BB11-D

DMA INTERFACE

MD-11-DZKDA-A

EP-DZKDA-A-DL

FICHE 1 OF 1

JUN 1978

digital

MADE IN USA

IDENTIFICATION

PRODUCT CODE:	MAINDEC 11-DZKDA-A-D
PRODUCT NAME:	8811-D DMA INTERFACE
DATE CREATED:	APRIL 19, 1974
MAINTAINED BY:	DIAGNOSTIC ENGINEERING
AUTHOR:	MIKE STURAK

1. ABSTRACT

THIS PROGRAM CONTAINS VARIOUS TESTS DESIGNED TO COMPLETELY DIAGNOSE THE BB_{11}-D DIRECT MEMORY ACCCESS DEVICE WHEN STRUCTURED IN "LOOP BACK" MODE (DIAGNOSTIC CABLE FROM LOCATION D_1 ($M9_100$) J_1 TO EF_81 ($M15_82$) AND STARTED AT PROGRAM LOCATION 200.

WHEN THE BB11-D (KIT) IS IN "LOOP BACK" CONFIGURATION, THE FOLLOWING SIGNALS ARE EQUIVALENT.

(CSR)	SIGNAL NAME	(CSR)	SIGNAL NAME
9	LAST XFER	12	X_0F (EXTERNAL OVERFLOW)
2	FNCT	3	STATUS
11	DA OUT	10	INPUT (DA IN-NOT IN)
	DATA ACCEPTED	7	READY

THE FOLLOWING TESTS ARE PERFORMED UPON THE BB11-D, AND ARE LISTED IN THE ORDER IN WHICH THEY ARE EXECUTED (SA 200).

TEST 0 - " INIT "
TEST 2 - CSR BITS) 2 (FNCT), AND 3 (STAT)
TEST 3 - CSR BITS) 0 (IN), AND 7 (RDY)
TEST 5 - " DATA AVAILABLE OUT "
TEST 7 - DATA BUFFER REGISTER WRAP AROUND
TEST 8 - DATA OUTPUT (PART 1)
ONE NBR (OUT) W/O WORD COUNT OVERFLOW
TEST 9 - DATA OUTPUT (PART 2)
ONE NBR OUTPUT WITH WORD COUNT OVERFLOW
TEST 10 - INPUT
TEST 11 - OUTPUT/INPUT (SIMULTANEOUS)
TEST 12 - DATA VERIFICATION (PART 1)
TEST 13 - DATA VERIFICATION (PART 2)
TEST 14 - INTERRUPT ENABLE (PART 1) NO INTERRUPT EXPECTED
TEST 15 - INTERRUPT ENABLE (PART 2) FROM WORD COUNT OVERFLOW
TEST 16 - INTERRUPT ENABLE (PART 3) FROM SPECIAL CONDITION
TEST 17 - INTERRUPT ENABLE (PART 4) PRIORITY LEVEL OF KIT
TEST 18 - INTERRUPT ENABLE (PART 5) PRIORITY LEVEL 5 VERIFICATION.
TEST 19 - LAST TRANSFER - EXTERNAL OVERFLOW

THIS PROGRAM ALSO CONTAINS A SPECIAL CONVERSATION EXERCISE DESIGNED TO ASYNCHRONOUSLY TRANSFER DATA (OUT/IN) WHEN STARTED AT PROGRAM LOCATION 282.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11
BB11-D (KIT)
DIAGNOSTIC "LOOP BACK" CABLE

2.2 STORAGE

THIS PROGRAM REQUIRES 4K.

2.3 CONFIGURATION

2.3.1 CONFIGURATION FOR STARTING ADDRESS 200 (DIAGNOSTIC)

CONNECT THE DIAGNOSTIC "LOOP BACK" CABLE FROM BB11-D LOGIC LOCATION D1 (M9100) J1 TO $EF0_1$ (M1502). DISCONNECT ANY CABLE FROM D1 (M9100) J2.

2.3.2 CONFIGURATIONS FOR STARTING ADDRESS 202 (CONVERSATION EXERCISE)

(A). THE SAME AS PARAGRAPH 2, 3, 1, OR
(B). CONNECT TO LOGIC LOCATION OF "THIS" BB11-D, D1 (M9100) J2, THE CABLE FROM THE "OTHER" BB11-D EF01 (M1502). DISCONNECT ANY CABLE FROM D1 (M9100) J1 OF ALL BB11-D'S.

2.4 PRELIMINARY PROGRAMS

ALL DIAGNOSTICS FOR THE MAINTENANCE OF THE PDP-11 SHOULD HAVE BEEN RUN IN THEIR ENTIRETY SOME TIME PRIOR TO THE EXECUTION OF THIS DIAGNOSTIC.

3. LOADING PROCEDURE

THIS PROGRAM IS IN .ABS FORMAT. FOLLOW THE DIRECTIONS FOR THE PARTICULAR ABSOLUTE LOADER BEING USED.

4. STARTING PROCEDURE

STARTING ADDRESS 200 IS THE BB11-D DIAGNOSTIC.
STARTING ADDRESS 202 IS THE SPECIAL CONVERSATION EXERCISE.

(A) PLACE WITHIN PROGRAM LOCATION 1200 THE OCTAL EQUIVALENT OF THE BB11-D DEVICE ADDRESS (IF NOT 160000).

(B) CONFIGURE THE DATA SWITCHES TO REPRESENT THE OCTAL EQUIVALENT OF THE STARTING ADDRESS 200, OR 202.

(C) PRESS "LOAD ADDRESS". PRESS "START".

THE PROGRAM IS NOW RUNNING.

5. PROGRAM AND/OR OPERATOR ACTION

5.1 DATA SWITCH SETTINGS

15 = 0 - HALT UPON DETECTION OF ERROR
14 = 0 - SCOPE LOOP ON ERROR (LATCHED BY FIRST ERROR)
13 = 1 - HALT AT END OF EACH PROGRAM PASS (SA 200), OR
- HALT AT END OF RECIEVE DATA COMPARE (SA 202)
12 = 0 - PRINT ERROR INFORMATION AT DETECTION
11 = 1 - BURST MODE TRANSMISSIONS (SA 202)

5.2 MISCELLANEOUS PRINTOUTS

THE PROGRAM WILL PRINT ON THE TELEPRINTER THE INTRODUCTORY MESSAGE " TOGGLE INTO PROGRAM LOCATION XXXX THE BB11-D DEVICE ADDRESS 00 IF NOT 760500 " AFTER THE FIRST PROGRAM START 200.

THE PROGRAM WILL INFORM THE ATTENDANT OF THE BB11-D VECTOR ADDRESS BY PRINTING ON THE TELEPRINTER THE MESSAGE " BB11-D VECTOR ADDRESS IS XXXX " UPON FIRST DETECTION OF A BB11-D DEVICE INTERRUPT AFTER START 200, OR START 202.

THE PROGRAM WILL PRINT THE MESSAGE " WAITING FOR OTHER KIT " (STARTING ADDRESS 202) EACH TIME THE " OTHER " KIT BECOMES, OR FAILS TO BECOME READY. (STATUS BIT 03 = 1 INDICATES THE KIT IS READY).

THE "OTHER" KIT IS NOT READY WHEN THE DATA SWITCH 13=1, OR 15=1, OR WHEN A BB11-D ERROR HAS BEEN DETECTED AND DATA SWITCH 12=0.

5.3 NON RECOVERABLE PROGRAM HALTS

PROGRAM LOCATION	DESCRIPTION
2	UNASSIGNED TRAP
4	STACK POINTER OVERFLOW
12	RESERVED INSTRUCTION TRAP
14	TRACE TRAP
24	POWER FAILURE
32	EMT TRAP
60	UNEXPECTED TELEPRINTER INTERRUPT

5.4 RESTRICTIONS (APPLICABLE ONLY FOR STARTING ADDRESS 202)

THE PROGRAM WILL "HANG" WAITING FOR CSR BIT 7 (READY) IF THE HARDWARE CONFIGURATION IS THAT DESCRIBED WITHIN PARAGRAPH 2.3.2 (A), AND DATA SWITCH 11 (BURST MODE) IS POSITIONED TO REPRESENT A BINARY 1. THIS IS ALSO TRUE IF THE HARDWARE CONFIGURATION IS THAT DESCRIBED WITHIN PARAGRAPH 2.3.2 (B), AND DATA SWITCH 11 OF BOTH PDP-11'S ARE CONFIGURED TO REPRESENT A BINARY 1.

5.9 LOOP ON TEST

THE CORRECT EXECUTION OF EACH BB11-D SUBTEST IS DEPENDENT UPON THE EXECUTION OF THE PREVIOUS TESTS (IF ANY). THEREFORE, THERE IS NO "LOOP ON TEST OK" DATA SWITCH PROVISION.

IF HOWEVER, LOOPING ON A PASSING TEST (OR PORTION OF A PASSING TEST) IS DESIRED WITHOUT LOOPING THE ENTIRE DIAGNOSTIC (DATA SWITCH 13 = 0), PLACE WITHIN THE PROGRAM LOCATION FOLLOWING A "SCOPE", OR "SUBSCOPE" "TRAP" TRAP, OF THE TEST OR PORTION OF TEST DESIRED TO CYCLE UPON, THE OCTAL EQUIVALENT OF A "JMP @ .+2" ($_{10}$137) FOLLOWED BY THE ADDRESS OF THE FIRST INSTRUCTION OF THAT TEST (ALWAYS THE ADDRESS OF A "SCOPE" TRAP);

REFERENCING PARAGRAPH 7.4 (FUNCTIONAL DIAGRAM), TO "LOOP ON SUBTEST OK", (IN THIS EXAMPLE SUBTEST #2), PLACE WITHIN THE NEXT PROGRAM LOCATION FOLLOWING THE "SUBSCOPE" TRAP TRAP AN OCTAL 137, (THE ADDRESS OF E2+4) FOLLOWED BY THE FIRST ADDRESS OF THE LOOP (OR AT LEAST TO THE PREVIOUS "SCOPE" TRAP).

```
TA:  SCOPE
       .
       .
       .
       .
       .
       .
       .
       .
       .
       .
E2:  ERROR+X
     SUBSCOPE
     < 137 >
     < THE OCTAL EQUIVALENT
     OF THE ADDRESS OF "TA" >
       .
       .
       .
       .
       .
       .
       .
       .
       .
       .
       .
```

NOTE: THE TESTS FOLLOWING THE "JMP @ .+2" ARE NOT EXECUTED UNTIL THE ORIGINAL CONTENTS OF THE 2 PROGRAM LOCATIONS MODIFIED TO "LOOP ON TEST OK" ARE RESTORED.

6. ERRORS

6.1 OVERVIEW

THE FOLLOWING ERRORS ARE LISTED IN THE ORDER IN WHICH THEY MAY BE DETECTED.

E1	WC NOT 0 AFTER INIT
E2	BAR NOT 0 AFTER INIT
E3	DBR NOT 0 AFTER INIT
E4	CSR NOT = X6 1 0 AFTER INIT
E10	CSR NOT = X2 0 4
E12	CSR NOT = X2 1 0
E20	CSR NOT = X 1 1
E21	WC NOT = -177776
E22	BAR NOT = 0
E23	CSR NOT = X2 1 0
E30	CSR NOT = 126410
E31	WC NOT = -177776
E34	DBR NOT = 0
E32	BAR NOT = 0
E33	CSR NOT = 122610
E35	CSR NOT = 610
E50	THE CONTENTS OF THE DBR NOT = CONTENTS OF R0
E57	CSR NOT = 126412
E60	WC NOT = 177777
E61	BAR NOT = @BUFFER+2
E62	CSR NOT = 127410
E63	WC NOT = 0
E64	BAR NOT = @BUFFER+2
E65	CSR NOT = 122610
E70	CSR NOT = 11
E72	DBR NOT = @BUFFER+2
E74	WC NOT = 177777
E100	CSR NOT = 210
E102	BAR NOT = @BUFFER+4
E110	CSR NOT = 210
E112	DATA TRANSMITTED NOT = DATA RECIEVED
E116	CSR NOT = 210
E120	DATA TRANSMITTED NOT = DATA RECIEVED
E131	AN UNEXPECTED BB11-D INTERRUPT OCCURED
E130	CSR NOT = X3 1 0
E132	CSR NOT = X2 1 0
E140	NO BB11-D INTERRUPT OCCURED
E142	CSR NOT = X3 1 0
E143	NO BB11-D INTERRUPT OCCURED
E144	CSR NOT = 1 2 2 X1 1 0
E146	NO BB11-D INTERRUPT OCCURED
E152	KIT NOT ON PL= 5 (UNEXPECTED INTERRUPT)
E160	CSR NOT = 127410
E162	CSR NOT = 110210
E164	CSR NOT = 210
E202	OUTPUT BUST HAVE ADDRESSED " NEX "
E206	INPUT MUST HAVE ADDRESSED " NEX "
E210	DATA ERROR (DATA "IN" NOT = DATA EXPECTED)
*E376	UNEXPECVED BB11-D INTERRUPT (MAY OCCUR ANYTIME)

NOTE: CSR BIT "X" IS CSR BIT 08 (LAST DIRECTION) MAY REPRESENT A BINARY 1 OR 0.

6.2 ERROR TYPEOUT FORMAT

E (ERRORX) SCOPE (PCSCOPE) PASS (PASSLSB) ERROR (ERRORS)

BUFFER (BADDRESS) EXPECTED (GOOD) ACTUAL (BAD)

THE FOLLOWING INFORMATION IS PRINTED ON DETECTION OF A BB11-D ERROR IF DATA SWITCH 12 REPRESENTS A BINARY 0.

E	ERRORX	- THE E# CORRESPONDING TO AN ERROR DESCRIBED WITHIN THE PRECEEDING ERROR ABSTRACT.
SCOPE	PCSCOPE	- THE OCTAL EQUIVALENT TO THE FIRST ADDRESS OF THE LATCHED SCOPE LOOP.
PASS	PASSLSB	- PROGRAM PASS COUNTER FOR SA 200 (1-65534(10)).
ERROR	ERRORS	- ERROR COUNTER WITHIN A SCOPE LOOP (1-65534(10)).

THE FOLLOWING INFORMATION IS AMMENDED TO THE ABOVE PRINTOUT ON DETECTION OF A BB11-D DATA ERROR.

BUFFER	BADDRESS	- THE OCTAL ADDRESS CONTAINING THE DATA IN ERROR
EXPECTED	GOOD	- THIS OCTAL EQUIVALENT REPRESENTS THE DATA EXPECTED.
ACTUAL	BAD	- THIS OCTAL EQUIVALENT REPRESENTS THE ACTUAL DATA (RESULT OF DATA TEST).

6.3 ERROR RECOVERY

DATA SWITCH	BINARY REPRESENTATION
12	0

UPON THIS PROGRAMS DETECTION OF A BB11-D IRREGULARITY, AN ERROR MESSAGE WILL ENSUE ON THE TELEPRINTER (IF DATA SWITCH 12 = 0)...

15	0

...THEN THE PDP WILL HALT (IF DATA SWITCH 15 = 0)...

14	0

...THE FIRST BB11-D ERROR DETECTED LATCHES THIS PROGRAM INTO A SCOPE LOOP (THE FIRST ADDRESS OF THAT LOOP BEING THE CONTENTS OF PROGRAM LOCATION PCSCOPE) EVEN IF THE ERROR IS INTERMITTANT AS WOULD BE INDICATED BY THE ERROR COUNTER INCREMENTING, (THE CONTENTS OF PROGRAM LOCATION ERRORS).

THE OPERATOR MUST PLACE THE DATA SWITCH 14 IN THE POSITION REPRESENTING A " 1 " WHICH WOULD ABORT THE LATCHED SCOPE LOOP AND CONTINUE THIS PROGRAM (AS THOUGH NO ERROR HAS EVER DETECTED).

6.4 NO TYPEOUT (ERROR PARAMATER RECOVERY)

IF AN ERROR HAS OCCURED, BUT THE STATE OF DATA SWITCH 12 IS REPRESENTING A BINARY 1 (DISABELING ANY ERROR TYPECUT), OR THE SYSTEM CONFIGURATION HAS NO "HARD COPY" OUTPUT DEVICE FOR THE DEPOSIT OF ANY ERROR INFORMATION, THEN PLACE DATA SWITCH 15 TO REPRESENT A BINARY 0 (HALTING THE PDP AT THE DETECTION OF A 8811-D ERROR) AND EXAMINE FOR THE CONTENTS OF THE PROGRAM LOCATIONS DESCRIBED WITHIN PARAGRAPH 6.2 (ERROR TYPEOUT FORMAT) FOR THAT SPECIFIC INFORMATION RELATING TO THAT ERROR CONDITION.

IN SUMMATION, EXAMINING FOR THE CONTENTS OF PROGRAM LOCATION "ERRORX" WOULD REVEAL THE E#, AND OF "PCSCOPE", THE FIRST ADDRESS OF THE LATCHED SCOPE LOOP, AND ETC., ETC., ETC..

7. "TRAP" TRAP DISCUSSION

THE FOLLOWING PARAGRAPHS (THRU 7.3) ARE ATTEMPTED AS A SIMPLIFIED FUNCTIONAL OUTLINE OF THIS PROGRAMS ERROR, AND AUTOMATIC SCOPE LOOPING PROCEEDURES.

7.1 THE MNUMONICS "ERROR", "SCOPE" AND "SUBSCOPE" ARE "TRAP" TRAPS WHICH VECTOR THROUGH PROGRAM LOCATION 34 TO TRAPROUTINE " ERRORCATCHER " WHERE THE INTERROGATION OF THE IDENTIFIER " X " (CODE 104400+X,) WHERE X = 0 TO 377 OCTAL, PROCEEDS TO DETERMINE WHICH SUB-TRAP-ROUTINE IS EXECUTED. THE GENERATED ASSEMBLY CODE FOR THE MNUMONIC " SCOPE " IS 104400 ("TRAP" TRAP), FOR THE MNUMONIC " SUBSCOPE " IS 104777, AND FOR THE MNUMONIC " ERROR " IS 104400+X, WHERE X = 1 TO 376.

(A) IF THE CONTENTS OF PROGRAM LOCATION ERRORX IS EQUIVALENT TO THE OCTAL VALUE OF:

(1) 0, GOTO SUB-TRAP-ROUTINE " XSCOPE " (PARAGRAPH 7.2), OR
(2) 377, GOTO SUB-TRAP-ROUTINE " XSSCOPE " (PARAGRAPH 7.3).

(B) THE CONTENTS OF PROGRAM LOCATION "ERRORX" IS EQUIVALENT TO THE OCTAL VALUE BETWEEN 1 AND 376. THE CONTENTS OF PROGRAM LOCATION "PCSSCOPE" IS EQUIVALENT TO THE ADDRESS OF THE PROGRAM LOCATION+2 CONTAINING THE MNUMONIC "SCOPE", OR "SUBSCOPE" THEREFORE THE CONTENTS OF THE PROGRAM LOCATION "ERRORS" IS INCREMENTED.

(C) IF THE CONTENTS OF PROGRAM LOCATION " ERRORX " IS EQUIVALENT OF 200 THRU 277, CSR BIT 2 (FNCT) IS CLEARED (SA 202).

(D) IF THE DATA SWITCH 12 = 1, GOTO PARAGRAPH (E), OTHERWARDS AN APPROPRIATE ERROR MESSAGE IS PRINTED (REFER TO PARAGRAPH 6.2 FOR "ERROR TYPEOUT FORMAT ").

(E) IF DATA SWITCH 15 = 0, THE PDP IS HALTED.

(F) RTI (FROM ERROR+X "TRAP" TRAP)

7.2 SCOPE (SUB-TRAP-ROUTINE "XSCOPE")

(G) IF THE CONTENTS OF PROGRAM LOCATION ERRORS IS NOT EQUIV-
ALENT TO 0, GOTO PARAGRAPH (J).

(H) NO ERRORS HAVE BEEN DETECTED, OR DATA SWITCH 14 = 1 ABORTING
THE SCOPE LOOP ALREADY IN PROGRESS. REPLACE THE CONTENTS
OF PROGRAM LOCATION ERRORS WITH 0, AND REPLACE THE CONTENTS
OF PROGRAM LOCATION PCSSCOPE WITH THE CONTENTS OF THE CON-
TENTS OF THE STACK POINTER (RETURN PC) FOR LATER USE AS THE
FIRST ADDRESS OF A SCOPE LOOP IF AN ERROR IS DETECTED.

(I) RTI (FROM SCOPE "SUB-TRAP" TRAP)

(J) THIS IS A SCOPE LOOP. IF THE DATA SWITCH 14 = 1, GOTO
" NOSCOPE ", (PARAGRAPH H).

(K) REPLACE THE CONTENTS OF THE CONTENTS OF THE STACK POINTER
WITH THE CONTENTS OF PROGRAM LOCATION " PCSSCOPE " (1ST
LOCATION OF THE SCOPE LOOP).

(L) RTI (TO LATCHED SCOPE LOOP)

7.3 SUBSCOPE (SUB-TRAP-ROUTINE "XSSCOPE")

IF THE CONTENTS OF PROGRAM LOCATION "ERRORS" = 0, AN RTI IS
PERFORMED.

SOME TYPE OF BB11-D ERROR EXISTS. IF THE RETURN PC (THE
CONTENTS OF THE CONTENTS OF THE STACK POINTER) IS EQUIV-
ALENT TO THE CONTENTS OF PROGRAM LOCATION PCSSCOPE, THEN
THIS BB11-D ERROR IS THE ERROR WHICH IS THE ADJITATION OF
THE SCOPE LOOP, AND NOT SOME PREVIOUS ERROR WHICH OCCURED
WHILE ALREADY WITHIN THE SCOPE LOOP, THEREFORE GOTO PARA-
GRAPH (J) " SCOPING ", OTHERWARDS AN RTI IS PERFORMED.

7.4 FUNCTIONAL DIAGRAM

THIS PARAGRAPH IS INTENDED TO RENDER AN OVERVIEW OF THE FUNCTIONAL DESCRIPTION OF PARAGRAPHS 7.1 THRU 7.3 .

" T " INDICATES THE START OF A PARTICULAR BB11-C TEST,
" E " SIGNIFIES AN ERROR "TRAP" TRAP.

```
TA:     SCOPE               ; NOTE #1

        <<<<< SUBTEST 1 OF A >>>>>

E1:     ERROR*X
        SUBSCOPE            ; NOTE #2

        <<<<< SUBTEST 2 OF A >>>>>

E2:     ERROR*X
        SUBSCOPE            ; NOTE #2

        <<<<< SUBTEST 3 OF A >>>>>

E3:     ERROR*X


TB:     SCOPE               ; NOTE #3

        <<<<< SUBTEST 1 OF B >>>>>

E4:     ERROR*X
        SCOPE               ; NOTE #4

        <<<<< SUBTEST 2 OF B >>>>>

E5:     ERROR*X

        ...ETC...
```

NOTE #1: THIS IS THE INITIAL ENTRY INTO PROGRAM CODE "XSCOPE" WHICH SETS THE ADDRESS OF "TA+2" (RETURN PC) INTO PROGRAM LOCATION "PCSSCOPE" FOR SCOPE LOOPING FOR THIS TEST,

NOTE #2: IF A BB11-D ERROR EXISTS, SUB-SCOPE TO PROGRAM LOCATION TA+2.

NOTE #3: IF A BB11-D ERROR EXISTS, SCOPE LOOP TO PROGRAM LOCATION TA+2. OTHERWARDS, THIS IS THE INITIAL ENTRY INTO PROGRAM CODE "XSCOPE" ...FOR THIS TEST,

NOTE #4: IF A BB11-D ERROR EXISTS, SCOPE LOOP TO PROGRAM LOCATION TB+2.

8. DISCUSSION

8.1 CSR BIT ALLOCATIONS

THE CONTROL/STATUS REGISTER (CSR) IS THE ONLY BYTE ADDRESSABLE REGISTER. WRITING INTO THE LOW BYTE (OR INTO THE ENTIRE WORD) WILL AUTOMATICALLY CLEAR "SPECIAL CONDITION", AND "NEX" AND/OR "XOF" IF PREVIOUSLY SET.

CSR	MNUMONIC	
0	IN	R/W
1	OUT	R/W
2	FNCT	R/W
3	STAT	R
4	RESERVED	
5	RESERVED	
6	ENB	R/W
7	READY	R
8	LAST DIRECTION	R
9	LAST TRANSFER	R
10	INPUT	R/WRITE 0
11	DATA AVAILABLE (OUT)	R/WRITE 0
12	XOF (EXTERNAL OVFL)	R/CLEARED BY WRITING LOW BYTE OF CSR
13	INPUT DEMAND	R
14	NEX	R/CLEARED BY WRITING LOW BYTE OF CSR
15	SPECIAL CONDITION	R/CLEARED BY WRITING LOW BYTE OF CSR

PROGRAMMING NOTE: CSR BITS 10 (INPUT) AND 11 (DATA AVAILABLE OUT) MAY BE CLEARED BY WRITING A 0 INTO THAT PARTICULAR BIT, THEREFORE, WHENEVER ADDRESSING THE CSR, AND THE STATE OF CSR BITS 10, AND 11 ARE IMPORTANT, WRITE THESE BITS TO A 1. (WRITING THESE BITS TO A 1 DOES NOT SET THEM, IT ONLY ASSURES THAT THEY WON'T BE CLEARED).

9. TEST DESCRIPTIONS

9.1 DIAGNOSTIC TEST DESCRIPTIONS

THE FOLLOWING PARAGRAPHS ARE ATTEMPTED TO SUMMARIZE EACH DIAGNOSTIC TEST. THE E#'S WITHIN THE <> ARE INTENDED TO INDICATE WHICH ERRORS MAY BE DETECTED.

9.1.1 TEST 0 (T0) - " INIT "

THIS PROGRAM ISSUES THE PDP-11 "RESET" INSTRUCTION, THEN TESTS THE BB11-D WORD COUNT REGISTER TO = 0, THE BUS ADDRESS REGISTER TO = 0, THE DATA BUFFER REGISTER TO = 177777, AND FINALLY THE CSR TO = X_2 1 0 (WHERE X=BIT 0 LAST DIRECTION = 1 OR 0).

" RESET " WILL BE ISSUED WITHIN THE SCOPE LOOP.

<E1, E2, E3, E4>

9.1.2 TEST 2 (T2) - CSR BITS 2 (FNCT), AND 3 (STAT)

THIS PROGRAM FLIPS BIT 2 (FNCT) OF THE CSR TESTING BIT 3 (STAT) TO BE THE COMPLEMENT OF THE "FNCT" BIT.

<E10, E12>

9.1.3 TEST 3 (T3) - CSR BITS 0 (IN), AND 7 (RDY)

THE BB11-D IS "READY" WHEN "IN", AND "OUT" BOTH = 0, AND THERE IS NO "DATA AVAILABLE (OUT)".

THIS PROGRAM FLIPS THE CSR BIT 0 (IN), AND TESTS THE CSR BIT 7 (RDY) TO = 0.

NO INTERRUPTS OR DMA'S ARE EXPECTED, THEREFORE THE WORD COUNT REGISTER AND BUS ADDRESS REGISTER SHOULD REMAIN UNCHANGED.

<E20, E21, E22, E23>

9.1.4 TEST 5 (T5) - DATA AVAILABLE OUT

THIS PROGRAM WRITES INTO THE BB11-D'S DATA BUFFER REGISTER WHICH SHOULD RAISE "DATA AVAILABLE OUT".

THE COMMAND STATUS REGISTER SHOULD = 126410,
THE WORD COUNT REGISTER SHOULD STILL = 177776,
THE DATA BUFFER REGISTER SHOULD = 0,
THE BUS ADDRESS REGISTER SHOULLD STILL = 0,

15 SPECIAL CONDITION (1 BECAUSE "INPUT DEMAND")
13 INPUT DEMAND (1 BECAUSE "INPUT"=1, AND "IN"=0)
11 DATA AVAILABLE OUT (1 BECAUSE OF WRITING INTO THE DBR)
10 INPUT (1 BECAUSE "DATA AVAILABLE OUT"
 0 LAST DIRECTION (1=OUT)
 7 READY (0 BECAUSE "IN"=0, "OUT"=0, AND DATA AVAILABLE =1)
 3 STATUS (1 BECAUSE FNCT=0)

<E30, E31, E34, E32, E33, E35>

9.1.5 TEST 7 DATA BUFFER (CABLE WRAP AROUND)

WITH THE DIAGNOSTIC "WRAP AROUND" CABLE CONNECTED FROM BB11-D LOGICAL LOCATION D1 (M9100) J1 TO EF01 (M1502), WRITING INTO THE DATA BUFFER WILL PLACE THAT DATA ONTO THE DATA LINES (OUT), THEREFORE, THE CONTENTS OF THE DATA BUFFER SHOULD BE EQUIVALENT TO THE DATA OUT.

<E50>

9.1.6 TEST 8 OUTPUT (PART 1)

THIS TEST ENABLES FOR A 2 WORD DATA TRANSFER OUT, BUT BECAUSE DATA IS NOT ACCEPTED, THERE SHOULD BE NO WORD COUNT OVERFLOW.

THE CSR SHOULD = 126412;
THE WORD COUNT REGISTER SHOULD = 177777,
THE BUS ADDRESS REGISTER SHOULD = #BUFFER+2,

<E57, E60, E61>

9.1.7 TEST 9 OUTPUT (PART 2)

THIS TEST ENABLES FOR A 1 WORD DATA TRANSFER OUT EXPECTING A WORD COUNT OVERFLOW.

THE CSR SHOULD = 127413;
THE WORD COUNT REGISTER SHOULD HAVE OVERFLOWED TO 0,
THE BUS ADDRESS REGISTER SHOULD = #BUFFER+2,

THIS TEST THEN CLEARS CSR BIT 11 (DATA AVAILABLE OUT), WHICH CLEARS CSR BIT 9 (LAST TRANSFER), AND BECAUSE CSR BIT 2 (IN), AND 1 (OUT) WERE BOTH PREVIOUSLY = 0, CSR BIT 7 (READY) BECOMES SET.

THE CSR SHOULD = 122613;

<E65>

9.1.8 TEST 10 INPUT

THIS TEST ENABLES BIT 0 (IN) THEN RAISES DATA AVAILABLE OUT BY WRITING INTO THE BB11-D DATA BUFFER REGISTER, ONE DMA TRANSFER SHOULD OCCUR.

THE CSR SHOULD = 11;
THE BUSS ADDRESS REGISTER SHOULD = #BUFFER+2,
THE WORD COUNT REGISTER SHOULD = 177777,

<E70, E72, E74>

[redacted] TEST 11 OUTPUT/INPUT (SIMULTANEOUSLY)

THIS TEST ENABLES BOTH CSR BITS 0 (IN), AND 1 (OUT). TWO DMA TRANSFERS SHOULD OCCUR (1 OUT, AND 1 IN).

THE CSR SHOULD = 2107
THE BUS ADDRESS REGISTER SHOULD = @BUFFER+4.

<E100, E102>

9.1.10 TEST 12 DATA TEST (PART 1)

THIS TEST ENABLES FOR 34 DMA TRANSFERS (17 OUT, AND 17 IN) THEN COMPARES DATA TO THE FOLLOWING EXPECTED PATTERN:

(FIRST) 0 1 2 4 10
20 40 100 200 400
1000 2000 4000 10000 20000
40000 100000 (LAST WORD)

<E110, E112>

9.1.11 TEST 13 DATA TEST (PART 2)

THIS TEST ENABLES FOR 64 DMA TRANSFERS (32 OUT, AND 32 IN) COMPARING DATA WITH THE EXPECTED WORST CASE PATTERN OF A COMPLEMENTING FLOATING " 1 ".

(FIRST) -2, 1 -3, 2 -5, 4
-11, 10 -21, 20 -41, 40
-101, 100 -201, 200 -401, 400
-1001, 1000 -2001, 2000 -4001, 4000
-10001, 10000 -20001, 20000 -40001, 40000
-100001, 100000 (LAST WORD)

<E116, E120>

9.1.12 TEST 14 INTERRUPT ENABLE (PART 1)

THIS TEST FLIPS CSR BIT 6 (INTERRUPT ENABLE) EXPECTING NO DEVICE INTERRUPT.

<E131, E138, E132>

9.1.13 TEST 15 INTERRUPT ENABLE (PART 2)

THIS TEST ENABLES THE BB11-D EXPECTING A DEVICE INTERRUPT (FROM SPECIAL CONDITION WORD COUNT OVERFLOW) AT ANY PRIORITY LEVEL FOR NOW.

<E148, E142>

9.1.14 TEST 16 INTERRUPT ENABLE (PART 3)

THIS TEST ENABLES THE BB11-D EXPECTING A DEVICE INTERRUPT FROM SPECIAL CONDITION INPUT DEMAND AT ANY PRIORITY LEVEL FOR NOW.

<E143, E144>

9.1.15 TEST 17 BB11-D INTERRUPT ENABLE PRIORITY LEVEL (PART 4)

THIS TEST ENABLES THE BB11-D EXPECTING A DEVICE INTERRUPT FROM SPECIAL CONDITION AT PRIORITY LEVELS 5, 6, OR 7 FOR NOW.

<E146>

9.1.16 TEST 18 INTERUPT ENABLE (PART 5)

THIS TEST ENABLES THE BB11-D NOT EXPECTING A DEVICE INTERRUPT AT PRIORITY LEVELS 6, OR 7 BECAUSE THE PDP PRIORITY IS 5.

<E152>

9.1.17 TEST 19 EXTERNAL OVERFLOW (XOF)

THIS IS A TEST FOR EXTERNAL OVERFLOW (XOF) CSR BIT 12. A ONE WORD DMA OUTPUT IS ENABLED EXPECTING A BB11-D WORD COUNT OVERFLOW AND LAST TRANSFER, THEN, WITHOUT CLEARING LAST TRANSFER, A ONE WORD DMA DATA INPUT IS ENABLED EXPECTING EXTERNAL OVERFLOW.

<E168, E162, E164>

9.2 SPECIAL CONVERSATION EXERCISE DESCRIPTION (STARTING ADDRESS 202)

BASICALLY, EACH KIT, WITH INTERRUPT ENABLE, WILL TRANSMIT TO THE "OTHER" KIT. UPON THE DETECTION OF "INPUT DEMAND", EACH KIT WILL DISABLE "OUT", AND ENABLE "IN", THUS INPUTTING DATA. IF AFTER 20US BETWEEN DMA INPUT TRANSFERS (TWICE THE MAXIMUM TRANSFER TIME OF 10US) THE INPUT TRANSFERS CEASE, EACH KIT WILL DISABLE "IN", AND RESUME TRANSMISSIONS TO THE "OTHER" KIT AT THE POINT AT WHICH THEY WERE SUSPENDED.

THE DATA INPUTTED IS COMPARED TO THE DATA EXPECTED AT THE DETECTION OF "EXTERNAL OVERFLOW". ANY ERROR MESSAGE, IF APPLICABLE, WILL ENSUE.

THIS ASYNCHRONOUS OPERATION MAY BE DEFERRED SOMEWHAT BY PLACING DATA SWITCH 11 TO REPRESENT A BINARY 1 THUS CAUSING A "BURST" TYPE OF OPERATION.

BASICALLY, EACH KIT, WITH INTERRUPT ENABLE WILL TRANSMIT TO THE "OTHER" KIT. UPON THE DETECTION OF "INPUT DEMAND", THE BINARY STATE OF DATA SWITCH 11 IS TESTED, AND IF = 1, "THIS" KIT WILL *NOT* DISABLE "OUT", BUT WILL CONTINUE TRANSMITTING UNTIL A WORD COUNT OVERFLOW OCCURS, THEN INPUT DEMAND IS RECOGNIZED AND APPROXIMATELY 1 DMA "IN" OCCURS.

IN SUMMARY, THE PDP WHOSE DATA SWITCH 11 REPRESENTS A BINARY 1 WILL IGNORE "INPUT DEMAND"'S UNTIL ALL "N" WORDS ARE TRANSMITTED TO THE "OTHER" KIT, THEN 1 WORD "IN" IS ACCEPTED (APPROXIMATELD 1 OR 1+X DEPENDING UPON THE LENGTH OF THE PROGRAM SEQUENCE IN WHICH THE "OTHER" KIT IS COMPARING ALL OF THOSE "N" DATA WORDS RECIEVED). THEREFORE, "THIS" KIT WOULD TRANSFER "N" DATA WORDS, "N" TIMES, IN WHICH "N" WORD COUNT OVERFLOW'S WOULD OCCUR BEFORE "THIS" KIT DETECTED 1 EXTERNAL OVERFLOW (XOF) CAUSED BY A WORD COUNT OVERFLOW OF THE "OTHER" KIT. THE VALUE OF "N" IN THIS PROGRAM IS 32.

NOTE: REFERENCE PARAGRAPH 9.4 FOR OPERATIONAL DATA SWITCH 11 RESTRICTION.

10. LISTING

```
                                        .TITLE MAINDEC-11-DZKDA-A
                                        .ENABLE ABS
                                        .ENABLE AMA
                                .NLIST SEQ,LOC

                        ; TEST 0 - " INIT "

001434                          E1      ; WC NOT 0 AFTER INIT
001446                          E2      ; BAR NOT 2 AFTER INIT
001462                          E3      ; DBR NOT 2 AFTER INIT
001474                          E4      ; CSR NOT = X0 1 0  AFTER INIT

                        ; TEST 2 - CSR BITS: 2 (FNCT), AND 3 (STAT)

001514                          E10     ; CSR NOT = X2 0 4
001532                          E12     ; CSR NOT = X2 1 0

                        ; TEST 3 - CSR BITS: 0 (IN), AND 7 (RDY)

001572                          E20     ; CSR NOT = X 1 1
001606                          E21     ; WC NOT = 177776
001620                          E22     ; BAR NOT = 0
001640                          E23     ; CSR NOT = X2 1 0

                        ; TEST 5 - " DATA AVAILABLE OUT "

001702                          E30     ; CSR NOT = 126410
001716                          E31     ; WC NOT = 177776
001730                          E34     ; DBR NOT = 0
001742                          E32     ; BAR NOT = 0
001764                          E33     ; CSR NOT = 122610
002006                          E35     ; CSR NOT = 610

                        ; TEST 7 - DATA BUFFER REGISTER WRAP AROUND

002042                          E50     ; THE CONTENTS OF THE DBR NOT = CONTENTS OF R0

                        ; TEST 8 - DATA OUTPUT (PART 1)
                        ;          ONE NPR (OUT) W/O WORD COUNT OVERFLOW

002122                          E57     ; CSR NOT = 126412
002136                          E60     ; WC NOT = 177777
002152                          E61     ; BAR NOT = @BUFFER+2

                        ; TEST 9 - DATA OUTPUT (PART 2)
                        ;          ONE NPR OUTPUT WITH WORD COUNT OVERFLOW

002220                          E62     ; CSR NOT = 127410
002232                          E63     ; WC NOT = 0
002246                          E64     ; BAR NOT = @BUFFER+2
002270                          E65     ; CSR NOT = 122610
```

```
                        ; TEST 10 - INPUT

002336                          E70     ; CSR NOT = 11
002352                          E72     ; DBR NOT = #BUFFER+2
002366                          E74     ; WC NOT = 177777

                        ; TEST 11 - OUTPUT/INPUT

002436                          E100    ; CSR NOT = 210
002452                          E102    ; BAR NOT = #BUFFER+4

                        ; TEST 12 - DATA VERIFICATION TEST (PART 1)

002554                          E110    ; CSR NOT = 210
002602                          E112    ; DATA TRANSMITTED NOT = DATA RECIEVED
                                        ; THE CONTENTS OF BUFFER+X NOT = THE CONTENTS OF BUFFER+X+2

                        ; TEST 13 - DATA VERIFICATION (PART 2)

002712                          E116    ; CSR NOT = 210
002740                          E120    ; DATA TRANSMITTED NOT = DATA RECIEVED
                                        ; THE CONTENTS OF BUFFER+X NOT = THE CONTENTS OF BUFFER+X+2

                        ; TEST 14 - INTERRUPT ENABLE (NONE)

002774                          E131    ; AN UNEXPECTED BB11-D INTERRUPT OCCURED
003006                          E130    ; CSR NOT = X3 1 0
003026                          E132    ; CSR NOT = X2 1 0

                        ; TEST 15 - INTERRUPT ENABLE (EXPECTING AN INTERRUPT FROM WCO)

003112                          E140    ; NO INTERRUPT OCCURED
003124                          E142    ; CSR NOT = X3 1 0

                        ; TEST 16 - INTERRUPT ENABLE (FROM SPECIAL CONDITION)

003200                          E143    ; NO BB11-D INTERRUPT OCCURED
003212                          E144    ; CSR NOT = 1 2 2 X1 1 0

                        ; TEST 17 - PRIORITY LEVEL OF KIT

003276                          E146    ; NO INTERRUPT OCCURED

                        ; TEST 18 - PRIORITY LEVEL 5 VERIFICATION

003364                          E152    ; KIT NOT ON PL= 5 (UNEXPECTED INTERRUPT)

                        ; TEST 19 - LAST TRANSFER - EXTERNAL OVERFLOW (XOF)

003432                          E160    ; CSR NOT = 127410
003462                          E162    ; CSR NOT = 110210
003502                          E164    ; CSR NOT = 210
```

```
                        ; STARTING ADDRESS 2P2
                        ; CONVERSATION BETWEEN 2 OR MORE BB11-D'S
      003774                    E202    ; SPECIAL CONDITION = 1, MUST BE " NEX "
      004176                    E206    ; SPECIAL CONDITION = 1 (NOT " XOF " )
                                        ; MUST BE " NEX "
      004250                    E210    ; DATA ERROR
  118                           .LIST SEQ,LOC
```

```
                                ; DATA SWITCH DEFINITIONS
                                ;
100000                          SW15=100000
040000                          SW14=40000
020000                          SW13=20000
010000                          SW12=10000
004000                          SW11=4000
                                ;SW10=2000
                                ;SW9=1000
000400                          SW8=400
                                ;SW7=200
                                ;SW6=100
                                ;SW5=40
                                ;SW4=20
                                ;SW3=10
                                ;SW2=4
                                ;SW1=2
                                ;SW0=1
                                ;
                                ; DATA SWITCH OPERATIONAL DEFINITIONS
                                ;
                                ; SW15 = 0      -       HALT ON ERROR
                                ; SW14 = 0      -       SCOPE LOOP ON ERROR (LATCHED BY FIRST ERROR)
                                ; SW13 = 1      -       HALT AT THE END OF EACH PROGRAM PASS
                                ; SW12 = 0      -       PRINT ERROR INFORMATION AT DETECTION
                                ;
                                ; SPECIAL DATA SWITCH FOR STARTING ADDRESS 202
                                ;
                                ; BURST MODE
                                ;
                                ; THIS BB11-D WILL NOT HONOR " INPUT DEMAND " UNTIL ALL DATA HAS BEEN
                                ; TRANSMITTED
                                ;
                                ; SW11 = 1      -       BURST MODE TRANSMISSION
```

```
                        .=0
000000  000002          .+2                     ; UNASSIGNED TRAP
000002  000000          HALT
000004  000006          .+2                     ; SP OVERFLOW, BUS ERROR TRAP
000006  000000          HALT
000010  000012          .+2                     ; RESERVED INSTRUCTION TRAP
000012  000000          HALT
000014  000016          .+2                     ; TRACE TRAP
000016  000000          HALT

                        ; MACRO LOAD THE INTERUPT VECTOR ADDRESSES (20-777) WITH THE " IOT " TRAP

                        .REPT 174

                        .+2
                        4                       ; IOT (TRAPPED TO PREVIOUS ADDRESS)

                        .ENDR

                        .=20

000020  004730          KITCATCHER              ; IOT TRAP (KIT INTERRUPTS VIA IOT)
000022  000340          340
000024  000026          .+2                     ; POWER FAIL TRAP
000026  000000          HALT
000030  000032          .+2                     ; EMT TRAP
000032  000000          HALT

                        .=34

000034  004342          ERRORCATCHER            ; TRAP TRAP
000036  000340          340

                        .=64

000064  000066          .+2
000066  000000          HALT

                        .=200

000200  000401          BR 1$
000202  000402          BR 2$

000204  000137  001254  1$:     JMP $200
000210  000137  003534  2$:     JMP $202

                        ; .=774

                        ;       KITCATCHER      ; KIT 11/D VECTOR ADDRESS IF NO JUMPERS CUT
                        ;       340             ; ON THE M7821 MODULE
```

```
                        001200          .=1200
                                        ; BB11-D REGISTER ADDRESSES (ONLY THE CSR IS BYTE ADDRESSABLE)
                                        ;
001200  160000          WC:     160000  ; WORD COUNT REGISTER
001202  160002          BAR:    160002  ; BUS ADDRESS REGISTER
001204  160004          CSR:    160004  ; CONTROL-STATUS REGISTER
001206  160005          CSR1:   160005  ;       HIGH BYTE ADDRESS OF CSR
001210  160006          DBR:    160006  ; DATA BUFFER REGISTER
                                        ;
                                        ; XWC
                                        ; XBAR
                                        ; XCSR
                                        ; XCSR1
                                        ; XDBR
                                        ;
                        001200          .=1200
001200  000000          .WORD 0
                                        ;
                        001220          .=1220
                                        ;
                                        ; PDP-11 SOFTWARE REGISTER ADDRESSES
                                        ;
001220  177566          TPB:    177566  ; TELEPRINTER BUFFER
001222  177564          TPS:    177564  ; TELEPRINTER STATUS REGISTER
001224  177562          TKB:    177562  ; KEYBOARD BUFFER
001226  177560          TKS:    177560  ; KEYBOARD STATUS REGISTER
                                        ;
                                        ;
                                        ; REGISTER DEFINITIONS
                                        ;
                        000007          PC=%7
                        000006          SP=%6
                        000005          R5=%5
                        000004          R4=%4
                        000003          R3=%3
                        000002          R2=%2
                        000001          R1=%1           ; WORK: T7, T12, T13
                        000000          R0=%0           ; WORK: T12, T13
                                        ;
                                        ; STANDARD ADDRESS DEFINITIONS
                                        ;
                        177776          PSW=177776      ; PROCESSOR STATUS WORD
                        177570          SWR=177570      ; SWITCH REGISTER
                                        ;
                                        ; "TRAP" TRAP EQUALITIES
                                        ;
                        104400          ERROR=TRAP      ; ERROR=TRAP 1 TO TRAP 376
                        104400          SCOPE=TRAP      ;SCOPE=TRAP 0
                        104777          SUBSCOPE=TRAP+377       ;SUBSCOPE=TRAP 377
```

```
                                ERRORS:         0               ; IF >0 SOME TYPE OF KIT11/D ERROR EXISTS
                                ERRORX:         0               ; CONTAINS THE KIT11D ERROR #
                                ;  IRQ = 100000+X, PRINT KIT VECTOR ADDRESS AT DETECTION OF INTERRUPT
                                ;  IRQ = X-100000, A KIT INTERRUPT IS EXPECTED
                                IRQ:            0               ; = 1 IF A KIT INTERUPT IS EXPECTED
                                IRQOK:          0               ; THE ADRS FOR RETURN FROM AN EXPECTED KIT INTERUPT
                                LINECOUNT:      11#             ; 72 CHARACTER TELEPRINTER LINE
                                PASSLSB:        0               ; PASS COUNTER LSB TO 177777 OCTAL
                                PCSCOPE:        #               ; CONTAINS THE FIRST ADDRSS OF THE " SCOPE " LOOP
                                PCSSCOPE:       0               ; ADDRESS+2 OF THE COMMAND " SUBSCOPE " OR " SCOPE "

                                ;                       WORD COUNT REGISTER = 16XXX0

                                ; THIS 16 BIT REGISTER IS TO BE LOADED WITH THE 2'S COMPLEMENT OF THE
                                ; NUMBER OF WORDS TO BE TRANSFERRED.  CAUTION, UNLESS THE KIT IS MODIFIED
                                ; IT CAN ADDRESS ONLY UP TO 32K WORDS,  THEREFORE, BIT 15 SHOULD ALWAYS
                                ; BE LOADED WITH A  "ONE"  TO KEEP THE TOTAL NUMBER OF TRANSFERS TO 32K
                                ; OR LESS. (THIS REGISTER IS NOT BYTE ADDRESSABLE).

                                ;                       BUS ADDRESS REGISTER = 16XXX2

                                ; THIS 15 BIT REGISTER IS TO BE LOADED WITH THE ADDRESS THAT SPECIFIES
                                ; THE MEMORY LOCATION  "TO"  OR  "FROM"  WHICH THE FIRST DATA WORD IS TO
                                ; BE TRANSFERRED.  THIS REGISTER IS INCREMENTED AFTER EACH TRANSFER.
                                ; CAUTION, IF THIS REGISTER INCREMENTS TO ITS 32K ADDRESS LIMIT AND THE
                                ; WORD COUNT REGISTER HAS NOT INCREMENTED TO ZERO AND THERE HAS BEEN NO
                                ; WORD COUNT OVERFLOW, THE ADDRESS WILL  "WRAP AROUND"  TO LOCATION 0 ON
                                ; THE NEXT TRANSFER.  (THIS REGISTER IS NOT BYTE ADDRESSABLE).

                                ;                       CONTROL/STATUS REGISTER = 16XXX4

                                ; THIS 16 BIT REGISTER IS THE ONLY REGISTER WHICH IS BYTE ADDRESSABLE.

                                ; WRITING INTO THE LOW BYTE WILL CLEAR " NEX ", AND " XOF ",
```

```
                                        ; THIS ERROR IS FROM AN UNEXPECTED KIT VECTOR ADDRESS INTERRUPT
                                        ; ENTRY TO HERE MAY BE FROM ANYWHERE WITHIN THIS PROGRAM
                                        E376:   ERROR*376
                                                SUBSCOPE
                                        ; STARTING ADDRESS 200
001254  012737  100000  001234  S200:   MOV #SW15, IRQ          ; IRQ = 100000
001262  005037  001242                  CLR PASSLSB
001266  012706  001200          CONTINUE: MOV #1200, SP         ; PRIME THE STACK POINTER = 1200
001272  112737  000340  177776          MOVB #340, PSW          ; SET THE STATUS PRIORITY=7
001300  005037  001230                  CLR ERRORS              ; PRIME FOR ANY ERROR COUNT
001304  012737  000110  001240          MOV #110, LINECOUNT     ; INITIALIZE LINECOUNT FOR 72 CHARACTERS
001312  012737  001250  001236          MOV #E376, IRQOK        ; FOR INVALAD KIT VECTOR INTERRUPT
001320  005737  001200                  TST WC
001324  001020                          BNE XS200
001326  004537  005006                  JSR R5, PRINT
001332  005627                          MHELP1                  ; "TOGGLE INTO PROGRAM
001334  012737  001200  004726          MOV #WC, VA             ; LOCATION 1200 THE
001342  004537  005100                  JSR R5, TYPEOCTAL       ; BB11-D DEVICE ADDRESS
001346  004726                          VA                      ; IF NOT 160000
001350  004537  005006                  JSR R5, PRINT
001354  005670                          MHELP2
001356  000000                          HALT
001360  012737  160000  001200          MOV #160000, WC
001366  013700  001200          XS200:  MOV WC, R0
001372  005720                          TST (R0)+               ;     " ADD #2, (R0) "
001374  010037  001262                  MOV R0, BAR             ; BAR ADDRESS = (WC) + 2
001400  005720                          TST (R0)+
001402  010037  001264                  MOV R0, CSR             ; CSR ADDRESS = (WC) + 4
001406  105720                          TSTB (R0)+
001410  010037  001266                  MOV R0, CSR+1           ; CSR (HIGH BYTE) = (WC) + 5
001414  105720                          TSTB (R0)+
001416  010037  001210                  MOV R0, DBR             ; DBR ADDRESS = (WC) + 6
```

```
                                ;              TEST 8 = " INIT "
                                ;
                                ; THE STATUS OF THE KIT'S REGISTERS AFTER INITIALIZE SHOULD BE:
                                ;
                                ; WORD COUNT = 0, BUS ADDRESS REGISTER = 0,
                                ;
                                ; DATA BUFFER R = 177777, AND CSR = X2 1 0
                                ;
                                ; (NOTE: " RESET " WILL BE ISSUED WITHIN THE SCOPE LOOP).
                                ;
001422  104400          T8:     SCOPE                   ; TO PRIME FIRST " SCOPE " LOOP EVER
                                ;
001424  000005                  RESET                   ; ISSUED NOW AND IN SCOPE LOOP
                                ;
001426  005777  177546          TST @ WC                ; TEST WORD COUNT = 0
001432  001401                  BEQ .+4
001434  104401          E11:    ERROR+1                 ; THE WORD COUNT NOT = 0 AFTER " INIT "
001436  104777                  SUBSCOPE                ; LOOP
                                ;
001440  005777  177536          TST @ BAR               ; TEST BUS ADDRESS R = 0
001444  001401                  BEQ .+4
001446  104402          E21:    ERROR+2                 ; THE BUS ADDRESS R NOT = 0 AFTER " INIT "
001450  104777                  SUBSCOPE                ; LOOP
                                ;
001452  022777  177777  177530  CMP #-1, @ BBR          ; TEST THE DATA BUFFER R = 177777
001460  001401                  BEQ .+4
001462  104403          E31:    ERROR+3                 ; THE DATA BUFFER R NOT = -1 AFTER " INIT "
001464  104777                  SUBSCOPE
                                ;
001466  004537  005412          JSR R5, XCSR
001472  000210                  210
001474  104404          E41:    ERROR+4
```

```
                                        ;       TEST 2 - CSR BITS: 2 (FNCT), AND 3 (STAT)
                                        ;
                                        ; THE STATE OF BIT 3 IS THE COMPLEMENT OF BIT 2
                                        ;
001476  104400                  T21:    SCOPE
                                        ;
001500  052777  000004  177476          BIS #4, @ CSR           ; SETTING BIT 2 SHOULD CLEAR BIT 3
                                        ;
001506  004537  005412                  JSR R5, XCSR
001512  000204                          204
001514  104410                  E10:    ERROR+10
001516  104400                          SCOPE
                                        ;
                                        ; CLEARING BIT 2 SHOULD SET BIT 3
                                        ;
001520  005077  177460                  CLR @ CSR
                                        ;
001524  004537  005412                  JSR R5, XCSR
001530  000210                          210
001532  104412                  E12:    ERROR+12
```

```
                                        ;              TEST 3 - CSR BITS; 0 (IN), 7 (RDY)
                                        ;
                                        ; SETTING BIT 0 (IN) OF THE CSR SHOULD CLEAR BIT 7 (RDY)
                                        ;
                                        ; NO INTERUPTS OR DMA'S ARE EXPECTED
                                        ;
001534  104400          T3:     SCOPE
                                        ;
001536  005077  177442          CLR @ CSR               ; THE CSR IS DEFAULT TO 612
                                        ;
001542  012777  177776  177438  MOV #-2, @ WC            ; SET THE WORD COUNT R = -2
001550  005077  177426          CLR @ BAR               ; SET THE BUS ADDRESS R = 0
001554  005277  177424          INC @ CSR               ; SET BIT 0 (IN) OF THE CSR = 1
                                        ;
001560  004737  000406          JSR PC, STALL           ; 10.5 US TIME PER XFER
                                        ;
001564  004537  005412          JSR R5, XCSR
001570  000011                  11
001572  104420          E20:    ERROR+20
001574  104777                  SUBSCOPE                ; LOOP
                                        ;
001576  027727  177376  177776  CMP @ WC, #-2           ; THE WORD COUNT SHOULD STILL = -2
001604  001401                  BEQ .+4
001606  104421          E21:    ERROR+21                ; THE WORD COUNT NOT = -2
001610  104777                  SUBSCOPE                ; LOOP
                                        ;
001612  005777  177364          TST @ BAR               ; THE BUS ADDRESS R SHOULD STILL = 0
001616  001401                  BEQ .+4
001620  104422          E22:    ERROR+22                ; THE BUS ADDRESS R NOT = 0
001622  104400                  SCOPE                   ; LOOP
                                        ;
                                        ; NOW CLEAR BIT 0 (IN) OF THE CSR, THEREBY SETTING BIT 7 (RDY) = 1
                                        ;
                                        ; BIT 0 (LAST DIRECTION) SHOULD PREVIOUSLY BE = 1
                                        ;
001624  042777  000001  177352          BIC #1, @ CSR           ; CLEARING BIT 0 SHOULD SET BIT 7
                                        ;
001632  004537  005412          JSR R5, XCSR
001636  000210                  210
001640  104423          E23:    ERROR+23
                                        ;
```

```
                                ;              TEST 9 -  DATA AVAILABLE OUT
                                ;
                                ; WRITING INTO THE KITS DATA BUFFER REGISTER SHOULD RAISE " DATA AVAILABLE ",
                                ;
                                ; THE CSR SHOULD = 126410
                                ;
                                ; 15 SPECIAL CONDITION ( 1 HERE BY INPUT DEMAND)
                                ; 13 INPUT DEMAND ( 1 HERE BECAUSE INPUT=1, AND IN=0)
                                ; 11 DATA AVAILABLE OUT (BECAUSE OF THE COMMAND CLR @ DBR)
                                ; 10 INPUT ( 1 HERE BY DATA AVAILABLE OUT)
                                ;  8 LAST DIRECTION (=OUTPUT)
                                ;= 7 READY ( 0 BECAUSE IN=0, OUT=0, AND DATA AVAILABLE = 1)
                                ;  3 STAT (BECAUSE BIT 2 FNCT=0)
                                ;
                                ; NO INTERUPTS OR DMA CYCLES ARE EXPECTED TO OCCUR
                                ;
001642  104400                  T51:  SCOPE
                                ;
001644  005077  177334                CLR @ CSR                 ; THE CSR IS DEFAULT TO X2 1 0
                                ;
001650  012777  177776  177322        MOV #-2, @ WC             ; SET THE WORD COUNT R = 177776
001656  005077  177328                CLR @ BAR                 ; SET THE BUS ADDRESS R = 0
001662  005077  177322                CLR @ DBR                 ;=WRITE 0'S INTO THE DATA BUFFER R
                                ;
001666  004737  005406                JSR PC, STALL             ; 10.5 US TIME PER TRANSFER
                                ;
001672  027727  177306  126410        CMP @ CSR, #126410
001700  001401                       BEQ .+4
001702  104438                  E38: ERROR+38
001704  104777                       SUBSCOPE                  ; LOOP
                                ;
001706  027727  177266  177776        CMP @ WC, #-2
001714  001401                       BEQ .+4
001716  104431                  E31: ERROR+31                  ; THE WORD COUNT R NOT STILL = -2
001720  104777                       SUBSCOPE                  ; LOOP
                                ;
001722  005777  177262                TST @ DBR
001726  001401                       BEQ .+4
001730  104434                  E34: ERROR+34                  ; THE DATA BUFFER REGISTER NOT = 0
001732  104777                       SUBSCOPE
                                ;
001734  005777  177242                TST @ BAR
001740  001401                       BEQ .+4
001742  104432                  E32: ERROR+32                  ; THE BUS ADDRESS R NOT STILL = 0
001744  104400                       SCOPE                     ; LOOP
```

```
                                        ; BECAUSE # IN #, AND # OUT # WERE BOTH PREVIOUSLY # 0,
                                        ; CLEARING BIT 11 (DATA AVAILABLE OUT), SETS BIT 7 (READY)

                                        ; THE CSR SHOULD # 122618

                                        ; 15 SPECIAL CONDITION ( 1 BECAUSE INPUT DEMAND=1)
                                        ; 13 INPUT DEMAND ( 1 BECAUSE INPUT=1, AND IN=0)
                                        ; 12 INPUT ( 1 BECAUSE OF DATA AVAILABLE OUT)
                                        ;  8 LAST DIRECTION (=OUTPUT)
                                        ;  7 READY ( 1 BECAUSE IN=0, OUT=0, AND DATA AVAILABLE OUT=0)
                                        ;  3 STAT ( 1 BECAUSE BIT 2 FNCT=0)

001746  142777  000010  177232          BICB #10, @ CSR1        ; CLEAR BIT 11 (DATA AVAILABLE OUT)

001754  027727  177224  122610          CMP @ CSR, #122610
001762  001401                         BEQ .+4
001764  104433                 E33:    ERROR+33                ; CSR NOT # 122610
001766  104400                         SCOPE

                                        ; CLEARING BIT 12 (INPUT) WILL CLEAR BIT 13 (INPUT DEMAND), AND
                                        ; BIT 15 (SPECIAL CONDITION)

001770  142777  000004  177210          BICB #4, @ CSR1         ; CLEAR BIT 12 (INPUT)

001776  027727  177202  000610          CMP @ CSR, #610
002004  001401                         BEQ .+4
002006  104435                 E35:    ERROR+35                ; CSR NOT # 610
002010  104400                         SCOPE
```

```
                                ;                       TEST 7 - DBR
                                ;
                                ; WITH THE DIAGNOSTIC INPUT-OUTPUT CABLE CONNECTED, WRITING INTO THE
                                ; DATA BUFFER REGISTER WILL PLACE ON THE DATA LINES OUT OF THE KIT THE
                                ; SAME DATA, THEREFORE READING FROM THE DATA BUFFER R SHOULD HEAD THE
                                ; SAME DATA WRITTEN.
                                ;
                                ; THE CONTENTS OF R1 WILL CONTAIN THE DATA TO BE WRITTEN:
                                ;
                                ; 1
                                ; 2
                                ; 4
                                ; 10
                                ; 20
                                ; 40
                                ; 100
                                ; 200
                                ; 400
                                ; 1000
                                ; 2000
                                ; 4000
                                ; 10000
                                ; 20000
                                ; 40000
                                ; 100000
                                ;
                                ; THE DATA BUFFER HAS PREVIOUSLY BEEN TESTED IN TEST 5 FOR ALL 0'S
                                ;
002812  005077  177166  T7:     CLR @ CSR               ; THE CSR DEFAULTS TO 610
                                ;
002816  012701  000001          MOV #1, R1              ; R1 IS A WORK REGISTER
                                ;
002822  104400          T7L:    SCOPE                   ; SETS (PCSCOPE) = .+2
                                ;
002824  010177  177160          MOV R1, @ CBR           ;@WRITE INTO THE KITS DATA BUFFER R
                                ;
003030  004737  005406          JSR PC, STALL
                                ;
003034  020177  177150          CMP R1, @ CBR
003040  001401                  BEQ .+4
003042  104458          E50:    ERROR+50                ; THE DATA READ NOT = TO THE DATA WRITTEN
003044  104777                  SUBSCOPE                ; SCOPE LOOP IF (ERRORS) > 0
                                ;
003046  006301                  ASL R1                  ; SHIFT THE CONTENTS OF R1 FOR NEXT BIT
003050  001364                  BNE T7L                 ; UNTIL (R1) = 0
```

E3

```
                                ; TEST 8 (OUTPUT PART 1) ; 1=NPR (OUT) W/O WORD COUNT OVERFLOW
                                ;
                                ; NO INTERUPTS ARE EXPECTED
                                ;
002052  104400                  T8:     SCOPE
                                ;
002054  005077  177124                  CLR @ CSR               ; CSR DEFAULT TO X2 1 0
                                ;
002060  012777  177776  177112          MOV #-2, @ WC           ; SET THE WC R = 177776
002066  012777  005752  177106          MOV #BUFFER, @ BAR      ; SET THE BUS ADDRESS R = ADDRESS: BUFFER
002074  052777  000002  177102          BIS #2, @ CSR           ; SET BIT 1 (OUT)
                                ;
002102  004737  005406                  JSR PC, STALL           ; 10.5 US TIME PER TRANSFER
002106  004737  005406                  JSR PC, STALL
                                ;
                                ;
                                ; CSR SHOULD = 126412
                                ;
                                ; 15 SPECIAL CONDITION ( 1 HERE FROM INPUT DEMAND)
                                ; 13 INPUT DEMAND ( 1 BECAUSE INPUT=1, AND IN 0)
                                ; 11 DATA AVAILABLE OUT ( 1 BECAUSE OUT=1)
                                ; 10 INPUT (BECAUSE DA OUT SET DA IN)
                                ;  8 LAST DIRECTION (=OUTPUT)
                                ;  3 STAT ( 1 BECAUSE FNCT=0)
                                ;  1 OUT
                                ;
                                ; WC SHOULD = -1
                                ; BAR SHOULD = #BUFFER+2
                                ;
002112  027727  177866  126412          CMP @ CSR, #126412
002120  001401                          BEQ .+4
002122  104457                  E57:    ERROR 57                ; CSR NOT = 126412
002124  104777                          SUBSCOPE
                                ;
002126  022777  177777  177044          CMP #-1, @ WC
002134  001401                          BEQ .+4
002136  104460                  E60:    ERROR 60                ; WC NOT = 177777
002140  104777                          SUBSCOPE
                                ;
002142  022777  005754  177032          CMP #BUFFER+2, @ BAR
002150  001401                          BEQ .+4
002152  104461                  E61:    ERROR 61                ; BAR NOT = #BUFFER+2
```

```
                                ; TEST 9 (OUTPUT PART 2) ; 1-NPR (OUT) WITH WORD COUNT OVFLO
                                ;
                                ; CSR SHOULD = 127418
                                ;
                                ; 15 SPECIAL CONDITION ( 1 BECAUSE OF INPUT DEMAND)
                                ; 13 INPUT DEMAND ( 1 BECAUSE INPUT=1, AND IN=0)
                                ; 11 DATA AVAILABLE OUT
                                ; 10 INPUT ( 1 BECAUSE OF DATA AVAILABLE OUT)
                                ;  9 LAST XFER ( 1 BECAUSE OF WORD COUNT OVERFLOW)
                                ;  8 LAST DIRECTION ( =OUTPUT)
                                ;  7 READY ( 1 BECAUSE OUT=0 FROM WORD COUNT OVERFLOW)
                                ;  3 STAT ( 1 BECAUSE BIT 2 FNCT = 0)
                                ;
                                ; WC SHOULD = 0
                                ; BAR SHOULD = #BUFFER+2
                                ;
002154  104400                  T9:     SCOPE
                                ;
002156  005077  177022                  CLR @ CSR               ; CSR DEFAULTS TO X2 1 0
                                ;
002162  012777  177777  177010          MOV #-1, @ WC           ; SET WC = 177777
002170  012777  005752  177004          MOV #BUFFER, @ BAR      ; SET BAR = #BUFFER
002176  052777  000002  177000          BIS #2, @ CSR           ; SET BIT 2 (OUT) = 1
                                ;
002204  004737  005406                  JSR PC, STALL           ; 10.5 US TIME PER TRANSFER
                                ;
002210  027727  176770  127410          CMP @ CSR, #127410
002216  001401                          BEQ .+4
002220  104402                  E62:    ERROR+62                ; CSR NOT = 127410
002222  104777                          SUBSCOPE
                                ;
002224  005777  176750                  TST @ WC
002230  001401                          BEQ .+4
002232  104403                  E63:    ERROR+63                ; WC NOT = 0
002234  104777                          SUBSCOPE
                                ;
002236  022777  005754  176736          CMP #BUFFER+2, @ BAR
002244  001401                          BEQ .+4
002246  104404                  E64:    ERROR+64                ; BAR NOT = #BUFFER+2
002250  104400                          SCOPE
```

```
                                        ; CLEARING BIT 11 (DATA AVAILABLE OUT) CLEARS BIT 9 (LAST TRANSFER), AND
                                        ; BECAUSE " IN " AND " OUT " WERE BOTH PREVIOUSLY = 0, SETS BIT 7 (READY)
                                        ;
                                        ; THE CSR SHOULD = 122610
                                        ;
                                        ; 15 SPECIAL CONDITION ( 1 BECAUSE INPUT DEMAND)
                                        ; 13 INPUT DEMAND ( 1 BECAUSE INPUT=1, AND IN=0)
                                        ; 10 INPUT ( 1 BECAUSE OF DATA ABAILABLE OUT)
                                        ;  9 LAST DIRECTION ( =OUTPUT)
                                        ;  7 READY ( 1 BECAUSE IN=0, OUT=0, AND DATA AVAILABLE OUT=0)
                                        ;  3 STAT ( 1 BECALSE BIT 2 FNCT = 0)
                                        ;
002252  142777  000010  176726                BICB #10, @ CSR1        ; CLEAR BIT 11 (DATA AVAILABLE OUT)
                                        ;
002260  027727  176720  122610                CMP @ CSR, #122610
002266  001401                               BEQ .+4
002270  104465                       E65:    ERROR+65                ; CSR NOT = 122610
```

```
                                        ;                       TEST 10 (INPUT)
                                        ;
                                        ; SET BIT 0 (IN) OF THE CSR THEN RAISE DATA AVAILABLE OUT BY WRITING INTO THE DBR
                                        ;
002272  104400                  T10:    SCOPE
                                        ;
002274  005077  176704                  CLR @ CSR               ; CSR = X2 1 2
                                        ;
002300  012777  005752  176674          MOV #BUFFER, @ BAR      ; BUS ADDRESS R = #BUFFER
002306  012777  177776  176664          MOV #-2, @ WC           ; WORD COUNT R = 177776
002314  005277  176664                  INC @ CSR               ; SET BIT 0 (IN)
002320  005077  176664                  CLR @ DBR               ;*RAISE DATA AVAILABLE OUT
                                        ;
002324  004737  005406                  JSR PC, STALL           ; 10.5 US TIME PER TRANSFER
                                        ;
                                        ; THE CSR SHOULD = 11
                                        ;
002330  004537  005412                  JSR R5, XCSR
002334  000011                          11
002336  104478                  E70:    ERROR*70
002340  104777                          SUBSCOPE
                                        ;
                                        ; THE BUS ADDRESS SHOULD = #BUFFER+2
                                        ;
002342  022777  005754  176632          CMP #BUFFER+2, @ BAR
002350  001401                          BEQ .+4
002352  104472                  E72:    ERROR*72
002354  104777                          SUBSCOPE
                                        ;
                                        ; THE WORD COUNT REGISTER SHOULD = -1
                                        ;
002356  022777  177777  176614          CMP #-1, @ WC
002364  001401                          BEQ .+4
002366  104474                  E74:    ERROR*74
```

```
                                        ;       TEST 11 (OUTPUT AND INPUT SIMULTANEOUSLY)
                                        ;
                                        ; SET BITS 0 (IN), AND 1 (OUT) OF THE CSR SIMULTANEOUSLY
                                        ;
002370  104400                  T11:    SCOPE
                                        ;
002372  005077  176686                  CLR @ CSR               ; CSR = X2 1 0
                                        ;
002376  012777  177776  176574          MOV #-2, @ WC           ; SET WORD COUNT R = 177776
002404  012777  005752  176570          MOV #BUFFER, @ BAR      ; SET BUS ADDRESS R = #BUFFER
002412  052777  000003  176564          BIS #3, @ CSR           ; SET OUT, AND IN
                                        ;
002420  004737  005406                  JSR PC, STALL           ; 10.5 US TIME PER TRANSFER
002424  004737  005406                  JSR PC, STALL           ; 10.5 US TIME PER TRANSFER
                                        ;
                                        ; THE CSR SHOULD = 210
                                        ;
002430  004537  005412                  JSR R5, XCSR
002434  000210                          210
002436  104000                  E1001:  ERROR+100
002440  104777                          SUBSCOPE
                                        ;
                                        ; THE BAR SHOULD = #BUFFER+4
                                        ;
002442  022777  005756  176532          CMP #BUFFER+4, @ BAR
002450  001401                          BEQ .+4
002452  104002                  E1021:  ERROR+102
002454  104400                          SCOPE
                                        ;
002456                          X11=.
```

```
                        .NLIST SEQ,LOC
                        ;               TEST 12 - DATA TEST (PART 1)
                        ; WRITE THE DATA PATTERN AS INDICATED BELOW
                        ;       BUFFER          WORD#           PATTERN
 003752                 ;       BUFFER          ;1      INPUT   0
 003756                 ;       BUFFER+4        ;2      INPUT   1
 003762                 ;       BUFFER+10       ;3      INPUT   2
 003766                 ;       BUFFER+14       ;4      INPUT   4
 003772                 ;       BUFFER+20       ;5      INPUT   10
 003776                 ;       BUFFER+24       ;6      INPUT   20
 004002                 ;       BUFFER+30       ;7      INPUT   40
 004006                 ;       BUFFER+34       ;8      INPUT   100
 004012                 ;       BUFFER+40       ;9      INPUT   200
 004016                 ;       BUFFER+44       ;10     INPUT   400
 004022                 ;       BUFFER+50       ;11     INPUT   1000
 004026                 ;       BUFFER+54       ;12     INPUT   2000
 004032                 ;       BUFFER+60       ;13     INPUT   4000
 004036                 ;       BUFFER+64       ;14     INPUT   10000
 004042                 ;       BUFFER+70       ;15     INPUT   20000
 004046                 ;       BUFFER+74       ;16     INPUT   40000
 004052                 ;       BUFFER+100      ;17     INPUT   100000
                        ;       .LIST SEQ,LOC
768
761             002456          .=X11
```

```
763  002456  012701  003752          T12:    MOV #BUFFER, R1
764  002462  005000                          CLR R0                  ; FIRST WORD = 0 (AND C BIT)
765  002464  000261                          SEC                     ; SET CARRY BIT IN PROCESSOR STATUS WORD
766
767  002466  010021                  T12L:   MOV R0, (R1)+           ; BUFFER, BUFFER+4, BUFFER+6, ETC
768  002470  012721  000000                  MOV #0, (R1)+           ; MOV INST SO NOT TO CLEAR C BIT WITHIN PSW
769  002474  006100                          ROL R0
770  002476  001373                          BNE T12L
771
772  002500  104400                          SCOPE                   ; SETS (PCSCOPE) = .+2
773
774  002502  005077  176476                  CLR @ CSR               ; CSR = X2 1 0
775
776  002506  012777  177736  176464          MOV #-42, @ WC          ; 17 WORDS TO TRANSFER X 2
777  002514  012777  003752  176460          MOV #BUFFER, @ BAR
778  002522  052777  000003  176454          BIS #3, @ CSR           ; SET IN, AND OUT
779
780  002530  012701  177736                  MOV #-42, R1            ; STALL FOR 17 WORDS (OUT+IN)
781
782  002534  005777  176440          1$:     TST @ WC
783  002540  001402                          BEQ 2$
784  002542  005201                          INC R1
785  002544  001373                          BNE 1$
786
787                                  ; ALL 34 WORDS SHOULD HAVE BEEN TRANSFERRED OUT THEN IN BY NOW
788
789                                  ; THE CSR SHOLD = 210
790
791  002546  004537  003412          2$:     JSR R5, XCSR
792  002552  000210                          210
793  002554  104510                  E110:   ERROR+110
794  002556  104777                          SUBSCOPE
795
796                                  ; COMPARE THE DATA
797
798                                  ; THE CONTENTS OF BUFFER SHOULD = THE CONTENTS OF BUFFER+2
799
800                                  ; ...ETC...
801
802  002560  012701  003752                  MOV #BUFFER, R1
803  002564  004737  003432          XT12L:  JSR PC, COMPARE
804  002570  001004                          BNE E112
805  002572  022701  006056                  CMP #BUFFER+184, R1
806  002576  001372                          BNE XT12L
807  002600  000401                          BR OK112
808
809  002602  104512                  E112:   ERROR+112
810  002604  104400                  OK112:  SCOPE
811
812          002606                  X12=.
```

```
                        .NLIST SEQ,LOC
                        ;       TEST 13 - DATA TEST (PART 2)
                        ; WRITE THE DATA PATTERN AS INDICATED BELOW
                        ;       BUFFER          WORD#           PATTERN

005752                  BUFFER          #1              177776
005756                  BUFFER+4        #3              1
005762                  BUFFER+10       #5              177775
005766                  BUFFER+14       #7              2

005772                  BUFFER+20       #11             177773
005776                  BUFFER+24       #13             4
006002                  BUFFER+30       #15             177767
006006                  BUFFER+34       #17             10

006012                  BUFFER+40       #21             177757
006016                  BUFFER+44       #23             20
006022                  BUFFER+50       #25             177737
006026                  BUFFER+54       #27             40

006032                  BUFFER+60       #31             177677
006036                  BUFFER+64       #33             100
006042                  BUFFER+70       #35             177577
006046                  BUFFER+74       #37             200

006052                  BUFFER+100      #41             177377
006056                  BUFFER+104      #43             400
006062                  BUFFER+110      #45             176777
006066                  BUFFER+114      #47             1000

006072                  BUFFER+120      #51             175777
006076                  BUFFER+124      #53             2000
006102                  BUFFER+130      #55             173777
006106                  BUFFER+134      #57             4000

006112                  BUFFER+140      #61             167777
006116                  BUFFER+144      #63             10000
006122                  BUFFER+150      #65             157777
006126                  BUFFER+154      #67             20000

006132                  BUFFER+160      #71             130000
006136                  BUFFER+164      #73             40000
006142                  BUFFER+170      #75             077777
006146                  BUFFER+174      #77             100000

862                     .LIST SEQ,LOC
863             002606  .=X12
```

```
865  002606  012701  005752          T13:    MOV #BUFFER, R1
866  002612  012700  000001                  MOV #1, R0
867  002616  005100                  1$:     COM R0
868  002620  010021                          MOV R0, (R1)+
869  002622  005021                          CLR (R1)+
870  002624  005100                          COM R0
871  002626  010021                          MOV R0, (R1)+
872  002630  005021                          CLR (R1)+
873  002632  006300                          ASL R0
874  002634  001370                          BNE 1$
875
876  002636  104400                          $COPE
877
878  002640  005077  176340                  CLR @ CSR
879
880  002644  012777  177700  176326          MOV #-100, @ WC          ; 64 WORDS (32 OUT/32 IN)
881  002652  012777  005752  176322          MOV #BUFFER, @ BAR
882  002660  052777  000003  176316          BIS #3, @ CSR
883
884  002666  012701  177000                  MOV #-1000, R1
885
886  002672  005777  176302          2$:     TST @ WC
887  002676  001402                          BEQ 3$
888  002700  005201                          INC R1
889  002702  001373                          BNE 2$
890
891                                  ; ALL 32 X 2 WORDS SHOULD HAVE BEEN XFERRED BY NOW
892
893                                  ; THE CSR SHOULD = 210
894
895  002704  004537  005412          3$:     JSR R5, XCSR
896  002710  000210                          210
897  002712  104516                  E116:   ERROR+116
898  002714  104777                          SUBSCOPE
899
900                                  ; COMPARE DATA
901
902  002716  012701  005752                  MOV #BUFFER, R1
903  002722  004737  005432          XT13L:  JSR PC, COMPARE
904  002726  001004                          BNE E120
905  002730  022701  006152                  CMP #BUFFER+200, R1
906  002734  001372                          BNE XT13L
907  002736  000401                          BR T14
908
909  002740  104520                  E120:   ERROR+120
```

```
                                        ;               TEST 14 - INTR ENABLE
                                        ;               (NO INTERUPTS SHOULD OCCUR)
                                        T14:    SCOPE
                                                MOV #E131, @#IRQOK      ; PC RETURN IF UNEXPECTED IRQ
                                                BICB #340, @#PSW        ; SET PSW PRIORITY = 0
                                                MOV #100, @#176216      ; SET BIT 6 (ENB)
                                                JSR PC, STALL
                                                BR E131+2
                                        E131:   ERROR 131
                                                SUBSCOPE
                                        ; THE CSR SHOULD = X3 1 0
                                                JSR R5, XCSR
                                                310
                                        E130:   ERROR 130
                                                SUBSCOPE
                                        ; CLEAR BIT 6 (ENB) OF THE CSR
                                                BIC #100, @#CSR
                                                JSR R5, XCSR
                                                210
                                        E132:   ERROR 132
```

```
                                ;       TEST 15 - INTR ENABLE PART 1 (FROM WORD COUNT OVERFLOW)
                                ;
                                ; TEST THE KITS (BB11-D) VECTOR ADDRESS
                                ;
                                ; THE VECTOR ADDRESS UP TO THIS TEST IS UNKNOWN
                                ;
                                ; SET IRQ=1 TO EXPECT A  KIT INTERUPT
                                ;
                                ;               (FROM ANY PRIO LEVEL FOR NOW)
                                ;
                                ;               THEN ANY " IOT " TRAP IS VALAD
                                ;
                                T15:    SCOPE
                                ;
                                        CLR @ CSR
                                ;
                                        MOV #E148+2, IRQOK      ; SETUP FOR PC RETURN IF INTERRUPT OK
                                        BICB #340, PSW          ; PRIORITY LEVEL = 0
                                        BIS #1, IRQ             ; PRIME FOR EXPECTED KIT INTERRUPT
                                        MOV #-2, @ WC
                                        MOV #BUFFER, @ BAR
                                        MOV #103, @ CSR         ; SET BIT 6 (ENB)
                                ;
                                        JSR PC, STALL
                                        JSR PC, STALL
                                ;
                                ;
                                ; AN INTERUPT SHOULD HAVE OCCURED BY NOW
                                ;
                                E148:   ERROR+148
                                        SUBSCOPE
                                ;
                                ; THE CSR SHOULD = 318
                                ;
                                        JSR R5, XCSR
                                        318
                                E142:   ERROR+142
```

```
                                ;TEST 16 - INTERRUPT ENABLE PART 2 (FROM SPECIAL CONDITION)
                                
003126  104400          T16:    SCOPE
                                
003130  012737  003202  001236        MOV #E143+2, IRQOK
003136  142737  000340  177776        BICB #340, PSW
003144  052737  000001  001234        BIS #1, IRQ
003152  005077  176822                CLR @ WC
003156  005077  176820                CLR @ BAR
003162  012777  000100  176814        MOV #100, @ CSR
003170  005077  176814                CLR @ DBR
                                
003174  004737  003406                JSR PC, STALL
                                
                                ; AN INTERRUPT SHOULD HAVE OCCURED BY NOW
                                
003200  104543          E143:   ERROR+143
003202  104777                  SUBSCOPE
                                
                                ; THE CSR SHOULD = 126110
                                
                                ; 15 SPECIAL CONDITION ( 1 BECAUSE OF INPUT DEMAND)
                                ; 13 INPUT DEMAND ( 1 BECAUSE INPUT = 1, AND IN = 0)
                                ; 11 DATA AVAILABLE OUT
                                ; 10 INPUT ( 1 BECAUSE DATA AVAILABLE OUT)
                                ;  0 LAST DIRECTION = X
                                ;+ 7 READY ( 0 BECAUSE IN=0, OUT=0, AND DATA AVAILABLE OUT=1)
                                ;  3 STAT ( 1 BECAUSE FNCT = 0)
                                
003204  004537  003412                JSR R5, XCSR
003210  126110                        126110
003212  104544          E144:   ERROR+144
```

```
1015                                    ;       TEST 17 - PRIORITY LEVEL INTERUPTS 5 TO 7 EXPECTED
1016
1017                                    ;           (THE KIT SHOULD INTERUPT IN PL-5 )
1018
1019  003214  104400                    T17:    SCOPE
1020
1021  003216  005077  175762                    CLR @ CSR
1022
1023  003222  012737  003300  001236            MOV #E146+2, IRQOK      ; FOR PC RETURN IF INTERRUP OK
1024  003230  112737  000200  177776            MOVB #200, PSW          ; SET PRIORITY LEVEL = 4
1025  003236  052737  000001  001234            BIS #1, IRC
1026  003244  012777  177776  175726            MOV #-2, @ WC
1027  003252  012777  005752  175722            MOV #BUFFER, @ BAR
1028  003260  012777  000103  175716            MOV #103, @ CSR         ; SET BIT 6 (ENB)
1029
1030  003266  004737  005406                    JSR PC, STALL
1031  003272  004737  005406                    JSR PC, STALL
1032
1033                                    ; AN INTERUPT SHOULD HAVE OCCURED BY NOW
1034
1035  003276  104546                    E146:   ERROR+146
1036
1037                                    ;       TEST 18 - PRIORITY LEVEL 5 ONLY
1038
1039                                    ; SETTING THE PSW TO = PRIORITY = 5, NO INTERRUPT SHOULD OCCUR
1040
1041  003300  104400                    T18:    SCOPE
1042
1043  003302  005077  175676                    CLR @ CSR
1044
1045  003306  012737  003314  001236            MOV #E152, IRQOK        ; FOR PC RETURN IF UNEXPECTED IRQ
1046  003314  112737  000240  177776            MOVB #240, PSW          ; SET PRIORITY LEVEL = 5
1047  003322  052737  000001  001234            BIS #1, IRC
1048  003330  012777  177776  175642            MOV #-2, @ WC
1049  003336  012777  005752  175636            MOV # BUFFER, @ BAR
1050  003344  012777  000103  175632            MOV #103, @ CSR         ; SET BIT 6 (ENB)
1051
1052  003352  004737  005406                    JSR PC, STALL
1053  003356  004737  005406                    JSR PC, STALL
1054
1055  003362  000401                            BR E152+2               ; OK, NO INTERUPT OCCURED
1056
1057                                    ; NO INTERUPTS SHOULD HAVE OCCURED
1058
1059  003364  104552                    E152:   ERROR+152
```

```
                                        ;              TEST 19 - LAST TRANSFER/XOF
                                        T19:    SCOPE
003370  005077  175610                          CLR @ CSR
003374  012777  177777  175576                  MOV #-1, @ WC
003402  012777  005752  175572                  MOV #BUFFER, @ BAR
003410  052777  000002  175566                  BIS #2, @ CSR
003416  004737  005406                          JSR PC, STALL
003422  027727  175556  127410                  CMP @ CSR, #127410
003430  001401                                  BEQ .+4
003432  104568                          E168:   ERROR*168
003434  104777                                  SUBSCOPE
                                        ; NOW WITHOUT CLEARING LAST TRANSFER, SET " IN "
                                        ; THEREBY RAISING EXTERNAL OVERFLOW (XOF)
003436  012777  177776  175534                  MOV #-2, @ WC
003444  005277  175534                          INC @ CSR
003450  004737  005406                          JSR PC, STALL
003454  004537  005412                          JSR R5, XCSR
003460  110210                                  110210
003462  104562                          E162:   ERROR*162
003464  104400                                  SCOPE
                                        ; TO CLEAR " XOF " WRITE INTO THE LOW BYTE OF THE CSR
003466  052777  000010  175510                  BIS #10, @ CSR          ; TRY TO WRITE BIT 3 (STAT)
003474  004537  005412                          JSR R5, XCSR
003500  000210                                  210
003502  104564                          E164:   ERROR*164
                                        ; IF DATA SWITCH 13 = 1 DO NOT HALT AT THE END OF THIS PASS
003504  104400                                  SCOPE
003506  032737  020000  177570                  BIT #SW13, SWR
003514  001401                                  BEQ .+4
003516  000000                                  HALT
003520  005237  001242                          INC PASSLSB
003524  000137  001266                          JMP CONTINUE
```

```
1110                                    ; STARTING ADDRESS 2S2
1111
1112                                    ; CONVERSATION BETWEEN 2 OR MORE BB11-D'S
1113
1114            003752                  OUTBUFFER=BUFFER                 ; O-UTPUT BUFFER
1115            003652                  INBUFFER=OUTBUFFER-100           ; I-NPUT BUFFER
1116  003530    177740                  KWC:    -40                      ; OUTPUT WC (CONSTANT AND RESTORE)
1117  003532    003752                  KBAR:   OUTBUFFER                ; OUTPUT BAR (CONSTANT AND RESTORE)
1118
1119  003534    112737  000340  177776  S2S2:   MOVB #340, PSW
1120
1121  003542    012701  003752                  MOV #BUFFER, R1
1122  003546    012700  000001                  MOV #1, R0
1123  003552    005100          1$:             COM R0
1124  003554    010021                          MOV R0, (R1)+
1125  003556    005100                          COM R0
1126  003560    010021                          MOV R0, (R1)+
1127  003562    006300                          ASL R0
1128  003564    001372                          BNE 1$
1129
1130  003566    012737  100000  001234          MOV #SW15, IRQ           ; IRQ = 100000
1131  003574    012737  000110  001240          MOV #110, LINECOUNT      ; INITIALIZE LINECOUNT FOR 72 CHARACTERS
1132  003602    005037  001242                  CLR PASSLSB
1133  003606    005037  001230                  CLR ERRORS               ; PRIME FOR ANY ERROR COUNT
1134  003612    012737  177737  004144          MOV #-41, INWC           ; "INPUT" WORD COUNT -41
1135  003620    012737  003652  004146          MOV #INBUFFER, INBAR     ; "INPUT" BUSS ADDRESS REGISTER BUFFER
1136
1137  003626    012737  177740  003530  XS2S2: MOV #-40, KWC
1138  003634    012737  003752  003532          MOV #OUTBUFFER, KBAR
1139
1140                                    ; IF THE STATUS BIT = 0, THE OTHER KIT IS NOT READY, WAIT
1141
1142  003642    104400                  OUT:    SCOPE
1143
1144  003644    012706  001200                  MOV #1200, SP
1145  003650    112777  000004  175326          MOVB #4, @ CSR
1146
1147  003656    004737  004300                  JSR PC, WAIT2S2
1148
1149  003662    013777  003530  175310          MOV KWC, @ WC
1150  003670    001756                          BEQ XS2S2                ; (KWC) = 0
1151  003672    013777  003532  175302          MOV KBAR, @ BAR
1152  003700    012737  003736  001236          MOV #ODIRECTION, IRQOK
1153  003706    052737  000001  001234          BIS #1, IRQ
1154  003714    052777  000102  175262          BIS #102, @ CSR          ; " OUT "
1155  003722    112737  000200  177776          MOVB #200, PSW           ; PRIORITY 4
1156
1157  003730    004737  004300          2S1:    JSR PC, WAIT2S2
1158  003734    000775                          BR 2S
```

```
1160                                    ; A PRIORITY LEVEL # 5 INTERRUPT OCCURED FROM THE BB11-D
1161
1162                                    ; IF " INPUT DEMAND " = 1, THEN STOP TRANSMISSIONS AND INPLT DATA
1163
1164                                    ; IF " NEX " THEN ERROR 202
1165
1166  003736  032777  020000  175240    ODIRECTION: BIT #SW13,@ CSR
1167  003744  001410                           BEQ 1$                  ; INPUT DEMAND = 0
1168
1169                                    ; IF SW 11 = 1, IGNORE ALL INPUT DEMANDS UNTIL WORD COUNT OVERFLOW (READY = 1)
1170
1171  003746  032737  004000  177570            BIT #SW11,SWR
1172  003754  001412                           BEQ IN                  ; SW11 = 0
1173  003756  105777  175222                   TSTB @ CSR              ; WAIT FOR BIT 07 (READY)
1174  003762  100375                           BPL .-4
1175  003764  000406                           BR IN
1176
1177  003766  005777  175212            1$:    TST @ CSR
1178  003772  100315                           BPL XS202               ; MUST BE WORD COUNT OVERFLOW
1179
1180                                    ; * * * ERROR * * *
1181
1182                                    ; NOT " INPUT DEMAND " OR  WORD COUNT OVERFLOW
1183
1184                                    ; THE BB11-D MUST HAVE TRIED TO ADDRESS NON EXTENDED MEMORY
1185
1186  003774  104402                    E202:  ERROR*202               ; " SPECIAL CONDITION " = 1,
1187  003776  104777                           SUBSCOPE                 ; MUST BE = NEX " BECAUSE DIRECTION WAS = OUT
1188                                           ; THEREFORE " XOF " COULD NOT CAUSE SC
1189  004000  000712                           BR XS202
```

```
1191                                    ; INPUT DEMAND = 1
1192
1193                                    ; SAVE THE OUT WORD COUNT, AND THE OUT BUS ADDRESS REGISTER
1194
1195                                    ; ENABLE " IN "
1196
1197  004002  042777  000102  175174  IN1:    BIC #102, @ CSR         ; DISABLE " OUT"
1198
1199  004010  017737  175164  003530          MOV @ WC, XWC
1200  004016  017737  175160  003532          MOV @ BAR, XBAR
1201  004024  013777  004144  175146          MOV INWC, @ WC          ; WC = -41 (MAX), XOF SHOULD OCCUR AT -40
1202  004032  013777  004146  175142          MOV INBAR, @ BAR
1203  004040  012737  004152  001236          MOV #IDIRECTION, IRQOK
1204  004046  052737  000001  001234          BIS #1, IRQ
1205  004054  112777  000101  175122          MOVB #101, @ CSR
1206  004062  112737  000200  177776          MOVB #200, PSW          ; PRIORITY 4
1207
1208                                    ; IF AFTER 10 US THE CONTENTS OF THE WORD COUNT REGISTER IS NOT EQUIV-
1209                                    ; ALENT TO THE CONTENTS+2 OF THE WORD COUNT REGISTER PRIOR TO THE
1210                                    ; STALL, THE DATA WAS STOPPED COMING IN
1211
1212  004070  017737  175104  004150  1S1:    MOV @ WC, XWC
1213  004076  004737  003406          JSR PC, STALL           ; 10 US
1214  004102  004737  003406          JSR PC, STALL           ; 10 US
1215  004106  023777  004150  175864          CMP XWC, @ WC
1216  004114  001365                  BNE 1S                  ; DATA STILL COMING IN
1217
1218                                    ; DATA STOPPED COMING IN, SAVE "INWC", AND "INBAR"; THEN TRANSMIT
1219
1220  004116  042777  000101  175860          BIC #101, @ CSR
1221  004124  017737  175850  004144          MOV @ WC, INWC
1222  004132  017737  175844  004146          MOV @ BAR, INBAR
1223
1224  004140  000137  003642                  JMP OUT
1225
1226  004144  177737                  INWC:   -41
1227  004146  006052                  INBAR:  INBUFFER
1228  004150  000000                  XWC:    0
```

```
1230                                    ; A PRIORITY LEVEL 5 INTERRUPT OCCURED FROM THE BB11-D
1231
1232                                    ; IF # XOF # = 1, COMPARE DATA
1233
1234                                    ; IF NOT XOF, THEN MUST BE " NEX ", ERROR 286
1235
1236                                    ; ALL DATA MUST BE "IN", RESET "INWC", AND "INBAR"
1237
1238  004192  012737  177737  004144  IDIRECTION: MOV #-41, INWC
1239  004200  012737  004852  004146        MOV #INBUFFER, INBAR
1240  004206  032777  010000  175010        BIT #SW12, @ CSR
1241  004214  001004                       BNE ICOMPARE            ; XOF = 1, COMPARE DATA
1242
1243  004216  104406                E286:   ERROR#286               ; INPUT ADDRESSED NON EXTENDED MEMORY
1244  004220  104777                       SUBSCOPE                ; XOF=0, THEREFORE ILLEGAL NCO=1
1245
1246  004222  000137  003642                JMP OUT
1247
1248  004226  012701  005750        ICOMPARE: MOV #OUTBUFFER-2, R1
1249  004232  005721                ICLOOP: TST (R1)+               ; ADD #2, R1
1250  004234  010137  005346                MOV R1, BADDRESS
1251  004240  011137  005462                MOV (R1), GOOD
1252  004244  016137  000100  005456        MOV 100(R1), BAD
1253  004252  026111  000100                CMP 100(R1), (R1)
1254  004256  001004                       BNE E218
1255  004260  022701  004850                CMP #BUFFER+70, R1
1256  004264  001362                       BNE ICLOOP
1257  004266  000401                       BR OK218
1258
1259  004270  104410                E218:   ERROR#218               ; DATA ERROR (GOOD=BAD)
1260  004272  104777                OK218:  SUBSCOPE
1261
1262                                    ; DATA SWITCH 13 = 1 TO HALT AT END OF RECEIVE DATA COMPARE
1263
1264  004274  032737  020000  177570        BIT #SW13, SWR
1265  004302  001404                       BEQ 1$
1266  004304  042777  000004  174712        BIC #4, @ CSR
1267  004312  000000                       HALT
1268  004314  000137  003642          1$:   JMP OUT
1269
1270                                    ; WAIT HERE FOR THE KIT TO BECOME READY, OR FOR ANY INTERRUPT
1271
1272  004320  004737  005466        WAIT282: JSR PC, STALL          ; 10 US
1273  004324  032777  000010  174672        BIT #10, @ CSR
1274  004332  001412                       BEQ 1$
1275  004334  005201                       INC R1
1276  004336  001370                       BNE WAIT282
1277  004340  032737  010000  177570        BIT #SW12, SWR
1278  004346  001364                       BNE WAIT282
1279  004350  004537  005006                JSR R5, PRINT
1280  004354  005577                       MWAITING
1281  004356  000760                       BR WAIT282
1282  004360  000207                1$:   RTS PC
```

```
1284                                    ; TRAP (ERROR) CATCHES THE CODES 104400 TO 104777
1285
1286  004342  011637  001232      ERRORCATCHER: MOV @ SP, ERRORX  ; (ERRORX) = X ADDRESS * 2
1287  004346  162737  000002  001232          SUB #2, ERRORX      ; (ERRORX) = X ADDRESS
1288  004354  017737  174652  001232          MOV @ ERRORX, ERRORX  ; (ERRORX) = CONTENTS OF THE X ADDRESS
1289  004362  042737  104400  001232          BIC #ERROR, ERRORX  ; SAVING BITS 7 THRU 0
1290
1291                                    ; IF (ERRORX) = 0 = SCOPE
1292                                    ; IF (ERRORX) = 377 = SUBSCOPE
1293                                    ; IF (ERRORX) = 1 TO 376 = SOME KIND OF KIT11D ERROR
1294
1295  004370  001527                          BEQ XSCOPE          ; " SCOPE "
1296  004372  022737  000377  001232          CMP #377, ERRORX
1297  004400  001535                          BEQ XSSCOPE         ; " SUBSCOPE "
1298
1299                                    ; (ERRORX) = 1 TO 376 = A KIT11D ERROR
1300
1301                                    ; SET " PCSSCOPE " JUST IN CASE THE PC (NOW RESIDING WITHIN THE SP)
1302                                    ; IS EQUIVALENT TO THE ADDRESS OF X ERROR+2
1303
1304  004402  011637  001246                  MOV @ SP, PCSSCOPE
1305  004406  062737  000002  001246          ADD #2, PCSSCOPE    ; NOW LOOKS LIKE X SUBSCOPE+2, OR X SCOPE+2
1306  004414  005237  001238      INCERRORS: INC ERRCNT           ; +1 TO THE ERROR TABULATOR
1307  004420  001775                          BEQ INCERRORS       ; SO ERRORS DO NOT = 0
1308
1309                                    ; IF ERRORS ARE ANY CODE 200, WRITE THE " FNCT " BIT = 0
1310
1311                                    ; (I.E, MAKING  ..THIS.. KIT NOT READY FOR S202)
1312
1313  004422  032737  000100  001232          BIT #100, ERRORX
1314  004430  001007                          BNE 1$
1315  004432  032737  000200  001232          BIT #200, ERRORX
1316  004440  001403                          BEQ 1$
1317
1318  004442  042777  000004  174534          BIC #4, @ CSR
```

```
1328                                    ; DATA SWITCH 12 = 0, PRINT APPROPRIATE ERROR MESSAGE
1329
1330  004490  032737  010000  177570  15$:    BIT #SW12, SWR
1331  004496  001006                         BNE 3$
1332  004500  004537  005006                 JSR R5, PRINT
1333  004504  005212                         MENUMBER
1334  004506  004537  005100                 JSR R5, TYPEOCTAL
1335  004512  000172                         ERRORX                  ; (ERRORX) IS THE OPERAND TO BY OCTALIZED
1336  004514  004537  005006                 JSR R5, PRINT
1337  004520  005220                         MSCOPE                  ; STARTING ADDRESS OF " SCOPE " LOOP
1338  004524  004537  005100                 JSR R5, TYPEOCTAL
1339  004530  000174                         PCSCOPE
1340  004532  004537  005006                 JSR R5, PRINT
1341  004536  005230                         MPASS                   ; PASS # IN OCTAL (177777 MAX)
1342  004540  004537  005100                 JSR R5, TYPEOCTAL
1343  004544  000176                         PASSL$B
1344  004546  004537  005006                 JSR R5, PRINT
1345  004552  005241                         MITERATION              ; ITERATION THROUGH ERROR SCOPE LOOP
1346  004554  004537  005100                 JSR R5, TYPEOCTAL
1347  004560  000200                         ERRORS
1348
1349                                    ; IF THE ERROR IS #112, OR E128, OR E218, ALSO PRINT DATA ERROR (GOOD=BAD)
1350
1351  004562  022737  000112  001232         CMP #112, ERRORX
1352  004570  001410                         BEQ 2$                  ; E112
1353  004572  022737  000128  001232         CMP #128, ERRORX
1354  004600  001404                         BEQ 2$                  ; E128
1355  004602  022737  000210  001232         CMP #210, ERRORX
1356  004610  001022                         BNE 3$
1357
1358  004612  004537  005006          2$:    JSR R5, PRINT           ; E210
1359  004616  005251                         MBUFFER
1360  004620  004537  005100                 JSR R5, TYPEOCTAL
1361  004624  005460                         BADDRESS
1362  004626  004537  005006                 JSR R5, PRINT
1363  004632  005262                         MGOOD
1364  004634  004537  005100                 JSR R5, TYPEOCTAL
1365  004640  005462                         GOOD
1366  004642  004537  005006                 JSR R5, PRINT
1367  004646  005271                         MBAD
1368  004650  004537  005100                 JSR R5, TYPEOCTAL
1369  004654  005456                         BAD
1370
1371                                    ; DATA SWITCH 15 = 0 HALT AT DETECTION OF ERROR
1372
1373  004656  032737  100000  177570  3$:    BIT #SW15, SWR
1374  004664  001001                         BNE .+4
1375  004666  000000                         HALT
1376  004670  000002                         RTI
```

```
1370                                    ;               * SCOPE *
1371
1372  004650  009737  001230          XSCOPE: TST ERRORS
1373  004654  001015                          BNE SCOPING             ; TO A SCOPE LOOP
1374
1375                                    ; NO ERRORS HAVE BEEN DETECTED HERE,
1376                                    ; JUST SET PCSCOPE = THE FIRST ADDRESS OF THE SCOPE LOOP
1377                                    ; (IN CASE ANY ERRORS ARE EVER DETECTED LATER)
1378
1379  004656  009037  001230          NOSCOPE: CLR ERRORS
1380  004662  011637  001244                   MOV @ SP, PCSCOPE
1381  004666  000002                          RTI
1382
1383                                    ;               * SUBSCOPE *
1384
1385  004670  009737  001230          XSSCOPE: TST ERRORS
1386  004674  001001                          BNE .+4
1387  004676  000002                          RTI                     ; NO ERRORS EXIST - EXIT
1388
1389                                    ; ERRORS DO EXIST
1390
1391                                    ; IF THIS ERROR ADDRESS IS THE SAME ADDRESS WITHIN THE PROGRAM
1392                                    ; LOCATION " PCSSCOPE ", THEN THIS IS A SCOPE LOOP, IF NOT - EXIT
1393
1394  004700  021637  001246                   CMP @ SP, PCSSCOPE
1395  004704  001401                          BEQ .+4
1396  004706  000002                          RTI                     ; IT ISN'T - EXIT
1397
1398                                    ; THIS IS A SCOPING LOOP
1399
1400  004710  032737  040000  177570  SCOPING: BIT #SW14, SWR
1401  004716  001397                          BNE NOSCOPE
1402  004720  013716  001244                  MOV PCSCOPE, @ SP
1403  004724  000002                          RTI                     ; EXIT
```

MY

```
1405  004726  000000                  VA:     0                       ; KIT11D VECTOR ADDRESS
1406
1407                                  ; IF THE CONTENTS OF PROGRAM LOCATION .IRQ. = 1, ANY IOT TRAP IS VALAD,
1408                                  ; AND IT IS ASSUMED TO BE A KIT VECTOR ID TRAP.
1409
1410  004730  011637  004726          KITCATCHER: MOV @SP, VA         ; KIT 11D VECTOR ADDRESS
1411  004734  162737  000004  004726          SUB #4, VA              ; BECAUSE PC=VA+2+2
1412  004742  012746  000340                  MOV #340, -(SP)         ; RESET PRIORITY = 7 (AT EXIT)
1413  004746  013746  001236                  MOV IRQOK, -(SP)        ; RETURN PC
1414  004752  005737  001234                  TST IRQ
1415  004756  100010                          BPL 1$
1416
1417                                  ; PRINT THE KIT'S VECTOR ADDRESS ON THE TTY
1418
1419  004760  004537  005006                  JSR R5, PRINT
1420  004764  005464                          .VECTOR                 ; ADDRESS OF TEXT MESSAGE
1421  004766  004537  005100                  JSR R5, TYPEOCTAL
1422  004772  004726                          VA                      ; VA (VECTOR ADDRESS) TO BE TYPED AS OCTAL
1423  004774  004737  005312                  JSR PC, CRLF
1424  005000  005037  001234          1$:     CLR IRQ                 ; RESET IRQ
1425  005004  000002                          RTI
```