SLAC - PUB - 5967
October 1992
(I)

OVERVIEW OF THE SCALABLE COHERENT INTERFACE, IEEE STD 1596 (SCI)*

David B. Gustavson, Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94309
David V. James, Apple Computer Inc., One Infinite Loop, Cupertino, CA 95014
Hans A. Wiggers, Hewlett Packard Company, 1501 Page Mill Road, Palo Alto, CA 94304

Abstract

The Scalable Coherent Interface standard defines a new generation of interconnection that spans the full range from supercomputer memory 'bus' to campus-wide network.

SCI provides bus-like services and a shared-memory software model while using an underlying packet protocol on many independent communication links. Initially these links are 1 GByte/s (wires) and 1 GBit/s (fiber), but the protocol scales well to future faster or lower-cost technologies. The interconnect may use switches, meshes, and rings.

The SCI distributed-shared-memory model is simple and versatile, enabling for the first time a smooth integration of highly parallel multiprocessors, workstations, personal computers, I/O, networking and data acquisition.

I. INTRODUCTION

A. *New Approach Needed for Buses*

The Scalable Coherent Interface project arose out of advanced computer-bus work that was being done in the Futurebus+ working group in 1987, when it became clear that practical and fundamental limitations of buses would prevent any bus from meeting the needs of the coming generations of microprocessor-based multiprocessors.

A new approach was needed, one that could provide bus-like services but without the bus limitations. These goals and an outline of the solution were developed by a study group chaired by Paul Sweazey, the Futurebus Cache Coherence Task Group Leader, then at National Semiconductor.

In July 1988 an IEEE working group, P1596, was formed to develop this outline into a new 'bus' standard. The work was technically stable by January 1991 when it was sent out for a vote of the balloting body, and received final approval as a completed ANSI/IEEE standard in March 1992.

The approach SCI adopted was to use a collection of unidirectional point-to-point links instead of a bus. This solves the fundamental bus bottleneck because the multiple links permit multiple concurrent transmissions, and solves the practical signaling problems as well.

Figure 1 shows an SCI subrack, which is based on ANSI/IEEE Std 1301.1, an all-metric standard designed for international acceptance. The narrow SCI links use few connector pins, leaving space free for application-dependent I/O connections or multiple SCI connections. The physical constraints of backplane buses make packages like this very attractive; however, SCI's point-to-point links work well in any geometry, including cable connections, so the motivation for using subracks of this sort is somewhat reduced.

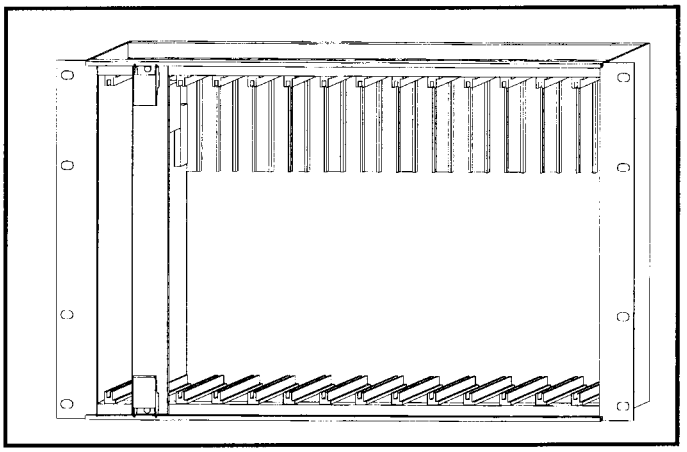

Figure 1: Type 1 Module and a Typical Subrack

B. *New Approaches Bring New Problems*

Several new problems were created by the multiple-link approach, however: the links have to be connected with one another; the familiar 'snooping' methods for maintaining the consistency of multiple cache memories rely on observing all traffic at the bus bottleneck, incompatible with a high performance interconnect; and mutual exclusion mechanisms based on bus locking also fail.

The problem of interconnecting these links was solved by defining their behavior at an interface, so that they can be routed through a switch (for high-end systems) and by supporting their connection as rings (for low-end systems) or meshes of rings (intermediate systems).

Cache coherence was implemented by using a directory instead of snooping. Whenever a cache copies data, it enters itself into a doubly linked list of caches that have copies of that data. Then when one processor modifies the data, it uses the directory to find all the other copies and mark them invalid, so that they will be updated before they are used again. The directory is distributed across the system, not

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

Invited paper presented at the Nuclear Science Symposium, Orlando, Florida, October 25–31, 1992

centralized or concentrated at the memory. The storage required by the pointers of the linked list is provided by those caches that share the data, so there is always precisely the needed amount of storage for the particular sharing situation. The operations that update this list are also distributed, and any memory access touches memory only once, so hot spots are not aggravated by the SCI coherence protocols.

Locks for mutual exclusion have historically relied on locking the single system bus for the duration of operations like Read-Modify-Write, again relying in a fundamental way on the bus bottleneck. In powerful systems with concurrent signal paths, this mechanism must be improved. The solution is to export the atomic lock operation through an arbitrary interconnect to the controller at the destination address. For example, in SCI the Compare&Swap operation sends the new data and the compare value to a remote controller where the operation is executed atomically. The old data is then returned. To the interconnect, this transaction looks like any other, and poses no special problems. However, it does require support from the destination controller.

SCI supports Masked Swap, Compare&Swap, and Fetch&Add lock primitives. These are powerful tools for multiprocessor environments, with the power to arrange consensus among an arbitrary number of processors [1]. They are also just what one needs for handling shared lists efficiently. Shared lists allow multiple processors to enqueue work for a DMA controller, for example, and allow multiple DMA controllers to enqueue interrupts and completion-status reports for a processor to handle.

If lock operations are to be performed on addresses of data that are coherently cached, the locks may be implemented by fetching an exclusive copy of that cache line, performing the lock in software, and then releasing the line for sharing by others. It is important, however, that user code be independent of the cache coherence strategy chosen by the operating system. The processor interface should dynamically select the appropriate way to execute the lock, based, for example, on memory-management page-attribute bits that specify whether the referenced page is to be accessed coherently.

II. Current Status

The Scalable Coherent Interface was approved as IEEE Std 1596-1992 in March, 1992. Several related projects that are extensions or accessories to SCI are still under way.

The standard is now in the process of final editing for publication, which should occur early in 1993. Early users are impatiently awaiting interface chips, which are expected late in 1992. Extensive simulations of SCI have been performed, including a University of Wisconsin simulation of the coherence protocols that runs on a Thinking Machines CM-5 at about half real speed, and a Dolphin SCI simulation of the Verilog model of the interface chip that tests the interaction of five interface ICs in a ring connection.

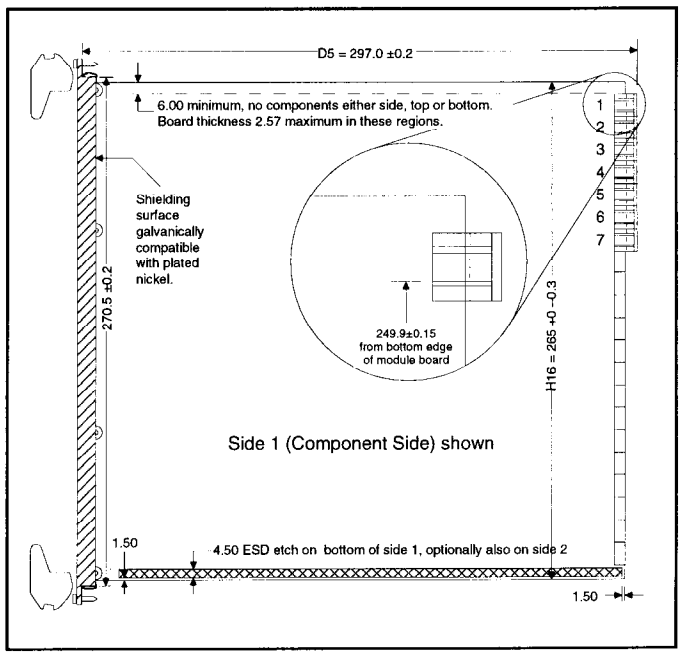

Figure 2: SCI Module Board

An SCI interface chip incorporates transceivers, buffers, and protocol logic. The first will be GaAs by Vitesse, from designs by Dolphin SCI Technology of Oslo, Norway. The input and output links operate at 1 GByte/s each, and the back-end interface uses a conventional TTL 64-bit-wide path at 500 MByte/s. That is a good match to current processors, and is particularly appropriate for use in ring configurations.

These chips can be run at 1/8 speed and used as the protocol engine for driving the parallel/serial/parallel G-Link chips from Hewlett Packard, which are already on the market. They operate at Gbit/s speeds to implement SCI fiber (SCI-FI) or Serial HIPPI links, or in pairs to replace a ribbon cable.

As of this writing, an agreement is in place for a large manufacturer to produce CMOS SCI chips. These will have lower speed, but also lower power and lower cost. They should be well suited for driving SCI fiber links via G-Link or other serializer chips. Details will have to await the December announcement date. These CMOS chips are likely to be faster, less expensive, and easier to use than computer bus interfaces.

There are SCI development projects under way in several computer companies, but only Convex has publicly revealed its plans, using SCI as the interconnect for a super-multiprocessor based on HP's PA-RISC processors. Also, the US Navy is looking favorably on SCI for its High Speed Data Transfer Network, and should make its final decision (among finalists ATM, SCI, and FibreChannel) in December. The Canadian Navy is leading work on SCI Real-Time extensions.

A large SCI prototyping project is under way at CERN, called RD24. This will experiment with using SCI on long fiber links in the LHC environment. A great deal of simulation work has been done in this project. Interfaces between SCI and several other buses are also under development.

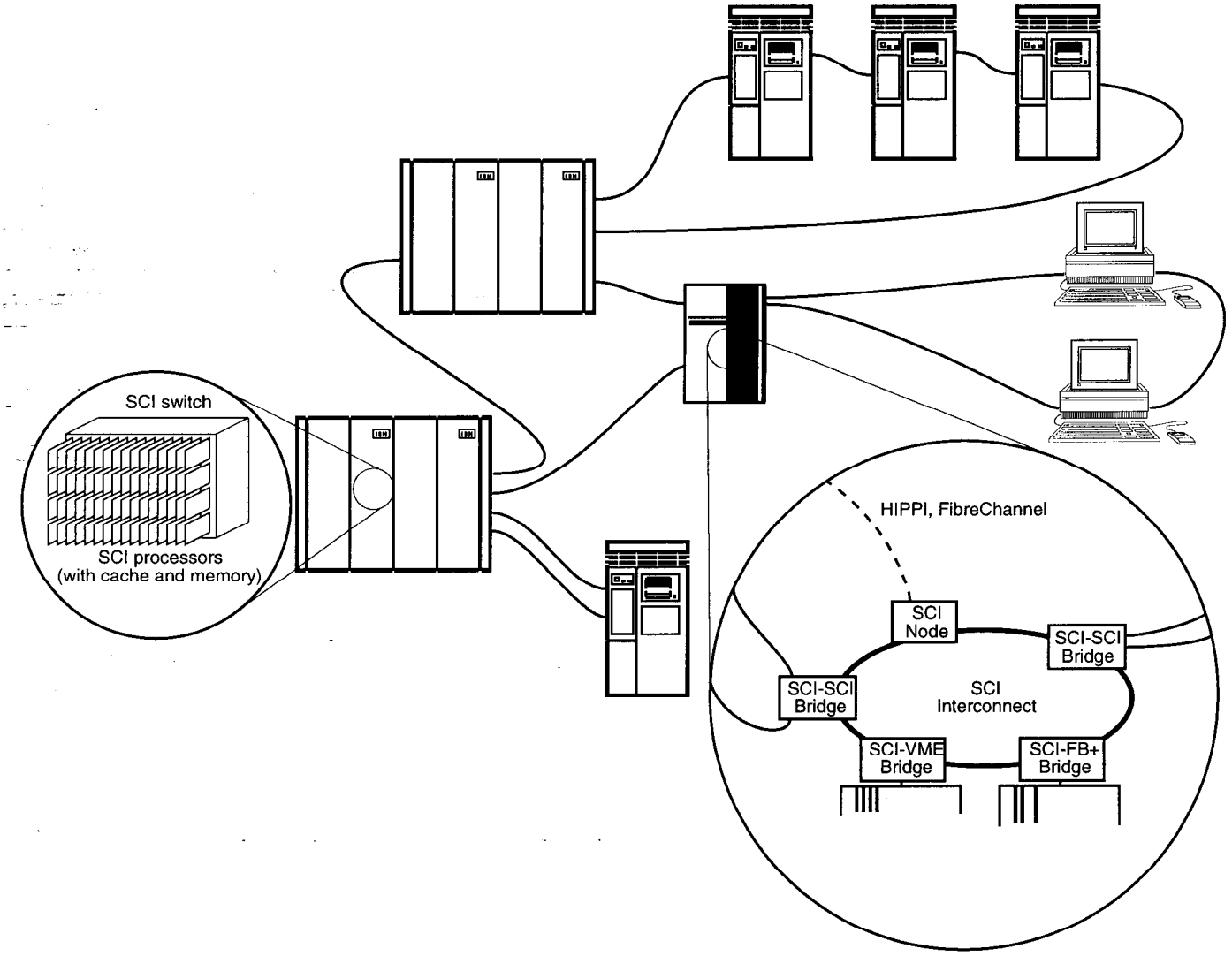

Figure 3: SCI Application Domain: SuperMultiProcessor, I/O, Workstations, LAN

SLAC is starting to put together a collaboration with industry to build a distributed seamless multiprocessor for research purposes, similar to that shown in Figure 3. This would involve a super-multiprocessor and a set of workstations using compatible processors and connected via SCI, as well as a set of workstations using other processors. Such a facility will be useful for learning how best to use shared-memory multiprocessing for the very demanding applications expected in next-generation High-Energy Elementary Particle Physics facilities.

III. Conclusion

SCI is a stable, approved standard with a very large investment in simulations for proving its protocols and interface chip designs. Its interface fits on a single chip, which thus relieves the casual user from having to understand the details. As these chips become available over the next months and the prototype systems are brought online, SCI will become increasingly attractive for a wide variety of uses.

Many applications benefit from SCI's link technology, which frees the user from backplane constraints and allows moving data at high speed over considerable distance. SCI is the first 'bus' to provide such compatible links and support them fully in its protocols. Over the next few years, as SCI volumes increase and interface chip costs drop, it will begin to displace conventional buses in many applications.

For details, or to participate, please contact:

David B. Gustavson
IEEE P1596 (SCI) Chairman
1946 Fallen Leaf Lane
Los Altos, CA 94024-7206 U. S. A.
tel: (415) 961-3539 fax: (415) 961-3530
email: dbg@slac.stanford.edu

IV. References

[1] Herlihy, M. "Wait-Free Synchronization," *ACM Transactions on Programming Languages and Systems 13,* 1 (Jan. '91) 124–149.