NoteTaker
A Portable Computing System

XEROX PALO ALTO RESEARCH CENTER
Systems Science Laboratory
LSI Systems Area
March 11, 1978

For Xerox Internal Use Only

Subject: **NoteTaker Debugging Plan**

From: Douglas Fairbairn

filed on [IVY] < fairbairn > nt > NTDebugplan.nt

The following describes in more or less detail the plan of how the NoteTaker boards and system will be debugged. This plan of course represents a compromise between time and thoroughness. As we learn more of the typical failure modes of the system, we can modify the procedures as appropriate. The testing will be divided into three phases: chip testing, board testing and system testing. Successful passage of early tests is assumed in later tests.

I. Chip Testing:

The only chips which will be tested individually will be the 8086s. In that the parts are new and we have little idea of what their likely failure modes might be, it seems worthwhile to characterize them for successful execution of a basic instruction set over a range of frequencies. It is not my intention to test all their features during these tests, but merely to gain a feeling for their relative tolerance to frequency and supply voltage variations.

A. Test Goals: Establish the limits of power supply voltage and frequency over which the 8086 will execute a basic set of instructions.

B. Test Setup: We should try to use the setup developed at ED when they were evaluating 8086s. It has the capability of varying the two parameters of interest in a controlled way. We could also use their instruction test software I presume.

C. Test Procedure:

1. Run the 8086 throught the standard instruction test for 5-10 seconds at nominal voltage (5.0 volts) and frequency (5.0 Mhz.).

2. Increase and decrease the supply voltage to find the limits which the chip will operate over at nominal frequency.

3. Reset voltage to nominal and vary the frequency to find the operational limits. It may not be practical to find the lower limit. If the system works for about 60 seconds at 100 Khz., that shuld be a sufficient test. Note that at the lower frequencies it is advisable to run the system longer so as to execute an adequate number of instructions.

D. Documentation: Each of the chips should be numbered in a permanent way and the results of the test recorded in a clear way for future reference.

II. Board Testing

We should continue to check out the boards individually as we have done so far. The boards will be tested in either the first INOVA chassis or the wood mock-up. These chassis will be assumed to be functionally interchangable unless we discover otherwise. The power supplies will be set to 5.0 volts, 12.0 volts, -20.0 volts *at the backplane with all the boards plugged in.* All 8086s will be run at 5.0 Mhz. and the memory system at 18 Mhz.

A. Emulation Processor Module:

1. Test Goals: Verify operation of emulation processor board in known environment.

2. Test Setup: Standard debug chassis with either old or new (in-chassis) debug board. The new debug board is to be plugged into the battery module slot.

3. Test Procedure:

- Insert in system and run initialization program. This will test the operation of the 8086 and all the local memory. In the event of failure:

    - Verify that all mods have been made correctly by double-checking buzz list. We need a list which shows what shuld be connected and what *should not* be onnected.

    - Inspect board for chips plugged in backwards, bent pins, etc.

- The software should verify the operation of the NMI (non-maskable interrupt).

- Check operation of interrupt circuitry by first checking the 60 hz interrupt. This check could be added to initialization software. It should send specific message to operator if it doesn't work.

- Operator runs memory diagnostic on main memory (about 5 passes through *all* memory.) The CRT should be actively refreshing at this time to furnish bus contention.

- Verify ability to disable local memory and run completely out of main memory.

- The illegal address traps (interrupts) in locations FFFC0-FFFDF should be tested.

- Operator removes local memory chip #e8 and tests the local parity error reporting and interrupt hardware. The test software should have loaded itself into main memory and then tested local memory to force parity error. It could/should instruct operator what to do at each step. Software should verify it can clear parity error, disable their occurence, and mask out the interrupt as well. The operator then replaces #e8 and removes #e4 to test parity detection on other byte of word.

- The test program should now enter a loop where it is toggling the *ProcLock* bit and the operator should manually check that it is occuring properly on the bus. This might be accomplished easiest by bringing this signal out on the in-chassis debug board to the rear connector where it could be easily viewed.

- Operator grounds one of the bit lines in main memory to check opertion of main memory parity error interrupt. The test software should also check to see that it can properly determine the bit which was bad and the region in which it occured.

4. Documentation: Wouldn't it be nice if the program kept track of the histroy of debugging a board. It could tell when faults occurred and ask the operator to type in what was done to rectify it. All this would be written on a file. This would be useful in gathering statistics, etc. for later planning.

B. I/O Processor Module:

1. Test Goals: Verify operation of I/O processor board in known environment.

2. Test Setup: Standard debug chassis with either old or new (in-chassis) debug board. The new debug board is to be plugged into the battery module slot. The disk/display board should not be plugged into chassis at all.

3. Test Procedure:

- Insert in system and run initialization program. This will test the operation of the 8086 and all the local memory. In the event of failure:

  - Verify that all mods have been made correctly by double-checking buzz list. We need a list which shows what shuld be connected and what *should not* be onnected.

  - Inspect board for chips plugged in backwards, bent pins, etc.

nsertion.

The same intialization operation should be tried from the external boot switch to verify this mode of startup.

- The software should verify the operation of the NMI (non-maskable interrupt).

- Check operation of interrupt circuitry by checking the 60 hz interrupt. This check could be added to initialization software. It should send specific message to operator if it doesn't work.

Check voltages on on-board regulators (+and- 15v and +5 volts)

Run IOPMemTest through all of memory 5 times, move code and repeat.

The ROM should be disabled now to verify access to all main memory. Enable ROM again.

The EIA UART should be turned around on itself and operation verified by sending several hundred characters through the interface. Verify operation of EIA interrupt and ability to mask that interrupt. The EIA controller is on the Disk/Display board and should be known to work at this point. This test will verify the ability of the 8086 to talk to a device on the DD board.

Verify operation of DAC, Sample/hold, and amplifier. Verify basic operation of the speaker. In order to more accurately characterize the D/A, take output from headphone jack and feed it into the A/D converter (on the disk/display board). The A/D converter input will only tolerate a + or -5v swing so the output of the D/A must be scaled through a 2:1 resistor network in order to stay within the range of the A/D. Alternatively, the software can send out a series of sine waves of maximum amplitude. Operator then looks at waveform with scope to verify no clipping, proper wave shape, etc.

Verify that the FIFO clock on the D/A converter can be set to all its frequencies. This should be checked by operator with prompts from test program.

Visually inspect tablet for defects, etc. Run tablet test to verify operation over whole display and that there are no obvious bad spots.

Remove a specific memory chip from main-memory and let test program verify that the error is reported properly and that the interrupt can be masked off.

4. Documentation:

C. Disk/Display Module:

1. Test Goals: Verify operation of disk/display board in known environment.

2. Test Setup: Standard debug chassis with either old or new (in-chassis) debug board. The new debug board is to be plugged into the battery module slot.

3. Test Procedure:

· Plug in board and verify that system can still be initialized. If this step fails, check that all mods have been made correctly and that there are no bent pins, bad chips, etc.

Check operation of control register on the various things it controls. Make sure power can be turned on and off to the disk. Then make sure the bit clock and A/D converter speeds can be set correctly. Verify that the drive select and side select bits for the disk are OK. Verify ability to select different analog channels on A/D converter.

Start up CRT and verify its operation. Make sure to check that the ability to block the display with a limited number of scan lines also works.

Verify that the A/D converter will properly check the +5 and +12 power supply voltages and that the tablet works.

Loop the EIA interface back on itself and verify that it performs correctly.

Run thorough series of checks on disk.

4. Documentation:

D. Memory Data Module:

1. Test Goals: Verify that data module can correct all single bit errors, detect all double bit errors without correcting them, and properly report the locations and types of errors detected.

2. Test Setup: Memory test chassis with new Mex board and a standard battery of tests run automatically from Mex.

3. Test Procedure:

Determine a complete set of tests, including byte reads and writes, page reads and writes, a write followed by all reads, etc. Note that error reporting can also be handled by Mex. It should perhaps do this automatically when it sees an error. To check the error correcting capability, one of the bit lines must be manually grounded to simulate a fault.

4. Documentation:

E. Memory Address Module:

1. Test Goals: Verify that data module can correct all single bit errors, detect all double bit errors without correcting them, and properly report the locations and types of errors detected.

2. Test Setup: Memory test chassis with new Mex board and a standard battery of tests run automatically from Mex.

3. Test Procedure:

Determine a complete set of tests, including byte reads and writes, page reads and writes, a write followed by all reads, etc. Note that error reporting can also be handled by Mex. It should perhaps do this automatically when it sees an error. To check the error correcting capability, one of the bit lines must be manually grounded to simulate a fault.

4. Documentation:

F. Battery Module:

1. Test Goals:

Verify that batteries are fully charged, can power the unit for a minimum period of time and that the charging mechanism recharges them in the required period of time.

2. Test Setup: Fuly working NoteTaker interfaced to Alto with debug board.

3. Test Procedure:

   Check to see if batteries are fully charged. If they are, as determined by A/D converter, instruct operator to turn off AC line. Verify that computer is still operational. If so, let it monitor battery voltage and periodically report that to the Alto. The operator wil have to verify that the battery voltage profile is within reasonable limits. If the system is left in this configuration and the battery output drops too low, the power supply will quit functioning and the battery drain will be drop to a minimum. When the AC line is turned on again and the battery starts recharging, the voltage profile should again be monitored periodically, both with the charger on and off. This is because the charger will cause the batteries to reach a much higher voltage than they would by themselves.

4. Documentation:

III. System Testing

A. Test Goals: Verify ability of individually tested boards to function as a unit.

B. Test Setup: Final package and power supplies.

C. Test Procedure:

With a debugger board in the battery slot, go through each of the individual tests generated for the EP, IOP, and DD boards.

Replace debugger board with batteries and boot system from floppy disk. Verify total operation on some Smalltalk program running for a minimum of 24 hours.

Run a battery test program as described above to re-verify operation of batteries.

Plug an Alto display in the external monitor connector and verify that the CRT controller can be reprogrammed to run this external display.

Plug in a TI terminal and verify that this can be driven over the EIA port.

Connect output of D/A converter from heaphone jack into the external connector for the A/D converter and verify these external channels.

D. Documentation: