ERR AFT CLR" - ADDRESSING THE ICAR WOULD NOT CLEAR OUT THE ERROR INTERRUPT BIT.

"DATA VALID TRUE" - DATA VALID TESTED TRUE WITHIN M8894 MICRODIAGNOSTIC WHEN THE DATA SHOULD HAVE BEEN FALSE.

"INPUT ADDR ERR" - IN M8896 MICRODIAGNOSTIC WHEN WRAPPING THE INPUT MODULE ADDRESS BACK, THEY CAME BACK OUT OF SEQUENCE. CHECK EXP'D REC'D.

"NO MOD INT" - MODULE INTERRUPT WAS EXPECTED BUT NOT POSTED, CHECK ERRPC IN LISTING FOR FURTHER EXPLANATION.

"OUT BUSY STUCK" - OUTPUT BUSY POSTED LONGER THEN SPECIFIED TIME.

"FME ERROR" - FORCED MASTER ERROR IN M8894 MICRODIAGNOSTIC WAS INITIATED BUT IT DID NOT OCCUR, THAT IS A MODULE INTERRUPT WAS TO BE POSTED BY THE MICROCODE AT THIS TIME INDICATING THE FME EVENTS TOOK PLACE. THE MODULE INTERRUPT DID NOT OCCUR.

"FME NO TIMEOUT" - FORCE MASTER ERROR. SHOULD HAVE INITIATED A TIMEOUT ON THE M8894 BUT IT DID NOT.

"SL ACK ON BD CRC" - WHEN THE BAD CRC WAS GENERATED BY THE MASTER THE SLAVE SHOULD NOT HAVE ACK-NOWLEDGED THE MESSAGE, BUT THE SLAVE DID RESPOND TO THE BAD CRC.

"FSE ERROR" - FORCED SLAVE ERROR IN M8894. MICRODIAGNOSTIC WAS INITIATED BUT IT DID NOT OCCUR, THAT IS A MODULE INTERRUPT WAS TO BE POSTED BY THE MICROCODE AT THIS TIME INDICATING THE FSE EVENTS TOOK PLACE. THE MODULE INTERRUPT DID NOT OCCUR.

"FSE TIMEOUT" - THE BAD CRC FROM THE SLAVE CAUSED THE MASTER TO TIMEOUT WHICH IT SHOULDN'T HAVE. THE MASTER SHOULD HAVE IGNORED THE BAD CRC AND REASK FOR THE MESSAGE.

"FSE NO ERROR" - THE SUPPOSED BAD CRC GENERATED BY THE SLAVE WAS GOOD AT THE MASTER END OR THE SLAVE WILL NOT GENERATE A BAD CRC.

"TIMEOUT ERROR" - A TIMEOUT OF THE MASTER OCCURRED IN THE M8898 LOOP BACK TEST. INDICATING THE M8898 (SLAVE) DID NOT LOOP BACK THE DATA.

"DATA VALID ERROR" - A BAD CRC WAS RETURNED BY THE M8898 (SLAVE) DURING THE M8898 LOOP BACK TEST.

"ICAR ERROR" - THE DATA LOOPED BACK AND LOADED INTO THE ICAR DURING THE M8898 TEST WAS INCORRECT. BITS 15 AND 12 ARE NOT CHECKED DUE TO THE FACT THEY ARE NOT DIRECTLY LOADED DURING THIS TEST.

DATA SENT = BIT 7-0 THAT WERE SENT OUT IN THE LAST BYTE
OF THE OUTGOING MESSAGE.

EXP'D COUNTS = 16 BIT WORD THAT SHOULD HAVE BEEN LOADED
INTO THE ICAR.

REC'D COUNTS = 16 BIT WORD THAT WAS ACTUALLY IN ICAR.

THE CORRELATION BETWEEN THE DATA SEND AND ICAR EXPECTED
IS AS FOLLOWS. FOR THE 8 BIT BYTE OUT (BITS 7-0) THE
EXPECTED IS:

BITS Y,1,0,Y,4,5,6,7,X,X,X,X,3,2,1,0

WHERE Y = NOT TESTED
WHERE X,X,X,X = FILE BOX UNDER TEST

"MOD ADDR ERROR" = DURING THE M8098 TEST THE DATA IS ALSO
LOADED INTO THE 16 X 16 FILE IMAGE ON THE M8094, THIS
DATA WAS IN ERROR.

MODULE ADDRESS = ADDRESS WITHIN RAM WHERE THE DATA WAS
READ.

DATA SENT = 8 BIT BYTE SENT.

EXP'D CONTS = 16 BIT EXP'D CONTENTS OF MODULE ADDRESS
BITS 7-0, 7-0.

REC'D CONTS = 16 BIT CONTENTS THAT WERE READ.

"DA NOT SET ON TTY TEST" = DATA AVAILABLE WAS NOT POSTED
IN THE REMOTE TTY LOOP BACK TEST AFTER A BYTE WAS SENT
OUT.

"TBMT WAS SET AFTER TTY TRANS" = UART TRANSMITTER BUFFER
WAS EMPTY IMMEDIATELY AFTER IT WAS LOADED WITH A CHARACTER
TO BE TRANSMITTED.

"TBMT WAS CLEAR AFTER TTY TRANS TIME" = UART TRANSMITTER
BUFFER WAS STILL FULL AFTER TTY TRANSMISSION TIME.

"TTY SEND RECEIVE ERROR" = ERROR WAS DETECTED IN THE
REMOTE TTY LOOP DATA TEST.

SENT = 8 BIT BYTE THAT WAS SENT

REC'D = 8 BIT BYTE THAT WAS RECEIVED

"TTY INTERRUPT SEND RECEIVE ERROR" = DATA ERROR IN REMOTE
TTY INTERRUPT DRIVEN TEST.

"RIF BIT WOULD NOT SET" = RIF BIT OF CSR WOULD NOT SET.

"RIF BIT WOULD NOT CLEAR" = RIF BIT OF CSR WOULD NOT CLEAR.

"MAINT BIT 3 ON" - MAINTENANCE BIT 3 IS USED AS AN INIT-
IALIZE ON THE M8094, UPON SETTING IT, IT SHOULD CLEAR
ITSELF OUT.

"PWR FAIL BIT ON" - POWER FAIL INDICATOR BIT OF THE CSR
IS SET.

"ICSR REG, ERROR" - BIT IN CSR IS IN ERROR.

EXP'D - WHAT THE CSR SHOULD BE
REC'D - WHAT WAS READ

"NO TBMT INTERRUPT" - WITH UART TRANSMITTER BUFFER EMPTY
AND TBMT INTERRUPT ENABLE SET THERE WAS NO INTERRUPT.

"TBMT NOT SET" - TRANSMITTER EMPTY BUFFER NEVER SET.

"TBMT INT ERROR" - WITHIN MICRODIAGNOSTIC OF M8094 TEST OF TBMT
WAS FALSE.

"NO BMT INT" - WITH UART TRANSMITTER BUFFER LOADED
AND BMT INTERRUPT ENABLE SET THERE WAS NO INTERRUPT.

8. DIAGNOSTIC STRUCTURE

8.1 INITIAL DIALOGUE

THE INITIAL DIALOGUE IS AIMED TO GIVE THE USER A QUICK
VERIFICATION OF HOW THE SYSTEM IS CONFIGURED AND HOW THE
INDIVIDUAL FILE BOX(ES) ARE JUMPED WITH RESPECT TO IN-
TERRUPT VECTOR AND INTERRUPT PRIORITY LEVEL. THIS SECTION
PROVES THAT THE ICR FUNCTIONS PROPERLY AS A UNIBUS DEVICE
IN AS MUCH AS ADDRESSING AND INTERRUPTING.

8.2 CSR TEST

THE TEST OF THE CSR TESTS THAT CSR BITS SET AND CLEAR
PROPERLY AND INDIVIDUALLY.

8.3 M8094 TEST

THE MICRODIAGNOSTIC RESIDENT IN THE M8094 TESTS THAT FLOWS
THROUGH THE M8094 ARE FUNCTIONING PROPERLY. IT WILL ALSO
VERIFY THAT THE M8094 CRC CHECKING LOGIC IS WORKING
CORRECTLY, AND THAT THE M8094 LOOPS DATA PROPERLY. SEE
SECTION 9.1 FOR M8094 MICRODIAGNOSTIC CODE AND FLOWCHARTS.

8.4 M8096 TEST

THE M8096 MICRODIAGNOSTIC TESTS CERTAIN FLOWS ON THE M8096
WITHOUT INTEFERING WITH MODULES IN THE FILE BOX. THE INPUT
MODULE RANGE JUMPERS ARE CHECKED AS WELL AS THE FACT THAT
THE FOLLOWING CIRCUITRY IS FUNCTIONING CORRECTLY. THE
MODULE ADDRESSES ARE ALSO VERIFIED THAT CERTAIN BIT
PATTERNS LOAD AND READ PROPERLY. THIS TEST ALSO ALLOWS
THE USER TO TRANSMIT A FIXED PATTERN VIA THE SWITCH
REGISTER (SW08 AND SW07-00) OR VARY THE BITS FOR SCOPING
PURPOSES AGAIN BY THE SWITCH REGISTER. SEE SECTION 9.2
FOR M8096 MICRODIAGNOSTIC CODE AND FLOWCHARTS.

8.5 REMOTE TERMINAL TEST

THIS TEST EXERCISES THE UART AND ITS ASSOCIATED LOGIC, AS WELL AS THE WHOLE LINK, WITH ALL DATA PATTERNS IN BOTH FLAG AND INTERRUPT MODES. THIS TEST WILL ALSO RECORD ALL LINE ERRORS INDUCED BY SURROUNDING NOIES. (LINE ERRORS INDICATE ONLY THAT THE MASTER OR SLAVE HAD TO RESEND A MESSAGE NOT THAT ANY DATA WAS LOST).

9. MISCELLANEOUS

9.1 M8094 MICRODIAGNOSTIC

9.1.1 M8094 MICRODIAGNOSTIC STRUCTURE

THE MICRODIAGNOSTIC WITHIN THE M8094 IS CONTROLLED BY MAINTENANCE BIT 2 OF THE ICR'S CSR. SETTING THIS BIT TO A 1 WILL CAUSE THE MICROCODE TO ENTER THE MICRO-DIAGNOSTIC. TOGGLING THIS BIT WILL THEN ALLOW THE MICRO-DIAGNOSTIC TO RUN IN SYNCHRONIZATION WITH THE PDP11 DIAGNOSTIC.

THE M8094 MICRODIAGNOSTIC PERFORMS BASIC CHECKS OF THE M8094 SUCH AS MODULE INTERRUPT, TIMEOUT FLOP OPERATION, ERROR FLOP OPERATION, SEQUENCE COMPARE, DATA VALID AND TBMT. NEXT THE M8094 CRC LOGIC IS TESTED AS BOTH MASTER AND SLAVE CRC ERRORS ARE FORCED AND VERIFIED. THESE ARE DONE BY SETTING THE M8094 MAINTENANCE BIT AND FORCING THE TT BITS TO CERTAIN CONDITIONS WITH DATA BITS 6,7 OF THE OUTGOING MESSAGE AS 0,0 OR 1,1 (SEE SECTION 9.1.2 FOR EXACT TT AND DATA BIT PATTERNS). M8094 IS THEN PUT INTO A LOOP BACK MODE, WHERE THE SLAVE TURNS THE DATA AROUND AND THE MICRODIAGNOSTIC LOADS IT INTO THE ICAR AND ADDRESS RAM WHICH IS READ AND VERIFIED BY THE PDP11 DIAGNOSTIC.

9.1.2 M8094 DIAGNOSTIC FEATURES

THE FOLLOWING CRC TEST FUNCTIONS ARE DEFINED IF THE MAINTENANCE BIT IS SET:

1. MASTER MODE:

   A. FORCE AN OUTGOING CRC ERROR IN THE MASTER IF: TT=1.

2. SLAVE MODE:

   A. FORCE AN OUTGOING CRC ERROR IN THE SLAVE IF BITS 6 AND 7 OF THE LAST DATA BYTE ARE BOTH ONE.

   B. INHIBIT CRC CHECK IF BITS 6 AND 7 OF THE LAST DATA BYTE ARE BOTH ZERO.

MAINTENANCE -- THE MAINTENANCE BIT CAUSES THE SLAVE TO REPEAT THE MESSAGE BACK TO THE MASTER. IF DATA IS PART OF THE MESSAGE, THE LAST DATA BYTE RECEIVED IS SEND BACK 'N' TIMES WHERE 'N' EQUALS THE NUMBER OF DATA BYTES INDICATED BY THE RECEIVED LENGTH BITS.

ICR11 DIAGNOSTIC MICROCODE (M8094)

LOCATION	INSTRUCTION	CODE	COMMENT
0	TST MAINT	00000011	MAINT SET?
1	CJMP .+2	11110011	YES, CONT
2	CLR PC	1101XXXX	NO, START AGAIN
3	LOAD MODINT	00010001	CREATE INTERRUPT
4	TST MAINT	00000011	MAINT SET?
5	CJMP .-1	11110100	YES, HANG
6	TST TIMOUT	00000000	TIMEOUT SET?
7	CJMP .+2	11111001	YES, CREATE INT.
10	JMP .+2	11101010	NO, DON'T INT
11	LOAD MODINT	00010001	CREATE INTERRUPT
12	TST MAINT	00000011	MAINT SET?
13	CJMP .+4	11111111	YES, CONT
14	JMP .-2	11101010	NO, HANG
15			
16			
17	INC SEQ	1010XXXX	INC SEQUENCE
0	TST SEQ	00000001	SEQUENCE COMP?
1	CJMP .+5	11110110	YES, CREATE INT.
2	INC SEQ	1010XXXX	INC SEQUENCE
3	TST SEQ	00000001	SEQUENCE COMP?
4	CJMP .+2	11110110	YES, CREATE INT
5	JMP .+2	11100111	NO, DON'T INT
6	LOAD MODINT	00010001	CREATE INTERRUPT
7	TST MAINT	00000011	MAINT SET?
10	CJMP .-1	11110111	YES, HANG
11	LOAD ERROR	0011XXXX	CREATE ERROR INTERRUPT
12	TST MAINT	00000011	MAINT SET?
13	CJMP .+4	11111111	YES, CONT
14	JMP .-2	11101010	NO, HANG
15			
16			
17	TST DATA VALID	00000010	DATA VALID TRUE?
0	CJMP .+2	11110010	YES, DON'T INT.
1	LOAD MODINT	00010001	CREATE INTERRUPT
2	TST MAINT	00000011	MAINT SET?
3	CJMP .-1	11110010	YES, HANG
4	CLK FLAGS	0100XXXX	CLOCK FLAGS
5	TST TBMT	00000101	TSMT TRUE
6	CJMP .+2	11111000	YES, INTERRUPT
7	JMP .+2	11101001	NO, DON'T INTR.
10	LOAD MODINT	00010001	CREATE INTERRUPT
11	TST MAINT	00000011	MAINT SET?
12	CJMP .+2	11111100	YES, R
13	JMP .-2	11101001	NO, HANG

```
14      TST TBMT        00000101        TBMT SET?
15      CJMP .-1        11111100        YES, WAIT
16      CLK FLAGS       0100XXXX        CLOCK FLAGS
17      TDP 1011        00101011        ISSUE TDP

0       TST RDP         00000111        RECV DATA PRESENT?
1       CJMP .+4        11110101        YES, CONT
2       TST TIMEOUT     00000000        TIMEOUT?
3       CJMP .+4        11110111        YES, INDICATE
4       JMP .-4         11100000        LOOP BACK
5       SET MODINT      00010001        SET MOD INTR
6       JMP .+2         01101000        CONT
7       SET ERROR       0011XXXX        SET ERROR
10      TST MAINT       00000011        MAINT SET?
11      CJMP .+2        11111011        YES, THEN LOOP
12      JMP .+2         11101100        NO, CONT
13      CIF 12          10001010        LOOP
14      TST TBMT        00000101        TBMT SET?
15      CJMP .-1        11111100        YES, HANG
16      CLK FLAGS       0100XXXX        CLOCK FLAGS
17      TDP 1001        00101001        ISSUE TDP

0       TST TIMEOUT     00000000        TIMEOUT SET?
1       CJMP .+4        11110101        YES, SET DA
2       TST RDP         00000111        RECV DATA PRESENT?
3       CJMP .+4        11110111        YES, CONT
4       JMP .-4         11100000        LOOP
5       SET DA          01110000        SET DA
6       JMP .+4         11101010        CONT
7       TST DATA VALID  00000010        DATA VALID?
10      CJMP .+2        11111010        YES, CONT
11      SET MODINT      00010001        NO, SET MOD INT
12      JMP .+2         11101100        CONT
13      CIF 15          10001011        LOOP
14      TST MAINT       00000011        MAINT SET?
15      CJMP .+2        11111111        YES, CONT
16      JMP .-3         11101011        NO, LOOP
17      CLK FLAGS       0100XXX1        CLOCK FLAGS

0       CLR MODINT      00010000        CLEAR MOD INTR
1       TDP 1010        00101010        ISSUE TDP
2       TST RDP         00000111        RECV DATA PRESENT?
3       CJMP    .+5     11111000        YES, CONT
3       CJMP .+5        11111000        YES, CONT
4       TST TIMEOUT     00000000        TEST TIMEOUT
5       CJMP .+2        11110111        SET, INDICATE
6       JMP .-4         11100010        LOOP ON TEST
7       CIF 16          10001110        INDICATE TIMEOUT
10      TST RDP         00000111        WAIT TIL RDP
11      CJMP .-1        11111000        GOES AWAY
12      TST DATA VALID  00000010        TEST DATA VALID
13      CJMP .+2        11111101        CONT
14      CIF 16          10001110        DATA INVALID
15      LD GEN          0110XXXX        LOAD ICAR
16      LD MEM          1001XXXX        LOAD MEM FILE
17      NOP             1011XXXX        NO GENERATION
```

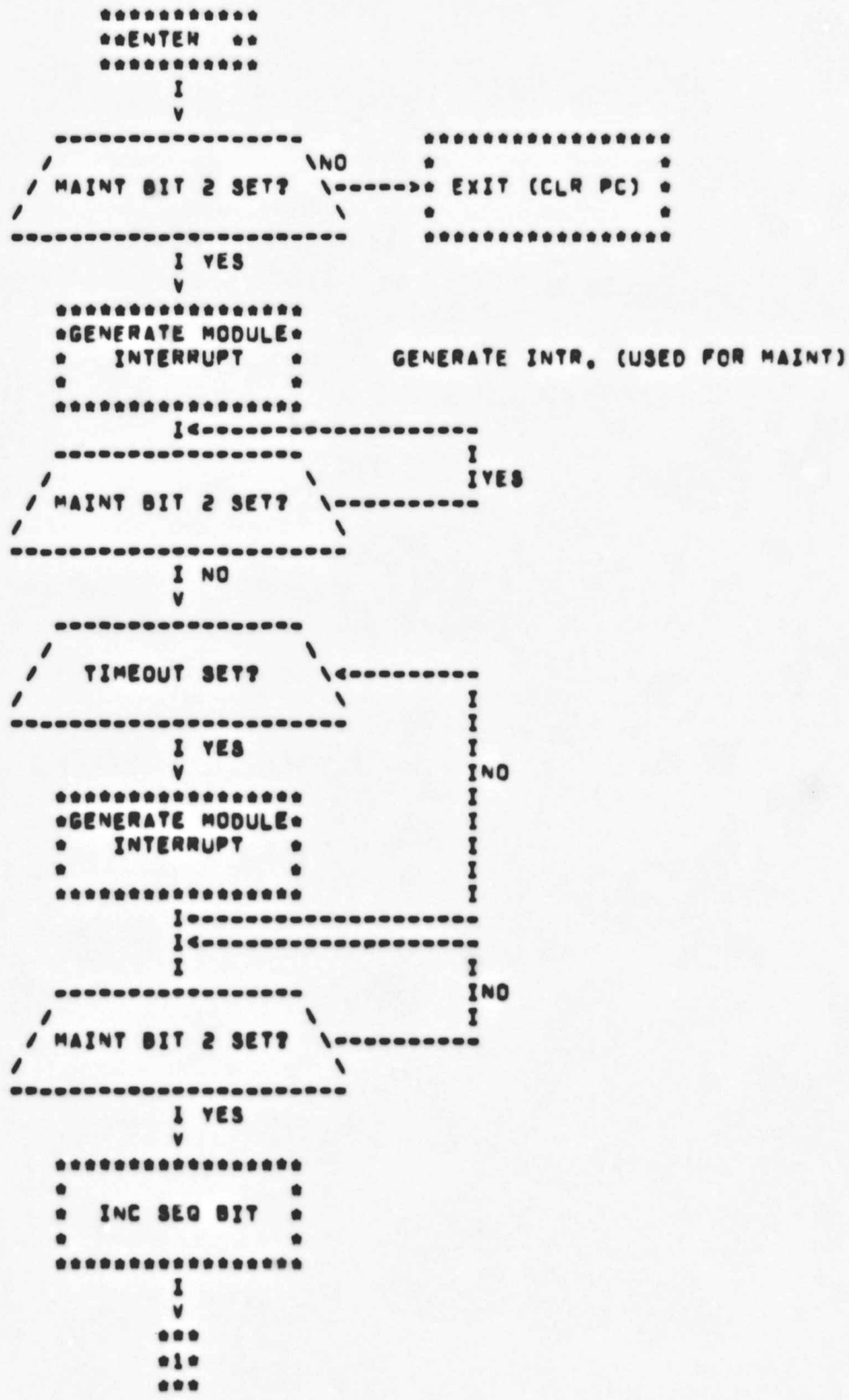
```
0       LOAD MODINT     00010001        LOAD MOD INTR
1       TST TBMT        00000101        TBMT SET?
2       CJMP .-1        11110001
3       TST MAINT       00000011        MAINT SET?
4       CJMP .+12       11111110        LOOP ON TEST
5       CLR PC          1101XXXX        RETURN
6
7
10      LD DA           01110000        TIMEOUT INDICATION
11      JMP .-11        11100000        CONT
12      LD ERROR        0011XXXX        DATA INVALID
13      JMP .-13        11100000        CONT
14
15      JMP .-3         11101010        DATA INVALID
16      CIF 14          10001100        LOOP ON TEST
17
```

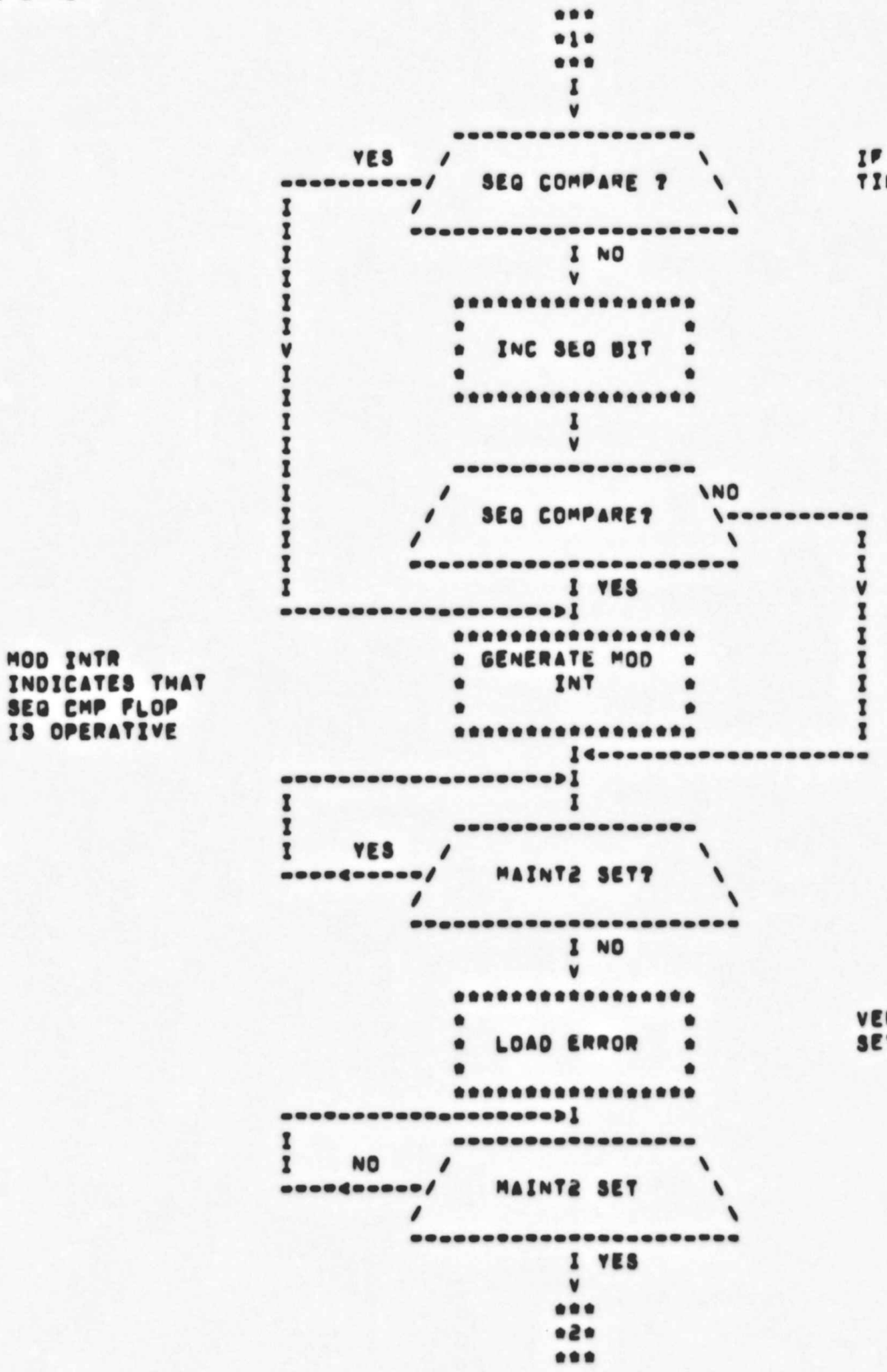
FLOW CHART
****

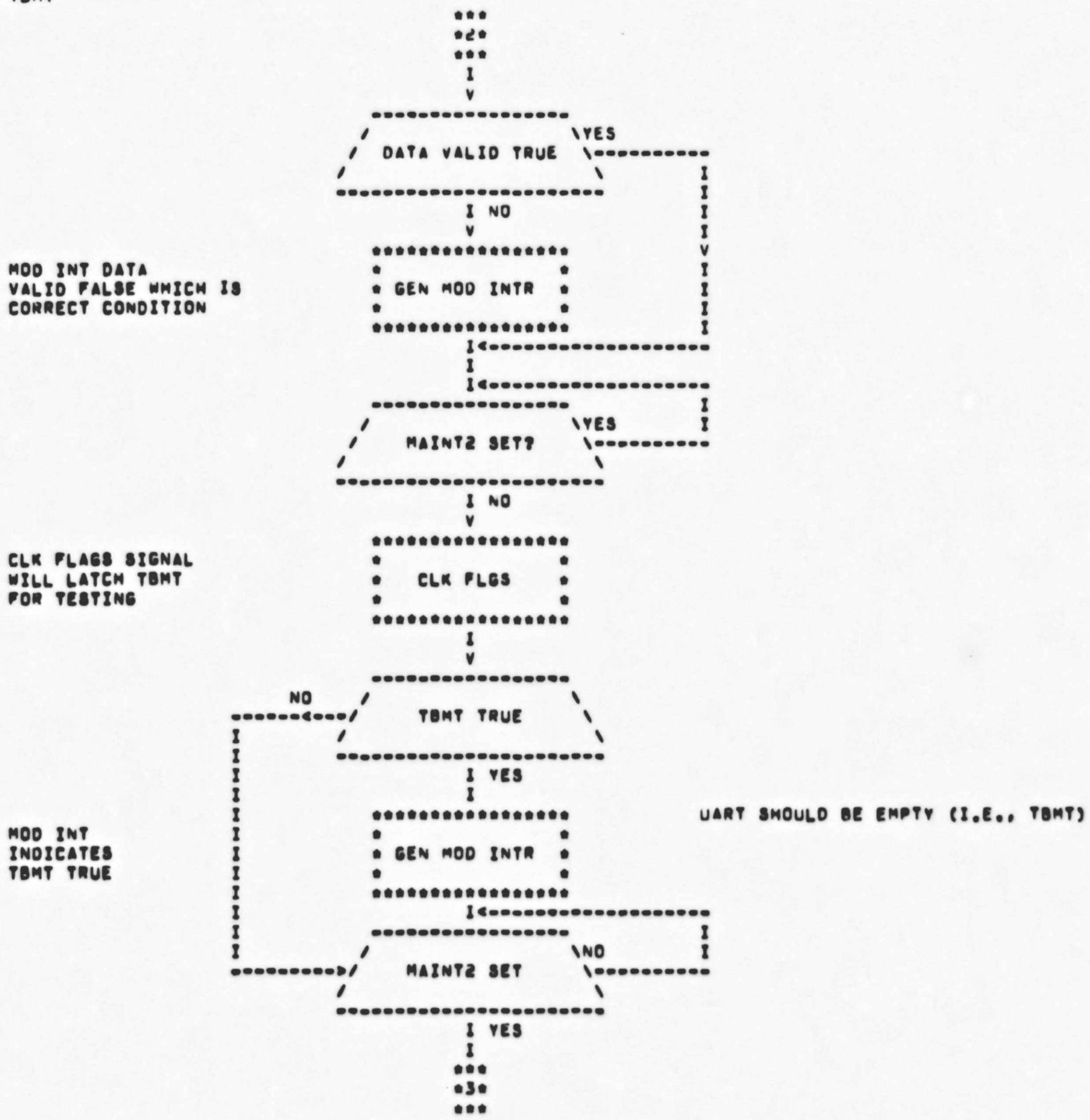
8094 DIAGNOSTIC MICRO-FLOW

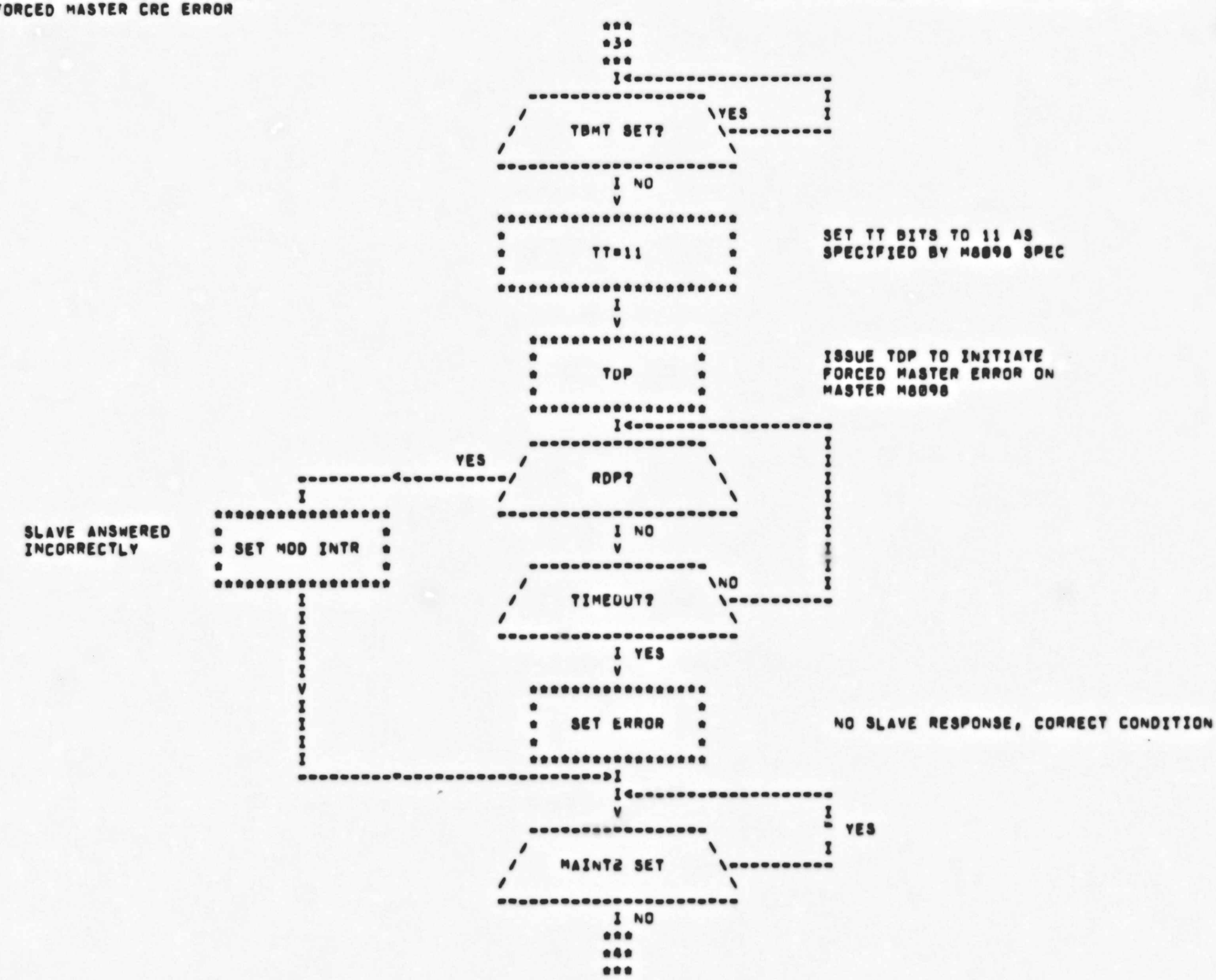
****

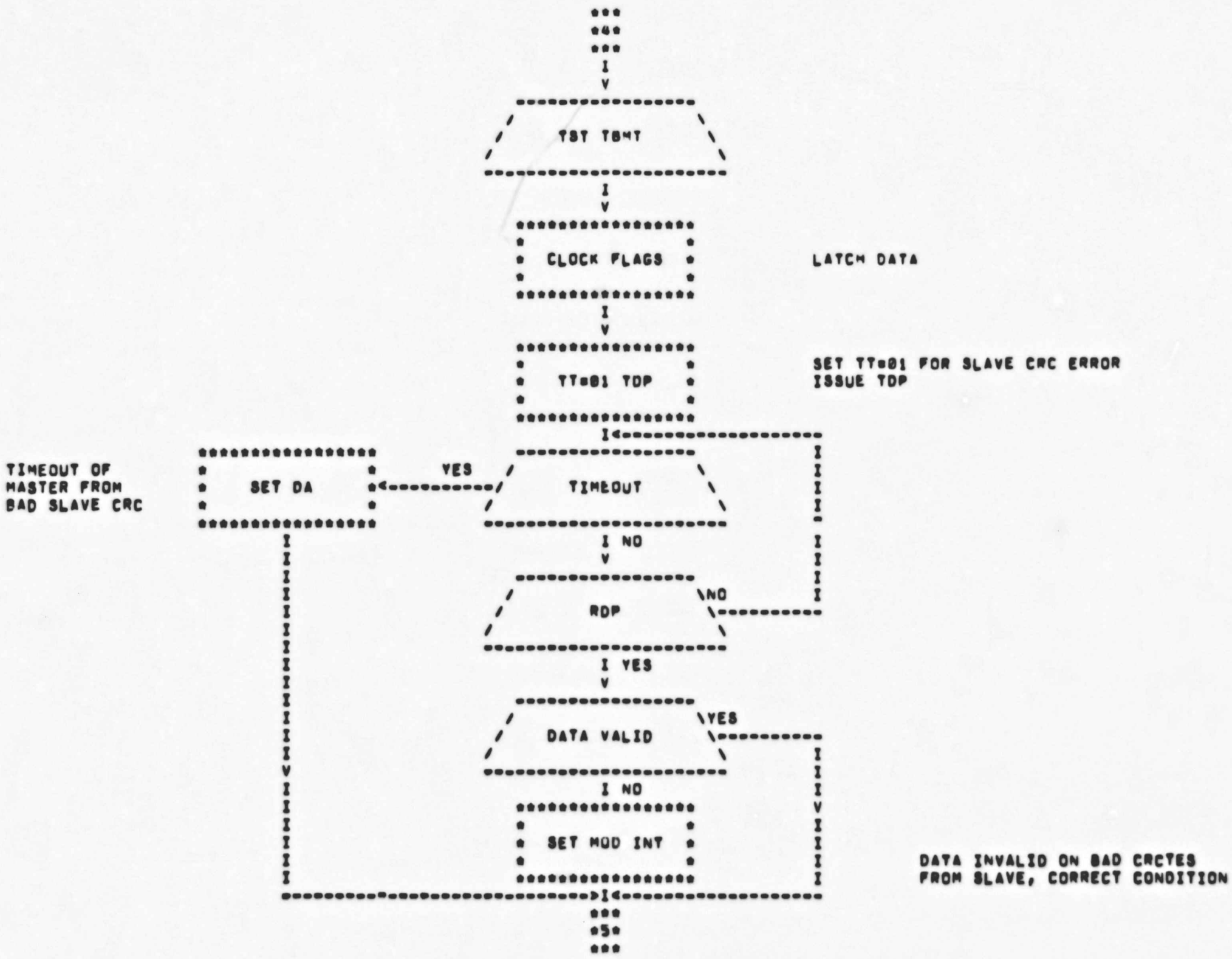
TABLE OF CONTENTS
****

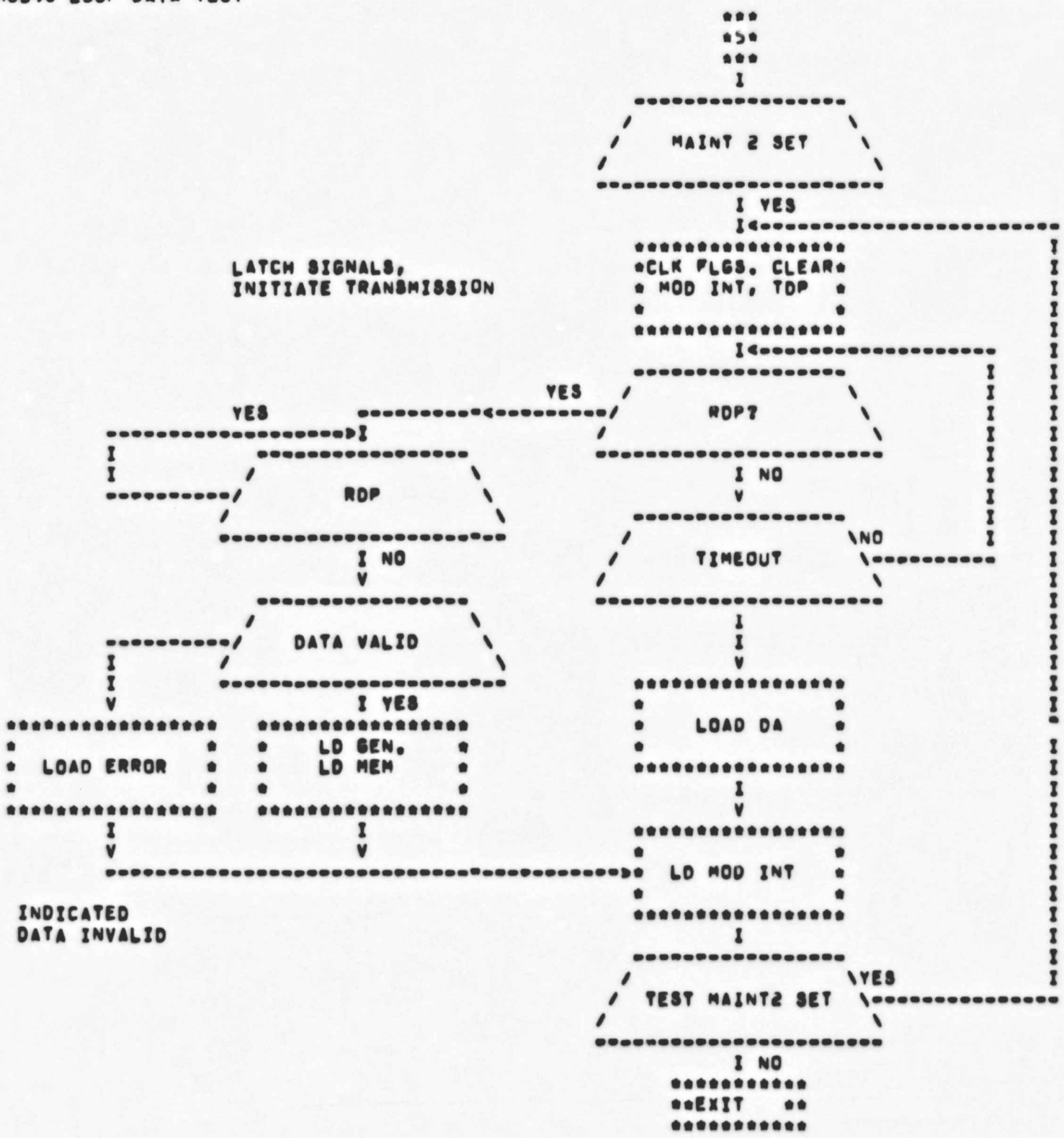

ENTER
MAINT BIT 2 SET?
NO
EXIT (CLR PC)
YES
GENERATE MODULE INTERRUPT
GENERATE INTR. (USED FOR MAINT)
MAINT BIT 2 SET?
YES
NO
TIMEOUT SET?
YES
NO
GENERATE MODULE INTERRUPT
MAINT BIT 2 SET?
YES
NO
MODULE INTERRUPT INDICATES TIMEOUT PRESENT, WHICH SHOULD HAPPEN WITH NO TDP FROM M8094
INC SEQ BIT
SEQUENCE COMPARE FLIP-FLOP CHECK
1


1
SEQ COMPARE ?
YES
NO
IF IT DOESN'T COMPARE THIS TIME, TRY ONE MORE
INC SEQ BIT
SEQ COMPARE?
NO
YES
MOD INTR INDICATES THAT SEQ CMP FLOP IS OPERATIVE
GENERATE MOD INT
MAINT2 SET?
YES
NO
LOAD ERROR
VERIFY THAT ERROR BIT WILL SET IN ICR CSR
MAINT2 SET
NO
YES
2


2
DATA VALID TRUE
YES
NO
MOD INT DATA VALID FALSE WHICH IS CORRECT CONDITION
GEN MOD INTR
MAINT2 SET?
YES
NO
CLK FLAGS SIGNAL WILL LATCH TBMT FOR TESTING
CLK FLGS
TBMT TRUE
NO
YES
UART SHOULD BE EMPTY (I.E., TBMT)
MOD INT INDICATES TBMT TRUE
GEN MOD INTR
MAINT2 SET
NO
YES
3


3
TBMT SET?
YES
NO
TT=11
SET TT BITS TO 11 AS SPECIFIED BY M8098 SPEC
TDP
ISSUE TDP TO INITIATE FORCED MASTER ERROR ON MASTER M8098
RDP?
YES
NO
SLAVE ANSWERED INCORRECTLY
SET MOD INTR
TIMEOUT?
NO
YES
SET ERROR
NO SLAVE RESPONSE, CORRECT CONDITION
MAINT2 SET
YES
NO
4


TST TBMT
CLOCK FLAGS
LATCH DATA
TT=01 TDP
SET TT=01 FOR SLAVE CRC ERROR
ISSUE TDP
TIMEOUT OF MASTER FROM BAD SLAVE CRC
SET DA
YES
TIMEOUT
NO
RDP
NO
YES
DATA VALID
YES
NO
SET MOD INT
DATA INVALID ON BAD CRCTES FROM SLAVE, CORRECT CONDITION
5


5
MAINT 2 SET
I YES
CLK FLGS, CLEAR
* MOD INT, TDP *
LATCH SIGNALS,
INITIATE TRANSMISSION
YES
RDP?
YES
RDP
I NO
TIMEOUT
NO
I NO
DATA VALID
I YES
LOAD ERROR
LD GEN.
LD MEM
LOAD DA
LD MOD INT
INDICATED
DATA INVALID
TEST MAINT2 SET
YES
I NO
**EXIT **

K2

9.2 M8096 MICRODIAGNOSTIC

9.2.1 M8096 MICRODIAGNOSTIC STRUCTURE

THE M8096 MICRODIAGNOSTIC IS CONTROLLED BY MAINTENANCE BITS 0 AND 1. THESE BITS ARE PASSED TO M8096 WHICH THEN ENTERS THE MICRODIAGNOSTIC WHICH PERFORMS TWO BASIC CHECKS OF THE M8096. THESE CHECKS ARE TEST OF THE M8096 POLLING SCHEME AND THE ABILITY TO READ ALL MODULE ADDRESS AND CONTENTS OF THE FILE BOX.

9.2.2 M8096 MICRODIAGNOSTIC CODE

ICR11 DIAGNOSTIC CODE (M8096)

LOCATION	INSTRUCTION	CODE	COMMENT
0	NOP	1100XXXX	NOP
1	NOP	1100XXXX	NOP
2	TST DATA VALID	00000111	TEST DATA VALID
3	CJMP .+2	11110101	TRUE, CONT
4	JMP .-2	11100010	NO, HANG
5	TST MAINT	00001001	MAINT SET?
6	CJMP .+2	11111000	YES
7	CLR PC	1101XXXX	NO, RETURN
10	TST LMD	00000000	LMD SET? (INDICATING SYNC WITH PDP11 DIAGNOSTIC)
11	CJMP .+2	11111011	YES, CONT
12	JMP .+4	11101110	NO, JUST TDP
13	SET MODINT	01100001	SET MOD INTERRUPT
14	INC MOD ADDR	0101XXXX	INC INPUT ADDRESS
15	SEL M ADDR	00100000	SEL POLL ADDRESS
16	TDP	01110100	TRANS DATA PRESENT
17	NOP	1100XXXX	NOP
0	JMP .+2	11100010	SKIP CIF
1	CIF 14	10001100	CHANGE TO 14
2	TST DONE	00001000	DONE?
3	CJMP .+2	11110101	YES, CONT
4	JMP .-2	11100010	NO, WAIT
5	CLR MODINT	01100000	CLEAR MOD INT
6	TST DATA VALID	00000111	DATA VALID?
7	CJMP .+2	11111001	YES, CONT
10	JMP .-2	11100110	NO, WAIT
11	TST MAINT	00001001	MAINT SET?
12	CJMP .-11	11110001	YES, GO TO CIF
13	NOP	1100XXXX	NOP
14	NOP	1100XXXX	NOP
15	NOP	1100XXXX	NOP
16	NOP	1100XXXX	NOP
17	NOP	1100XXXX	NOP

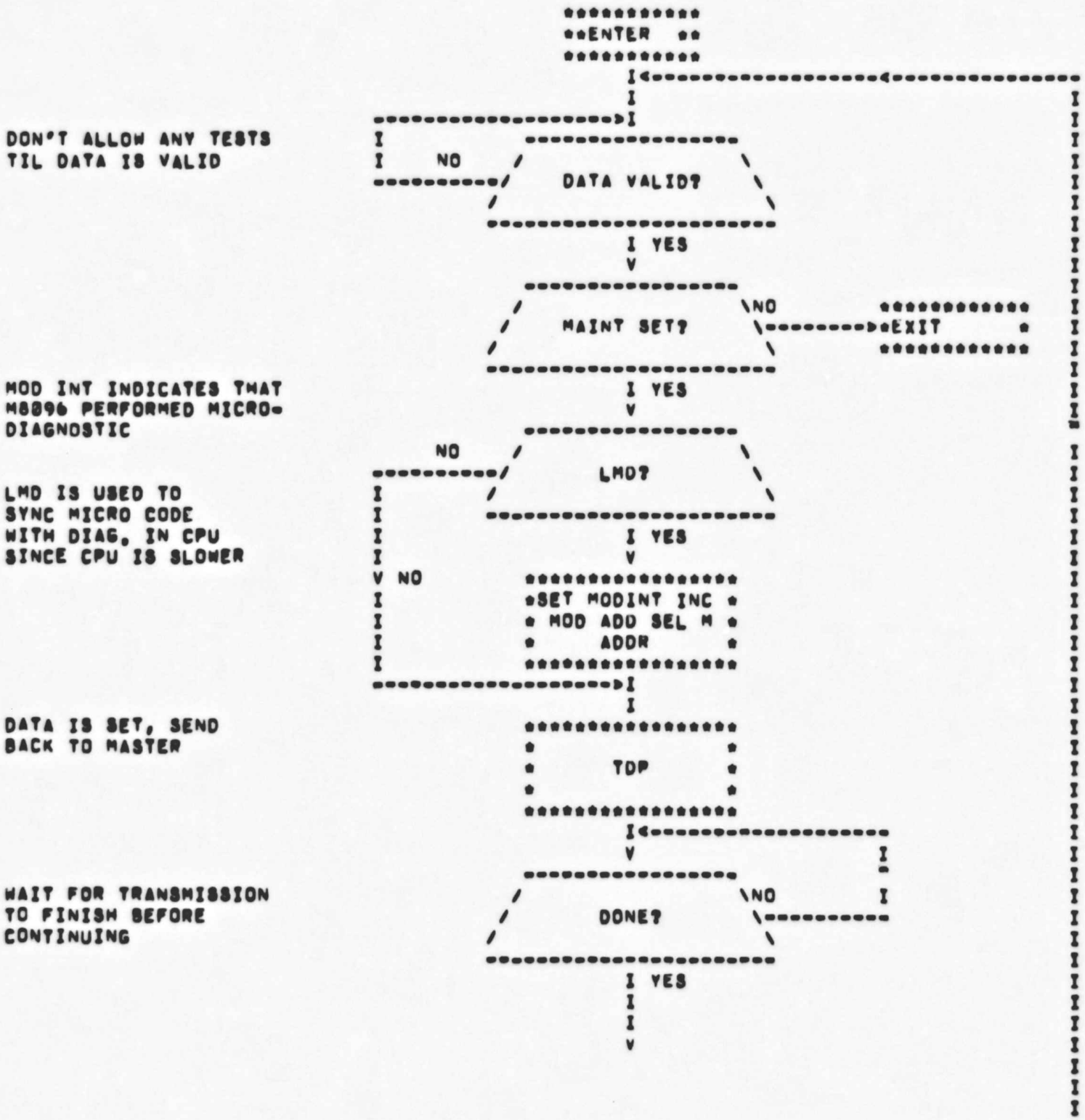
0	TST RMD	00000001	RMD SET? (INDICATING SYNC WITH PDP11 DIAGNOSTIC)	SEQ 0026
1	CJMP .+3	11110100	YES, JUST TDP	
2	SET MOD INT	01100001	SET MOD INT	
3	SEL L ADDR	00100001	SET INPUT ADDR	
4	TDP	01110100	TRANS DATA PRESENT	
5	TST DONE	00001000	DONE	
6	CJMP .+2	11111000	YES, CONT	
7	JMP .-2	11100101	NO, WAIT	
10	CLR MODINT	01100000	CLR MOD INT	
11	TST DATA VALID	00000111	DATA VALLID?	
12	CJMP .+2	11111100	YES, CONT	
13	JMP .-2	11101001	NO, WAIT	
14	TST MAINT	00001001	MAINT SET?	
15	CJMP .+2	11111111	YES, CONT	
16	JMP .-16	11100000	NO, LOOP	
17	NOP	1100XXXX	NOP	
0	TDP	01110100	TRANS DATA PRESENT	
1	TST DONE	00001000	DONE?	
2	CJMP .+2	11110100	YES, CONT	
3	JMP .-2	11100001	NO, WAIT	
4	TST DATA VALID	00000111	DATA VALID?	
5	CJMP .+2	11110111	YES, CONT	
6	JMP .-2	11100100	NO, WAIT	
7	TST MAINT	00001001	MAINT SET?	
10	CJMP .-10	11110000	YES, HANG	
11	TDP	01110100	NO, TRANS DATA PRES	
12	TST DONE	00001000	DONE?	
13	CJMP .+2	11111101	YES, CONT	
14	JMP .-2	11101010	NO, WAIT	
15	CLR PC	1101XXXX	RETURN	
16				
17				

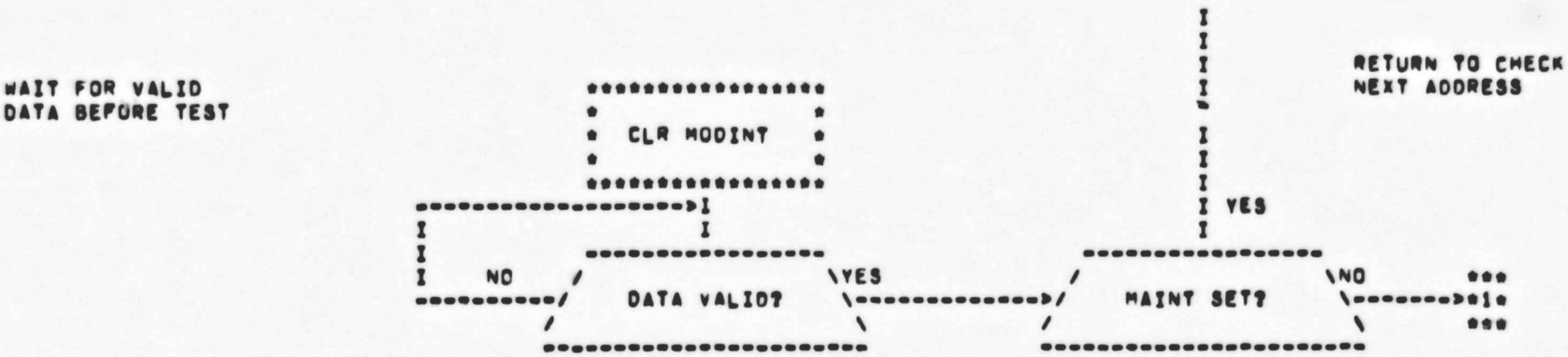
FLOW CHART
****

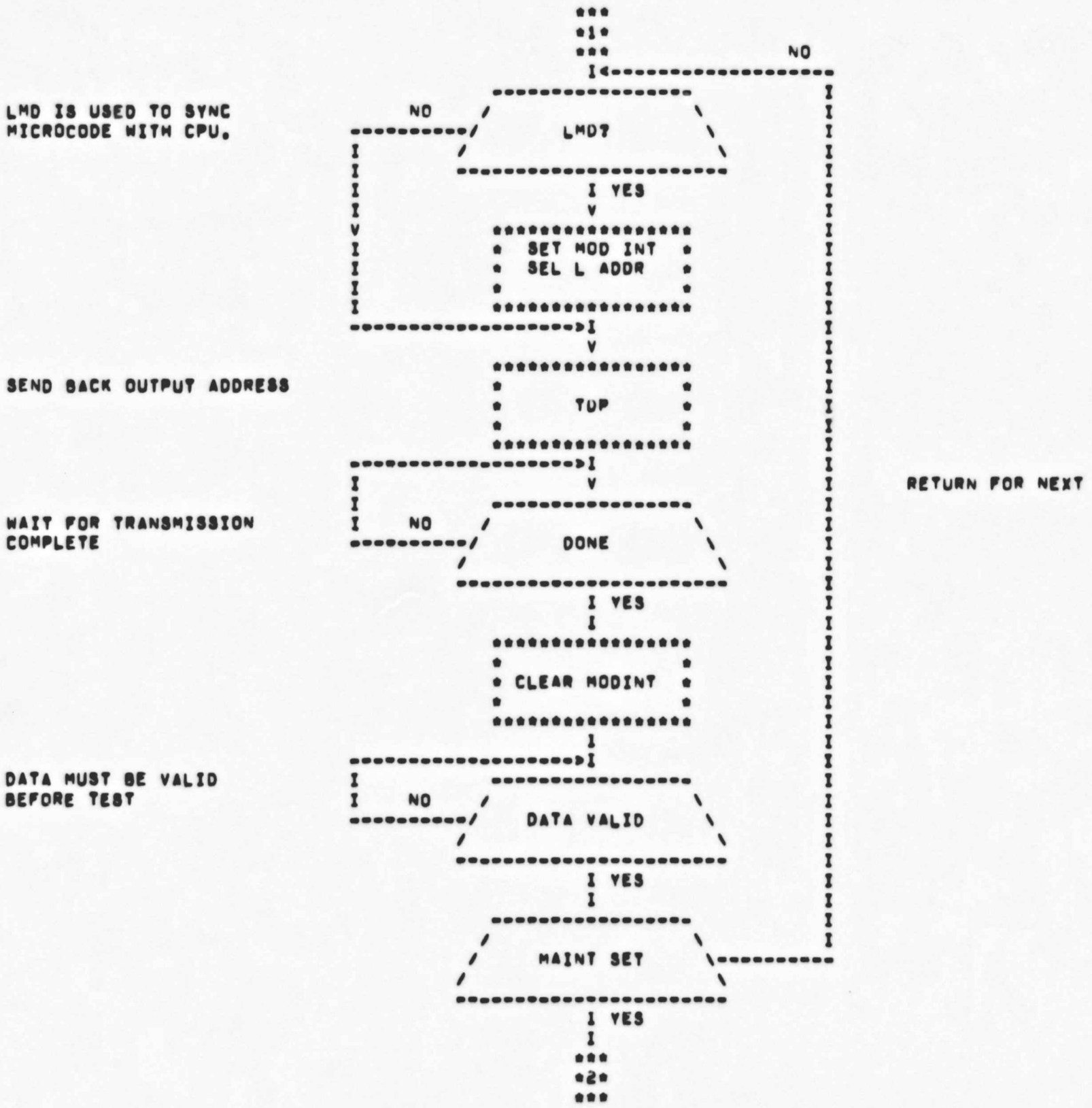
M8096 DIAGNOSTIC MICRO-FLOW

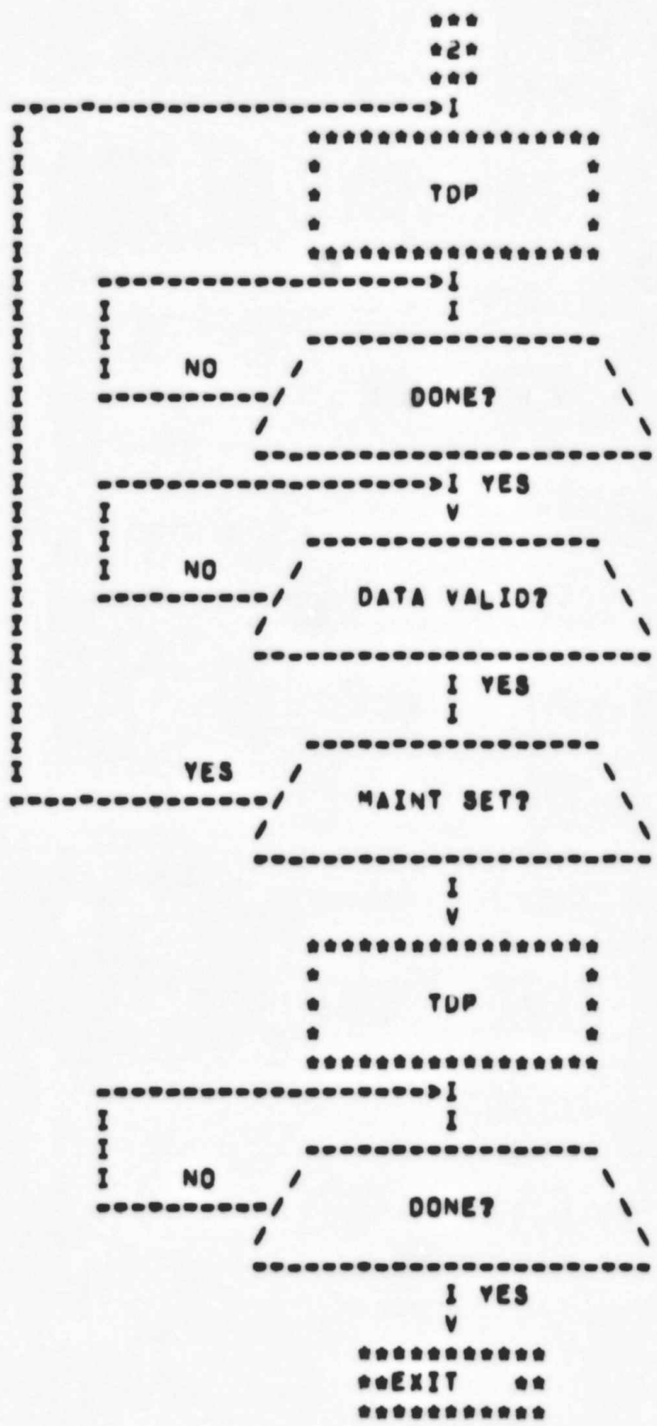
****

TABLE OF CONTENTS
****


ENTER
DON'T ALLOW ANY TESTS TIL DATA IS VALID
NO
DATA VALID?
YES
MAINT SET?
NO
EXIT
MOD INT INDICATES THAT M8096 PERFORMED MICRO-DIAGNOSTIC
NO
LMD?
LMD IS USED TO SYNC MICRO CODE WITH DIAG. IN CPU SINCE CPU IS SLOWER
YES
NO
SET MODINT INC
MOD ADD SEL M
ADDR
DATA IS SET, SEND BACK TO MASTER
TOP
WAIT FOR TRANSMISSION TO FINISH BEFORE CONTINUING
NO
DONE?
YES


WAIT FOR VALID
DATA BEFORE TEST
CLR MODINT
NO
DATA VALID?
YES
YES
MAINT SET?
NO
RETURN TO CHECK
NEXT ADDRESS


LMD IS USED TO SYNC MICROCODE WITH CPU.
NO
LMD?
I YES
SET MOD INT
SEL L ADDR
SEND BACK OUTPUT ADDRESS
TDP
RETURN FOR NEXT
WAIT FOR TRANSMISSION COMPLETE
NO
DONE
I YES
CLEAR MODINT
DATA MUST BE VALID BEFORE TEST
NO
DATA VALID
I YES
MAINT SET
I YES
1
2



DUMMY MESSAGE BACK

THIS EXIT SCHEME ALLOWS SMOOTH RETURN TO M8096 MICROCODE

DUMMY MESSAGE

```
                        ;
                        ;ICR-11 DIAGNOSTIC
                        ;
                        
                        .TITLE  MAINDEC-11-DZIRA-A
                        ;*COPYRIGHT (C) 1975
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY DAN DEKNIS
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-B1),AUG 29,1975.
                        ;*
000001                  STN=1
160000                  SSWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
                        
                        .EVEN
                        
                        
                                .SBTTL  SWITCH REGISTER CONTROL SETTINGS
                        
                                ;SW00   - FILE
                        
                                ;SW01   - BOX
                        
                                ;SW02   - UNDER
                        
                                ;SW03   - TEST
                        
                                ;SW04   -0-     TEST ALL FILES ON SYSTEM
                                ;       -1-     TEST FILE  SPECIFIED BY SW03-00
                        
                                ;SW05   -0-
                                ;       -1-     LOOP ON TEST SPECIFIED BY SW07,06
                        
                                ;SW06   - TEST          SW07 = 0  SW06 = 0  CSR TEST
                                ;                       SW07 = 0  SW06 = 1  M8094 TEST
                                ;SW07   - NUMBER        SW07 = 1  SW06 = 0  M8096 TEST
                                ;                       SW07 = 1  SW06 = 1  REMOTE TTY TEST
                        
                                ;SW08   -0-
                                ;       -1-     LOOP ON M8098 DATA LOOP BACK TEST
                                ;               (USES SW08-07 FOR DATA PATTERN)
                                ;               NOTE:  SW06,07  SHOULD BE 0,1 OR 1,0
                                ;                      SW06,07 AS 0,0 WILL DEFAULT TO 0,1
                                ;                      SW06,07 AS 1,1 WILL DEFAULT TO 1,0
                        
                                ;SW09   -0-     HALT AT END OF PASS
                                ;       -1-     LOOP ON DIAGNOSTIC
                        
                                ;SW10   -0-     PRINT ON LOCAL CONSOLE
                                ;       -1-     PRINT ON LOCAL AND REMOTE TERMINAL
                        
                                ;SW11   -0-
                                ;       -1-
```

```
                                        ;SW12   -0-     QUICK VERIFY
                                        ;       -1-     LONG TEST
                                        ;
                                        ;SW13   -0-
                                        ;       -1-     SUPPRESS ERROR TYPEOUT
                                        ;
                                        ;SW14   -0-
                                        ;       -1-     SCOPE LOOP
                                        ;
                                        ;SW15   -0-
                                        ;       -1-     HALT ON ERROR
                                        ;
                                        ;
                                        ;
                                .SBTTL  TRAP CATCHER
                                        ;
  000000                                .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
  000174                                .=174
000174  000000                        DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                        SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                        ;
                                .SBTTL  STARTING ADDRESS(ES)
000200  000137  002070                JMP     @#START         ;;JUMP TO STARTING ADDRESS OF PROGRAM
                                        ;
                                        ;
  000204                                .=204
                                        ;
000204  000137  005554                JMP     @#TEST2
                                        ;
  000210                                .=210
                                        ;
000210  000137  003002                JMP     @#START1
                                        ;
  000214                                .=214
                                        ;
000214  000137  011330                JMP     @#TEST3
                                        ;
  000220                                .=220
                                        ;
000220  000137  007032                JMP     @#TTYSTR
                                        ;
                                ;START IF POWER FAIL TEST WANTED
                                        ;
  000224                                .=224
                                        ;
000224  000137  002064                JMP     @#SETPWF
                                        ;
                                        ;
                                        ;
                                .SBTTL  BASIC DEFINITIONS
                                        ;
```

```
                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                  STACK=  1100
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
                        .EQUIV  R6,SP           ;;STACK POINTER
                        .EQUIV  R7,PC           ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
```

```
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
            100000              BIT15=  100000
            040000              BIT14=  40000
            020000              BIT13=  20000
            010000              BIT12=  10000
            004000              BIT11=  4000
            002000              BIT10=  2000
            001000              BIT09=  1000
            000400              BIT08=  400
            000200              BIT07=  200
            000100              BIT06=  100
            000040              BIT05=  40
            000020              BIT04=  20
            000010              BIT03=  10
            000004              BIT02=  4
            000002              BIT01=  2
            000001              BIT00=  1
                                .EQUIV  BIT09,BIT9
                                .EQUIV  BIT08,BIT8
                                .EQUIV  BIT07,BIT7
                                .EQUIV  BIT06,BIT6
                                .EQUIV  BIT05,BIT5
                                .EQUIV  BIT04,BIT4
                                .EQUIV  BIT03,BIT3
                                .EQUIV  BIT02,BIT2
                                .EQUIV  BIT01,BIT1
                                .EQUIV  BIT00,BIT0

                                ;*BASIC "CPU" TRAP VECTOR ADDRESSES
            000004              ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
            000010              RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
            000014              TBITVEC=14              ;;"T" BIT
            000014              TRTVEC= 14              ;;TRACE TRAP
            000014              BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
            000020              IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
            000024              PWRVEC= 24              ;;POWER FAIL
            000030              EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
            000034              TRAPVEC=34              ;;"TRAP" TRAP
            000060              TKVEC=  60              ;;TTY KEYBOARD VECTOR
            000064              TPVEC=  64              ;;TTY PRINTER VECTOR
            000240              PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
            000030              .=30
000030  014500                          EDISPT
000032  000000                          0
```

K3

```
                        ;*****************************************************************

                        .SBTTL  COMMON TAGS

                        ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                        ;*USED IN THE PROGRAM.

                                .=1100
001100                  SCMTAG:                 ;;START OF COMMON TAGS
001100  000000         SPASS:  .WORD   0        ;;CONTAINS PASS COUNT
001102  000             STSTNM: .BYTE   0        ;;CONTAINS THE TEST NUMBER
001103  000             SERFLG: .BYTE   0        ;;CONTAINS ERROR FLAG
001104  000000         SICNT:  .WORD   0        ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000         SLPADR: .WORD   0        ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000         SLPERR: .WORD   0        ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000         SERTTL: .WORD   0        ;;CONTAINS TOTAL ERRORS DETECTED
001114  000             SITEMB: .BYTE   0        ;;CONTAINS ITEM CONTROL BYTE
001115  001             SERMAX: .BYTE   1        ;;CONTAINS MAX. ERRORS PER TEST
001116  000000         SERRPC: .WORD   0        ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000         SGDADR: .WORD   0        ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000         SBDADR: .WORD   0        ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000         SGDDAT: .WORD   0        ;;CONTAINS 'GOOD' DATA
001126  000000         SBDDAT: .WORD   0        ;;CONTAINS 'BAD' DATA
001130  000000                 .WORD   0        ;;RESERVED--NOT TO BE USED
001132  000000                 .WORD   0
001134  000000                 .WORD   0
001136  177570         SWR:    .WORD   DSWR     ;;ADDRESS OF SWITCH REGISTER
001140  177570         DISPLAY: .WORD  DDISP    ;;ADDRESS OF DISPLAY REGISTER
001142  177560         STKS:   177560           ;;TTY KBD STATUS
001144  177562         STKB:   177562           ;;TTY KBD BUFFER
001146  177564         STPS:   177564           ;;TTY PRINTER STATUS REG. ADDRESS
001150  177566         STPB:   177566           ;;TTY PRINTER BUFFER REG. ADDRESS
001152  000             SNULL:  .BYTE   0        ;;CONTAINS NULL CHARACTER FOR FILLS
001153  002             SFILLS: .BYTE   2        ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001154  012             SFILLC: .BYTE   12       ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001155  000             STPFLG: .BYTE   0        ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001156  077             SQUES:  .ASCII  /?/      ;;QUESTION MARK
001157  015             SCRLF:  .ASCII  <15>     ;;CARRIAGE RETURN
001160  000012         SLF:    .ASCIZ  <12>     ;;LINE FEED
```

```
                        ;*****************************************************************

                        .SBTTL  ERROR POINTER TABLE

                        ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                        ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                        ;*LOCATION SITEMB, THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                        ;*NOTE1:        IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
                        ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                        ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                        ;*      DH              ;;POINTS TO THE DATA HEADER
                        ;*      DT              ;;POINTS TO THE DATA
                        ;*      DF              ;;POINTS TO THE DATA FORMAT

001162                  SERRTB:


                        ;
                        ;

                                .SBTTL  ERROR DEFINITIONS

                        ;ERROR 1 - OUTPUT IS BUSY (BIT 15) IS SET

001162  022136                  EM1
001164  000000                  0
001166  000000                  0
001170  000000                  0

                        ;ERROR 2 - RESET NOT READ AS 0

001172  022155                  EM2
001174  000000                  0
001176  000000                  0
001200  000000                  0

                        ;ERROR 3 - TTY ENABLE NOT SET

001202  022173                  EM3
001204  000000                  0
001206  000000                  0
001210  000000                  0

                        ;ERROR 4 - TTY ENABLE NOT CLEAR

001212  022213                  EM4
001214  000000                  0
001216  000000                  0
001220  000000                  0

                        ;ERROR 5 - TIMEOUT FLOP INOPERATIVE

001222  022265                  EM5
001224  000000                  0
```

```
                                ;ERROR 6 - SEQUENCE COMPARE FLOP INOPERATIVE
                                ;ERROR 7 - TBMT INTERRUPT DID NOT INTERRUPT
                                ;ERROR 10 - ERRONOUS CAUSING ERROR
                                ;ERROR 11 - ERROR BIT DID NOT INTERRUPT
                                ;ERROR 12 - MODULE INTERRUPT DID NOT CLEAR WITH RIF
                                ;ERROR 13 - ERROR INTERRUPT BIT DID NOT CLEAR WITH ADDRESSING ICAR
                                ;ERROR 14 - DATA VALID WAS TRUE WHEN IT WASN'T SUPPOSE TO BE
                                ;ERROR 15 - TBMT WAS NOT SET
```

```
                                ;ERROR 16 - TBMT WOULD NOT INTERRUPT
001332  022477                          EM16
001334  000000                          0
001336  000000                          0
001340  000000                          0

                                ;ERROR 17 - BMT WILL NOT INTERRUPT
001342  022513                          EM17
001344  000000                          0
001346  000000                          0
001350  000000                          0

                                ;ERROR 20 - INPUT MODULE ADDRESS ERROR
001352  023214                          EM20
001354  025416                          DM1
001356  025620                          DT1
001360  000000                          0

                                ;ERROR 21 - MODULE INTERRUPT WILL NOT INTERRUPT
001362  023233                          EM21
001364  000000                          0
001366  000000                          0
001370  000000                          0

                                ;ERROR 22 - MODULE ADDRESS WRAP AROUND INCORRECT
001372  023246                          EM22
001374  025416                          DM1
001376  025620                          DT1
001400  000000                          0

                                ;ERROR 23 - TTY SEND RECEIVE ERROR
001402  023266                          EM23
001404  025416                          DM1
001406  025620                          DT1
001410  000000                          0

                                ;ERROR 24 - BMT INTERRUPTED WITH OUTPUT BUSY
001412  023315                          EM24
001414  000000                          0
001416  000000                          0
001420  000000                          0

                                ;ERROR 25 - NO INTERRUPT FROM MULTIPLE INTERRUPTS
001422  023331                          EM25
001424  000000                          0
```

```
                                        0
                                        0
                        ;ERROR 26 - MULTIPLE MODULE INTERRUPTS OCCURED
                                        EM26
                                        0
                                        0
                                        0
                        ;ERROR 27 - MODULE INTERRUPT ENABLE TOGGLE DID NOT CAUSE INTERRUPT
                                        EM27
                                        0
                                        0
                                        0
                        ;ERROR 30 - ERROR INTERRUPT ENABLE TOGGLE DID NOT CAUSE INTERRUPT
                                        EM30
                                        0
                                        0
                                        0
                        ;ERROR 31 - BMT INTERRUPT ENABLE TOGGLE DID NOT CAUSE INTERRUPT
                                        EM31
                                        0
                                        0
                                        0
                        ;ERROR 32 - OUTPUT NOT BUSY WITH MOV INSTRUVTION
                                        EM32
                                        0
                                        0
                                        0
                        ;ERROR 33 - DATA AVAILABLE NOT SET ON TTY TEST
                                        EM33
                                        0
                                        0
                                        0
                        ;ERROR 34 - MODULE INTERRUPT NOT POSTED
                                        EM34
                                        0
                                        0
                                        0
                        ;ERROR 35 - OUTPUT BUSY STUCK
                                        EM35
                                        0
```

```
 482  001526  000000                    0
 483  001530  000000                    0
 484
 485                                ;ERROR 36 - TTY INTERUPT SEND RECIEVE ERROR
 486
 487  001532  023466                    EM36
 488  001534  025416                    DH1
 489  001536  025620                    DT1
 490  001540  000000                    0
 491
 492                                ;ERROR 37 - FORCED MASTER ERROR FAILURE
 493
 494  001542  023527                    EM37
 495  001544  000000                    0
 496  001546  000000                    0
 497  001550  000000                    0
 498
 499                                ;ERROR 40 - FORCED MASTER ERROR DID NOT CAUSE TIMEOUT
 500
 501  001552  023541                    EM40
 502  001554  000000                    0
 503  001556  000000                    0
 504  001560  000000                    0
 505
 506                                ;ERROR 41 - SLAVE RESPONDED ON FORCED MASTER ERROR
 507
 508  001562  023560                    EM41
 509  001564  000000                    0
 510  001566  000000                    0
 511  001570  000000                    0
 512
 513                                ;ERROR 42 - FORCED SLAVE ERROR FAILURE
 514
 515  001572  023601                    EM42
 516  001574  000000                    0
 517  001576  000000                    0
 518  001600  000000                    0
 519
 520                                ;ERROR 43 - FORCED SLAVE CRC ERROR CAUSED TIMEOUT
 521
 522  001602  023613                    EM43
 523  001604  000000                    0
 524  001606  000000                    0
 525  001610  000000                    0
 526
 527                                ;ERROR 44 - FORCED SLAVE CRC ERROR DID NOT CAUSE ERROR
 528
 529  001612  023627                    EM44
 530  001614  000000                    0
 531  001616  000000                    0
 532  001620  000000                    0
 533
 534                                ;ERROR 45 - TIMEOUT ERROR IN M8098 LOOP TEST
 535
 536  001622  023644                    EM45
 537  001624  000000                    0
```

```
000000                          0
000000                          0

                        ;ERROR 46 - DATA VALID WAS FALSE IN M8098 LOOP TEST

023662                          EM46
000000                          0
000000                          0
000000                          0

                        ;ERROR 47 - ICAR IN LOOP TEST WAS BAD

023703                          EM47
025532                          DM3
025642                          DT3
000000                          0

                        ;ERROR 50 - BAD DATA IN MODULE ADDRESS

023716                          EM50
025434                          DM2
025630                          DT2
000000                          0

                        ;ERROR 51 - TEN CONSECUTIVE LINE ERRORS IN INPUT MODULE RANGE TEST

023735                          EM51
000000                          0
000000                          0
000000                          0

                        ;ERROR 52 - TEN CONSECUTIVE LINE ERRORS IN MODULE WRAP AROUND TEST

024016                          EM52
000000                          0
000000                          0
000000                          0

                        ;ERROR 53 - TBMT WAS TRUE AFTER TTY TRANSMISSION

024105                          EM53
000000                          0
000000                          0
000000                          0

                        ;ERROR 54 - TBMT WAS FALSE AFTER TTY TRANSMISSION TIME

024124                          EM54
000000                          0
000000                          0
000000                          0

                        ;ERROR 55 - RIF BIT NOT SET

024142                          EM55
000000                          0
```

```
001726  000000                  0
001730  000000                  0

                        ;ERROR 56 - RIF BIT NOT CLEAR WITH MODULE

001732  024162                  EM56
001734  000000                  0
001736  000000                  0
001740  000000                  0

                        ;ERROR 57 - TTY INTERRUPT TEST HUNG

001742  024176                  EM57
001744  000000                  0
001746  000000                  0
001750  000000                  0

                        ;ERROR 60 - MAINTENANCE BIT 3 STUCK ON AFTER ISSUE

001752  024226                  EM60
001754  000000                  0
001756  000000                  0
001760  000000                  0

                        ;ERROR 61 - POWER FAIL BIT IS ON

001762  024245                  EM61
001764  000000                  0
001766  000000                  0
001770  000000                  0

                        ;ERROR 62 - ICR11 CSR IN ERROR

001772  024265                  EM62
001774  025416                  DM1
001776  025620                  DT1
002000  000000                  0
                        ;ERROR 63 - UNEXPECTED MODULE INTERRUPT FROM ICR DURING REMOTE TTY TEST

002002  024336                  EM63
002004  000000                  0
002006  000000                  0
002010  000000                  0

                        ;ERROR 64 - SPURIOUS INTERRUPT IN REMOTE TTY TEST

002012  024316                  EM64
002014  000000                  0
002016  000000                  0
002020  000000                  0


                        ;R4 IS USED THROUGHOUT THIS TO KEEP TRACK OF WHICH FILE BOX IS UNDER
                        ;TEST, ALL PASS COUNTS, ERRORS AND LINE ERRORS ARE BUFFERED FOR EACH
```

```
                                ;ICR FILE BOX THAT IS BEING TESTED
                                
                                
                                .SBTTL  ICSR/ICAR BIT EQUIVALENTS
                                ;
            100000              OUTBSY=BIT15                ;ICSR - OUTPUT BUSY
            040000              MAINT3=BIT14                ;ICSR - MAINT BIT 03
            020000              MAINT2=BIT13                ;ICSR - MAINT BIT 02
            010000              ERRBIT=BIT12                ;ICSR - ERROR INTERRUPT
            004000              MAINT1=BIT11                ;ICSR - MAINT BIT 01
            002000              PWRFL=BIT10                 ;ICSR - POWER FAIL INTERRUPT
            001000              TBMTEN=BIT9                 ;ICSR - TBMT INTERRUPT ENABLE
            000400              MAINT0=BIT8                 ;ICSR - MAINT BIT 00
            000200              MODINT=BIT7                 ;ICSR - MODULE INTERRUPT
            000100              XRESET=BIT6                 ;ICSR - RESET
            000040              TTYEN=BIT5                  ;ICSR - TTY ENABLE
            000020              PWFEN=BIT4                  ;ICSR - POWER FAIL INTERRUTP ENA.
            000010              BMTEN=BIT3                  ;ICSR - BMT INTERRUPT ENABLE
            000004              MODEN=BIT2                  ;ICSR - MODULE INTERRUPT ENABLE
            000002              ERREN=BIT1                  ;ICSR - ERROR INTERRUPT ENABLE
            000001              XRIF=BIT0                   ;ICSR - RESET INTERRUPT FLAG
            022626              POP2SP=022626               ;POP STACK
            000000              PASCNT=0                    ;PASS COUNT OFFSET TO R4
            000002              ERRTOT=2                    ;ERROR COUNT OFFSET TO R4
            000004              ERRCNT=4                    ;LINE ERROR COUNT OFFSET TO R4
            000006              PERHGH=6                    ;HIGH INPUT MODULE ADDRESS RANGE
            000010              PERLOW=10                   ;LOW INPUT MODULE ADDRESS RANGE
                                
                                ;
                                ;
            100000              XTBMT=BIT15                 ;ICAR - TBMT
            010000              DA=BIT12                    ;ICAR - DA
                                ;
                                ;
                                ;
                                ;
002022  171000                  ICMOD:  171000              ;MODULE ADDRESS
002024  171776                  ICSR:   171776              ;ICR CSR
002026  171774                  ICAR:   171774              ;ICR ADDRESS REGISTER
002030  000000                  ICRRUN: 0                   ;ICR COUNT ON SYSTEM
002032  000000                  PWFFLG: 0                   ;PWR FAIL TEST INDICATOR
002034  000000                  ICRCNT: 0                   ;ICR COUNT
002036  026502                  MFR11K: 11586.              ;CONSTANT FOR 110 MSEC
002040  000474                  MFR1MK: 316.                ;1 MSEC
002042  000236                  MFR500: 158.                ;500 USEC
002044  000077                  MFR200: 63.                 ;200 USEC
002046  000037                  MFR100: 31.                 ;100 USEC
002050  000006                  MFR20:  6.                  ;20 USEC
002052  000000                  BOXCNT: 0
002054  000000                  TENCNT: 0
002056  000000                  SW9FF:  0                   ;SW9 PREVIOUSLY SET
002060  000000                  SRTFF:  0                   ;PASS RUNNING FLOP
002062  000000                  ERRFLR: 0
                                
                                ;START 224 BRINGS YOU HERE FOR TEST OF POWER FAIL
                                
```

```
                                                SETPWF: INC     PWFFLG          ;POWER FAIL TEST REQUIRED

                                                        .SBTTL  ADDRESS MAP OF ICR


                                                START:
002070                                                  MOV     #SCMTAG,R0      ;;FIRST LOCATION TO BE CLEARED
002070  012700  001100                                  CLR     (R0)+           ;;CLEAR MEMORY LOCATION
002074  005020                                          CMP     #SBUDAT,R0      ;;DONE?
002076  022700  001126                                  BNE     .-6             ;;LOOP BACK IF NO
002102  001374                                          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
002104  012700  001100                                  MOV     #SERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002110  012737  015056  000030                          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002116  012737  000340  000032                          MOV     #STRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002124  012737  016154  000034                          MOV     #340,@#TRAPVEC+2;LEVEL 7
002132  012737  000340  000036                          MOV     #SPWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
002140  012737  016020  000024                          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002146  012737  000340  000026                          MOV     @#4,-(SP)       ;;SAVE ERROR VECTOR
002154  013746  000004                                  MOV     @#6,-(SP)
002160  013746  000006                                  MOV     #648,4          ;;SET UP TIME OUT VECTOR
002164  012737  002200  000004                          TST     @#SWR           ;;TRY TO REFERENCE HARDWARE SWR
002172  005777  176740                                  BR      65$             ;;BRANCH IF NO TIMEOUT TRAP OCCURS
002176  000407                                  001136  MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
002200  012737  000176  001136  64$:                    MOV     #DISPREG,DISPLAY ;;POINT TO SOFTWARE DISPLAY REG
002206  012737  000174  001140                          CMP     (SP)+,(SP)+     ;;RESTORE STACK
002214  022626                                  000006  MOV     (SP)+,@#6       ;;RESTORE ERROR VECTOR
002216  012637  000006          65$:                    MOV     (SP)+,@#4
002222  012637  000004

                                                ;DETERMINE WHAT PDP11 WE ARE RUNNING ON FOR TIMING CONSTANTS

002226  012737  000006  000004  FIND11: MOV     #6,@#4
002234  012737  000002  000012                          MOV     #RTI,@#12
002242  000262                                          SEV
002244  074000                                          XOR     R0,R0
002246  102512                                          BVS     2$
002250  013700  002050                                  MOV     MFR20,R0                ;MUST BE 11/40,11/45 RECAL CONSTANTS
002254  010037  014416                                  MOV     R0,FR20
002260  060037  014416                                  ADD     R0,FR20
002264  060037  014416                                  ADD     R0,FR20
002270  013700  002046                                  MOV     MFR100,R0
002274  010037  014412                                  MOV     R0,FR100
002300  060037  014412                                  ADD     R0,FR100
002304  060037  014412                                  ADD     R0,FR100
002310  013700  002044                                  MOV     MFR200,R0
002314  010037  014414                                  MOV     R0,FR200
002320  060037  014414                                  ADD     R0,FR200
002324  060037  014414                                  ADD     R0,FR200
002330  013700  002042                                  MOV     MFR500,R0
002334  010037  014410                                  MOV     R0,FR500
002340  060037  014410                                  ADD     R0,FR500
002344  060037  014410                                  ADD     R0,FR500
002350  013700  002040                                  MOV     MFR1K,R0
002354  010037  014406                                  MOV     R0,FR1000
002360  060037  014406                                  ADD     R0,FR1000
002364  060037  014406                                  ADD     R0,FR1000
002370  013700  002036                                  MOV     MFR11K,R0
```

```
   762  002374  010037  014420                MOV     R0,FRM110
   763  002400  060037  014420                ADD     R0,FRM110
   764  002404  060037  014420                ADD     R0,FRM110
   765
   766                                  ;NOW SEE IF RUNNING ON 11/70, IF SO CHANGE CONSTANTS BY 50%
   767
   768  002410  013746  000004                MOV     @#4,-(SP)       ;SAVE TIMEOUT VECTOR
   769  002414  013746  000006                MOV     @#6,-(SP)       ;SAVE
   770  002420  012737  002460  000004        MOV     #3$,@#4         ;SETUP IF NOT 11/70
   771  002426  005737  177760                TST     @#177760        ;TRY TO ADDRESS 11/70 MEM SIZE REG
   772  002432  006337  014416                ASL     FR20            ;SUCCESSFUL, SHIFT ALL CONSTANTS BY TWO
   773  002436  006337  014412                ASL     FR100
   774  002442  006337  014414                ASL     FR200
   775  002446  006337  014410                ASL     FR500
   776  002452  006337  014406                ASL     FR1000
   777  002456  000401                        BR      .+4
   778  002460  022626                3$:     POP2SP                  ;RESET STACK
   779  002462  012637  000006                MOV     (SP)+,@#6       ;RESTORE 6
   780  002466  012637  000004                MOV     (SP)+,@#4       ;RESTORE 4
   781  002472  000417                        BR      1$
   782  002474  013737  002050  014416  2$:   MOV     MFR20,FR20      ;11/05  11/20 CONSTANTS
   783  002502  013737  002044  014414        MOV     MFR200,FR200
   784  002510  013737  002046  014412        MOV     MFR100,FR100
   785  002516  013737  002040  014406        MOV     MFR1K,FR1000
   786  002524  013737  002036  014420        MOV     MFR11K,FRM110
   787  002532  005037  000012          1$:   CLR     @#12
   788  002536  005037  014464                CLR     PRRMT1
   789  002542  005037  014462                CLR     PRRMT           ;CLEAR PRINT TO REMOTE END
   790  002546  104413  022656                TYPE,HEADER
   791  002552  012700  171774                MOV     #171774,R0      ;START OF ICSR,ICAR
   792  002556  005037  002034                CLR     ICRCNT          ;CLEAR ICR COUNT
   793  002562  005002                        CLR     R2
   794  002564  005003                        CLR     R3              ;SET TABLE
   795  002566  012737  002710  000004        MOV     #FILNFD,@#4     ;SET TIMEOUT VECTOR
   796  002574  005010                FILFND: CLR     (R0)            ;TRY TO ADDRESS FILE
   797                                  ;
   798                                  ;THERE'S SOMETHING OUT THERE, DETERMINE IF IT'S ICS OR
   799                                  ;ICR
   800                                  ;
   801  002576  010037  002024                MOV     R0,ICSR
   802  002602  062737  000002  002024        ADD     #2,ICSR         ;CREATE ICSR
   803  002610  052777  000040  177206        BIS     #40,@ICSR       ;TRY TO SET BIT 5
   804  002616  032777  000040  177200        BIT     #40,@ICSR       ;BIT 5 SET?
   805  002624  001004                        BNE     5$              ;1
   806  002626  032777  020000  177170        BIT     #20000,@ICSR    ;IT DIDN'T WRITE BUT IT
   807  002634  001026                        BNE     FILNFD+2        ;MAY BE BAD; CHECK BIT 13
   808  002636  005237  002034          5$:   INC     ICRCNT          ;INDICATE SUCCESS
   809  002642  104413  022547                TYPE,FILE
   810  002646  010246                        MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
   811                                                                  ;;FILE BOX
   812  002650  104402                        TYPOS                   ;;GO TYPE--OCTAL ASCII
   813  002652     002                        .BYTE   2               ;;TYPE 2 DIGIT(S)
   814  002653     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
   815                                                                  ;ICAR
   816  002654  104413  022563                TYPE,ICARMD
   817  002660  010046                        MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
```

```
                                TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                                TYPE,ICSRMD
                                                        ;ICSR
                                ADD     #2,R0
                                MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                                SUB     #2,R0
                                                        ;ADDRESS
                                INC     R3              ;INDICATE SUCCESS IN LIST
                                BR      .+4             ;
                        FILNFD: POP2SP                  ;POP STACK
                        1$:     ASL     R3              ;INDICATE NO ICR
                                SUB     #10,R0          ;UPDATE FOR NEW ICAR
                                INC     R2              ;INCREMENT FILE #
                                CMP     #14,R2          ;DONE?
                                BNE     FILFND          ;NO
                                MOV     R3,SYSMAP       ;STORE FILE LIST
                                TYPE,   STANMS
                                MOV     #6,@#4          ;MOVE .+2 BACK
                                TST     ICRCNT          ;ANY FILES FOUND
                                BNE     FILLST+2
                        GOOF:   TST     BOXCNT
                                BNE     CONTST
                                TYPE,NOBOX              ;BOX ASKED TO TEST DOES NOT EXIST
                                HALT                    ;HALT
                                JMP     .-2             ;
                        FILLST: 0                       ;LIST FOR FILE INDICATION
                        CONTST:
                                TYPE,CRLF

                                .SBTTL  INITIAL SETUP FOR PROGRAM


                        START1: MOV     #STACK,SP       ;SET STACK POINTER
                                CLR     SW9FF           ;CLEAR THAT SW9 IS SET
                                BIT     #SW9,@#SWR      ;TEST IF SW 9 IS SET
                                BEQ     .+6
                                INC     SW9FF           ;INDICTAE PRESENCE OF SW 9
                        ;
                        ;INITIAL CLEAR OF ALL FILE BOX(ES) INFORMATION I.E. PASS COUNTS, ERROR COUNTS
                        ;LINE ERROR COUNTS

                                MOV     #INFLST,R0      ;BEGINNING OF BUFFER
                                MOV     #70.,R1         ;BUFFER LENGTH
                                CLR     (0)+            ;CLEAR LOCATION IN BUFFER
                                DEC     R1              ;DEC BUFFER LENGTH
                                BNE     .-4             ;CONTINUE TILL DONE


                        START4: CLR     TIMTRP          ;CLEAR TIMEOUT INDICATOR
```

```
003050  005037  002060                  CLR     SWTFF           ;INDICATE FIRST BOX FOR PASS SUMMARY
003054  005037  017204                  CLR     WRKFLP          ;PRINT REMOTELY "TESTING"
003060  005037  014474                  CLR     PWRFLP          ;POWER FAIL INDICATOR
003064  005037  002030                  CLR     ICRRUN
003070  012737  020000  014376          MOV     #20000,XMASK    ;SET UP MASK
003076  013737  014436  002774          MOV     SYSMAP,FILLST   ;GET SYSTEM CONFIG.
003104  005037  014400                  CLR     FILUTS          ;FILE BOX UNDER TEST
003110  012704  017210                  MOV     #INFLST,R4      ;INITIAL ADDRESS OF INFORMATION BUFF
003114  012737  000015  014402          MOV     #15,FILCNT      ;SET MAXIMUM FILE TO 15(BASE 8)
003122  012737  171774  002026          MOV     #171774,ICAR    ;INITIAL ICAR
003130  012737  171776  002024          MOV     #171776,ICSR    ;INITIAL ICSR
003136  012737  171000  002022          MOV     #171000,ICMOD   ;INITIAL MODULE ADDRESS

003144  005037  002052          START2: CLR     BOXCNT          ;CLEAR THAT BOXES EXIST
003150  005037  014426                  CLR     MICFLP          ;RUNNING MICRODIAGNOSTIC INDICATOR
003154  012737  000001  014454          MOV     #1,PASLG        ;SET FOR QUICK VERIFY
003162  032777  010000  175746          BIT     #SW12,@SWR      ;LONG PASS WANTED
003170  001403                          BEQ     2$              ;NO, THEN 2$
003172  062737  000143  014454          ADD     #99.,PASLG      ;YES THEN SET LONG PASS
                                        ;
                                        ;WRITE LOCATIONS "230-1000" WITH ",,+2!4", PC+2, IOT TRAP
                                        ;SINCE EACH PASS WILL OVER WRITE SOME LOCATIONS
                                        ;
003200  012700  000230          2$:     MOV     #230,R0
003204  012701  000232                  MOV     #232,R1
003210  010120                  1$:     MOV     R1,(R0)+
003212  012720  000004                  MOV     #4,(R0)+
003216  022121                          CMP     (R1)+,(R1)+
003220  022700  001000                  CMP     #1000,R0
003224  003371                          BGT     1$
003226  012737  177777  014424          MOV     #-1,FIRST       ;SET TO PRINT FILE BOX ID ONLY ON FIRST ERROR
003234  012737  000340  177776          MOV     #340,@#PSW

003242  032777  000020  175666  ONEBOX: BIT     #SW4,@SWR       ;SINGLE BOX OR ALL TO BE TESTED?
003250  001430                          BEQ     TEST1           ;ALL THEN TEST1
003252  017700  175660                  MOV     @SWR,R0         ;GET BOX TO BE TESTED
003256  012701  010000                  MOV     #10000,R1       ;SET MASK TO FIRST BOX
003262  042700  177760                  BIC     #177760,R0      ;CLEAR ALL BUT BOX SWITHES
003266  001412                          BEQ     1$              ;FILE BOX 0, GO TEST
003270  020027  000015                  CMP     R0,#15          ;IS BOX LEGAL
003274  002404                          BLT     2$              ;YES, CONT AT 2$
003276  104413  024533                  TYPE,   NOEXST          ;BOX ASKED TO TEST DOES NOT EXIST
003302  000000                          HALT
003304  000776                          BR      .-2
003306  006201                  2$:     ASR     R1              ;SHIFT MASK RIGHT TIL BOX FOUND
003310  005300                          DEC     R0              ;HAS BOX BEEN FOUND YET
003312  001375                          BNE     2$              ;NO, THEN KEEP LOOKING
003314  005037  002774          1$:     CLR     FILLST          ;CLEAR SYSTEM CONFIGURATION SO AS TO
003320  050137  002774                  BIS     R1,FILLST       ;TO TEST ONLY BOX ON SYSTEM
003324  012737  000001  002034          MOV     #1,ICRCNT       ;SET ICRCNT TO ONE
```

```
003332 012737 000340 177776 TEST11: MOV     #340,@#PSW      ;INHIBIT INTERRUPTS
003340 006237 014376        2$:     ASR     XMASK           ;ROTATE RIGHT TO FIND FILE BOX
003344 033737 014376 002774         BIT     XMASK,FILLST    ;FILE PRESENT
003352 001022                       BNE     1$              ;FILE PRESENT?
003354 162737 000010 002024         SUB     #10,ICSR        ;UPDATE ICSR
003362 162737 000010 002026         SUB     #10,ICAR        ;UPDATE ICAR
003370 062737 000040 002022         ADD     #40,ICMOD       ;UPDATE FILE BOX ADDRESS
003376 062704 000012                ADD     #12,R4          ;MOVE INFORMATION POINTER
003402 005237 014400                INC     FILUTS          ;FILE UNDER TEST
003406 005337 014402                DEC     FILCNT          ;RUN OUT
003412 001352                       BNE     2$
003414 000137 002754                JMP     GOOF
003420 000240               1$:     NOP
                            ;
                            ;FILE BOX,ICAR,ICSR FOUND - START TEST
                            ;
003422 005764 000000                TST     PASCNT(4)       ;DOING FIRST PASS FOR THIS FILE
003426 001402                       BEQ     .+6             ;PRINT "TESTING....."
003430 000137 004014                JMP     VECTST          ;NO, THEN JUST GET VECTOR
003434 012737 000001 002052         MOV     #1,BOXCNT       ;SET THAT BOX PRESENT
003442 005037 014462                CLR     PRRMT
003446 012737 000001 014464         MOV     #1,PRRMT1       ;PRINT TO REMOTE END
003454 104413 022630                TYPE,TESTIN             ;PRINT "TESTING...."
003460 013700 014400                MOV     FILUTS,R0
003464 010046                       MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
                                                            ;;FILE BOX
003466 104402                       TYPOS                   ;;GO TYPE--OCTAL ASCII
003470    002                       .BYTE   2               ;;TYPE 2 DIGIT(S)
003471    000                       .BYTE   0               ;;SUPPRESS LEADING ZEROS
003472 005000                       CLR     R0
003474 005300                       DEC     R0
003476 001376                       BNE     .-2




003500 000137 004014                JMP     VECTST


                                    .SBTTL  TEST OF CSR

003504 012737 017775 017774 LPTST:  MOV     #BUFBEG-1,BUFMES ;RESET BUFFER FOR MESSAGES
003512 012737 000001 014464         MOV     #1,PRRMT1       ;ALLOW REMOTE PRINTING
003520 005037 014462                CLR     PRRMT

                            ;TEST OF POWER FAIL BIT

003524 012737 003524 014372 RWBIT:  MOV     #RWBIT,SCOLOP

003532 032777 002000 176264         BIT     #PWRFL,@ICSR    ;IS PWR FAIL ON
003540 001401                       BEQ     1$

                            ;
                            ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                            ;
```

L4

```
  986  003542  104061                        ERROR   61              ;PWR FALL BIT IN CSR IS SET
  987
  988                                  ;
  989                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
  990                                  ;
  991
  992  003544  104412                  181     SCOPEX
  993
  994                                  ;TEST THAT MAINT BIT 3 CLEARS ITSELF OUT
  995
  996  003546  012737  003546  014372  381     MOV     #38,SCOLOP
  997  003554  052777  040000  176242          BIS     #MAINT3,@ICSR
  998  003562  032777  040000  176234          BIT     #MAINT3,@ICSR
  999
 1000  003570  001401                          BEQ     28
 1001
 1002                                 ;
 1003                                 ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
 1004                                 ;
 1005
 1006  003572  104060                        ERROR   60              ;MAINT3 BIT OF CSR SHOULD CLEAR CSR AND ITSELF OUT
 1007                                                                 ;WHEN ISSUED
 1008
 1009                                 ;
 1010                                 ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
 1011                                 ;
 1012
 1013  003574  104412                  281     SCOPEX
 1014
 1015
 1016  003576  005037  001124          MAINT3: CLR     $GOODAT         ;SET EXPECTED
 1017  003602  012737  003576  014372          MOV     #MAINT3,SCOLOP  ;SET SCOPE
 1018  003610  013700  014406                  MOV     @R1000,R0
 1019  003614  005777  176204          788:    TST     @ICSR           ;IS OUTPUT BUSY?
 1020  003620  100003                          BPL     .+10            ;NO, CONT
 1021  003622  005300                          DEC     R0              ;YES, INC TIMEOUT
 1022  003624  001373                          BNE     788
 1023
 1024  003626  104035                        ERROR   35              ;OUTPUT BUSY
 1025
 1026  003630  017700  176172                  MOV     @ICAR,R0
 1027
 1028  003634  052777  040000  176162          BIS     #MAINT3,@ICSR   ;ISSUE MAINT RESET
 1029  003642  017737  176156  001126          MOV     @ICSR,$GOODAT   ;SEE IF CSR IS CLEAR AFTER RESET (MAINT3)
 1030  003650  001401                          BEQ     18
 1031
 1032                                 ;
 1033                                 ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
 1034                                 ;
 1035
 1036  003652  104062                        ERROR 62                ;MAINT RESET DID NOT CLEAR ICR CSR
 1037
 1038                                 ;
 1039                                 ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
 1040                                 ;
 1041  003654  104412                  181     SCOPEX
```

M14

```
                                ;TEST THAT ALL ENABLES WILL SET AND CLEAR ICSR BITS
                                ;INVOLVED  ARE BITS 0,1,2,3,4,5,9
                                ;
003656  012700  003776          BITTST: MOV     #BITLST,R0      ;GET LIST
003662  012701  000007                  MOV     #7,R1           ;SET LENGTH
003666  012002                  4$:     MOV     (0)+,R2         ;GET BIT
003670  012737  003670  014372  1$:     MOV     #1$,SCOLOP      ;SET SCOPE LOOP
003676  005077  176122                  CLR     @ICSR           ;CLEAR CSR
003702  005037  001124                  CLR     $GDDAT          ;CLEAR EXPECTED
003706  005037  001126                  CLR     $BDDAT
003712  050237  001124                  BIS     R2,$GDDAT       ;SET EXPECTED
003716  050277  176102                  BIS     R2,@ICSR        ;SET BIT IN CSR
003722  030277  176076                  BIT     R2,@ICSR
003726  001003                          BNE     2$
003730  040237  001126                  BIC     R2,$BDDAT
                                ;
                                ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                ;
003734  104062                          ERROR 62                ;BIT WAS NOT SET IN ICR CSR
                                ;
                                ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                ;
003736  104412                  2$:     SCOPEX
                                ;
003740  040237  001124                  BIC     R2,$GDDAT       ;CLEAR BIT
003744  040277  176054                  BIC     R2,@ICSR
003750  030277  176050                  BIT     R2,@ICSR        ;TEST IF BIT IS CLEAR
003754  001403                          BEQ     3$
003756  050237  001126                  BIS     R2,$BDDAT
                                ;
                                ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                ;
003762  104062                          ERROR 62                ;BIT WAS NOT CLEAR IN ICR CSR
                                ;
                                ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                ;
003764  104412                  3$:     SCOPEX
                                ;
003766  005301                          DEC     R1              ;DONE TESTING
003770  001336                          BNE     4$
                                ;
003772  000137  004346                  JMP     TENTST
```

```
1098
1099  003776  000001                  BITLST: XRIF
1100  004000  000002                          ERREN
1101  004002  000004                          MODEN
1102  004004  000010                          BMTEN
1103  004006  000020                          PWFEN
1104  004010  000040                          TTYEN
1105  004012  001000                          TBMTEN
1106
1107
1108                                  .SBTTL  ICR VECTOR ADDRESS AND PRIORITY LEVEL
1109
1110                                  ;
1111                                  ;VECTOR AND PRIORITY LEVEL
1112                                  ;
1113
1114          004014                  VECTST=.
1115
1116  004014  005237  014426                  INC     #ICFLP
1117  004020  005037  004146                  CLR     CONTSW
1118  004024  012737  004150  000020  VECT1:  MOV     #VECFND,@#20    ;MOVE VECTOR ROUTINE TO IOT TRAP
1119  004032  012737  000340  177776          MOV     #340,@#PSW      ;PROCESS STATUS WORD - PRIOR = 7
1120  004040  012703  000007                  MOV     #7,R3           ;PRIORITY LEVEL IN R5
1121  004044  013700  014410          1$:     MOV     FR500,R0
1122  004050  052777  000004  175746          BIS     #MODEN,@ICSR    ;MODULE INTERRUPT BIT
1123  004056  052777  020000  175740          BIS     #MAINT2,@ICSR   ;MAINTENANCE BIT 0
1124  004064  005300                          DEC     R0
1125  004066  001376                          BNE     .-2
1126  004070  052777  040000  175726          BIS     #MAINT3,@ICSR   ;NO INTERRUPT - RESET
1127  004076  005303                          DEC     R3              ;DECREMENT PRIORITY LEVEL
1128  004100  100404                          BMI     VECT2           ;
1129  004102  162737  000040  177776          SUB     #40,@#PSW
1130  004110  000755                          BR      1$              ;LOOP - LOWER PRIORITY
1131  004112  005737  004146          VECT2:  TST     CONTSW
1132  004116  001402                          BEQ     .+6
1133  004120  000137  004024                  JMP     VECT1
1134  004124  104413  022072                  TYPE,VECMES             ;TYPE THAT ICR WILL NOT INTERRUPT
1135  004130  000000                          HALT                    ;HALT, CONT WILL FORCE SCOPE LOOP
1136  004132  104413  024775                  TYPE,   CONTOP
1137  004136  005237  004146                  INC     CONTSW          ;SET CONTSW
1138  004142  000137  004024                  JMP     VECT1
1139
1140  004146  000000          CONTSW: 0
1141
1142                                  ;
1143                                  ;VECTOR AND PRIORITY LEVEL FOUND, CONTINUE TESTING
1144                                  ;
1145                                  ;
1146                                  ;
1147  004150  052777  040000  175646  VECFND: BIS     #MAINT3,@ICSR
1148  004156  010337  017172                  MOV     R3,PRILVL       ;SAVE PRIORITY LEVEL FOUND
1149  004162  005203                          INC     R3              ;BUMP UP TO FORM BR LEVEL
1150  004164  011605                          MOV     (SP),R5         ;GET VECTOR ADDRESS
1151  004166  162705  000004                  SUB     #4,R5
1152  004172  010537  014404                  MOV     R5,VECTOR
1153  004176  022626                          POP2SP                  ;POP STACK
```

```
1154  004200  022626                          POP2SP
1155  004202  005764  000000                  TST     PASCNT(4)       ;DOING FIRST PAS FOR THIS FILE
1156  004206  001013                          BNE     1$              ;NO, CONT DON'T PRINT INFORMATION
1157  004210  104413  022004                  TYPE,   VECMD           ;PRINT VECTOR
1158  004214  010546                          MOV     R5,-(SP)        ;;SAVE R5 FOR TYPEOUT
1159  004216  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1160  004220  104413  022034                  TYPE,PRIOR              ;PRIORITY
1161  004224  010346                          MOV     R3,-(SP)        ;;SAVE R3 FOR TYPEOUT
1162  004226  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1163  004230  005001                          CLR     R1
1164  004232  005301                          DEC     R1
1165  004234  001376                          BNE     .-2
1166  004236  012737  000340  177776  1$:     MOV     #340,@#PSW      ;PRIORITY LEVEL = 7
1167
1168  004244  012700  000230          VECRET: MOV     #230,R0         ;WRITE OVER WITH .+2, HALT
1169  004250  012701  000232                  MOV     #232,R1
1170  004254  010120                  9$:     MOV     R1,(R0)+
1171  004256  012720  000000                  MOV     #0,(R0)+
1172  004262  022121                          CMP     (R1)+,(R1)+
1173  004264  022700  001000                  CMP     #1000,R0
1174  004270  003371                          BGT     9$
1175
1176
1177  004272  012777  016714  010104          MOV     #ICRSRV,@VECTOR ;MOVE ICR SERVICE ROUTINE
1178  004300  013700  014404                  MOV     VECTOR,R0       ;MOV
1179  004304  005720                          TST     (R0)+
1180  004306  012710  000340                  MOV     #340,(R0)       ;MOV 340 IN PSW PICKUP
1181  004312  006337  017172                  ASL     PRILVL
1182  004316  006337  017172                  ASL     PRILVL
1183  004322  006337  017172                  ASL     PRILVL
1184  004326  006337  017172                  ASL     PRILVL
1185  004332  006337  017172                  ASL     PRILVL          ;ALIGN PRIORITY LEVEL
1186
1187
1188
1189  004336  005037  014472                  CLR     REMOTE
1190  004342  000137  003504                  JMP     LPTST
1191
1192                                  ;
1193                                  ;TEST THAT TTY ENABLE SETS AND CLEARS
1194                                  ;
1195
1196  004346                          TENTST:
1197  004346  012700  017776                  MOV     #BUFBEG,R0      ;CLEAR OUT ERROR STORAGE BUFFER
1198  004352  012720  100601          9$:     MOV     #100601,(0)+
1199  004356  020027  021750                  CMP     R0,#BUFEND      ;DONE WITH BUFFER
1200  004362  001373                          BNE     9$
1201  004364  005037  014462                  CLR     PRRMT           ;DON'T BUFFER
1202  004370  012737  000001  014464          MOV     #1,PRRMT1       ;SET PRINT TO REMOTE END
1203  004376  005237  014426                  INC     MICFLP
1204  004402  013737  014454  014456          MOV     PASLG,PASRUN
1205          004410                  LOOP1=.
1206  004410  012737  004410  014372  7$:     MOV     #7$,$COLOP
1207  004416  052777  000040  175400          BIS     #TTYEN,@ICSR    ;SET TTY ENABLE
1208  004424  032777  000040  175372          BIT     #TTYEN,@ICSR    ;IS IT SET
1209  004432  001001                          BNE     4$
```

```
1210
```

```
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
004434  104003                                ERROR   3               ;TTY ENABLE NOT SET

                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;

004436  104412                          48$:    SCOPEX

004440  042777  000040  175356                BIC     #TTYEN,@ICSR    ;CLEAR TTY ENABLE
004446  032777  000040  175350                BIT     #TTYEN,@ICSR    ;IS IT CLEAR
004454  001401                                BEQ     5$

                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
004456  104004                                ERROR 4                 ;TTY ENABLE NOT CLEAR

                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;

004460  104412                          58$:    SCOPEX

                                        ;
                                        ;TEST THAT OUTPUT BUSY IS CLEAR
                                        ;
004462  012737  004462  014372          68$:    MOV     #68,SCOLOP

004470  005777  175330                        TST     @ICSR           ;CHECK BIT 15, OUTPUT BUSY
004474  100001                                BPL     1$

                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
004476  104001                                ERROR 1                 ;OUTPUT BUSY IS SET WITH NO PRIOR TRANSMISSION

                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;

004500  104412                          18$:    SCOPEX

004502  052777  040000  175314                BIS     #MAINT3,@ICSR

                                        ;
                                        ;CHECK THAT RIF BIT WILL SET
                                        ;
004510  012737  004510  014372          738$:   MOV     #738,SCOLOP
004516  052777  000001  175300                BIS     #XRIF,@ICSR
004524  032777  000001  175272                BIT     #XRIF,@ICSR
004532  001001                                BNE     71$

                                        ;
```

```
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
004534  104055                                ERROR   55              ;RIF BIT NOT SET

                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
004536  104412                          718:    SCOPEX

                                        ;
                                        ;CHECK THAT RIF WILL CLEAR WITH ADDRESSING MODULE
                                        ;
004540  012737  004540  014372          748:    MOV     #748,SCOLOP
004546  052777  000001  175250                  BIS     #XRIF,@ICSR
004554  017700  175242                          MOV     @ICMOD,R0
004560  032777  000001  175236                  BIT     #XRIF,@ICSR
004566  001401                                  BEQ     728

                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
004570  104056                                ERROR   56              ;RIF WOULD NOT CLEAR

                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
004572  104412                          728:    SCOPEX


                                        ;
                                        ;CHECK THAT BMT WILL INTERRUPT WHEN ENABLED
                                        ;
004574  104415                          BMTTST: CKCNC
004576  012737  004576  014372          108:    MOV     #108,SCOLOP     ;SET SCOPE LOOP
004604  005037  017206                          CLR     INTFLG          ;CLEAR INTERRUPT OCCURANCE FLAG
004610  013700  014414                          MOV     FR20B,R0
004614  005777  175284                  708:    TST     @ICSR           ;IS OUTPUT BUSY
004620  100003                                  BPL     .+10            ;IF NO CONT
004622  005300                                  DEC     R0              ;INC WATCHDOG
004624  001373                                  BNE     708             ;NOT TIMED OUT THEN 708

                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
004626  104035                                ERROR   35              ;OUTPUT BUSY IS STUCK ON

                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
004630  104412                                SCOPEX

004632  013700  014416                          MOV     FR20,R0
004636  005077  175162                          CLR     @ICSR           ;CLEAR ICR CSR
004642  052777  000010  175154                  BIS     #BMTEN,@ICSR    ;SET BMT INTERRUPT ENABLE
```

```
1323  004650  013737  017172  177776          MOV     PRILVL,@#PSW    ;LOWER PRIORITY LEVEL
1324  004656  005300                          DEC     R0
1325  004660  001376                          BNE     .-2
1326  004662  005737  017206                  TST     INTFLG          ;DID INTERRUPT OCCUR?
1327  004666  001001                          BNE     9$              ;YES, OKAY GO TO 9$
1328
1329                                  ;
1330                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1331                                  ;
1332  004670  104017                          ERROR   17              ; BMT WOULD NOT INTERRUPT
1333
1334                                  ;
1335                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1336
1337  004672  104412                  9$:     SCOPEX          ;SCOPE LOOP
1338
1339                                  ;
1340                                  ;CHECK THAT IF BMT INTERRUPT ENABLE IS RAISED LOWERED RAISED THAT
1341                                  ;BMT WILL INTERRUPT ONCE AGAIN
1342                                  ;
1343
1344  004674  042777  000010  175122  14$:     BIC     #BMTEN,@ICSR            ;CLEAR BMT ENABLE
1345  004702  052777  000001  175114          BIS     #XRIF,@ICSR             ;SET RIF BIT
1346  004710  005777  175106                  TST     @ICMOD                  ;CLEAR RIF BIT
1347  004714  005037  017206                  CLR     INTFLG                  ;CLEAR INT FLG
1348  004720  012737  004674  014372          MOV     #14$,$COLOP
1349  004726  005077  175072                  CLR     @ICSR                   ;CLEAR ICSR
1350  004732  052777  000010  175064          BIS     #BMTEN,@ICSR            ;SET BMT ENABLE
1351  004740  013737  017172  177776          MOV     PRILVL,@#PSW            ;LOWER PSW
1352  004746  013700  014416                  MOV     FR2@,R0
1353  004752  005300                          DEC     R0
1354  004754  001376                          BNE     .-2
1355  004756  012737  000340  177776          MOV     #340,@#PSW
1356  004764  005737  017206                  TST     INTFLG                  ;DID INTERRUPT OCCUR
1357  004770  001001                          BNE     15$                     ;YES OK,GO TO 15$
1358
1359                                  ;
1360                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1361                                  ;
1362  004772  104031                          ERROR 31                        ;BMT WOULD NOT INTERRUPT
1363
1364                                  ;
1365                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1366                                  ;
1367  004774  104412                  15$:    SCOPEX
1368
1369
1370  004776  012737  004776  014372  8$:     MOV     #8$,$COLOP
1371
1372  005004  032777  000100  175012          BIT     #XRESET,@ICSR   ;RESET IS READ AS "0"
1373  005012  001401                          BEQ     2$
1374
1375                                  ;
1376                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1377                                  ;
1378  005014  104002                          ERROR 2                 ;RESET READ AS "1"
```

```
1379
1380                                    ;
1381                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1382                                    ;
1383  005016  104412                    28$:    SCOPEX
1384                                    ;
1385                                    ;THE FOLLOWING SECTION IS DEPENDENT ON THE MICRODE FOR THE
1386                                    ;INTERRUPTS. ICR CSR BIT 13 (MAINT2) IS THE KEY BIT THAT THE
1387                                    ;MICROCODE FOLLOWS AS THE 11 CODE PROGRESSES
1388                                    ;
1389                                    ;
1390                                    ;TEST THAT BOUNCING MODULE INTERRUPT ENABLE WILL CAUSE
1391                                    ;ANOTHER INTERRUPT
1392                                    ;
1393
1394  005020  012737  000001  014464            MOV     #1,PRRMT1       ;ALLOW PRINT TO REMOTE END
1395  005026  012737  000001  014462            MOV     #1,PRRMT        ;BUFFER ERRORS THAT OCCUR
1396  005034  012737  005034  014372  INTST:    MOV     #INTST,SCOLOP   ;SET SCOPE LOOP
1397  005042  052777  040000  174754            BIS     #MAINT3,@ICSR   ;ISSUE MAINT RESET
1398  005050  005077  174750                    CLR     @ICSR           ;CLEAR ICR CSR
1399  005054  005037  017206                    CLR     INTFLG
1400  005060  013737  017172  177776            MOV     PRILVL,@#PSW    ;ALOW INTERRUPTS
1401  005066  052777  000004  174730            BIS     #MODEN,@ICSR    ;SET MOD INT ENABLE
1402  005074  052777  020000  174722            BIS     #MAINT2,@ICSR   ;SET MAINT BIT TO START MICROCODE
1403  005102  013700  014410                    MOV     FR5@@,R0
1404  005106  005300                            DEC     R0
1405  005110  001376                            BNE     .-2
1406  005112  005737  017206                    TST     INTFLG          ;INTERRUPT OCCUR?
1407  005116  001001                            BNE     18
1408
1409                                    ;
1410                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1411                                    ;
1412  005120  104021                            ERROR 21                ;MODULE INTERRUPT WOULD NOT INTERRUPT
1413                                                                    ;MODULE INTERRUPT SHOULD HAVE BEEN CAUSED
1414                                                                    ;BY THE MICROCODE
1415
1416                                    ;
1417                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1418                                    ;
1419
1420  005122  104412                    18$:    SCOPEX
1421
1422  005124  023727  017206  000001            CMP     INTFLG,#1       ;DID MULTIPLE INTERRUPTS OCCUR
1423  005132  001401                            BEQ     28
1424
1425                                    ;
1426                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1427                                    ;
1428  005134  104026                            ERROR 26                ;MODULE INTERRUPT OCCURED INDICATING BOUNCE
1429
1430                                    ;
1431                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1432                                    ;
1433  005136  104412                    28$:    SCOPEX
1434
```

```
1435                                    ;
1436                                    ;CHECK THAT TOGGLING MOD EN W/O RIF DOES CAUSE INTERRUPT
1437                                    ;
1438
1439  005140  013700  014416                   MOV     FR20,R0
1440  005144  005037  017206                   CLR     INTFLG
1441  005150  042777  000004  174646           BIC     #MODEN,@ICSR    ;TOGGLE MODULE INTERRUPT ENABLE
1442  005156  052777  000004  174640           BIS     #MODEN,@ICSR
1443  005164  005300                           DEC     R0
1444  005166  001376                           BNE     .-2
1445  005170  005737  017206                   TST     INTFLG
1446  005174  001001                           BNE     3$
1447
1448                                    ;
1449                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1450                                    ;
1451  005176  104027                           ERROR 27                ;MODULE INTERRUPT DID NOT HAPPEN
1452
1453                                    ;
1454                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1455                                    ;
1456  005200  104412                   3$:     SCOPEX
1457
1458  005202  042777  000004  174614           BIC     #MODEN,@ICSR
1459  005210  052777  000001  174606           BIS     #XRIF,@ICSR     ;SET RIF BIT
1460  005216  005777  174600                   TST     @ICMOD  ;CLEAR RIF BIT
1461  005222  052777  040000  174574           BIS     #MAINT3,@ICSR   ;INITIALIZE ICR CONTROLLER
1462  005230  005077  174570                   CLR     @ICSR           ;EXIT MICROCODE
1463
1464                                    ;
1465                                    ;TEST OF INTERRUPT LOGIC WITH MULTIPLE INTERRUPTS
1466                                    ;BMT AND MODULE INTERRUPTS ARE USED FOR THIS TEST
1467                                    ;ONCE EITHER INTERRUPT OCCURS TOGGLING EITHER ENABLE
1468                                    ;SHOULD  CAUSE ANOTHER INTERRUPT
1469                                    ;MICROCODE IS USED IN THIS SECTION TO CAUSE INTERRUPTS
1470
1471  005234  012737  005234  014372  INTST1: MOV     #INTST1,SCOLOP  ;SET SCOPE LOOP
1472  005242  012737  000340  177776           MOV     #340,@#PSW      ;INHIBIT INTERRUPTS
1473  005250  005077  174558                   CLR     @ICSR           ;CLEAR ICR CSR
1474  005254  013700  014414                   MOV     FR200,R0
1475  005260  005777  174548           4$:     TST     @ICSR           ;OUTPUT BUSY?
1476  005264  100003                           BPL     5$              ;NO, CONTINUE
1477  005266  005300                           DEC     R0
1478  005270  001373                           BNE     4$
1479
1480                                    ;
1481                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1482                                    ;
1483  005272  104035                           ERROR 35                ;OUTPUT BUSY STUCK ON
1484                                    ;
1485                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1486                                    ;
1487  005274  104412                   5$:     SCOPEX
1488
1489  005276  052777  020000  174520           BIS     #MAINT2,@ICSR   ;SET MICROCODE MAINT BIT
1490  005304  005037  017206                   CLR     INTFLG          ;CLEAR INTERRUPT OCCURANCE FLAG
```

```
1491  005310  013737  017172  177776          MOV     PRILVL,@#PSW    ;ALLOW INTERRUPTS
1492  005316  013700  014416                  MOV     FR2#,R#
1493  005322  052777  000014  174474          BIS     #MODEN+BMTEN,@ICSR      ;SET BMT AND MODULE INTERRUPT ENABLES
1494  005330  005300                          DEC     R#
1495  005332  001376                          BNE     .-2
1496  005334  005737  017206                  TST     INTFLG          ;DID INTERRUPT OCCUR?
1497  005340  001001                          BNE     1$
1498
1499                                  ;
1500                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1501                                  ;
1502  005342  104025                          ERROR 25                ;NO INTERRUPT WITH MULTIPLE ENABLES
1503                                  ;
1504                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1505                                  ;
1506  005344  104412                  1$:     SCOPEX
1507
1508                                  ;INTERRUPT OCCURED; LET'S BOUNCE MOD ENABLE
1509                                  ;W/O RIF
1510
1511  005346  013700  014416                  MOV     FR2#,R#
1512  005352  005037  017206                  CLR     INTFLG
1513  005356  042777  000004  174440          BIC     #MODEN,@ICSR    ;BOUNCE MODULE INTERRUPT ENABLE
1514  005364  052777  000004  174432          BIS     #MODEN,@ICSR
1515  005372  005300                          DEC     R#
1516  005374  001376                          BNE     .-2
1517  005376  005737  017206                  TST     INTFLG          ;INTERRUPT OCCUR?
1518  005402  001401                          BEQ     2$
1519
1520                                  ;
1521                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1522                                  ;
1523  005404  104027                          ERROR 27                ;BOUNCED MODULE INTERRUPT ENABLE--NO INTERRUPT
1524                                  ;
1525                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1526                                  ;
1527  005406  104412                  2$:     SCOPEX
1528
1529  005410  013700  014416                  MOV     FR2#,R#
1530  005414  005037  017206                  CLR     INTFLG
1531  005420  042777  000014  174376          BIC     #MODEN+BMTEN,@ICSR      ;BOUNCE BMT ENABLE
1532  005426  052777  000010  174370          BIS     #BMTEN,@ICSR
1533  005434  005300                          DEC     R#
1534  005436  001376                          BNE     .-2
1535  005440  005737  017206                  TST     INTFLG          ;INTERRUPT OCCUR?
1536  005444  001001                          BNE     3$
1537
1538                                  ;
1539                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1540                                  ;
1541  005446  104031                          ERROR 31                ;BOUNCED BMT INTERRUPT ENABLE - NO INTERRUPT
1542                                  ;
1543                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1544                                  ;
1545  005450  104412                  3$:     SCOPEX
1546
```

```
1547  005452  042777  000014  174344          BIC     #MODEN+BMTEN,@ICSR      ;CLEAR ENABLES
1548  005460  052777  000001  174336          BIS     #XRIF,@ICSR             ;SET RIF BIT
1549  005466  005777  174330                  TST     @ICMOD                  ;CLEAR RIF BIT
1550  005472  052777  040000  174324          BIS     #MAINT3,@ICSR           ;RESET MICROCODE IN CONTROLLER
1551  005500  005077  174320                  CLR     @ICSR
1552  005504  005337  014456                  DEC     PASRUN                  ;DEC PASS COUNT
1553  005510  001402                          BEQ     .+6                     ;DONE, .+6
1554  005512  000137  004410                  JMP     LOOP1                   ;LOOP BACK
1555  005516  032777  000040  173412          BIT     #SW5,@SWR               ;LOOP ON TEST SET
1556  005524  001002                          BNE     .+6                     ;YES, SEE IF THIS ONE
1557  005526  000137  011330                  JMP     TEST3                   ;NO, CONT
1558  005532  017700  173400                  MOV     @SWR,R0         ;PICK UP SWR
1559  005536  042700  177477                  BIC     #177477,R0              ;CLEAR UNWANTED BITS
1560  005542  001002                          BNE     .+6                     ;RESULT 0, FOR TEST 0
1561  005544  000137  003504                  JMP     LPTST                   ;YES, LOOP ON TEST 0
1562  005550  000137  011330                  JMP     TEST3
1563
1564
1565                                  .SBTTL  TEST OF MICRO-DIAGNOSTIC IN M8096
1566                                  ;
1567                                  ;THIS SECTION USES THE MAINTENENCE MICROCODE OF THE REMOTE
1568                                  ;CONTROL BOARD (M8096). M8094 IS RUNNING ITS OWN CODE.
1569                                  ;MICROCODE IS KEYED ON ICR CSR BITS 8 AND 11
1570                                  ;
1571
1572  005554  012737  000340  177776  TEST2:  MOV     #340,@#PSW      ;INHIBIT INTERRUPTS
1573  005562  000240                          NOP
1574  005564  000240                          NOP
1575  005566  000240                          NOP
1576  005570  005037  014466                  CLR     REMPRE
1577  005574  005037  014452                  CLR     PRIIND
1578  005600  012737  000001  014462          MOV     #1,PRRMT        ;BUFFER ERRORS
1579  005606  012737  000001  014464          MOV     #1,PRRMT1       ;DON'T ALLOW PRINT TO REMOTE
1580  005614  104415                          CKCNC
1581  005616  013737  014454  014456          MOV     PASLG,PASRUN
1582  005624  016437  000002  014442          MOV     ERRTOT(4),ERTTL ;SAVE PRESENT ERROR LEVEL FOR SCOPE PURPOSES
1583          005632                  LOOP2=.
1584  005632  005037  014422                  CLR     FLAG1           ;CLEAR FIRST GOOD FLAG
1585  005636  012737  000012  014460  18:     MOV     #10.,ERRLOP
1586  005644  012737  005644  014372  958:    MOV     #958,SCOLOP
1587  005652  052777  004400  174144          BIS     #MAINT0+MAINT1,@ICSR ;SET MAINT BITS
1588  005660  012705  004000                  MOV     #4000,R5        ;SET MAINT1 BIT IN R5
1589  005664  000240                          NOP
1590                                  ;
1591                                  ;TEST FOR MODULE INPUT RANGE
1592                                  ;
1593  005666  012701  000050                  MOV     #40.,R1
1594  005672  012702  017532                  MOV     #ADDTAB,R2
1595  005676  013700  014410          628:    MOV     FR500,R0        ;SET TIMEOUT FOR TESTING
1596  005702  005777  174116                  TST     @ICSR           ;OF OUTPUT BUSY
1597  005706  100003                          BPL     .+10
1598  005710  005300                          DEC     R0
1599  005712  001373                          BNE     .-10
1600
1601                                  ;
1602                                  ;\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
```

```
1603                                    ;
1604  005714  104035                            ERROR 35                ;OUT
1605
1606                                    ;
1607                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1608                                    ;
1609  005716  013700  014410                  MOV     FR500,R0
1610  005722  010077  174074                  MOV     R0,@ICMOD
1611  005726  032777  000200  174070  6001:   BIT     #MODINT,@ICSR   ;MODINT INDICATES M8096 HAS EXECUTED PROPER CODE
1612  005734  001014                          BNE     81$
1613  005736  005300                          DEC     R0              ;INC WATCHDOG
1614  005740  001372                          BNE     600$
1615
1616  005742  052777  000001  174054          BIS     #XRIF,@ICSR     ;SET RIF
1617  005750  005777  174046                  TST     @ICMOD          ;INTIATE RIF
1618  005754  040577  174044                  BIC     R5,@ICSR        ;CLEAR MAINT 1 TO STALL 96 MICROCODE
1619
1620                                    ;
1621                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1622                                    ;
1623  005760  104034                            ERROR   34              ;MODULE INTERRUPT NOT POSTED BY M8096
1624
1625                                    ;
1626                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1627                                    ;
1628  005762  000137  006344                  JMP     158
1629
1630
1631  005766  017722  174032          81$:    MOV     @ICSR,(2)+      ;STORE RESULTS
1632  005772  017722  174030                  MOV     @ICAR,(2)+
1633  005776  052777  000001  174020          BIS     #XRIF,@ICSR     ;SET RIF
1634  006004  005777  174012                  TST     @ICMOD          ;INTIATE RIF
1635  006010  005301                          DEC     R1              ;DONE?
1636  006012  001331                          BNE     62$             ;LOOP
1637
1638                                    ;
1639                                    ;ALL READS ARE DONE LET'S CLEAR MAINT BIT AND VERIFY
1640                                    ;THAT EVERY THING IS ALL DONE
1641                                    ;
1642
1643  006014  040577  174004                  BIC     R5,@ICSR        ;CLEAR MAINT1 TO STALL '96 MICROCODE
1644  006020  012705  000046                  MOV     #38.,R5
1645  006024  012702  017542                  MOV     #ADDTAB+18,R2
1646  006030  012200                  88$:    MOV     (2)+,R0         ;GET READ
1647  006032  032700  010000                  BIT     #ERRBIT,R0      ;ERR BIT SET?
1648  006036  001402                          BEQ     82$             ;NO, CCNT
1649  006040  005264  000004                  INC     ERRCNT(4)       ;YES, INDICATE IT
1650  006044  012200                  82$:    MOV     (2)+,R0         ;GET ICAR
1651  006046  005305                          DEC     R5              ;DONE
1652  006050  001454                          BEQ     118             ;YES, EXIT
1653
1654  006052  042700  177760                  BIC     #177760,R0      ;CLEAR UNWANTED BITS
1655  006056  005737  014422                  TST     FLAG1           ;FIRST GOOD READ?
1656  006062  001012                          BNE     9$              ;NO, THEN 9$
1657  006064  010001                          MOV     R0,R1           ;R1 IS FIRST ADDR
1658  006066  010037  001124                  MOV     R0,$GDDAT       ;$GDDAT IS LAST RECV ADDR
```

```
1659  006072  010037  014432                  MOV     R0,MGH          ;STORE AS HIGH VALUE
1660  006076  010037  014434                  MOV     R0,LOW          ;STORE AS LOW VALUE
1661  006102  005237  014422                  INC     FLAG1           ;SET FIRST RECEPTION FLAG
1662  006106  000431                          BR      10$             ;BRANCH
1663
1664  006110  020001                  98$:    CMP     R0,#1           ;IS NEW READ TO FIRST READ
1665  006112  001433                          BEQ     11$             ;YES, 11$
1666  006114  020037  014434                  CMP     R0,LOW          ;IS NEW LOW
1667  006120  002002                          BGE     99$             ;NO, DON'T SWAP
1668  006122  010037  014434                  MOV     R0,LOW          ;YES, SWAP
1669  006126  020037  014432          9981:   CMP     R0,MGH          ;IS NEW HIGH
1670  006132  003402                          BLE     98$             ;NO, DON'T SWAP
1671  006134  010037  014432                  MOV     R0,MGH          ;YES, SWAP
1672  006140  010003                  985:    MOV     R0,R3           ;LETS CHECK IF GOOD
1673  006142  163703  001124                  SUB     SGDDAT,R3       ;THAT POLL ADDRESSES
1674  006146  022703  000001                  CMP     #1,R3           ;OCCUR IN SUCCESSION
1675  006152  001407                          BEQ     10$             ;YES THEN 10$
1676  006154  005700                          TST     R0              ;REPEAT?
1677  006156  001405                          BEQ     10$             ;YES
1678  006160  005237  001124                  INC     SGDDAT
1679  006164  010037  001126                  MOV     R0,SBDDAT
1680
1681                                  ;
1682                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1683                                  ;
1684  006170  104020                          ERROR   20              ;INPUT ADDRESSES NOT IN SUCCESSION
1685
1686                                  ;
1687                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1688                                  ;
1689  006172  010037  001124          10$:    MOV     R0,SGDDAT       ;MOVE THIS ADDRESS TO
1690  006176  000137  006030          968:    JMP     @#8$            ;LOOP BACK
1691
1692                                  ;
1693                                  ;PRINT RESULTS OF TEST
1694                                  ;
1695
1696  006202  042777  004405  173614  11$:    BIC     #MAINT1!MAINT0!XRIF!MODEN,@#CSR ;CLEAR MAINT BITS RIF MOD INT ENA
1697  006210  005737  014466                  TST     REMPRE          ;REMOTE PRESENT
1698  006214  001043                          BNE     12$             ;YES CONT
1699  006216  005764  000000                  TST     PASCNT(4)       ;FIRST PASS
1700  006222  001040                          BNE     12$             ;NO, DON'T PRINT RESULTS
1701  006224  005737  014452                  TST     PRIIND          ;ARE WE LOOPING ON THIS TEST
1702  006230  001035                          BNE     12$             ;YES, DON'T PRINT
1703  006232  013702  014432          14$:    MOV     MGH,R2          ;SET UP TO PRINT
1704  006236  013701  014434                  MOV     LOW,R1
1705  006242  013764  014432  000006          MOV     MGH,PERMGH(4)
1706  006250  013764  014434  000010          MOV     LOW,PERLOW(4)
1707  006256  005037  017204                  CLR     WRKFLP
1708  006262  032777  020000  172646          BIT     #SW13,@#SWR
1709  006270  001030                          BNE     13$
1710  006272  104413  023124                  TYPE,   RESULT
1711  006276  010146                          MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
1712  006300  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1713  006302  104413  023173                  TYPE,   FINMES
1714  006306  010246                          MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
```

M6

```
1715  006310  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1716  006312  104413  022000                  TYPE,   CRLF
1717  006316  005237  014466                  INC     REMPRE
1718  006322  000413                          BR      13$
1719  006324  023764  014432  000006  12$:    CMP     MGM,PERMGM(4)   ;RESULTS PRINTED ONLY
1720  006332  001337                          BNE     14$             ;IF DIFFERENT FROM 1ST PASS
1721  006334  023764  014434  000010          CMP     LOW,PERLOW(4)
1722  006342  001333                          BNE     14$
1723  006344  042777  004405  173452  15$:    BIC     #MAINT0+MAINT1+XRIF+MODEN,@ICSR
1724                                          ;
1725                                          ;TEST OF ADDRESS WRAP AROUND CAPABILITY IN M8096
1726                                          ;
1727
1728  006352  013737  002022  001124  13$:    MOV     ICMOD,SGDDAT    ;SET UP GOOD DATA
1729  006360  012702  000020                  MOV     #20,R2          ;SET UP COUNT
1730  006364  012737  000012  014460  ADWR1:  MOV     #10.,ERRLOP
1731  006372  012737  006372  014372  ADWR2:  MOV     #ADWR2,SCOLOP
1732  006400  052777  000007  173416          BIS     #XRIF+MODEN+ERREN,@ICSR
1733  006406  013700  014414                  MOV     FR200,R0        ;SET TIMEOUT
1734  006412  005777  173406          70$:    TST     @ICSR           ;IS OUTPUT BUSY
1735  006416  100003                          BPL     .+10            ;IF NO CONT
1736  006420  005300                          DEC     R0              ;INC WATCHDOG
1737  006422  001373                          BNE     70$             ;NOT TIMED OUT THEN 70$
1738
1739                                          ;
1740                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1741                                          ;
1742  006424  104035                          ERROR   35              ;OUTPUT BUSY ERROR
1743
1744                                          ;
1745                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1746                                          ;
1747  006426  052777  004000  173370          BIS     #MAINT1,@ICSR   ;SET MAINT BIT FOR NEXT TEST
1748  006434  000240                  ADDWRP: NOP
1749  006436  000240                          NOP
1750  006440  000240                          NOP
1751  006442  010077  172456                  MOV     R0,@SGDDAT      ;SEND TRANSMISSION
1752  006446  013700  014410                  MOV     FR500,R0        ;SET TIMEOUT
1753  006452  032777  000200  173344  60$:    BIT     #MODINT,@ICSR   ;MOD INT KEYS M8096 HAS EXECUTED CODE
1754  006460  001003                          BNE     .+10
1755  006462  005300                          DEC     R0              ;INC WATCH DOG
1756  006464  001372                          BNE     60$
1757
1758                                          ;
1759                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1760                                          ;
1761  006466  104034                          ERROR   34              ;MODULE INTERRUPT NOT POSTED BY M8096
1762
1763                                          ;
1764                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1765                                          ;
1766  006470  000240                          NOP
1767  006472  013700  014414                  MOV     FR200,R0        ;SET TIMEOUT
1768  006476  005777  173322          70$:    TST     @ICSR           ;IS OUTPUT BUSY
1769  006502  100003                          BPL     .+10            ;IF NO CONT
1770  006504  005300                          DEC     R0              ;INC WATCHDOG
```

```
1771  006506  001373                          BNE     70$             ;NOT TIMED OUT THEN 70$
1772
1773                                  ;
1774                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1775                                  ;
1776  006510  104035                          ERROR   35              ;OUTPUT BUSY ON
1777
1778                                  ;
1779                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1780                                  ;
1781
1782  006512  032777  010000  173304          BIT     #ERRBIT,@ICSR   ;ERROR SET
1783  006520  001415                          BEQ     97$
1784  006522  005264  000004                  INC     ERRCNT(4)       ;INDICATE ERROR ON LINE
1785  006526  017700  173274                  MOV     @ICAR,R0
1786  006532  042777  004000  173264          BIC     #MAINT1,@ICSR
1787  006540  013700  014410                  MOV     FR500,R0
1788  006544  005300                          DEC     R0
1789  006546  001376                          BNE     .-2
1790  006550  000137  006646                  JMP     1$
1791
1792
1793  006554  017737  173246  001126  97$:   MOV     @ICAR,$BDDAT    ;READ ICAR
1794  006562  042777  004000  173234          BIC     #MAINT1,@ICSR   ;CLEAR MAINT BIT
1795  006570  013701  014410                  MOV     FR500,R1
1796  006574  005301                          DEC     R1
1797  006576  001376                          BNE     .-2
1798  006600  042737  177760  001126          BIC     #177760,$BDDAT  ;CLEAR BITS
1799  006606  006337  001126                  ASL     $BDDAT          ;SHIFT LEFT TO ALIGN
1800  006612  053737  002022  001126          BIS     ICMOD,$BDDAT    ;BIT SET IN OFFSET
1801  006620  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;DID WE GET RIGHT ADDRESS
1802  006626  001401                          BEQ     2$
1803
1804                                  ;
1805                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1806                                  ;
1807  006630  104022                          ERROR   22              ;ADDRESS THAT CAME BACK IS NOT CORRECT
1808
1809                                  ;
1810                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1811                                  ;
1812
1813  006632  062737  000002  001124  2$:     ADD     #2,$GDDAT       ;SETUP FOR NEXT ADDRESS
1814  006640  005302                          DEC     R2              ;DONE?
1815  006642  001250                          BNE     ADWR1           ;LOOP BACK
1816
1817
1818  006644  000240                          NOP
1819
1820                                  ;TEST IS DONE RELEASE
1821                                  ;
1822  006646  052777  000400  173150  1$:     BIS     #MAINT0,@ICSR   ;SET MAINT BIT
1823  006654  013700  014410                  MOV     FR500,R0
1824  006660  005300                          DEC     R0              ;WAIT
1825  006662  001376                          BNE     .-2
1826  006664  042777  000400  173132          BIC     #MAINT0,@ICSR   ;CLEAR MAINT BIT
```

```
1827  006672  013700  014410                  MOV     FR500,R0
1828  006676  005300                          DEC     R0
1829  006700  001376                          BNE     .-2
1830  006702  052777  040000  173114          BIS     #MAINT3,@ICSR   ;RESET M8094
1831  006710  013700  014410                  MOV     FR500,R0
1832  006714  005300                          DEC     R0              ;WAIT
1833  006716  001376                          BNE     .-2
1834  006720  017700  173102                  MOV     @ICAR,R0        ;CLEAR ANY ERRORS PENDING
1835                                                                  ;DUE TO RELEASE
1836  006724  005237  014452                  INC     PRIIND
1837  006730  005337  014456                  DEC     PASRUN
1838  006734  001402                          BEQ     .+6
1839  006736  000137  005632                  JMP     LOOP2
1840  006742  026437  000002  014442          CMP     ERRTOT(4),ERTTL ;IS ERROR COUNT SAME AS WHEN WE ENTERED
1841  006750  001406                          BEQ     99$             ;YES EXIT
1842  006752  032777  040000  172156          BIT     #SW14,@SWR      ;NO, THEN CHECK FOR POSSIBLE SCOPE LOOP
1843  006760  001402                          BEQ     .+6
1844  006762  000137  005632                  JMP     LOOP2
1845  006766  032777  000040  172142  9981:   BIT     #SW5,@SWR       ;LOOP ON TEST SET
1846  006774  001411                          BEQ     3$              ;NO, CONT
1847  006776  017700  172134                  MOV     @SWR,R0         ;GET SWR
1848  007002  042700  177477                  BIC     #177477,R0      ;CLEAR UNWANTED BITS
1849  007006  022700  000200                  CMP     #200,R0         ;TEST 2 DESIRED
1850  007012  001002                          BNE     .+6
1851  007014  000137  005554                  JMP     TEST2
1852  007020  000240                  3$:     NOP
1853  007022  000240                          NOP
1854  007024  000240                          NOP
1855  007026  000137  007036                  JMP     TTYTST          ;GO TO NEXT TEST
1856
1857
1858
1859                                  .SBTTL  TEST OF TTY LOOP BACK
1860                                  ;
1861                                  ;REMOTE TTY TEST - SHORTING BLOCK IS ASSUMED PRESENT IN M8096
1862                                  ;
1863
1864  007032  005064  000000          TTYSTR: CLR     PASCNT(4)
1865  007036  032777  002000  172072  TTYTST: BIT     #SW10,@SWR      ;RUNNING REMOTELY?
1866  007044  001402                          BEQ     .+6             ;NO, DO TTY TEST
1867  007046  000137  010304                  JMP     ENDPAS          ;YES, CAN'T DO TTY TEST
1868  007052  012706  001100                  MOV     #1100,SP        ;SET STACK
1869  007056  005037  014464                  CLR     PRRMT1          ;DON'T ALLOW REMOTE TYPE
1870  007062  005037  014462                  CLR     PRRMT           ;DON'T BUFFER ERRORS
1871  007066  005737  014472                  TST     REMOTE          ;DID OPERATOR SET UP TEST WRONG?
1872  007072  001402                          BEQ     .+6             ;NO, CONT
1873  007074  000137  010304                  JMP     ENDPAS          ;YES, DON'T DO TEST
1874  007100  005764  000000                  TST     PASCNT(4)       ;FIRST PASS
1875  007104  001002                          BNE     TTYL            ;YES, DON'T TYPE HEADER
1876  007106  104413                          TYPE
1877  007110  025035                          TSTTTY
1878          007112                  TTYL=.
1879  007112  012737  000001  014466          MOV     #1,REMPRE
1880  007120  052777  000041  172676          BIS     #TTYEN+XRIF,@ICSR       ;CLEAR ANY PENDING DA
1881  007126  017700  172670                  MOV     @ICMOD,R0
1882  007132  005000                          CLR     R0                      ;WAIT
```

```
1883  007134  005200                          INC     R0
1884  007136  001376                          BNE     .-2
1885  007140  017700  172656                  MOV     @ICMOD,R0               ;CLEAR AGAIN
1886  007144  005000                          CLR     R0                      ;WAIT
1887  007146  005200                          INC     R0
1888  007150  001376                          BNE     .-2
1889  007152  042777  000040  172644          BIC     #TTYEN,@ICSR            ;CLEAR ENABLE
1890  007160  012737  000340  177776  18$     MOV     #340,@#PSW              ;INHIBIT INTERRUPTS
1891  007166  005037  014460                  CLR     ERRLOP
1892  007172  005037  001124          58$     CLR     $GDDAT                  ;GDDAT USED FOR PATTERN
1893  007176  012737  007176  014372  38$     MOV     #38,$COLOP
1894  007204  052777  001041  172612          BIS     #TTYEN+TBMTEN+XRIF,@ICSR        ;SET BITS IN CSR
1895  007212  013700  014410                  MOV     FR500,R0
1896  007216  032777  100000  172602          BIT     #XTBMT,@ICAR            ;TRANSMITTER BUFFER EMPTY?
1897  007224  001003                          BNE     .+10
1898  007226  005300                          DEC     R0
1899  007230  001372                          BNE     .-12
1900
1901                                  ;
1902                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1903                                  ;
1904  007232  104015                          ERROR 15                ;TBMT NEVER SET
1905
1906                                  ;
1907                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1908                                  ;
1909  007234  042777  001000  172562          BIC     #TBMTEN,@ICSR           ;CLEAR ENABLE
1910
1911  007242  013700  014410                  MOV     FR500,R0
1912  007246  005777  172552                  TST     @ICSR                   ;OUTPUT NOT BUSY?
1913  007252  100003                          BPL     .+10
1914  007254  005300                          DEC     R0
1915  007256  001373                          BNE     .-10
1916
1917                                  ;
1918                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1919                                  ;
1920  007260  104035                          ERROR 35                ;OUTPUT BUSY
1921
1922                                  ;
1923                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1924                                  ;
1925  007262  104411                          SCOTTY
1926
1927  007264  013777  001124  172530          MOV     $GDDAT,@ICMOD   ;SEND PATTERN
1928  007272  052777  001000  172524          BIS     #TBMTEN,@ICSR
1929  007300  032777  100000  172520          BIT     #XTBMT,@ICAR            ;TRANSMITTER BUFFER SET?
1930  007306  001401                          BEQ     96$
1931
1932                                  ;
1933                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
1934                                  ;
1935  007310  104053                          ERROR   53                      ;TBMT SET AFTER TRANS
1936
1937                                  ;
1938                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
```

```
                                        ;
007312  104411                  9651:   SCOTTY
007314  042777  001000  172502          BIC     #TBMTEN,@ICSR
007322  017700  172474                  MOV     @ICMOD,R0
                                        ;
007326  013700  014420                  MOV     FRM110,R0
007332  032777  010000  172464  995:    BIT     #ERRBIT,@ICSR   ;ERROR BIT SET?
007340  001402                          BEQ     97$             ;NO,THEN97$
007342  005264  000004                  INC     ERRCNT(4)       ;YES, THEN ACCOUNT FOR IT
007346  032777  010000  172452  975:    BIT     #DA,@ICAR       ;DA SET?
007354  001016                          BNE     98$
007356  005300                          DEC     R0              ;INC WATCHDOG
007360  001364                          BNE     99$
                                        ;
007362  005737  014466                  TST     REMPRE          ;IT MAY BE OPERATOR ERROR, DON'T FLAG
007366  001012                          BNE     98$+2           ;YET
007370  005237  014460                  INC     ERRLOP
                                        ;
007374  013700  014420                  MOV     FRM110,R0               ;SET TO DELAY 110 MSEC
007400  005300                          DEC     R0                      ;TO ALLOW ANY LATE DA'S
007402  001376                          BNE     .-2
007404  017700  172412                  MOV     @ICMOD,R0               ;CLEAR DA IF THERE
                                        ;
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
007410  104033                          ERROR   33              ;DA NOT SET IN TTY LOOP TEST
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
007412  104411                  9885:   SCOTTY
                                        ;
007414  052777  001000  172402          BIS     #TBMTEN,@ICSR
007422  032777  100000  172376          BIT     #XTBMT,@ICAR    ;TRANSMITTER EMPTY?
007430  001001                          BNE     95$             ;YES, CONT
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
007432  104054                          ERROR   54              ;TBMT NOT SET
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
007434  104411                  955:    SCOTTY
                                        ;
007436  042777  001000  172360          BIC     #TBMTEN,@ICSR
007444  052777  000001  172352          BIS     #XRIF,@ICSR     ;SET RIF BIT
007452  017737  172344  001126          MOV     @ICMOD,$BDDAT   ;READ TTY
                                        ;
007460  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;READ OKAY
007466  001425                          BEQ     2$              ;YES CONT?
```

```
1995
1996  007470  005737  001124                  TST     SGDDAT          ;IS GOOD DATA 0?
1997  007474  001016                          BNE     4$              ;NO PRINT ERROR
1998  007476  023727  001126  000377          CMP     SBDDAT,#377     ;IS BAD DATA 377?
1999  007504  001012                          BNE     4$              ;NO, PRINT ERROR
2000                                                                  ;GOOD DATA = 0   BAD DATA = 337
2001                                                                  ;APPEARS LIKE PLUG NOT IN END
2002                                                                  ;OPERATOR ERROR, TELL AND SKIP TEST
2003
2004  007506  005764  000000                  TST     PASCNT(4)       ;FIRST PASS?
2005  007512  001005                          BNE     6$              ;NO, SKIP MESSAGE
2006  007514  012737  000001  014472          MOV     #1,REMOTE       ;SET OPERATOR ERROR INDICATOR
2007  007522  104413  024567                  TYPE,   NOPLUG          ;TYPE MESSAGE
2008  007526  000137  010304          6$:     JMP     ENDPAS          ;END PASS
2009
2010  007532  005737  014460          4$:     TST     ERRLOP
2011  007536  001001                          BNE     2$
2012
2013                                  ;
2014                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2015                                  ;
2016  007540  104023                          ERROR 23                ;INCORRECT DATA REC'D FROM TTY
2017
2018                                  ;
2019                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2020                                  ;
2021  007542  104411                  2$:     SCOTTY
2022  007544  005037  014466                  CLR     REMPRE
2023
2024  007550  005237  001124                  INC     SGDDAT
2025  007554  005037  014460                  CLR     ERRLOP
2026  007560  032737  000400  001124          BIT     #400,SGDDAT     ;DONE WHOLE TEST (0-377)
2027  007566  001603                          BEQ     3$              ;NO, CONT
2028  007570  000137  007574                  JMP     TTYINT          ;DO INTERRUPT TEST
2029
2030                                  ;
2031                                  ;TTY INTERRUPT TEST    TEST OF REMOTE TTY UNDER INTERRUPT CONTROL
2032                                  ;
2033
2034  007574  005037  014426          TTYINT: CLR     MICFLP          ;CLEAR MICRO CODE FLOP
2035  007600  005037  017174                  CLR     DATSNT          ;DATA ON IT'S WAY FLOP
2036  007604  005237  017176                  INC     TTYFLP          ;SET TTY FLOP
2037  007610  005037  001124                  CLR     SGDDAT          ;CLEAR PATTERN
2038  007614  005077  172204                  CLR     @ICSR           ;CLEAR ICR CSR
2039  007620  052777  000040  172176  TTY2:   BIS     #TTYEN,@ICSR    ;ENABLE TTY
2040  007626  013700  014410                  MOV     FR500,R0
2041  007632  005777  172166                  TST     @ICSR
2042  007636  100003                          BPL     .+10
2043  007640  005300                          DEC     R0
2044  007642  001373                          BNE     .-10
2045
2046                                  ;
2047                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2048                                  ;
2049  007644  104035                          ERROR 35                ;OUTPUT BUSY STUCK
2050
```

```
2051                                    ;
2052                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2053                                    ;
2054  007646  013777  001124  172146          MOV     $GDDAT,@ICMOD   ;START THE TRANSMISSION
2055
2056  007654  052777  000006  172142  TTY1:   BIS     #MODEN+ERREN,@ICSR      ;SET INTERRUPT ENABLES
2057  007662  013737  017172  177776          MOV     PRILVL,@#PSW    ;ALLOW INTERRUPTS
2058  007670  005000                          CLR     R0
2059  007672  032737  000400  001124  2$:     BIT     #400,$GDDAT     ;DONE
2060  007700  001003                          BNE     1$
2061  007702  005200                          INC     R0
2062  007704  001372                          BNE     2$
2063
2064                                    ;
2065                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2066                                    ;
2067  007706  104057                          ERROR   57              ;TTY TEST HUNG
2068
2069                                    ;
2070                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2071                                    ;
2072          007710                    TTYEND=.
2073
2074  007710  012737  000340  177776  1$:     MOV     #340,@#PSW      ;INHIBIT INTERRUPTS
2075  007716  005077  172102                  CLR     @ICSR           ;CLEAR CSR
2076  007722  005037  017176                  CLR     TTYFLP          ;CLEAR TTY FLAG
2077  007726  032777  000040  171202          BIT     #SW5,@#SWR      ;LOOP ON TEST SW SET
2078  007734  001411                          BEQ     5$              ;NO, CONT
2079  007736  017700  171174                  MOV     @#SWR,R0 ;GET SWR
2080  007742  042700  177477                  BIC     #177477,R0
2081  007746  022700  000300                  CMP     #300,R0
2082  007752  001002                          BNE     .+6
2083  007754  000137  007112                  JMP     TTYL            ;YES
2084  007760  005764  000008          5$:     TST     PASCNT(4)       ;FIRST PASS
2085  007764  001404                          BEQ     6$
2086  007766  005037  002032                  CLR     PWFFLG          ;CLEAR PWR FLG
2087  007772  000137  010304                  JMP     ENDPAS
2088  007776  005737  002032          6$:     TST     PWFFLG          ;PERFORM POWER FAIL TEST
2089  010002  001002                          BNE     .+6             ;YES, GO DO IT
2090  010004  000137  010304                  JMP     ENDPAS          ;NO, PRINT END PASS
2091
2092                                    ;TEST OF POWER FAIL BIT ON ICR
2093
2094  010010  104413  025245                  TYPE,   PWRMS1          ;PRINT "POWER DOWN.....
2095  010014  005037  014464                  CLR     PRRMT1          ;DONT PRINT TO REMOTE END
2096  010020  005037  014462                  CLR     PRRMT
2097  010024  032777  002000  171772          BIT     @#PWRFL,@ICSR   ;SENSE POWER FAIL
2098  010032  001774                          BEQ     .-6             ;WAIT FOR IT
2099  010034  104413  025161                  TYPE,   PWRMES  ;PRINT POWER FAIL SENSED
2100  010040  052777  040000  171756          BIS     #MAINT3,@ICSR
2101  010046  104413  025332                  TYPE,   PWRMS2          ;PRINT "POWER UP.......
2102  010052  032777  010000  171744  4$:     BIT     #ERRBIT,@ICSR   ;IS IT UP
2103  010060  001010                          BNE     3$              ;NO, CLEAR ERROR AND TRY AGAIN
2104  010062  052777  040000  171734          BIS     #MAINT3,@ICSR   ;SYSTEM RESET
2105  010070  012737  000001  014464          MOV     #1,PRRMT1       ;ALLOW REMOTE PRINT
2106  010076  000137  010304                  JMP     ENDPAS          ;GO DO END PRINT
```

```
2107  010102  005777  171720          38:     TST     @ICAR           ;CLEAR ERROR
2108  010106  013700  014414                  MOV     FR200,R0
2109  010112  005300                          DEC     R0
2110  010114  001376                          BNE     .-2
2111  010116  000755                          BR      48
2112
2113
2114
2115
2116
2117                                  ;
2118
2119                                  ;ROUTINE TO PRINT SUMMARY
2120                                  ;
2121
2122  010120  104413  024405          SUMPRT: TYPE,   RUNSUM
2123  010124  013700  014400                  MOV     FILUTS,R0       ;GET FILE UNDER TEST
2124  010130  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
2125                                                                  ;;FILE BOX
2126  010132  104402                          TYPOS                   ;;GO TYPE--OCTAL ASCII
2127  010134     002                          .BYTE   2               ;;TYPE 2 DIGIT(S)
2128  010135     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
2129  010136  104413  024436                  TYPE,   RUNCON
2130  010142  016401  000000                  MOV     PASCNT(4),R1
2131  010146  016402  000002                  MOV     ERRTOT(4),R2
2132  010152  016403  000004                  MOV     ERRCNT(4),R3
2133  010156  005064  000000                  CLR     PASCNT(4)
2134  010162  005064  000002                  CLR     ERRTOT(4)
2135  010166  005064  000004                  CLR     ERRCNT(4)
2136  010172  005237  002060                  INC     SRTFF           ;INDICATE PAS REPORT START
2137  010176  104413  024442                  TYPE,   PASMES
2138  010202  010146                          MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
2139  010204  104404                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2140  010206  004737  010254                  JSR     PC,WATROU
2141  010212  104413  024463                  TYPE,   ERMES
2142  010216  010246                          MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
2143  010220  104404                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2144  010222  004737  010254                  JSR     PC,WATROU
2145  010226  104413  024507                  TYPE,   TRERR
2146  010232  010346                          MOV     R3,-(SP)        ;;SAVE R3 FOR TYPEOUT
2147  010234  104404                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2148  010236  004737  010254                  JSR     PC,WATROU
2149  010242  104413  022000                  TYPE,   CRLF
2150  010246  104413  022000                  TYPE,   CRLF
2151  010252  000207                          RTS     PC
2152
2153
2154                                  ;ROUTINE TO WAIT BETWEEN DISPLAYING RUN SUMMARY
2155
2156  010254  032777  002000  170654  WATROU: BIT     #SW10,@SWR
2157  010262  001407                          BEQ     28
2158  010264  012700  000010                  MOV     #10,R0
2159  010270  005001                          CLR     R1
2160  010272  005301                  18:     DEC     R1
2161  010274  001376                          BNE     .-2
2162  010276  005300                          DEC     R0
```

```
2163  010300  001374                        BNE     1$
2164  010302  000207              2$:       RTS     PC
2165
2166
2167
2168
2169                                        .SBTTL  END OF PASS ROUTINE
2170
2171                              ;
2172
2173  010304  004737  010720      ENDPAS: JSR     PC,PRTMES           ;PRINT ANY ERRORS BUFFERED
2174  010310  005264  000000              INC     PASCNT(4)           ;INC PASS COUNT
2175  010314  005037  014462              CLR     PRRMT               ;ALLOW PRINT TO REMOTE TTY
2176  010320  012737  000001  014464      MOV     #1,PRRMT1           ;
2177  010326  032777  000020  170682  1$: BIT     #SW4,@#SWR          ;TESTING ONE FILE BOX?
2178  010334  001502                      BEQ     SYSTST              ;NO, THEN CHECK ENTIRE SYSTEM
2179          010336              ENDPS1=.
2180  010336  032777  001000  170572      BIT     #SW9,@#SWR          ;LOOP ON DIAGNOSTIC
2181  010344  001440                      BEQ     PRTSUM              ;NO, THEN PRINT SUMMARY
2182  010346  032777  002000  170562      BIT     #SW10,@#SWR         ;REMOTE TTY REQUESTED
2183  010354  001432                      BEQ     2$                  ;NO, THEN GO TO 2$
2184  010356  005737  017204              TST     WRKFLP              ;IS "TESTING" DISPLAYED
2185  010362  001002                      BNE     3$                  ;YES, THEN GO FLASH IT
2186  010364  104413  025146              TYPE,   WRKMES              ;PRINT TESTING
2187  010370  032777  010000  170540  3$: BIT     #SW12,@#SWR
2188  010376  001404                      BEQ     5$
2189  010400  104413  025146              TYPE,   WRKMES
2190  010404  000137  003504              JMP     LPTST
2191  010410  005237  017204      5$:     INC     WRKFLP              ;BLINK WORKING IF RT02
2192  010414  112737  000216  017200      MOVB    #216,BLINK          ;IF NOT SEND NULLS
2193  010422  032737  000001  017204      BIT     #1,WRKFLP
2194  010430  001402                      BEQ     4$
2195  010432  005237  017200              INC     BLINK
2196  010436  104413  017200      4$:     TYPE,   BLINK
2197  010442  000137  003504      2$:     JMP     LPTST               ;LOOP
2198  010446  004737  010120      PRTSUM: JSR     PC,SUMPRT           ;PRINT SUMMARY
2199  010452  000240              WAIT2:  NOP
2200  010454  012777  011116  003722      MOV     #ICRWT,@#VECTOR     ;SET UP FOR ICR TTY AND
2201  010462  012737  011316  000060      MOV     #TTYWT,@#60         ;CONSOLE TTY INTERRUPTS,
2202  010470  052777  000040  171326      BIS     #TTYEN,@#ICSR
2203  010476  017700  171320              MOV     @#ICMOD,R0
2204  010502  042777  000040  171314      BIC     #TTYEN,@#ICSR
2205  010510  104413  024736              TYPE,   WAITIN
2206  010514  052777  000026  171302  WAIT1: BIS     #ERREN+MODEN+PWFEN,@#ICSR  ;ALLOW ICR INTERRUPTS
2207  010522  052777  000100  170412      BIS     #100,@#STKS         ;ALLOW TTY INTERUPTS
2208  010530  005037  177776              CLR     @#PSW               ;CLEAR PSW
2209  010534  000001                      WAIT
2210  010536  000000                      HALT
2211  010540  000776                      BR      .-2
2212  010542  022737  000001  002034  SYSTST: CMP     #1,ICRCNT           ;ONE BOX ON SYSTEM
2213  010550  001002                      BNE     1$                  ;NO, MORE THEN 1$
2214  010552  000137  010336              JMP     ENDPS1
2215  010556  032777  001000  170352  1$: BIT     #SW9,@#SWR          ;LOOPING
2216  010564  001013                      BNE     2$                  ;YES DO NOT PRINT SUMMARY
2217  010566  005737  002056              TST     SW9FF               ;WAS SW 9 EVER SET
2218  010572  001406                      BEQ     5$                  ;NO, ALLOW PRINT
```

```
2219  010574  005737  002030              TST     ICRRUN                  ;FIRST SUMMARY
2220  010600  001403                      BEQ     5$                      ;YES, ALLOW PRINT
2221  010602  005737  002060              TST     SRTFF                   ;IN MIDDLE OF BOXES
2222  010606  001402                      BEQ     2$                      ;YES, DONT PRINT YET
2223  010610  004737  010120      5$:     JSR     PC,SUMPRT               ;PRINT SUMMARY FOR THIS BOX
2224  010614  005237  002030      2$:     INC     ICRRUN                  ;INC RUN COUNT
2225  010620  023737  002030  002034      CMP     ICRRUN,ICRCNT           ;DONE ALL BOXES
2226  010626  001421                      BEQ     3$
2227  010630  062737  000040  002022      ADD     #40,ICMOD               ;SET TO NEXT FILE BOX MODULE
2228  010636  162737  000010  002024      SUB     #10,ICSR                ;SET TO NEXT ICR CSR
2229  010644  162737  000010  002026      SUB     #10,ICAR                ;SET TO NEXT ICR ADDR BUFFER
2230  010652  062704  000012              ADD     #12,R4                  ;MOVE FILE BOX ID POINTER
2231  010656  005337  014402              DEC     FILCNT                  ;DECREASE FILE COUNT
2232  010662  005237  014400              INC     FILUTS                  ;BUMP FILE UNDER TEST
2233  010666  000137  003144              JMP     START2
2234  010672  032777  001000  170236  3$:     BIT     #SW9,@SWR               ;LOOP SWITCH SET
2235  010700  001005                      BNE     4$                      ;YES, CONT
2236  010702  005764  000000              TST     PASCNT(4)               ;PASS COUNT 0, INDICATING PRINT
2237  010706  001002                      BNE     .+6                     ;NO, THEN FORCE LOOP
2238  010710  000137  010452              JMP     WAIT2                   ;NO, SET UP TO LOAD OR RERUN
2239  010714  000137  003044      4$:     JMP     STARTA                  ;LOOP ON DIAGNOSTIC
2240
2241
2242
2243                                      ;
2244                                      ;ROUTINE TO PRINT THE ERROR BUFFER THAT WAS STORED WHILE THE
2245                                      ;DIAGNOSTIC WAS IN MAINTENENCE MODE INHIBITING THE USE OF THE REMOTE
2246                                      ;TERMINAL
2247                                      ;
2248
2249  010720  032777  002000  170210  PRTMES: BIT     #BIT10,@SWR             ;REMOTE REQUESTED
2250  010726  001001                      BNE     .+4                     ;YES, CONT
2251  010730  000207              1$:     RTS     PC                      ;NO, EXIT
2252  010732  012700  017776              MOV     #BUFBEG,R0              ;SET UP START
2253  010736  022710  100601              CMP     #100601,(0)             ;ANYTHING THERE
2254  010742  001772                      BEQ     1$                      ;NO, EXIT
2255
2256  010744  052777  001040  171052  2$:     BIS     #TTYEN+TBMTEN,@ICSR     ;THIS IS JUST A SIMPLE
2257  010752  032777  010000  171046          BIT     #DA,@ICAR               ;TYPE ROUTINE TO THE ICR TTY
2258  010760  001407                      BEQ     7$
2259  010762  017702  171034              MOV     @ICMOD,R2
2260  010766  020227  000203              CMP     R2,#203
2261  010772  001002                      BNE     .+6
2262  010774  000137  016270              JMP     XCKPW1
2263  011000  013702  014420      7$:     MOV     FRM110,R2
2264  011004  032777  100000  171014  5$:     BIT     #XTBMT,@ICAR
2265  011012  001004                      BNE     6$
2266  011014  005302                      DEC     R2
2267  011016  001372                      BNE     5$
2268  011020  000137  016270              JMP     XCKPW1
2269  011024  042777  001000  170772  6$:     BIC     #TBMTEN,@ICSR
2270  011032  112001                      MOVB    (0)+,R1
2271  011034  010177  170762              MOV     R1,@ICMOD
2272  011040  121027  000015              CMPB    (0),#15
2273  011044  001404                      BEQ     3$
2274  011046  121027  000201              CMPB    (0),#201
```

```
2275  011052  001334                          BNE     2$
2276  011054  000414                          BR      4$
2277  011056  012702  000004          3$:     MOV     #4,R2
2278  011062  005001                          CLR     R1
2279  011064  005201                          INC     R1
2280  011066  001376                          BNE     .-2
2281  011070  005302                          DEC     R2
2282  011072  001374                          BNE     .-6
2283  011074  020827  021773                  CMP     R0,@#BUFFIN-1
2284  011100  001321                          BNE     2$
2285  011102  005037  017204                  CLR     WRKPLP
2286  011106  042777  000040  170710  4$:     BIC     #TTYEN,@#ICSR
2287  011114  000207                          RTS     PC
2288
2289
2290  011116  022626                  ICRWT:  POP2SP                  ;ICR INTERRUPTS HERE (FROM QUESTION R OR L?)
2291  011120  032777  002000  170676          BIT     #PWRFL,@#ICSR   ;PWR FAIL INTERRUPT
2292  011126  001405                          BEQ     1$              ;NO, CONTINUE
2293  011130  012737  000001  014474          MOV     #1,PWRFLP       ;INDICATE POWER FAIL
2294  011136  000137  016530                  JMP     @#STRT          ;GO TYPE MESSAGE
2295  011142  032777  000200  170654  1$:     BIT     #MODINT,@#ICSR  ;MOD INT?
2296  011150  001445                          BEQ     ICRWT2          ;NO, CONT (CHECK ERROR)
2297  011152  032777  002000  167756          BIT     #SW10,@#SWR     ;IS REMOTE TTY ON SYSTEM
2298  011160  001441                          BEQ     ICRWT2          ;NO, THEN GO SKIP DA
2299  011162  032777  010000  170636          BIT     #DA,@#ICAR      ;DO SET
2300  011170  001435                          BEQ     ICRWT2          ;NO, GO CHECK ERROR
2301  011172  052777  000041  170624          BIS     #TTYEN+XRIF,@#ICSR  ;GET TO READ ICR TTY
2302  011200  017700  170616                  MOV     @#ICMOD,R0
2303  011204  042777  000040  170612          BIC     #TTYEN,@#ICSR
2304  011212  042700  000240          ICRWT3: BIC     #240,R0         ;LOOK FOR 'R' OR 'L'
2305  011216  022700  000114                  CMP     #'L,R0
2306  011222  001407                          BEQ     ICRWT1          ;'L' GO LOAD
2307  011224  022700  000122                  CMP     #'R,R0
2308  011230  001002                          BNE     .+6
2309  011232  000137  016634                  JMP     RST1            ;'R' GO START
2310  011236  000137  010514                  JMP     WAIT1           ;WAIT FOR NEXT
2311  011242  005037  014462          ICRWT1: CLR     PRRMT           ;PRINT 'LOADING'
2312  011246  012737  000001  014464          MOV     #1,PRRMT1
2313  011254  104413  024657                  TYPE,   LOADIN
2314  011260  000137  033000                  JMP     GOLOAD          ;GO TO ABS LOADER
2315  011264  032777  010000  170532  ICRWT2: BIT     #ERRBIT,@#ICSR  ;CHECK FOR ERROR
2316  011272  001402                          BEQ     2$
2317  011274  017700  170526                  MOV     @#ICAR,R0
2318  011300  052777  000001  170516  2$:     BIS     #XRIF,@#ICSR    ;CLEAR INT, EXIT
2319  011306  005777  170510                  TST     @#ICMOD
2320  011312  000137  010514                  JMP     WAIT1
2321
2322
2323
2324  011316  022626                  TTYWT:  POP2SP                  ;CONSOLE TTY INTERRUPTS HERE
2325  011320  017700  167620                  MOV     @#TKB,R0
2326  011324  000137  011212                  JMP     ICRWT3
2327
2328
2329
2330                                  .SBTTL  TEST OF M8094 MICRO-DIAGNOSTICS
```

```
                                        ;
                                        ;LET'S START TEST OF MICRO-CODE
                                        ;ON THE M8094 - MAINTENANCE BIT 2, BIT 13 OF ICSR
                                        ;
                                        ;
                                        ;THIS SECTION RUNS IN SYNCH WITH THE MICROCODE IN THE M8094
                                        ;
011330  012737  000001  014464  TEST3:  MOV     #1,PRRMT1       ;BUFFER ERRORS
011336  005037  014462                  CLR     PRRMT           ;REMOTE PRINT
011342  104415                          CKCNC
011344  005764  000000                  TST     PASCNT(4)       ;FIRST PASS?
011350  001002                          BNE     .+6             ;NO, CONT
011352  104413                          TYPE                    ;TYPE HEADER
011354  022227                          M8094
011356  012737  000001  014462          MOV     #1,PRRMT        ;NO REMOTE PRINT
011364  012737  011364  014372  TEST3L: MOV     #TEST3L,SCOLOP  ;SET SCOPE LOOP
011372  013737  014454  014456          MOV     PASLG,PASRUN    ;GET RUN LENGTH
        011400                  LOOP3:.
011400  052777  040000  170416          BIS     #MAINT3,@ICSR   ;ISSUE MICRO INIT
011406  012737  000340  177776          MOV     #340,@#PSW      ;SET LEVEL TO 7
011414  052777  020000  170402          BIS     #MAINT2,@ICSR   ;SET MAINT BIT
011422  052777  000001  170374          BIS     #1,@ICSR        ;SET RIF BIT
011430  005777  170366                  TST     @ICMOD          ;CLEAR RIF BIT
011434  013700  014410                  MOV     FR500,R0        ;WAIT 10 US
011440  005300                          DEC     R0
011442  001376                          BNE     .-2
011444  013737  017172  177776          MOV     PRILVL,@#PSW    ;SET PRIORITY LEVEL
011452  005037  017206                  CLR     INTFLG          ;CLEAR INTERRUPT OCCURRANCE FLAG
011456  013700  014416                  MOV     FR20,R0
011462  052777  000004  170334          BIS     #MODEN,@ICSR    ;ENABLE MODULE TO INTERRUPT
011470  042777  020000  170326          BIC     #MAINT2,@ICSR   ;CLEAR MAINT. MODE
011476  005300                          DEC     R0
011500  001376                          BNE     .-2
011502  005737  017206                  TST     INTFLG          ;TEST IF INTERRUPT OCCURRED
011506  001001                          BNE     1$              ;YES
                                                                ;NO, INTERRUPT
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
011510  104005                          ERROR 5                 ;INTERRUPT FROM MICRO CODE DID NOT OCCUR
                                                                ;TIMEOUT FLOP ON M8094 WAS FALSE
                                                                ;WHEN IT SHOULD HAVE BEEN TRUE
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
011512  104407          1$:             SCOP94
011514  032777  000200  170302          BIT     #MODINT,@ICSR   ;CHECK THAT MODULE CAUSED
011522  001001                          BNE     2$              ;INTERRUPT
011524  104010                          ERROR 10                ;INTERRUPT WAS NOT CAUSED BY MODULE INTERRUPT
011526  104407          2$:             SCOP94
011530  052777  000001  170266          BIS     #1,@ICSR        ;SET RIF BIT
011536  005777  170260                  TST     @ICMOD          ;CLEAR RIF BIT
```

```
2387  011542  032777  000200  170254          BIT     @MODINT,@ICSR   ;CHECK IF MOD INT IS STILL SET
2388  011550  001401                          BEQ     .+4             ;NO, OK
2389
2390                                          ;
2391                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2392                                          ;
2393  011552  104012                          ERROR 12                ;MOD INT SET AFTER RIF
2394
2395                                          ;
2396                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2397                                          ;
2398  011554  104407                          SCOP94
2399
2400  011556  005037  017206                  CLR     INTFLG          ;CLEAR INTERRUPT OCCURRANCE FLAG
2401  011562  013700  014416                  MOV     FR2@,R0
2402  011566  052777  000004  170230          BIS     @MODEN,@ICSR    ;ENABLE MODULE INTERRUPT
2403  011574  052777  020000  170222          BIS     @MAINT2,@ICSR   ;SET MAINT MODE
2404  011602  005300                          DEC     R0
2405  011604  001376                          BNE     .-2
2406  011606  005737  017206                  TST     INTFLG          ;INTERRUPT OCCUR?
2407  011612  001001                          BNE     38
2408                                                                  ;NO
2409
2410                                          ;
2411                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2412                                          ;
2413  011614  104006                          ERROR 6                 ;INTERRUPT NOT PRESENT
2414                                                                  ;SEQUENCE COMPARE FLOP ON M8094 DID
2415                                                                  ;NOT TEST TRUE IN MICROCODE
2416
2417                                          ;
2418                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2419                                          ;
2420
2421  011616  104407  38:                     SCOP94
2422  011620  032777  000200  170176          BIT     @MODINT,@ICSR   ;CHECK FOR MODULE INTERRUPT
2423  011626  001001                          BNE     TEST3A
2424
2425                                          ;
2426                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2427                                          ;
2428                                                                  ;MODULE INTERRUPT DIDN'T CAUSE INTERRUPT
2429  011630  104010                          ERROR 10                ;MODULE INTERRUPT
2430
2431                                          ;
2432                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2433                                          ;
2434  011632  104407  TEST3A:                  SCOP94
2435  011634  052777  000001  170162          BIS     #1,@ICSR        ;SET RIF BIT
2436  011642  005777  170154                  TST     @ICMOD          ;CLEAR RIF BIT
2437
2438  011646  032777  000200  170150          BIT     @MODINT,@ICSR   ;CHECK IF MOD INT IS STILL SET
2439  011654  001401                          BEQ     .+4             ;NO, OK
2440                                          ;
2441                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2442                                          ;
```

```
2443
2444  011656  104012                          ERROR 12                ;MOD INT SET AFTER RIF
2445
2446                                          ;
2447                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2448                                          ;
2449  011660  104407                          SCOP94
2450
2451                                          ;
2452                                          ;TEST ERROR BIT IS CAPABLE OF INTERRUPTING
2453                                          ;
2454
2455  011662  005037  017206                  CLR     INTFLG          ;CLEAR INTERRUPT OCCURRENCE FLAG
2456  011666  042777  000004  170130          BIC     #MODEN,@ICSR    ;CLEAR MOD INT ENABLBE BIT
2457  011674  052777  000002  170122          BIS     #ERREN,@ICSR    ;SET ERROR INT ENABLE BIT
2458  011702  013700  014416                  MOV     FR20,R0
2459  011706  042777  020000  170110          BIC     #MAINT2,@ICSR   ;CLEAR MAINT BIT
2460  011714  005300                          DEC     R0
2461  011716  001376                          BNE     .-2
2462  011720  005737  017206                  TST     INTFLG          ;DID INTERUPT OCCUR
2463  011724  001001                          BNE     1$              ;YES, THEN 1$
2464
2465                                          ;
2466                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2467                                          ;
2468  011726  104011                          ERROR 11                ;ERROR BIT DID NOT INTERRUPT
2469
2470                                          ;
2471                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2472                                          ;
2473  011730  104407                          1$:     SCOP94
2474
2475                                          ;
2476                                          ;TEST ERROR BIT ACTUALLY CAUSED INTERRUPT
2477                                          ;
2478
2479  011732  005037  017206                  CLR     INTFLG          ;CLEAR INTERRUPT OCCURRANCE FLAG
2480  011736  032777  010000  170060          BIT     @ERRBIT,@ICSR   ;TEST ERROR BIT
2481  011744  001001                          BNE     2$              ;SET, THEN 2$
2482
2483                                          ;
2484                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2485                                          ;
2486  011746  104010                          ERROR 10                ;ERROR BIT WAS NOT CAUSE OF INTERRUPT
2487
2488                                          ;
2489                                          ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2490                                          ;
2491  011750  104407                          2$:     SCOP94
2492
2493                                          ;
2494                                          ;
2495                                          ;CHECK THAT BOUNCING ERROR INTERRUPT ENABLE DOES CAUSE
2496                                          ; ANOTHER INTERRUPT
2497                                          ;
2498
```

```
2499  011752  013700  014416                  MOV     FR20,R0
2500  011756  042777  000002  170040          BIC     #ERREN,@ICSR
2501  011764  052777  000002  170032          BIS     #ERREN,@ICSR
2502  011772  005300                          DEC     R0
2503  011774  001376                          BNE     .-2
2504  011776  005737  017206                  TST     INTFLG
2505  012002  001001                          BNE     99$
2506
2507                                  ;
2508                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2509                                  ;
2510  012004  104030                          ERROR 30                ;BOUNCED ERROR ENABLE - NO INTERRUPT
2511
2512                                  ;
2513                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2514                                  ;
2515  012006  104407                  99$:    SCOP94
2516
2517                                  ;CHECK ERROR BIT CLEARS WITH ADDRESSING ICAR
2518                                  ;
2519
2520  012010  005777  170012                  TST     @ICAR           ;ACCESS ICAR
2521  012014  032777  010000  170002          BIT     #ERRBIT,@ICSR   ;ERROR BIT STILL SET
2522  012022  001401                          BEQ     3$              ;NO, THEN 3$
2523
2524                                  ;
2525                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2526                                  ;
2527  012024  104013                          ERROR 13                ;ERROR BIT DID NOT CLEAR WITH ADDRESSING ICAR
2528
2529                                  ;
2530                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2531                                  ;
2532
2533  012026  104407                  3$:     SCOP94
2534  012030  052777  000001  167766          BIS     #XRIF,@ICSR     ;SET RIF BIT
2535  012036  005777  167760                  TST     @ICMOD          ;ACCESS MODULE TO CLEAR RIF BIT
2536
2537                                  ;
2538                                  ;DATA VALID SHOULD BE FALSE, LETS CHECK IT
2539                                  ;
2540
2541  012042  042777  000002  167754          BIC     #ERREN,@ICSR    ;CLEAR ERROR INT ENABLE
2542  012050  052777  000004  167746          BIS     #MODEN,@ICSR    ;SET MODULE INTERRUPT ENABLE
2543  012056  013700  014416                  MOV     FR20,R0
2544  012062  052777  020000  167734          BIS     #MAINT2,@ICSR   ;SET MAINT BIT
2545  012070  005300                          DEC     R0
2546  012072  001376                          BNE     .-2
2547  012074  005737  017206                  TST     INTFLG          ;INTERRUPT OCCUR
2548  012100  001001                          BNE     4$
2549
2550                                  ;
2551                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2552                                  ;
2553  012102  104014                          ERROR 14                ;DATA VALID WAS TRUE
2554
```

```
2555                                    ;
2556                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2557                                    ;
2558  012104  104407                    48$:    SCOP94
2559                                    ;
2560                                    ;TBMT SHOULD BE TRUE
2561                                    ;
2562
2563
2564  012106  005037  017206                  CLR     INTFLG          ;CLEAR INTERRUPT OCCURRENCE FLAG
2565  012112  052777  000001  167704          BIS     #XRIF,@ICSR     ;SET RIF
2566  012120  005777  167676                  TST     @ICMOD          ;CLEAR RIF
2567  012124  052777  001004  167672          BIS     #MODEN+TBMTEN,@ICSR     ;SET MOD. INT. ENABLE
2568  012132  013700  014416                  MOV     FR20,R0
2569  012136  042777  020000  167660          BIC     #MAINT2,@ICSR   ;SET MAINT BIT
2570  012144  005300                          DEC     R0
2571  012146  001376                          BNE     .-2
2572  012150  005737  017206                  TST     INTFLG          ;INTERRUPT TAKE PLACE?
2573  012154  001001                          BNE     58              ;YES, THEN 58
2574
2575                                    ;
2576                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2577                                    ;
2578  012156  104015                          ERROR 15                ;TBMT WAS NOT TRUE
2579
2580                                    ;
2581                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2582                                    ;
2583  012160  104407                    58$:    SCOP94
2584
2585
2586  012162  005037  017206                  CLR     INTFLG          ;CLEAR INTERRUPT OCCURRENCE FLAG
2587  012166  032777  100000  167632          BIT     #100000,@ICAR   ;DID TBMT REALLY CAUSE THE INTERRUPT
2588  012174  001001                          BNE     78
2589
2590                                    ;
2591                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2592                                    ;
2593  012176  104015                          ERROR 15                ;INTERRUPT WAS NOT CAUSED BY TBMT
2594
2595                                    ;
2596                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2597                                    ;
2598  012200  104407                    78$:    SCOP94
2599
2600  012202  052777  000001  167614          BIS     #XRIF,@ICSR
2601  012210  005777  167606                  TST     @ICMOD
2602  012214  042777  001000  167602          BIC     #TBMTEN,@ICSR
2603
2604                                    .SBTTL   CHECK OF ICR11 ERROR GENERATION CAPABILITIES
2605
2606                                    ;TEST OF ICR11 TO FORCE CRC ERRORS
2607                                    ;
2608                                    ;MASTER END IS DONE FIRST, MICROCODE IS CAPABLE OF CAUSEING
2609                                    ;MASTER TO SEND BAD CRC TO SLAVE AND SLAVE SHOULD NOT RESPOND
2610                                    ;
```

```
2611
2612  012222  005037  017206          TEST38: CLR     INTFLG          ;CLEAR INTERRUPT OCCURANCE
2613  012226  012737  000001  017202          MOV     #1,TIMTRP
2614  012234  012777  000000  167560          MOV     #0,@ICMOD       ;SEND OUT
2615  012242  052777  000006  167554          BIS     #MODEN+ERREN,@ICSR      ;SET ERROR ENABLE AND MODULE ENABLE
2616  012250  013700  014410                  MOV     FR500,R0
2617  012254  052777  020000  167542          BIS     #MAINT2,@ICSR   ;SET MAINT BIT FOR MICROCODE
2618  012262  005300                          DEC     R0              ;WAIT FOR INTERRUPT, INC WATCH DOG
2619  012264  001376                          BNE     .-2
2620
2621  012266  005737  017206                  TST     INTFLG          ;DID INTERRUPT OCCUR?
2622  012272  001001                          BNE     1$
2623
2624                                  ;
2625                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2626                                  ;
2627  012274  104037                  1$:     ERROR 37                ;MASTER CRC ERROR FAILURE
2628
2629                                  ;
2630                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2631                                  ;
2632  012276  104407                  18:     SCOP94
2633
2634  012300  032777  010000  167516          BIT     #ERRBIT,@ICSR   ;DID ERROR BIT INTERRUPT INDICATING
2635  012306  001001                          BNE     2$              ;TIMEOUT OF MASTER
2636
2637
2638                                  ;
2639                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2640                                  ;
2641  012310  104040                  2$:     ERROR 40                ;NO TIMEOUT FROM MASTER
2642
2643                                  ;
2644                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2645                                  ;
2646  012312  104407                  28:     SCOP94
2647
2648  012314  032777  000200  167502          BIT     #MODINT,@ICSR   ;DID MODULE INTERRUPT INDICATING
2649  012322  001401                          BEQ     3$              ;THAT DATA WAS PRESENT
2650
2651                                  ;
2652                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2653                                  ;
2654  012324  104041                          ERROR 41                ;SLAVE RESPONDED TO BAD CRC
2655                                  ;
2656                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2657                                  ;
2658
2659  012326  104407                  38:     SCOP94
2660
2661                                  ;
2662                                  ;NOW CHECK THAT A BAD CRC FROM THE SLAVE WILL NOT FUNCTION PROPERLY
2663                                  ;AT THE MASTER END
2664                                  ;
2665
2666  012330  005737  014374                  TST     TENCN1
```

```
2667  012334  001003                          BNE     TEST3C
2668  012336  012737  000012  014374          MOV     #10.,,TENCN1
2669  012344  005037  017206          TEST3C: CLR     INTFLG          ;CLEAR  INTERRUPT OCCURANCE
2670  012350  052777  000001  167446          BIS     #XRIF,@ICSR             ;SET RIF BIT
2671  012356  005777  167440                  TST     @ICMOD          ;CLEAR RIF BIT
2672  012362  017700  167440                  MOV     @ICAR,R0        ;CLEAR ERROR BIT
2673  012366  052777  000040  167430          BIS     #TTYEN,@ICSR
2674  012374  017700  167422                  MOV     @ICMOD,R0       ;CLEAR DA
2675  012400  042777  000040  167416          BIC     #TTYEN,@ICSR
2676  012406  013700  014410                  MOV     FR500,R0
2677  012412  012777  000300  167402          MOV     #300,@ICMOD     ;SET DATA BITS 6,7
2678  012420  042777  020000  167376          BIC     #MAINT2,@ICSR   ;SET MAINT BIT FOR MICROCODE
2679  012426  005300                          DEC     R0              ;WAIT FOR INTERRUPT
2680  012430  001376                          BNE     .-2
2681
2682  012432  005737  017206                  TST     INTFLG          ;DID INTERRUPT OCCUR
2683  012436  001005                          BNE     1$
2684
2685  012440  005337  014374                  DEC     TENCN1          ;ALLOW 10 CONSECUTIVE ERRORS TO GO BY
2686  012444  001401                          BEQ     5$              ;FOR NOISE IMMUNITY
2687  012446  000426                          BR      8$
2688                                  ;
2689                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2690                                  ;
2691  012450  104042                  5$:     ERROR 42                ;SLAVE  FORCED ERROR FAILURE
2692
2693                                  ;
2694                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2695                                  ;
2696  012452  104407                  1$:     SCOP94
2697
2698  012454  032777  010000  167344          BIT     #DA,@ICAR       ;SEE IF DA IS SET
2699  012462  001411                          BEQ     2$              ;INDICATING     THAT    MASTER TIMED OUT
2700
2701  012464  052777  000040  167332          BIS     #TTYEN,@ICSR    ;CLEAR DA
2702  012472  017700  167324                  MOV     @ICMOD,R0
2703  012476  042777  000040  167320          BIC     #TTYEN,@ICSR
2704
2705                                  ;
2706                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2707                                  ;
2708  012504  104043                          ERROR 43                ;MASTER TIMED OUT WITH BAD CRC FROM SLAVE
2709
2710                                  ;
2711                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2712                                  ;
2713  012506  104407                  2$:     SCOP94
2714
2715  012510  032777  000200  167306          BIT     #MODINT,@ICSR   ;TEST MODULE INTERRUPT INDICATING
2716                                                                  ;THAT CRC FORM SLAVE WAS BAD AS EXPECTED
2717  012516  001001                          BNE     4$
2718
2719                                  ;
2720                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2721                                  ;
2722  012520  104044                          ERROR   44
```

```
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
012522  104407                  48:     SCOP94
                                        ;
012524  052777  000041  167272  88:     BIS     #XRIF+TTYEN,@ICSR
012532  017700  167264                  MOV     @ICMOD,R0
012536  042777  000040  167260          BIC     #TTYEN,@ICSR
012544  017700  167256                  MOV     @ICAR,R0
                                        ;
                                        ;TEST OF M8098'S ABILITY TO TURN LAST DATA BYTE AROUND AND REPEAT
                                        ;IT 4 TIMES, ALL PATTERNS  OF BITS 0-5
                                        ;WILL BE SENT AND READ BACK IN THE ICAR AND MODULE ADDRESSES
                                        ;BITS 7 AND 6 CANNOT BE 11 OR 00
                                        ;
012550  012737  000012  002054          MOV     #10.,TENCNT
012556  005037  014444          TEST3D: CLR TEMP1       ;CLEAR STARTING PATTERN
012562  005037  014460                  CLR     ERRLOP          ;ERROR INDICATOR FOR THIS LOOP
012566  012737  000340  177776          MOV     #340,@PSW       ;INHIBIT INTERRUPTS
012574  052777  001000  167222          BIS     #TBMTEN,@ICSR   ;SET TBMT ENABLE
012602  012737  000001  014446          MOV     #1,TRACK        ;SET LOOP INDICATOR
012610  052737  000100  014444          BIS     #BIT6,TEMP1     ;SET BIT 14 FOR FIRST LOOP
012616  052777  000006  167200          BIS     #MODEN+ERREN,@ICSR      ;SET ENABLES
012624  013777  014444  167170  38:     MOV     TEMP1,@ICMOD    ;SEND PATTERN
012632  052777  020000  167164          BIS     #MAINT2,@ICSR   ;SET MAINT BIT TO START MICROCODE
012640  013700  014410          18:     MOV     FR500,R0
012644  032777  000200  167152          BIT     #MODINT,@ICSR
012652  001003                          BNE     .+10
012654  005300                          DEC     R0
012656  001372                          BNE     18+4
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
012660  104034                          ERROR 34
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
012662  104407                          SCOP94
                                        ;
                                        ;TEST FOR ERRORS WITHIN MICRO CODE
                                        ;IF DA IS POSTED THEN TIMEOUT WAS DETECTED
                                        ;IF ERROR INTERRUPT IS POSTED THEN DATA VALID WAS FALSE
                                        ;
012664  032777  010000  167132          BIT     #ERRBIT,@ICSR   ;TEST FOR ERROR INTERRUPT
012672  001404                          BEQ     48
```

```
                                INC     ERRLOP
                                INC     ERRCNT(4)
                        4$:     BIT     #DA,@ICAR       ;TEST FOR DA
                                BEQ     2$
                                BIS     #TTYEN,@ICSR            ;CLEAR DA
                                MOV     @ICMOD,R0
                                BIC     #TTYEN,@ICSR
                                CKPWF
                        7$:     DEC     TENCNT          ;FOR NOISE IMMUNITY
                                BNE     63$
                                INC     ERRLOP
                                INC     ERRCNT(4)
                        ;
                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                        ;
                                ERROR 45                ;TIMEOUT ERROR
                        ;
                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                        ;
                        2$:     SCOP94
                                TST     ERRLOP
                                BEQ     82$
                        63$:    MOV     @ICAR,R0                ;CLEAR ERROR
                                BIS     #XRIF,@ICSR             ;SET RIF
                                MOV     @ICMOD,R0               ;CLEAR RIF , MOD INTERRUPT
                                CLR     ERRLOP
                                BIT     #SW8,@SWR
                                BEQ     .+6
                                JMP     SWLOOP
                                TST     TENCNT
                                BNE     .+6
                                JMP     72$
                                JMP     99$             ;ERROR SENSED, RE-SEND MESSAGE


                        ;DATA HAS BEEN LOADED INTO THE ICAR IN THE FOLLOWING PATTERN
                        ;(BITS 15-0), WITH THOSE BITS OF THE OUTGOING MESSAGE
                        ;PATTERN IS 6,1,0,7,4,5,6,7,X,X,X,X,3,2,1,0.
                        ;IT HAS ALSO BEEN LOADED INTO THE MODULE ADDRESS AS 7-0,7-0
                        ;X,X,X,X IS THE FILE BOX ADDRESS UNDER TEST

                        ;CHECK ICAR FOR IT BEING CORRECT

                        82$:    MOV     #10.,TENCNT
                                MOV     TEMP1,$GDDAT            ;CALCULATE GOOD DATA,
                                                        ;GOOD LUCK IN FOLLOWING THIS
                                BIC     #177760,$GDDAT  ;CLEAR 15-4
                                MOV     ICMOD,R0        ;GET BITS 7-4
```

```
2835  013070  006200                          ASR     R0              ;DIVIDE FILE BOX BY 2
2836  013072  042700  177417                  BIC     #177417,R0      ;CLEAR 15-8, 3-0
2837  013076  050037  001124                  BIS     R0,$GDDAT       ;BITS 7-0 ARE FORMED
2838  013102  032737  000200  014444          BIT     #BIT7,TEMP1     ;TEST BIT 7 OF MESSAGE SENT
2839  013110  001403                          BEQ     5$              ;NOT SET, THEN 5$
2840  013112  052737  010400  001124          BIS     #10400,$GDDAT   ;SET, THEN SET REQ'D BITS
2841  013120  032737  000100  014444  5$:     BIT     #BIT6,TEMP1     ;TEST BIT 6 OF MESSAGE SENT
2842  013126  001403                          BEQ     6$              ;NOT SET, THEN 6$
2843  013130  052737  101000  001124          BIS     #101000,$GDDAT  ;SET, THEN SET REQ'D BITS
2844  013136  032737  000040  014444  6$:     BIT     #BIT5,TEMP1     ;TEST BIT 5
2845  013144  001403                          BEQ     7$
2846  013146  052737  002000  001124          BIS     #2000,$GDDAT
2847  013154  032737  000020  014444  7$:     BIT     #BIT4,TEMP1     ;TEST BIT 12
2848  013162  001403                          BEQ     8$
2849  013164  052737  004000  001124          BIS     #4000,$GDDAT
2850  013172  032737  000002  001124  8$:     BIT     #BIT1,$GDDAT
2851  013200  001403                          BEQ     9$
2852  013202  052737  040000  001124          BIS     #BIT14,$GDDAT
2853  013210  032737  000001  001124  9$:     BIT     #BIT0,$GDDAT
2854  013216  001403                          BEQ     10$
2855  013220  052737  020000  001124          BIS     #BIT13,$GDDAT
2856
2857                                  ;
2858                                  ;GOOD ICAR IS FORMED  LET'S GET ACTUAL ICAR AND COMPARE
2859                                  ;
2860
2861  013226  017737  166574  001126  10$: MOV #ICAR,$BDDAT
2862  013234  042737  110000  001124          BIC     #BIT15+BIT12,$GDDAT
2863  013242  042737  110000  001126          BIC     #BIT15+BIT12,$BDDAT
2864  013250  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE READ WITH THAT FORMED
2865  013256  001401                          BEQ     11$
2866
2867                                  ;
2868                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2869                                  ;
2870  013260  104047                          ERROR 47                ;BAD ICAR ON READ
2871
2872                                  ;
2873                                  ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2874                                  ;
2875  013262  104407                  11$:    SCOP94
2876
2877                                  ;READ MODULE DATA FOR PROPER DATA
2878
2879  013264  013737  014444  001124          MOV     TEMP1,$GDDAT    ;FORM $GDDAT OUT OF TEMP1
2880  013272  013700  014444                  MOV     TEMP1,R0        ;$GDDAT = TEMP1 BITS 7-0 TWICE
2881  013276  000300                          SWAB    R0
2882  013300  050037  001124                  BIS     R0,$GDDAT
2883  013304  013737  001124  014458          MOV     $GDDAT,ADDR     ;NOW CALCULATE MODULE ADDRESS
2884  013312  042737  177760  014458          BIC     #177760,ADDR    ;OF WHERE DATA SHOULD BE
2885  013320  006337  014458                  ASL     ADDR
2886  013324  053737  002022  014458          BIS     ICMOD,ADDR      ;ADD IN FILE BOX OFFSET
2887  013332  005137  001124                  COM     $GDDAT          ;COMPLIMENT DATA PATTERN
2888  013336  017737  001106  001126          MOV     @ADDR,$BDDAT    ;READ DATA
2889  013344  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE
2890  013352  001401                          BEQ     12$
```

```
2891
2892                                    ;
2893                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2894                                    ;
2895  013354  104050                          ERROR 50                ;BAD DATA IN MODULE ADDRESS
2896
2897                                    ;
2898                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2899                                    ;
2900  013356  104407                    128$:   SCOP94
2901
2902                                    ;
2903                                    ;TESTS ARE DONE  FORM NEXT PATTERN AND LOOP
2904                                    ;
2905
2906          013360                    SWLOOP=.
2907
2908  013360  032777  000400  165550          BIT     #SW8,@SWR               ;LOOP ON DATA TEST
2909  013366  001434                          BEQ     81$                     ;NO THEN 81$
2910  013370  117737  165542  017174          MOVB    @SWR,DATSNT             ;GET SWR LOW BYTE
2911  013376  117737  165534  014444          MOVB    @SWR,TEMP1
2912  013404  042737  177477  017174          BIC     #177477,DATSNT          ;GET BITS 6,7
2913  013412  001004                          BNE     84$                     ;NOT ZERO THEN  84$
2914  013414  052737  000200  014444          BIS     #BIT07,TEMP1            ;ZERO, DEFAULT TO SET BIT 7
2915  013422  000412                          BR      83$                     ;CONT
2916  013424  022737  000300  017174  84$:    CMP     #300,DATSNT             ;BITS 6,7  =11
2917  013432  001006                          BNE     83$                     ;NO THEN 83$
2918  013434  042737  000300  014444          BIC     #300,TEMP1
2919  013442  052737  000100  014444          BIS     #BIT06,TEMP1            ;SET BIT 6
2920  013450  013737  014444  017174  83$:    MOV     TEMP1,DATSNT            ;SET  DATA SENT
2921  013456  000412                          BR      99$
2922  013460  005737  014446          81$:    TST     TRACK           ;FIRST OR SECOND LOOP?
2923  013464  001432                          BEQ     13$             ;SECOND, THEN 13$
2924  013466  062737  000001  014444          ADD     #BIT0,TEMP1     ;INC NEW PATTERN
2925  013474  032737  000100  014444          BIT     #BIT6,TEMP1     ;DONE?
2926  013502  001413                          BEQ     14$             ;YES, THEN 14$, NEXT LOOP
2927  013504  013777  014444  166310  99$:    MOV     TEMP1,@ICMOD    ;MOVE OUT NEXT PATTERN
2928  013512  042777  001000  166304          BIC     #TBMTEN,@ICSR   ;TOGGLE TBMT ENABLE TO ALLOW
2929  013520  052777  001000  166276          BIS     #TBMTEN,@ICSR   ;MICRO CODE TO PROGRESS
2930  013526  000137  012640                  JMP     1$              ;LOOP
2931  013532  005037  014444          14$:    CLR     TEMP1           ;CLR PATTERN
2932  013536  005037  014446                  CLR     TRACK           ;INDICATE SECOND LOOP
2933  013542  052737  000200  014444          BIS     #BIT7,TEMP1     ;CREATE NEXT PATTERN
2934  013550  000755                          BR      99$             ;SEND PATTERN
2935  013552  062737  000001  014444  13$:    ADD     #BIT0,TEMP1     ;INCREMENT PATTERN
2936  013560  032737  000100  014444          BIT     #BIT6,TEMP1     ;DONE?
2937  013566  001746                          BEQ     99$             ;NO, THEN SEND NEXT PATTERN
2938  013570  042777  021000  166226  72$:    BIC     #TBMTEN+MAINT2,@ICSR    ;YES,, CLEAR TBMTEN AND MAINT2 TO EXIT MICROCODE
2939
2940  013576  005037  017206                  CLR     INTFLG
2941  013602  005077  166216                  CLR     @ICSR
2942  013606  052777  040000  166210          BIS     #MAINT3,@ICSR
2943  013614  005037  017202                  CLR     TIMTRP
2944  013620  104415                          CKCNC
2945  013622  005337  014456                  DEC     PASRUN
2946  013626  001402                          BEQ     .+6
```

```
2947  013630  000137  011400                    JMP     LOOP3
2948
2949
2950  013634  017700  166166                    MOV     @ICAR,R0
2951  013640  032777  000040  165270            BIT     #SW5,@SWR
2952  013646  001002                            BNE     .+6
2953  013650  000137  013676                    JMP     90$
2954  013654  017700  165256                    MOV     @SWR,R0
2955  013660  042700  177477                    BIC     #177477,R0
2956  013664  022700  000100                    CMP     #100,R0
2957  013670  001002                            BNE     .+6
2958  013672  000137  011364                    JMP     TEST3L
2959                                    ;
2960                                    ;CHECK THAT A MOV INSTRUCTION TO AN ICR OUTPUT MODULE WILL SET
2961                                    ;OUTPUT BUSY
2962                                    ;
2963
2964
2965
2966
2967  013676  012737  013676  014372  90$:    MOV     #90$,SCOLOP
2968  013704  013700  014414                    MOV     FR2@0,R0
2969  013710  005777  166110          70$:    TST     @ICSR           ;IS OUTPUT BUSY
2970  013714  100003                            BPL     .+10            ;IF NO CONT
2971  013716  005300                            DEC     R0              ;INC WATCHDOG
2972  013720  001373                            BNE     70$             ;NOT TIMED OUT THEN 70$
2973
2974                                    ;
2975                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2976                                    ;
2977  013722  104035                            ERROR   35              ;LINE ACTIVE TOO LONG
2978
2979                                    ;
2980                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2981                                    ;
2982  013724  104412                            SCOPEX
2983  013726  013700  014412                    MOV     FR1@0,R0
2984  013732  052777  000040  166064            BIS     #TTYEN,@ICSR    ;SET TTY ENABLE
2985  013740  012777  000000  166054            MOV     #0,@ICMOD
2986  013746  005777  166052          98$:    TST     @ICSR
2987  013752  100403                            BMI     97$
2988  013754  005300                            DEC     R0
2989  013756  001373                            BNE     98$
2990
2991                                    ;
2992                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2993                                    ;
2994  013760  104032                            ERROR 32                ;LINE DID NOT GO ACTIVE WITH MOV INSTRUCTION
2995
2996                                    ;
2997                                    ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
2998                                    ;
2999  013762  104412          97$:    SCOPEX
3000  013764  042777  000040  166032            BIC     #TTYEN,@ICSR    ;CLEAR TTY ENABLE
3001
3002                                    ;
```

```
                                        ;CHECK THAT BMT WILL NOT INTERRUPT WITH LINE BUSY
                                        ;
                                        BMTINT:
013772                                  
013772  012737  013772  014372  128:    MOV     #128,SCOLOP     ; SET SCOPE LOOP
014000  005077  166020                  CLR     @ICSR           ;CLEAR ICR CSR
014004  013700  014406                  MOV     FR1000,R0
014010  005777  166010          70$:    TST     @ICSR           ;IS OUTPUT BUSY
014014  100003                          BPL     .+10            ;IF NO CONT
014016  005300                          DEC     R0              ;INC WATCHDOG
014020  001373                          BNE     70$             ;NOT TIMED OUT THEN 70$
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
014022  104035                          ERROR   35                      ;OUTPUT BUSY STUCK
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
014024  104412                          SCOPEX
014026  005037  017206                  CLR     INTFLG          ;CLEAR INTERRUPT FLAG
014032  052777  000040  165764          BIS     #TTYEN,@ICSR    ;SET TTY ENABLE
014040  012777  000000  165754          MOV     #0,@ICMOD       ;CAUSE LINE TO GO ACTIVE
014046  042777  000040  165750          BIC     #TTYEN,@ICSR    ;CLEAR TTY ENABLE
                                                                ;WAIT FOR IT TO GO ACTIVE
014054  013700  014412                  MOV     FR100,R0
014060  005777  165740          80$:    TST     @ICSR
014064  100403                          BMI     .+10
014066  005300                          DEC     R0
014070  001373                          BNE     80$
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
014072  104032                          ERROR   32              ;OUTPUT BUSY WOULD NOT GO ACTIVE
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
014074  013700  014416                  MOV     FR20,R0
014100  013737  017172  177776          MOV     PRILVL,@#PSW    ;LOWER PSW
014106  052777  000010  165710          BIS     #BMTEN,@ICSR    ;ALLOW INTERRUPT IF PENDING
014114  005300                          DEC     R0
014116  001376                          BNE     .-2
014120  042777  000010  165676          BIC     #BMTEN,@ICSR    ;CLEAR BMT ENABLE
014126  012737  000340  177776          MOV     #340,@#PSW
014134  005737  017206                  TST     INTFLG          ;INTERRUPT OCCUR
014140  001401                          BEQ     11$             ;NO, OKAY THEN GO TO 11$
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
014142  104024                          ERROR 24                ;BMT INTERRUPTED WITH LINE BUSY
```

```
                                        ;
                                        ;/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
                                        ;
014144  104412                          118:    SCOPEX
014146  005037  017206                          CLR     INTFLG
014152  000137  005554                          JMP     TEST2
                                        ;
                                        ;
                                        ;
                                        ;
                                                .SBTTL  SCOPE LOOP ROUTINES
                                        ;SCOPE LOOP ROUTINE FOR M8094 MICRO DIAGNOSTICS
                                        ;
014156  104414                          XSCO94: CKPWF
014160  032777  040000  164750                  BIT     #SW14,@SWR      ;SW 14 SET
014166  001001                                  BNE     .+4
014170  000002                                  RTI                     ;NO EXIT
014172  011637  014232                          MOV     (SP),SCTMP      ;GET SCOPE LOCATION
014176  162737  000004  014232                  SUB     #4,SCTMP        ;SUB 4 TO LOOK LIKE ERR LOC
014204  023737  001116  014232                  CMP     $ERRPC,SCTMP    ;EQUAL TO ERR PC
014212  001401                                  BEQ     .+4
014214  000002                                  RTI
                                                POP2SP                  ;FIX STACK
014216  022626
014220  052777  040000  165576                  BIS     #MAINT3,@ICSR   ;RESET 94
014226  000137  011364                          JMP     TEST3L          ;LOOP
                                        SCTMP:  0
                                        ;
                                        ;SCOPE LOOP FOR M8096 MICRO DIAGNOSTICS
                                        ;
014234  104414                          XSCO96: CKPWF
014236  032777  040000  164672                  BIT     @SW14,@SWR      ;SW 14 SET
014244  001001                                  BNE     .+4
014246  000002                                  RTI                     ;NO EXIT
014250  022626                                  POP2SP                  ;FIX STACK
014252  000177  000114                          JMP     @SCOLOP         ;LOOP
                                        ;
                                        ;SCOPE LOOP FOR TTY TEST
                                        ;
014256  104414                          XSCTTY: CKPWF                   ;CHECK FOR TTY INPUT
014260  032777  040000  164650                  BIT     @SW14,@SWR      ;SW 14 SET
014266  001001                                  BNE     .+4
014270  000002                                  RTI                     ;NO EXIT
014272  011637  014232                          MOV     (SP),SCTMP      ;GET SCOPE LOCATION
014276  162737  000004  014232                  SUB     #4,SCTMP        ;SUB 4 TO LOOK LIKE ERR LOC
014304  023737  001116  014232                  CMP     $ERRPC,SCTMP    ;EQUAL TO ERR PC
014312  001401                                  BEQ     .+4
014314  000002                                  RTI
```

```
3115  014316  022626                        POP2SP                  ;FIX STACK
3116  014320  000177  000046                JMP     @SCOLOP         ;LOOP
3117
3118                                  ;
3119                                  ;REGULAR SCOPE LOOP FOR REMAINDER OF TEST
3120                                  ;
3121
3122  014324  104414                  XSCOPE: CKPWF
3123  014326  032777  040000  164602        BIT     #SW14,@SWR      ;SW 14 SET
3124  014334  001001                        BNE     .+4
3125  014336  000002                        RTI                     ;NO EXIT
3126  014340  011637  014232                MOV     (SP),SCTMP      ;GET SCOPE LOCATION
3127  014344  162737  000004  014232        SUB     #4,SCTMP        ;SUB 4 TO LOOK LIKE ERR LOC
3128  014352  023737  001116  014232        CMP     @ERRPC,SCTMP    ;EQUAL TO ERR PC
3129  014360  001401                        BEQ     .+4
3130  014362  000002                        RTI
3131
3132  014364  022626                        POP2SP                  ;FIX STACK
3133  014366  000177  000000                JMP     @SCOLOP         ;LOOP
3134
3135                                  ;
3136  014372  000000                  SCOLOP: 0
3137
3138
3139
3140  014374  000000                  TENCN1: 0
3141  014376  000000                  XMASK:  0                       ;MASK TO FIND BOX
3142  014400  000000                  FILUTS: 0                       ;FILE UNDER TEST
3143  014402  000000                  FILCNT: 0                       ;FILE COUNT
3144  014404  000000                  VECTOR: 0                       ;ICR VECTOR
3145  014406  000474                  FR1000: 316.                    ;TIME CONSTANTS
3146  014410  000236                  FR500:  158.                    ;
3147  014412  000037                  FR100:  31.                     ;
3148  014414  000077                  FR200:  63.                     ;
3149  014416  000006                  FR20:   6.                      ;
3150  014420  026502                  FRM110: 11586.                  ;
3151
3152  014422  000000                  FLAG1:  0                       ;SOFT FF
3153  014424  000000                  FIRST:  0                       ;SOFT FF
3154  014426  000000                  MICFLP: 0                       ;MICROCODE INDICATOR
3155  014430  000000                  FLAG:   0                       ;SOFT FF
3156  014432  000000                  HGH:    0                       ;HGH ADDRESS ON '96
3157  014434  000000                  LOW:    0                       ;LOW ADDRESS ON '96
3158  014436  000000                  SYSMAP: 0                       ;SYSTEM CONFIGARATION
3159  014440  000000                  ERRFLP: 0                       ;ERROR FF
3160  014442  000000                  ERTTL:  0                       ;ERROR TOTAL
3161  014444  000000                  TEMP1:  0                       ;TEMP STORAGE
3162  014446  000000                  TRACK:  0                       ;USED BY '98 LOOP
3163  014450  000000                  ADDR:   0                       ;USED BY '98 LOOP
3164  014452  000000                  PRIIND: 0                       ;USED FOR '96 RANGE
3165  014454  000000                  PASLG:  0                       ;PASS LENGTH
3166  014456  000000                  PASRUN: 0                       ;PASS COUNT (LOOP)
3167  014460  000000                  ERRLOP: 0                       ;ERROR FF
3168  014462  000000                  PRRMT:  0                       ;TTY REMOTE - DATA TO BE BUFFERED =1 YES  =0 NO
3169  014464  000000                  PRRMT1: 0                       ;TTY REMOTE - DATA ALLOWED AT ALL 1=YES 0=NO
3170  014466  000000                  REMPRE: 0                       ;REMOTE FF
```

```
3171  014470  000000                  TTYTMP: 0                       ;TIMER
3172  014472  000000                  REMOTE: 0                       ;OPERATOR SW REG ERROR FF
3173  014474  000000                  PWRFLP: 0                       ;PWR FAIL FF
3174  014476  000000                  TEMP:   0                       ;TEMP STORAGE
3175
3176
3177  014500  000000                  EDISPT: 0
3178
3179                                  ;;*****************************************************************************
3180
3181                                  .SBTTL  TYPE ROUTINE
3182
3183                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE, MESSAGE MUST TERMINATE WITH A 0 BYTE.
3184                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
3185                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
3186                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
3187                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
3188                                  ;*
3189                                  ;*CALL:
3190                                  ;*1) USING A TRAP INSTRUCTION
3191                                  ;*      TYPE    ,MESADR                 ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
3192                                  ;*OR
3193                                  ;*      TYPE
3194                                  ;*      MESADR
3195                                  ;*
3196                                  ;*2) USING A JSR INSTRUCTION
3197                                  ;*      MOV     PS,-(SP)                ;PUSH PROCESSOR STATUS WORD ON THE STACK
3198                                  ;*      JSR     PC,$TYPE                ;CALL TYPE ROUTINE
3199                                  ;*      MESSADR                         ;FIRST ADDRESS OF MESSAGE
3200
3201  014502  105737  001155          $TYPE:  TSTB    $TPFLG                  ;IS THERE A TERMINAL?
3202  014506  100002                          BPL     1$                      ;BR IF YES
3203  014510  000000                          HALT                            ;HALT HERE IF NO TERMINAL
3204  014512  000407                          BR      3$                      ;LEAVE
3205  014514  010046                  1$:     MOV     R0,-(SP)                ;SAVE R0
3206  014516  017600  000002          9$:     MOV     @2(SP),R0               ;GET ADDRESS OF ASCIZ STRING
3207  014522  112046                  2$:     MOVB    (R0)+,-(SP)             ;PUSH CHARACTER TO BE TYPED ONTO STACK
3208  014524  001005                          BNE     4$                      ;BR IF IT ISN'T THE TERMINATOR
3209  014526  005726                          TST     (SP)+                   ;IF TERMINATOR POP IT OFF THE STACK
3210  014530  012600                          MOV     (SP)+,R0                ;RESTORE R0
3211  014532  062716  000002          3$:     ADD     #2,(SP)                 ;ADJUST RETURN PC
3212  014536  000002                          RTI                             ;RETURN
3213  014540  004737  014572          4$:     JSR     PC,7$                   ;GO TYPE THIS CHARACTER
3214  014544  123726  001154          5$:     CMPB    $FILLC,(SP)+            ;IS IT TIME FOR FILLER CHARS.?
3215  014550  001364                          BNE     2$                      ;IF NO GO GET NEXT CHAR.
3216  014552  013746  001152                  MOV     $NULL,-(SP)             ;GET # OF FILLER CHARS. NEEDED
3217                                                                          ;AND THE NULL CHAR.
3218  014556  105366  000001          6$:     DECB    1(SP)                   ;DOES A NULL NEED TO BE TYPED?
3219  014562  002770                          BLT     5$                      ;BR IF NO--GO POP THE NULL OFF OF STACK
3220  014564  004737  014572                  JSR     PC,7$                   ;GO TYPE A NULL
3221  014570  000772                          BR      6$                      ;LOOP
3222  014572  122766  000216  000002  7$:     CMPB    #216,2(SP)
3223  014600  001412                          BEQ     9$
3224  014602  122766  000217  000002          CMPB    #217,2(SP)
3225  014610  001406                          BEQ     9$
3226  014612  105777  164330          99$:    TSTB    @$TPS                   ;WAIT UNTIL PRINTER IS READY
```

```
 3227  014616  100375                        BPL     99$
 3228  014620  116677  000002  164322        MOVB    2(SP),@STPB             ;LOAD CHAR TO BE TYPED INTO DATA REG.
 3229  014626  032777  002000  164302  98$:  BIT     #BIT10,@SWR
 3230  014634  001507                        BEQ     6$
 3231  014636  005737  014464                TST     PRRMT1                  ;REMOTE TTY ALLOWED AT THIS TIME
 3232  014642  001504                        BEQ     8$
 3233  014644  005737  014462                TST     PRRMT                   ;SHOULD WE BUFFER?
 3234  014650  001412                        BEQ     10$                     ;NO, CONT
 3235  014652  005237  017774                INC     BUFMES                  ;INC FOR CORRECT STORAGE
 3236  014656  022737  021747  017774        CMP     #BUFEND-1,BUFMES        ;AT END?
```

```
3237  014664  003473                        BLE     8$                      ;YES DONE!
3238  014666  116677  000002  003100        MOVB    2(SP),@BUFMES           ;BUFFER
3239  014674  000467                        BR      8$                      ;CONT
3240  014676  017746  165122          10$:  MOV     @ICSR,-(SP)             ;SAVE OLD ICSR
3241  014702  052777  001040  165114        BIS     #TBMTEN+TTYEN,@ICSR     ;SET TTY ENABLE, TBMT READ ENABLE
3242  014710  016677  000004  165104        MOV     4(SP),@ICMOD            ;SEND DATA TO ICR TTY
3243  014716  013737  014420  014470        MOV     FRM110,TTYTMP
3244  014724  005037  002062                CLR     ERRFLR                  ;CLEAR ERROR COUNT
3245  014730  032777  010000  165066  13$:  BIT     #ERRBIT,@ICSR           ;ERROR BIT SET
3246  014736  001413                        BEQ     14$                     ;NO, THEN 14$
3247  014740  005237  002062                INC     ERRFLR                  ;LOG ERROR
3248  014744  005777  165056                TST     @ICAR                   ;CLEAR ERROR
3249  014750  022737  000012  002062        CMP     #10.,ERRFLR             ;TEN (10) CONSECUTIVE LINE ERRORS
3250  014756  001364                        BNE     13$                     ;NO, THEN CHECK LINE AGAIN
3251  014760  022626                        POP2SP                          ;ADJUST STACK
3252  014762  000137  016474                JMP     ERRLIN
3253  014766  005037  002062          14$:  CLR     ERRFLR                  ;CLEAR ERROR COUNT
3254  014772  032777  002000  165024        BIT     #PWRFL,@ICSR            ;IS PWR FAIL BIT SET
3255  015000  001403                        BEQ     15$                     ;NO, THEN 15$
3256  015002  022626                        POP2SP
3257  015004  000137  016530                JMP     RSTRT
3258  015010  032777  100000  165010  15$:  BIT     #XTBMT,@ICAR            ;TRANSMITTOR BUFFER EMPTY
3259  015016  001014                        BNE     12$
3260  015020  005337  014470                DEC     TTYTMP
3261  015024  001341                        BNE     13$
3262  015026  005037  014464                CLR     PRRMT1
3263  015032  022626                        POP2SP
3264  015034  012600                        MOV     (SP)+,R0
3265  015036  022626                        POP2SP
3266  015040  104413  024676                TYPE,   TBMTMS
3267  015044  000000                        HALT
3268  015046  000776                        BR      .-2
3269  015050  012677  164750          12$:  MOV     (SP)+,@ICSR             ;SEND OLD CSR
3270  015054  000207                  8$:   RTS     PC
3271
3272                                        ;;*****************************************************************************
3273
3274
3275                                        ;
3276
3277
3278                                        ;;*****************************************************************
3279
3280                                        .SBTTL  ERROR HANDLER ROUTINE
3281
3282                                        ;*SW15=1        HALT ON ERROR
3283                                        ;*SW13=1        INHIBIT ERROR TYPEOUTS
3284                                        ;;GO TO $ERRTYP ON TYPEOUTS
3285
3286  015056                                SERROR:
3287  015056  105237  001103          7$:   INCB    $ERFLG                  ;SET THE ERROR FLG
3288  015062  001775                        BEQ     7$                      ;DON'T LET FLAG GO TO ZERO
3289  015064  013737  001102  001140        MOV     $TSTNM,@#DISPLAY        ;DISPLAY TEST NUMBER AND ERROR
3290  015072  005264  000002                INC     ERRTOT(4)               ;INC THE ERROR COUNT
3291  015076  011637  001116                MOV     (SP),$ERRPC             ;GET ADDRESS OF ERROR
3292  015102  162737  000002  001116        SUB     #2,$ERRPC
```

```
3293  015110  117737  164002  001114          MOVB    @SERRPC,SITEMB          ;STRIP AND SAVE THE ITEM CODE
3294  015116  032777  020000  164012          BIT     #SW13,@SWR              ;SKIP TYPEOUT IF SET
3295  015124  001016                          BNE     2$
3296  015126  005237  014424                  INC     FIRST                   ;HAVE WE PRINTED FILE BOX ASSOCIATED WITH ERROR
3297  015132  001007                          BNE     1$                      ;YES CONTINUE
3298  015134  104413  024357                  TYPE,   FILET
3299  015140  013746  014400                  MOV     FILUTS,-(SP)    ;;SAVE FILUTS FOR TYPEOUT
3300  015144  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3301  015146  104413  022000                  TYPE    ,CRLF
3302  015152  004737  015174          1$:     JSR     PC,@#SERRTYP
3303  015156  104413  022000                  TYPE,   CRLF
3304  015162  005777  163750          2$:     TST     @SWR
3305  015166  100001                          BPL     3$
3306  015170  000000                          HALT
3307  015172  000002                  3$:     RTI
3308
3309
3310                                  ;;*****************************************************************
3311
3312                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
3313
3314  015174                          SERRTYP:
3315  015174  005037  017204                  CLR     #RKFLP
3316  015200  104413  022000                  TYPE    ,CRLF                   ;"CARRIAGE REURN" & "LINEFEED"
3317  015204  010046                          MOV     R0,-(SP)                ;SAVE R0
3318  015206  005000                          CLR     R0                      ;PICKUP THE ITEM INDEX
3319  015210  153700  001114                  BISB    @#SITEMB,R0
3320  015214  001004                          BNE     1$
3321                                                                          ;IF ITEM NUMBER IS ZERO, JUST
3322                                                                          ;TYPE THE PC OF THE ERROR
3323  015216  013746  001116                  MOV     SERRPC,-(SP)    ;;SAVE SERRPC FOR TYPEOUT
3324                                                                  ;;ERROR ADDRESS
3325  015222  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3326  015224  000433                          BR      6$
3327  015226  005300                  1$:     DEC     R0                      ;ADJUST THE INDEX SO THAT IT WILL
3328  015230  006300                          ASL     R0                      ;WORK FOR THE ERROR TABLE
3329  015232  006300                          ASL     R0
3330  015234  006300                          ASL     R0
3331  015236  062700  001162                  ADD     #SERRTB,R0              ;FORM TABLE POINTER
3332  015242  012037  015252                  MOV     (R0)+,2$                ;PICKUP "ERROR MESSAGE" POINTER
3333  015246  001411                          BEQ     3$                      ;SKIP TYPEOUT IF NO POINTER
3334  015250  104413                          TYPE                            ;TYPE THE ERROR MESSAGE
3335  015252  000000                  2$:     .WORD   0                       ;ERROR MESSAGE POINTER GOES HERE
3336  015254  104413  024305                  TYPE,   PCPRT                   ;PRINT ERRPC
3337  015260  013746  001116                  MOV     SERRPC,-(SP)    ;;SAVE SERRPC FOR TYPEOUT
3338                                                                  ;;ERROR PC
3339  015264  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3340  015266  104413  022000                  TYPE    ,CRLF                   ;"CARRIAGE RETURN" "LINEFEED"
3341  015272  012037  015302          3$:     MOV     (R0)+,4$                ;PICK UP "DATA HEADER" POINTER
3342  015276  001404                          BEQ     5$                      ;SKIP TYPEOUT IF ZERO
3343  015300  104413                          TYPE                            ;TYPE THE "DATA HEADER"
3344  015302  000000                  4$:     .WORD   0                       ;"DATA HEADER" POINTER GOES HERE
3345  015304  104413  022000                  TYPE    ,CRLF                   ;"CARRIAGE RETURN" "LINEFEED"
3346  015310  011000                  5$:     MOV     (R0),R0                 ;PICKUP "DATA TABLE POINTER
3347  015312  001004                          BNE     7$                      ;GO TYPE THE DATA
3348  015314  012600                  6$:     MOV     (SP)+,R0                ;RESTORE R0
```

```
3349  015316  104413  022000                  TYPE    ,CRLF                   ;TYPE CARRIAGE RETURN LF
3350  015322  000207                          RTS     PC
3351  015324                          7$:
3352  015324  013046                          MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
3353  015326  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3354  015330  005710                          TST     (R0)                    ;IS THERE ANOTHER NUMBER
3355  015332  001770                          BEQ     6$
3356  015334  104413  015342                  TYPE    ,8$                     ;TYPE 2 SPACES
3357  015340  000771                          BR      7$
3358  015342  020040      000         8$:     .ASCIZ  /  /
3359
3360          015346                          .EVEN
3361
3362                                  ;*******************************************************************
3363
3364                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3365
3366                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3367                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
3368                                  ;*STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3369                                  ;*CALL:
3370                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3371                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
3372                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3373                                  ;*      .BYTE   M               ;;M=1 OR 0
3374                                  ;*                                      ;;1=TYPE LEADING ZEROS
3375                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
3376                                  ;*
3377                                  ;*STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3378                                  ;*STYPOS OR STYPOC
3379                                  ;*CALL:
3380                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3381                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
3382                                  ;*
3383                                  ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3384                                  ;*CALL:
3385                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3386                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
3387
3388  015346  017646  000000          STYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
3389  015352  116637  000001  015571          MOVB    1(SP),$0FILL    ;;LOAD ZERO FILL SWITCH
3390  015360  112637  015573                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
3391  015364  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
3392  015370  000406                          BR      STYPON
3393  015372  112737  000001  015571  STYPOC: MOVB    #1,$0FILL       ;;SET THE ZERO FILL SWITCH
3394  015400  112737  000006  015573          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
3395  015406  112737  000005  015570  STYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
3396  015414  010346                          MOV     R3,-(SP)        ;;SAVE R3
3397  015416  010446                          MOV     R4,-(SP)        ;;SAVE R4
3398  015420  010546                          MOV     R5,-(SP)        ;;SAVE R5
3399  015422  113704  015573                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
3400  015426  005404                          NEG     R4
3401  015430  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
3402  015434  110437  015572                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
3403  015440  113704  015571                  MOVB    $0FILL,R4       ;;GET THE ZERO FILL SWITCH
3404  015444  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
```

```
 3405  015450  005003                        CLR     R3              ;;CLEAR THE OUTPUT WORD
 3406  015452  006105                1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 3407  015454  000404                        BR      3$              ;;GO DO MSB
 3408  015456  006105                2$:     ROL     R5              ;;FORM THIS DIGIT
 3409  015460  006105                        ROL     R5
 3410  015462  006105                        ROL     R5
 3411  015464  010503                        MOV     R5,R3
 3412  015466  006103                3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 3413  015470  105337  015572                DECB    $OMODE          ;;TYPE THIS DIGIT?
 3414  015474  100016                        BPL     7$              ;;BR IF NO
 3415  015476  042703  177770                BIC     #177770,R3      ;;GET RID OF JUNK
 3416  015502  001002                        BNE     4$              ;;TEST FOR 0
 3417  015504  005704                        TST     R4              ;;SUPPRESS THIS 0?
 3418  015506  001403                        BEQ     5$              ;;BR IF YES
 3419  015510  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 3420  015512  052703  000060                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 3421  015516  052703  000040        5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 3422  015522  110337  015566                MOVB    R3,8$           ;;SAVE FOR TYPING
 3423  015526  104413  015566                TYPE    ,8$             ;;GO TYPE THIS DIGIT
 3424  015532  105337  015570        7$:     DECB    $OCNT           ;;COUNT BY 1
 3425  015536  003347                        BGT     2$              ;;BR IF MORE TO DO
 3426  015540  002402                        BLT     6$              ;;BR IF DONE
 3427  015542  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 3428  015544  000744                        BR      2$              ;;GO DO THE LAST DIGIT
 3429  015546  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
 3430  015550  012604                        MOV     (SP)+,R4        ;;RESTORE R4
 3431  015552  012603                        MOV     (SP)+,R3        ;;RESTORE R3
 3432  015554  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 3433  015562  012616                        MOV     (SP)+,(SP)
 3434  015564  000002                        RTI                     ;;RETURN
 3435  015566     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 3436  015567     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 3437  015570     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 3438  015571     000                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 3439  015572  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 3440                                ;*****************************************************************
 3441
 3442                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 3443
 3444                                ;+THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 3445                                ;+SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 3446                                ;+NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 3447                                ;+BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 3448                                ;+REPLACED WITH SPACES.
 3449                                ;+CALL:
 3450                                ;+      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 3451                                ;+      TYPDS                   ;;GO TO THE ROUTINE
 3452
 3453  015574                        $TYPDS:
 3454  015574  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 3455  015576  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 3456  015600  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 3457  015602  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 3458  015604  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 3459  015606  012746  020200                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 3460  015612  016605  000020                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
```

```
3461  015616  100004                          BPL     1$              ;;BR IF INPUT IS POS.
3462  015620  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
3463  015622  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
3464  015630  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
3465  015632  012703  016010                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
3466  015636  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
3467  015642  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
3468  015644  016001  016000                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
3469  015650  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
3470  015652  002402                          BLT     4$              ;;BR IF DONE
3471  015654  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
3472  015656  000774                          BR      3$
3473  015660  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
3474  015662  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
3475  015664  001002                          BNE     5$              ;;FALL THROUGH IF 0
3476  015666  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
3477  015670  100407                          BMI     7$              ;;BR IF YES
3478  015672  106316                  5$:     ASLB    (SP)            ;;MSD?
3479  015674  103003                          BCC     6$              ;;BR IF NO
3480  015676  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
3481  015704  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
3482  015710  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
3483  015714  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
3484  015716  005720                          TST     (R0)+           ;;JUST INCREMENTING
3485  015720  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
3486  015724  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
3487  015726  003002                          BGT     8$              ;;GO TO EXIT
3488  015730  010502                          MOV     R5,R2           ;;GET THE LSD
3489  015732  000764                          BR      6$              ;;GO CHANGE TO ASCII
3490  015734  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
3491  015736  100003                          BPL     9$              ;;BR IF NO
3492  015740  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
3493  015746  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
3494  015750  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
3495  015752  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
3496  015754  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
3497  015756  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
3498  015760  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
3499  015762  104413  016010                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
3500  015766  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
3501  015774  012616                          MOV     (SP)+,(SP)
3502  015776  000002                          RTI                     ;;RETURN TO USER
3503  016000  023420                  $DTBL:  10000.
3504  016002  001750                          1000.
3505  016004  000144                          100.
3506  016006  000012                          10.
3507  016010  000004                  $DBLK:  .BLKW   4
3508                                  ;*****************************************************************
3509
3510                                  .SBTTL  POWER DOWN AND UP ROUTINES
3511
3512                                  ;POWER DOWN ROUTINE
3513  016020  012737  016146  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
3514  016026  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
3515  016034  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
3516  016036  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
```

```
3517  016040  010246                        MOV     R2,-(SP)          ;;PUSH R2 ON STACK
3518  016042  010346                        MOV     R3,-(SP)          ;;PUSH R3 ON STACK
3519  016044  010446                        MOV     R4,-(SP)          ;;PUSH R4 ON STACK
3520  016046  010546                        MOV     R5,-(SP)          ;;PUSH R5 ON STACK
3521  016050  010637  016152                MOV     SP,$SAVR6         ;;SAVE SP
3522  016054  012737  016066  000024        MOV     #$PWRUP,@#PWRVEC  ;;SET UP VECTOR
3523  016062  000000                        HALT
3524  016064  000776                        BR      .-2               ;;HANG UP
3525
3526                                ;POWER UP ROUTINE
3527  016066  013706  016152        $PWRUP: MOV     $SAVR6,SP         ;;GET SP
3528  016072  005037  016152                CLR     $SAVR6            ;;WAIT LOOP FOR THE TTY
3529  016076  005237  016152        1$:     INC     $SAVR6            ;;WAIT FOR THE INC
3530  016102  001375                        BNE     1$                ;;OF  WORD
3531  016104  012605                        MOV     (SP)+,R5          ;;POP STACK INTO R5
3532  016106  012604                        MOV     (SP)+,R4          ;;POP STACK INTO R4
3533  016110  012603                        MOV     (SP)+,R3          ;;POP STACK INTO R3
3534  016112  012602                        MOV     (SP)+,R2          ;;POP STACK INTO R2
3535  016114  012601                        MOV     (SP)+,R1          ;;POP STACK INTO R1
3536  016116  012600                        MOV     (SP)+,R0          ;;POP STACK INTO R0
3537  016120  012737  016020  000024        MOV     #$PWRDN,@#PWRVEC  ;;SET UP THE POWER DOWN VECTOR
3538  016126  012737  000340  000026        MOV     #340,@#PWRVEC+2   ;;PRIO:7
3539  016134  104413                        TYPE                      ;;REPORT THE POWER FAILURE
3540  016136  023060                $PWRMG: .WORD   MES39             ;;POWER FAIL MESSAGE POINTER
3541  016140  012716                        MOV     (PC)+,(SP)        ;;RESTART AT START1
3542  016142  003002                $PWRAD: .WORD   START1            ;;RESTART ADDRESS
3543  016144  000002                        RTI
3544  016146  000000                $ILLUP: HALT                      ;;THE POWER UP SEQUENCE WAS STARTED
3545  016150  000776                        BR      .-2               ;; BEFORE THE POWER DOWN WAS COMPLETE
3546  016152  000000                $SAVR6: 0                         ;;PUT THE SP HERE
3547
3548
3549
3550                                ;*****************************************************************
3551
3552                                .SBTTL  TRAP DECODER
3553
3554                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3555                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3556                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
3557                                ;*GO TO THAT ROUTINE.
3558
3559  016154  010046                $TRAP:  MOV     R0,-(SP)          ;;SAVE R0
3560  016156  016600  000002                MOV     2(SP),R0          ;;GET TRAP ADDRESS
3561  016162  005740                        TST     -(R0)             ;;BACKUP BY 2
3562  016164  111000                        MOVB    (R0),R0           ;;GET RIGHT BYTE OF TRAP
3563  016166  006300                        ASL     R0                ;;POSITION FOR INDEXING
3564  016170  016000  016176                MOV     $TRPAD(R0),R0     ;;INDEX TO TABLE
3565  016174  000200                        RTS     R0                ;;GO TO ROUTINE
3566
3567
3568                                .SBTTL  TRAP TABLE
3569
3570                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3571                                ;*BY THE "TRAP" INSTRUCTION.
3572
```

```
                                        ;       ROUTINE
                                        ;       -------
016176                                  STRPAD:
016176  014502                                  STYPE   ;;CALL*TYPE     TRAP*0(104400)  TTY TYPEOUT ROUTINE
016200  015372                                  STYPOC  ;;CALL*TYPOC    TRAP*1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
016202  015346                                  STYPOS  ;;CALL*TYPOS    TRAP*2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
016204  015406                                  STYPON  ;;CALL*TYPON    TRAP*3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
016206  015574                                  STYPDS  ;;CALL*TYPDS    TRAP*4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)
016210  017372                                  XSAVRE  ;;CALL*SAVREG   TRAP*5(104405)  ROUTINE TO SAVE REGISTER
016212  017446                                  XRESRG  ;;CALL*GETREG   TRAP*6(104406)  ROUTINE TO RESTORE REGISTERS
016214  014156                                  XSCO94  ;;CALL*SCOP94   TRAP*7(104407)  M8094 MICRO CODE SCO94 ROUTINE
016216  014234                                  XSCO96  ;;CALL*SCOP96   TRAP*10(104410) M8096 MICRO CODE SCOPE ROUTINE
016220  014256                                  XSCTTY  ;;CALL*SCOTTY   TRAP*11(104411) TTY SCOPE ROUTINE
016222  014324                                  XSCOPE  ;;CALL*SCOPEX   TRAP*12(104412) GENERAL SCOPE ROUTINE
016224  014502                                  STYPE   ;;CALL*TYPE     TRAP*13(104413) TTY & REMOTE TTY TYPE ROUTINE
016226  016232                                  XCKPWF  ;;CALL*CKPWF    TRAP*14(104414) PWR FAIL SENSE ROUTINE
016230  016302                                  XCKCNC  ;;CALL*CKCNC    TRAP*15(104415) CNTRL C CHECK


                                        .SBTTL  EXIT REQUEST ROUTINES(POWER FAIL AND CONTROL C)


016232  032777  002000  162676  XCKPWF: BIT     @BIT10,@SWR             ;IS BIT 10 SET
016240  001001                          BNE     .+4                     ;YES CONTINUE
016242  000002                          RTI                             ;NO, RETURN
016244  005737  017202                  TST     TIMTRP
016250  001401                          BEQ     .+4
016252  000002                          RTI
016254  032777  002000  163542          BIT     @PWRFL,@ICSR            ;IS POWER FAIL BIT SET
016262  001001                          BNE     .+4                     ;YES, CONTINUE
016264  000002                          RTI                             ;NO, EXIT
016266  022626                          POP2SP
016270  052777  040000  163526  XCKPW1: BIS     @MAINT3,@ICSR           ;ISSUE "RESET"
016276  000137  016530                  JMP     RSTRT                   ;GO AND RESTART PROGRAM


016302  032777  002000  162626  XCKCNC: BIT     @SW10,@SWR
016310  001006                          BNE     3$
016312  105777  162624                  TSTB    @STKS
016316  001403                          BEQ     3$
016320  017700  162620                  MOV     @STKB,R0
016324  000423                          BR      4$
016326  032777  002000  162602  3$:     BIT     @BIT10,@SWR             ;IS BIT 10 SET
016334  001001                          BNE     .+4                     ;YES, CONTINUE
016336  000002                          RTI                             ;NO, EXIT
016340  013700  014420                  MOV     FRM110,R0
016344  005300                          DEC     R0
016346  001376                          BNE     .-2
016350  032777  010000  163450          BIT     @DA,@ICAR               ;REMOTE TTY DATA AVAILABLE
016356  001001                          BNE     .+4                     ;YES CONTINUE
016360  000002                          RTI                             ;NO,EXIT
016362  052777  000041  163434          BIS     #TTYEN*XRIF,@ICSR               ;ENABLE TTY READ
016370  017700  163426                  MOV     @ICMOD,R0               ;READ
016374  042700  000200          4$:     BIC     #200,R0
016400  042777  000040  163416          BIC     #TTYEN,@ICSR            ;CLEAR READENABLE
```

```
3629  016406  022700  000003                    CMP     #3,R0           ;WAS IT A CNTL C
3630  016412  001416                            BEQ     1$              ;YES SERVICE IT
3631  016414  022700  000020                    CMP     #20,R0          ;CNTRL P
3632  016420  001401                            BEQ     2$              ;YES, SERVICE IT
3633  016422  000002                            RTI                     ;NO, EXIT
3634  016424  012737  000001  014464  2$:       MOV     #1,PRRMT1
3635  016432  005037  014462                    CLR     PRRMT
3636  016436  104413  025132                    TYPE,   CNTNLP
3637  016442  022626                            POP2SP
3638  016444  000137  010446                    JMP     PRTSUM
3639  016450  012737  000001  014464  1$:       MOV     #1,PRRMT1       ;SET UP TO PRINT TO REMOTE
3640  016456  005037  014462                    CLR     PRRMT
3641  016462  104413  025140                    TYPE,   CNTRLC          ;ECHO ^C
3642  016466  022626                            POP2SP                  ;ADJUST STACK
3643  016470  000137  010452                    JMP     WAIT2
3644
3645  016474  032777  002000  163322  ERRLIN:  BIT     @PWRFL,@ICSR    ;IS PWR FAIL BIT ALSO SET
3646  016502  001402                            BEQ     .+6             ;NO, THEN IT MUST BE A BAD LINE
3647  016504  000137  016530                    JMP     RSTRT           ;RESTART
3648  016510  005037  014464                    CLR     PRRMT1
3649  016514  005037  014462                    CLR     PRRMT
3650  016520  104413  025367                    TYPE,   NOISY
3651  016524  000137  016544                    JMP     RST4
3652
3653
3654
3655
3656
3657  016530  005037  014464          RSTRT:   CLR     PRRMT1          ;INHIBIT TYPEOUT TO REMOTE TTY
3658  016534  005037  014462                    CLR     PRRMT
3659  016540  104413  025161                    TYPE,   PWRMES          ;PRINT POWER FAIL SENSED
3660  016544  052777  040000  163252  RST4:    BIS     @MAINT3,@ICSR   ;ISSUE RESET
3661  016552  013700  014420                    MOV     FRM110,R0
3662  016556  005300                            DEC     R0
3663  016560  001376                            BNE     .-2
3664  016562  012700  000002          RST3:    MOV     #2.,R0          ;SET UP DELAY
3665  016566  005001                            CLR     R1
3666  016570  032777  010000  163226  RST2:    BIT     @ERRBIT,@ICSR   ;ERROR BIT SET
3667  016576  001416                            BEQ     RST1            ;NO, CONT
3668  016600  005777  163222                    TST     @ICAR           ;CLEAR ERROR
3669  016604  013702  014414                    MOV     FR200,R2
3670  016610  005302                            DEC     R2
3671  016612  001376                            BNE     .-2
3672  016614  005301                            DEC     R1
3673  016616  001364                            BNE     RST2
3674  016620  005300                            DEC     R0
3675  016622  001362                            BNE     RST2
3676  016624  104413  025212                    TYPE,   ICRDWN
3677  016630  000137  016562                    JMP     RST3
3678  016634  052777  040000  163162  RST1:    BIS     @MAINT3,@ICSR
3679  016642  012737  000001  014464            MOV     #1,PRRMT1
3680  016650  012737  000000  014462            MOV     #0,PRRMT
3681  016656  013700  014420                    MOV     FRM110,R0
3682  016662  005300                            DEC     R0
3683  016664  001376                            BNE     .-2
3684  016666  005737  014474                    TST     PWRFLP          ;WHEN ANSWERING QUES DID
```

```
                                                                        ;PWR FAIL CAUSE US TO GET HERE
016672  001404                          BEQ     1$                      ;NO, THEN RESTART DIAGNOSTIC
016674  005037  014474                  CLR     PWRFLP                  ;CLEAR INDICATOR
016700  000137  010452                  JMP     WAIT2                   ;REASK QUESTION
016704  104413  025076          1$:     TYPE,   RSTMES
016710  000137  003002                  JMP     START1
                                .SBTTL  ICR-11 INTERRUPT SERVICE ROUTINE
                        ;
016714  005237  017206          ICRSRV: INC     INTFLG                  ;INDICATE INTERRUPT
016720  005737  014426                  TST     MICFLP                  ;IN MICRO CODE SECTION?
016724  001401                          BEQ     .+4                     ;NO, 4$
016726  000002                          RTI                             ;YES, EXIT
016730  005737  017176                  TST     TTYFLP                  ;IN TTY SECTION
016734  001001                          BNE     .+4                     ;YES CONT
016736  000002                          RTI                             ;NO,EXIT
016740  032777  010000  163056          BIT     @ERRBIT,@ICSR           ;DID ERROR CAUSE INTERRUPT
016746  001405                          BEQ     1$                      ;NO, THEN CHECK FOR OTHERS
016750  017700  163052                  MOV     @ICAR,R0                ;REMOVE ERROR
016754  005264  000004                  INC     ERRCNT(4)               ;RECORD ERROR
016760  000002                          RTI                             ;RETURN FROM ERROR INTERRUPT
016762  052777  000001  163034  1$:     BIS     #XRIF,@ICSR             ;SET RIF BIT TO CLEAR INTERRUPT
016770  032777  000200  163026          BIT     #MODINT,@ICSR           ;MODULE INTERRUPT THE CAUSE
016776  001416                          BEQ     2$                      ;NO, THEN 2$.....BAD
017000  032777  010000  163020          BIT     @DA,@ICAR               ;DID DA CAUSE IT
017006  001016                          BNE     3$                      ;YES, THEN GO SERVICE DA AT 3$
017010  032777  100000  163010          BIT     @XTBMT,@ICAR            ;WAS IT TBMT
017016  001040                          BNE     4$                      ;YES THEN GO SERVICE TBMT AT 4$
                        ;
                        ;IT WASN'T DA OR TBMT THAT SET MOD INT SO IT MUST HAVE BENN A
                        ;REAL MODULE INTERRUPT, INDICATE IT SO
                        ;
017020  005777  162776                  TST     @ICMOD                  ;READ TO CLEAR RIF BIT AND INTERRUPT
017024  017700  162776                  MOV     @ICAR,R0
017030  104063                          ERROR   63                      ;UNEXPECTED MODULE INTERRUPT, INPUT PROBABLY
017032  000401                          BR      99$
017034  104064                  2$:     ERROR   64                              ;SPURIOUS ICR INTERRUPT
017036  022626                  99$:    POP2SP
017040  000137  007620                  JMP     TTY2
                        ;
                        ;DA WAS SET   READ IN DATA AND COMPARE
                        ;
```

```
3741  017044  017737  162752  001126  38$:    MOV     @ICMOD,$BDDAT           ;READ
3742  017052  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
3743  017060  001401                          BEQ     68
3744
3745  017062  104036                          ERROR   36                      ;ERROR BAD READ
3746
3747  017064  000240                  68$:    NOP
3748  017066  005237  001124                  INC     $GDDAT
3749  017072  005037  017174                  CLR     DATSNT
3750  017076  032737  000400  001124          BIT     #400,$GDDAT
3751  017104  001405                          BEQ     48
3752  017106  022626                  78$:    POP2SP
3753  017110  017700  162712                  MOV     @ICAR,R0
3754  017114  000137  007710                  JMP     TTYEND
3755
3756                                  ;
3757                                  ;
3758                                  ;TBMT TRUE   SEND NEW PATTERN
3759                                  ;
3760
3761  017120  005737  017174          48$:    TST     DATSNT
3762  017124  001015                          BNE     88
3763  017126  013700  014410                  MOV     FR500,R0
3764
3765  017132  005777  162666                  TST     @ICSR
3766  017136  100003                          BPL     .+10
3767  017140  005300                          DEC     R0
3768  017142  001373                          BNE     .-10
3769
3770  017144  104035                          ERROR 35
3771
3772  017146  013777  001124  162646          MOV     $GDDAT,@ICMOD
3773  017154  005237  017174                  INC     DATSNT
3774  017160  017700  162642          88$:    MOV     @ICAR,R0
3775  017164  022626                          POP2SP
3776  017166  000137  007654                  JMP     TTY1
3777
3778
3779  017172  000000                  PRILVL: 0
3780  017174  000000                  DATSNT: 0
3781  017176  000000                  TTYFLP: 0
3782  017200  000000                  BLINK:  0
3783  017202  000000                  TIMTRP: 0
3784  017204  000000                  WRKFLP: 0
3785  017206  000000                  INTFLG: 0
3786
3787
3788                                  ;
3789
3790                                  ;ICR INFORMATION FOR EACH ICR FILE UNDER TEST
3791
3792  017210  000000                  INFLST: .WORD   0
3793
3794  017212  000070                          .BLKW   56.
3795
3796                                  ;SAVE REGISTERS
```

```
                                        ;
017372  012637  017530          XSAVREG: MOV     (SP)+,SAVEPC
017376  012637  017526                   MOV     (SP)+,SAVPSW
017402  012637  017524                   MOV     (SP)+,SAV2PC
017406  012637  017522                   MOV     (SP)+,SAV2SW
017412  010146                           MOV     R1,-(SP)
017414  010246                           MOV     R2,-(SP)
017416  010346                           MOV     R3,-(SP)
017420  010446                           MOV     R4,-(SP)
017422  010546                           MOV     R5,-(SP)
017424  013746  017522                   MOV     SAV2SW,-(SP)
017430  013746  017524                   MOV     SAV2PC,-(SP)
017434  013746  017526                   MOV     SAVPSW,-(SP)
017440  013746  017530                   MOV     SAVEPC,-(SP)
017444  000002                           RTI
                                        ;
                                        ;RESTORE REGISTERS
                                        ;
017446  012637  017530          XRESRG:  MOV     (SP)+,SAVEPC
017452  012637  017526                   MOV     (SP)+,SAVPSW
017456  012637  017524                   MOV     (SP)+,SAV2PC
017462  012637  017522                   MOV     (SP)+,SAV2SW
017466  012605                           MOV     (SP)+,R5
017470  012604                           MOV     (SP)+,R4
017472  012603                           MOV     (SP)+,R3
017474  012602                           MOV     (SP)+,R2
017476  012601                           MOV     (SP)+,R1
017500  013746  017522                   MOV     SAV2SW,-(SP)
017504  013746  017524                   MOV     SAV2PC,-(SP)
017510  013746  017526                   MOV     SAVPSW,-(SP)
017514  013746  017530                   MOV     SAVEPC,-(SP)
017520  000002                           RTI
017522  000000          SAV2SW:  0
017524  000000          SAV2PC:  0
017526  000000          SAVPSW:  0
017530  000000          SAVEPC:  0

017532  000000          ADDTAB:  0

        017774                   .=.+168.

017774  000000          BUFMES:  0
017776  000000          BUFBEG:  0

        021750                   .=.+1000.

021750  005015  051105  047522  BUFEND:  .ASCII  <15><12>/ERROR BUFFER FULL/
021756  020122  052502  043106
021764  051105  043040  046125
021772     114
021773     201                           .BYTE   201
021774  100601          BUFFIN:  100601

                                        .EVEN
```

```
                                        0
                                ;
                                        .SBTTL  ASCII MESSAGES
                                ;

022000  006415  000012          CRLF:   .ASCIZ  <15><15><12>
022004  006415  044412  051103  VECMD:  .ASCIZ  <15><15><12>/ICR VECTOR ADDRESS= /
022034  006415  044412  052116  PRIOR:  .ASCIZ  <15><15><12>/INTERRUPT PRIORITY LEVEL= /
022072  006415  046412  042117  VECMES: .ASCIZ  <15><15><12>/MOD. INT. WILL NOT INT - ABORTED/
022136  051503  020122  044502  EM1:    .ASCIZ  /CSR BIT 15 SET/
022155     122  051505  052105  EM2:    .ASCIZ  /RESET BIT SET/
022173     124  054524  042440  EM3:    .ASCIZ  /TTY ENA NOT SET/
022213     124  054524  042440  EM4:    .ASCIZ  /TTY ENA SET/
022227     015  005015  040515  M8094:  .ASCIZ  <15><15><12>/MAINTENANCE MODE ACTIVATED/
022265     124  047515  052125  EM5:    .ASCIZ  /TMOUT FLP NOT SET/
022307     123  050505  041440  EM6:    .ASCIZ  /SEQ COMPARE ERR/
022327     124  046502  020124  EM7:    .ASCIZ  /TBMT INT ERR/
022344  051105  047522  042516  EM10:   .ASCIZ  /ERRONEOUS INT/
022362  047516  044440  052116  EM11:   .ASCIZ  /NO INT FROM ERR/
022402  047111  020124  042523  EM12:   .ASCIZ  /INT SET AFT RIF/
022422  051105  020122  042523  EM13:   .ASCIZ  /ERR SET AFT CLR/
022442  040504  040524  053040  EM14:   .ASCIZ  /DATA VALID TRUE/
022462  041124  052115  047040  EM15:   .ASCIZ  /TBMT NOT SET/
022477     116  020117  041124  EM16:   .ASCIZ  /NO TBMT INT/
022513     116  020117  046502  EM17:   .ASCIZ  /NO BMT INT/
022526  006415  050012  051501  PASS:   .ASCIZ  <15><15><12>/PASS COUNT = /
022547     015  043012  046111  FILE:   .ASCIZ  <15><12>/FILE BOX /
022563     040  041511  051101  ICARMD: .ASCIZ  / ICAR=/
022572  044440  051503  036522  ICSRMD: .ASCIZ  / ICSR=/
022601     015  047012  020117  NOBOX:  .ASCIZ  <15><12>/NO ICR'S - ABORTED/<15><12>
022630  006416  005015  042524  TESTIN: .ASCIZ  <16><15><15><12>/TESTING FILE BOX /
022656  005015  041511  030522  HEADER: .ASCII  <15><12>/ICR11 CONTROLLER DIAGNOSTIC MAINDEC-11-DZIRA-A/<15><12>
022740  005015  025052  025052          .ASCIZ  <15><12>/**********ICR11 SYSTEM MAP**********/<15><12>
023010  005015  005015  025052  STARMS: .ASCIZ  <15><12><15><12>/**********************************/
023060  006415  051012  041505  MES39:  .ASCIZ  <15><15><12>/RECOVERED FROM POWER FAILURE /<15><15><12>
023124  005015  047111  052520  RESULT: .ASCII  <15><12>/INPUT MODULE ADDRESS RANGE/<15><12>
023162  052123  051101  020124          .ASCIZ  /START   /
023173     015  005015  044506  FINMES: .ASCIZ  <15><15><12>/FINISH  /
023207     040  020040  000040  BLANK:  .ASCIZ  /    /
023214  047111  052520  020124  EM20:   .ASCIZ  /INPUT ADDR ERR/
023233     116  020117  047515  EM21:   .ASCIZ  /NO MOD INT/
023246  052517  050124  052125  EM22:   .ASCIZ  /OUTPUT ADDR ERR/
023266  052124  020131  042523  EM23:   .ASCIZ  /TTY SEND RECEIVE ERROR/
023315     102  052115  044440  EM24:   .ASCIZ  /BMT INT ERR/
023331     115  046125  020124  EM25:   .ASCIZ  /MULT INT ERR/
023346  052515  052114  044440  EM26:   .ASCIZ  /MULT INT OCC/
023363                          EM27:
023363                          EM30:
023363     111  052116  020122  EM31:   .ASCIZ  /INTR ENA ERR/
023400  047516  020124  052502  EM32:   .ASCIZ  %NOT BUSY W/ MOV%
023420  040504  047040  052117  EM33:   .ASCIZ  /DA NOT SET ON TTY TEST/
        023233                  EM34=EM21
023447     117  052125  041040  EM35:   .ASCIZ  /OUT BUSY STUCK/
023466  052124  020131  047111  EM36:   .ASCIZ  /TTY INTERRUPT SEND RECEIVE ERROR/
023527     106  042515  042440  EM37:   .ASCIZ  /PME ERROR/
```

```
023541     106  042515  047040  EM40:   .ASCIZ  /FME NO TIMEOUT/
023560  046123  040440  045503  EM41:   .ASCIZ  /SL ACK ON BD CRC/
023601     106  042523  042440  EM42:   .ASCIZ  /FSE ERROR/
023613     106  042523  052040  EM43:   .ASCIZ  /FSE TIMEOUT/
023627     106  042523  047040  EM44:   .ASCIZ  /FSE NO ERROR/
023644  044524  042515  052517  EM45:   .ASCIZ  /TIMEOUT ERROR/
023662  040504  040524  053040  EM46:   .ASCIZ  /DATA VALID ERROR/
023703     111  040503  020122  EM47:   .ASCIZ  /ICAR ERROR/
023716  047515  020104  042101  EM50:   .ASCIZ  /MOD ADDR ERROR/
023735     124  047105  041440  EM51:   .ASCIZ  /TEN CONSECUTIVE LINE ERRORS IN MODULE RANGE TEST/
024016  042524  020116  047503  EM52:   .ASCIZ  /TEN CONSECUTIVE LINE ERRORS IN MODULE WRAP AROUND TEST/
024105     124  046502  020124  EM53:   .ASCIZ  /TBMT TTY TRANS/
024124  041124  052115  052040  EM54:   .ASCIZ  /TBMT TTY TIME/
024142  044522  020106  044502  EM55:   .ASCIZ  /RIF BIT NOT SET/
024162  044522  020106  044502  EM56:   .ASCIZ  /RIF BIT SET/
024176  052124  020131  047111  EM57:   .ASCIZ  /TTY INTERRUPT TEST HUNG/
024226  040515  047111  020124  EM60:   .ASCIZ  /MAINT BIT 3 ON/
024245     120  051127  043040  EM61:   .ASCIZ  /PWR FAIL BIT ON/
024265     111  051503  020122  EM62:   .ASCIZ  /ICSR REG, ERROR/
024305     040  042440  051122  PCPRT:  .ASCIZ  /  ERRPC /
024316  006415  051412  052520  EM64:   .ASCIZ  <15><15><12>/SPUR ICR INT/
024336  006415  052412  042516  EM63:   .ASCIZ  <15><15><12>/UNEXP MOD INT/
024357     015  005015  044506  FILET:  .ASCIZ  <15><15><12>/FILE BOX IN ERROR /
024405     015  005015  025012  RUNSUM: .ASCIZ  <15><15><12><12>/*RUN SUMMARY - FILE /
024436  006452  000012          RUNCON: .ASCIZ  /*/<15><12>
024442  040520  051523  041440  PASMES: .ASCIZ  /PASS COUNT      /
024463     015  005015  051105  ERMES:  .ASCIZ  <15><15><12>/ERROR COUNT     /
024507     015  005015  044514  TRERR:  .ASCIZ  <15><15><12>/LINE ERROR COUNT/
024533     015  005015  047516  NOEXST: .ASCIZ  <15><15><12>/NON-EXISTENT FILE BOX/
024564  020040     000          SPACE:  .ASCIZ  /  /
024567     015  005015  044123  NOPLUG: .ASCIZ  <15><15><12>/SHORTING PLUG NOT IN REMOTE END - TTY TEST ABORTED/<15><12>
024657     015  005015  047514  LOADIN: .ASCIZ  <15><15><12>/LOADING..../
024676  006415  051012  046505  TBMTMS: .ASCIZ  <15><15><12>/REMOTE TTY HUNG--DESELECT IT/
024736  006415  051012  051105  WAITIN: .ASCIZ  <15><15><12>/RERUN OR LOAD FIELD TEST?  /
024775     015  005015  047503  CONTOP: .ASCIZ  <15><15><12>/CONT BY OPERATOR...LOOPING/<15><12>
025035     015  005015  042522  TSTTTY: .ASCIZ  <15><15><12>/REMOTE TTY WRAP AROUND TEST/<15><12>
025076  006416  005015  005015  RSTMES: .ASCIZ  <16><15><15><12><15><12>/RESTARTING DIAGNOSTIC/
025132  006415  057012  000120  CNTRLP: .ASCIZ  <15><15><12>/^P/
025140  006415  057012  000103  CNTRLC: .ASCIZ  <15><15><12>/^C/
025146  006415  052012  051505  WRKMES: .ASCIZ  <15><15><12>/TESTING/
025161     015  005015  041511  PWRMES: .ASCIZ  <15><15><12>/ICR POWER FAIL SENSED/
025212  006415  044412  051103  ICRDWN: .ASCIZ  <15><15><12>%ICR POWER OFF/LINE OPEN%
025245     015  005015  041511  PWRMS1: .ASCII  <15><15><12>/ICR POWER FAIL TEST/
025273     015  050012  053517          .ASCIZ  <15><12>/POWER DOWN REMOTE END OF ICR/
025332  005015  047520  042527  PWRMS2: .ASCIZ  <15><12>/POWER REMOTE END BACK UP/<15><12>
025367     015  005015  047516  NOISY:  .ASCIZ  <15><15><12>%NOISY/OPEN LINE%


3859                                            .EVEN
3860
3861  025412  006415  000012          DMB:    .ASCIZ  <15><15><12>
3862  025416  054105  023520  020104  DMI:    .ASCIZ  /EXP'D   REC'D/
3863  025424  020040  042522  023503
3864  025432  000104
```

```
 3865  025434  047515  052504  042514  DM2:    .ASCII  /MODULE   DATA   EXP'D   REC'D/<15><15><12>
 3866  025442  020040  042040  052101
 3867  025450  020101  020040  054105
 3868  025456  023520  020104  020040
 3869  025464  042522  023503  006504
 3870  025472  005015
 3871  025474  042101  051104  051505          .ASCIZ  /ADDRESS  SENT   CONTS   CONTS/
 3872  025502  020123  051440  047105
 3873  025510  020124  020040  047503
 3874  025516  052116  020123  020040
 3875  025524  047503  052116  000123
 3876  025532  042040  052101  020101  DM3:    .ASCII  / DATA    EXP'D    REC'D/<15><15><12>
 3877  025540  020040  042440  050130
 3878  025546  042047  020040  020040
 3879  025554  042522  023503  006504
 3880  025562  005015
 3881  025564  051440  047105  020124          .ASCIZ  / SENT    CONTS    CONTS/
 3882  025572  020040  041440  047117
 3883  025600  051524  020040  020040
 3884  025606  047503  052116  000123
 3885
 3886                                          .EVEN
 3887
 3888  025614  000000                  DM4:    0
 3889
 3890  025616  000000                  DT0:    0
 3891  025620  001124  001126  000000  DT1:    $GDDAT,$BDDAT,0,0
 3892  025626  000000
 3893  025630  014450  014444  001124  DT2:    ADDR,TEMP1,$GDDAT,$BDDAT,0
 3894  025636  001126  000000
 3895  025642  014444  001124  001126  DT3:    TEMP1,$GDDAT,$BDDAT,0
 3896  025650  000000
 3897  025652  000000                  DT4:    0
 3898
 3899
 3900
 3901
 3902
 3903
 3904          033000                          .=33000
 3905
 3906                                          .SBTTL  FIELD TEST LOAD ROUTINES/ABS LOADER
 3907
 3908
 3909
 3910                                  ;
 3911                                  ;ROUTINE TO LOAD FIELD TEST
 3912                                  ;
 3913  033000  052777  040000  147816  GOLOAD: BIS     #MAINT3,@ICSR   ;RESET
 3914  033006  013737  002024  033154          MOV     ICSR,SVCSR      ;SAVE PRESENT CSR IN CASE OF CKSUM ERROR
 3915  033014  013737  002026  033156          MOV     ICAR,SVCAR
 3916  033022  013737  014404  033152          MOV     VECTOR,SVVEC
 3917  033030  013737  002022  033160          MOV     ICMOD,SVMOD
 3918  033036  013737  001136  033150          MOV     SWR,TSWR
 3919
 3920  033044  000137  033232                  JMP     L01
```

```
                                        ;
                                        ;PRINT BAD CHECKSUM MESSAGE
                                        ;
033050  010046                  CKSMBD: MOV     R0,-(SP)        ;SAVE R0
033052  010146                          MOV     R1,-(SP)        ;SAVE R1
033054  012700  033166                  MOV     #CKMES,R0       ;GET MESSAGE
033060  112001                  3$:     MOVB    (0)+,R1         ;GET CHAR
033062  001426                          BEQ     2$              ;ZERO TERMINATOR
033064  010177  000072                  MOV     R1,@TPB
033070  105777  000070          1$:     TSTB    @TPS
033074  100375                          BPL     1$
033076  032777  002000  000044          BIT     #SW10,@TSWR
033104  001765                          BEQ     3$
033106  052777  001040  000040          BIS     #TBMTEN+TTYEN,@SVCSR
033114  032777  100000  000034          BIT     #XTBMT,@SVCAR
033122  001774                          BEQ     .-6
033124  010177  000030                  MOV     R1,@SVMOD
033130  042777  001040  000016          BIC     #TTYEN+TBMTEN,@SVCSR
033136  000758                          BR      3$
033140  012601                  2$:     MOV     (SP)+,R1
033142  012600                          MOV     (SP)+,R0
033144  000137  033246                  JMP     LD2

033150  000000                  TSWR:   0
033152  000000                  SVVEC:  0
033154  000000                  SVCSR:  0
033156  000000                  SVCAR:  0
033160  000000                  SVMOD:  0
033162  177566                  TPB:    177566
033164  177564                  TPS:    177564

033166  006415  041012  042101  CKMES:  .ASCIZ  <15><15><12>/BAD CHECKSUM/
033174  041440  042516  045503
033202  052523  000115

                                        .EVEN

                                        ;
                                        ;PDP-11 ABSOLUTE BINARY LOADER -- V005A  MODIFIED FOR THE
                                        ;PURPOSE OF LOADING ICR11 FIELD TEST PROGRAM (MD-11-DZIRB)
                                        ;REMOTELY. LOADER WILL SELF START THE FIELD TEST IF THE CHECKSUM
                                        ;IS OK . IF THERE IS A CHECKSUM ERROR, IT WILL INDICATE SO ON BOTH
                                        ;LOCAL AND REMOTE TTYS.

                                        ;DEFINITIONS FOR THE ABS LOADER ONLY

000000                          CKSM=   %0              ;CHECKSUM IS KEPT IN R0
000001                          ADR=    %1              ;LOAD ADDRESS
000002                          BC=     %2              ;BYTE COUNT
000003                          BYT=    %3              ;CONTENTS OF BYTE
000004                          R4=     %4              ;SCRATCH
000005                          PTR=    %5              ;READ SUBR PNTR
000006                          SP=     %6              ;STACK POINTER
```

```
                    000007                  PC=     %7              ;PROGRAM COUNTER
                    000200                  LOAD=200
                    017776                  DEV=17776
                    000162                  ICRVEC=162              ;VECTOR FOR FIELD TEST
                    000164                  REMFF=164               ;REMOTE LOAD INDICATOR
                    000166                  ICARLD=166              ;ICAR FOR FIELD TEST
                    000170                  ICSRLD=170              ;ICSR FOR FIELD TEST
                    000172                  ICSMLD=172              ;MODULE ADDRESS FOR FIELD TEST


033206  000012                      .BLKW   10.

033232  010706          LD1:        MOV     PC,SP   ;SET STACK POINTER
033234  024646                      CMP     -(SP),-(SP)             ;TO START AT LD1-4
033236  012705  033326              MOV     #READ,PTR
033242  005001                      CLR     ADR                     ;CLEAR LOAD ADDRESS
033244  005016          LD1B:       CLR     @SP                     ;USING ADDRESS ON TAPE

                                    ;
                                    ;LOOK FOR THE BEGINNING OF A BLOCK
                                    ;

033246  005000          LD2:        CLR     CKSM                    ;INIT CHECK SUM
033250  004715                      JSR     PC,@PTR                 ;CHECK FOR 1 (START OF BLOCK)
033252  105303                      DECB    BYT                     ;LOOP UNTIL 1 FOUND
033254  001374                      BNE     LD2
033256  004715                      JSR     PC,@PTR                 ;READ ANOTHER FRAME

                                    ;
                                    ;INPUT AND SAVE BYTE COUNT, IF BYTE COUNT IS 6 GO PROCESS JUMP
                                    ;

033260  004737  033356                  JSR     PC,GWRD             ;GET FUUL BYTE COUNT
033264  010402                          MOV     R4,BC
033266  162702  000004                  SUB     #4,BC               ;SUB 4 TO BYTE COUNT CORRECT
033272  022702  000002                  CMP     #2,BC               ;WAS IT SIX (6)
033276  001440                          BEQ     JMP1                ;JUMP BLOCK
033300  004737  033356                  JSR     PC,GWRD             ;GET LOAD ADDRESS
033304  061604                          ADD     @SP,R4              ;ACTUAL ADDRESS
033306  010401                          MOV     R4,ADR              ;PUT ADDRESS IN PROPER CELL

                                    ;
                                    ;READ IN REMAINDER OF TAPE
                                    ;IF LOADER GOES TO "BAD" A CHECKSUM ERROR HAS OCCURRED
                                    ;

033310  004715          LD3:        JSR     PC,@PTR                 ;READ A FRAME
033312  002003                      BGE     LD4                     ;BR IF MORE DATA REMAINS
033314  105700                      TSTB    CKSM                    ;CHECKSUM CORRECT
033316  001753                      BEQ     LD2
033320  000464          BAD:        BR      BDCKSM                  ;CHECK SUM ERROR

033322  110321          LD4:        MOVB    BYT,(ADR)+              ;STORE 8 BITS
```

```
4033  033324  000771                          BR      LD3
4034
4035                                  ;
4036                                  ;INPUT A FRAME, DECREMENT BYTE COUNT, AND ACCUMULATE CHECKSUM
4037                                  ;
4038
4039  033326  012703  177550          READ:   MOV     #177550,BYT             ;DEVICE ADDRESS TO BYT
4040  033332  105213                          INCB    @BYT                    ;READER RUN
4041  033334  105713                  R12:    TSTB    @BYT
4042  033336  100376                          BPL     R12
4043  033340  116303  000002                  MOVB    2(BYT),BYT
4044  033344  060300                          ADD     BYT,CKSM
4045  033346  042703  177400                  BIC     #177400,BYT
4046  033352  005302                          DEC     BC
4047  033354  000207                          RTS     PC
4048
4049                                  ;
4050                                  ;ASSEMBLE ONE FULL WORD OF DATA
4051                                  ;
4052
4053  033356  012637  033426          GWRD:   MOV     (SP)+,TMP
4054  033362  004715                          JSR     PC,@PTR
4055  033364  010304                          MOV     BYT,R4
4056  033366  004715                          JSR     PC,@PTR
4057  033370  000303                          SWAB    BYT
4058  033372  050304                          BIS     BYT,R4
4059  033374  013707  033426                  MOV     TMP,PC
4060
4061                                  ;
4062                                  ;CHECK CORRECTNESS OF JUMP ADDRESS
4063                                  ;
4064
4065  033400  004737  033356          JMP1:   JSR     PC,GWRD
4066  033404  004715                          JSR     PC,@PTR
4067  033406  105700                          TSTB    CKSM
4068  033410  001343                          BNE     BAD
4069  033412  006204                          ASR     R4
4070  033414  103002                          BCC     JMP11
4071  033416  000404                          BR      FLDST
4072  033420  000711                          BR      LD1B
4073  033422  006204                  JMP11:  ASR     R4
4074  033424  000114                          JMP     @R4
4075
4076  033426  000000                  TMP:    .WORD   0
4077
4078  033430  012737  000001  000164  FLDST:  MOV     #1,REMFF                ;SETUP ICSR,VECTOR,ICAR FOR FIELD TEST
4079  033436  013737  033154  000170          MOV     SVCSR,ICSRLD
4080  033444  013737  033156  000166          MOV     SVCAR,ICARLD
4081  033452  013737  033160  000172          MOV     SVMOD,ICSMLD
4082  033460  013737  033152  000162          MOV     SVVEC,ICRVEC
4083  033466  000137  000200                  JMP     LOAD                    ;GO START FIELD TEST
4084  033472  000137  033050          BDCKSM: JMP     CKSMBD
4085
4086          033476                  SENDAD=.
4087          000001                          .END
```

```
ADDR    014450
ADDTAB  017532
ADDWRP  006434
ADR   =%000001
ADWR1   006364
ADWR2   006372
BAD     033320
BC    =%000002
BDCKSM  033472
BITLST  003776
BITTST  003656
BIT0  = 000001
BIT00 = 000001
BIT01 = 000002
BIT02 = 000004
BIT03 = 000010
BIT04 = 000020
BIT05 = 000040
BIT06 = 000100
BIT07 = 000200
BIT08 = 000400
BIT09 = 001000
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BLANK   023207
BLINK   017200
BMTEN = 000010
BMTINT  013772
BMTTST  004574
BOXCNT  002052
BPTVEC= 000014
BUFBEG  017776
BUFEND  021750
BUFFIN  021774
BUFMES  017774
BYT   =%000003
CKCNC = 104415
CKMES   033166
CKPWF = 104414
CKSM  =%000000
CKSMBD  033050
CNTRLC  025140
CNTRLP  025132
CONTOP  024775
CONTST  002776
CONTSW  004146
CRLF    022000
DA    = 010000
DATSNT  017174
DDISP = 177570
DEV   = 017776
DH0     025412
DH1     025416
DH2     025434
DH3     025532
DH4     025614
DISPLA  001140
DISPRE  000174
DSWR  = 177570
DT0     025616
DT1     025620
DT2     025630
DT3     025642
DT4     025652
EDISPT  014500
EMTVEC= 000030
EM1     022136
EM10    022344
EM11    022362
EM12    022402
EM13    022422
EM14    022442
EM15    022462
EM16    022477
EM17    022513
EM2     022155
EM20    023214
EM21    023233
EM22    023246
EM23    023266
EM24    023315
EM25    023331
EM26    023346
EM27    023363
EM3     022173
EM30    023363
EM31    023363
EM32    023400
EM33    023420
EM34  = 023233
EM35    023447
EM36    023466
EM37    023527
EM4     022213
EM40    023541
EM41    023560
EM42    023601
EM43    023613
EM44    023627
EM45    023644
EM46    023662
EM47    023703
EM5     022265
EM50    023716
EM51    023735
EM52    024016
EM53    024105
EM54    024124
EM55    024142
EM56    024162
EM57    024176
EM6     022307
EM60    024226
EM61    024245
EM62    024265
EM63    024336
EM64    024316
EM7     022327
ENDPAS  010304
ENDPS1= 010336
ERMES   024463
ERRBIT= 010000
ERRCNT= 000004
ERREN = 000002
ERRFLP  014440
ERRFLR  002062
ERRLIN  016474
ERRLOP  014460
ERRTOT= 000002
ERRVEC= 000004
ERTTL   014442
FILCNT  014402
FILE    022547
FILET   024357
FILFND  002574
FILLST  002774
FILNFD  002710
FILUTS  014400
FIND11  002226
FINMES  023173
FIRST   014424
FLAG    014430
FLAG1   014422
FLOST   033430
FRM110  014420
FR100   014412
FR1000  014406
FR20    014416
FR200   014414
FR500   014410
GETREG= 104406
GOLOAD  033000
GOOF    002754
GWRD    033356
HEADER  022656
HGM     014432
ICAR    002026
ICARHD  022563
ICARLD= 000166
ICMOD   002022
ICRCNT  002034
ICRDWN  025212
ICRRUN  002030
ICRSRV  016714
ICRVEC= 000162
ICRWT   011116
ICRWT1  011242
ICRWT2  011264
ICRWT3  011212
ICSRLD= 000172
ICSR    002024
ICSRHD  022572
ICSRLD= 000170
INFLST  017210
INTFLG  017206
INTST   005034
INTST1  005234
IOTVEC= 000020
JMP1    033400
JMP11   033422
LD1     033232
LD10    033244
LD2     033246
LD3     033310
LD4     033322
LOAD  = 000200
LOADIN  024657
LOOP1 = 004410
LOOP2 = 005632
LOOP3 = 011400
LOW     014434
LPTST   003504
MAINTS  003576
MAINT0= 000400
MAINT1= 004000
MAINT2= 020000
MAINT3= 040000
MES39   023060
MFR10K  002040
MFR100  002046
MFR11K  002036
MFR20   002050
MFR200  002044
MFR500  002042
MICFLP  014426
MODEN = 000004
MODINT= 000200
M8094   022227
NOBOX   022601
NOEXST  024533
NOISY   025367
NOPLUG  024567
ONEBOX  003242
OUTBSY= 100000
PASCNT= 000000
PASLG   014454
PASMES  024442
PASRUN  014456
PASS    022526
PC    =%000007
PCPRT   024305
PERHGH= 000006
PERLOW= 000010
PIRQ  = 177772
PIRQVE= 000240
POP2SP= 022626
PRIIND  014452
PRILVL  017172
PRIOR   022034
PRMT    014462
PRMT1   014464
PRTMES  010720
PRTSUM  010446
PR0   = 000000
PR1   = 000040
PR2   = 000100
PR3   = 000140
PR4   = 000200
PR5   = 000240
PR6   = 000300
PR7   = 000340
PS    = 177776
PSW   = 177776
PTR   =%000005
PWFEN = 000020
PWFFLG  002032
PWRFL = 002000
PWRFLP  014474
PWRMES  025161
PWRMS1  025245
PWRMS2  025332
PWRVEC= 000024
```

```
READ    033326          STARTA  003044          TBMTEN= 001000          TYPON = 104403          $ERRTY  015174
REMFF = 000164          START1  003002          TBMTMS  024676          TYPOS = 104402          $ERTTL  001112
REMOTE  014472          START2  003144          TEMP    014476          VECFND  004150          $FILLC  001154
REMPRE  014466          STKLMT= 177774          TEMP1   014444          VECHD   022004          $FILLS  001153
RESULT  023124          SUMPRT  010120          TENCNT  002054          VECMES  022072          $GDADR  001120
RESVEC= 000010          SVCAR   033156          TENCN1  014374          VECRET  004244          $GDDAT  001124
RSTMES  025076          SVCSR   033154          TENTST  004346          VECTOR  014404          $HD   = 000003
RSTRT   016530          SVMOD   033160          TESTIN  022630          VECTST= 004014          $ICNT   001104
RST1    016634          SVVEC   033152          TEST1   003332          VECT1   004024          $ILLUP  016146
RST2    016570          SWLOOP= 013360          TEST2   005554          VECT2   004112          $ITEMB  001114
RST3    016562          SWR     001136          TEST3   011330          WAITIN  024736          $LF     001160
RST4    016544          SWREG   000176          TEST3A  011632          WAIT1   010514          $LPADR  001106
RUNCON  024436          SW0   = 000001          TEST3B  012222          WAIT2   010452          $LPERR  001110
RUNSUM  024405          SW00  = 000001          TEST3C  012344          WATROU  010254          $NULL   001152
RWBIT   003524          SW01  = 000002          TEST3D  012556          WRKFLP  017204          $OCNT   015570
R0    =%000000          SW02  = 000004          TEST3L  011364          WRKMES  025146          $OMODE  015572
R1    =%000001          SW03  = 000010          TIMTRP  017202          XCKCNC  016302          $PASS   001100
R12     033334          SW04  = 000020          TKVEC = 000060          XCKPWF  016232          $PWRAD  016142
R2    =%000002          SW05  = 000040          TMP     033426          XCKPW1  016270          $PWRDN  016020
R3    =%000003          SW06  = 000100          TPB     033162          XMASK   014376          $PWRMG  016136
R4    =%000004          SW07  = 000200          TPS     033164          XRESET= 000100          $PWRUP  016066
R5    =%000005          SW08  = 000400          TPVEC = 000064          XRESRG  017446          $QUES   001156
R6    =%000006          SW09  = 001000          TRACK   014446          XRIF  = 000001          $SAVR6  016152
R7    =%000007          SW1   = 000002          TRAPVE= 000034          XSAVRE  017372          $SETUP= 000016
SAVEPC  017530          SW10  = 002000          TRERR   024507          XSCOPE  014324          $STUP = 177777
SAVPSW  017526          SW11  = 004000          TRTVEC= 000014          XSCO94  014156          $SWR  = 160000
SAVREG= 104405          SW12  = 010000          TSTTTY  025035          XSCO96  014234          $TKB    001144
SAV2PC  017524          SW13  = 020000          TSWR    033150          XSCTTY  014256          $TKS    001142
SAV2SW  017522          SW14  = 040000          TTYEN = 000040          XTBMT = 100000          $TN   = 000001
SCOLOP  014372          SW15  = 100000          TTYEND= 007710          $BDADR  001122          $TPB    001150
SCOPEX= 104412          SW2   = 000004          TTYFLP  017176          $BDDAT  001126          $TPFLG  001155
SCOP94= 104407          SW3   = 000010          TTYINT  007574          $CMTAG  001100          $TPS    001146
SCOP96= 104410          SW4   = 000020          TTYL  = 007112          $CM3  = 000000          $TRAP   016154
SCOTTY= 104411          SW5   = 000040          TTYSTR  007032          $CRLF   001157          $TRP  = 000016
SCTMP   014232          SW6   = 000100          TTYTMP  014470          $DBLK   016010          $TRPAD  016176
SETPWF  002064          SW7   = 000200          TTYTST  007036          $DTBL   016000          $TSTNM  001102
SP    =%000006          SW8   = 000400          TTYWT   011316          SENDAD= 033476          $TYPDS  015574
SPACE   024564          SW9   = 001000          TTY1    007654          $ERFLG  001103          $TYPE   014502
SRTFF   002060          SW9FF   002056          TTY2    007620          $ERMAX  001115          $TYPOC  015372
STACK = 001100          SYSMAP  014436          TYPDS = 104404          $ERROR  015056          $TYPON  015406
STARMS  023010          SYSTST  010542          TYPE  = 104413          $ERRPC  001116          $TYPOS  015346
START   002070          TBITVE= 000014          TYPOC = 104401          $ERRTB  001162          $0FILL  015571
.     = 033476
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*DZIRAA,DZIRAA/SOL_DZIRAA
RUN-TIME: 36 24 1 SECONDS
RUN-TIME RATIO: 198/62=3.0
CORE USED: 16K (31 PAGES)