Design and Implementation of Xcent-Net

Kyoung Park, Jong-Seok Hahn, Won-Sae Sim, and Woo-Jong Hahn

Abstract

Xcent-Net is a new system network designed to support a clustered SMP called SPAX(Scalable Parallel Architecture based on Xbar) that is being developed by ETRI. It is a duplicated hierarchical crossbar network to provide the connections among 16 clusters of 128 nodes. Xcent-Net is designed as a packet switched, virtual cut-through routed, point-to-point network. Variable length packets contain up to 64 bytes of data. The packets are transmitted via full duplexed, 32-bit wide channels using source synchronous transmission technique. Its plesiochronous clocking scheme eliminates the global clock distribution problem. Two level priority-based round-robin scheme is adopted to resolve the traffic congestion. Clear-to-send mechanism is used as a packet level flow control scheme. Most of functions are built in Xcent router, which is implemented as an ASIC. This paper describes the architecture and the functional features of Xcent-Net and discusses its implementation.

Ⅰ. Introduction

Over the past few years, parallel computing has expanded from the realm of highly specialized computing systems used primarily for research and military applications, to the world of high-end commercial enterprise computing. Symmetric multiprocessing (SMP), also known as shared memory multiprocessing and massively parallel processing(MPP) are the two most important forms of parallel processing in the high-performance computing industry[1, 2].

MPP with message-passing paradigm has advantage in achieving scalable performance. However, the burden on programmers to distribute the computations and data sets over the nodes causes the lack of programmability. On the contrary, SMP is suitable for general-purpose multi-user applications where programmability is one of the major concerns. A major shortcoming of SMP is poor scalability relative to MPP[2].

Recently, clustered SMP systems, so called CSMP hereafter, are becoming increasingly popular. CSMP is a hybrid architecture to overcome the shortcomings of both architectures. CSMP involves a high speed system interconnect connection of a number of SMP nodes which are more powerful than the nodes of a MPP. A typical CSMP is designed around a group of workstations connected through a high speed LAN. The main problem with such design is that the hardware and software latencies are so large that a significant percentage of CPU cycles are devoted to serve the communication through the LAN[3, 4].

We have explored the requirements of system interconnection for SPAX that is a CSMP running commercial applications. Latency, bandwidth, availability and implementation feasibility are discussed as the requirements of it[5, 6]. Concerning those, we designed and implemented Xcent-net that is a duplicated hierarchical crossbar network providing the connections among 16 clusters of 128 nodes. It is a packet switched, virtual cut-through routed, point-to-point network that has the aggregated bandwidth of 67.584 Gbytes/sec.

The purpose of this paper is to discuss the design and the implementation of Xcent-Net. The architecture of SPAX and Xcent-Net are presented in section 2. The functions of Xcent-Net are described in section 3. In section 4, we discuss the physical implementation of Xcent-Net and finally address the conclusions in section 5.

Ⅱ. The Architecture of SPAX and Xcent-Net

SPAX is a CSMP providing scalability and high performance with the dedicated system network, Xcent-Net. The entire system can be composed of up to 16 clusters. As shown in Figure 1, each cluster consists of 8 SMP nodes which can be any combination of processing nodes, input/output nodes and communication control nodes. Figure 2 shows the structure of a node which comprises 4 Pentium-Pro microprocessors and shared local memory[5, 6].

A group of processing nodes executes collections of closely coupled tasks that communicate frequently by fine grained messages in SPAX. The hardware support of inter-node communication is essential to the performance of the system. Therefore, Xcent-Net which is a duplicated hierarchical crossbar network as shown in Figure 3, is developed to provide high bandwidth and low latency

Manuscript received July 15, 1996; accepted October 17, 1997.

The Author is with Processor Section, ETRI, Taejon 305-600, Korea.

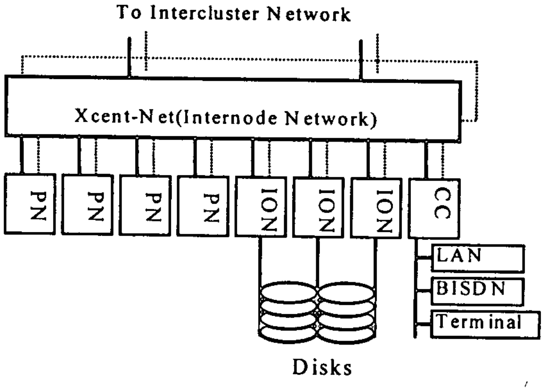


Fig. 1. Structure of a typical cluster.

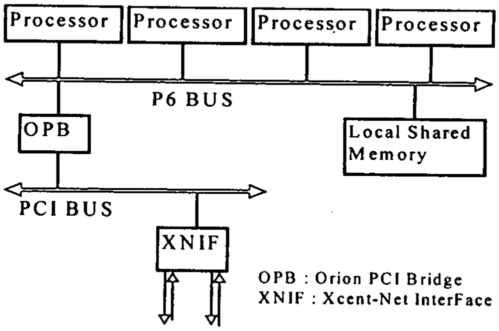


Fig. 2. Structure of a SMP node.

inter-node communication.

Crossbar networks provide the highest bandwidth and the lowest latency compared with others. A crossbar network is visualized as a single-stage switch network which provides permutation connections among nodes. Therefore, the n×n crossbar connects at most n pairs of one-to-one connections at a time[2, 7, 8]. That is the major reason the router of Xcent-Net is designed based on 10× 10 crossbar, which is depicted in section 4.

Scalability is a critical requirement for parallel processing systems to provide the smooth growth for customers as their computing requirements grow. However, the crossbar network is the most expensive one to build, due to the fact that its hardware complexity increases exponentially[2]. Xcent-Net provides the cost-effective scalable interconnection. It is divided into two hierarchies; internode network and intercluster network. Since the amounts of accesses to the remote cluster are small percentage in typical applications, a 8 : 2 fat-tree[2, 5, 6] is adopted as the hierarchy of Xcent-Net.

An interconnection network with hundreds of nodes must have built-in reliability to ensure high availability of systems[6]. For high availability, Xcent-Net supports dual paths to every node through a duplicated hierarchical crossbar network. In normal operation, traffic is spread across the two networks to achieve the optimal load distribution, but either one can take over the whole load after a failure.

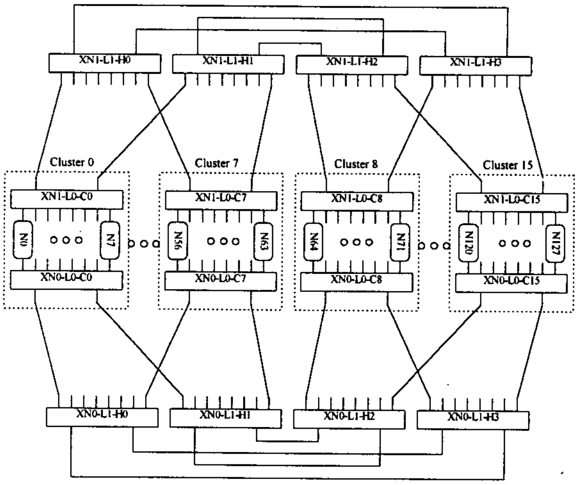
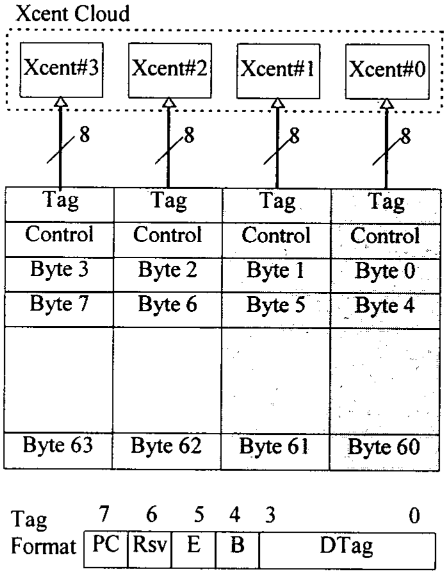


Fig. 3. Architecture of Xcent-Net.



Tag	Tag	Tag	Tag
Control	Control	Control	Control
Byte 3	Byte 2	Byte 1	Byte 0
Byte 7	Byte 6	Byte 5	Byte 4
Byte 63	Byte 62	Byte 61	Byte 60

Tag	7	6	5	4	3 0
Format	PC	Rsv	E	B	DTag

Fig. 4. Packet structure.

III. Functions of Xcent-Net

In commercial systems running mainly OLTP applications, short messages and short I/O transfers are common, and the architecture must not penalize short message latency even if a long transfer is currently in progress. Design for short latency imposes requirements such as short packet size and efficient routing method[9].

Xcent-Net is a virtual cut-through routed[10, 11], packet-switched, point-to-point network. Source synchronous transmission technique is adopted to move data through the physical links[9, 12].

The element of Xcent-Net, Xcent router, is a 10×10 crossbar router which has separate I/O ports of a byte wide data path. Xcent cloud consists of four Xcent routers to provide 32 bits of data paths.

1. Packet

The messages initiated by a node are broken into a number of packets. The packet is a basic unit of information exchange among nodes. Figure 4 shows the structure of the packet used in Xcent-Net.

Each packet has tag flits, a control flit, and data flits. Four tags in a tag flit contain the routing information, which are used by Xcent cloud to decide the output port. All four tags are identical in a tag flit. The control flit is used by the node to identify the type of messages and the sequence number of the packet in a message. The maximum payload of a single packet is 64 bytes to reduce the worst case latency. Restricting the packet length also improves fairness in usage of the links and reduces the buffer size.

The followings explain the contents of the tag.

o PC : Packet Class

PC makes an identification between system packet and normal packet.

o E : Emergency

When E field is set, the packet has the higher priority on arbitration

o B : Broadcast

When B field is set, the packet is broadcasted to all output ports regardless of Dtag.

o Dtag : Destination Tag

Four bits of Dtag field specifies a destination output port among ten output ports

o Rsv : Reserved for future use

2. *Source Based Routing*

Xcent-Net uses source based routing; the entire path of a packet through Xcent-Net is completely determined by routing information placed in the packet tag flits by the sender[13]. This approach has several benefits. The traffic characteristics can be considered in the routing of a packet. For example, different types of messages can be routed through separate dual networks. Source based routing is flexible since major changes in network routing policy can be accomplished simply by modifying the route generation algorithm used in the packet sender. Also it can easily supports system reconfiguration to isolate malfunctioned network links.

Xcent-Net uses multiple tag flits to specify the path of a packet through Xcent-Net. With up to 4 tag flits, the sender can transfer a packet to any other node or nodes in SPAX. Use-and-drop mechanism is adopted to handle the multiple tag flits.

Figure 5 shows this mechanism in Xcent-Net. The sender places 4 tag flits in a packet to specify the routing path to the receiver. The first tag flit, Tag0 is used to find the correct path through the internode stage and discarded by the internode stage. Then the second tag flit, Tag1 is used in next stage. Finally, the receiver gets the packet that has only control and data flits.

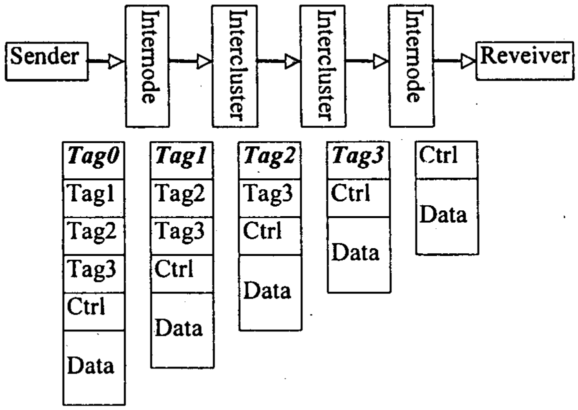


Fig. 5. Use-and-Drop mechanism for multiple tag flits.

3. *Arbitration*

The arbitration mechanism of Xcent-Net is two level priority-based round-robin method, which is built in Xcent router. The priority scheme is very efficient way to reduce the latency of short messages when a long message is currently in progress. A round-robin scheme guarantees that no packet is starved for service.

According to the emergency field in the tag, the priority of the packet is classified, and the requests with same priority are resolved by round-robin method. To prevent the starvation due to the priority, new requests with higher priority are not serviced until currently pending requests are resolved.

Other important features of the arbitration are broadcasting and adaptive ports selecting. Xcent-Net supports broadcasting from a node to all other nodes. When a broadcasting request occurs, Xcent router replicates the packet to all nodes after currently pending requests are resolved. Internode network and intercluster network have 8 child ports and 2 parent ports due to fat-tree like hierarchy. The parent ports are designed as adaptive ports to increase efficiency of Xcent-Net. On request to parent ports, Xcent router makes a connection to an idle port among 2 parent ports.

4. *Buffering and Flow Control*

Virtual cut-through method offers a compromise by combining the store-and-forward and wormhole routing[2, 9]. Xcent-Net uses virtual cut-through method to reduce the network latency and handle the blocking efficiently.

The virtual cut-through in Xcent-Net is implemented using packet buffers and clear-to-send flow control mechanism. Each port of Xcent router has an input packet buffer to store a packet. When a packet arrives at the input port, incoming flits are enqueued in the packet buffer. The tag flit of a packet comes out of the packet buffer to request arbitration as soon as it is enqueued. After the arbitration to set up the desired path, the remaining flits in a packet buffer are transferred while the packet is still being enqueued.

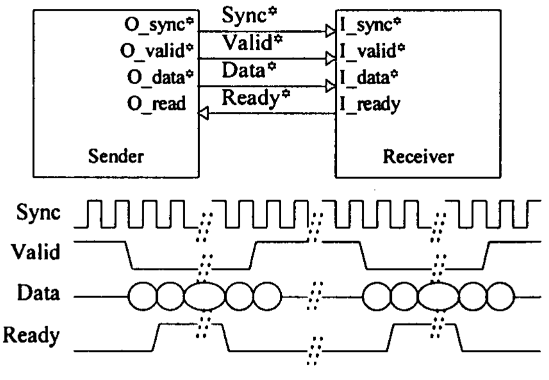


Fig. 6. Source synchronous transmission.

The input port of a receiver sends the flow control signal to the output port of a sender to notify the availability of a buffer. The sender always checks the flow control signal before the transmission of a packet; Clear-to-send flow control.

5. Plesiochronous Clocking

One of the most difficult tasks in putting together a large scale interconnection network is a global clock distribution[13]. Xcent-Net fixes up this problem by using plesiochronous clocking. Each Xcent cloud uses a different local oscillator with the same nominal frequency. There is no global clock and the clock distribution problem is entirely avoided.

In Xcent-Net, the clocks of Xcent clouds are all free running. Source synchronous transmission technique is used to transmit data via plesiochronous interfaces between a sender and a receiver. In source synchronous transmission, a sender transmits data with the clock as a synchronization signal which a receiver latches incoming data with[13]. This makes it easier to scale the system to higher frequency, and avoids the performance losses of asynchronous networks caused by handshaking.

As shown in Figure 6, there are two synchronizing signals to provide the timing information regarding data movements between a sender and a receiver. The sender transmits the data with its local clock as a flit synchronization signal called *Sync**. The additional synchronization signal named *Valid** is transmitted to synchronize the packet frame. Then, the receiver uses this incoming *Sync** signal to sample the incoming data and *Valid** signal to identify the first flit and the last flit of an incoming packet.

Ⅳ. Implementation

The major components of Xcent-Net are Xcent router, backplane and cable. Xcent routers perform the switch functions including buffering, arbitration, routing, flow control and transmission. Xcent routers are mounted on the backplane providing the physical connections for internode network or intercluster network. The cables are used for connection between the backplanes.

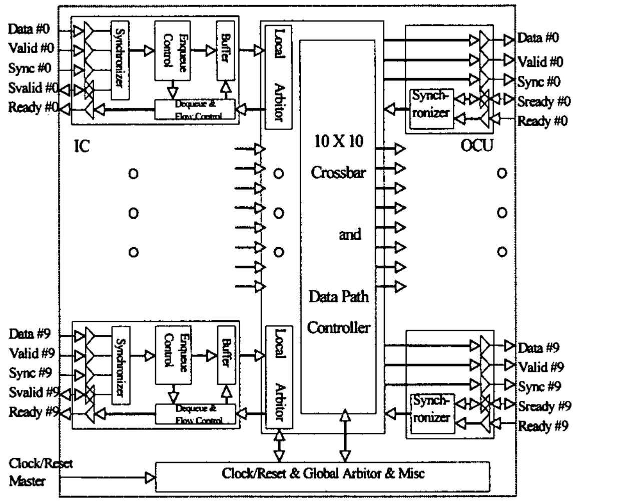


Fig. 7. Architecture of Xcent router.

1. Xcent Router

The element of Xcent-Net is Xcent router. Most of the functions described in section 3, are implemented in a Xcent router. As shown in Figure 7, Xcent router consists of 10 input ports with a packet buffer, 10 output ports, 10 local arbitors, a single global arbitor and a 10×10 crossbar with its data path controller.

The input port is composed of a synchronizer, a packet buffer, an enqueue controller and a dequeue/flow controller. An incoming packet is sampled with the incoming *Sync** signal and stored in the packet buffer which is implemented with a dual ported static RAM. The writing port of the packet buffer is controlled by the enqueue controller with the incoming *Sync** signal. The reading port is controlled by the dequeue/flow controller with its local clock. The flits of a packet are synchronized with its local clock just by passing through the packet buffer. *Valid** signal is synchronized with local clock by the synchronizer to decide the start point of dequeueing flits.

After the synchronization of the incoming *Valid** signal, the tag flit is transferred from the packet buffer to the local arbitor or the global arbitor according to the broadcasting field. After arbitration, the dequeue/flow controller read out the remaining flits from the packet buffer and transmits them through the crossbar. When the arbitor fails to set up the desired crossbar path, the flit flow of a packet is blocked in the buffer.

The dequeue/flow controller controls packet flow using a dedicated flow signal called *Ready**, which is back propagated from the input port of a receiver to the output port of a sender.

Ten local arbitors connected into each input port arbitrates based on round-robin with two level priorities. A global arbitor performs the arbitration for broadcasting. The crossbar and data path controller set up and provide the data path according to the status of local arbitors and a global arbitor.

The output port drives the outgoing data with its *Sync** and

Table 1. Cycle count through Xcent router.

Cycle No.	Description
1	latch
2-3	*Valid** synchronization
4	synchronization of Xcent cloud
5	arbitration
6	crossbar set up
7	data out

Table 2. Implementation details of Xcent router ASIC.

Target library	LCA300K(LSI)
Technology	0.6μ CMOS gatearray
Metal layer	2
Gate count	50K gates
Signal count	276
I/O buffer	TTL (4mA output driver)
Package	383 CPGA(22×22)
Operation speed	66MHz
Operation voltage	5V±10%
Power consumption	3.5W

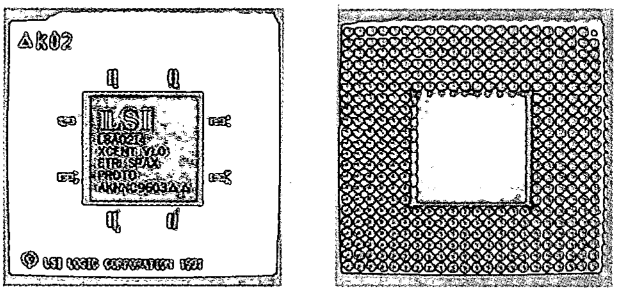


Fig. 8. Photograph of Xcent router ASIC.

*Valid** signal. Also the output port samples the back propagated flow control signal from a receiver and sends it to the local arbitors and the global arbitor to control the packet flow.

The latency through Xcent router is 7 clock cycles without blocking as described in Table 1. The latency of a packet through Xcent-Net, Tp can be calculated as Formula 1. Therefore, the latency of a maximum sized packet through the longest path is 45 cycles.

Tp = (# of stages) X 7 + (# of data flits + 1) --- Formula. 1

The design capture of Xcent router is done with Verilog hardware description language. After logic simulation, the design source is synthesized using LSI's LCA300K ASIC library. The result of synthesis is fabricated with 0.6 Double metaled CMOS gatearray and packaged in Ceramic PGA of 383 pins. The implementation details of Xcent router are summarized in Table 2. Figure 8 shows the photograph of Xcent router ASIC.

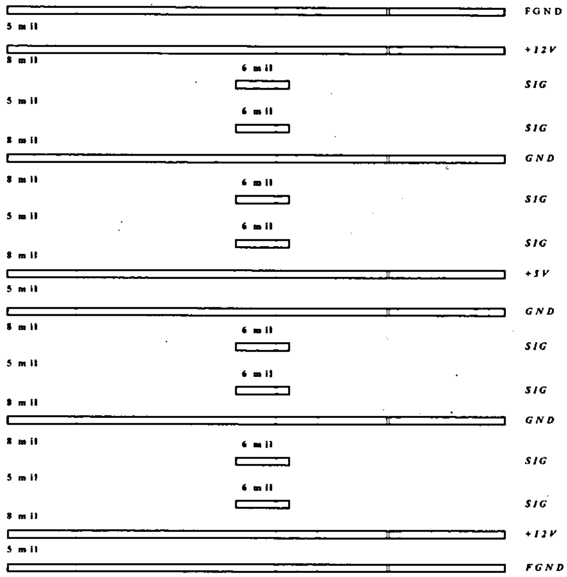


Fig. 9. Stackup of 16 layered backplane.

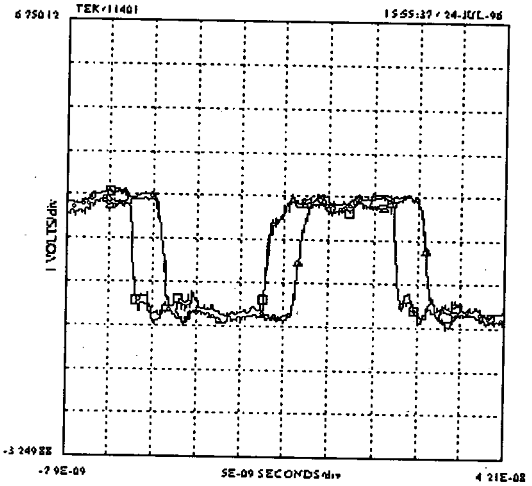


Fig. 10. Driven line measurement of a signal in backplane.

2. Backplane and Cable

The backplane that mounts two Xcent clouds(8 Xcent routers) and connectors is 16 layered, impedance controlled PCB. Specially the skew among the signals in a same port is controlled accurately to support source synchronous transmission. The single-ended impedance of the backplane is 53Ω which is created using 6 mils traces of one ounce copper in a dual stripline configuration. Figure 9 shows the stackup of 16 layered backplane.

To reduce the crosstalk between adjacent stripline layers, all traces on consecutive signal layers are routed othogonally. On parallel routed traces, no adjacent trace is closer then 50 mils edge-to-edge. Xcent routers on the backplane drives the signals

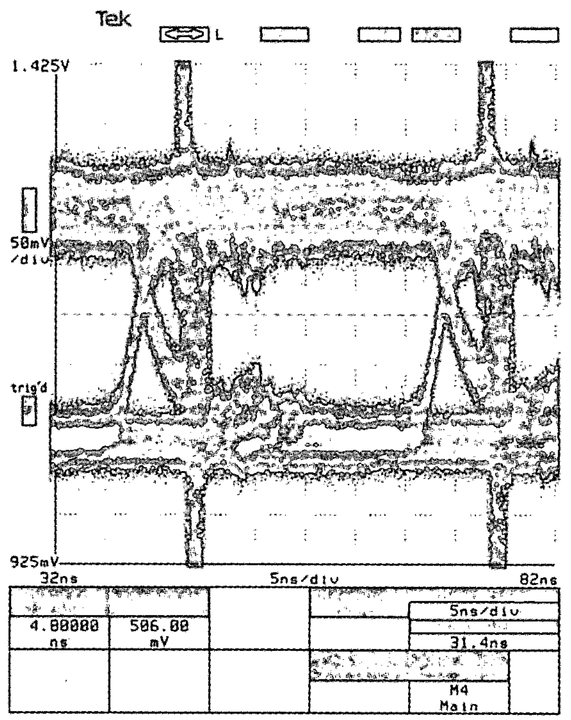


Fig. 11. Driven line measurement of LVDS signal.

using 4 mA single-ended TTL technology. To reduce the reflections, the corners are made at 45 degree angle. The skew among the signals in a same port is controlled less than 300 ps.

Figure 10 is the measured result of a signal that is driven by output buffer of Xcent router. The waveform marked with squares is a 4 mA TTL output signal that is driven by Xcent router through a 20 inches single stripline with 33 Ω termination. The probing point of the waveform marked with triangles, is the connector pin in the backplane. As shown in the figure, the waveform is virtually square, with only a slight undershoot on falling edge. The propagation delay through the backplane is about 2 ns.

The LVDS(Low Voltage Differential Signal) is adopted for driving cables between the backplanes. The signals driven by Xcent routers are converted to LVDS signals by the cable interface cards those are slotted in the backplanes. To reduce the attenuation through the cable path which has 10 meters of the maximum travel length, 30 gagues or larger wires are used. Generally shielded twisted pair cables are used. Figure 11 is the measured result of a LVDS signal that is driven by cable interface card. The waveform is a pair of 33 MHz LVDS signal driven by the DS90C031 driver of National semiconductor through 10 meters of twisted pair wire. The waveform is probed at the cable interface card of receiver. As shown in the figure, the pair of differential signals is well equalized and the voltage difference is about 230 mV. The voltage attenuation of the receiving point is about 170 mV.

Table 3. Features of Xcent-Net.

Network features	
Architecture	Duplicated hierarchical crossbar
Configuration	16 clusters of 128 nodes
Clocking scheme	Plesiochronous
Transmission	Source synchronous
Switching method	Packet switching with Virtual cut-through
Flow control	Clear-to-send
Routing method	Source based routing
Arbitration	Two level priority-based round-robin
Static performance	
Operation clock	33 MHz
Data path width	4 bytes
Bandwidth of a channel	33 MHz×4 bytes × 2 = 264 Mbytes/sec
Bandwidth of a node port	264 Mbytes/sec × 2 = 528 Mbytes/sec
Bandwidth of a cluster	528 Mbytes/sec × 10 = 5.28 Gbytes/sec
Aggregated Bandwidth	528 Mbytes/sec × 128 = 67.584 Gbytes/sec
Shortest latency	24 clocks
Longest latency	45 clocks

V. Conclusion and Future Work

Xcent-Net, a duplicated hierarchical crossbar network, is designed and implemented as the system network of SPAX being developed by ETRI. Up to 16 clusters of 128 nodes are interconnected through it. As summarized in Table 3, it is a packet-switched, virtual cut-through routed, point-to-point network. The packets containing up to 64 bytes of data are transmitted among nodes using source based routing and use-and-drop mechanism. To eliminate the global clock distribution problem, plesiochronous clocking scheme and source transmission technique are used. The switch functions including buffering, arbitration, routing, flow control and transmission are implemented on Xcent router as an ASIC. Xcent routers are mounted on the backplane providing the physical connections for internode network or intercluster network. The cables are used for connection between the backplanes.

Xcent-Net which operates at 33 MHz is integrated with Xcent routers, backplanes and cables. In a Xcent cloud which operates at 33 MHz, a channel bandwidth and the aggregated bandwidth reach to 264 Mbytes/sec and 2.64 Gbytes/sec respectively. Therefore, an internode network or an intercluster network with dual channels provides 5.28 Gbytes/sec of aggregated bandwidth, and 528 Mbytes/sec of bandwidth is available to each node. The aggregated bandwidth of Xcent-Net configured with 16

Table 4. Comparison with other system networks.

	Xcent-Net	SP-switch	BYNET	2D Mesh
System	SPAX	IBM	NCR	Unisys
Max. Nodes	128	512	128	256
Aggregate Bandwidth	67.6 GB/s	76.8 GB/s	8 GB/s	5.6 GB/s
Bandwidth per Node	528 MB/s	150 MB/s	64 MB/s	175 MB/s
Bandwidth of A router	660 MB/s	1.2 GB/s	–	875 MB/s

clusters of 128 nodes, reaches to 67.584 Gbytes/sec.

Table 4 is a comparison with other system networks in terms of various bandwidths. As summarized in it, Xcent-Net provides a node with much higher bandwidth than any other system networks those are currently available.

We finished testing its functionality and measuring the physical characteristics regarding the signal integrity. Currently, we integrate a whole system and try to port ETRI's proprietary operating system that is developed based on UnixWare-MK.

To enhance the performance of Xcent-Net, we will try to scale up the operation frequency over 66 MHz. For better understanding of the dynamic performance, we will also try to measure various performance parameters using benchmarks on the actual system. We will compare the results of measurements with the them of previous work done by ETRI to get the design parameters in early design stage[14].

References

[1] L. Hennessy and D. A. Patterson, Computer Architecture; A Quantitative Approach, pp. 635-693, Morgan Kaufmann Publishers. Inc., San francisco, California, 1996.

[2] Kai Hwang, Advanced Computer Architecture:Prallelism, Scalability, Programmability, pp. 3-96, McGraw-Hill, Singapore, 1993.

[3] R. P. Martin, "HPAM:An Active Message layer for a Network of HP Workstation," proc. of Hot Interconnect II, pp. 40-58, Aug. 1994.

[4] J. C. Hoe, "Network Interface for Message-Passing Parallel Computation on a Workstation Cluster," proc. of Hot Interconnect II, pp. 154-163, Aug. 1994.

[5] B. J. Min, S. S. Shin and K. W. Rim, "Design and Analysis of a Multiprocessor System with Extended Fault Tolerance," proc. of IEEE 5th workshop on FTDCS, pp. 301-307, Aug. 1995.

[6] W. J. Hahn, S. H. Yoon and K. W. Rim, "Design of the New Parallel Processing Architecture for Commercial Applications," Jounal of the KITE, Vol. 33-B, No. 5, pp. 897-908, May 1996.

[7] K. Park, J. S. Hahn, W. S. Sim, W. J. Hahn, "Design of a 10×10 Crossbar Router," proc. of ICEIC '95, pp. 475-478, Aug. 1995.

[8] Y. Tarmir and H. C. Chi, "Symmetric Crossbar Arbitor for VLSI Communication Switches," IEEE Tr. on parallel and distributed systems, Vol. 4, No. 1, pp. 13-27, 1993.

[9] R. Horst and et. al., "Tnet:A Reliable System Area Network for I/O and IPC," proc. of Hot Interconnect II, pp. 95-105, Aug. 1994.

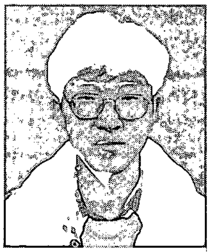
[10] P. Kermani and L. Kleinrock, "Virtual cut-through:A new computer communication switching technique," Computer Networks, Vol. 3, pp 267-186, Sept. 1976.

[11] C. B. Stunkel and et. al., "Architecture and Implementation of Vulcan," proc. of 8th IPPS, pp. 268-274, April, 1994.

[12] W. J. Dally, "Architecture and Implementation of the Reliable Router," proc. of Hot Interconnect II, pp. 122-133, Aug. 1994.

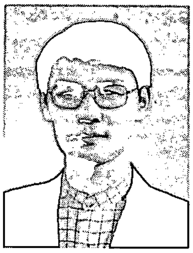
[13] G. A. Boughton, "Arctic Routing Chip," proc. of Hot Interconnect II, pp. 164-172, Aug. 1994.

[14] J. Kim, "Performance Study of Packet Switching Multistage Interconnection Networks," ETRI Journal, Vol. 16, No. 3, pp. 27-41, Oct. 1994.



Kyoung Park is a research staff with ETRI, where he has been involved in the development of high performance parallel processing system since 1993. He participated in the testing and analysis of TICOM in 1993. From 1994, he is working on developing parallel processing system called SPAX, and also working on developing next generation microprocessor. His research interests include large scaled parallel architecture and advanced microprocessor architecture. He received the B.S. and M.S. degree in computer engineering from ChonBuk national university in 1991 and 1993 respectively. He is a member of IEEE and KITE.

Jong-Seok Han received the B.S. degree in computer science from Ajou University in 1989, and the M.S. degree from Yonsei University in 1992. Since 1992 he is a member of technical staff at Computer Division, Electronics and Telecommunication Research Institute(ETRI). His current research interests are in parallel processing architecture, fault-tolerant interconnection network and VLSI ASIC design. He is a member of IEEE, KISS, KITE and the Korean Professional Engineers Association.



Won-Sae Sim has the M.S. and B.S. degree in electronics engineering from ChonBuk national university. He is a senior research engineer of computer development at ETRI. He has joined to three general purpose computer development project for 9 years. His experiences and interests are design of SMP system, system bus design, bus monitoring tool design, backplane PCB design and high speed digital signal integrity.

Woo-Jong Hahn received the B.S., M.S. and Ph.D. degree from Korea University in 1981, 1984 and 1995 respectively. From 1984, he is working on Electronics and Telecommunications Research Institute in Taejon, Korea. From 1986 to 1988, he was working on 64-bit processor and workstation server development project at AIT in Cupertino, CA., USA. From 1988 to 1991, he was working on developing SMP server, so called TICOM, and directed development and test of Processor Board. From 1991 to 1994, he coordinated hardware development and directed development of Memory Board of the next version of TICOM. From 1994, he is working on developing parallel processing architecture, so called SPAX, and also directing development of Interconnection Network. He is a Principal Member of Technical Engineering Staff at ETRI. His professional interests include computer architecture, processor architecture, memory hierarchy in a large scale system, interconncection network, multimedia server. To contact him, email to wjhan@computer.etri.re.kr or fax with +82-42-860-6645.