

**XEROX**  
**PALO ALTO RESEARCH CENTER**  
*Systems Science Laboratory*  
*LSI Systems Area*  
May 8, 1978

**For Xerox Internal Use Only**

To:           Distribution  
From:         Doug Fairbairn  
Subject:       Desk-Top NoteTaker Proposal

stored: [IVY]<fairbairn>dtnt>dtntprop.memo

In discussing the NoteTaker design with various people, it has become clear that there is a critical need for a machine of the NoteTaker's class for use as a low cost, stand-alone work station. This machine would fill the large gap between a simple keystroke-capture station and the present D0 machine. ASD, among others, has expressed a strong interest in such a machine.

After carefully examining the requirements of the portable and desk-top versions of this machine, I have concluded that the two systems should be compatible at the module or board level, not at the package level. This of course means that the systems would be software compatible as well. The packaging needs of the two systems are just too different to allow for compatibility at this level. The desk-top system requires a large (15") display, an optional floppy disk and no batteries. The portable version needs a small (7") display, requires a mini-floppy disk, and has a significant volume set aside for batteries. The two machines might share the same power supply, however, keeping in mind that the batteries are optional.

The PC boards presently envisioned for use in the portable NoteTaker are an awkward size: 5" x 17". This size is not consistent with the needs of the desk-top version, and represents a manufacturing problem for either system. An additional negative aspect of these boards is that they are too large to include only one or two subsystems. Using Alto technology (MSI TTL), a new subsystem generally required 60-90 new chips. This was about the number of chips which fit on a board so that was a reasonable board size. With the level of integration proposed for the NoteTaker, each subsystem will require roughly 20-40 chips, thus making smaller boards the more logical choice.

Based primarily on the modularity and manufacturability arguments above, I propose that the two systems use a common module size of approximately 8.5" x 5". Amazingly enough, this is the same length to width ratio which Xerox standards propose as the maximum (1.7). This module size is the same height as the NoteTaker boards, but is half the length. The boards will have a 100 pin edge connector along the 5" side. In the current NoteTaker package, they would fit into the package as shown in Fig. 1. Eight of these modules could be plugged into the NoteTaker. The two separate back-planes which the cards plug into are connected by a flexible cable. A significant feature of this arrangement is that there is an unobstructed airflow over the cards.

A short historical perspective on how the NoteTaker architecture evolved would probably be helpful at this point. I spent a fair amount of time looking at the various processor-memory configurations which seemed reasonable for the NoteTaker system. The one which had the most appeal is diagrammed in Fig. 2. This diagram was generated when I was investigating how the D0/LSI components could be integrated into the NoteTaker architecture. In general this architecture is characterized by a large common memory shared

by multi-processors. The various processors have their own local memory as well. This local memory can take the form of a hardware-managed cache, a simple RAM, etc. In this particular diagram, the emulation processor has a content-addressable cache and the I/O processor has a straight RAM.

When it became clear that there were not going to be enough people to push the D0/LSI program ahead in the near future, I made the decision to merely replace the custom emulation processor with the 8086 machine. The I/O and memory systems remained essentially unchanged.

It is interesting to note that this architecture is essentially an updated version of the Alto architecture in the following respect. The Alto had several tasks running essentially simultaneously which communicated through main memory. These tasks of course did not run exactly concurrently because they all shared the same processor. In the NoteTaker architecture, the processes still communicate through main memory but in fact can now run truly simultaneously because of the multiprocessor configuration.

In summary, the current architecture is one we can grow with. It will support multi-processors and a large main memory (1M word). The processors can be of varying power and complexity, making use of various types of local memory. One can even choose to make the overall system simpler. The large main memory can be replaced by a very small one, possibly without error correction. The processors can use ROM to hold most of their local code with only the minimum amount of RAM.

A description of the potential module types is given below.

### 1.0 Main Memory

It will be assumed in this discussion that we are talking about a memory system with single bit error correction and double bit detection. The memory will operate as a 32 bit wide system with 7 error correction bits. The options which are available to us are 16K RAMs in a 16 pin package and 2 16K RAMs in an 18 pin package. Early next year 64K RAMs will become available in limited quantities and we will have the same packaging options for those chips. The following discussion will assume 16K RAMs.

The ideal memory module would include 64K x 16 bits of RAM using 18 pin packages, some address and data buffering, and a custom LSI chip to do memory data error correction. Without the custom LSI chip, all 32 bits from the memory would have to be brought to a connector on the back of the board. It would then connect to a module which did the error correction. After the error correction was done, the 32 bits would be multiplexed onto a 16 bit data bus available to all the cards in the system.

The memory timing circuits would share a board with the the system bus controller and the error logging processor.

### 2.0 Processors

There will be several processors in the system, each with its own local memory. They will all have access to the main memory and will communicate as described in the NoteTaker System Manual. The local memory may consist of RAM, PROM or both. 4K is probably the maximum amount of local memory which can be currently provided, although this can grow later on when 16K static chips become available and also if PROM or ROM can be used instead of RAM.

### 3.0 Sample System Configurations

A sample system configuration using all off-the-shelf components and 16K RAMs is shown below. This system could be built in moderate to large quantities in early 1979.

Function	# of Modules
128K x 16 Memory	2

Memory Control	1
Memory error corr.	1
Emulation Processor	1
Disk Controller	1/2
Display Controller	1/2
Gen. Purpose I/O	1/4
Ethernet Controller	3/4
<u>Total Modules:</u>	<u>7</u>

A system which could be available in small quantities in mid-1979 and which used a custom data error correction chip and display controller, 64K RAMs, a more capable floppy disk controller, etc. is shown below:

<u>Function</u>	<u># of Modules</u>
256K x 16 Memory	1
Memory/System Control	1
Emulation Processor	3/4
Display Controller	1/4
Disk Controller	1/4
Gen. Purpose I/O	1/4
Ethernet Controller	1/2
<u>Total Modules:</u>	<u>4</u>

The total power drain for these systems, including the mini-floppy disk but not the display is in the area of 50-100 watts.

These are obviously just sample systems, Other combinations and more custom LSI are possible. We need to identify more carefully where the costs are in these systems to see where the most leverage is.

#### 4.0 Future Options

All of the above systems include an 8086 processor as the emulation processor. If a more powerful processor is desired, there are the following options:

1) Use a bit-slice processor. This would require two boards in the above system and could be built out of existing components. The power requirements of this option are relatively high. The 8086 processor board will require 5-7 watts while the bit-slice approach would likely use about 30 watts. This last issue is only of great importance to the portable NoteTaker.

The computational power of this machine in the Mesa environment would probably be somewhat greater than the Alto. It may be that the most increase in power can be gained from a fast front-end processor for the 8086 which helps do the Mesa address calculations and perhaps helps on dispatching.

2) Design our own custom processor: e.g. D0/LSI. This would result in a very powerful system but would require at least a year of work on the part of several people.

3) Wait until Intel or some other vendor comes out with a new series of micro-processors. These may appear in 1979 and might be micro-programmable. They also may not be all that helpful. Three big maybes.

I will be looking into these and other alternatives in the next few weeks and will try to identify where the most leverage is in designing a machine to run Mesa.

#### 5.0 Software Compatibility

The only language presently being designed to run on the NoteTaker system is Smalltalk. If there were an interest, it would probably not be too difficult to modify the BCPL compiler to generate 8086 code instead of Nova code. There would also be utilities to be rewritten

and the run-time stuff to be redone. The recompiled code would almost certainly run faster on the NoteTaker than the Alto. This BCPL conversion seems a very short-sighted goal, however.

The real question is how well an 8086-based system can run Mesa, or at least a subset of it. I cannot presently give a realistic estimate of this.

## 6.0 Conclusions

I am strongly in favor of providing as much compatibility between the portable and desk-top versions of the NoteTaker as possible. I am also interested in proceeding as rapidly as possible through the first design of the NoteTaker system as it is presently configured. The emphasis in the near term will be to get the electrical design correct, and to build two functional prototypes. We cannot settle on the final physical packaging until Inova is farther along and until the needs of the desk-top version become clearer. I'd like to hear your reactions to the ideas presented here.

### Distribution:

- L. Conway
- J. Ellenby
- H. Hall
- A. Kay
- T. Mott
- B. Sutherland
- L. Tesler

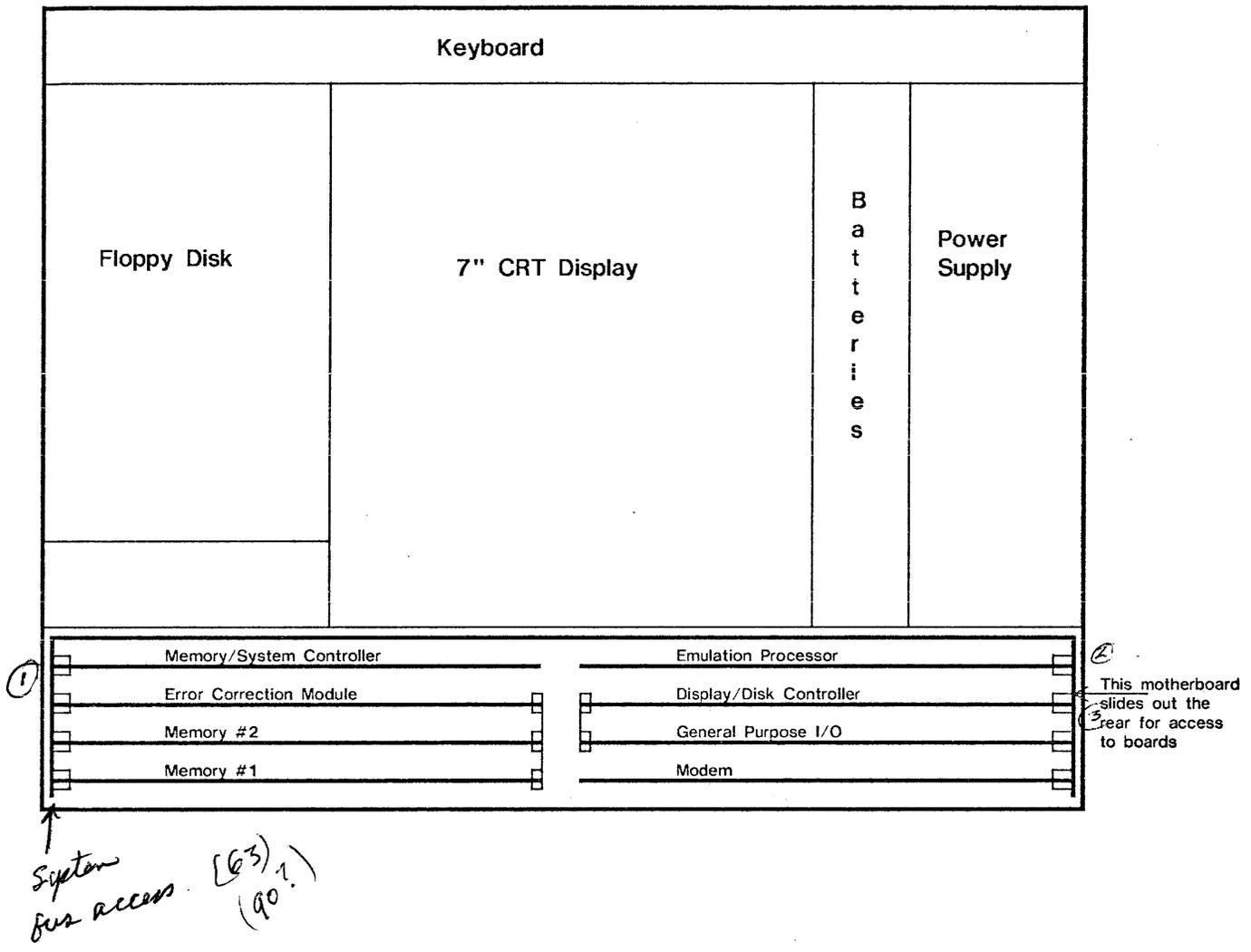
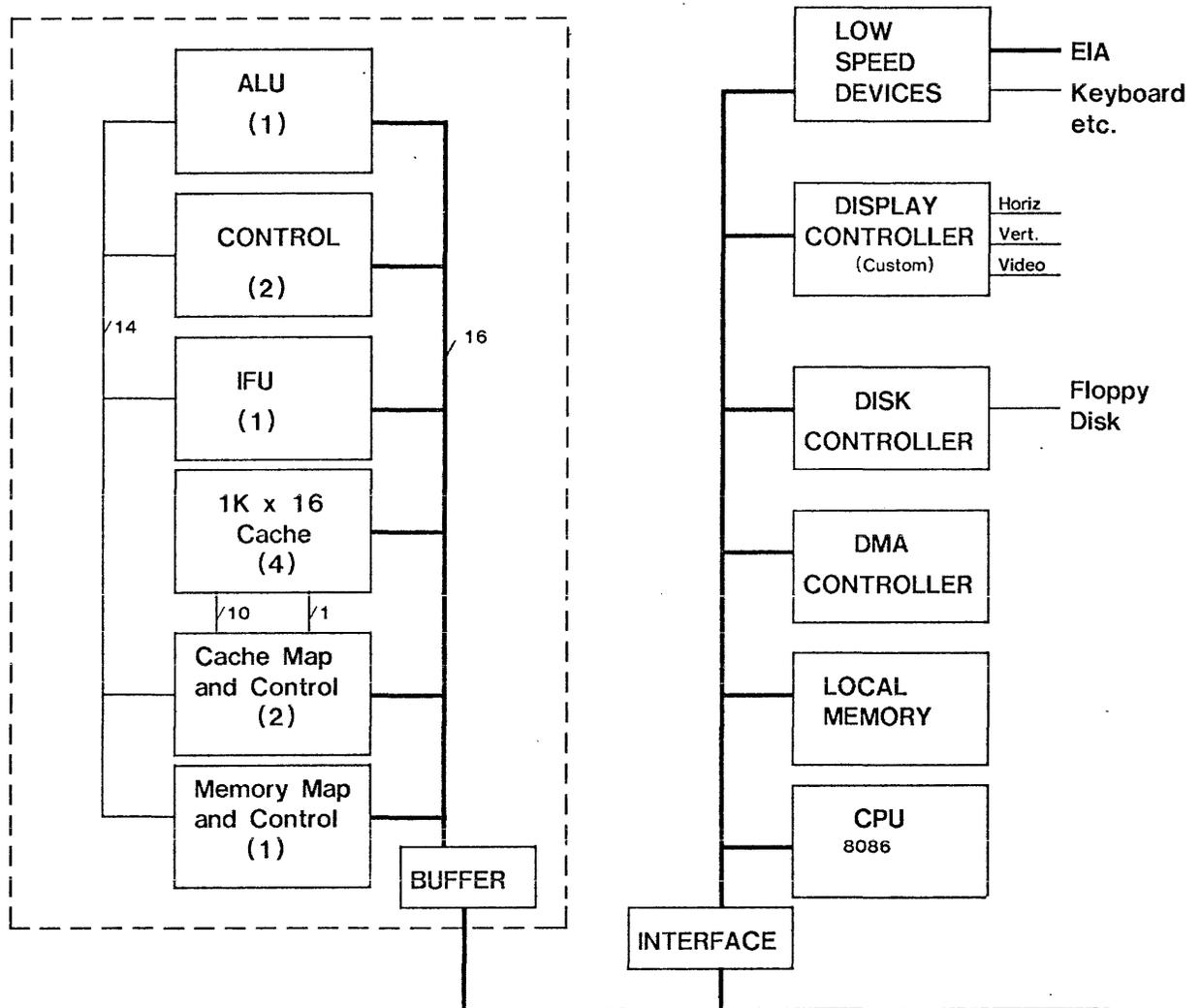


Fig. 1 - Module Placement in Portable NoteTaker

Fig. 2 - D0/NoteTaker Block Diagram



**ALU:**

- Mesa Stack
- Shifter/Masker
- Arithmetic Unit
- 2 Registers

**CONTROL:**

- Next Address Calculation
- Subroutines

**CACHE MAP/CONTROL:**

- Mesa base registers
- Adder for VA calculation
- Map for cache: (21A+7D)64

**CACHE:**

- Can be either custom or off the shelf RAM.

**MEMORY MAP/CONTROL**

- Mesa base registers
- Adder for VA calculation
- Map for memory: (16A+12D)64W
- Strokes out memory address in 8 bit bytes

**IFU:**

- Cache for 4 to 8 bytecodes
- Ability to put translate from bytecodes to microcode addresses
- Ability to place needed offsets on D bus from bytecode stream