have very high latencies (>10 µsec) and relatively low throughput (155 or 622 Mbits/sec). In addition, ATM systems require more conventional protocol software that cannot rely on the switch to drop or corrupt messages.

At the other end of the spectrum, S-Connect can compare favorably to interconnect systems designed for massively parallel multiprocessors. Dual-TIC S-Connect networks offer about the same bisection bandwidth as the Cray T3D MPP, however at about 5-10x higher HW latencies. However, one router in T3D needs 3 ECL-gate-arrays and 192 wires to connect to other nodes, all of which are high quality, matched transmission lines. In the largest, dual-TIC configuration, one S-Connect nodes needs 2 inexpensive, low-power CMOS chips and 16 wires. While the wires in S-Connect also need to be high quality transmission lines, they do not need to have the same length. Similarly, S-Connect systems compare favorably in terms of cost, performance and functionality to the Intel Paragon interconnect, the Meiko CS-2, N-Cube systems, and the IBM SP-2.

7 Conclusion

The S-Connect system is a high-performance distributed switch which can be used as the interconnect fabric for a local area, distributed shared memory multiprocessor as well as for conventional networks-of-workstations, where high throughput and low latencies are needed. The TIC achieves a high level of integration by using multiple integrated high-speed serial links that can interface directly with high speed (>1Gb/s) optical fiber systems. The adaptive routing algorithm developed for S-Connect does not depend on any particular topology and achieves near optimal throughput even in the case of irregular, fine grained loads. Virtual cut-through is used to minimize latency and was shown to be superior to wormhole routing at the expense of higher complexity. The TIC provides support for time critical traffic through the use of four priority levels and by providing global synchronization. Because of the use of new technologies and a high level of integration, S-Connect systems are 10x more cost effective than other interconnect systems with supercomputer class throughput.

8 Acknowledgments

The TIC team would like to express their gratitude for the contributions of Prof. David Dill and his team, who help with the formal verification of the TIC logic, Dr. Kenneth McMillan, who applied his SMV system to verify parts of the router core and Han Yang, who assisted in the development of the channel protocol. Special thanks belong to Wayne Rosing, Dr. Doug Boyle and Dr. Jeff Rulifson, who supported this work within Sun.

The high speed serial link circuit were developed by Prof. D.K. Jeong and his student at the National University in Seoul, South Korea, and implemented by LSI Logic corporation.

S-Connect is a research project of the technology development organization of Sun Microsystems Computer Corporation and is currently not a commercially available product.

References

[1] The First Networks of Workstations Workshop, October 1994, San Jose, California.

[2] Nowatzyk, A., Aybay, G., Browne, M, Kelly, E., Parkin, M., Radke, W., Vishin, S., *S3.mp: Current Status and Future Directions*, Shared Memory Multiprocessor Workshop, International Symposium on Computer Architecture, Chicago, IL, May 1994.

[3] Nowatzyk, A., Parkin, M., *The S3.mp Interconnect System*, Hot Interconnect Symposium, Stanford, CA, August 1993.

[4] Rambus System Specification, 1993, Rambus Inc., Mountain View, CA

[5] James, D. V., Laundrie, A. T., Gjessing, S., Sohi, G. S., *Scalable Coherent Interface*, IEEE Computer #6, Vol. 23, pp74-77.

[6] Futurebus+, P896.2: *Physical Layer and Profile Specification*, June 1991, IEEE Computer Society

[7] Stevens, K.S., *The Communication Framework for a Distributed Ensemble Architecture*, AI Technical Report 47, SRI, February 1986

[8] Dally, W. J., *A VLSI Architecture for Concurrent Data Structures*, PhD Thesis, California Institute of Technology, 1986

[9] Kim, J. H., Liu, Z., Chien, A. A., *Compressionless Routing: A Framework for Adaptive and Fault-Tolerant Routing*, International Symposium on Computer Architecture, Chicago, IL, May 1994.

[10] Kermani, P., Kleinrock, L., *Virtual Cut-Through: A New Computer Communication Switching Technique*, Computer Networks, 3:267-286, 1979.

[11] Lenoski, D. *The Design and Analysis of DASH: A Scalable Directory-Based Multiprocessor.* PhD Dissertation, Stanford University, December 1991.

[12] Agarwal, A.,Kubiatowicz J., Kranz, D., Lim, B., Yeung, D., D'Souza, G., Parkin, M. *Sparcle: An Evolutionary Processor Design for Large-Scale Multiprocessors*. IEEE Micro, June 1993, pages 48-61.

[13] Nowatzyk, A. *Communications Architecture for Multiprocessor Networks*. PhD Dissertation, Carnegie Mellon University, December 1989.

[14] *Myrinet Product Information*, Myricom Inc, Arcadia, California.

[15] Guenter, K. D., *Prevention of Deadlocks in Packet-Switched Data Transport Systems*, IEEE Transactions on Communications C-29 (4):512, April 1981

delay. Because of this, messages are delayed so the header arrives at the specified time slot for the corresponding inbound channel. This leads to the variance in node traversal latencies.

The α-boards have a demonstrated DMA speed of >80 Mbytes/sec. It turns out that they can receive data about 30% faster than they can send data, which is due to the larger number of cycles required on the Mbus to perform a read operation. It requires about 4 programmed I/O references (load/store from/to status registers) to the α-board to initiate a DMA operation because the α-board does not chain DMA descriptors, which will be added later.

4.2 Simulation results

The TIC bisection bandwidth and latency for 2D meshes, 4D meshes and the average of random interconnect topologies are given in Figure 18. The shown bandwidth includes only the data portion of single packets that are addressed to random destinations and does not include bandwidth used for flow control and CRC. To present a conservative estimate, the bandwidth is derated to 80%.

The reference points for other interconnect systems in Figure 18 use 100% of the raw physical bandwidth with no deductions for any communication overhead, congestion or router inefficiencies. They represent the upper performance limits which cannot be reached in real applications.

5 S-Connect Applications

S-Connect is the switching fabric for the S3.mp distributed, shared memory multiprocessor [2]. Several of the design decisions were motivated by this primary application, for example the message sizes are optimized for carrying cache lines and coherency control messages. Strict priorities were added to avoid deadlock of the cache coherency protocol. The S3.mp architecture, which is beyond the scope of this paper, delivers more than 100 Mbyte/sec user program accessible bandwidth at latencies of about 1 µsec. Due to the nature of the S-Connect system, the S3.mp multiprocessor can be assembled out of cost-effective workstations that are spatially distributed. The very same architecture also can be used to large multiprocessors with communication performance that is competitive with the best commercially available MPP systems, while costing about 10 times less per node.

The capabilities of the TIC chip are also being demonstrated in form of a Mbus DMA adapter card that uses Altera FPGA's and one TIC chip. This card allows cache coherent data transfers between workstations and high speed I/O devices (HDTV video sources and high resolution frame buffers). This card can send data at about 90 Mbytes/sec while concurrently receiving data at 90 Mbytes/sec, a rate that is limited by the Mbus and the memory subsystem of the workstation, rather than by the TIC. This demo card is essentially equivalent to The IBM SP-2 multicomputer, however it does not need any external switch and achieves higher throughput at a fraction of the cost.

A more specialized TIC application under development is the use of S-Connect fabrics for ATM switches and scalable TCP/IP routers. In this role, the TIC is integrated into subscriber cards where several ATM channels a connected to one TIC and some amount of buffer memory. A specialized protocol is run in firmware to implement the switching functions. The scalability of TIC based interconnects allows this ATM switch to be extended in small increments without saturating any busses that are currently used inside small switches.

Future versions of the TIC will not be an independent ASIC, rather they will become a specialized macrocell that is incorporated into other chips that require a high performance, scalable interconnect system.

6 Related Work and Discussion

Among the many high performance networks that are currently being proposed and implemented, Myrinet [14] is the most similar to S-Connect. Myrinet evolved from the family of Torus routing chips developed at Caltech [8] and inherited many of their characteristics, such as worm-hole routing, self-timed circuits, and a preference for the mesh topology. S-Connect achieves higher performance than Myrinet system, even if the channel speed were set to the same bandwidth. This due to three main reasons:

- S-Connect prefers random topologies, which have diameters that scale with the logarithm of the number of nodes. This means that messages have to traverse fewer nodes and occupies less channel time.
- S-Connect does not use wormhole routing, which more than doubles the saturation throughput. Unlike Myrinet, which relies on source directed routing, the adaptive routing algorithm in S-Connect can base routing decisions dynamically on the local traffic situation.
- S-Connect operates synchronously on fixed length messages. This means that the available fiber capacity is optimally scheduled without any gaps between messages. Asynchronous systems avoid the synchronization necessary to inject messages and might achieve lower latency in the unloaded case, but practically all client machines operate synchronously and require synchronization circuitry with associated delays to interface to an asynchronous component.

Other differences include the lack of priorities in Myrinet as well as the performance monitoring and diagnostic support. It is fair to say that S-Connect is a smarter router that can deal with less sophisticated interfaces, which is reflected in the fact that the TIC chip used more gates.

ATM networks originated in the telecommunication industry and were designed to address the more general problem of constructing large, wide area networks. As a consequence, it is more suited for WAN/LAN applications, which can tolerate the higher latency, the lack of HW flow control, and the message size/structure which does not match fine-gained, irregular traffic. Current ATM switches

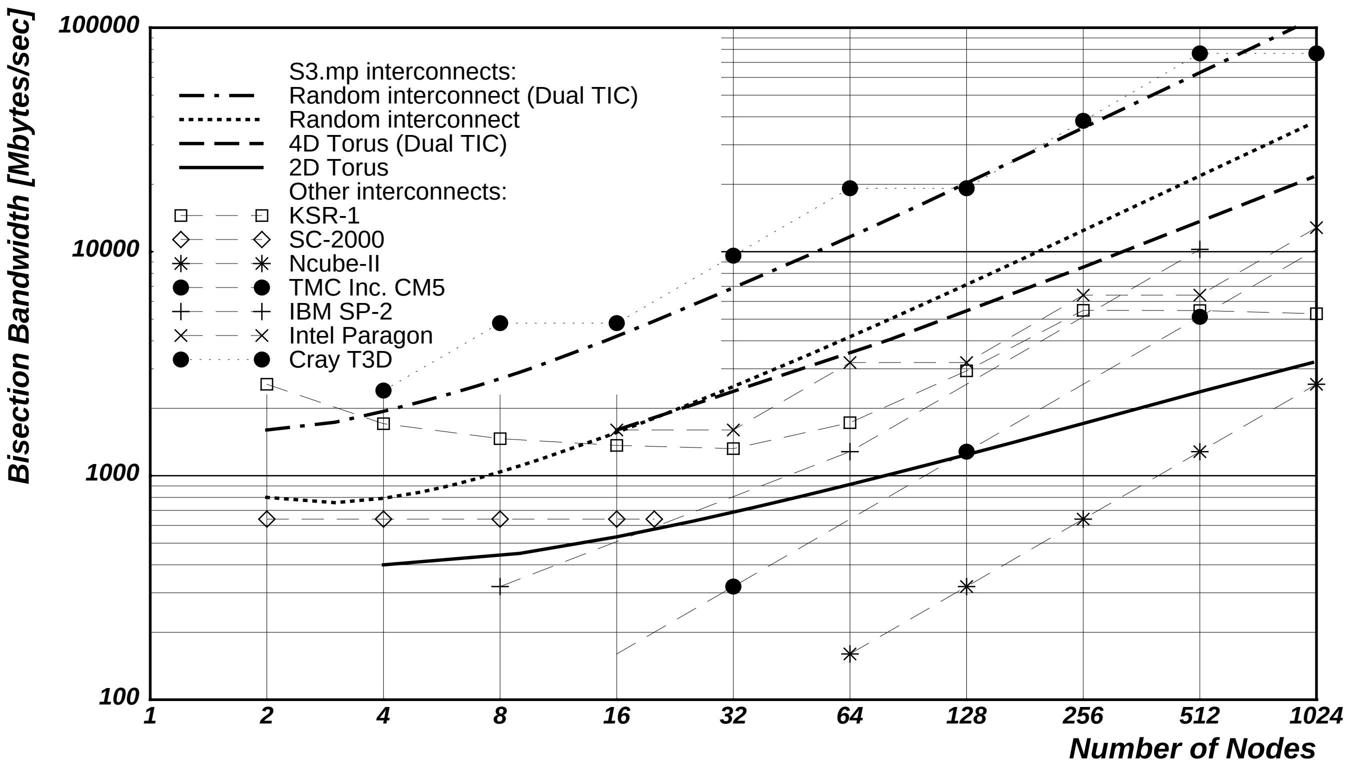
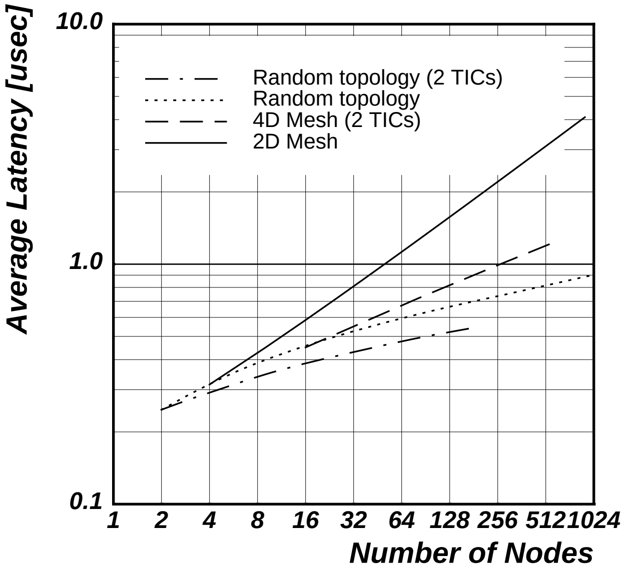


FIGURE 18 : Estimated TIC Performance

carried in the DMA packets so that out-of order delivery is supported. The initiator of a DMA transfer needs to ask the destination node for a valid buffer address prior to initiating the transfer. Because the Mbus in the SS-10 has a usable memory bandwidth of about 100 Mbytes/sec, the α-boards use only one of the parallel ports into the TIC chip (the other port is left unconnected). Availability of VXCO's limits the speed of the α-board to 51.840 Mhz (custom VXCO's have long lead times).

These limitations of the α-board result from to the use of off-the-shelf components and from a fairly simple design in order minimize the design and implementation time, and to allow room for experimentation. The next generation of TIC boards will have better functionality and will use higher integration so that the interface board will occupy a single width Mbus module (currently, the α-board is about twice as large and uses only one side of the board to facilitate easy probing and instrumentation).

The global clock synchronization resulted in a measured clock jitter between any two nodes that is about 100 ps, which is much less than expected. It is also independent of the topology.

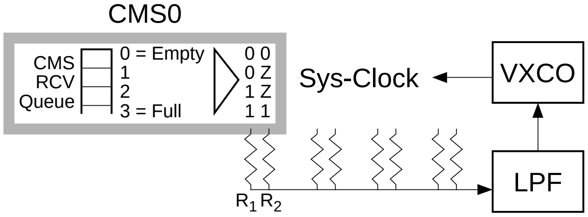


Single packets are sent by successively writing to two 64 bit registers. The measured latency from the time of the second store to the time that the destination is ready to supply the message, is 1.7 µs, which is dominated by the FIFO and interface logic of the α-board. Operating at 51.840 Mhz and using 2 m of cables between nodes, each node traversal adds about 310 ns, which is dominated by the serialization overhead in the media interface. The actual latency depends on the phase of the routing cycles of adjacent nodes and on the length of the channel and ranges from 180 ns to 330 ns. While each TIC will run at exactly the same clock speed, the phase relation of adjacent nodes depends on the media errors, the determination that a transfer is completed depends on the number of errors that occurred. Once sent, a message and its associated buffer are locked by the CM until it can determine that the transfer succeeded or failed. Whenever a packet transmission fails, the packet is returned to the router and a new routing decision is made. Each of these failures causes the age field of the packet to be incremented, so that packets cannot perpetually retry.

3.3 Clock distribution and synchronization

S-Connect switching fabrics operate isochronously, which is essential for the efficient operation of the TIC (simpler channel protocols, less overhead, no need for low level flow control, etc.). Clock distribution uses the method described in Section 2.5. The critical component is the phase/frequency comparators for each channel. The TIC uses the receive fifo depth in each CMS as a phase/frequency comparator. The depth increases if the local clock frequency is too slow and it decreases if the local oscillator is too fast.

FIGURE 17 : Distributed Phase Lock Loop



The TIC averages the fifo depth of all active channels by using a simple 2 bit digital to analog converter, that consists of two external resistors, R_1 and R_2. Given that the fifo under-/overflow logic does not wrap around, this realizes a P/F comparator with a range of 4 rad. By choosing $R_1 > R_2$, the transfer function is distorted such that the loop gain increases with the deviation from the fifo half-full point, which results in a lock frequency that is moved towards the mean of the center frequencies of all nodes. This feature improves stability because it moves the operating point away from the limits of the VXCO tuning range.

3.4 TIC control facilities

All functions of the TIC are controlled through special control messages that are interpreted by the TIC. Control messages can be sent to a TIC through any channel, so that the TIC can be remotely configured. TIC control functions include:

- Read and write access to the routing table. This is used to download and verify routing table entries.
- Forwarding of control packets over a specific channel. This is used to bootstrap the system without a valid routing table. Essentially, this allows source directed routing to map the network topology.
- Access to error and status registers. For each CM/CMS a number of error conditions are recorded. Channels can be enabled or disabled explicitly.
- PLL diagnostics. By turning the outbound data stream off, the distributed PLL is disabled and the rate of fifo under- or overflows can be used to compare the free running clock frequencies.
- Performance monitoring.A programmable counter can be connected to 32 different event sources. For example, this facility can be used to measure the traffic over a particular channel.
- Time keeping: The TIC includes a 16 bit timer that can be read remotely. The times can also be used to time stamp control packets to compare the timer of two adjacent nodes. The timer can be adjusted by adding to it atomically. This is used by S3.mp nodes to provide a globally synchronized notion of time.

4 S-Connect Performance

The first lot of TIC chips was received in November ‘94 and was successfully tested. The test setup was limited to 50 Mhz. The first TIC-based interface boards (α-run) became operational in February ‘95. As of March ‘95, a TIC interconnected cluster of 4 SparcStations-10’s is used to evaluate the performance and to investigate a number of implementation issues (cables, electro-magnetic emissions, compatibility with common fiber optic tranceivers, packaging issues, cooling, etc). Performance estimates for larger configurations are based on simulations.

4.1 α-boards

A small number of α-boards were assembled to gain experience that will lead to the design of a TIC based network interface board which will be made available to research collaborations. The α-board uses standard components (FPGAs and FIFO chips) to implement a DMA engine that is attached to the Mbus of a SS-10 or SS-20 workstation. It allows to send and receive control messages under programmed I/O and it has the ability to copy a region of 32 to 4096 bytes to a remote node in a single DMA operations. DMA transfers are currently not allowed to cross page boundaries and must transfer an integral number of cache lines (32 bytes). The DMA transfer uses cache coherent Mbus transactions to ensure that the most recent data is fetched and that stale data is invalidated during a transfer. The α-board uses a FIFO to attached to the TIC so that the Mbus can operate asynchronously to the TIC, which is globally synchronized. Unsolicited packet arrival and DMA completion can cause interrupts. The α-board has provisions for a counter array that is indexed with the source address of a DMA-packet and which can be used to signal DMA completion at the destination site. The destination address for a DMA transfer is generated by the sender and is

[1] The CM protocol is actually more complicated because the TIC supports quad-messages, which are trains of 4 messages that traverse the interconnect as one unit. Since the entire quad messages must be transferred atomically, adjacent CM slots are coupled in this case.

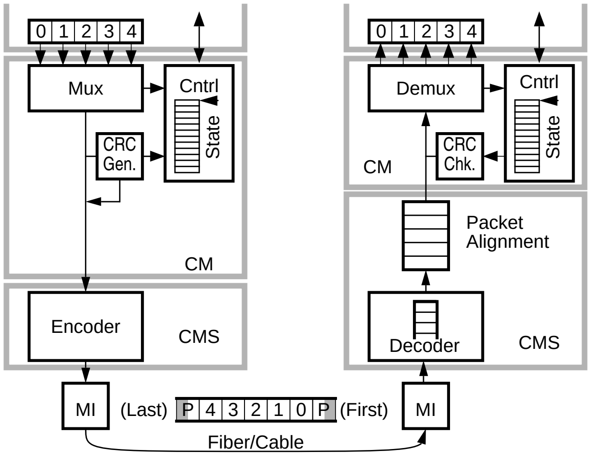
tained in hardware. Since a virtual cut-through algorithm is used, enqueue and dequeue operations can overlap, so that sending of a packet may commence before it is entirely received. Because of this overlap, a CRC error on an incoming packet causes an abort bit to be set in the corresponding outgoing packet so the receiving logic at the next node is able to discard the packet. This bypass logic allows messages to emerge from the TIC chip before they are completely received, resulting in routing latencies of less than 6 cycles.

All of these operations (receiving, routing, enqueueing/dequeueing) take place concurrently at a rate that matches the total packet throughput rate. This performance can be achieved because all packets are of equal size and arrive at precisely scheduled time slots.

3.2 Channel protocol

Each channel is composed of two components: the channel module (CM) that executes the packet level protocol and a channel maintenance subsystem (CMS) that includes all the low level functions associated with bit serial data transmission (Figure 14).

FIGURE 14 : Channel Maintenance Subsystem



Attached to the CMS is the media interface, which include the high speed serializer and deserializer, as well as the low voltage swing line drivers, impedance match and termination circuitry, etc. The complete MI occupies about 1.6 mm^2 per channel plus a central PLL circuit that is shared by all 4 channels. The power consumption is bout 400 mW / channel, which compares favorably to I/O drivers that support 220 Mbyte/sec.

The CM operates on the premise that it is connected to a pair of 16 bit wide data paths that synchronously accept and supply messages. Every routing cycle starts with the first part of the message header. The CM further relies on the fact that the channel has a fixed delay, which is an integral multiple of the routing cycle. Hence the channel appears as if it were a conveyor belt with a fixed and known number of slots that can hold one message (Figure 16).

FIGURE 15 : Eye Pattern of Serial I/O Signal

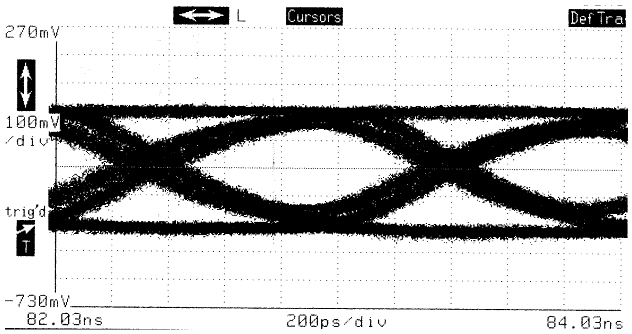
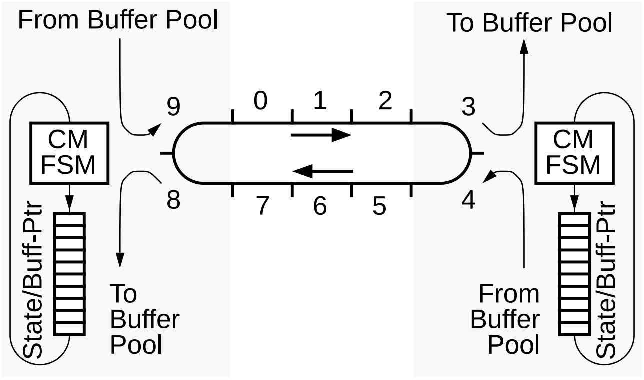


FIGURE 16 : TIC Channel Protocol



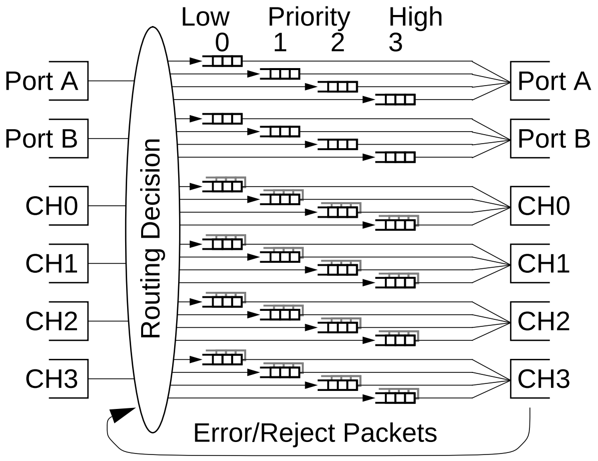
Message transmission is not assumed to be error free, rather the CM verifies packet integrity. Packets which are rejected by a channel module due to lack of buffers or a transmission error are placed in a reject queue by the sender. Rejected messages go through a routing cycle to determine a new transmit channel and are then inserted at the head of the corresponding outbound queue. Receive errors also corrupt the piggyback acknowledgment which causes the original packet sent in the corresponding time slot to be retransmitted. In the event of a duplicate transmission the receiver will discard the packet, because the sequence id bit of the incoming message will not match the expected sequence id bit. Essentially, a 1 bit windowing protocol is used for each slot. The entire protocol overhead is 6 bits/message, which includes the piggyback acknowledgment and a late-abort feature, which allows the annihilation of an outgoing message in case the received message was corrupted so that the outbound message is valid (= the ack-bit for the reverse channel is good) but is discarded upon reception.

While the number of slots depends on the physical length of the channel, the actual CM protocol is oblivious to the number of packets in transit. For each slot, the CM maintains an independent set of state variables in the form of a shift register that matches the channel delay. The CMS synchronizes these state variables, so that the CM operates as a collection of independent protocol engines, each dealing with a unity delay channel.[1]

The CM protocol has the property that it neither drops nor duplicates messages. In the presence of transmission each combination of 4 priority levels and 2 virtual channels. The transmitters serve their queues in strict priority order: as long as there is pending traffic in a higher priority queue, no lower priority queue will be served. The parallel ports do not support virtual channels, but are otherwise functionally equivalent. The virtual channels are served fairly. However if a virtual channel has no pending traffic, the entire channel capacity is available to the other virtual channel.

The routing decision is based on the packet priority, the amount of traffic pending for each viable path and the packet age. A new routing decision is made for packets that were rejected due to transmission errors or due to congestion at the next TIC. Requeued traffic has priority over newly inserted traffic. This routing algorithm differs from the one described in Figure 4 through the omission of step 1: the transmitter work assignment proceeds in a fixed order, which is hardwired into the pipeline. This results in slightly lower efficiency, however this effect is countered by having a much larger buffer pool: the TIC chip has 64 buffers while the simulations were based on only 4 buffers.

FIGURE 11 : Logical TIC Structure



The TIC recognizes 4 levels of priority such that higher priority traffic can't be blocked by congestion at a lower priority level. To guarantee that low priority traffic can't block higher levels, some buffers are reserved. This means that not all 64 buffers can be used for low priority messages.

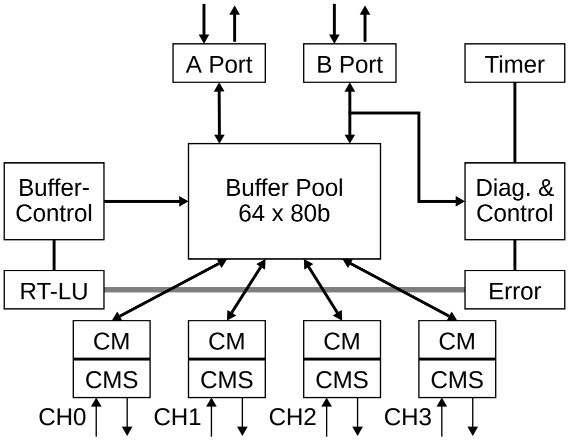
3.1 TIC details

At the core of the TIC is a common buffer pool that is shared by all 6 ports (Figure 12). Rather than implementing 40 independent queues, these queues use linked lists that are maintained in the buffer control logic. By sharing buffers, the required amount of on chip memory is greatly reduced.

Each of the 4 serial links has a logically independent transmitter and receiver. Receivers accept messages, verify their integrity (CRC codes are used to ensure a high degree of robustness), and deposit them into the buffer area. Concurrently, the transmitters retrieve messages from the buffers and send them to other nodes.

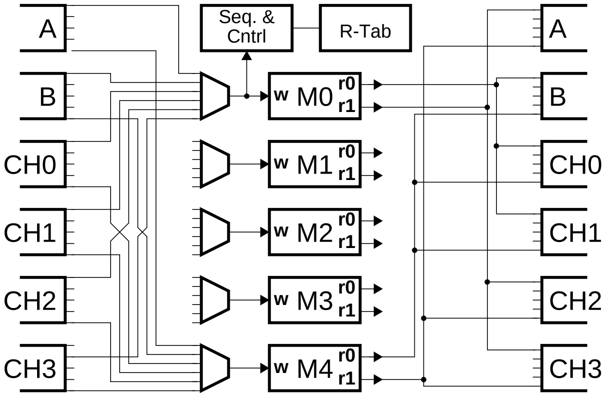
The location in the register file to store the incoming message is computed in advance based on a bit vector representing the free buffers. Even though the buffer pool is shared by all queues, additional buffer reservation logic guarantees that for each priority level and for each virtual channel a minimal number of buffers are available so that traffic from different classes cannot cause resource dependencies that lead to deadlocks.

FIGURE 12 : Simplified TIC Block Diagram



The buffer pool uses 5 independent memory banks that are used in different stages of the pipeline (Figure 13). At the write port of the first memory bank, the newly received message data is inspected by the router control logic. The first 16-bit piece of data contains the destination address, priority, and type. Concurrently with the write operation, the destination address is used to perform a routing table lookup. The routing table entry consists of an 8 bit virtual channel mask, that determines which virtual channel may be used or which parallel port should receive the packet.

FIGURE 13 : TIC Buffer Pool Structure



The routing table lookup, the packet priority, the queue sizes and the source virtual channel determine which queue should receive a packet. In the case of rejected messages (due to congestion or due to transmission problems), preference is given to a channel other than the one tried in the last attempt. The queue structure uses linked lists that are main-transmission error about once in 10^{15} years of continuous operation. Detected transmission errors are dealt with through retransmission, using a protocol that has been formally verified to never duplicate or drop messages.

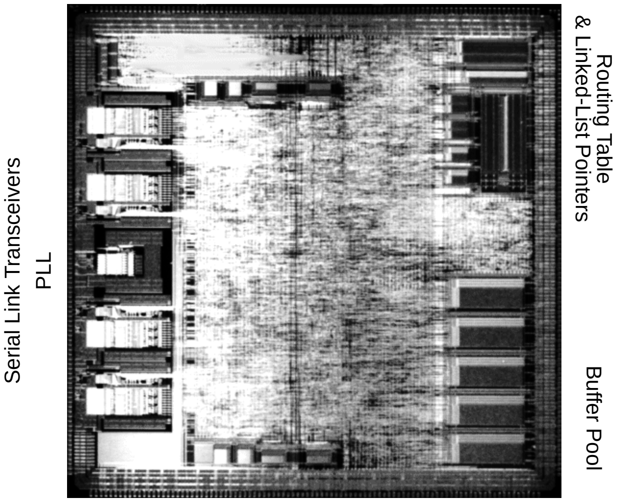
At the system level, facilities were added that allow to diagnose and isolate problems. For example, a faulty oscillator could prevent global synchronization by forcing a node to transmit at a speed that is outside of the tuning range of the other nodes.To diagnose performance problems, programmable counters were added that can be used to gather traffic statistics.

All of these facilities are accessible from any point in the system by means of special in-band control messages. This means that only one node in a system needs to be attached to a controlling computer, which can explore the network topology, compute routing tables, verify operations, log errors, monitor performance, etc. without any other means of communicating to S-Connect nodes.

3 Implementation: The TIC Chip

The TIC chip is the first implementation of an S-Connect router with 6 ports, two of which are parallel ports that are intended to be attached to the local host interface. The TIC is the first router chip ever to integrate multiple serial I/O channels onto one inexpensive CMOS chip that operate at speeds > 1 Gbit/sec and are compatible with fiber optic transceivers as well as conventional cables.

FIGURE 7 : TIC Chip Microphotograph

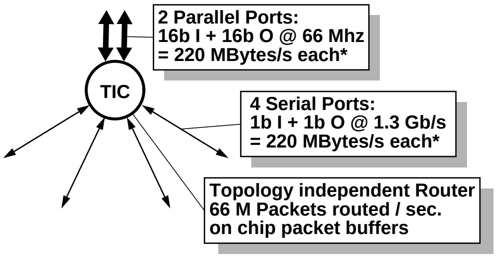


Vital Statistics

Chip Size:	11.8 x 11.8 mm
Technology:	0.65 µm CMOS, 2 Metal Layers
Power Supply:	4W @ 5V
# of Logic Gates:	48,000
Design Style:	Custom + Gate Array
On-chip Memory:	7520 bits
External Clock:	66 Mhz
Test Support:	Full ATPG Scan

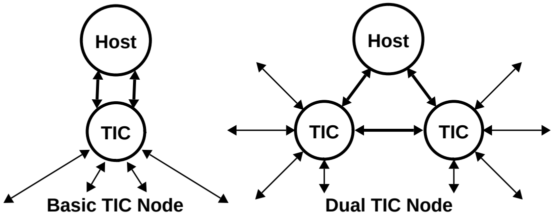
From the user perspective, the TIC is a simple building block to construct arbitrary, high performance switching fabrics, without any complicated glue logic. The local host interface allows the concurrent injection of two messages while two other messages are being received. This host interface uses a simple parallel port, complete with flow control signals and means to control message priorities.

FIGURE 8 : The TIC chip[1]



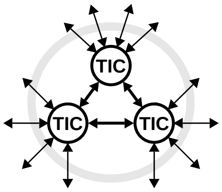
The parallel ports use a symmetric interface with just 2 control wires (*valid:* a message is ready, *accept*: the message has been read), so that the TIC/host port can be directly connected to another TIC/host, without any other circuits. The TIC logic can be programmed to present transient traffic on a specific port so that it is possible to construct dual-TIC nodes (Figure 9) that have 8 channels to other nodes. These higher dimensional topologies boost bandwidth significantly and are appropriate for large, high end configurations.

FIGURE 9 : TIC based node configurations



The same feature allows the construction of pure switching nodes, where 2 or 3 TICs can be used as a 8/12 ported switch. While S-Connect systems usually don't require switches, there are cases (repeaters to extend the range, wiring closets or distribution panels) where switches are useful, in particular when such switches are cheap. The 12 ported switch in Figure 10 requires 3 TIC chips, one VXCO (crystal oscillator) and a box with power supply: no other active electronics of any kind areneeded.

FIGURE 10 : 12 port, TIC based router



Logically, the TIC has 6 packet sources and 6 packet drains that are connected through a set of 40 queues (Figure 11). For each serial outbound channel there is one queue for

[1] The bandwidth figures are the usable packet throughput. The parallel ports transfer data only on 5 out of 6 cycles. The serial links carry an additional 16 CRC and handshake bits, that are not included.

nels is compared to the local clock such that the differences are used to adjust the local clock, which in turn is exported to all attached nodes.

To the demonstrate proper synchronization of the circuit in Figure 6, a system of n nodes is analyzed. The absolute phase $\phi_i(t)$, $i = 1..n$ of each oscillator (VCO) depends on the time t. If the system was turned on at $t = 0$, the phases are:

$$\phi_i(t) = \phi_i(0) + \int_0^t \left(\omega_i + K_0 K_d \sum_{i \neq j} a_{ij} \; (\phi_j(x) + d_{ij} - \phi_i(x)) \right) dx \quad (1)$$

ω_i is the frequency [rad/sec] of the i^{th} oscillator if the control voltage is set to 0. K_o is the conversion factor for the VCO in rad per volt, and K_d is the sensitivity of the phase detector in volt per rad. At this point, a linear phase detector is assumed. It can be shown that limiting the output of the phase detector will still result in a proper synchronization. This limiting is essentially the characteristic of a frequency / phase detector. A plain phase detector would produce ambiguous and non-monotonic outputs that invalidate this analysis. The product $K = K_o K_d$ is the dimensionless loop gain.

The network topology is specified by the coefficients a_{ij} which is the number of channels from node j to node i. Each channel has a certain delay d_{ij} which is expressed in terms of a phase shift of a signal with the steady state operating frequency ω.

Differentiation of Equation 1 yields a non-homogeneous linear equation system:

$$\frac{\partial}{\partial t} \vec{\phi}(t) = \mathbf{A} \vec{\phi}(t) + \vec{F} \quad (2)$$

with:

$$\vec{\phi} = \left[\phi_1 \; ... \; \phi_n \right]^T$$

$$\mathbf{A} = \left[a_{ij} \right]_{i,j=1}^{n} \quad ; \quad a_{ii} = -\sum_{i \neq j} a_{ij} \quad (3)$$

$$\vec{F} = \left[f_1 \cdots f_n \right]^T \quad ; \quad f_i = \omega_i + K \sum_{i \neq j} a_{ij} d_{ij}$$

The structure of **A** is critical to the remaining steps. **A** represents a strongly connected graph (i.e., the inter-cluster network) and is therefore irreducible. Because of the assumption that a channel from node i to node j implies the existence of a channel in reverse direction, $\mathbf{A} = \mathbf{A}^T$. Furthermore since **A** is real, it is also hermitian. Therefore **A** has only real eigenvalues. The diagonal elements of **A** are negative semi-dominant, hence **A** is negative semi-definite: all eigenvalues $\lambda_1 ... \lambda_n$ of A are ≤ 0. Equation 3 also implies that one eigenvalue is 0 and that the corresponding eigenvector is $\vec{1}$ because $\mathbf{A}\vec{1} = 0$.

Let $\mathbf{P} = \left[\vec{x}_1 \cdots \vec{x}_n \right]$ be the matrix of right eigenvectors of **A**, $\mathbf{D} = \mathrm{diag}[\lambda_1 ... \lambda_n]$ be a diagonal matrix composed of the corresponding eigenvalues, and $\mathbf{Q} = \left[\vec{y}_1 \cdots \vec{y}_n \right]$ be the matrix of left eigenvectors. Therefore $\mathbf{A} = \mathbf{PDQ}^T$. Rewriting Equation 2 yields:

$$\frac{\partial}{\partial t} \mathbf{Q}^T \vec{\phi}(t) = \mathbf{DQ}^T \vec{\phi}(t) + \mathbf{Q}^T \vec{F} \quad (4)$$

This decoupples the equation system into n independent differential equations. Introducing $z_i(t) = \vec{y}_i^{\,T} x_i(t)$ and $g_i = \vec{y}_i^{\,T} \vec{F}$ leads to:

$$\frac{\partial}{\partial t} z_i(t) = \lambda_i z_i(t) + g_i \quad (5)$$

if $\lambda_i = 0$:

$$z_i(t) = t g_i + c_i = t \frac{1}{n} \sum_{l=1}^{n} f_i + c_i \quad (6)$$

otherwise:

$$z_i(t) = c_i e^{\lambda_i t} - \frac{g_i}{\lambda_i} \quad (7)$$

Backsubstitution with $\vec{\phi}(t) = \mathbf{P}\vec{z}(t)$ gives the phase functions for each oscillator. The integration constants $c_i, i = 1 \cdots n$ could be determined by the state of the system at $t = 0$. However, the precise form of the phase function is not required to demonstrate that the system synchronizes properly, because the phase functions of any system are mere linear combinations of functions described by either Equation 6 or 7.

Equation 6 yields the steady state operating frequency ω, which is the coefficient of t:

$$\omega = \frac{1}{n} \sum_{i=1}^{n} \left(\omega_i + K \sum_{j \neq i} a_{ij} d_{ij} \right) \quad (8)$$

This is mainly the average of the open loop frequencies of the individual oscillators. The delays of each channel contribute to a net increase of the operating frequency. This sets an upper limit of the loop gain because of the narrow tuning range. It turns out that this limit is largely irrelevant because the loop gain is subject to the normal stability consideration that governs the selection of loop gain, natural loop frequency, lowpass filter cut-off frequency, etc. which constrains K even further. A low gain, low natural loop frequency is desirable to make the system less sensitive to transient errors. This reduces the capture range (a non-issue here) and increases the time required for synchronization.

Equation 7 describes the turn-on transient. This is essentially an exponentially decaying function of time because $\lambda_i < 0$. The non-zero eigenvalues are proportional to K, so that higher loop gain reduces the duration of the synchronization time. It is interesting that the convergence of the system is an exponential function in time.

2.6 Error Detection and Recovery

Spatial distribution exposes the interconnect system to more error sources. These range from nodes that are turned off, over ordinary transmission errors to nodes that do not operate correctly. S-Connect uses about 25% of its logic to achieve reliable operation.

At the channel level, messages are guarded by a CRC code. Assuming a bit error rate of 10^{-9}, a fairly pessimistic value for fiber optic systems, a channel will fail to detect a resource assignment requires solving a bipartite cardinality matching problem in each routing cycle. Unfortunately, the time complexity is cubic in the number of channels.

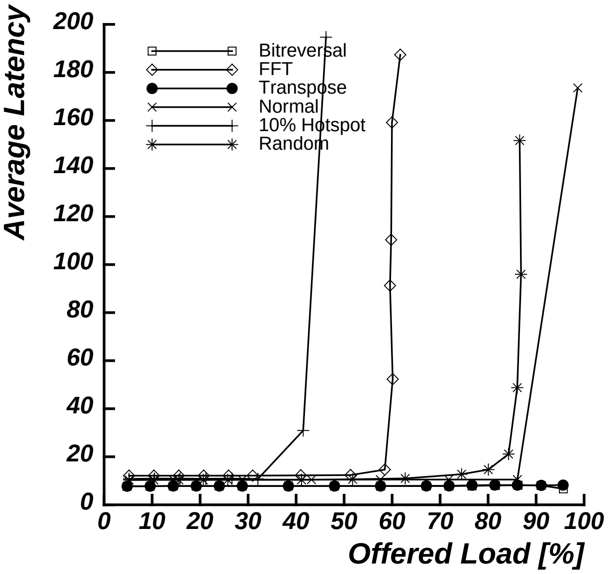
FIGURE 4 : Near Optimal Adaptive Routing

1. Sort transmitters: fewest pending messages first
   messages with multiple options are counted more than once
2. For each transmitter with non-zero pending messages, in the order established by step 1 do:
   1. Among the pending messages, locate the one that has a) the highest priority, b) the oldest age, c) the fewest options, and d) is first by any arbitrary, but defined order. Criteria a-d are used in the given order until a unique selection is made.
   2. Send the selected message and update all data structures
3. Save all unsent messages in the buffer pool until all buffers are filled
4. Send any excess messages to a random destination.

Figure 4 outlines a near optimal adaptive algorithm, in the sense that simulation results were within the statistical error limits of simulations that were based on a router that solves the optimal assignment problem. In step 4, messages can be send to a node that is not specified by the routing table. However, this is a rather infrequent event, that occurs in large systems in about 1% of all messages under saturation conditions.

Figure 5 shows simulation results with the complete algorithm in place for a 64 node, hypercube connected interconnect system, exchanging 10 unit messages and using a buffer pool of 10 buffers (one for each channel plus 4 spares).

FIGURE 5 : Adaptive Router Performance



2.3 Priorities and Resource Allocation

Since the routing algorithm considers priorities as part of the message to channel mapping process, it is easy to design the rest of the router to obey strict priorities, by adding a priority based buffer reservation logic. Priorities in S-Connect guarantee that a message of priority *n+1* will be delivered, even if the entire system has been saturated with messages of priority *n* or lower, and no node is removing any low priority message. This feature solves a deadlock problem that is inherent in cache coherent, distributed shared memory architectures layered on top of a switching fabric without broadcast capability. This problem has been addressed by duplicating the entire interconnect system [11] or by detecting deadlocks and resorting to a software recovery mechanism [12]. Either approach is unnecessary in S-Connect.

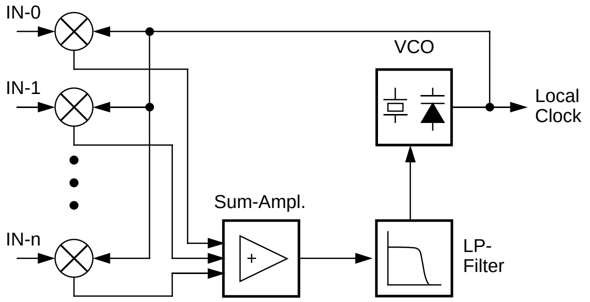
Multiple priorities also proved useful to support real-time applications (video, multimedia, etc.), where traffic with predictable bandwidth must be delivered on schedule.

2.4 Deadlock Avoidance

S-Connect relies on resource ordering to avoid cyclic dependencies [15]. This is accomplished by multiplexing two virtual channels over each physical channel. For each node and each destination, the routing table indicates which virtual channel must be used. While all virtual channel and priorities use the same buffer pool, a reservation logic that is based on the count of packets within each category assures proper operation.

Given an arbitrary interconnect topology it is always possible to prevent deadlocks if packets are not constrained to the shortest route. For example, if all traffic is directed to a designated node such that traffic to the node uses virtual channel 0 and traffic from that node uses virtual channel 1, the resource dependency graph is acyclic and no deadlock exists. This increases the length that a message has to traverse by no more than a factor of two. However in practice, virtual channel assignments can be computed that preserve the shortest path routing. S-Connect uses a heuristic that computes a virtual channel assignment that uses shortest path routing and that permits most alternate paths for adaptive routing [13].

FIGURE 6 : Distributed Phase Lock Loop



2.5 Isochronicity: How to Stay Synchronized

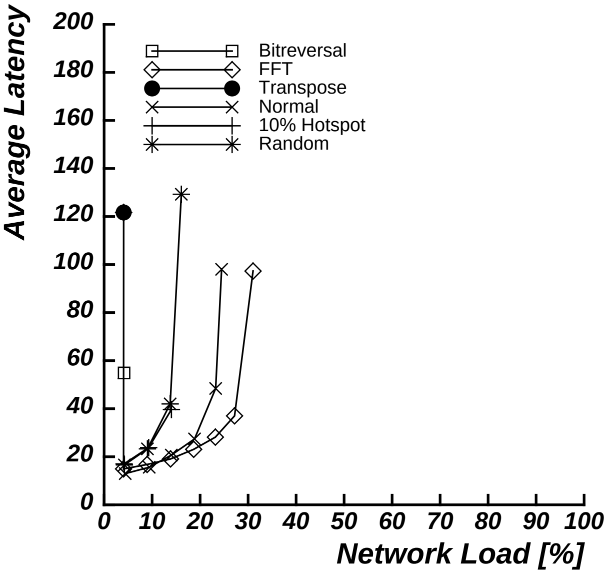
S-Connect operates synchronously, which allows the pipelines of all routers to operate in lock-step fashion without synchronization losses. Other benefits include the availability of a global time and simplified flow-control, because all transmitters and receivers operate at exactly the same speed. This synchronization is achieved through a distributed phase lock loop that relies on the fact that the serial data transmission also exports the clock of the sending node. The phase relation of the clocks of all incoming chan-

2.1 Wormhole Routing Considered Inefficient

Most current router designs are based on the wormhole routing algorithm [7,8], which owns much of its popularity due to its simplicity. This simplicity translates to a short critical path in the case where the router performance is limited by the speed of the router logic. The downside of wormhole routing is its relatively low throughput, generally below 40%, because the communication channels serve also as buffers: a blocked “worm” will tie up resources downstream, which could otherwise be used for traffic headed towards uncontended regions of the interconnect. This deliberate design choice was justified originally by assuming a bandwidth surplus [8]. Subsequent proposals for adaptive wormhole routing algorithms improved behavior in the case of traffic contention, but did not substantially increase the saturation thoughput [9].

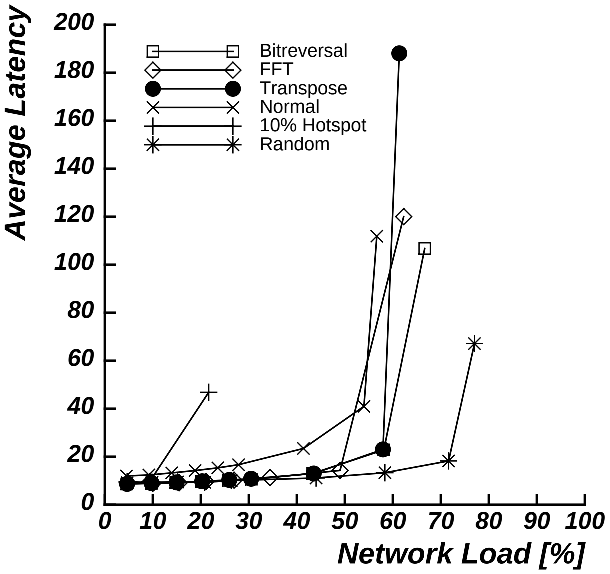
Wormhole routing was derived from virtual cut-through routing [10], which resorts to store-and-forward routing in the case of output contention, and which was considered to be too complex to implement in hardware. The S-Connect system is not limited by the speed of the router logic, rather by the speed at which data can be sent to another chip, a situation common to most advanced CMOS technologies where gate delays are in the sub-ns range, while I/O drivers require about 10 times as much time. Hence the increased complexity due to the implementation of full VC-routing did not limit the cycle time, but greatly improved performance under high traffic loads.

FIGURE 2 : Wormhole Routing Latency vs. Load



Figures 2 and 3 show the performance of wormhole vs. VC-routers for 256 node systems subjected to “difficult” traffic patterns. The random traffic sends messages with a mean length of 10 to equally distributed destinations. the hot-spot traffic singles out a particular node, which receives higher traffic. FFT and Bit-reversal are traffics associated with data structures used in fast fourier transform codes. The transpose traffic results from the transposition of distributed, dense matrices. The normal traffic is associated with divide and conquer algorithms that distribute or combine data, without broadcasting. Details on these traffic loads are given in [13].

FIGURE 3 : VC-Routing Latency vs. Load



Besides poor performance in channel capacity limited systems, wormhole routing also cannot deal with imperfect channels that have transmission delays in excess of the message size. A 100 bit message spans only 18m (60ft) on a fiber operated at 1 Gb/sec, so that by the time the possibly corrupted message arrives, the sending router has lost track of it if more than 18m of fiber were used. Fixing this problem in the wormhole router paradigm requires at least as much logic as building a VC-router that handles the buffering naturally.

2.2 Adaptive Routing Algorithms

Conceptually, the adaptive routing algorithms in S-Connect operate cyclically, where at the beginning of each routing cycle the messages from all inbound channels are collected. Subsequently, the disposition of these new messages and of the ones temporarily stored in the buffer pool are decided by assigning messages to outbound channels, to on-chip buffers, or by rejecting the message. By the end of each routing cycle, messages are transferred to the outbound channels.

In practice, the router employs a pipeline so that message processing is distributed over a number of stages. Furthermore, pipeline bypasses are in place to minimize the cut-though latency: a message headed for an idle channel does not need to go through the buffer pool, rather it can be sent to the channel directly.

Ideally, the router tries to keep all its outbound channels busy. Given that each message should only be sent to a specific subset of the transmitters, namely those that are specified in the destination-indexed routing table, optimal

1.2 Pro & Con of Bit-Serial Data Transmission

The merits of bit-serial data transmissions over parallel connections depend on many technology dependent factors, hence it is necessary to define the context. For the following discussion, we assume technologies that are common to current workstations and high end PCs, (CMOS, surface mount packaging, air-cooling, etc.). Furthermore, we assume that the cost for the interconnect system may not dominate and that it is desirable to achieve the highest possible performance.

CMOS high speed serial links currently offer bandwidths in the range of 1 to 2 Gbits/sec with serialization latencies of approximately 5 system clock cycles [3]. The transceiver circuit for each link requires about 1 mm^2 of chip area and 400 mW of power. High performance parallel interfaces such as those used by Rambus, SCI and FutureBus can operate at about 500 Mhz with latencies of about 2 system clock cycles[4,5,6]. The chip area and power requirements are similar. However, the number of I/O pins is increased. Furthermore, the wires connecting two parallel ports need to have carefully matched delays.

For a circuit board environment where data has to travel about 10 cm, parallel connections offer roughly twice the bandwidth with half the latency for a fixed cost. However the situation changes once the system size increases, because the cost for the wires, which is negligible within one board, becomes significant. Connectors, RF quality transmission line designs, data skew, clock distribution and synchronization become important issues. The fact that bit-serial data transmission provides at least four times more bandwidth per wire means that for systems of about 10 boards, both methods offer the same bandwidth/$.

For systems that extend beyond one box, the cost for parallel interfaces increases significantly because they require high quality cables and connectors with multiple, matched transmission lines, such as those use for HPPI systems. Bit serial systems have a clear advantage due to smaller cables and connections. Moreover, bit serial signals allow transformer coupling, which avoids the need for galvanic connections, which decreases EMI susceptibility and spurious emissions.

Finally, once the domain of the interconnect system is extended beyond the machine room, parallel connections are no longer cost-effective. Practically all existing and planned cable plants use bit-serial media (coaxial/TWP cables or fiber-optics).

2 The S-Connect Switching Fabric

S-Connect switching fabrics are composed of routing elements with limited fan-out that are connected by bidirectional, full duplex channels of fixed, but potentially large delay. No a-priori assumption is made about the topology, which means that each routing element includes a routing table that maps messages to outbound channels. This avoids the need for source directed routing and allows each node to base its routing decisions on the local traffic situation.

At the core of each routing element is a pipelined switch that moves messages from the receivers of the attached channels to the appropriate transmitters. Due to being full duplex, the receiver/transmitter pair at each end of a channel can cooperate to piggyback handshake and flowcontrol information on messages travelling in the reverse direction. Through means described below, the pipelines of all switches operate synchronously.

FIGURE 1 : Router Core Characteristics

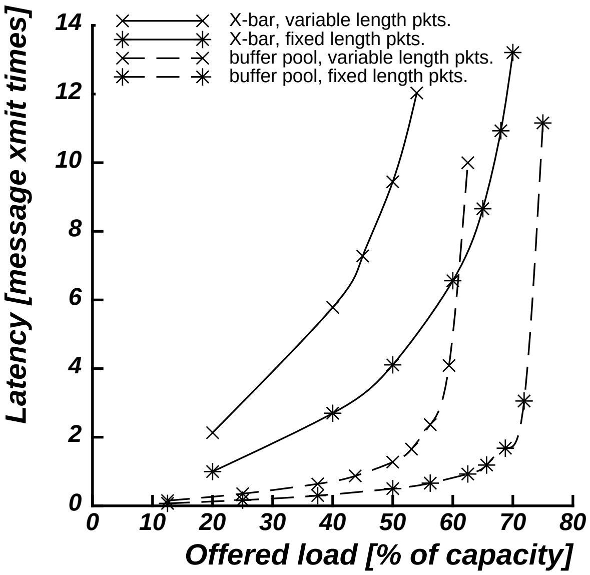


Figure 1 shows the reason for using a fixed message length and for using a shared buffer pool at the core of S-Connect routers: given an ideal, 8 by 8, order preserving router, the latency is shown as a function of the offered load, assuming a random distribution of destinations. In the case of variable length messages, it is assumed that the mean length is 1 and that it has a negative exponential distribution. Other length distributions show similar degradations, in particular bimodal distributions hurt performance if relatively few long messages are added to a traffic of predominantly short messages. The graphs for the cross-bar switch assumes input buffering while the buffer pool is essentially equivalent to an input- and output-buffered crossbar with zero buffer transfer times. In this case, fixed length messages offer higher saturation throughput (75% vs. 64%). While the saturation throughput for either structure is the same, the buffer pool approach has lower latencies.

The actual router differs from the ideal model by having more than one buffer per channel and by not preserving message order which allows adaptive control algorithms. Combining these features result in better channel utilization and higher saturation throughput.

S-Connect: from Networks of Workstations to Supercomputer Performance

Andreas G. Nowatzyk, Michael C. Browne,
Edmund J. Kelly[†], Michael Parkin

Sun Microsystems Computer Corporation
[†]Sun Microsystems Incorporated
e-mail contact: agn@acm.org

Abstract

S-Connect is a new high speed, scalable interconnect system that has been developed to support networks of workstations to efficiently share computing resources. It uses off-the-shelf CMOS technology to directly drive fiber-optic systems at speeds greater than 1 Gbit/sec and can realize bisection bandwidths comparable to high-end MPP systems while being >10x more cost-effective. S-Connect systems do not rely on centralized switches, but rather are composed of adaptive, topology independent routing elements that are integrated into each node. The S-Connect routing algorithm is optimized for fine grained, irregular traffic and is designed to support high traffic loads, that can utilize most of the physically available bandwidth. Such traffic is typical of a distributed shared memory system, which is one of the intended applications. S-Connect innovations include a novel distributed phase locking method that allows global synchronization, HW support for multiple message priorities, in-band monitoring and control facilities, and a low overhead channel protocol that supports multiple in-transit messages on the same fiber.

The first version of the S-Connect switching element has been successfully implemented in a commercial, 0.65 μm CMOS process.

1 Introduction and Motivation

Most of the prior research on high performance switching fabrics assumed complete control over the interconnect topology and relatively short connections between the switching elements. The result of this effort is a solid understanding on how to construct interconnect systems for traditional MPP machines, such as the Cray T3D, Intel Paragon, TMC's CM-5, IBM's SP-2 etc. However, once the scope of the problem is extended to include networks of workstation or high end PCs, the interconnect architecture must be able to deal with nodes that are physically separated by up to about 100m. Furthermore, viable interconnect topologies are constrained by the layout of the cable plant, the geographical distribution of machines, domains of administrative control, etc.

To appear in the proceeding of the
22nd annual International Symposium on Computer Architecture, June 1995.

On the other end of the spectrum, traditional and future networking technologies, such as ATM, are being proposed. However, these methods provide only a small fraction of the realizable bandwidth of a typical fiber optic system. For example, future ATM switches are expected to achieve latencies of 5 to 7 μsec [1]. Speeds are only slowly moving to a single 622 Mb.sec connection for each node while the per port cost for such system is >> $1K per port. S-Connect offers about 10x lower latencies, up to 10x more bandwidth, and eliminates the need for a separate switch, while reducing the cost to that of one interface chip.

Building a low latency, high bandwidth interconnect system solves only half of the problem of providing the user with an integrated environment with efficient access to all computing resources. For example, the processing of conventional networking protocol stacks can easily limit performance. Hence S-Connect is specifically designed to be used with systems that provide low overhead communication directly to the user process, for example in the form of distributed shared memory [2].

1.1 The Case for Spatially Distributed Processing

Foremost among the reasons that workstations and PCs rose to dominate computing are their relatively low cost and incremental scalability. Adding one workstation will not stand out in most budgets. Control over these machines tend to rest with the user, resulting in predictable performance and availability. Among the technical reasons that favor distributed deployment of computing resources is the fact that high resolution display devices require considerable bandwidth. Multimedia, integrated cameras, scanners, and video conferencing applications are likely to increase the demand for local processing capacity. It appears unlikely that this trend will be reversed in the near future, hence a sizable fraction of memory, CPUs, and I/O devices will stay outside of machine rooms, distributed over entire buildings.

Unfortunately, the vast majority of the distributed machines will receive rather poor utilization. The machine configuration will generally be upgraded to comfortably deal with the largest common application, even if the most frequent applications (screen savers, e-mail, word processing, spreadsheets, etc.) require far less memory and CPU power. This leads to a large pool of potential compute resources. Using these resources requires means to access them efficiently and in a manner that will not degrade the interactive responsiveness.