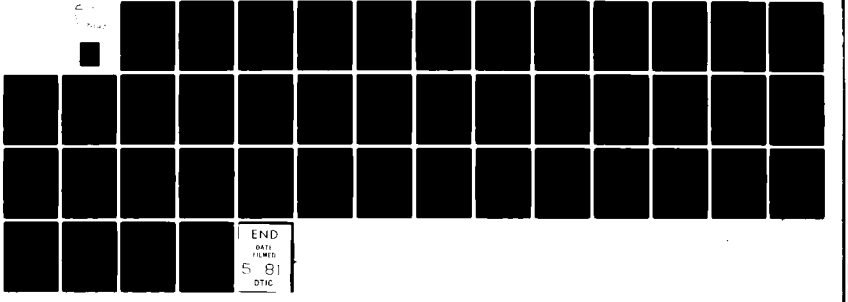
AD-A097 897 GENERAL ELECTRIC CO PITTSFIELD MA ORDNANCE SYSTEMS F/G 9/2
ELECTRICAL CHARACTERIZATION OF SINGLE CHIP MICROPROCESSORS AND --ETC(U)
FEB 81 B W HAJDUK, T M OSTROWSKI, D TUCKER F30602-78-C-0189
UNCLASSIFIED RADC-TR-81-10 NL



END
DATE FILMED
5 81
DTIC

17. Verify the RESET function. Before reset is applied all circuitry affected by it should be in the state opposite to what it will be after reset.

18. Check flag or status registers after all instructions to verify that each bit can be set or reset by each op code that should affect it and that some instructions to not affect it.

19. Verify that the circuitry that generates the status flags operates properly. For example, to check the zero flag each bit in the accumulator should assume the one state while all others are zero.

20. Verify conditional branch logic by attempting jumps, calls, returns, etc. for both the true and false condition of the flags. The flags not under test should be such that, if they are incorrectly selected, the operation is not performed.

21. Verify interrupts are acknowledged if they are applied at the appropriate sample time and ignored otherwise. Also verify nesting and priority of interrupts if they exist.

22. Verify that the device handles illegal op codes properly - i e ignores and fetches next instruction, performs trap, etc - if possible. For some devices it is not possible to test illegal op codes because their effect on the device is not known or specified.

23. If device has decimal adjust instruction verify that the circuitry that decodes the upper and lower digits operates properly and use all of the constants that are added to the digits. This has to be performed after both add and subtract instructions.

24. Verify that all constants generated by the device are used. These include decimal adjust constants, constants used for offsets, etc.

25. If the device has a ROM for program memory execute the program or as a minimum verify the contents of the ROM.

26. Verify any special features of the instructions. These are determined by reviewing the device data sheet or description of the instructions. Examples are counting and termination of repeated instructions, non acceptance of interrupts during enable interrupt instructions, etc.

In performing the preceding tests it is important to have the results visible; i.e., available on output pins. It does no good to apply a set of conditions to an internal circuit if the outputs that result from these conditions cannot be observed.

During test development it would be advantageous to list the various tests in tables and check them off as they are performed. Since the number of tables required would vary with the device under test, they are not specified here.

It is important to remember that this is a general guideline and that not all areas apply to all devices and some may require expansion to accommodate device differences.

APPENDIX A

CHECKLIST FOR THE 8048 AND 8035L MICROPROCESSORS

I. 64 x 8 Resident RAM Array (Working Registers)

A. Perform #15 in "Procedure for LSI Testing" (PLT)

II. Conditional Branch Logic

A. Conditional branches must be executed for both a true and false condition. The flags not under test should be such that, if they are incorrectly selected the opposite operation is performed.

III. Timer/Event Counter

A. Verify counting mode from both T1 and ALE ÷ 32 by incrementing counter from both of these.

B. Verify overflow flag (JUMP on both timer flag and interrupt).

C. Verify 8 bit counter.

1. Presettable (Write and Read 0-over-0, 0-over-1, 1-over-0, and 1-over-1)

2. Readable (Read above preset patterns)

3. Increment and verify 0, 1, 3, 7, ---255, 256.

IV. Timing/Circuits

A. The ability to switch between input and output mode must be verified. The disable mode, where T0 is an input, is checked during the verification of the branch instruction. The enable mode, where T0 is the clock output, must be verified by comparing the output voltage levels. The ability to switch between these two modes must also be verified.

V. Instruction Register and Decoder

A. Execute all instruction types
(#2 and 13 of PLT)

VI. Registers and Latches (#10 and 14 of PLT)

A. Port 1 and Port 2

B. Bus latch

C. Accumulator

D. Temporary register

E. Program status word (PSW)

1. Stack pointer logic

VII. Program Counter

A. The ability to read and write a 0-over-0, 0-over-1, 1-over-0, and 1-over-1 in every bit in the program counter register must be verified. The reading and writing operations in the program counter are performed by the CALL (PC→ Working register) and the RETURN (Working register→ PC) instructions respectively.

B. The verification of the counting circuitry (ripple counter) is accomplished by applying every combination of the data and carry-in to every bit. For example a four bit counter must increment, and verify the results, of the following data:

0 0 0 0
0 0 0 1
0 0 1 1
0 1 1 1
1 1 1 1

VIII. INTERRUPT and Timer Overflow Interrupt Circuitry (#3 of PLT)

A. Ensure that the microprocessor acknowledges interrupts and that the functions of the interrupt are executed correctly. The disable mode should also be verified; that is, the interrupts should be disabled, interrupts applied, and then the microprocessor should be checked to ensure that the interrupt was not acknowledged in any way.

In both modes, the functions affected by the interrupt should be such that acknowledging the interrupt will change the normal operations of the microprocessor.

IX. Verify Operating Modes

A. RESET

All the reset states in the microprocessor must be verified. These include the following functions:

1. Sets program counter to zero
2. Sets stack pointer to zero
3. Selects register bank 0
4. Selects memory bank 0
5. Sets BUS to high impedance state (except when EA = 5V)
6. Sets Ports 1 and 2 to input mode
7. Disables interrupts (timer and external)
8. Stops timer
9. Clears timer flag
10. Clears F0 and F1
11. Disables clock output from T0

In order to verify the reset mode, these functions should not be in their reset states if reset were not acknowledged. In other words, the program counter should not be at its maximum count, stack pointer should not be at zero, selected register and memory banks should not be 0, etc. when the reset is applied for verifying its operation.

B. Single Step

1. Ensure that in this mode the processor is halted after executing each instruction.

C. Power Down

The storage capability of the 64 x 8 RAM must be verified after the power down mode. The RAM contents should contain worst case data for this mode of operation. Worst case is applicable to both supply current and storage capability.

D. External Access

1. This mode is verified during the initial testing where all the instructions are inputted externally.

E. Read Internal Program Memory

1. This mode will be verified during the verification of the resident ROM where all the addresses will be read.

X. ALU Operation

A. ADD operation; #7 of PLT

B. AND, OR, and EXOR; #8 and #11 of PLT

C. Left and right shifts with and without carry;
#9 and #11 of PLT

D. CLR

1. Ensure that all accumulator bits can be cleared, and ensure that other instructions were not executed, such as complement instruction, instead of the clear.

E. SWAP

Uses a transparent temporary register.
Use #14 of PLT for each bit.

F. INCREMENT

Increment and verify execution on the following data:
(00, 01, 03, 07, 0F, 1F, 3F, 7F, FE, FF)

G. DECREMENT

Decrement and verify execution on the following data:
(00, 01, 02, 04, 08, 10, 20, 40, 80, FF)

H. DECIMAL ADJUST

To verify the operation of this instruction the following conditions should be converted:

Adjustment of least significant digit;

1. With AC = 0 the lower 4 bits of accumulator must contain the following digits

a) A or B,
b) C or D,
c) 8 or 9, and
d) any number between 2 and 7.

2. With AC = 1 the lower 4 bits of accumulator will be 0, 1, or 2

Adjustment of most significant digit after least significant digit has been converted:

1. With C = 0 the upper 4 bits of accumulator must contain the following digits

    a) A or B,
    b) C or D,
    c) 8 or 9, and
    d) any number between 2 and 7.

2. With C = 1 the upper 4 bits of accumulator will be 0, 1, 2, or 3.

Independent circuitry is used to implement the decimal adjust for both the most and least significant digits in the 8048 microprocessor. This independence must be verified. This is accomplished by executing the decimal adjust instructions with condition 2 and either condition 1c or 1d for the other digit. This condition should be tested for both digits. The remaining test conditions may be tested in any combination.

There are two conditions which may be advantageous in detecting unique faults. One is the application of conditions 1a and 1b on the lower digit and the value of 9 in the most significant digit prior to executing the decimal adjust instruction. The other is performing a decimal adjust with 1b and the value of F in the most significant digit prior to executing the decimal adjust instruction.

I. Transparent Temporary Registers (#10 of PLT as much as possible)

There are two transparent temporary registers used for executing ALU instructions

Data from internal data bus is loaded into TEMP1 and constants into TEMP2. Instructions and constants which utilized TEMP2 are:

Instruction	Constants (loaded into TEMP2)
INCREMENT	01
DECREMENT	01
DECIMAL ADJUST	06 and 60

XI. Resident ROM (1K x 8)

A. Read all the memory locations in the resident ROM in the Read Internal Program Memory mode outlined in the "User's Manual"

B. Execute part of program in resident ROM to verify this mode of operation. Otherwise the EA input is never low during testing.

C. Execute all testing at maximum frequency to verify maximum frequency of operation in the microprocessor and access time of memory.

XII. Miscellaneous Verifications

A. IC Pin Independence

1. #1 in PLT

B. High Impedance of Bus & Port Lines

1. #4 in PLT

C. Independence of Each Data Line

1. #5 in PLT

APPENDIX B

TESTS REQUIRED TO CHECK THE Z80

I. Reset function

A. Verify that reset

1. Forces the program counter to zero
2. Disables the interrupt enable flip-flop
3. Sets the I and R registers to zero
4. Sets interrupt mode 0
5. Places the address and data bus in the high impedance state and all control output signals in the inactive state.

II. ALU operation

A. Verify the following functions

1. Add and subtract
2. Increment and decrement
3. AND, OR, EXOR
4. Compare
5. Left and right shifts and/or rotates
6. Set, reset, and test bits

III. Conditional branch logic

A. Execute all jumps for both a true and a false condition. The flags not under test should be such that, if they are incorrectly selected the jump is not performed.

IV. Flag Register

A. Verify the integrity and independence of each bit.

B. Verify that each bit can be set and/or reset by each op code that should affect it.

C. Verify the operation of the circuitry that generates each of the flags.

D. Verify that certain op codes do not affect the flag register.

V. Interrupt circuitry

A. Verify that interrupts are acknowledged properly and ignored if they occur outside of an acceptable sample time.

B. Verify the enable and disable interrupt instructions.

C. Verify interrupt acknowledge modes 0, 1, and 2.

D. Verify nesting of interrupts.

E. Verify priority of $\overline{\text{BUSRQ}}$, $\overline{\text{NMI}}$, and $\overline{\text{INT}}$ inputs.

VI. Registers

A. Verify the integrity and independence of the bits in all registers.

B. Verify the exchange operation.

VII. Control logic

A. Perform all instructions at least once.

B. Verify operation of all addressing modes.

C. Verify special features of op codes

1. Compare - match and not match

2. String instructions - counts and whether all are necessary

3. Decimal adjust - constants added for various results

4. Shift instructions - that vacated bit positions are properly filled

5. Any others that exist

D. Halt mode and methods of exit

E. Refresh function

1. Generation of refresh address

2. Generation of $\overline{\text{RFSH}}$ signal

F. Stack pointer operation

VIII. Miscellaneous

A. Data path independence not previously checked

B. IC pin independence (adjacent pins as a minimum)

C. Tristate capability of outputs at times other than reset

APPENDIX C

8086 TEST PROGRAMS

This appendix contains the programs that were developed and used during the characterization of the 8086. The following describes the use of each program:

1. LEARN.EDT:86, pages C-2 to C-5, is used to learn the 8086 vectors using an SDK-86 single board computer as the stimulus.

2. LEARNP.EDT:86, page C-6, is the control program for using the PRAM start when ready mode.

3. CONVER.EDT:86, pages C-7 to C-12, is used to convert the patterns obtained from the learn program into mode 3 patterns for the 8086 functional test.

4. HEADER.PAT:86, pages C-13 and C-14, is used to generate the header information for the pattern file.

5. 8086.EDT:86, pages C-15 to C-30, is the test program used to characterize the 8086. It uses a GO/NOGO functional test with worst case input timing conditions. The outputs are strobed at the vendor specified delays and pass/fail data is recorded as temperature, Vcc, frequency, duty cycle, and input levels are varied.

6. 8086.PIN:86, pages C-31 and C-32, is the 8086 pin assignment program used with the test program.

```
1.0200 *
1.0300 *                TEKTRONIX S-3260 TEST PROGRAM
1.0400 *
1.0500 *                   CIRCUIT TEST ENGINEERING
1.0600 *              GENERAL ELECTRIC ORDNANCE SYSTEMS
1.0700 *                   PITTSFIELD, MASSACHUSETTS
1.0800 *
1.0900 *
1.1000 * . DUT PART OR DWG. #       :8086
1.1100 * . DUT DESCRIPTION          :16 BIT MICROPROCESSEOR
1.1200 * . DATE                     :6-MAY-80
1.1300 * . PROGRAMMER               :T.J.METZEL
1.1400 *
1.1500 * .  THIS PROGRAM IS USED TO LEARN 8086 VECTORS USING A
1.1600 * .  INTEL SDK-86 SINGLE BOARD COMPUTER AS THE STIMULUS
1.1700 *
1.1800 * .  THE PRAM IS USED IN A START-WHEN-READY MODE TO
1.1900 * .  RECOGNIZE THE ADDRESS OF THE FIRST OP CODE ON
1.2000 * .  THE ADDRESS BUS OF THE 8086
1.2010 * .
1.2100 * .   PATTERNS ARE STORE USING PATOUT TO LUN 2
1.2200 * .   THE VECOUT PROGRAM CONVERTS VECTORS TO A
1.2300 * .   HEX CODE AND PRINTS THEM TO THE LINE PRINTER
1.2400 * .
```

```
1.2500 * FOR SUPPORTING DOCUMENTS, CONSULT ETEC AND OTHER
1.2600 * CABINET FILES UNDER THE FOLLOWING IDENTIFIED TEST
1.2700 * SPECIFICATION NUMBER AND ADAPTER NUMBERS, AS WELL AS
1.2800 * LISTINGS OF THE FOLLOWING IDENTIFIED DISK OR MAGNETIC
1.2900 * TAPE FILES.
1.3000 *
1.3100 *
1.3200 * . TEST SPECIFICATION      :MIL-M-38510/530
1.3300 * . TEST TYPE/CONDITIONS    :CHARACTERIZATION
1.3400 * . PIN ASSIGNMENT FILE     :LEARN
1.3500 * . PAT FILE                :LEARN
1.3600 * . PRAM OBJECT FILE        :LEARNP
1.3700 * . RAM FILE                :LEARN
1.3800 * . THIS LISTING IS         :TEKTEST EDIT
1.3900 * . SOCKET CARD ASSEMBLY #  :2224

2.0100 * ************************************************
2.0200 *  TABLE OF CONTENTS
2.0300 * ************************************************
2.0400
2.0500 * SECTION      TEST
2.0600 *
2.0700 *    1         HEADER INFORMATION
2.0800 *    2         TABLE CONTENTS
2.0900 *    3         PINLIST
2.1000 *    4         SUBROUTINES AND FUNCTIONS
2.1100 *    5         CONSTANT DEFINITIONS
2.1200 *    6         ARRAY DECLARATIONS
2.1300 *   10         LOAD PRAM
2.1400 *   11         LEARN ROUTINE
2.1500 *   12         VECTOR STORAGE
```

```
3.0100 * ********************************************************
3.0200 *  PINLIST ASSIGNMENTS
3.0300 * ********************************************************
3.0400
3.0500 * PINLIST AD1=AD15I,AD14I,AD13I,AD12I,AD11I,AD10I /
3.0600 * AD9II,AD8I,AD7I,AD6I,AD5I,AD4I,AD3I,AD2I,AD1I,AD0I
3.0700   PINLIST AD0=AD15O,AD14O,AD13O,AD12O,AD11O,AD10O /
3.0800   AD9O,AD8O,AD7O,AD6O,AD5O,AD4O,AD3O,AD2O,AD1O,AD0O
3.0900   PINLIST A=A19,A18,A17,A16
3.1000   PINLIST ADR=A,AD0
3.1100   PINLIST INS=CLK,RESET,HOLD,TEST,READY,NMI,INTR,HLDA
3.1200   PINLIST OUTS=BHE,RD,WR,MIO,DTR,ALE,DEN,INTA,MN
3.1300   PINLIST ALLPINS=ADR,INS,OUTS
3.1400   PINLIST ALL=ADR,RESET,HOLD,TEST,READY,NMI,INTR,HLDA,B
         HE /
3.1500   RD,WR,MIO,DTR,ALE,DEN,INTA,MN

4.0100 * ********************************************************
4.0200 *  SUBROUTINES AND FUNCTION DECLARATIONS
4.0300 * ********************************************************
4.0400
4.0500   FUNCTION GETBIT(I,V,V,V):BARRAY
4.0600   FUNCTION RCALL(0)
4.0700
4.0800   SUBROUTINE BARRAY(I,V,V,V):BARRAY
4.0900   SUBROUTINE SREAD(P,V,V,V,V,I,V):SREAD
4.1000   SUBROUTINE PATOUT(N,V,I,V,V):PATIO
4.1100   SUBROUTINE RSTART(V),RLOADFILE(N,F),RHALT(V,V),STOPPR
         AM(0):PRMSUB

5.1000 * ************************************
5.2000 *  DEVICE SPECIFICATION CONSTANTS
5.3000 * ************************************
5.4000
5.5000   TCLCL= (406NS-80NS)
5.6000   START=120NS
5.7000   WINDOW= 20NS
5.8000   PATROW=512
5.9000   DLUN=2

6.1000 * ************************************
6.2000 *  ARRAY DECLARATIONS
6.3000 * ************************************
6.4000 *
6.5000   IARRAY DATA1(1283)
6.6000   BARRAY(DATA1,1,36,512)
```

```
10.0100 * ************************************
10.0200 *          LOAD PRAM                 *
10.0300 * ************************************
10.0400 *
10.0500   RHALT(0,#1777)
10.0600   RLOADFILE(EOD,LEARN)
10.0700   IF(NOT EOD) 10.1
10.0800   PRINT"FILE NOT FOUND-LEARN.RAM",CR
10.0900   STOP
10.1000   CONTINUE

11.0100 * ************************************
11.0200 *          LEARN ROUTINE             *
11.0300 * ************************************
11.0400 *
11.0500   CONNECT OUTPUT TO COMPARATOR ON ALL
11.0600   HICOMPARE = 2.4V ON ALL
11.0700   PHASE 11 = START FOR WINDOW
11.0800   PHASE 12 = START FOR WINDOW
11.0900   LOCOMPARE = .8V ON ALL
11.1000   PHASE 9 = START FOR WINDOW
11.1100   PHASE 10 = START FOR WINDOW
11.1200   COMPARE ALL WITH PATTERN
11.1300   CYCLE = TCLCL,EXTERNAL SYNC
11.1400   WHEN CALL 12.1
11.1410   RSTART(#0000)
11.1500   MOVE PRAM(0) TO ALL AND SAVE ERRORS
11.1600   IF(1<RCALL<6) 11.15,11.17
11.1700   PRINT "HALT ENCOUTERED WITH PRAM-PROGRAM ABORTED",CR
11.1800   STOP

12.1000 * ************************************
12.2000 *  VECTOR STORAGE
12.3000 * ************************************
12.4000
12.5000   SREAD(ALL,0,0,5,PATROW,DATA1,1)
12.6000   PATOUT(ERR,DLUN,DATA1,1,PATROW)
12.7000   IF(ERR) 12.8,12.9
12.8000   PRINT "ERROR ENCOUNTERD WITH PATOUT",CR
12.9000   CONTINUE

13.1000   ACCEPT"IF YOU WANT TO CONTINUE TYPE 1",T,CR
13.2000   IF(T) 13.5,13.3
13.3000   STOPPRAM
13.4000   STOP
13.5000   RETURN
```

```
DK0:LEARNP.EDT;86   PRAM V02.24  20-NOV-80  13:15  PAGE  1
DK0:LEARN.PAT;86

                                1.1000 *  8086 CONTROL PROGRAM USING THE S
                                          TART-WHEN-READY MODE
                                1.2000 *
                                1.3000    PINLIST ALLPIN = A19,A18,A17,A16,
                                          AD15Q,AD14Q,AD13Q/
                                1.4000    AD12Q,AD11Q,AD10Q,AD9Q,AD8Q,AD7Q,
                                          AD6Q,AD5Q,AD4Q,AD3Q /
                                1.5000    AD2Q,AD1Q,AD0Q,RESET,HOLD,TEST,RE
                                          ADY,NMI,INTR,HLDA,SHE /
                                1.6000    RD,WR,M1O,DTR,ALE,DEN,INTA,MN

      040000 000000             2.1000              DEFAULT IS ADVANCE SHIFT
                                                    REGISTER.

0000  000014 000777             3.1000    START:  LOAD 511 TO RA,HOLD SR.
                                1.0200    0000000000 0000000000 0000000000
0001  000060 000002             3.2000              LOAD FAIL TO TRAP,HOLD SR
                                                    .
                                1.0300    0000000000 0000000000 0000000000
0002  000320 000000             3.3000    FAIL:     CLEAR ERROR,TEST,TRAP ON
                                                    ERROR,HOLD SR.
                                1.0400    0000000001 1000000000 0001000011
                                3.4000 *  MATCH FOUND-RECORD VECTORS
0003  040210 120003             3.5000              TEST,DEC RA,IF RA NEZ GOT
                                                    O 3.
                                1.0500    0000000000 0000000000 0000000000
0004  003000 000000             3.6000              CALL(3),HOLD SR.
                                1.0600    0000000000 0000000000 0000000000
0005  000000 010005             3.7000              IF MOVE FLAG EQZ GOTO 5,H
                                                    OLD SR.
                                1.0700    0000000000 0000000300 0000000000
0006  000000 100000             3.8000              GOTO START,HOLD SR.
                                1.0800    0000000000 0000000000 0000000000


ERRORS DETECTED: 0.
PROGRAM LENGTH : 7.
PATTERN WIDTH HAS BEEN TRUNCATED ON LISTING ONLY.

EDT FILE NAME: DK0:LEARNP.EDT;86
PAT FILE NAME: DK0:LEARN.PAT;86
PIN FILE NAME: DK0:LEARN.PIN;86
RAM FILE NAME: DK0:LEARN.RAM;86
```

```
CONVER.EDT;86   SAUGAO V0.3            DISK NAME:  ETEC-24 TJW
DATE  20-NOV-80      TIME  08:17       PAGE    1 OF    6

    1.0100    PRINT<COLUM> ERASE
    1.0200 *
    1.0300 *                TEKTRONIX 8-3260 TEST PROGRAM
    1.0400 *
    1.0500 *                   CIRCUIT TEST ENGINEERING
    1.0600 *               GENERAL ELECTRIC ORDNANCE SYSTEMS
    1.0700 *                   PITTSFIELD, MASSACHUSETTS
    1.0800 *
    1.0900 *
    1.1000 * . DUT PART OR DWG. #       :8086
    1.1100 * . DUT DESCRIPTION          :16 BIT MICROPROCESSEOR
    1.1200 * . DATE                     :6-MAY-80
    1.1300 * . PROGRAMMER               :T.J.WETZEL
    1.1400 *
    1.1500 *
    1.1600 * THIS PROGRAM CONVERTS PATTERNS STORED USING
    1.1700 * THE LEARN PROGRAM INTO MODE 3 PATTERNS FOR
    1.1800 * USE IN THE 8086 FUNCTION TEST.
    1.1900 *
    1.2000 * BEFORE USING THIS CONVERSION PROGRAM THE FOLLOWING
    1.2100 * STEPS MUST BE DONE:
    1.2200 *       1. COLUMNS CONTAINING BIDIRECTIONAL DATA MUST
    1.2300 *          BE DUPLICATED.THIS IS DONE BY FOLLOWING STEPS
    1.2400 *                   EDIT
    1.2500 *                   INPUT FILENAME
    1.2600 *                   COLSEQ 5-20
    1.2700 *                   WIDTH,16
    1.2800 *                   SAVE NEW FILENAME
    1.2900 *                   EXIT
    1.3000 *         2.THE DUPLICATED COLUMNS MUST BE ADDED TO THE
    1.3100 *           ORIGINTAL PATTERN.
    1.3200 *                   EDIT
    1.3300 *                   INPUT FILENAME
    1.3400 *                   WIDTH,52
```

```
1.3500 *                    MERGE NEW FILENAME,37-52
1.3600 *                    SAVE FILENAME
1.3700 *          3.ASSIGN LUN 2 TO THE INPUT FILE AND LUN 3
1.3800 *            TO AN OUTPUT FILE AND RUN THIS PROGRAM.
1.3900 *
1.4000 *          4. HEADER INFORMATION AND PROPER RESET TIMING
1.4100 *            PATTERN CAN BE EASILY ADDED USING PAPER TAPE
1.4200 *                    BACKUP
1.4300 *                    OUTPUT DEVICE:PP
1.4400 *                    FILE NAME:HEADER.PAT
1.4500 *                    EDIT
1.4600 *                    INPUT FILENAME
1.4700 *                    ERASE 1,5
1.4800 *                    READ,TAPE
1.4900 *                    RESEQ
1.5000 *                    SAVE
1.5100 *
1.5200 * FOR SUPPORTING DOCUMENTS, CONSULT ETEC AND OTHER

1.5300 * CABINET FILES UNDER THE FOLLOWING IDENTIFIED TEST

1.5400 * SPECIFICATION NUMBER AND ADAPTER NUMBERS, AS WELL AS

1.5500 * LISTINGS OF THE FOLLOWING IDENTIFIED DISK OR MAGNETIC

1.5600 * TAPE FILES.

1.5700 *

1.5800 *

1.5900 * . TEST SPECIFICATION      :MIL-M-38510/530

1.6000 * . TEST TYPE/CONDITIONS    :CHARACTERIZATION

1.6100 * . PIN ASSIGNMENT FILE     :8086

1.6200 * . PATTERN FILE(S)         :PAT

1.6300 * . THIS LISTING IS         :TEKTEST EDIT

1.6400 * . SOCKET CARD ASSEMBLY #  :2024


2.0100 * ************************************************
2.0200 *  TABLE OF CONTENTS
2.0300 * ************************************************
```

```
2.0400
2.0500 * SECTION      TEST
2.0600 *
2.0700 *    1         HEADER INFORMATION
2.0800 *    2         TABLE CONTENTS
2.0900 *    3         PINLIST
2.1000 *    4         SUBROUTINES AND FUNCTIONS
2.1100 *    5         CONSTANT DEFINITIONS
2.1200 *    6         ARRAY DECLARATIONS
2.1300 *    7         CONTROL BIT DEFINITIONS
2.1400 *    8         HEADING PRINT ROUTINE

4.0100 * ************************************************
4.0200 *  SUBROUTINES AND FUNCTION DECLARATIONS
4.0300 * ************************************************
4.0400
4.0500   FUNCTION GETBIT(I,V,V,V):BARRAY
4.0600   FUNCTION GETROW(I,V,V,V):BARRAY
4.0700   SUBROUTINE SETBIT(V,I,V,V,V):BARRAY
4.0800   SUBROUTINE SETROW(V,I,V,V,V):BARRAY
4.0900   SUBROUTINE BARRAY(I,V,V,V):BARRAY
4.1000   SUBROUTINE SREAD(P,V,V,V,V,I,V):SREAD
4.1100   SUBROUTINE PATOUT(N,V,I,V,V):PATIO
4.1200   SUBROUTINE PATIN(N,V,I,V,V):PATIO
4.1300   SUBROUTINE CRDATE(V),CRTIME(V):TIME

6.1000 * ************************************
6.2000 *  ARRAY DECLARATIONS
6.3000 * ************************************
6.4000 *
6.5000   IARRAY DATA(3333)
6.6000   BARRAY(DATA,2,52,512)

14.1000 * ************************************************
14.2000 *      VECTOR RECOVERY
14.3000 * ************************************************
14.4000 *
14.5000   PATIN(ERR,2,DATA,1,512)
14.6000   IF(ERR) 14.7,14.8
14.7000   PRINT "ERROR ENCOUNTERED WITH PATIN",CR
14.8000   CONTINUE

15.1000 * ************************************************
15.2000 *       CONVERT COLUMNS 1-20 TO H'S AND L'S
15.3000 * ************************************************
15.4000   LOOP 15.8 J=1,512
15.5000   LOOP 15.7 I=1,20
```

```
15.6000    SETBIT(1,DATA,1,I,J)
15.7000    CONTINUE
15.8000    CONTINUE

16.1000 *  ************************************************************
16.2000 *       CONVERT COLUMNS 27-35 TO H'S AND L'S
16.3000 *  ************************************************************
16.4000    LOOP 16.8 J=1,512
16.5000    LOOP 16.7 I=27,35
16.6000    SETBIT(1,DATA,1,I,J)
16.7000    CONTINUE
16.8000    CONTINUE

20.0100 *  ************************************************************
20.0200 *      MODIFY FORCING DATA
20.0300 *  ************************************************************
20.0400    FLAG=0
20.0500    LOOP 20.2 J=1,512
20.0600 *  CHECK RD BIT TO DETERMINE DIRECTION OF DATA
20.0700 *  IF A READ IS BEING PERFORMED THEN LEAVE
20.0800 *  THE DATA ON FOR 3 CYCLES OTHERWISE
20.0900 *  INHIBIT THE DRIVERS
20.1000    IF(GETBIT(DATA,2,29,J)) 20.13,20.11
20.1100    FLAG=FLAG+1
20.1200    GOTO 20.2
20.1300    IF(FLAG EQ 2) 20.14,20.16
20.1400    FLAG=0
20.1500    GOTO 20.2
20.1600    LOOP 20.19 I=37,52
20.1700    SETBIT(1,DATA,1,I,J)
20.1800    SETBIT(0,DATA,2,I,J)
20.1900    CONTINUE
20.2000    CONTINUE

21.0100 *  ************************************************************
21.0200 *    MASK COMPARE DATA WHEN BUS FLOATS
21.0300 *  ************************************************************
21.0400    LOOP 21.12 J=1,512
21.0500 *  CHECK WRITE
21.0600    IF(NOT(GETBIT(DATA,2,30,J))) 21.12
21.0700 *  ALE CHECK
21.0800    IF(GETBIT(DATA,2,33,J)) 21.12
21.0900 *  CHANGE DATA COMPARE PATTERN TO MASK
21.1000    SETROW(0,DATA,1,5,J)
21.1100    SETROW(0,DATA,2,5,J)
21.1200    CONTINUE
```

```
22.0100 * ****************************************
22.0200 *    STATUS AND DATA NOT VALID UNTIL    *
22.0300 *    SECOND CLOCK AFTER ADDRESS IN A    *
22.0400 *           WRITE SEQUENCE              *
22.0500 * ****************************************
22.0600 *
22.0700   LOOP 22.19 J=1,512
22.0800 * CHECK WR
22.0900   IF(GETBIT(DATA,2,30,J)) 22.19
22.1000 * CHANGE DATA COMPARE PATTERN TO MASK
22.1100   SETROW (0,DATA,1,5,J)
22.1200   SETROW (0,DATA,2,5,J)
22.1300 * CHANGE STATUS COMPARE PATTERN TO MASK
22.1400   SETROW (0,DATA,1,1,J)
22.1500   SETROW (0,DATA,2,1,J)
22.1600 *  JUMP OVER SECOND VECTOR UNLESS ITS THE LAST ONE
22.1700   J=J+1
22.1800   IF(J EQ 512) 23.01
22.1900   CONTINUE

23.0100 * ****************************
23.0200 *      INVERT READY DATA     *
23.0300 *        (COLUMN #24)        *
23.0400 * ****************************
23.0500   LOOP 23.1 J=1,512
23.0600   IF(GETBIT(DATA,2,24,J)) 23.09
23.0700   SETBIT(1,DATA,2,24,J)
23.0800   GOTO 23.1
23.0900   SETBIT(0,DATA,2,24,J)
23.1000   CONTINUE

25.0100 * ************************************
25.0200 *  ALTER FIRST FIVE ROW TO INCLUDE A *
25.0300 *       A  RESET CYCLE               *
25.0400 * ************************************
25.0500 *
25.0600 *  SET ALL ADDRESS/DATA OUTPUTS LINES
25.0700 *     TO MASK (COLUMNS #1-20)
25.0800   LOOP 25.13 J=1,5
25.0900   LOOP 25.12 I=1,20
25.1000   SETBIT(0,DATA,1,I,J)
25.1100   SETBIT(0,DATA,2,I,J)
25.1200   CONTINUE
25.1300   CONTINUE
25.1400 *
25.1500 *  SET ALL DATA INPUT TO INHIBIT (COLUMNS #37-52)
25.1600   LOOP 25.21 J=1,5
```

```
25.1700    LOOP 25.2 I=37,52
25.1800    SETBIT(1,DATA,1,I,J)
25.1900    SETBIT(0,DATA,2,I,J)
25.2000    CONTINUE
25.2100    CONTINUE
25.2200 *
25.2300 *  SET STATUS OUTPUTS TO MASK (COLUMNS #27-35)
25.2400    LOOP 25.29 J=1,5
25.2500    LOOP 25.28 I=27,35
25.2600    SETBIT(0,DATA,1,I,J)
25.2700    SETBIT(0,DATA,2,I,J)
25.2800    CONTINUE
25.2900    CONTINUE
25.3000 *
25.3100 *  SET RESET TO ONE ON ROW 1-5
25.3200 *     (COLUMN #21)
25.3300    LOOP 25.35 J=1,5
25.3400    SETBIT(1,DATA,2,21,J)
25.3500    CONTINUE
25.3600 *
25.3700 *  SET VECTOR ADDRESSES TO FFFF0 ON VECTORS 6,10,14,18
25.3800    LOOP 25.43 I=1,16
25.3900    SETBIT(1,DATA,2,I,6)
25.4000    SETBIT(1,DATA,2,I,10)
25.4100    SETBIT(1,DATA,2,I,14)
25.4200    SETBIT(1,DATA,2,I,18)
25.4300    CONTINUE

40.1000 * ************************************
40.2000 *  VECTOR STORAGE
40.3000 * ************************************
40.4000
40.5000    PATOUT(ERR,3,DATA,1,512)
40.6000    IF(ERR) 40.7,40.8
40.7000    PRINT "ERROR #",ERR:I1C," ENCOUNTERED WITH PATOUT",CR
40.8000    CONTINUE
40.9000    STOP
```

```
DATE  24-NOV-80      TIME  08:18      PATTERN FILE  HEADER.PAT;86

  1                    0           0           0           0
    TO                 1           2           3           4
       47  1234567890 1234567890 1234567890 1234567890 1234567

  0.0001  *AAAAAAAAAA AAAAAAAAAA RHTRNIHBRW MDADIMDDDD DDDDDDDDD
  0.0002  *1111DDDDDD DDDDDDDDDD EOEEMNLHDR ITLENNIIII IIIIIIIII
  0.0003  *9876DDDDDD DDDDDDDDDD SLSAITDE   DREVT 1111 119876503
  0.0004  *    111111 9876543210 EDTD RA       A 5432 10
  0.0005  *    543210            T  Y

  1.0000    0000000000 0000000000 1000000000 000001LLLL LLLLLLL
  2.0000    0000000000 0000000000 1000000000 000001LLLL LLLLLLL
  3.0000    0000000000 0000000000 1000000000 000001LLLL LLLLLLL
  4.0000    0000000000 0000000000 1000000000 000001LLLL LLLLLLL
  5.0000    0000000000 0000000000 1000000000 000001LLLL LLLLLLL

  5.1000    0000000000 0000000000 0000000000 000001LLLL LLLLLLL
  5.2000    0000000000 0000000000 0000000000 000001LLLL LLLLLLL
  5.3000    0000000000 0000000000 0000000000 000001LLLL LLLLLLL
  5.4000    0000000000 0000000000 0000000000 000001LLLL LLLLLLL
  5.5000    0000000000 0000000000 0000000000 000001LLLL LLLLLLL

  5.6000    0000000000 0000000000 0000000000 000001LLLL LLLLLLL
  5.7000    0000000000 0000000000 0000000000 000001LLLL LLLLLLL
```

```
DATE  24-NOV-80      TIME  08:18      PATTERN FILE  HEADER.PAT;86

 48          0 0
    TO       4 5
        52   89012

   0.0001   *AAAAAAAAAA AAAAAAAAAA RHTRNIHHRW MDADIMDDDD DDDDDDDDD
   0.0002   *1111DDDDDD DDDDDDDDDD EOEEMNLHOR ITLENNIIII IIIIIIIII
   0.0003   *9876000000 0000000000 SLSATTDE   ORENT 1111 119876543
   0.0004   *    111111 9876543210 EDTD RA        A 5432 10
   0.0005   *    543210            T  Y

   1.0000    LLLLL
   2.0000    LLLLL
   3.0000    LLLLL
   4.0000    LLLLL
   5.0000    LLLLL

   5.1000    LLLLL
   5.2000    LLLLL
   5.3000    LLLLL
   5.4000    LLLLL
   5.5000    LLLLL

   5.6000    LLLLL
   5.7000    LLLLL
```

```
1.0100    PRINT<10> ERASE
1.0300 *
1.0400 *                TEKTRONIX S-3260 TEST PROGRAM
1.0500 *
1.0600 *                    CIRCUIT TEST ENGINEERING
1.0700 *                GENERAL ELECTRIC ORDNANCE SYSTEMS
1.0800 *                    PITTSFIELD, MASSACHUSETTS
1.0900 *
1.1000 *
1.1100 * . DUT PART OR DWG. #      :8086
1.1200 * . DUT DESCRIPTION         :16 BIT MICROPROCESSEOR
1.1300 * . DATE                    :18-JUN-80
1.1400 * . PROGRAMMER              :T.J.WETZEL
1.1500 *
1.1600 *  THIS PROGRAM IS A LIMIT FUNCTION TEST FOR THE 8086UP
1.1610 *  USING WORST CASE INPUT TIMING CONDITIONS PASS/FAIL
1.1620 *  INFORMATION IS RECORDED AS THE FOLLOWING PARAMETERS
1.1630 *  ARE VARIED: TEMPERATURE,VCC,FREQUENCY,DUTY CYCLE,AND
1.1640 *   LOGIC INPUT LEVELS.
1.1700 *
1.1800 *
1.1900 * FOR SUPPORTING DOCUMENTS, CONSULT ETEC AND OTHER
1.2000 * CABINET FILES UNDER THE FOLLOWING IDENTIFIED TEST
1.2100 * SPECIFICATION NUMBER AND ADAPTER NUMBERS, AS WELL AS
1.2200 * LISTINGS OF THE FOLLOWING IDENTIFIED CORE OR MAGNETIC
```

```
1.2300 * TAPE FILES.
1.2400 *
1.2500 *
1.2600 * . TEST SPECIFICATION      :MIL-M-38510/530
1.2700 * . TEST TYPE/CONDITIONS    :CHARACTERIZATION
1.2800 * . PIN ASSIGNMENT FILE     :8086
1.2900 * . PAT FILE                :8086
1.3000 * . TST FILE                :8086
1.3100 * . THIS LISTING IS         :TEKTEST EDIT
1.3200 * . SOCKET CARD ASSEMBLY #  :2058
1.3300 *
1.3400 * ************************************************
1.3410 *                PROGRAM STRUCTURE
1.3500 * ************************************************
1.3700 *
1.3800 *     TEMPERATURE CONTROL
1.3900 *           ICC MONITOR
1.4000 *
1.4100 *           VCC CONTROL
1.4200 *              VSS-VERIFICATION
1.4300 *              DRIVER SETUP
1.4400 *              COMPARATOR SETUP
1.4500 *              LOAD CARD SETUP
1.4600 *
1.4700 *               DRIVE LEVEL CONTROL
```

```
1.4800 *
1.4900 *                    FREQUENCY CONTROL
1.5000 *
1.5100 *                       DUTY CYCLE CONTROL
1.5200 *                          TEST SETUP
1.5300 *                          PHASE TIMING
1.5400 *                          PHASE CONNECTS
1.5500 *
1.5600 *                          TESTING CONTROL
1.5700 *                                ERROR ROUTINE
1.5800 *
1.5900 *


2.0100 * ************************************************
2.0200 *  TABLE OF CONTENTS
2.0300 * ************************************************
2.0400
2.0500 * SECTION      TEST
2.0600 *
2.0700 *    1         HEADER INFORMATION
2.0800 *    2         TABLE CONTENTS
2.0900 *    3         PINLIST
2.1000 *    4         SUBROUTINES AND FUNCTIONS
2.1100 *    5         CONSTANT DEFINITIONS
2.1200 *    6         ARRAY DECLARATIONS
2.1300 *   10         TEMPERATURE
2.1400 *   11         ICC MONITOR
2.1500 *   20         VCC CONTROL
2.1600 *   21         VCC VERIFACATION
2.1700 *   22         DRIVER SETUP
2.1800 *   23         COMPARATOR SETUP
2.1900 *   24         LOAD CARD SETUP
2.2000 *   28         DRIVE LEVEL CONTROL
2.2100 *   30         FREQUENCY CONTROL
```

```
2.2200 *  35        DUTY CYCLE CONTROL
2.2300 *  40        TESTING SETUP
2.2400 *  41        PHASE CONNECTS
2.2500 *  42        PHASE TIMING
2.2600 *  45        TESTING CONTROL
2.2700 *  49        ERROR ROUTINE

3.0100 * ************************************************
3.0200 *  PINLIST ASSIGNMENTS
3.0300 * ************************************************
3.0400
3.0500   PINLIST AD1=AD15I,AD14I,AD13I,AD12I,AD11I,AD10I /
3.0600   AD9I,AD8I,AD7I,AD6I,AD5I,AD4I,AD3I,AD2I,AD1I,AD0I
3.0700   PINLIST AD0=AD150,AD140,AD130,AD120,AD110,AD100 /
3.0800   AD90,AD80,AD70,AD60,AD50,AD40,AD30,AD20,AD10,AD00
3.0900   PINLIST A=A19,A18,A17,A16
3.1000   PINLIST ADR=A,AD0
3.1100   PINLIST INS=RESET,HOLD,TEST,READY,NMI,INTR
3.1200   PINLIST OUTS=HLDA,BHE,RD,WR,MIO,DTR,ALE,DEN,INTA
3.1300   PINLIST ALLPINS=A,AD0,INS,OUTS,MN,AD1
3.1400   PINLIST DATAI=AD1,INS,MN,CLK
3.1500   PINLIST DATAO=ADR,OUTS
3.1600   PINLIST ALL=ALLPINS,CLK

4.0100 * ************************************************
4.0200 *  SUBROUTINES AND FUNCTION DECLARATIONS
4.0300 * ************************************************
4.0400
4.0500   FUNCTION HPWRLS(V):HP6129
4.0600   FUNCTION GETBIT(I,V,V,V):BARRAY
4.0700   FUNCTION ATTEMP(0):TP450A
4.0800   FUNCTION MVS5(V),MVS7(V),IVS5(0),IVS7(0):MC3
4.0900   FUNCTION CURSEC(0):TIME
4.1000
4.1100   SUBROUTINE EVS5(0),EVS7(0):MC3
4.1200   SUBROUTINE DVS5(0):MC3
4.1300   SUBROUTINE CRDATE(V),CRTIME(V):TIME
4.1400   SUBROUTINE HPWRRV(V,V,V),HPWRRS(V),HPWRCL(V,V):HP6129
4.1500   SUBROUTINE BARRAY(I,V [illegible] V):BARRAY
4.1600   SUBROUTINE SREAD(P,V, [illegible] ,V,I,V):SREAD
4.1700   SUBROUTINE SETEMP(V),RDTEMP(N):TP450A

5.0100 * ************************************
5.0200 *  DEVICE SPECIFICATION CONSTANTS
5.0300 * ************************************
5.0400
5.0500   VCCNOM = 5.0V
```

```
5.0600    VCCMAX = 5.0V + .05*(5.0V)
5.0700    VCCMIN = 5.0V - .05*(5.0V)
5.0800    ICCMAX = 340MA
5.0900    ILIMIT = 1A
5.1000 *  DIGITAL INPUT LEVELS
5.1100    VIHMIN = 2.2V
5.1200    VIHMAX = 5.0V
5.1300    VILMIN = 0.0V
5.1400    VILMAX = 0.6V
5.1500 *  CLOCK INPUT LEVELS
5.1600    VCL = 0.5V
5.1700    VCH = 3.9V
5.1800 *  VCC CONSTANTS
5.1900    VMIN = 4.0V
5.2000    VMAX = 6.0
5.2100    VINC = 100MV
5.2200 *   FREQUENCY CONSTANTS
5.2300    FMIN=1000KHZ
5.2400    FMAX=7500KHZ
5.2500    FINC=500KHZ
5.2600 *   DUTY CYCLE CONSTANTS(PERCENT OF PERIOD)
5.2700    DMIN = .20
5.2800    DMAX = .60
5.2900    DINC = .05
5.3000 *   TEMPERATURE CONSTANTS
5.3100    TMIN=-55
5.3200    TMAX=125
5.3300 *   LOGICAL UNIT ASSIGNMENTS
5.3400    DLUN=2
5.3500    OLUN=10
5.3600 *   TIME OFFSET
5.3700    SW=10NS
5.3800 *   LOAD CARD CONSTANTS
5.3900    LOADV=2.5V
5.4000    LOADI= 1.33MA
5.4100    NOUT = 29

6.1000 * ************************************
6.2000 *       ARRAY DECLARATIONS
6.3000 * ************************************
6.4000
6.5000    IARRAY ERR(2000)
6.6000    BARRAY(ERR,2,29,5'6)
6.7000
6.8000    ARRAY TEM(7)
6.9000    PRESET TEM(1)= -55,-30,0,25,70,100,125
```

```
    9.1000 *  LOG SERIAL NUMBER
    9.2000    ACCEPT"ENTER DEVICE SERIAL NUMBER ",SN,CR
    9.3000    LOGMARKER<1>20
    9.4000    LOGPARAMETRIC<1,S"SN">SN
    9.5000    PRINT<OLUN>CR,CR,"SN ",SN,CR

   10.0100 * ****************************************
   10.0200 *      TEMPERATURE CONTROL
   10.0300 * ****************************************
   10.0400    LOOP 10.15 T=1,7
   10.0500    TEMP=TEM(T)
   10.0600    TFLAG=1
   10.0700    PRINT<OLUN>CR,CR,"TEMPERATURE = ",TEMP:10,"DEGREES C"
   10.0800    LOGMARKER<1>30
   10.0900    LOGPARAMETRIC<1,S"TEMP">TEMP
   10.1000     SETEMP(TEMP)
   10.1100 * CALL ICC MONITOR
   10.1200    CALL 11.01
   10.1300 * CALL VCC CONTROL
   10.1400    CALL 20.01
   10.1500    CONTINUE
   10.1600    INITIALIZE
   10.1700    SETEMP(25)
   10.1800    STOP

   11.0100 * ****************************************************
   11.0200 *      ICC MONITOR ROUTINE
   11.0300 * ****************************************************
   11.0400 *
   11.0500 *  MONITOR ICC WHILE TEMP IS CHANGING TO ENSURE
   11.0600 *  MAX POWER DISSIPATION IS NOT EXCEEDED
   11.0700
   11.0800    INITIALIZE
   11.0900
   11.1000    TCLCL =400NS
   11.1100    CYCLE=TCLCL
   11.1200    SETUP TO MEASURE CURRENT ON VCC FROM VS3=VCCMIN AT 34
              0MA
   11.1300    WAIT 100MS
   11.1400    VS3=VCCMAX,ICCMAX
   11.1500 *  SETUP OUT CLK
   11.1600    CONNECT INPUT TO DRIVER ON CLK
   11.1700    HIDRIVE = VIHMIN ON CLK
   11.1800    LODRIVE = VILMIN ON CLK
   11.1900    PHASE 13 =2/3*TCLCL FOR 1/3*TCLCL
   11.2000    FORCE CLK WITH ONE;R7
   11.2100 *  PUT OUT INTO RESET STATE
```

```
11.2200   CONNECT INPUT TO DRIVER ON RESET
11.2300   HIDRIVE =VIHMIN ON RESET
11.2400   LODRIVE = VILMAX ON RESET
11.2500   FORCE RESET WITH ONE
11.2600 *  PUT OUT INTO MIN MODE
11.2700   CONNECT INPUT TO DRIVER ON MN
11.2800   HIDRIVE = VIHMIN ON MN
11.2900   LODRIVE = VILMAX ON MN
11.3000   FORCE MN WITH ONE
11.3100   BURST ON
11.3200
11.3300   ICC=CURRENT
11.3400   IF(0MA<ICC<ICCMAX) 11.5,11.35,11.5
11.3500   IF(NOT(ATTEMP)) 11.33
11.3600 *
11.3700 * TIME OUT 5 MINUTES
11.3800   TIM=CURSEC
11.3900   ICC=CURRENT
11.4000   PRINT CR, (CURSEC-TIM),CURRENT
11.4100   IF(0MA<ICC<ICCMAX) 11.5,11.42,11.5
11.4200   IF((CURSEC-TIM) LT 300) 11.39
11.4300 *
11.4400   BURST OFF
11.4500   UNSET TO MEASURE CURRENT ON VCC FROM VS3
11.4600   INITIALIZE
11.4700   RETURN
11.4800 *
11.4900 *  EMERGENCY SHUTDOWN ROUTINE
11.5000   SETEMP(25)
11.5100   INITIALIZE
11.5200   PRINT "↑G↑G↑G↑G↑G↑G↑G↑G↑G↑G↑G"
11.5300   PRINT CR,CR,CR
11.5400   PRINT "ICC EXCEEDED MAX LIMIT DURING TEMP CHANGE",CR,
          CR
11.5500   PRINT "TEST ABORTED"
11.5600   STOP

20.0100 * ****************************************
20.0200 *            VCC CONTROL                 *
20.0300 * ****************************************
20.0400
20.0500   LOOP 20.22 VCC = VMIN,VMAX,VINC
20.0600   VCCVS5 = VCC
20.0700   ICCVS5 = ILIMIT
20.0800 * CALL VCC VERIFICATION - VS5
20.0900   CALL 21.01
20.1000   LOGMARKER<1>40
```

```
20.1100   LOGPARAMETRIC<1,S"VCC">VCC,ICC
20.1200   PRINT<OLUN>CR,CR,"VCC = ",VCC,"VOLTS  ",ICC,"AMPS"
20.1300 *  CALL DRIVER SETUP ROUTINE
20.1400   CALL 22.01
20.1500 *  CALL COMPARATOR SETUP ROUTINE
20.1600   CALL 23.1
20.1700 * CALL LOAD CARD SETUP
20.1800   CALL 24.01
20.1900 *  CALL DRIVE LEVEL CONTROL
20.2000   CALL 28.01
20.2100   DISCONNECT LOAD1 ON DATA0
20.2200   CONTINUE
20.2300   RETURN

21.0100 * ********************************
21.0200 *   VCC VERIFICATION - VSS      *
21.0300 * ********************************
21.0400
21.0500   EVSS
21.0600   HPWRRS(1)
21.0700   HPWRCL(1,ICCVSS)
21.0800   HPWRRV(1,1,VCCVSS)
21.0900   WAIT 1SEC
21.1000 * ICC=IVSS
21.1100   V1=MVSS(VCCVSS)
21.1200   IF((VCCVSS-(VCCVSS*.01))<V1<(VCCVSS+(VCCVSS*.01))) 21
          .17
21.1300   PRINT<OLUN> CR,"FAILED THE ",VCCVSS,"VOLT VEE-VSS VE
          IFICATION"
21.1400   PRINT<OLUN> CR, "TEST WAS ABORTED"
21.1500   DISPLAY 21, FAIL
21.1600   STOP
21.1700   DISPLAY 21, PASS
21.1800   RETURN

22.0100 * ********************************
22.0200 *     DRIVER  SETUP            *
22.0300 * ********************************
22.0400 *
22.0500   CONNECT INPUT TO DRIVER ON DATAI
22.0600 *
22.0700   HIDRIVE= VIHMIN ON DATAI
22.0800   LODRIVE = VILMAX ON DATAI
22.0900   INHIBIT DATAI WITH ONE
22.1000   CONNECT INPUT TO DRIVER ON CLK
22.1100   HIDRIVE = VCH ON CLK
22.1200   LODRIVE = VCL ON CLK
```

```
22.1300    INHIBIT CLK WITH ONE
22.1400 *
22.1500    RETURN

23.1000 * ****************************************
23.2000 *     COMPARATOR SETUP ROUTINE          *
23.3000 * ****************************************
23.4000 *
23.5000    CONNECT OUTPUT TO COMPARATOR ON DATA0
23.6000    LOCOMPARE = .45V ON DATA0
23.7000    HICOMPARE = 2.4V ON DATA0
23.8000    RETURN

24.0100 * ********************************
24.0200 *   LOAD CARD SETUP               *
24.0300 * ********************************
24.0400    CONNECT LOAD1 ON DATA0
24.0500    EVS7
24.0600    VS7=LOADV AT LOADI*NOUT
24.0700    WAIT 50MS
24.0800    V1=MVS7(LOADV)
24.0900    I1=IVS7
24.1000    IF(I1 GT LOADI*NOUT)24.11,24.14
24.1100    PRINT <OLUN> "DEVICE DRAWS MORE THAN",LOADI*NOUT,"A (
           F CURRENT"
24.1200    INITIALIZE
24.1300    STOP
24.1400    IF((LOADV-(LOADV*.01))<V1<(LOADV+(LOADV*.01))) 24.19
24.1500    PRINT<OLUN> CR,"FAILED THE ",LOADV,"VOLT (LOAD CARDS)
           -VS7 VERIFICATION"
24.1600    PRINT<OLUN> CR, "TEST WAS ABORTED"
24.1700    DISPLAY 24, FAIL
24.1800    STOP
24.1900    DISPLAY 24, PASS
24.2000    RETURN

28.0100 * ****************************************
28.0200 *           DRIVE LEVEL CONTROL         *
28.0300 * ****************************************
28.0400    HIDRIVE = 2.0V ON DATAI
28.0500    LODRIVE = 0.8V ON DATAI
28.0600    LOGMARKER<1>45
28.0700    LOGPARAMETRIC<1,S"DR">2.0
28.0800    PRINT<OLUN>CR,CR,"DRIVE LEVELS = 2.0V AND 0.8V"
28.0900 *  CALL FREQUENCY CONTROL
28.1000    CALL 30.01
28.1100    HIDRIVE = 2.22V ON DATAI
```

```
28.1200    LODRIVE = 0.6V ON DATA1
28.1300    LOGMARKER<1>45
28.1400    LOGPARAMETRIC<1,S"DR">2.22
28.1500    PRINT<OLUN>CR,CR,"DRIVE LEVELS = 2.22V AND 0.6V"
28.1600 *   CALL FREQUENCY CONTROL
28.1700    CALL 30.01
28.1800    HIDRIVE = 2.4V ON DATA1
28.1900    LODRIVE = 0.4V ON DATA1
28.2000    LOGMARKER<1>45
28.2100    LOGPARAMETRIC<1,S"DR">2.4
28.2200    PRINT<OLUN>CR,CR,"DRIVE LEVELS = 2.4V AND 0.4V"
28.2300 *  CALL FREQUENCY CONTROL
28.2400    CALL 30.01
28.2500    RETURN

30.0100 * ****************************************
30.0200 *   FREQUENCY CONTROL                      *
30.0300 * ****************************************
30.0400 *  PRINT OUT HEADING
30.0500    PRINT<OLUN>CR,"DUTY CYCLE          "
30.0600    LOOP 30.08 DCYCLE=DMIN,DMAX,DINC
30.0700    PRINT<OLUN>"    ",DCYCLE*100$IOC,"%"
30.0800    CONTINUE
30.0900    PRINT<OLUN>CR
30.1000    LOOP 30.18 FREQ=FMIN,FMAX,FINC
30.1100    PRINT<OLUN>CR,"FREQ=",FREQ,"    "
30.1200    TCLCL=1/FREQ
30.1300    LOGMARKER<1>50
30.1400    LOGPARAMETRIC<1,S"P">FREQ
30.1500    CYCLE = TCLCL
30.1600 *  CALL DUTY CYCLE CONTROL
30.1700    CALL 35.01
30.1800    CONTINUE
30.1900    RETURN

35.0100 * ****************************************
35.0200 *         DUTY CYCLE CONTOL                *
35.0300 * ****************************************
35.0400 *
35.0500 *  DUTY CYCLE IS IN PERCENT OF PERIOD
35.0600    LOOP 35.12 TCHCL=DMIN*TCLCL,DMAX*TCLCL,DINC*TCLCL
35.0700 *   TCHCL = CLK HIGH TIME
35.0800 *   TCLCH = CLK LOW TIME
35.0900    TCLCH=TCLCL-TCHCL
35.1000 *  CALL TESTING SETUP
35.1100    CALL 40.01
35.1200    CONTINUE
```

```
35.1300    RETURN

40.0100 * ************************************
40.0200 *      TESTING SETUP                  *
40.0300 * ************************************
40.0400 *
40.0500    ERFLG = 0
40.0600    IF(TFLAG EQ 1)40.08,40.1
40.0700 *  CALL PHASE CONNECTS
40.0800    CALL 41.01
40.0900 *
40.1000 * PASS ONE
40.1100 * CHECK ADDRESS/DATA LINE,BHE,WR
40.1200 * CALL PHASE TIMING=PASS1
40.1300    CALL 42.07
40.1400    FORCE ADI WITH PATTERN;RZ
40.1500    INHIBIT ADI WITH PATTERN
40.1600    FORCE INS WITH PATTERN
40.1700    FORCE CLK WITH ONE;RZ
40.1800    FORCE READY WITH ONE
40.1900    FORCE MN WITH PATTERN
40.2000    COMPARE ADR WITH PATTERN
40.2100    MASK ADR WITH PATTERN
40.2200    COMPARE BHE WITH PATTERN
40.2300    MASK BHE WITH PATTERN
40.2400    COMPARE WR WITH PATTERN
40.2500    MASK WR WITH PATTERN
40.2600    COMPARE HLDA WITH PATTERN
40.2700    MASK HLDA WITH PATTERN
40.2800 *  CALL TESTING CONTROL
40.2900    CALL 45.01
40.3000 *
40.3100 * PASS TWO
40.3200 * CHECK ALE
40.3300 * CALL PHASE TIMING=PASS2
40.3400    CALL 42.2
40.3500    FORCE ADI WITH PATTERN;RZ
40.3600    INHIBIT ADI WITH PATTERN
40.3700    FORCE INS WITH PATTERN
40.3800    FORCE CLK WITH ONE;RZ
40.3900    FORCE READY WITH ONE
40.4000    FORCE MN WITH PATTERN
40.4100    COMPARE ALE WITH PATTERN
40.4200    MASK ALE WITH PATTERN
40.4300 *  CALL TESTING CONTROL
40.4400    CALL 45.01
40.4500 *
```

```
40.4600 * PASS THREE
40.4700 * CHECK RD
40.4800 * CALL PHASE TIMING=PASS3
40.4900   CALL 42.37
40.5000   FORCE ADI WITH PATTERN;RZ
40.5100   INHIBIT ADI WITH PATTERN
40.5200   FORCE INS WITH PATTERN
40.5300   FORCE CLK WITH ONE;RZ
40.5400   FORCE READY WITH ONE
40.5500   FORCE MN WITH PATTERN
40.5600   COMPARE RD WITH PATTERN
40.5700   MASK RD WITH PATTERN
40.5800 *  CALL TESTING CONTROL
40.5900   CALL 45.01
40.6000 *
40.6100 * PASS FOUR
40.6200 * CHECK DTR,DEN
40.6300 * CALL PHASE TIMING=PASS4
40.6400   CALL 42.54
40.6500   FORCE ADI WITH PATTERN;RZ
40.6600   INHIBIT ADI WITH PATTERN
40.6710   FORCE INS WITH PATTERN
40.6800   FORCE CLK WITH ONE;RZ
40.6900   FORCE READY WITH ONE
40.7000   FORCE MN WITH PATTERN
40.7100   COMPARE DTR WITH PATTERN
40.7200   MASK DTR WITH PATTERN
40.7300   COMPARE DEN WITH PATTERN
40.7410   MASK DEN WITH PATTERN
40.7500 *  CALL TESTING CONTROL
40.7600   CALL 45.01
40.7700   PRINT<OLIN>NOT(ERFLG):I1,"  "
40.7800   LOGPARAMETRIC<1,8"0">TCHCL/TCLCL*100,NOT(ERFLG)
40.7900   RETURN

41.0100 * ****************************************
41.0200 *         PHASE  CONNECTS
41.0300 * ****************************************
41.0400
41.0500 * PHASE 1
41.0600   CONNECT TO PHASE ON NMI
41.0700 * PHASE 2
41.0800   CONNECT TO PHASE ON INTR
41.0900 * PHASE 3
41.1000   CONNECT TO PHASE ON CLK
41.1100 * PHASE 4
41.1200 * CONNECT TO PHASE ON RESET
```

```
41.1300 * PHASE 5
41.1400   CONNECT TO PHASE ON READY
41.1500 * PHASE 6
41.1600   CONNECT TO PHASE ON TEST
41.1700 * PHASE 7
41.1800 * CONNECT TO PHASE ON HLDA
41.1900 * PHASE 8
41.2000   CONNECT TO PHASE ON HOLD
41.2100 * PHASE 13 AND 14 (16 DATAI LINES)
41.2200   CONNECT TO DATAPHASE ON ADT
41.2300   TFLAG=0
41.2400   RETURN

42.0100 * ****************************************
42.0200 *         PHASE TIMING                   *
42.0300 * ****************************************
42.0400 *
42.0500 *  SW IS THE OFFSET NEEDED TO MAINTAIN DATA HOLD TIME
42.0600 *
42.0700 * PASS ONE ENTRY POINT
42.0800 * CLK
42.0900   PHASE 3 = TCLCH-SW FOR TCHCL
42.1000 * DATAI
42.1100   PHASE 13 = TCLCL-30NS-SW FOR 40NS
42.1200   PHASE 14 = TCLCL-30NS-SW FOR 40NS
42.1300 * SETUP THE COMPARE PHASES FOR ADR/DATA LINE,BHE,RD
42.1400   PHASE  9 =110NS-SW FOR 20NS
42.1500   PHASE 10 =110NS-SW FOR 20NS
42.1600   PHASE 11 =110NS-SW FOR 20NS
42.1700   PHASE 12 =110NS-SW FOR 20NS
42.1800   RETURN
42.1900 *
42.2000 * PASS TWO ENTRY POINT
42.2100 * CLK
42.2200   PHASE 3 = TCLCH-SW FOR TCHCL
42.2300 * DATAI
42.2400   PHASE 13 = TCLCL-30NS-SW FOR 40NS
42.2500   PHASE 14 = TCLCL-30NS-SW FOR 40NS
42.2600 * SETUP THE COMPARE PHASES FOR ALE
42.2700   IF(TCLCH LE 80NS+20NS) 42.3,42.28
42.2800   X=80NS
42.2900   GOTO 42.31
42.3000   X=TCLCH-20NS
42.3100   PHASE  9 =X-SW FOR 20NS
42.3200   PHASE 10 =X-SW FOR 20NS
42.3300   PHASE 11 =X-SW FOR 20NS
42.3400   PHASE 12 =X-SW FOR 20NS
```

```
42.3500    RETURN
42.3600 *
42.3700 * PASS THREE ENTRY POINT
42.3800 * CLK
42.3900    PHASE 3 = TCLCH-SW FOR TCHCL
42.4000 * DATAI
42.4100    PHASE 13 = TCLCL-30NS-SW FOR 40NS
42.4200    PHASE 14 = TCLCL-30NS-SW FOR 40NS
42.4300 * SETUP THE COMPARE PHASES FOR RD
42.4400    IF(TCLCL LE 165NS+20NS) 42.45,42.47
42.4500    X=TCLCL-20NS
42.4600    GOTO 42.48
42.4700    X=165NS
42.4800    PHASE  9 =X-SW FOR 20NS
42.4900    PHASE 10 =X-SW FOR 20NS
42.5000    PHASE 11 =X-SW FOR 20NS
42.5100    PHASE 12 =X-SW FOR 20NS
42.5200    RETURN
42.5300 *
42.5400 * PASS FOUR ENTRY POINT
42.5500 * CLK
42.5600    PHASE 3 = 0NS+SW FOR TCHCL
42.5700 * DATAI
42.5800    PHASE 13 = TCHCL-90NS+SW FOR 130NS
42.5900    PHASE 14 = TCHCL-90NS+SW FOR 130NS
42.6000 * SETUP THE COMPARE PHASES FOR DTR,DEN
42.6100    IF(TCLCL LE 110NS+20NS) 42.62,42.64
42.6200    X=TCLCL-20
42.6300    GOTO 42.65
42.6400    X=110NS
42.6500    PHASE  9 =X+SW FOR 20NS
42.6600    PHASE 10 =X+SW FOR 20NS
42.6700    PHASE 11 =X+SW FOR 20NS
42.6800    PHASE 12 =X+SW FOR 20NS
42.6900 * PHASE  9 =0NS FOR 40NS
42.7000 * PHASE 10 =0NS FOR 40NS
42.7100 * PHASE 11 =0NS FOR 40NS
42.7200 * PHASE 12 =0NS FOR 40NS
42.7300    RETURN

45.0100 * ************************************
45.0200 *      TESTING CONTROL               *
45.0300 * ************************************
45.0400 *
45.0500    LOAD FROM CORE DATA4E(1,511+4) TO ALLPINS WITH FI,CM
45.0600    MOVE REGISTER((511+4)) TO ALL WITH FI,CM AND SAVE ER
           ORS
```

```
45.0700    INHIBIT DATAI WITH ONE
45.0800    MASK DATAO WITH ONE
45.0900    IF(ERROR) 45.1,45.15
45.1000    ERFLG = 1
45.1100    SWITCH = AND(#000001,GETBUS(#177570))
45.1200    IF (SWITCH) 45.15
45.1300 *  CALL ERROR ROUTINE
45.1400    CALL 49.01
45.1500    RETURN

49.0100 * ****************************************
49.0200 *              ERROR ROUTINE              *
49.0300 * ****************************************
49.0400 *
49.0500    SPEAD(DATAO,1,29,5,511,ERR,2)
49.0600    LOOP 49.39 J=1,511
49.0700    PFLAG=0
49.0800    LOOP 49.38 I= 1,29
49.0900    IF(NOT(GETBIT(ERR,2,I,J))) 49.38
49.1000    IF(PFLAG) 49.14
49.1100    PRINT<OLUN> CR,CR,"***** FAILURE ON VECTOR",J:I3C," *
           ****",CR
49.1200    PRINT<OLUN>"PINS "
49.1300    PFLAG=1
49.1400    IF(1<I<4) 49.17
49.1500    IF(5<I<20) 49.19
49.1600    GOTO(I-20) 49.21,49.23,49.25,49.27,49.29,49.31,49.33,
           49.35,49.37
49.1700    PRINT<OLUN>"A",(20-I):I0C," "
49.1800    GOTO 49.38
49.1900    PRINT<OLUN>"AD",(20-I):I0C," "
49.2000    GOTO 49.38
49.2100    PRINT<OLUN>"HLDA "
49.2200    GOTO 49.38
49.2300    PRINT<OLUN>"BHE "
49.2400    GOTO 49.38
49.2500    PRINT<OLUN>"RD "
49.2600    GOTO 49.38
49.2700    PRINT<OLUN>"WR "
49.2800    GOTO 49.38
49.2900    PRINT<OLUN>"MIO "
49.3000    GOTO 49.38
49.3100    PRINT<OLUN>"DTR "
49.3200    GOTO 49.38
49.3300    PRINT<OLUN>"ALE "
49.3400    GOTO 49.38
49.3500    PRINT<OLUN>"DEN "
```

```
49.3600     GOTO 49.38
49.3700     PRINT<OLUN>"INTA "
49.3800     CONTINUE
49.3900     CONTINUE
49.4000     RETURN
```

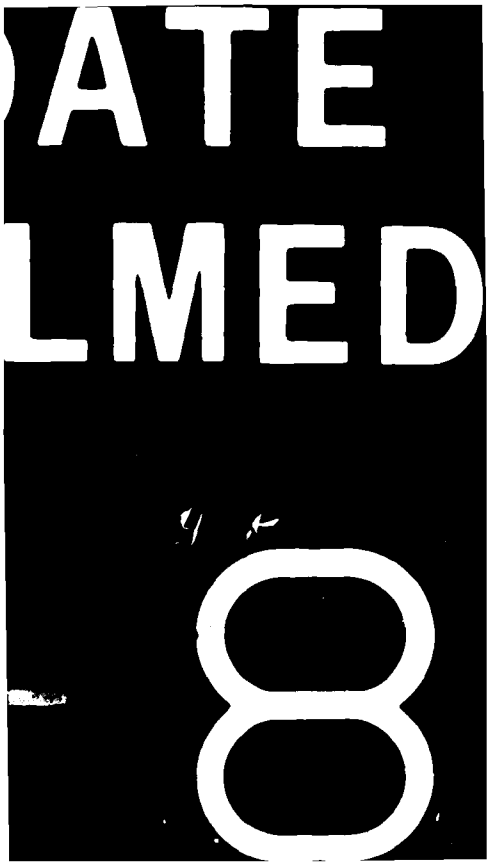
DK0: A6.PIN;86 DATE: 20-NOV-80 TIME: 13:16:27

LINE NUMBER		SECTOR NUMBER	PIN NAME	OUT PIN OR COMMENT
1.0100	*	PIN ASSIGNMENTS FOR 8086 UP		
1.0200	*			
1.0300		1WAI	AD7I	PIN9
2.0000		2XA0	AD7O	PIN9
3.0000		3YAI	AD6I	PIN10
4.0000		4ZA0	AD6O	PIN10
5.0000		5WAI	AD5I	PIN11
6.0000		6XA0	AD5O	PIN10
7.0000		7YAI	NMI	PIN17
8.0000	*			
9.0000		9WAI	AD4I	PIN12
10.0000		10XA0	AD4O	PIN12
11.0000		11YAI	AD3I	PIN13
12.0000		12ZA0	AD3O	PIN13
13.0000		13WAI	AD2I	PIN14
14.0000		14XA0	AD2O	PIN14
15.0000		15YAI	INTR	PIN18
16.0000	*			
17.0000		17WAI	AD1I	PIN15
18.0000		18XA0	AD1O	PIN15
19.0000		19YAI	AD0I	PIN16
20.0000		20ZA0	AD0O	PIN16
21.0000		21WAI	CLK	PIN19
22.0000	*			
23.0000	*			
24.0000	*			
25.0000		25WA0	DEN	PIN26
26.0000		26XA0	ALE	PIN25
27.0000		27YA0	INTA	PIN24
28.0000	*			
29.0000	*			
30.0000	*			
31.0000	*			
32.0000		32ZAI	RESET	PIN21
33.0000	*			
34.0000		34XAI	READY	PIN22
35.0000		35YA0	DTR	PIN27
36.0000		36ZA0	MIO	PIN28
37.0000		37WA0	WR	PIN29
38.0000		38XA0	RD	PIN32
39.0000		39YA0	MN	PIN33
40.0000		40ZA0	BHE	PIN37
41.0000		41WAI	TEST	PIN23
42.0000		42XA0	A19	PIN35
43.0000		43YA0	A18	PIN36

44.0000	44ZA0	A17	PIN37
45.0000	45WAI	AD15I	PIN36
46.0000	46XA0	AD15O	PIN39
47.0000	47YAI	AD8I	PIN8
48.0000	48ZA0	AD8O	PIN8
49.0000	49WA0	HLDA	PIN30
50.0000	50XA0	A16	PIN38
51.0000	51YAI	AD9I	PIN9
52.0000	52ZA0	AD9O	PIN7
53.0000	53WAI	AD10I	PIN6
54.0000	54XA0	AD10O	PIN6
55.0000	55YAI	AD11I	PIN5
56.0000	56ZA0	AD11O	PIN5
57.0000	57WAI	HOLD	PIN31
58.0000	58XA0	VCC	PIN40
59.0000	59YAI	AD12I	PIN4
60.0000	60ZA0	AD12O	PIN4
61.0000	61WAI	AD13I	PIN3
62.0000	62XA0	AD13O	PIN3
63.0000	63YAI	AD14I	PIN2
64.0000	64ZA0	AD14O	PIN2

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.


ATE
LMED
8