A Study of Switch Models for the Scalable Coherent Interface

B. Wu^{1, 2}, A. Bogaerts² and B. Skaali¹

*1) Department of Physics, University of Oslo, 0136 Oslo, Norway
Tel: +47 22 856428, Fax: +47 22 856422, Email: bin.wu@fys.uio.no*

2) ECP-DS, CERN, 1211 Geneva 23, Switzerland

Abstract

The Scalable Coherent Interface (SCI) specifies a topology-independent communication protocol with the possibility of connecting up to 64 K nodes. SCI switches are the key components in building large SCI systems effectively. This paper presents four alternative architectures for a possible SCI switch implementation. Of these four models, two could be seen as crossbar-based switches. The main difference between the two crossbar switch models is the different structure of queues: either a parallel structure, called SwitchLink or serial, called CrossSwitch. The other two models are a ring-based switch model, which is actually connecting four ports into an internal SCI ring and a bus-based switch model which uses an internal bidirectional bus. We will describe each model in detail and compare them by simulation.

Keywords

Scalable Coherent Interface, SCI switch, switch model, simulation, performance

1 THE SCALABLE COHERENT INTERFACE

Connecting several processors, memories and devices is often performed by a single bus for protocol simplicity and low cost. The drawback of the bus structure is bandwidth limitation because of its one-at-a-time feature. Some other multiprocessor systems do provide high performance, but have a very specific design and are often very expensive. With a distributed cache coherence protocol, 1 Gbyte/s per link speed (16 bits data parallel running at 500 MHz), The Scalable Coherence Interface (SCI), IEEE Standard 1596-1992, provides an effective solution to the problem. It uses point-to-point unidirectional links to connect up to 64 K nodes (IEEE Std. 1596 (1992) and Gustavson (1992)).

The basic building block of an SCI network is a ring of two or more nodes connected by unidirectional links. An SCI node would have the basic elements as shown in Figure 1.a. A two-node SCI ring is also shown in the Figure 1.b.

Figure 1 a) A block diagram of an SCI node; b) A two-node SCI ring.

SCI uses a packet-transmission protocol. Most SCI transactions such as read and write are split in two subactions, a request and a response packet with request echo and response echo respectively. The exception is the move transaction that is without a response subaction. A packet contains target address (16 bits of node address that results in a maximum of 64 K nodes in an SCI system, plus 48 bits of internal address), command, source address (16 bits) and control bits followed by 0, 16, 64 or 256 bytes of data and a CRC checksum.

An SCI node is able to transmit packets while concurrently accepting other packets addressed to itself and passing packets addressed to other nodes. Because an input packet might arrive while the node is transmitting an internally generated packet, queue storage (in FIFO order) is provided to hold the packets received while the packet is being sent. New send packets are inserted by a sending node from its output queue, subject to the bandwidth allocation protocol in SCI. The packet is saved until a confirming echo packet is received. The addressed target node strips the send packet and creates an echo packet, which is returned to tell the source node whether the send packet was accepted by target's input queue, or rejected due to queue overflow or other errors. The source either discards the corresponding send packet or retransmits it if the sending was unsuccessful. The sequence of a successful sending is shown in Figure 1.b. Since a node transmits only when its bypass FIFO is empty, the minimum bypass FIFO size is determined by the longest packet that originates from the node. Idle symbols received between packets provide an opportunity to empty the bypass FIFO in preparation for the next transmission.

Input and output queues are needed in order to match node processing rates to the higher link-transfer rate. Since nodes are normally capable of both sending and receiving, request and response subactions are processed through separate queues, which are not drawn in Figure 1, to avoid system deadlocks on these full-duplex nodes. The depth of queues is important, as the deeper the queue, the more packets the node can send and receive concurrently. For the output queues, more outstanding requests could be issued while waiting for the "good" echo of the first send packet. From the input queue side, more packets could be saved for processing if the back-end logic is not fast enough to keep up with SCI speed.

Near the incoming link of the node interface, there is an address decoder that checks the target address of the incoming packets. A decision is then made to accept the packets into the node's input queue or to send it through the bypass FIFO to downstream nodes.

2 SCI SWITCHES

2.1 Why SCI switch

The simplest SCI system is a ring, but an SCI ring structure is sensitive to hardware failures and limited by its peak load and large SCI rings do not scale (Scott, 1992). So different topologies are investigated to interconnect many nodes (Bothner and Hulaas (1991), Johnson and Goodman (1992) and Wu, et al., (1994)). An SCI switch that consists of two or more SCI node-like interfaces (called SCI ports) to different rings, with appropriate routing mechanisms, is a key component in building these topologies. The SCI standard, however, does not directly specify an SCI switch structure. A wide variety of mechanisms are possible.

2.2 SCI switch taxonomies

Generic switch architectures have several classifications based on different attributes (Feng (1981) and Newman (1992)). One classification that also applies to SCI is based on the internal structure of the switch fabric, i.e. time- or space-division methodology. In the first, the use of a physical resource, say conducting medium or memory, is multiplexed among several input-output connections, based on discrete time slots. A bus is an example of a physical conducting medium that can accommodate time-division multiplexing. In a space-division methodology, the switch fabric can support multiple connections at a given moment. The

Figure 2 a, b) Two taxonomies of SCI switch fabric.

connections are based on the availability of nonconflicting physical paths within the fabric. A crossbar is an example. Figure 2.a illustrates such a classification adapted for SCI-based network. Due to the complexity of SCI switches, switches with hundreds of input and output links are not realistic to build with the existing technology. The left part of the dashed line implies what is possible to implement on a single chip (including MCM) while the right reflects the possible implementations on a PCB, in a cluster system or a LAN system.

Time- and space-division multiplexing can be combined. For example, a space-division switching network may be interconnected by several time-division switch elements in a hierarchical fashion.

Another classification of SCI switches in two types is agent and agent-less types. An agent would be a switch that takes control of the operation, returning an echo to the originator and taking charge of forwarding the packet. Agent-less would be a non-intelligent switch, which just forwards packets and echoes which come to it.

This paper will focus on agent type operation and switch elements with a small number of ports. We use 4-switch as an example. The reason is that for a frequency of 500 MHz, the switch should be designed on a single chip. Only so could timing skew be tightly specified because components are inherently well matched in a single chip design.

2.3 General switch model

An SCI 4-switch is by definition a switch implementing SCI protocols on its four ports as shown in Figure 3. The interconnections between the four ports could be a bus, a ring of the time-division class, or a crossbar of the space-division switches.

Figure 3 a) A block diagram of SCI 4-switch; b) A block diagram of an SCI switch port.

An SCI port should have the same functionalities as an SCI node (Figure 1.b), except that the port is simpler in not having the cache coherency logic, but more complicated in back-side connection and routing logic designs (Figure 3.b).

2.4 CrossSwitch model

One of the SCI switch models is named CrossSwitch. The term “Cross” indicates the capability of connecting any input to any output. The block diagram of a CrossSwitch is the same as in Figure 3.a with crossbar interconnect. Not shown in Figure 4 are the bypass FIFOs.

Figure 4 CrossSwitch model data flow with crossbar structure.

It is quite obvious that there will be competition between any two packets that want to go to the same output queue simultaneously, even if the output queue has enough space for both packets. Thus, the loser in the competition will be stored in the input queue. To avoid overflow (which has the consequence that many retries will happen) a deep input queue is recommended.

2.5 SwitchLink model

The SwitchLink model can also be classified as a crossbar-interconnected switch. It solves the contention problem of the CrossSwitch by its internal parallelism. Figure 5 shows the detail of the parallel queue structure of the SwitchLink model.

With sixteen queues in each 4-switch, one can achieve complete parallelism. Packets from different input links go through its private path to its corresponding queue. No packet will be blocked for getting into an output queue even if two packets come from different ports, for example port1 and port3, are routed simultaneously to port2.

The drawback is that parallel queue structure means potential waste of resources if the traffic is not random which unfortunately it is very often the case. On the other hand, the input queue could be omitted.

Figure 5 SwitchLink data flow with parallel output queues structure.

2.6 Bus-based switch model

Putting everything on a single chip with so many interconnections makes the hardware implementation for SwitchLink or CrossSwitch very difficult. The internal-bus scheme is investigated because it is simple and cheap.

A bidirectional bus with an arbitration facility is used to connect SCI ports together (Figure 6). The transfer of SCI packets between different ports is similar to a burst write operation on most buses. In principle, the bus could connect many SCI switch ports and other devices, but with the same argument of using SCI instead of a bus, the “one-at-a-time” feature of the bus will be the great obstacle for achieving high performance. One of the solutions is to increase the bus speed. In Figure 10 we will see simulation results showing the effect of bus speed on the switch performance.

Figure 6 Internal-bus based switch model.

2.7 Ring-based switch model

Neither the crossbar models nor the bus-based model uses point-to-point technique for the interconnections, which implies that it is difficult (costly) to implement hardware, as logic running at 500 MHz has already pushed technology close to its limit. Point-to-point unidirectional communication could greatly reduce the non-ideal-transmission-line problems.

A switch model based on point-to-point connection could be like the one shown in Figure 7. The simplified switch port (s-port) may have the same CSR structure (IEEE Std. 1212, 1991) as the corresponding switch port and it can be implemented without queues. The back-to-back connections are point-to-point transmit channels which could be very fast.

Though SCI rings do not scale indefinitely, they provide better performance than a bus for random traffic pattern (Bogaerts, et al., 1992), as we will see in the simulation results.

Figure 7 A 4-switch with internal ring structure.

3 MODELLING AND SIMULATION

3.1 Simulation tools and simulator

SCILab (Wu and Bogaerts, 1995) is a simulation environment for studying the behavior of SCI based networks. It is aimed at studying the influence of link speed, interface design and SCI protocols under various loading conditions for different topologies. Effects due to the SCI bandwidth allocation, queue acceptance and cache coherency protocols are taken into account. Large SCI systems consisting of rings interconnected by switches can be investigated. Examples of SCI nodes whose behavior can be specified by user definable parameters are also included in the model.

The simulations produce statistics on the SCI link traffic, bandwidth available to SCI nodes such as processors and memories, and the latency of SCI packets traveling through a network from source to target.

We ran each of our simulations for 500 000 ns and each SCI link at the speed of 500 MHz with 16-bit-wide data path. The bypass FIFO and cable delay is 15 ns. Switch routing takes 15 ns and internal transmission of packets from input to output uses 10 ns. Packets are routed in virtual cut-through technique.

3.2 Simulation of a 4-port switch

The performance of a switch design could be tested with the setup shown in Figure 8, with a switch in the middle and four SCI rings connected to it. Each ring has only one processor/memory node that sends and receives packets through the switch. The system is heavily loaded with random traffic.

Figure 8 Switch performance test setup.

When we were simulating such a system, we did not allow a node to send packets to itself since a packet sent to a local node will only go through the bypass FIFO of the switch port. Our purpose is to saturate the switch and see the performance. Figure 9 gives a good indication of the performance (Throughput, Latency) of a single switch chip with the architecture of SwitchLink (SL), CrossSwitch (CS), Bus-based and Ring-based switch, respectively.

Simulations were performed for both one-packet-deep and three-packet-deep queues. It should be pointed out that when we say one-packet-deep queue, we mean one-packet-deep output queue and one-packet-deep input queue, except SwitchLink which uses four one-packet-deep output queues (without input queues, and thus use double the queue size in total).

Considering the effect of queue size on system performance, it is not so surprising to see that SwitchLink exhibits the highest throughput and the lowest latency. A Bus-based switch is limited by the bus speed of 1 Gbyte/s no matter how deep the queues are. The performance of

Figure 9 Performance evaluation for SL, CS, Bus- and Ring-based switches.

a Ring-based switch is not good, also due to the limited performance of a single ring. We should emphasize that, among these four models, CrossSwitch is the model that could provide relatively high performance with relatively low cost in the sense of queue size, if we compare with the normalized SwitchLink performance in Figure 9.

The performance of the single switch system can to some extent reflect the performance of a large SCI system, and this is proved by simulations (Wu and Bogaerts, 1993).

3.3 Bus speed

Because a bus is simple and easy to build in hardware but limited by its “one-at-a-time” feature, it is interesting to see whether increased bus speed could help to solve the bottleneck. Figure 10 presents the simulation results for different bus speeds.

Figure 10 System's total throughput versus internal-bus speed.

It is observed that increasing bus speed does not provide linear increase in system throughput. This is due to the transmission overhead, as well as the obligatory use of store-and-forward switching technique when the bus speed is faster than SCI speed.

4 SUMMARY AND CONCLUSIONS

While SCI does have characteristics that facilitate interconnection between SCI rings, inter-networking and switch architecture is not part of the SCI standard. This leaves a high degree of flexibility to the implementor. In this paper we presented and discussed four SCI switch models.

The SCI switch models have been strongly influenced by features of the SCI standard and SCI node chips to achieve high bandwidth and low latency with modest complexity. For comparison, we list some of the important aspects in Table 1.

Table 1 Evaluation of SwitchLink, CrossSwitch, Bus-based and Ring-based switch

Switch Model	SL	CS	Bus ^a	Ring
Complexity	o ^b	o	+	o
Throughput	+	+	-	-
Latency	+	+	o	-
Suitability for burst traffic	-	o	+	o
Suitability for random traffic	+	+	o	o
Performance Scalability vs. # of ports	+	+	-	-
# queues for a 4-switch (n queue) ^c	4x4xn	4x2xn	4x2xn	4x2xn ^d

a. Internal-bus runs at 1 Gbyte/s

b. +: good; o: average; -: poor.

c. kxm_n where k is the number of ports, m is the number of queues, n is the depth of queues in unit of SCI packets.

d. s-port does not have queues.

Though back-plane buses have severe limitations, we still find them quite attractive because of their simplicity and low cost when the number of connected devices is not large. Higher bus speed could also help in some sense to increase performance. Between cost and performance, there are trade-offs to make.

Though still at its early stage, a wide range of applications are possible for SCI now, such as SCI/Sbus, SCI/VME, SCI/PCI and SCI/ATM interface, etc. SCI rings of several nodes have been up and running for almost two years. SCI switches are also catching up, like the UNISYS 2x2 switch (Buggert, et al., 1994), the Dolphin CMOS switch and the TOPSCI 4-switch (Wu, 1994).

Switch designs should capitalize on several of the most important features of SCI, namely scalability, simplicity and high performance. Scalability allows cascading of elementary switches into a larger composite switch as well as the growth of a system which needs more SCI ports. Simplicity eases the design, implementation, and operation of systems requiring high throughput and low latency.

Even being quite familiar with the requirements for SCI switches and their behaviors, it is still a challenge to design switches operating at 1 Gbyte per second per link in GaAs technology, or 200 Mbyte per second in CMOS technology.

5 ACKNOWLEDGMENT

We are grateful to colleagues in RD24 group at CERN and EURIKA TOPSCI project members, for the ideas that assisted this work. SwitchLink idea was initiated by Roberto Divia at CERN and later improved during brain-storm discussions within RD24 group at CERN.

Bin Wu is supported by the Norwegian Research Council. This work is partly supported by Thomson TCS, France.

6 REFERENCES

- Bogaerts, A., Divia, R., Muller, H. and Renardy, J. (1992) SCI based Data Acquisition Architectures, *IEEE Transactions on Nuclear Science*, Vol. 39, No. 2, Apr. 1992
- Bothner, J. and Hulaas T. (1991) Various interconnects for SCI-based systems, *Proceedings of Open Bus Systems '91*, Paris, 1991
- Buggert, J., Desai, V., Herzberg, L. and Kibria, K. (1994) The Unisys Datapump and Switch, *Proceedings the 1st International Workshop on SCI-based High Performance Low-Cost Computing*, Santa Clara, CA., Aug. 1994
- Feng, T. (1981) A Survey of Interconnection Networks, *IEEE Computer*, December 1981, pp 12-27
- IEEE Std 1212 (1991) *Control and Status Register (CSR) Architecture for Microcomputer Buses*, IEEE Service Center
- IEEE Std. 1596 (1992) *The Scalable Coherent Interface*, IEEE Service Center
- Johnson, R. and Goodman, J. (1992) Synthesizing General Topologies from Rings, *Proceedings of ICCP*, Aug. 1992
- Gustavson, D.B. (1992) The Scalable Coherent Interface and related Standards Projects, *IEEE Micro*, Feb. 1992, pp. 10-22
- Newman, P. (1992) ATM Technology for Corporate Networks, *IEEE Comm. Magazine*, Vol. 30, N. 4, Apr. 1992, pp. 90-101
- Scott, S., Goodman, J., Vernon, M. (1992) Performance of the SCI Ring, *Proceedings IEEE ISCA 92*, Queensland, May 1992
- Wu, B. and Bogaerts, A. (1993) Several Details of SCI Switch Models, *RD24 note*, CERN, Nov. 1993
- Wu, B., Bogaerts, A., Kristiansen, E., Muller, H., Perea, E. and Skaali, B. (1994) Applications of the Scalable Coherent Interface in Multistage Networks, *Proceedings IEEE TENCON'94, Frontiers of Computer Technology*, Singapore, Aug. 22-26, 1994
- Wu, B. (1994) SCI Switches, *Proceedings Int'l Data Acquisition Conference on Event Building and Event Data Readout in Medium and High Energy Physics Experiments*, Batavia, IL. Oct 26-28, 1994
- Wu, B. and Bogaerts, A. (1995) SCILab-A Simulation Environment for the Scalable Coherent Interface, *Proceedings IEEE MASCOTS'95*, Durham, NC., Jan. 16-18 1995

7 BIOGRAPHY

Bin Wu is a Ph.D candidate at the University of Oslo. He received a research fellowship from the Norwegian Research Council in 1993 and he is a Scientific Associate at CERN, Geneva, Switzerland. His research interests include high-speed networks, switch technologies, VLSI design, network modelling and simulation. Wu received a BSc and a MSc degree from Department of Physics, University of Oslo, both in Electronic Engineering, in 1991 and 1992 respectively.

Andre Bogaerts graduated in 1970 in Physics and Mathematics at the University of Utrecht, the Netherlands. From 1970-1974 he was employed by the Dutch Organization for Fundamental Research (FOM) where he joined an international team for a High Energy Physics experiment at the Intersecting Storage Rings (ISR) at CERN, Geneva, Switzerland. Since 1974 he has been employed by CERN. His main interest is in Data Acquisition Systems for High Energy Physics Experiments. He has been involved in the standardization of the Scalable Coherent Interface (SCI) and, since 1992, he has been leading a team to investigate the use of SCI for future experiments at CERN's Large Hadron Collider (LHC).

Bernhard Skaali graduated as Master of Science in Physics in 1966 and Doctor of Philosophy in Nuclear physics in 1976, from the University of Oslo. Research assistant 1966-72, Senior lecturer 1973-82, and Professor from 1983 at the Department of Physics, University of Oslo. Current main research areas: instrumentation and DAQ for high energy physics. Experiments: LEP-DELPHI-CERN from 1985, LHC-ALICE from 1994. Stays abroad: Tandem Acc. Laboratory, Niels Bohr Institute, Copenhagen, 1969-70 (17 months), Scientific Associate, CERN SPS div, 1976-77 (12 months), CERN DD div 1980-81 (6 months), CERN PPE div 1988-89 (17 months).