SLAC-PUB-5513
May 1991
(A)

SLC'S ADAPTATION OF THE ALS HIGH PERFORMANCE SERIAL LINK*

K. Krauter
Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94309 USA

Abstract -

Keep It Simple Network (KISNet) is a very high speed, low overhead, primarily point-to-point network adapted from the LBL Advanced Light Source (ALS) control system. KISNet implements the ALS system in the SLAC Linear Collider (SLC) environment.

Commercially available communication hardware and protocols do not meet the per-small-packet transmission time requirements of the initial application that motivated the development of KISNet. Limited resources restricted what could be developed from scratch.

In the SLC environment, the communications system is used in a slightly different way than at ALS. The software is layered to enable insertion of future hardware as it becomes available. Several enhancements for the future of KISNet are under consideration.

KISNet REQUIREMENTS

The Fast Feedback Project (FFP) [1] for the SLAC Linear Collider (SLC) includes a new high speed micro-to-micro communication path. The microprocessors along the length of the SLC communicate with each other only via a single centralized VAX,® with the exception of a few micros currently in communication with each other over low-performance broadband CAT/CAR links. This intervention of the centralized VAX® during micro-to-micro communications could not be afforded by FFP, so a micro-to-micro communication path concept was born, and the architecture requirements were drafted.

Most important:

- Performance had to be very high for small packet throughput, on the order of a millisecond, in order to operate on the scale at which SLC beam operates (120 Hz).
- The new network had to be operational when FFP was scheduled for use, within about eight months of when the new network decision process was begun.
- The new network had to fit within the modest FFP budget.
- The new network had to be general enough that it would support more clients than just FFP.
- The new network had to be structured enough such that a minimum amount of software and hardware needed to be re-implemented in the future event of replacing the new network with yet a newer one.

Less important but still desirable:

- The new network should be easily extendable as new needs for it are identified in the future; this ties into the nature of FFP which will expand a few micros at a time.
- Budgetary and schedule constraints would be more easily met if the new network consisted of standard commercially available hardware and software, but the performance constraint makes this unlikely.

The single requirement which encompasses most of the above requirements may be stated in the well-known design principle of "Keep It Simple," which led to choosing the name K-I-S-Net for the new network.

DECIDING ON KISNet

General SLC-wide network architectures were almost all eliminated from consideration because of the expense of stringing cable the entire length of SLC. FFP required only a small minority of the micros be connected at first, anyway. Further, most of the general purpose networks have significant overhead processing per data packet, exceeding the FFP performance requirement. In addition to these general drawbacks, each proposed network had its own drawbacks.

An Ethernet connecting all the micros seemed a poor choice due to the nature of the communication expected by FFP: namely in between each beam, all the micros would want to talk to each other simultaneously, resulting in significant packet collision recovery overhead.

A Shared Memory architecture was proposed which involved relocating all the micros in one room where they would talk to each other via common memory, and talk to their respective devices via wire links from the central room to each device's location in SLC. This proposal exceeded the FFP budget and schedule requirements.

The choices were narrowed down to point-to-point communication protocols specifically between those micros that needed to talk to each other. Cable would only be strung between those micros that needed to talk to each other, which only becomes more costly than one SLC-wide general cable as many micros are involved. Only two nodes on each cable need be supported, minimizing collision errors and possibly packet overhead. Extension of the intramicro communication paths requires more cable be strung up, which was a penalty offset by the advantage of being minimal of the initial installation.

Intel's iNA® Ethernet product installed on a point-to-point basis removed the drawbacks mentioned above for an SLC-wide ethernet. However, performance numbers supplied by Intel (on the order of six milliseconds small data packet throughput) exceeded FFP's requirements.

LBL's Advanced Light Source (ALS) [2] project developed a special-purpose network applicable to a point-to-point installation, with many of the same performance and cost constraints as KISNet. Throughput time is on the order of a hundred microseconds per 100 bytes.

An agreement was made to pattern KISNet after the ALS network, buying into their hardware and firmware. Our own driver code would be necessary because of differences between the ALS and SLC environment.

---

* Work supported by Department of Energy contract DE-AC03-76SF00515.

Presented at the IEEE Particle Accelerator Conference, San Francisco, CA, May 6-9, 1991

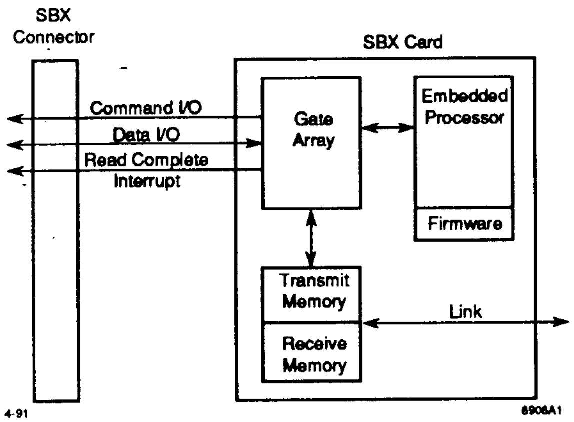

Figure 1. Node architecture.

ARCHITECTURE SUMMARY

Here is a summary of the ALS network architecture after which KISNet is patterned. The hardware at each node consists of an SBX® card with a gate array that handshakes with the microprocessor, some memory that is hooked to the link via DMA, and an embedded processor with firmware that implements RS485 SDLC link protocol. Refer to Fig. 1.

The SBX card's memory is divided into two parts: half for buffer data received, and half for buffer data to be transmitted. The buffer size limits the size of a data packet to 100 bytes. It takes thirty microseconds to download transmit data to the hardware. It takes 100 microseconds to transmit the data (the firmware will timeout a transmission after one millisecond). These times do not include interrupt overhead or other software execution times.

As used by KISNet, this network is one link with a master node on one end and a slave node on the other. The primary difference between these node types is that a master node may transmit data through to the slave node without solicitation. A slave node must hold data until the master sends a request, at which time the slave will transmit the data. In other words a slave must be polled for its data.

DIFFERENCES IN TARGET APPLICATION

ALS's network application has a single master "cluster controller" which talks to many slave device controllers. The master is a relatively slow IBM-PC® running Presentation Manager,® while the slaves are speedy Intel 80386 multibus based microprocessors running RMX.® Because the master is almost guaranteed to run slower than the slaves, and the communication direction is mostly in the polled direction from the slaves to the master, slave data can never overrun the master. The master polls the slaves for the data only as fast as the master can process it. The slaves wait for the poll from the master without losing data.

SLC's network application has a single master which talks to a single slave. Both master and slave are speedy Intel 80386 multibus based microprocessors running RMX. Data transmitted is very time-critical to the receiving end, and the fastest way to have data appear at the receiver is to send the data in the unpolled master-to-slave direction. This eliminates the overhead of a software polling cycle, although it does introduce the overhead of interrupt handling. The risk of receiver overruns is also introduced since the data is being transmitted asynchronously.

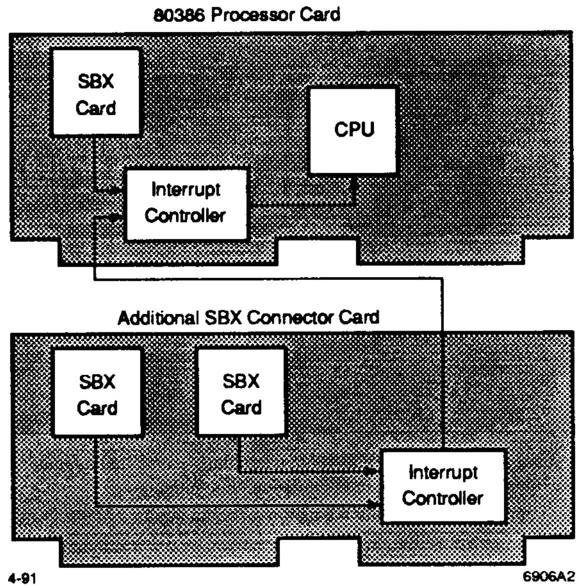

Figure 2. Target architecture.

At SLC, the polled slave-to-master direction is also supported, but is only used for non-time-critical transmissions. Multiple slaves per-master is supported, but not used yet.

Because KISNet is used point-to-point only, more than one KISNet network is required if a micro needs to talk to more than one micro. The KISNet hardware connects to one of SLC's 80386 microprocessors through its SBX connector, and so a micro needs as many SBX connectors as it has micros to talk to. SLC has older revisions of Intel's 80386 CPU card which have only one SBX connector on it; newer revisions of this card, such as are used by the ALS project, have more than one SBX connector. SLC uses a separate card with additional SBX connectors to support multiple KISNet ports. There is a slight performance degradation in the non-CPU-card KISNet ports since they must communicate with the CPU over the multibus backplane instead of directly. Refer to Fig. 2.

The FFP application sometimes must transmit data which is larger than the maximum memory size supported by the KISNet hardware. In this case the data is sent in multiple packets, which imposes a performance hit on the order of a millisecond per extra packet.

PROTOTYPE DEVELOPMENT

The ALS project was of a great deal of assistance during KISNet's design phase. Source code for the ALS high-level drivers and firmware documented the order in which I/O must be issued. ALS diagnostic source code demonstrated polled and unpolled communication. Discussions were held about applying the ALS network hardware to the SLC environment, with issues such as interrupt-driven I/O overhead, necessitated by unpolled communication, and firmware error recovery. Advice was given about parking at LBL.

Documentation was generated on the basis of the information received from ALS, including diagrams of the handshaking between the microprocessor and the firmware, flowcharts of the firmware source code, and block schematics of the gate array.

An IBM-PC platform was set up which held two KISNet ports. This platform was invaluable for quick experimentation, and verified some of the differences between KISnet and the original ALS network. Code developed on this platform was not directly transferable to the target systems on SLC; this was acceptable only because the turnaround of the development cycle on the PC platform was so much quicker than on the target system and because the PC platform had user interface support.

Once the KISNet design was moved from the prototyping platform onto the target system, debugging was needed only in those areas where the target system differed. These areas included interrupt handling, receiver overruns resulting from the asynchronous nature of the transmitting and receiving end, and the effect of higher priority interrupts interfering with both the interrupt-driven input and the atomic output of KISNet packets.

DIFFERENCES IN IMPLEMENTATION

The high-level driver code from the ALS system defined the way in which the hardware could be accessed. There are only a couple of areas in which the KISNet high-level driver code varies from the ALS drivers.

The primary difference is in error detection and recovery. This difference results from SLC responding to errors differently from ALS: the physically larger SLC machine makes it impractical to physically visit the hardware if an error occurs, so the hardware is given every possible chance to recover by itself. Potential infinite loops in the ALS drivers were given escape routes, indicating hardware malfunction and need for retry. When an error occurs, an attempt is made to automatically clear the hardware to determine if the error occurs again with the next operation. An unsolved firmware problem (which generates a firmware error whenever communication switches direction) was worked around in the drivers by anticipating the error, clearing it, and automatically trying again.

Other implementation differences exist in the application interface. Buffers and mailboxes allow the application to use the same interface to send data to a remote destination over KISNet, or to a local destination in another task on the same micro. Support for sequence numbers in the data packets was necessary for multi-packet data.

RESULTS

FFP is now running on the SLC using KISNet. The system works with only minor problems remaining. These are being addressed at the time of this writing.

While working with the firmware developed by ALS, a wish list of desirable additions has been generated. These features would make KISNet a more general-purpose network.

The software which attempts to clear the hardware is inadequate and needs to be replaced by a true hardware reset which may be stimulated by a software command to the firmware.

KISNet needs to query the firmware about the status of the previous write operation submitted to the firmware. Currently the only means by which this status may be obtained is to submit a new operation to the firmware; a workaround may be had by double buffering all write-data so the data from the previous write operation is not lost when the new data is written out and reveals the previous data to have failed.

A firmware retry mechanism is currently available for the write, however none is available for the master's poll operation. Currently, polls are manually retried from the driver.

REFERENCES

[1] S. Magyary et al., "Advanced Light Source Control System," in Proc. 1989 IEEE Particle Accelerator Conf., Floyd Bennett and Joyce Kopta, eds.

[2] T. Himel et al., "General, Database Driven Fast Feedback System for the Stanford Linear Collider," these proceedings.