XEROX

PALO ALTO RESEARCH CENTER
Systems Science Laboratory
LSI Systems Area
December 8, 1979

For Xerox Internal Use Only

To: Distribution

From: Doug Fairbairn

Subject: NoteTaker2 Electronics Changes

stored: <fairbairn>NT2ElectronicsChanges.memo

We are about to start on the redesign of the electronics for NoteTaker2. I have identified a number of areas which I feel changes are called for and those are outlined below. I want to stimulate some lively but short-term discussion. We need to settle these issues as soon as possible so the detailed design can proceed. The ideas will be presented and discussed with the SPG personnel on Monday, Dec. 10th. During the week of the 17th, I would like the interested parties in LRG to get together with the SPG people and work out the details. I am targeting the end of the year as discussion cut-off time. After that there will be a strong resistance to any substantive changes. We will publish the results of our discussions in El Segundo and everyone should feel free to let their opinions be known on any of these issues.

Architecture

The major changes in this area involve the structure of the I/O system and the protocols on the System Bus. Before I get into the details let me review the overall state of the system. The machine has 9 board slots, all of which are wired the same. The proposed "standard configuration" is as follows:

Processor module (2) - 8086 processor, 12K bytes of EPROM or RAM. These can be traded off in increments of 4K bytes.

Reset, Boot, INT

Proposal: 4K bytes of EPROM, 8K bytes of RAM.

→ 12/17 decision on how much RAM is required.

In addition, each processor will have a debugging interface which allows an Alto or another NoteTaker to be plugged into it directly to aid in software development, debugging, and also as an external device interface.

Terminators for system bus lines.

Memory Control module (1) - Generates all control signals for main memory. In addition, this board has a real-time clock and an illegal address trap. The real-time clock has its own battery back-up so it will maintain the time when the device is not plugged in or when the main batteries are dead. The illegal address trap will detect accesses to a certain range of addresses and return an interrupt to the currently active processor.

Memory Storage modules (2) - Contains 40 RAM chips and a custom error-correction chip. The RAMs may be either 32K or 64K devices. The boards will be able to support either kind. With 32K RAMs, these two boards will contain 256K bytes of RAM. This number doubles with 64K chips.

1K x 8 - $28.65

→ double sided / double density.

Disk/Display module (1) - Contains the logic for the display controller, disk controller, and keyboard

new address field length.

Ethernet module (1) - Contains the Ethernet interface, EIA interface, A/D and D/A subsystems as well as the tablet drivers.

> Note: All of these functions will NOT fit on one board if we cannot get a Shared Line Controller chip from ED. The liklihood of getting such a chip is decreasing although still possible.

As can be seen, we will be using 7 slots in the "standard configuration" if all goes well. The extra 2 slots can be used for more memory, additional processors or other interfaces.

The major architectural change which has been proposed is to interface the slow I/O devices to the system bus instead of the 8086 bus. This scheme has the advantage that all the I/O devices are always available to any processor and would be a cleaner environment in which to experiment with multi-processors. This scheme has some disadvantages which need to be analyzed. The first is that interfacing the I/O devices to the system bus will take more logic and it isn't clear we have room. The second concerns the potential congestion on the system bus. For the system to be workable at all, we will require that each I/O device generate a status bit or bits which can be made available to all processors on their local bus. This would mean that the processor would only have to make a system access if it was known there was data available or that the device was ready for new data.

The above proposal means that we would also have to allow for I/O accesses on the system bus, but I believe this is wise in any case. A new handshake protocol for bus tranfers rather than the synchronous one currently used would be more appropriate in this new environment.

If we adopt the proposed scheme, it would mean we would not have to take 8086 buses off the processor boards, and that the size of the rear connector could decrease from 50 pins to perhaps 20 or 30 pins. These pins would then be used primarily for the remote status bits. Alternatively, we could include both the system bus and 16 status lines in a 50 pin conector if that seemed desirable.

Current plans call for the local memory address space to occupy high memory. Is this OK???

How many addresses should be set aside for the illegal address trap (Larry Tesler)

Fabrication

What should our strategy be for debugging the system? Stitch weld to PC boards (or multi-wire)

Hardware details

These are just odds and ends we have to come to some decision about.

1. Which chips do we want to put in sockets and which soldered directly into the board. No sockets has the advantage that there is better air flow across the boards because of more space between the chips and the next board, plus we wouldn't have to worry about chips falling out when we carried them. The obvious disadvantage is that they are much more difficult to service. Another alternative is HolTite socket pins which fit right in the board.

2. Advanced low-power schottkey chips are becoming available. They have half the power consumption and twice the speed of devices we use now. We need to identify the parts which will give us the most system improvement and least hastles in purchasing, etc.

3. There is a serious need for a volume control for the audio output channel. My feeling is that we could find some place to put it facing the rear of the Ethernet module.

4. We need a timer on all system bus interfaces so a processor does not hang forever if the system bus is not responding.

5. We need a well-defined system bus protocol.

6. Do we want to include a way to selectvely power off total subsystems like the Ethernet while not in use?

Color codes on each plug 12/17 7. How are we going to label I/O connectors on the rear so people know what to plug in where?

8. What should our strategy on memory chips be? (32K vs. 64K)

9. The fact that the display processor has to be told where to start the bitmap each frame time has proved to be very inconvenient. It means for example that the display cannot run while the disk is running.

10. It would be even nicer if we could include the extra logic which would allow for interlaced bit maps in main memory.

11. ECC chip strategy. 13. Document dictionary and strategy

12. BASIC SWITCH SWITCH. 14. Drawing notes (on schematics)

Testing

1. How can Signature Analysis be applied to best advantage?

2. What can we do with the architecture to make remote diagnostics more practical?

3. What is the best way to employ LEDs to identify problems. 15. Standard chips? (S/L, 7474, etc.)

4. Should we put LEDs on the power supply to verify all outputs are working?

5.

We have significant interest in making NoteTaker2 as easy to service as possible. One easily stated goal is to make it possible for a user to identify the module or peripheral which is failing without any test equipment.

This list is by no means complete. Please forward your comments and additions to the distribution list of this memo.

→ 24 bit address required.

If we continue to include an I/O bus connection on the rear of all processor cards, the following pin list would meet the requirements:

1-16: CPU bus 00-15
17: CpuBusHiEnbl'
18: Read'
19: Write;
20: DEN'
21: DataT/R'
22: M/IO
23: ALE
24: Hold
25: HoldAck
26: Test'
27: NMI
28: Peripheral Clock
29-44: StatusLines00-15
45: EN' (used during interrupt ack)

cc: Larry Tesler, Bruce Horn, Dan Ingalls, Ted Kaehler, Jim Altoff, SPG (Albert Pae, Bobo Wang, Doug Stewart, Ron Freeman, Ed Wakida)