LA-7718-MS
Informal Report
UC-32
Issued: March 1979

The Integrated Computer Network High-Speed Parallel Interface

Roger B. Frank

THE INTEGRATED COMPUTER NETWORK HIGH-SPEED PARALLEL INTERFACE

by

Roger B. Frank

ABSTRACT

As the number and variety of computers within Los Alamos Scientific Laboratory's Central Computer Facility grows, the need for a standard, high-speed intercomputer interface has become more apparent. This report details the development of a High-Speed Parallel Interface (HSPI) from conceptual through implementation stages to meet current and future needs for large-scale network computing within the Integrated Computer Network.

INTRODUCTION

Computing requirements at the Los Alamos Scientific Laboratory (LASL) and at similar installations throughout the country have seen a trend toward large-scale networking. This trend has been accelerated as the magnitude of problems as well as their number have increased more quickly than mainframe technology.

Links between several large-scale computers allow many large tasks to be accomplished at one facility simultaneously. However, interconnecting valuable resources, such as Control Data Corporation (CDC) 7600s and Cray Research, Inc. (CRI) CRAY-1 computers, must be accomplished in an efficient manner to maximize the total capability of the network.

HISTORY

Computer manufacturers have traditionally attempted to create bigger, better, and faster mainframes. Computational speed for the central processor and input/output speed to attached peripherals were always maximized. Little effort was placed on intercomputer

capabilities. The task of connecting two computers of different manufacture has traditionally been left to the engineers at each facility.

When LASL had only a few computers, intercomputer connections were developed to meet the needs as seen at the time. As new computers from different manufacturers were acquired, more interfaces were developed. Some of those interfaces are typified by the following examples.

RTLI - Remote Terminal Line Interface. A synchronous serial link for communication between two PDP-11 processors.

CDC 6600 synchronizer to a Digital Equipment Corporation (DEC) PDP-11 coupler. A 12-bit parallel interface between a CDC 6600 and a PDP-11 Unibus.

CRAY-1 to System Engineering Laboratories (SEL) 32/55 interface. A CRI-designed 16-bit parallel interface with high-speed burst capabilities.

DESIGN PHILOSOPHY

It was clear that much could be gained by standardizing on an interface between computers. As the number and variety of computers grew, the task of designing, constructing and maintaining several different intercomputer interfaces became more difficult. As an example of this, a traditional network connecting four dissimilar computers is shown in Fig. 1. Notice that six different interfaces are required (types A, B, C, D, E, and F) for the four computers (numbered 1, 2, 3, and 4). In all, 12 different chassis need be designed, programmed, and tested.

If an interface could be designed for each computer's channel or internal bus that would translate their unique signals to a standard set, then any computer could easily be connected to any other computer. This can be accomplished only if each computer's interface can talk to a standard link. Figure 2 shows the same four computers connected through compatible interfaces. Notice that each computer has three copies of the same interface, which need be designed only once for each computer. A total of four designs must be completed: one for each dissimilar computer.

A real advantage is apparent when a new computer is added to an existing network that uses a standard interface (Fig. 3). To integrate the new computer, only one design need be undertaken: that of interfacing to the standard signal protocol. Then for each interconnection between the newly added computer and existing machines, only duplicates of the new machine's interface have to be constructed. Once this is accomplished, only additional copies of each existing computer's interface must be added to each computer, and no new designs need be undertaken on existing network computers.

A less obvious advantage to the standard interface approach applies to the testing phase of a new design. Since the interface

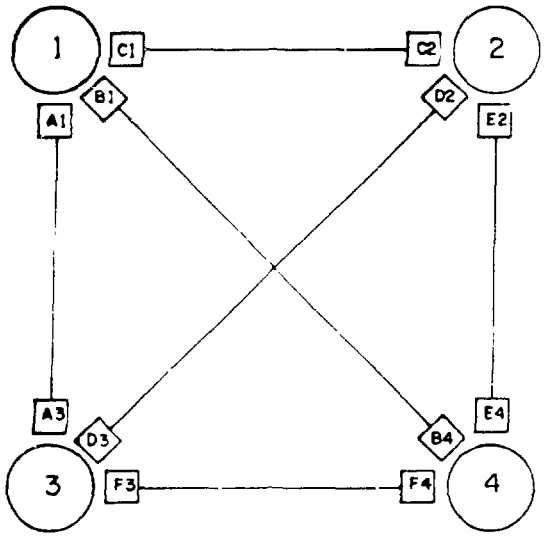

Fig. 1. Traditional network using several different interfaces.

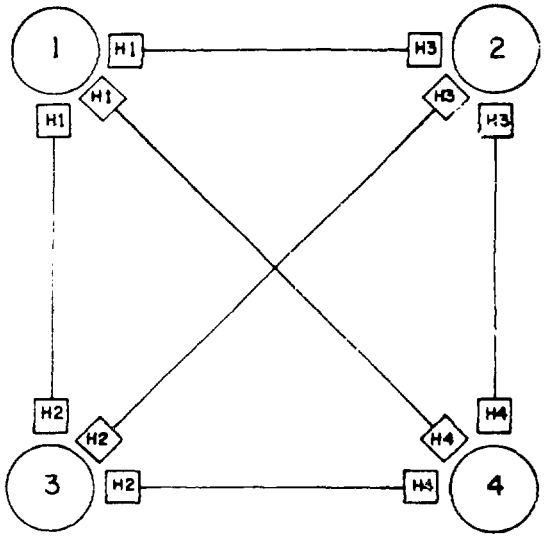

Fig. 2. Equivalent network using standard interfaces.

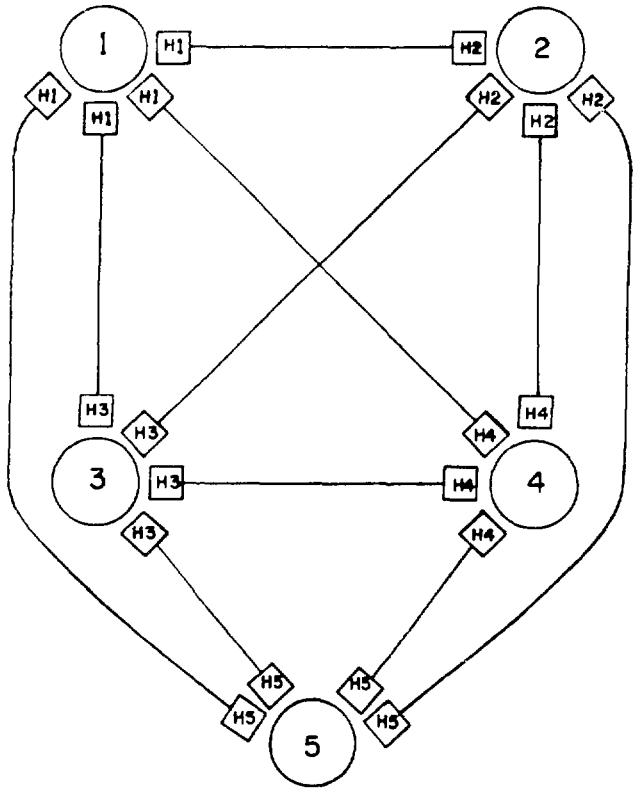

Fig. 3. Expanded network.

translates a computer's internal signals to a standard set, all testing can be accomplished either in a "loop" fashion, with the computer talking to itself, or with any other available computer, because the interface is the same in all cases. This is important when the cost of hardware development time on supercomputers is considered.

To utilize such a standard link, several factors must be considered. First, its speed should be faster than the fastest computers that will use it. If this were not true, then the fast computers would have to wait for the link to load or unload data. This should be avoided where possible. Data transmission rates for some typical LASL computers include:

CDC 7600	40 Mbits/s
CDC 6600	12 Mbits/s
CDC Cyber 73	24 Mbits/s
DEC PDP-11 Massbus	16 Mbits/s
DEC PDP-11 Unibus	3 Mbits/s
CRI CRAY-1	80 Mbits/s
SEL 32/55	40 Mbits/s

Another important characteristic is that it should be full duplex. Having independent transmission and reception paths significantly increases throughput capability, simplifies the hardware, and allows the use of well understood communications software.

A major consideration is that the standard link be as error-free as possible. Circuitry that can add to the reliability of the network and can also verify its own reliability must be included if the implementation is reasonable. Error correction can also be integrated into such a link for increased integrity.

Finally, the design philosophy must stress simplicity. A clean, uncomplicated design will be more reliable since fewer components will be required. Thus the probability of component failure will be reduced due to reduced complexity. Also, a modular approach to design and construction allows rapid checkout as well as circuit transportability to other HSPI designs. With simplicity and modularity in mind during the design phase, the cost of each additional interface will be minimized. An added benefit is increased maintainability and reduced mean-time-to-repair.

If the design is simple enough, the standard interface can be used for other than intercomputer communications. For example, a data collection system should be able to support the necessary logic to tie in to the standard interface specification.

IMPLEMENTATION

Using the guidelines outlined in the previous section, work was begun in the spring of 1978 on the first such interface. The High-Speed Parallel Interface (HSPI) was to incorporate the best features of proven interface designs developed at LASL and to add new features to produce a true state-of-the-art interface.

The design goals were further clarified into real-world numbers that would meet or exceed the requirements of the CCF computers. The design specifications for the HSPI became:

- full duplex, independent channels,
- automatic single-bit error detection and correction,
- automatic double-bit error detection,
- centralized reporting of errors with limited local logging,
- error-forcing capabilities for verification of error-checking circuitry, and
- rates up to 60 Mbits/s using 30 m cables or 40 Mbits/s using 100 m cables.

Given these design goals, a computer (the DEC PDP-11) was chosen for the first HSPI interface. Within the CCF there are 16 DEC PDP-11 computers. They are used as message synchronizers, as concentrators, and as security and status machines for the network. Thus it was decided that an intercomputer interface between PDP-11s with the capabilities of the HSPI should be constructed first.

This design effort was started in June 1978 and completed in October 1978. The design goals were met, and the PDP-11 HSPI is an example for follow-on HSPIs for other computers. Much of the circuitry developed is transportable in that the HSPI can be adapted to other machines by modifying only those portions associated with the particular computer's characteristics.

PDP-11 HSPI

The block diagram for the PDP-11 HSPI is shown in Fig. 4. The top half of the diagram represents the transmitter, while the lower half is the companion receiver. Both share a common Unibus interface, but using independent controllers, described below, can achieve full duplex capabilities.

The PDP-11 Unibus is an internal bus to the PDP-11 on which processor, memory, and peripherals reside. The HSPI interface is such a peripheral, assigned a block of 16 addresses in the PDP-11 peripheral address space. The Unibus Interface, shown to the left of the figure, is the part of the HSPI that is unique to the PDP-11. The Unibus interface handshakes with the Unibus to fetch words from memory for transmission to another computer through the HSPI as well as putting words received from another HSPI into the PDP-11's memory. These actions are initiated by either the transmitter (TX) or receiver (RX) controllers, depending on whether the TX controller needs data or the RX controller has data for the PDP-11.

The Unibus interface is a functional block with only a handful of control signals. To the transmitter, which fetches data from

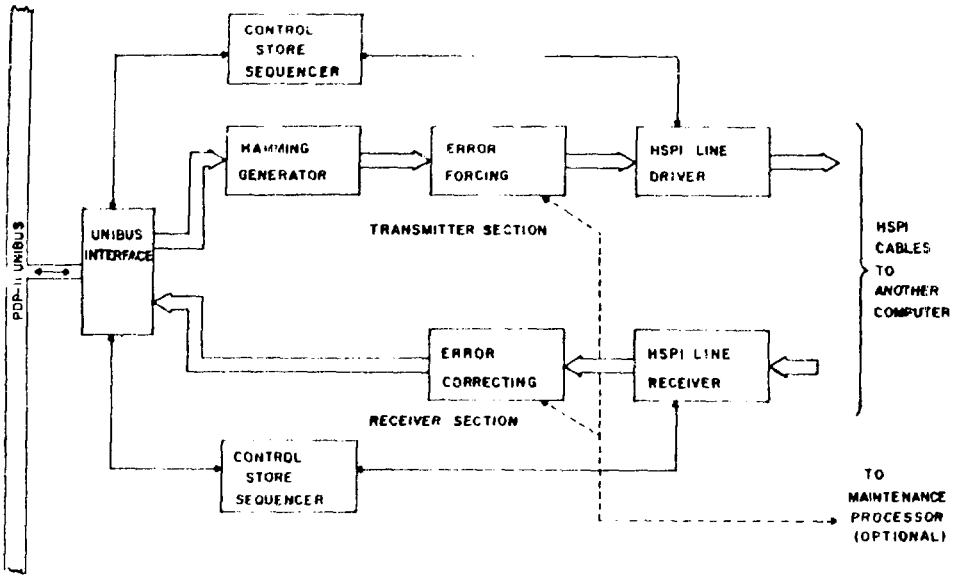

Fig. 4. PDP-11 High-Speed Parallel Interface.

the Unibus, the signals used are (1) a request for data from memory, sent to the Unibus Interface, (2) an acknowledgment that a word has been fetched from memory back from the Unibus Interface, and (3) an error signal in case the word was not fetched properly for any reason. For the receiver half of the interface, the similar signals are (1) that a word is ready to be transferred to Unibus memory, (2) an acknowledgment that the word has been placed in memory from the Unibus Interface, and (3) an error signal in case the word was not stored properly for any reason. With this modular approach, the Unibus Interface could be replaced by another computer's interface and the rest of the circuit would remain unchanged.

The Unibus Interface is the only common circuitry in the HSPI to both the TX and RX sections. Requests can be "stacked" in the Unibus Interface, so both the TX and RX circuits seem to have their own Unibus Interface. Speed limitations on the PDP-11 Unibus itself required a top access rate of approximately 3 Mbits/s for each half of the link, or about 6 Mbits/s total. Higher rates would lock out the Unibus from doing anything else while the HSPI was accessing the bus.

To the right of the Unibus Interface in the block diagram, the two data paths through the HSPI are shown. The top half is for transmission; the lower half is for receiving. Both are controlled by their own control store sequencers. Each sequencer has a programmable read-only memory, which is preprogrammed to control the data transfers through the unit. Both sequencers are identical with variation only in the read-only memories for each.

Looking at the transmitter first, data is requested from the Unibus interface by the sequencer. When it is available, it follows the data path to the Hamming Generator block. This circuitry

creates 6 additional bits of information that are passed on in parallel with the 16 bits of information from the Unibus. These 22 bits form the traditional Hamming code for single-bit error correction and double-bit error detection. This combined 22-bit code is processed throughout the HSPI between any two interfaces and allows for increased reliability and data integrity.

After the Hamming Generator, the data is routed through error-forcing circuitry. This circuitry can force a single-bit error as explained in detail later. The 22 bits of data, including the one bit in error, will be transmitted through the line drivers shown in the upper right of the diagram. The line drivers create differential signals that drive standard LASL coupler cables containing 26 twisted pairs of wire.

Transmitted data can go to any other HSPI interface in the CCF. The companion computer, with its HSPI, can simultaneously transmit data to the HSPI, which will receive it at the lower right corner of the block diagram.

The line receivers shown there convert the differential signals, used for high noise immunity, to standard logic levels. The 22 bits of information received are presented to the error-correction block, which examines the composite word and determines whether errors have occurred. Single bit errors are automatically corrected, whereas double-bit errors cannot be corrected and cause the PDP-11 to note the error through the sequencer.

Finally, under control of the receiver sequencer, data is presented to the Unibus interface and on to the PDP-11 memory through direct memory access.

Overlaying the data paths just described is an important feature designed into the HSPI for error handling. Additional logic is incorporated to support a maintenance processor, which connects to each HSPI chassis through a rear panel plug. The maintenance processor is not required for HSPI operation; however, when it is installed, it adds important capabilities to the HSPI. As mentioned above, logic exists in the transmitter for forcing errors. The maintenance processor is connected to this block and can force errors as often as it likes and in any bit position, even in the checking bits themselves. This data will be sent to the receiving HSPI, where its error-correcting circuitry will correct the error. The maintenance processor is also connected to the receiving HSPI. The error-correcting circuitry there will report the error to the maintenance processor after automatically correcting the faulty bit. This allows the maintenance processor not only to keep logs of unexpected errors that occur but to also verify that the error-correcting and detecting circuitry does indeed work by forcing errors at selected intervals. This can be done while production data is being shipped through the HSPIs.

FUTURE APPLICATIONS

The PDP-11 HSPI is the only production-ready HSPI at this time. Work is in progress to develop HSPIs for the IBM 370/148 and the SEL 32/55. Both designs should be complete by spring of 1979. Additional interfaces are in the planning stages. It is

anticipated that all new CCF intercomputer links will use the HSPI standard interface specification.

SUMMARY

The result of this network enhancement will be a faster, more reliable means of shipping data between the computers in the network. The use of a standard channel will allow simpler, more electrically reliable designs. The use of Hamming error correction and detection will provide increased data reliability. The use of a common parallel path will allow easy interconnect between dissimilar computers. Finally, the use of parallel data paths allows high rates compatible with the fastest network computers.

As the number and variety of computers at LASL has grown, the need for this type of interface has become steadily more apparent. By incorporating HSPIs into the network now, we hope to better meet the large-scale computing needs of LASL in the future.

BIBLIOGRAPHY

1. H. Ferman Kelso, "Overview of LASL Integrated Computer Network," Los Alamos Scientific Laboratory report LA-6576-MS (January 1977).

2. John A. Brockmeyer and A. Gene Dornhoff, "Computer Interfacing at LASL," Proc. AESOP Conf., Denver, Colorado, September 28-30, 1976 (U.S. Government Printing Office, 1977), Vol. 15, pp. 336-360.

3. D. E. Tolmie, "Using Minicomputers at LASL to Network Large Computers," Proc. Conf. Digital Equipment Computer Users Soc., Atlanta, Georgia, March 26, 1976 (Digital Equipment Corporation, 1976), Vol. 2, No. 4, pp. 935-940.

4. Norman R. Morse, "The Role of Supercomputers at LASL," Los Alamos Scientific Laboratory report LA-6750-MS (March 1977).

5. Fred W. Dorr, "Computer Science and Services Division CCF Five Year Plan," Los Alamos Scientific Laboratory report LA-6921-MS (August 1977)

6. Roger B. Frank, "High Speed Parallel Interface Design Specification," Los Alamos Scientific Laboratory, unpublished data, August 1978.

7. Donald E. Tolmie and Roger B. Frank, "The LASL Network Switch," presented at the Third Conference on Local Computer Networking, Minneapolis, Minnesota, October 23-24, 1978.