XEROX

**BUSINESS SYSTEMS**

*Systems Development Department*

| | | | |
|---|---|---|---|
| To: | Distribution | Date: | 11 May 1978 |
| From: | Bob Metcalfe | Org: | SDD/SD System Architecture |
| Subject: | **LSI Digital Processor Program** | Filed: | [Iris]<Metcalfe>LSI-11May.Bravo |

Distribution: Bob Belleville, Howard Kakita, Bill Kennedy, Walt Klein, Butler Lampson, Hal Lazar
David Liddle, Bill Lynch, Bob Metcalfe, Bob Spinrad, Chuck Thacker, Tim Townsend
Jerry Szelong, Jim White, Ron Wickham

This memo is an initial response to the request that I work with ED to develop task plans for the LSI Digital Processor Program to which monies, headcount, and Altos have been allocated in 1978 via a transfer agreement for SDD responsibility spending in ED. At our (MSI) Digital Processor Program Review on 5 May, Walt Klein emphasized the importance of SDD collaboration in ED's development of an LSI ESS; Chuck Thacker and I met today for two hours to begin this collaboration.

Chuck and I did not discuss, but I propose that two important items in our as yet unwritten 1978 task plans be (1) a jointly developed *LSI Digital Processor Requirements Specification* completed in September 1978, establishing processor structure and performance and (2) ED-developed detailed task plans in time for SDD approval prior to the Star Design Phase Review in February 1979, with UMC and performance commitments allowing accurate establishment of LSI cut-in schedules and program financials.

Before descending to detailed task plans for the remainder of 1978, Chuck and I thought it best to formulate and seek answers to a few major strategy questions. In the following, I attempt some formulations and answers, intending that none be taken as programmatic commitments. Further, Chuck and I seem to see eye to eye on almost all points discussed, but I have taken the liberty here to state things only as I see them, not attempting to express consensus. Your (written) comments are invited.

## Why are we doing LSI?

The objective of the LSI Digital Processor Program is to reduce Star's UMC. UMC reductions will come mainly from lower power supply and packaging requirements. ESS UMC should not be considered independently; significant, if not dominant, UMC contributions come from peripherals.

[more]

## Which hardware interfaces should we preserve?

There are a number of hardware interfaces manifest in the MSI D0 program:

--> O 850 Bus Interface (from OSD)
O RS232C Interface (EIA Standard)
--> O Xerox Wire Transceiver Cable Interface (4 wires)
O Xerox Wire Coaxial Cable Interface (1 wire)
--> O User Terminal Interface (from UTVFC, 7 wires)
--> O ROS Interface (from Xenia, 9 wires)
O Shugart 800 Floppy Disk Interface
O Shugart 400X Rigid Disk Interface
O Computer Magnetic Tape Interface (Pertec with Formatter)
O D0 BackPlane Interface to Memories
O D0 BackPlane Interface to Controllers
--> O Serial Interface for Peripherals (PDSI-like? HDLC? 3Q78 specification)
O *D0 Functional Specification* (Microcode Interface)

A set of major strategy questions follow from considering which of these interfaces to preserve -- to take as a constraint -- while developing the LSI ESS. There are several major alternatives:

(1) Preserve MSI PCBA interfaces, plug compatible board for board.
(2) Preserve MSI controller interfaces, say 1 board for CPU plus controllers.
(3) Preserve MSI ESS interfaces; LSI ESS serving all MSI cables.
--> (4) Preserve MSI ESS interfaces except for user terminal and disks.
(5) Redevelop processor and all peripherals for maximum UMC reduction.

All of these alternatives should be considered further, but from here it looks like the first two are unattractive. They have advantages for risk minimization and ease of cut-over, but major UMC reductions from power and packaging are ruled out.

The third alternative saves peripheral development costs and decouples peripheral UMC reductions from ESS development by preserving all external ESS interfaces. However, our peripherals are expensive and already headed for cost reduction. And power and packaging are a major problem with our peripherals, evidence the so-called suitcases associated with our full page display user terminal (UTFP) and impact character printer.

The fifth alternative is frightening on the face of it, which leaves the fourth, constructed for further discussion below.

[more]

## Which software interfaces should we preserve?

There are a number of software interfaces manifest in the OIS core programs:

O *D0 Functional Specification* (Microcode Interface)
--> O *OIS Digital Processor Principles of Operation* (Mesa bytecodes and IO)
--> O *Mesa Language Manual*
--> O *Pilot Functional Specification*
--> O *Common Software Functional Specification* (TBD)
--> O *OIS Communication Protocol Specification*
--> O *OIS Print File Format Specification* (*a la* Press format)
--> O *OIS Document File Format Specification* (*a la* Diamond format)
--> O *Star Functional Specification* (user interface)

A set of major strategy questions follow from considering which of these interfaces to preserve -- to take as a constraint -- while developing the LSI ESS. There are several major alternatives:

(1) Preserve MSI D0 microcode compatibility. No software changes.
(2) Preserve PrincOps compatibility with no required Mesa source changes.
(3) Change PrincOps; capture in Mesa compiler; no Mesa source changes.
--> (4) Preserve Mesa PrincOps; capture IO and memory changes in Pilot.
(5) Capture changes in Common Software; no application changes.
(6) Preserve user interface, but require all software to change.
(7) Change user interface and all underlying software.

The first alternative is too restrictive and forces an MSI design on an LSI technology. The second and third alternatives allow the underlying technology to be applied appropriately while preserving our sizable software investment. The fourth alternative, from here, appears to be the most desirable along the continuum from zero to total software revision. The fifth alternative deserves further consideration, but the sixth and seventh are included in this list solely for their shock value.

[more]

## What about LSI design automation tools?

The successful development of a multichip LSI system, especially on an aggressive schedule like ours, requires advanced design automation tools. Development of such tools has been started at Parc and is already being contemplated in ED. Our task plans for LSI should include the acquisition and application of LSI development software. For example, Icarus and Merlin.

## Do we have a "venture" program?

Recent experience has shown that it is not sufficient to say simply that a program is, or is not, a "venture" program. Our task plans for LSI should include an explicit enumeration of those venture or non-venture activities to be performed. It is proposed, in particular, that the program have three major phases: (1) concept and feasibility, lasting a year, (2) definition and design, lasting several months, and then (3) production.

## What about off-the-shelf LSI?

Our requirements are such that off-the-shelf LSI chips will not deliver adequate performance, at least for the processor and memory. However, use of off-the-shelf LSI for peripheral controllers should be enabled by the incorporation of a standard LSI chip interface, like the 8080 Bus Interface.

## One exciting target LSI system.

LSI will permit significant UMC reductions, mainly in the areas of power and packaging. Some predictions put potential UMC savings at better than 50%. MSI performance can be achieved using LSI, but UMC, development cost, and risk can be significantly reduced, it is proposed, by going for performance below that of the MSI D0, closer to that required for one human user per ESS.

One exciting target LSI system, whose feasibility needs to be determined soon, has a single board packaged with a user terminal and two floppies on a desk top. The processor would have performance sufficient to support one human user. A 128K main memory would take 32 64K chips. A rigid disk would not be needed because of the large main memory and single user performance. The CPU would consist of 5 to 10 custom LSI chips of 2 or 3 types. Other chips (some off-the-shelf LSI and others custom LSI and others MSI) would support controllers for a user terminal (with full page display, mouse, and keyboard), a Xerox Wire transceiver, the two floppies, and serial interfaces for remote communication and peripherals. The most common LSI ESS, then, would be a user terminal on a desk top with two cables coming out, one to the (15 amp?) wall socket and the other to the Xerox Wire. Even the floppies could be optional with swapping and user files coming over the Xerox Wire from serving storage devices on other ESSs. Cost-reduced peripherals would connect to the ESS through its remote peripheral interface, the very same presumably HDLC-like serial interface already planned for specification in 3Q78.

[end] [illegible]