Wayne R

Nat Mackay-Jewsbury TW B02

→ Gordon Bell

Written Comments on NAT's.. This is really distressing. I Forest got "too busy" to do any real reading for us - now we know why. Every little C+ some big) going on in the Bay Area We had better .... a get off the ... about SUNAX as a product in some form soon. I don't think we're going to know about this SCORPIO is very smart - not out till '86 or so

***** Draft *****

The SUN Workstation

A Terminal System for the Stanford University Network

Forest Baskett, Andreas Bechtolsheim,
Bill Nowicki, and John Scamons

Stanford University Computer Science Department

Draft of March 30, 1980

why not put Nebula in the UT200? I'm willing to give it a go, but we need to keep the drive up. — Wayne 5/30/80

We have designed Ethernet-based workstations that support clusters of graphical or conventional terminals. This terminal system gives users the capability to communicate with any host computer on the Stanford University Network (SUN), which is currently being installed.

The workstation provides up to 8 high-resolution graphical (bit-map) displays with 1280×819 pixels, or up to 32 medium-resolution channels with 640×409 pixels. Each station is controlled by a 16-bit microprocessor (an MC68000) that emulates standard terminals, such as Datamedias, and supports a network graphics protocol.

familia

The same basic hardware can also be configured to be a ASCII terminal/dial-up line concentrator, an Ethernet gateway, or a special device controller (e.g. for laser printers or disks). By adding memory, the hardware provides growth into a powerful personal computer system, suited for design automation (VLSI), advanced text processing (CRTEX), and other applications demanding dedicated computing power and high-resolution graphics.

↳ real-time TEX - what we want

A hardware and software prototype is scheduled for July 1, 1980. We plan to fabricate a small batch of workstations over the summer to support the VLSI design automation project. The gross cost of a station without displays is estimated to be $8000. Additional display channels are estimated to cost $500 for a low-resolution display and $2000 for a high-resolution display (excluding keyboards and monitors). Thus the gross cost of a terminal system with 16 low-resolution displays is estimated to be $16000 per user.

familiar?

1. Introduction

In the near future all major campus-wide computing resources will be connected by an Ethernet communication system. This network will provide us with an unprecedented level of system integration and thus will be extremely valuable to the entire user community.

The Ethernet [Metcalfe] is a bit-serial, broadcast, multi-drop packet switching network that allows up to 256 stations to be connected via a single coaxial cable, up to 1 km long, by simply tapping the wire. Access arbitration is achieved by deferring a transmission until the channel is idle, aborting a transmission if a collision occurs and retransmitting after a random interval.

The main advantage of the Ethernet is that it allows a large number of users and servers to communicate over a single channel at high speed. Essentially, the network offers the same communication mechanisms as a conventional time-sharing system. Computer mail and messages can be exchanged over the network, shared data bases can be kept at centralized disk-servers, and unique hardware devices such as laser-printers can be shared among their users. The Ethernet has a bandwidth of 3 Mbit/sec that allows to support all these services.

This network technology together with the advance of semiconductor technology is fundamentally changing the nature of computing facilities. It is becoming economically and organizationaly advantageous to distribute computing power. High-end microcomputers of today have equal performance to small main-frames of the past, and it is becoming practical to dedicate such systems for a single or a few users or for a given task. The availability of localized computing power is critically important for three purposes:

1) Interactive applications, such as program development or VLSI design, where the productivity of a researcher greatly depends on the response time of the system.
2) Compute-bound applications, such as simulations or AI programs, where the cycles provided by a powerful personal machine exceed those of a typical time-shared environment.
3) High-speed device applications, such as raster-scan graphics, laser-printers, and audio, with processing and latency demands that cannot be accomodated in time-shared systems.

In addition, a distributed system is inherently more reliable than a centralized one.

These considerations are now widely recognized and have been explored in a number of existing or proposed machines, such as the Alto [Thacker], Dorado [McDaniel], Lisp-Machine[Greenblatt], Nu-Computer system [Ward], PERQ [Rosen], and SPICE [Newell]. The approach we are proposing is most similar to the MIT Nu system, in that we desire a flexible solution for our immediate needs that will be available as soon as possible, utilizing standard, off-the-shelf LSI technology. We are currently less concerned about obtaining a very high-performance personal computer, such as a Dorado. Rather we see our system evolve naturally towards higher performance with the progress of VLSI semiconductor technology.

Our most immediate need is a low-cost, clustered terminal system, and a personal computer system with high-resolution graphics for design automation and advanced text processing. In addition, our hardware design can be configured as an Ethernet terminal concentrater and as an Ethernet gateway. A plan to build this hardware and the associated software is the core of this report.

A 780 for VLSI design — not a scorpio.

1.1. Current State of the SUN Ethernet

Currently, all the Xerox equipment (16 Altos, the DOVER printer, and the IFS file server) and the VLSI VAX are fully functional Ethernet devices. The SAIL computer is electrically connected through the front-end PDP-11, but the device driver for the Ethernet interface needs yet to be written. The other machines are awaiting the fabrication of our Ethernet interface boards. These interfaces will connect SCORE via the MassBus, the IBM 4331 via a Series/1 machine, and other equipment, such as the TI-990 machines, via their own I/O busses or the IEEE 488 standard bus.

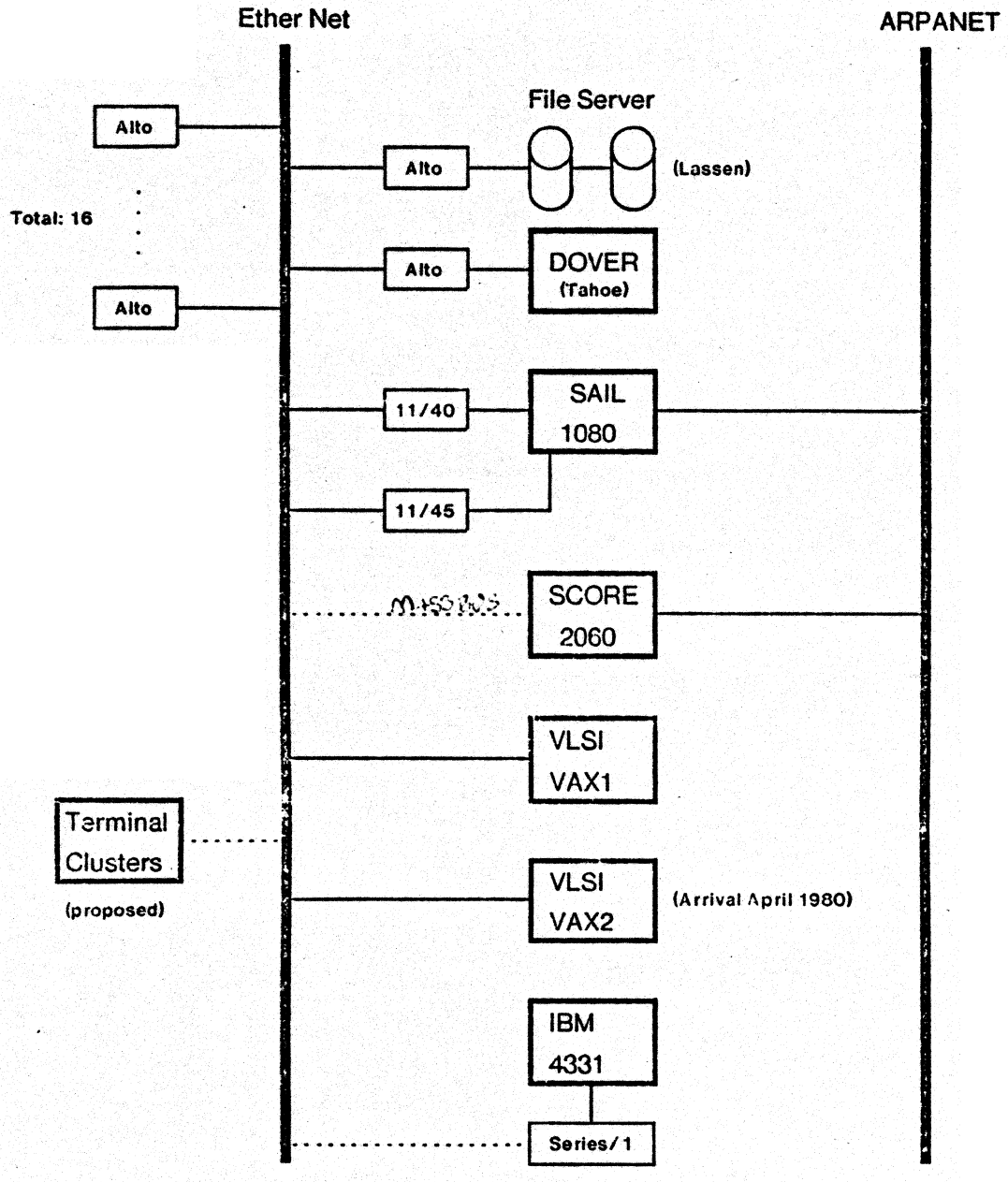


Figure 1: Status of SUNet March 1980 <SUN>proposal.fig1.sil

1.2. Plans for a Campus-Wide Ethernet

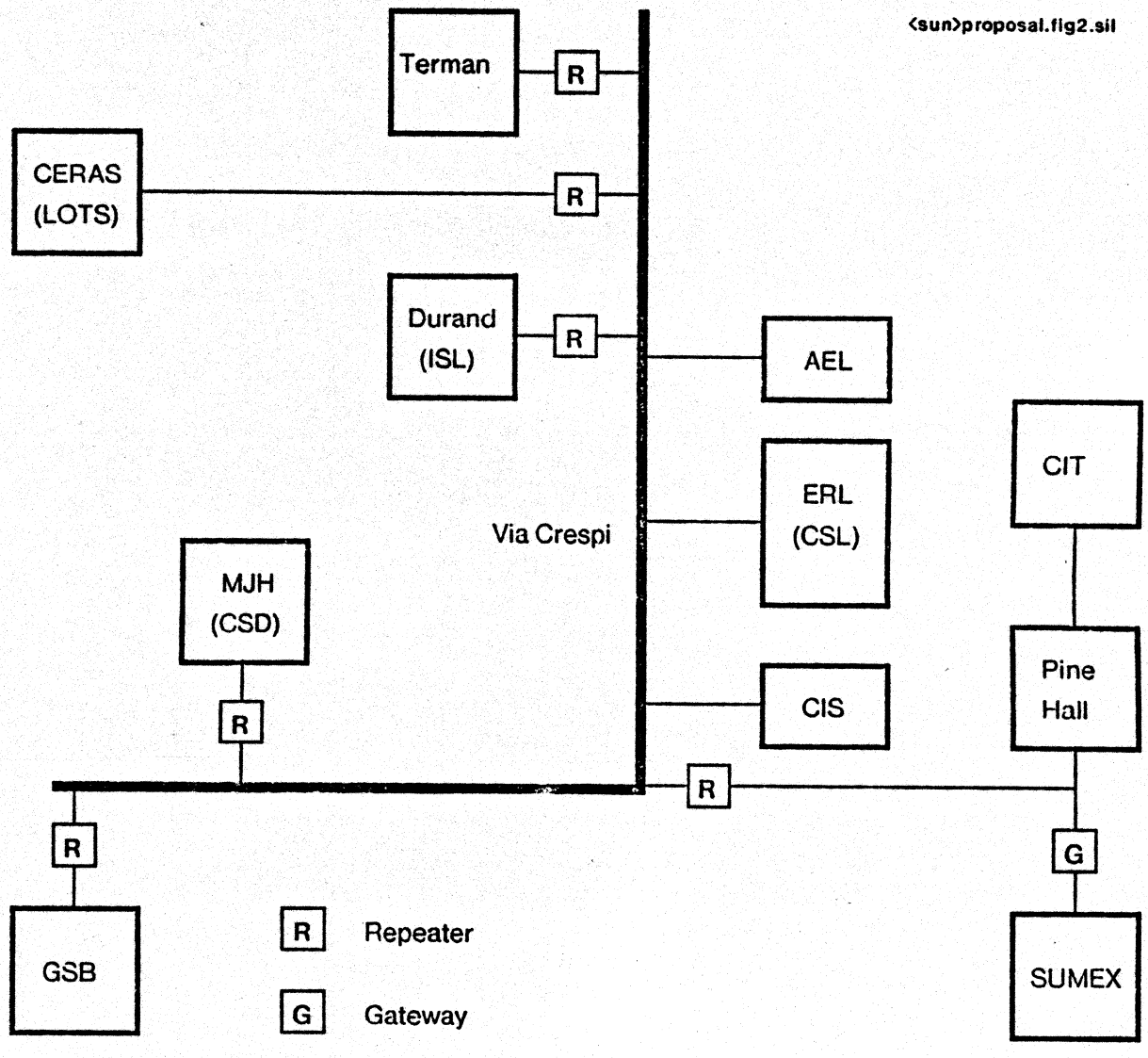
lets do it for TW

The current schedule for the installation of Ethernet cable calls for three phases:

Phase 1: Margaret Jacks Hall. This phase has been completed.

Phase 2: A "backbone" cable will run from the North corner of the main Quadrangle along *Via Crespi.* From this cable, "spurs" with Ethernet repeaters will connect Pine Hall, the Center for Integrated Systems Annex, the Electronics Research Laboratory, the Applied Electronics Laboratory, the Durand building, and the Terman building. Contractors are currently working on this phase. This phase is scheduled for completion by May 1, 1980.

Phase 3: The "backbone" cable will be connected to the Medical Center Facility (SUMEX), the Graduate School of Business, the Center for Information Technology in Forsythe Hall, and the LOTS computer in the CERAS building. SUMEX could possibly be connected during Phase 2 via Pine Hall. Phase 3 is scheduled for fiscal 1981.



2. The Architecture of the SUN Station

The main goals of the hardware designed are to provide a flexible and powerful human interface, to support the Ethernet communication architecture, and to provide localized processing power. We shall first discuss the interactive input/output and then briefly describe the actual hardware components and the suggested configurations.

2.1. Interactive Output

We plan to use bit-map raster-scan graphics displays as output devices that provide a choice of high-resolution or medium-resolution monochrome, greyscale, or color displays.

A raster display is one of the most general output device available today. It can represent characters of arbitrary size and style, vectors and curves, solid and grey areas, and it can simulate half-tone images [Thacker, Newman-Sproull]. A frame buffer display is also very economical: given current dynamic memory costs of less than 0.05 cents per bit, a single-bit per pixel frame buffer is about as expensive as the attached video monitor, be it a low-, medium, or high-resolution display. As the price of dynamic RAMs continues to decline, we expect to see increasingly large frame buffers in the future, providing enhanced resolution, grey-scale, and color.

Our design provides two kinds of display resolution:
High-resolution displays with 1024×1024 pixels (or 1280×819), and
Medium-resolution displays with 512×512 pixels (or 640×409).
For both resolutions there is the option of using multiple bits per pixel which can be mapped into colors or grey-scale through a look-up table.

The high-resolution display is the configuration of choice for design automation, text processing, and advanced program interfaces. A high-resolution frame buffer contains four times as many pixels as a Data-Disc display, two times as many pixels as an Alto display. It can either display a high-quality image of a standard sheet of paper in "portrait" mode, or two sheets side by side in "landscape" mode. The medium-resolution format is comparable to the current Data-Disc displays and will probably be used in "landscape" mode. It provides compatibility with low-cost video monitors, including our current Ball monitors, and is suited for economical color and grey-scale systems.

Besides the resolution of a raster display, its update speed is a crucial factor. Our goal was to change a complete high-resolution frame buffer without a noticable delay. Specifically, the entire screen should be scrolled within 64 milliseconds. The number of bits that must be accessed, shifted, masked, and modified in that time requires significant processing power. We have developed a novel frame buffer organization which reduces processing demands to a level where a single 16-bit microcomputer can serve a number of frame buffers. In brief, a small amount of special hardware implements a "RasterOp" rectangle manipulation function [Newman-Sproull] that makes it possible to modify rasters in the frame buffer at full memory bandwidth (32 Mbit/sec) without processor intervention. The host processor only needs to setup the source and destination location, the height and width of the rectangle, and the bit-operation desired. Excluding this overhead, a 16×16 character can be put into the frame buffer in 16 microseconds, and a 1024×1024 RasterOp takes 64 milliseconds.

2.2 Interactive Input

We plan to use unencoded keyboards in which any number of keys can be depressed simultaneously to represent desired functions. This provides edit keys, META keys, or any number of shift keys desired.

The issue of a "standard" keyboard is a difficult one. Most users can do with an Alto-like keyboard, maybe with additional programmable keys that can be redefined. The concept of an unencoded keyboard allows the internal microprocessor to simulate any desired keyboard to the host computer, but changing the key tops won't be as easy.

For design automation and other graphics applications, a pointing device, such as a mouse or a tablet, is essential. Unfortunately, pointing devices cost about $500 and are probably too expensive for use in the clustered terminal configuration.

2.3 Hardware Components

In brief, there are three main hardware components: a self-contained Ethernet interface, an MC68000 microcomputer system with serial and parallel ports, and a high-performance graphics subsystem. These components plug into an industry standard backplane, the Intel Multibus.

The Ethernet interface was designed to work with any host computer without making short latency or significant service demands. It is a full-duplex interface incorporating 16-word buffers for receiving and transmitting, collision filtering, programmable address filtering, hardware CRC generation and checking, and automatic retransmission. The address filter is specified as a bit-vector, which allows it to respond to any set of the 256 Ethernet addresses. Extensive self-test and diagnostic facilities are provided.

The MC68000 microcomputer system acts as the main controller of the station, handling the Ethernet communication and serving all the attached devices. It is a standard single-board computer with 64 kbytes of dynamic memory, two serial interfaces (for keyboards and tablets) and two parallel ports (for special devices such as laser printers). The design is optimized for speed and includes no memory management, which we felt not to be critical for the initial applications. We are also designing a virtual memory microcomputer system with a dual 68000 processor which is discussed in [Baskett].

The graphics subsystem consists of two modules, a graphics controller performing the raster operation and generating the sync signals, and a frame buffer module containing 1024×1024 bits which can be reconfigured into four channels of 512×512. Up to eight frame buffers can share one graphics controller module. For example, a display system with 16 medium resolution channels needs one graphics controller and four frame buffer modules. The frame buffers support refresh rates up to 64 Mbit/sec, making them compatible with practically all kinds of video monitors, interlaced and non-interlaced, color or monochrome.

All the hardware modules described plug into the Intel Multibus backplane. The Multibus is currently being standardized by the IEEE as the 796 Bus. It is an asynchronous bus with 16 data and 20 address lines, supporting transfers at a rate of up to 5 MHz. There is a second, non-

dedicated 60-pin backplane connector for intermodule communication. This connector is utilized in the graphics subsystem and we intend to use it as a future high-speed memory bus. Standard Multibus boards measure 12 by 6.75 inches and hold about 100 16-pin packages. The Multibus was selected over many alternative busses considered because of its performance, its board-size, and its wide acceptance in the marketplace. There are more than 50 suppliers of Multibus compatible components, ranging from bubble memory to speech recognition subsystems. This supplier base is an advantage when we need disk controllers or other I/O devices.

All hardware was designed with the SUDS design automation system. Most of the design was initially expressed in a graphical macro language similar to SCALD, but more geared towards efficient small-scale design. SUDS produces the wirelists for building wirewrap prototypes, and we plan to use SUDS for laying out the PC boards for the pilot production run.

2.4 Configuration of the Ethernet Stations

The following physical constraints apply to all configurations.

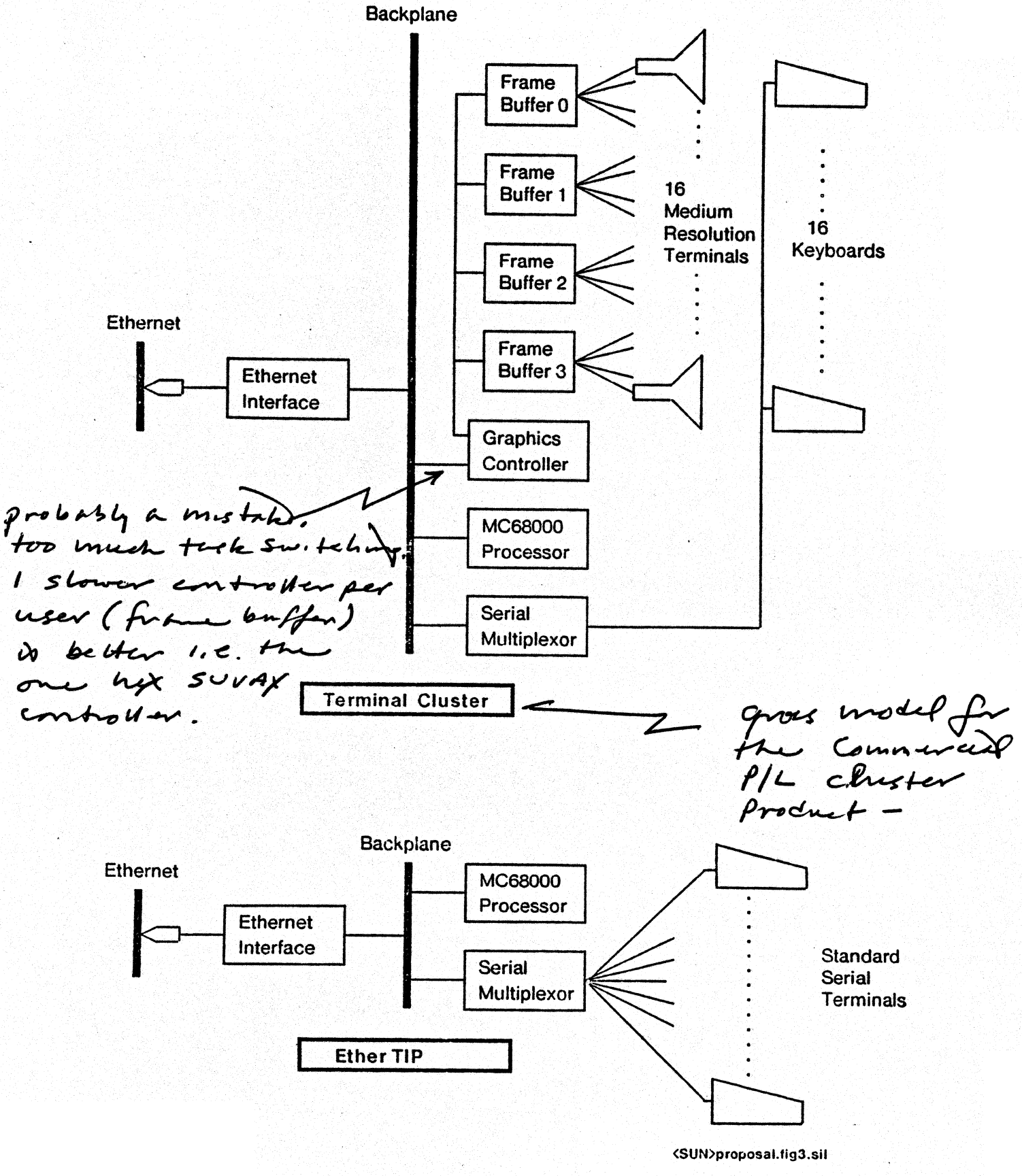
Maximum length from Ethernet Transceiver to Station:	40 feet
Maximum length from Station to Video Monitors:	150 feet
Physical size of terminal configuration:	20"x12"x15" (w,h,d)
Power consumption of 16-channel station (excl. monitors):	300 Watt, 110 Volt

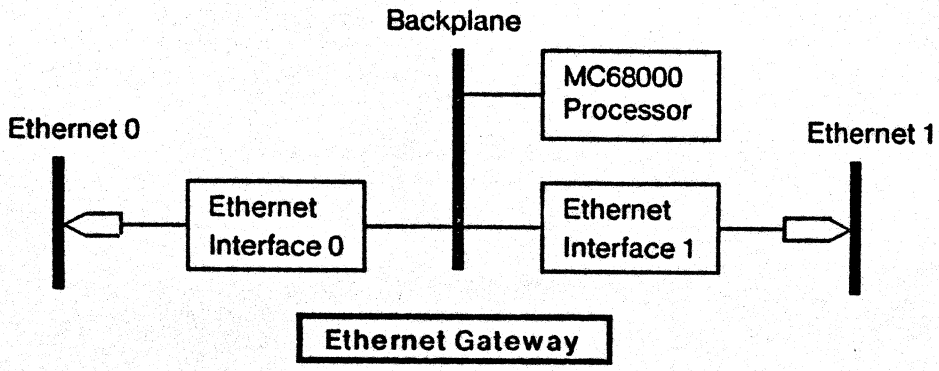
The Terminal Cluster configuration shown has 16 medium-resolution display channels. Since a station supports up to eight frame buffers, a maximum of 8 high-resolution or 32 medium-resolution displays are possible, as long as all displays have the same resolution. Because of wiring constraints, it is probably best if the station is located in the same room or nearby.

The Ethernet TIP configuration can be used where many conventional terminals are currently located, such as in ERL or Terman. One TIP can also replace the current Arpanet TIP or the dial-up lines to SCORE or SAIL. An Ethernet TIP provides a central place for dial-up access to all machines from home terminals, without the expense of adding line multiplexer and telephone equipment to machines like the VAXes and the IBM 4331.

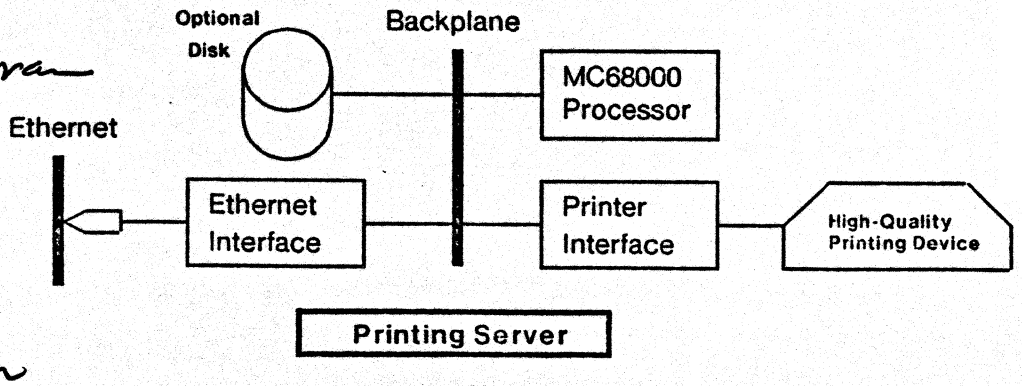
The Ethernet Gateway is simply a dedicated station with two Ethernet interfaces. Similarly, a printing server is a dedicated station with device specific interfaces. The department-wide utility of hardcopy devices coupled with the modest cost of the Ethernet station makes them especially attractive candidates for Ethernet devices.

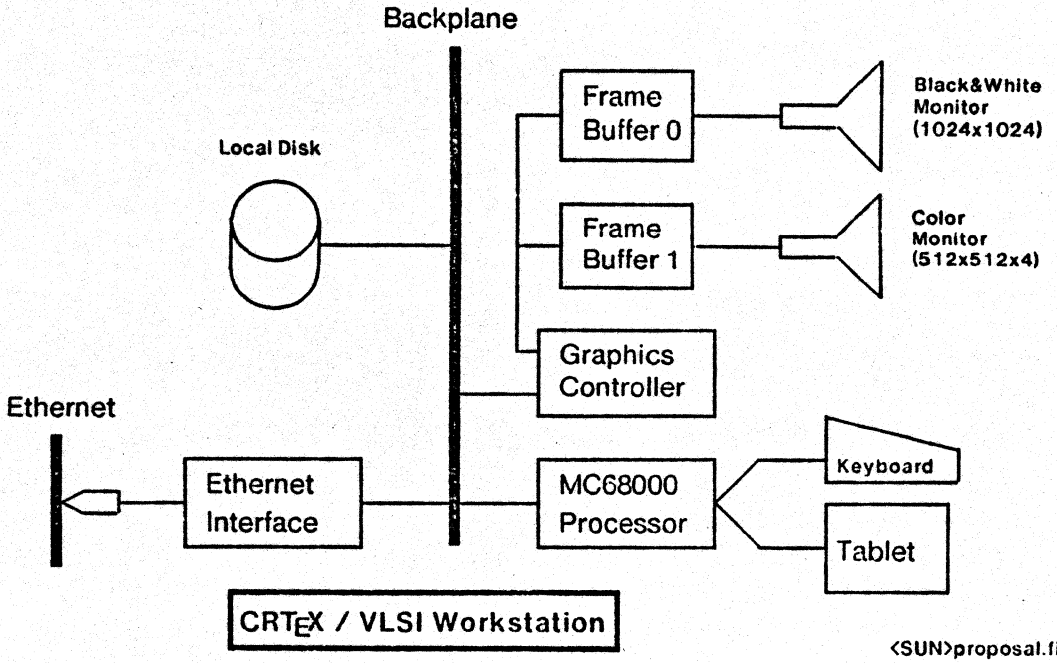
The CRTEX/VLSI workstation is a station with a high-resolution display, keyboard, and tablet. For VLSI design, we also want color graphics capabilities. When the hardware becomes available, we plan to use the virtual 68000 system with signifcant amounts of primary memory and a Manchester Disk for secondary storage.


Backplane
Frame Buffer 0
Frame Buffer 1
Frame Buffer 2
Frame Buffer 3
16 Medium Resolution Terminals
16 Keyboards
Ethernet
Ethernet Interface
Graphics Controller
MC68000 Processor
Serial Multiplexor
probably a mistake. too much task switching. 1 slower controller per user (frame buffer) do better i.e. than one NeX SUNAX controller.
Terminal Cluster
gives model for the Commercial P/L cluster Product -
Backplane
Ethernet
Ethernet Interface
MC68000 Processor
Serial Multiplexor
Standard Serial Terminals
Ether TIP
<SUN>proposal.fig3.sil



block diagram is identical to the one used in the SUVAX HW talk at Stratton





<SUN>proposal.fig4.sil

3. Software Architecture

We intend to use the Xerox PUP (*PARC Universal Packet*) architecture for Ethernet communication. PUP has been used inside Xerox since 1976 and is the basis for their internetwork software and hardware designs [Boggs]. The fundamental abstraction of PUP communication is an end-to-end media-independent internetwork datagram. In this case, the datagram takes the form of a single packet traversing the network. Higher levels of functionality are achieved by protocols that are strictly a matter of agreement among the communicating end processes. Thus, the PUP architecture is idealy suited for a dynamically changing environment. Given the research nature of our department, there is a great advantage in providing an expandable network architecture that will provide efficient interaction among users.

Network resources are best viewed as *User* and *Server* processes. The DOVER laser printer, for example, is a printing server that can respond to requests from any arbitrary host that implements the corresponding user process. This distinction allows a resource to function as an independent entity on the network, not directly tied to a particular mainframe or its operating system. The effort required in providing a new resource to a host is greatly simplified because the server process presents a high-level interface to the network. For example, the DOVER printing service was added to the VAX machine over a weekend by simply installing the proper user-mode software to generate PUPs on the Ethernet connection.

It is important to note that some local processing element must exist between every independent resource server and the Ethernet. Xerox has used the Alto minicomputer for this purpose. The DOVER printer and IFS file server each have their own Alto to handle service requests from the network. In our own environment, it is unacceptable to use Altos as server machines (we can't afford them). Although most of the departmental peripherals use an existing host machine as an interface, there is a critical need for a low-cost processing element to interface new devices that provide network services. The SUN terminal system can be easily reconfigured to implement a variety of servers.

For the display stations, only the basic PUP Telnet protocols need be implemented on the workstation's MC68000 processor. The SUN terminals could then be programmed to emulate currently available terminals, such as Datamedias, Telerays, Tektronix 4014 graphics terminals, or IBM displays. This would allow compatibility with the extensive software on existing host machines, an important consideration.

Development of protocols and interfaces to our current operating systems is progressing in parallel with the construction of the hardware. The implementation of the PUP software is designed to be portable to several machines besides the SUN terminal. In particular, software for the SUN terminal can be partially derived from other implementations of the PUP package (eg, the VAX project described below).

Specifically, there are five levels of software than must be implemented (see figure 5). The lowest level (PUP level 0) controls the Ethernet interface board. Although the Ethernet interface described in this proposal provides a "standard" for Ethernet interfacing, there are still small implementation differences between hosts. For example, the interface board is a memory bus device on SAIL, but a MassBus device on SCORE and the VAX. The level 0 software must deal with each host's I/O mechanisms. This component of the software is the simplest and will require one man-week per

host to write on systems with simple I/O mechanisms like Unix and SUN.

Level 1 functions provide a uniform interface to the PUP world. PUP packets are filtered and routed at this level. Most of the level 1 functions deal with gateway routing, which need only be implemented on gateway machines. Without gateway routing level 1 should require two man-week to implement. Software for gateway machines will require an additional three man-weeks.

Level 2 provides three protocol interfaces. The Rendezvous & Termination Protocol (RTP) initiates, manages, and terminates connections between hosts. Simple file transfers and DOVER spooling requests use the Easy File Transfer Protocol (EFTP). More complex functions such as Telnet and FTP require the Byte Stream Protocol (BSP). We have many implementation examples of this software. The Alto software for these protocols is written in both BCPL and Mesa. Versions of EFTP exist in the C language under Unix and in assembler code for PDP-11s. There is an ongoing project between Stanford and CMU to implement all four PUP levels in C under Unix. This software, along with MIT's C compiler for the MC68000, can also be used by the display station. In addition, we are exchanging additional software with CMU, MIT, and Berkeley on a regular basis. Stanford has developed software to print TEX, XGP, and TROFF files on the DOVER. CMU and MIT are working on PUP packages and printing software. Accordingly, level 2 functions can be implemented in three man-weeks.

Level 3 contains the user and server processes. They are: FTP User/Server for exchanging files, Telnet User/Server to allow remote logins between hosts, and an EmPress style printing program for the DOVER. FTP and Telnet require RTP and BSP from level 2, while EmPress needs only EFTP. The implementation times vary depending on the requirements of the host machine. Level 3 Telnet can be written in three man-weeks.

Level 4 is made up of specific terminal emulators and display packages. We can lessen the current terminal shortage by emulating existing terminals such as the DataMedia and Teleray. DataDisks and even III displays could be emulated (a real concern given the condition of the current equipment).

SUN Telnet, together with PUP levels 0, 1, 2, and an appropriate emulator will require 10 man-weeks of implementation time. More sophisticated applications, such as a gateway or Ether TIP, will require other level 3 components. Notice that once levels 0, 1, and 2 have been defined, additional functions are only a matter of writting the level 3 user and/or server programs.

The Telnet software developement schedule is shown in figure 6. It is calibrated in weeks, beginning April 1. These predictions reflect our best estimates given the limited manpower involved. Essentially there is one person writting the Unix/SUN software and one person doing the SAIL Telnet side. Additional help is expected in writing the terminal emulators. This schedule will be somewhat shortened if we can use the CMU level 1 and level 2 PUP package they are developing. Actual results are expected in early June provided the prototype hardware is ready.

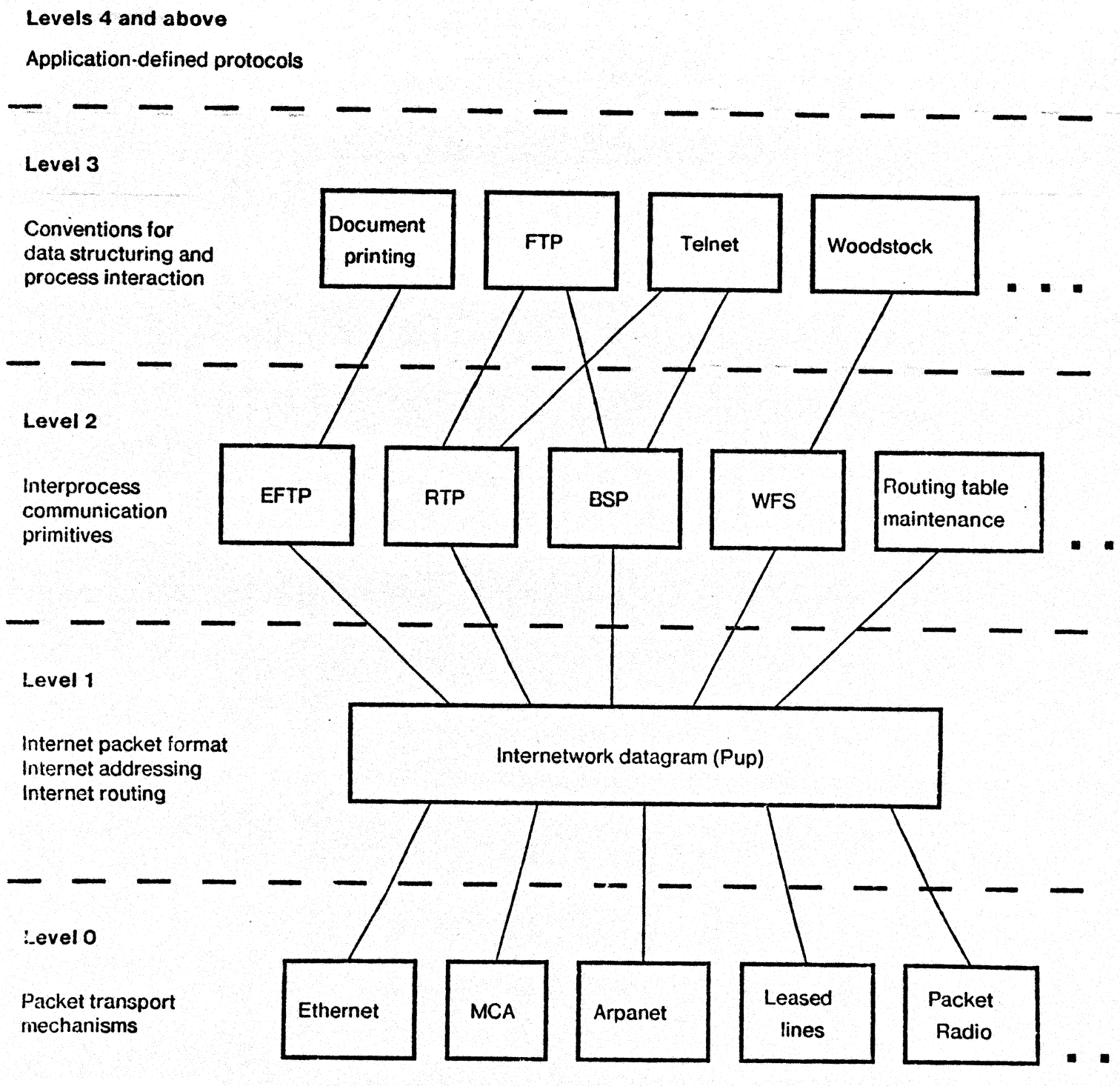


Figure 5. Pup Protocol Hierarchy

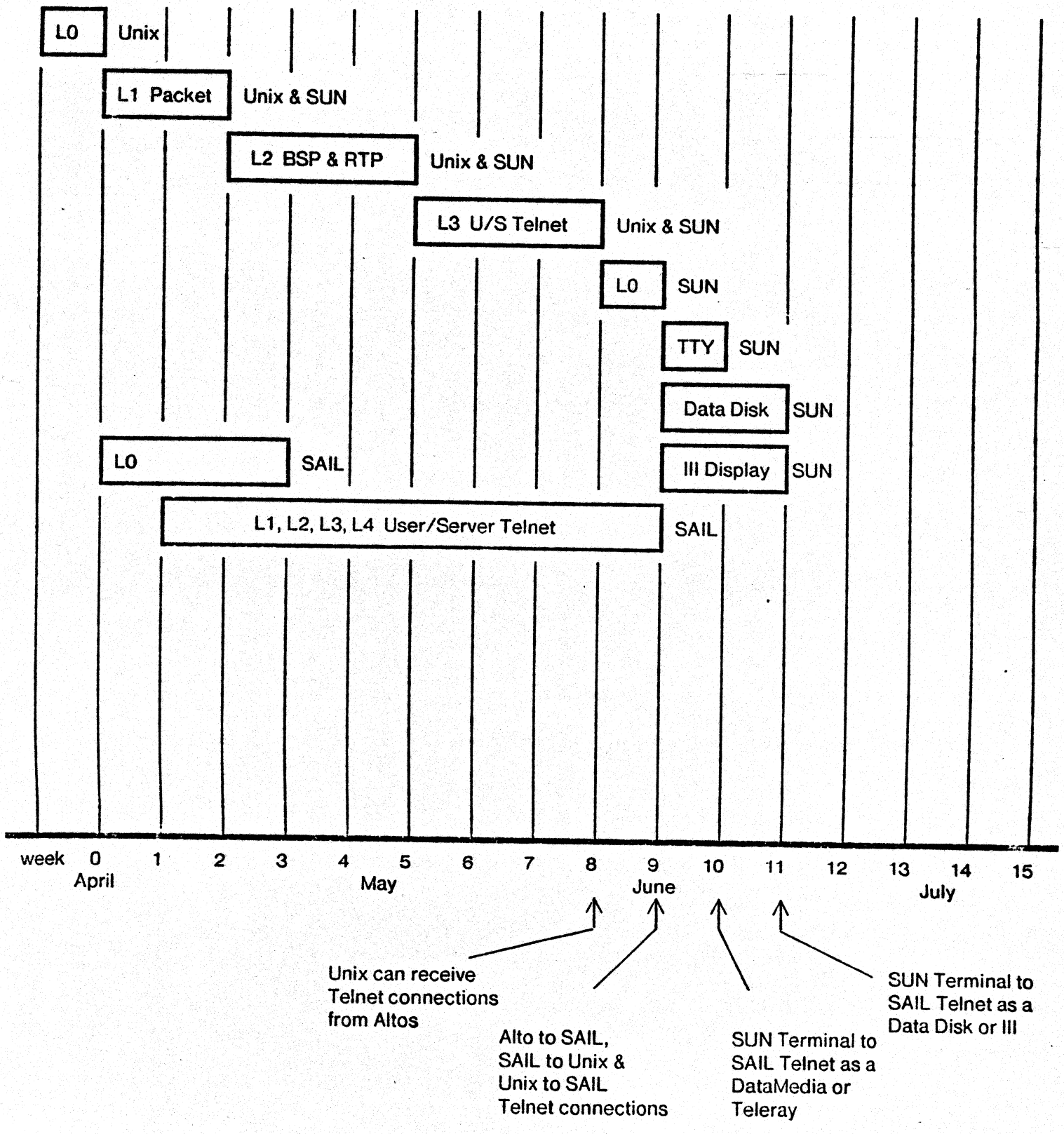
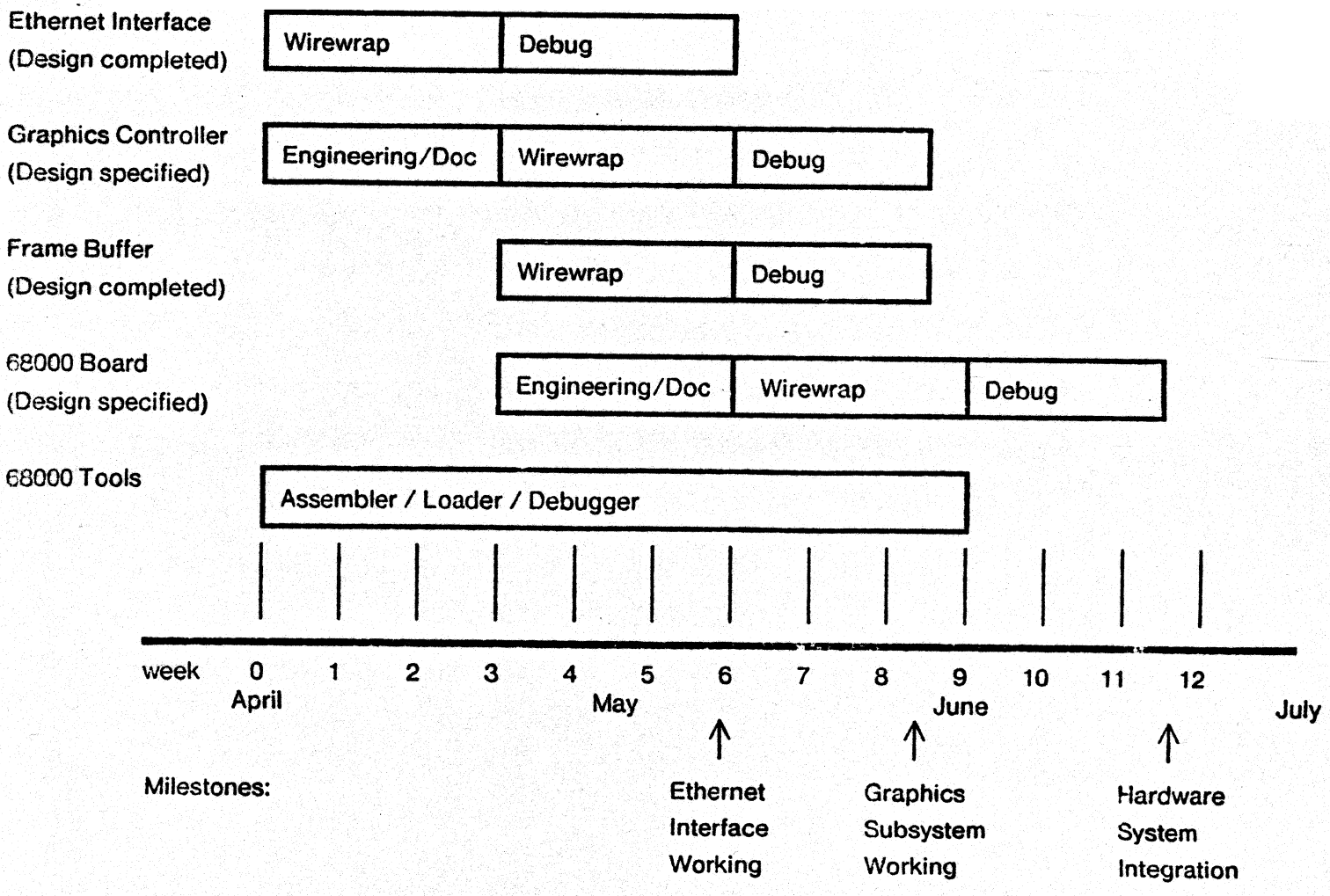


Figure 6. Telnet Software Developement Schedule

STANFORD CSD	Project SUNet	Reference Software Schedule	File proposal.fig6.sil	Designer John Seamons	Rev F	Date 3/31/80	Page 1



STANFORD CSD	Project SUNet	Reference Hardware Schedule	File proposal.fig7.sil	Designer A. Bechtolshein	Rev E	Date 3/26/80	Page 1

References

[Baskett]
Forest Baskett, Pascal and Virtual Memory in a Z8000 or MC68000 Based Design Station, *COMPCON* 1980.

[Boggs]
David R. Boggs, John F. Soch, Edward A. Taft, and Robert M. Metcalfe, Pup: An Internetwork Architecture, *IEEE Transactions on Communications*, April 1980.

[Greenblatt]
Richard Greenblatt, MIT's LISP Machine, *COMPCON* 1980.

[Metcalfe]
Robert Metcalfe and David Boggs, Ethernet: Distributed Packet Switching for Local Computer Networks, *Communications of the ACM*, volume 19 number 7, July 1976.

[McDaniel]
The Dorodo: A Compact, High-Performance Personal Computer for Computer Scientists, *COMPCON* 1980.

[Newell]
A. Newell, S. E. Fahlman, R. F. Sproull, and H. D. Wactlar, CMU Proposal for Scientific Personal Computing, *COMPCON* 1980.

[Newman-Sproull]
William M. Newman and Robert F. Sproull, *Principles of Interactive Computer Graphics*, 1979.

[Rosen]
Brian Rosen, PERQ: A Comercially Available Personal Scientific Computer, *COMPCON* 1980.

[Sproull]
Robert Sproull, Elaine Thomas, A Network Graphics Protocol, *Computer Graphics*, volume 8 number 3, Fall 1974.

[Thacker]
C. P. Thacker, E. M. McCreight, B. W. Lampson, R. F. Sproull, and D. R. Boggs, Alto: A personal Computer, *Computer Structures: Readings and Examples* (Siewiorek, Bell, and Newell, eds.) 1979.

[Ward]
Stephen Ward and Chris Terman, An Approach to Personal Computing, *COMPCON* 1980.