HPP Switch: A Novel High Performance Switch for HPC

Dawei Wang [123], Zheng Cao [123], Xinchun Liu [12], Ninghui Sun[12]
[1]*Institute of Computing Technology, Chinese Academy of Sciences*
[2]*Key Laboratory of Computer Architecture and System, Chinese Academy of Sciences*
[3]*Graduate University of the Chinese Academy of Sciences*
{wangdawei,cz,lxc,snh}@ncic.ac.cn

Abstract

The high performance switch plays a critical role in the high performance computer (HPC) system. The applications of HPC not only demand on the low latency and high bandwidth of the switch, but also need the effective support of collective communication, such as broadcast, multicast, and barrier etc. In this paper, HPP Switch, as the core component of interconnection network of a HPC prototype, is introduced to meet these requirements. It is with 38.4ns zero-load latency, 160Gbps aggregated bandwidth, 16 multicast groups and 16 barrier groups. HPP Switch is implemented in a 0.13um CMOS standard cell ASIC technology. The simulation results show that the multicast and barrier operations for 1024 nodes are finished within 2us, and the single stage of barrier operation only needs 128ns.

1. Introduction

The high performance switch is a critical component of interconnection network in high performance computer (*HPC*) system. It determines the point-to-point latency, bi-section bandwidth, scalability and reliability of the HPC system. With the increasing number of cores of the processor chip, the switch with the sufficient bandwidth to feed more data into CPU and the lower latency to communicate between nodes is becoming a greater concern. Furthermore, many large-scale HPC applications need the effective support of collective communication primitives, such as barrier, multicast, all-to-all, all-reduced [1~2].

The research and commercial implementation of the high performance switch is already with a lot of efforts. The latency was improved much, for example, Elite, the switch of QsNet, with 20ns zero-load latency [3]; and YARC, the high-radix switch of BlackWidow System, with 31.25ns [4]. With the improvement of high speed signaling technology [5], the bandwidth of the switch now progresses to many Terabits. For example, YARC, which integrates 192 6.25Gbps SerDes macros inside the chip, is up to 1.92Tb/s for bandwidth[1]. There are also supports for collective operations. For example, Infiniscale, the industry Infiniband switch chip, supports the unreliable multicast [6], and Elite supports reliable barrier and continuous port multicast [7]. Detailed performance parameters of current switch are showing in Table 1.

Table1: **The parameters of high performance switches**

Switch Name	Port NO.	Single Port/ Aggregated Bandwidth	Zero-Load latency	Supportive Collective
HPP Switch	16	5Gbps 160Gbps	38.4ns	Barrier, Multicast, Broadcast
XBar16 (Myrinet)	16	2Gbps 64Gbps[8]	150ns[9]	None
Infiniscale (IB)[10]	24	16Gbps 768Gbps	<200ns	Multicast Broadcast
Elite4 (Quadrics)	8	7.2Gbps 57.6Gbps	20ns	Continuous Multicast, Broadcast
BlackWidow	64	15Gbps 1.92Tbps	31.25ns	None

However, to the best of our knowledge, none of these switches meet the requirements of high performance computing simultaneously. To meet all these requirements in one switch chip, we propose HPP Switch, which is the core component of Dawning 5000A networks. The motivation of HPP switch is to provide a switch for the cluster interconnection with enhancement for multi-core and collective communica-

[1] The meaning of bandwidth in this paper is effective bandwidth which does not include 8/10b conversion loss.

1550-4794/08 $25.00 © 2008 IEEE
DOI 10.1109/HOTI.2008.17

tion in the low-cost approach. It is wished to being competitive with Infiniband technology for Cluster HPC. The HPP Switch makes two contributions:

- HPP Switch supports unicast, multicast and barrier communication patterns simultaneously, while providing low latency and high throughput;
- Provides reliable and efficient barrier at small hardware costs.

This paper describes the design and implementation of HPP Switch. The rest of the paper is organized as follows. The design of HPP Switch is described in Section 2. In Section 3, the evaluation and analysis of HPP Switch are provided. The ASIC implementation of HPP Switch is described in Section 4. Finally, the conclusion is presented in Section 5.

2. HPP Switch

2.1. Dawning 5000A Project

The Dawning 5000A is one of the R&D projects of Chinese 863 high-tech program. The target of this project is to build a 100TFlops HPC system, and a prototype for the peta-scale system. The HPP switch is the part of the prototype. HPP (*Hyper Parallel Processing*) is the system architecture of the prototype. The features of HPP include three-layer parallelism with *core-layer*, *intra-node* and *node-layer;* the global address space which enables the remote load/store and UDMA operations. The programming model can be either the message-passing such as MPI [11], or PGAS such as UPC [12]. The prototype consists of three kinds of chips, which are CPU chip, HPP Node Controller chip, and HPP Switch chip. The HPP prototype is like Figure 1. The CPU chip is designed with the quad-core Opteron. The HPP Node Controller supports the global address space and hardware locks. It also contains four network interface controllers (NIC). The system is interconnected by HPP Switch in Fat-Tree topology. The HPP interconnection is consisted of four independent HPP Switches instead of single switch with 4x channels, and the supports for global synchronization. The purpose is to improve the communication throughput between multi-core chips.

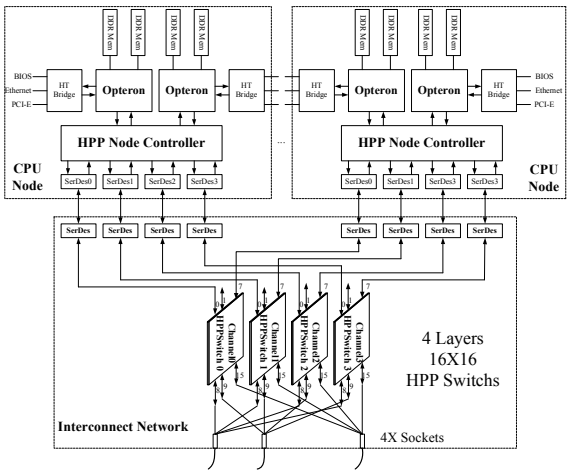


Figure1: **HPP prototype**

2.2. Basic Design

HPP Switch is a 16 port switch with Input-Queued [13] switching and source-address routing. Each port has a bidirectional bandwidth of 10Gbps, and the aggregated bandwidth of the switch is 160Gbps. The topology of HPP interconnection is Fat-Tree, scaling up to thousands of nodes.

The HPP Switch has 4-layer independent crossbars for data exchanging. Each layer crossbar has a corresponding virtual channels (*VC*).The four virtual channel are two unicast data VCs, one unicast/multicast mixed VC (which is capable of sending unicast and multicast packets simultaneously), and one synchronization VC. Each virtual channel has an exclusive receiving buffer. Packets only transmit in one specific virtual channel and its corresponding crossbar.

The packets at the head of an input buffer are transmitted into the routing module, which requests the arbiter at the destination port for transmitting the data. The arbitration scheme is base on the Matrix Arbiter [14], which keeps the strong fairness to all requestors. The granted packets will be transmitted to the transmitting buffer. In the Tx Module, there is Scheduler. The selection algorithm is that Barrier VC has the highest priority, and the Round-Robin is adopted for 3 data VCs. The internal structure is shown in Figure 2.

The two unicast and one unicast/multicast mixed virtual channels are used to diminish the performance degradation caused by head of line blocking [13]. The mixed virtual channel can also send multicast packets.

The synchronization virtual channel provides the hardware level support for barrier operations.

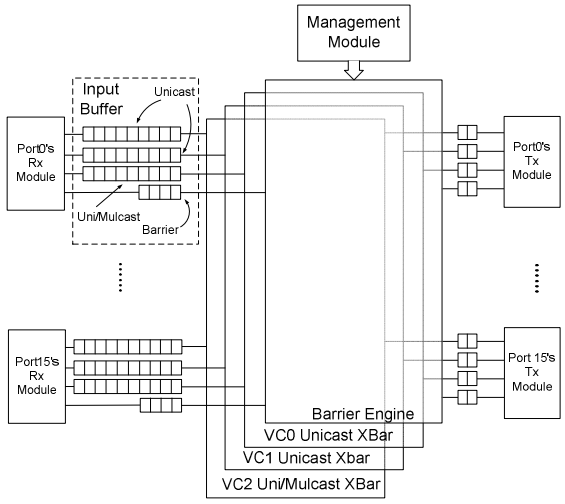


Figure 2: **Micro-Architecture of HPP Switch**

The packets pass the HPP Switch in the form of a 12 stage pipeline, among which the receiving LCL (link control layer) takes 2 stages, the input queuing 3 stages, the routing-granting 4 stages, the transmitting buffer 2 stages, and the transmitting LCL takes 1 stage. When there is no blocking, the latency of a single stage is 38.4 ns. Figure 3 shows the detail of the pipeline.

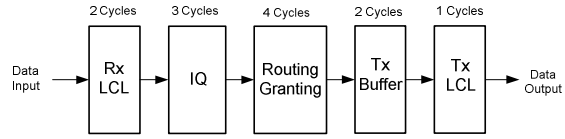
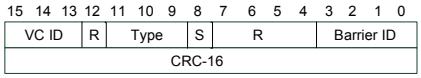


Figure 3: **Internal pipeline states of HPP Switch**

In order to reduce the delay on switching further more, HPP Switch adopts transmitting show-ahead scheme, which can reduce the switching delay to 11 cycles. The show-ahead scheme works in the following way. The starting flit of the packet is sent whether there is packet to be transmitted or not, so that once there is a packet arrived at the next cycle, it can be transmitted immediately, instead of waiting for the starting flit. If the show-ahead misses, the switch will cancel the transmission automatically. Under the condition of low workload, the switching latency can be reduced by 8.33%.

HPP Switch supports both global broadcast and multicast operations. A unicast/multicast mixed crossbar is integrated within the switch, so that the multicast packets received from one port can be output to several ports. Each port of the switch can belong to any multicast group, and one port can belong to several different groups. Until now, HPP Switch is able to support up to 16 multicast groups. As for the problem of multicast deadlock [15], HPP Switch has solved it using RBB (*Resource Bulletin Board*) algorithm.

Barrier is the synchronization point of parallel programs. HPP Switch accelerates barrier using the tree structure, and improves the reliability using retransmission and dun. HPP Switch can support up to 16 barrier groups. The barrier packet format shows in Figure 4. The Type region has 3 bits, to identify the Barrier's type from 4 types, namely, Combine packet, Distribute packet, Combine-ACK packet, and Distribute-Dun packet. The S region has 1 bit, indicating the operation number of Barrier, to distinguish two continuous barrier operations. The Barrier ID region indicates the barrier group which the current barrier operation belongs to.



R: Reserved
S: Sequence Number

Figure 4: **Barrier packet format of HPP Switch**

HPP Switch uses the VCT (Virtual Cut-through) switching method. Although the Wormhole switching can save resources for buffer, the improvement of circuit integration and the increase of memory resource on chip have made such saving not critical. Instead, VCT method can reduce the cost on switching among several virtual channels, to make highly use of bandwidth, and also prevent deadlocks from occurring. For these reasons, HPP chose VCT in practice.

The absolute credit-base method is adopted for the link level flow control of HPP Switch. The credit size is 64Bytes.

HPP adopts the outband mechanism for group management. Outband management can flexibly configure members of multicast and synchronization groups, and monitor the statistic registers inside the switch, taking records of current network traffic information.

2.3. Multicast Communication

Multicast communication is a common communicating primitive of collective communications. The implementation method can be classified into two classes: software-based and hardware-based. The soft-

ware-based multicast usually implements in the Spanning Tree algorithm, with high overhead. The hardware-based multicast is usually implemented in network cards [16] or switches [17~18]. The hardware implementation could diminish the software cost and accelerate the multicast operation. For example, Elite, the Quadrics Switch, supports the hardware multicast. But its multicast ports need to be continuous [18]. One multicast operation might be segmented into several hardware-based sub-multicasts. As a result, the calculation spending for route selection is increased, and application performance is reduced.

HPP Switch adopts CAM (*Content Addressed Memory*) to perform multicast routing, which supports any address pattern, no matter continuous or discrete. Each multicast packet contains a multicast group ID. The switches in each layer of the multi-stage network search for corresponding output ports from such multicast group IDs, then send the packet This ensures that every path from the source to the destination nodes is the shortest. The whole network contains 16 different multicast group IDs.

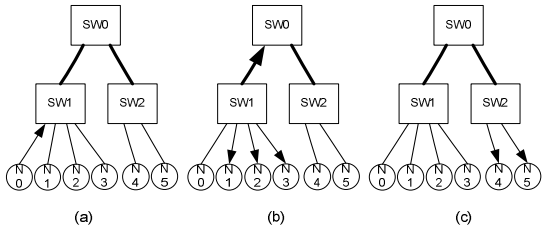


Figure 5: **Multicast transaction of HPP Switch**

Figure 5 shows the process of multicast communication. N0 ~ N5 are the members of the same multicast group. (a) First Step: N0 sends a multicast packet out. (b) Second Step: when the packet reaches SW1, the output ports are searched according to the multicast group ID, and SW1 sends the packet to both upper level and lower level respectively, still using the multicast group ID. (c) Last Step: this multicast packet is passed through SW0, and reaches SW2; finally SW2 makes a last sending process according to the multicast group ID.

To increase the multicast throughput as much as possible, HPP Switch allows output ports of different multicast group to send packets simultaneously, on condition that no output port contention occurs. Sending multiple multicast packets simultaneously may lead to deadlock [15]. HPP Switch adopts the RBB to solve this problem. RBB is a global resource register, which take records on currently available output ports of the switch. When a multicast packet request is to be sent, it cannot get permission until all output ports it requests for are guaranteed to be available. If more than one port has got permission, arbitration will be performed by the order of their priority. RBB will remove the destination ports which the packet passed arbitration requests, and after transmission accomplished, RBB will extend the released port list. Therefore, the operation to send one multicast packet to the requested output port becomes an atomic one and central controlled, and the deadlock is prevented.

2.4. Barrier Communication

Barrier is used for global synchronous point. Reducing the barrier operation overhead will significantly improve the performance of the HPC system.

HPP Switch adopts tree-based barrier scheme. In this approach, the barrier is composed of two phases as shown in Figure 6: (a) *combine phase* and (b) *distribution phase*. In the combine phase, each switch node in the barrier tree receives combine packets from its child nodes and forwards a combine packet to its parent node. In the distribution phase, each switch node in the barrier tree receives a distribution packet from its parents and forwards distribution packets to its child nodes.

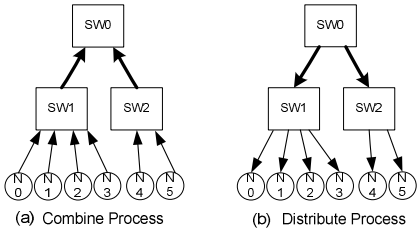


Figure 6: **Barrier Transaction of HPP Switch**

There are two types of reliability problems in the barrier operation. Firstly, the barrier packets may get lost or corrupted in transit. To solve this problem, *ACK/Timeout* and DUN/Timeout mechanism are used to ensure the reliable transfer of barrier packets. In the combine phase, if the node, which has sent out a combine packet, does not receive ACK packet after timeout, it will resend a combine packet and reset the time counter. After successful receiving combine ACK packet, the node starts DUN timer and waits for distribution packet. If timeout, the node will send a DUN

packet to its parent; then the parent will resend a distribution packet again.

The second problem is the consecutive barrier operations executed by the same barrier group may be out of order. During the barrier period, some processes, which have completed the distribution phase, may start next barrier operation before the whole parallel application achieves the barrier point. Resent mechanism makes the problem even more serious. To solve this problem, one-bit sequence number is used to distinguish the consecutive barrier operation.

3. Performance Evaluation

In this section, the behavior of HPP Switch is evaluated in different scenarios. A simulator of HPP Switch is developed at the register transfer level with cycle accuracy. First, the simulator is described. Second, the latency and throughput are evaluated by varying key parameters, such as number of virtual channels and input buffer size. Some hints are given too. Third, the interaction between unicast and barrier is studied. Last, the comparisons of the scalability of multicast and barrier operations with others are given.

3.1. Simulator

The architecture and design of a 16-port HPP Switch are simulated. Each port is connected with an injector and a monitor. The injector can inject unicast and barrier packets concurrently. The injector is able to inject unicast packets with uniform distribution of packet destinations and virtual channels in a given rate, in order to evaluate the full range of traffic, from low load to saturation. The monitor receives the packets passed out from the simulator, and makes the statistics.

The process of warming up is necessary for the simulator to arrive the steady state. Once in the steady state, the performance statistics is triggered. To guarantee the simulation precision, each simulation experiment lasts tens to hundreds of million cycles.

3.2. Unicast Throughput

The impact of number of virtual channel and input buffer size on unicast throughput is analyzed.

3.2.1. Number of Virtual Channels

Head-of-line (HOL) blocking in Unicast has greatly influence on the throughput of Input-Queued of switch. By adding certain numbers of virtual channel can greatly improved unicast throughput [19], but the number of VC is related with the switch port number. Sancho [19] indicates that when the switch port is 8, 2 virtual channels are good enough to get high throughput. Elite [3], QsNet 8 port switch, also has 2 virtual channels. To get the suit number of virtual channels for 16-port switch, the experiments are done. The throughput is shown in Figure 7 with 256B packet.

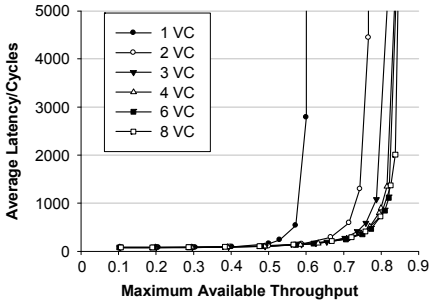


Figure 7: **Throughput with different number of VCs**

Increasing the number of VCs can significantly increase the maximum throughput. As Figure 7 shows, being compared with 1 VC, the network throughput with 2 VCs can increase by a factor of 1.27, 1.40 times with 3 VCs, 1.42 times with 4 VCs, and 1.44 with 8VCs. Because more VCs are used, more separate data paths can be used to route packets from source port to destination port, thus HOL effect is reduced greatly. However, more than 4 VCs make little contributes to the network throughput. The experiments with the longer and shorter packets get the similar results. The reason is that there are other factors causing performance reduction such as output contentions and flow control overhead, which can not be solved by adding more VCs. Therefore, HPP Switch chooses 3 VCs configuration.

3.2.2 Buffer Size

Buffer size of input queue also has distinct effect on throughput of switch. In this experiment, 3 VCs is used as the configuration, and the injector offers the maximum workload. In Figure 8, the throughput varia-

tion, when the buffer size and packet length change, is shown. The throughputs of 8~64B packet length get the saturation condition in 512B buffer size. The increasing of buffer size cannot improve the maximum throughput. The throughputs of 128~1024B packet length get the saturation in 2KB buffer size. The reason for this is that if the buffer is too small, the performance degradation is main caused by flow control scheme. A lot of cycles have been wasted to wait for flow control packets updating credits information, in the meanwhile, the number of flow control packets increase greatly. Since MTU of HPP interconnection network is 1KB, 4KB buffer size is chosen for HPP Switch.

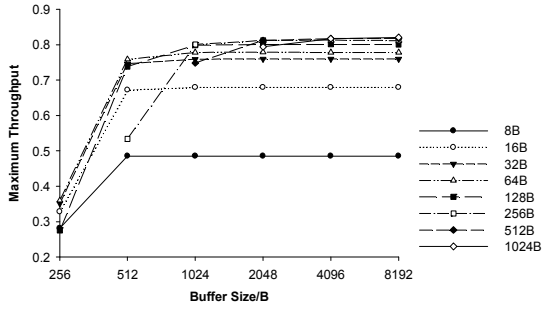


Figure 8: **Throughput with different buffer sizes**

3.3. Interaction between Unicast and Barrier

HPP Switch deals with the unicast packets and barrier packets in the same time, thus the output contention effect will affect the single-stage barrier. In this section, the interaction between unicast and barrier is studied.

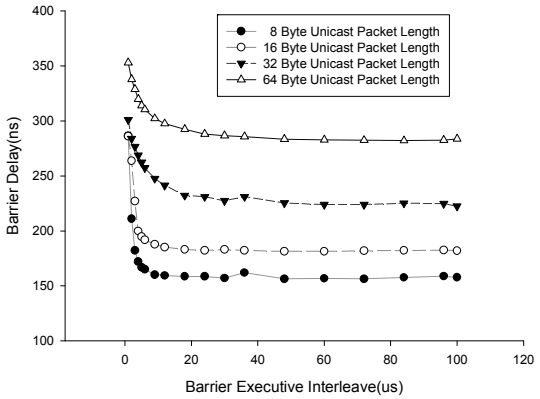
In the experiment, unicast packets, whose destination ports are uniformly covered 16 ports, are generated in random. Sixteen barrier groups, with 15 leaf ports and 1 parent port involved, are all used. The average latency of 16 groups is used to evaluate the interference.

$$T_{avg} = \sum_{i=0}^{15} \overline{T(bi)}/16$$

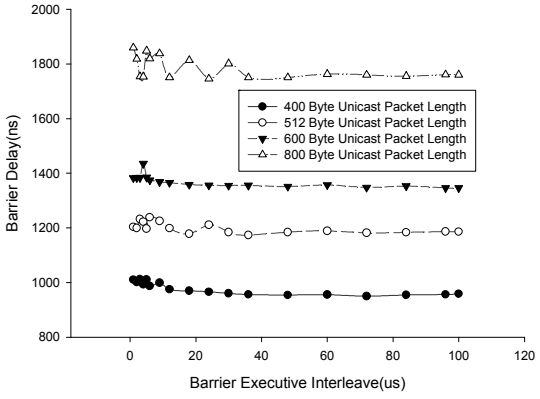
$$\overline{T(bi)} = \sum_{j=1}^{n} (T(cj)+T(dj)) / n$$

T(bi) is the average value of delays of ith barrier group. T(cj) is the delay of Barrier Combine time. T(dj) is the delay of Barrier Distribute time. N is the times of barrier operation.

Under the interference of the maximum unicast workload, the barrier latency is shown in Figure 9 by changing the interval between two consecutive barrier operations. The barrier latency is coming down to convergence with the increase of interval. When the interval is above 50us, the latencies stay steady. The interval in real applications is usually several milliseconds. Thus, the convergent value can be treated as the real single-stage barrier delay.



(a)



(b)

Figure 9: **Barrier latency with different intervals of (a) short packet or (b) long packet**

However, the convergent value varies with different unicast packet lengths. In Figure 10 it is shown that the barrier latency increases linearly with unicast packet length. That is because the barrier packets have the highest priority in HPP Switch. The barrier packets only need to wait one unicast packet at most. Without the effect of unicast, single-stage barrier delay is 128ns.

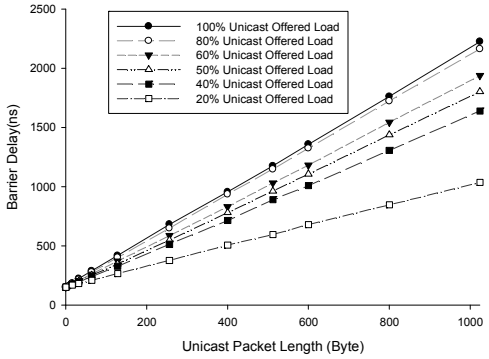


Figure 10: **Barrier delay with unicast packet length**

In the meanwhile, since barrier packet has higher priority than unicast packet, it will affect unicast performance too. The decreasing rate of unicast bandwidth is shown in Figure 11. There is a little impact on unicast, and the mean decreasing rate is only 1.14%. The reason is that the length of barrier packet is very short, only 5 16-bit flits. So, it is not necessary to implement barrier in an independent physical channel.

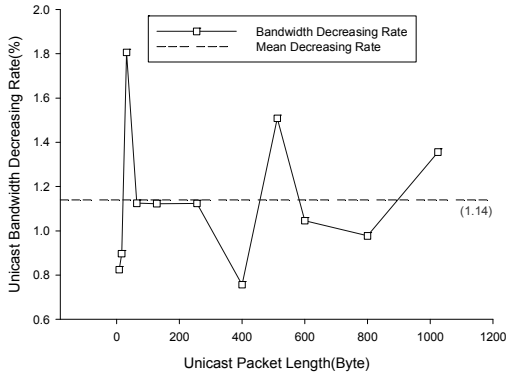


Figure 11: Decreasing rate of unicast bandwidth

3.4. Scalability of Collective Operations

In this section, the scalability of multicast and barrier is analyzed based on LogP [20] model. Assuming: the number of nodes is P; the transmit overhead is Os; the receive overhead Or. So the latency L is $(2\times\lceil\log_8 P\rceil\text{-}1)\times(d+l)+l$. The d, latency of multicast or barrier, can be calculated from HPP Switch design. The l, latency of assuming 8 meters wire, can be calculated by the 5ns/m velocity of signal on wire. For other parameters, such as CPU and NIC overhead, use the value mentioned in [21].

For broadcast operation, the approaches of Sequence, Optimal Spanning Tree BCAST [22], and HPP Switch (HW Mul) are compared. For barrier operation, Central Counter (Cen Cnt), Pair Wise (PW) [23] and HPP Switch (HW Barrier) are compared. The result is shown in Figure 12.

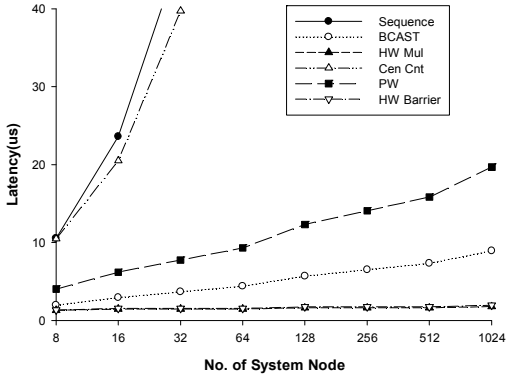


Figure12: **Comparison of Multicast and Barrier**

For other approaches, the latency scales up when system scales up. The latency of HPP Switch is approximately constant while system scales from 8 to 1024. The latencies of multicast and barrier are under 2us for 1024 nodes.

4. ASIC Implementation

The trade-offs and challenges in ASIC implementation of HPP Switch are discussed.

HPP Switch is implemented in 0.13um CMOS standard-cell ASIC technology with targeting system frequency of 312.5MHz. The silicon area is 64 mm^2. HPP Switch has about 4M logic gates, and the power consumption is 5.6W. The design adopts SMIC Logic013GHVT Process 1.2-Volt SAGE-X v2.0 Standard Cell Library with 8 layers of copper. Memory module is generated by Artisan Dual Port SRAM Generator, and small 17x16 FIFO was built by Register File since the larger overhead of BIST (Built-In Self Test) and other test circuit for memory.

The core area (excluding IO cells) of HPP Switch chip is 29.2mm^2. The Dual Port RAM's area is 15.78mm^2, which occupies most of the core area--up to 54%. This is the reason why the receive module takes the majority of chip area. These RAMs should be given a proper floorplan to facilitate the routing and to minimize the interconnect wire. Because the transmit module contains FIFOs which are implemented by

registers, so it takes the largest occupation of resources of the registers. The area of Barrier module is only 2.1% of core chip area. Area, logical cells, register cells and macro of module of HPP Switch chip are showed in Table2.

Table 2: **Resource statistics of HPP Switch**

Module	Logical	Register	RAM	Area(um^2)
Transmit	56,048	**30,432**	0	2,043,616
Receive	**57,360**	18,112	64	**17,184,160**
Barrier	24,141	8,951	0	625,887
Multicast XBar	25,717	2,856	0	502,925
Unicast XBar	40,300	8,224	0	822,608
Management	1,054	125	0	25,004
Total	213,623	68795	64	29,160,000

HPP Switch has 16 ports, each of which can simultaneously transmit and receive 16-bit data. Including control signals, the total number of the IOs in the chip is 646. In addition, it has more than 200 power IOs. So the chip has more than 900 IOs in total, with a Pad-Limit structure. Although we used double padding, the chip size is still up to 8x8 mm^2. The large area of the chip makes the interconnect wire longer and wire delay larger. As being more and more significant, wire delay makes it difficult for timing closure. In the final result, the longest wire length is 3.012mm which is long enough to have some troubles in timing closure. In the design, the method of inserting buffer to cut the long wire and increasing the width of clock wire is used, in order to meet the timing constrain.

The chip has too many signals which arrive simultaneously; the crosstalk noise problem has to be considered. The long interconnect wires has enhanced the effect of crosstalk noise too. So, the noise problem has to be dealt with during the physical design flow. The measure such as setting minimum transition time and enlarging the spacing of clock wire is taken to reduce the noise. After the physical design, the noise problem and optimized the existing violations are checked. This is an iteration work for timing closure and noise.

The final layout is illustrated by Figure 13. The outer ring of the core is RX module, which directly connects to the SRAM outside of it. The inner ring is Tx module. In the middle of the RX and TX ring are 3 stand-alone crossbars and Barrier Module. Because these four modules connect to all 16 ports, they are in the center of the chip to minimize the interconnect wire to the port.

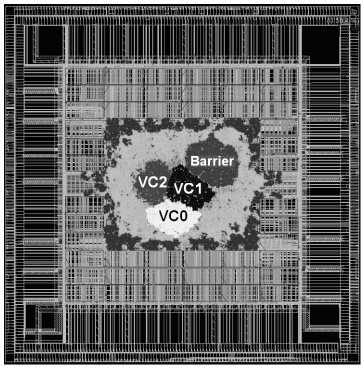


Figure 13: **Layout of HPP Switch**

5. Conclusion

As the core component of HPP interconnect network, HPP Switch provides 16 ports with 38.4ns zero-load latency, 160Gbps aggregated bandwidth, 16 reliable barrier groups and 16 multicast groups. The simulator proves that 3 virtual channels are best performance-cost choice for 16 input-buffered switches, and 4KB input buffer is sufficient enough for 1KB MTU packet to achieve the highest unicast throughput. Because the barrier packet is quite small and the interval between barrier operations is several milliseconds, the interference between barrier and unicast is minimal. It is not worth to use separate physical channel to implement barrier operation. HPP Switch is a low cost approach for high performance switch of scalable cluster system.

Future work includes more experiments on the ASIC chip and on HPP interconnection network to validate the performance of HPP Switch. The next generation HPP Switch for the real peta-scale HPC will double the port number and the port bandwidth.

Acknowledgements

The authors would like to express the gratitude to Dr. Xuejun An, Panyong Zhang and Dingshan You for their tremendous help. This research is supported by the National High Technology Research and Development Program (863 Program) under the contract No. 2006AA01A102 and NSF China under the contract No. 60633040.

Reference

[1] S. Coll, "A Strategy for Efficient and Scalable Collective Communication in the Quadrics Network", PhD

Thesis. University Politécnica de Valencia, Jul. 2005

[2] R. Riesen, “Communication patterns”, In Workshop on Communication Architecture for Clusters (CAC’06), Rhodes Island, Greece, Apr. 2006.

[3] Quadrics, “QsNetII: A network for Supercomputing Applications”, http://www.quadrics.com.

[4] S. Scott, D. Abts, J. Kim, and W. J. Dally, “The BlackWidow High-radix Clos Network”, In Proc. of the 33rd Annual International Symp. on Computer Architecture (ISCA ’06), Boston, MA, Jun. 2006. pp. 16–28,

[5] M. J. E. Lee, W. J. Dally, R. Farjad-Rad, H.T. Ng, R. Senthinathan, J. H. Edmondson, and J. Poulton, “CMOS High-Speed I/Os – Present and Future”. In International Conf. on Computer Design, San Jose, CA, 2003, pp. 454–461.

[6] J. Liu, A.R. Mamidala and D.K. Panda, “Fast and Scalable MPI-Level Broadcast using Infiniband’s Hardware Multicast Support” in Proc. of International Parallel and Distributed Processing Symposium (IPDPS ’04), New Mexico, Apr. 2004

[7] F. Petrini, J. Fernandez, E. Frachtenberg, and S. Coll, “Scalable Collective Communication on the ASCI Q Machine”, in Proc. Symp. High Performance Interconnects (HotI '03), Aug. 2003.

[8] http://www.myricom.com/myrinet/overview/

[9] J. Flich, P. López, M.P. Malumbres, and J. Duato, “Boosting the Performance of Myrinet Networks” IEEE Trans. Parallel and Distributed Systems, vol. 13, no. 7, Jul. 2002.

[10] http://www.mellanox.com/pdf/products/silicon/InfiniScaleIII.pdf

[11] http://www-unix.mcs.anl.gov/mpi/

[12] http://upc.lbl.gov/

[13] C. Minkenberg, “On packet switch design”, PhD Thesis, Eindhoven University of Technology, 2001.

[14] Li-Shiuan Peh, “Flow Control and Micro-Architectural Mechanisms for Extending the Performance of Interconnection Networks”, PhD Thesis, Stanford University, Aug. 2001.

[15] V. Varavithya and P. Mohapatra, “Asynchronous Tree-Based Multicasting in Wormhole-Switched MINs”, IEEE Trans. Parallel and Distributed Systems, 1999, vol. 10, no. 11, pp. 1159-1178.

[16] W. Yu, D. Buntinas, D.K. Panda, “High Performance and Reliable NIC-Based Multicast over Myrinet/GM-2”, in Proc. of the International Conference on Parallel Processing (ICPP’03), Oct. 2003

[17] Jiuxing Liu, Amith R Mamidala and Dhabaleswar K Panda. Fast and Scalable MPI-Level Broadcast using InfiniBand’s Hardware Multicast Support in proceedings of International Parallel and Distributed Processing Symposium, IPDPS 2004

[18] S. Coll, J. Duato, F. Petrini and F. J. Mora. “Scalable Hardware-Based Multicast Trees”. in Proc. of Supercomputing, Phoenix, Nov. 2003

[19] J. C. Sancho, J. Flich, A. Robles, P. Lopez, and J. Duato, “Analyzing the Influence of Virtual Lanes on the Performance of Infiniband Networks” in Proc. of the 16th International Parallel and Distributed Processing, 2002.

[20] D. Culler, R. Karp and D. Patterson, “LogP: Towards a realistic model of parallel computation”, in Proc. of the 4th ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming, New York, May1993, pp. 1-12.

[21] P. Zhang, C. Ma, J. Ma, Q. Li and D. Meng, “HPPNET: A Novel Network for HPC and Its Implication for Communication Software”, in International Workshop on Communication Architecture for Clusters (CAC), Apr. 2008

[22] A. Bar-Noy and S. Kipnis, “Designing Broadcasting Algorithms in the Postal Model for Message-Passing Systems”, in Proc. Fourth Ann. ACM Symp. Parallel Algorithms and Architectures (SPAA '92), Jun. 1992, pp. 13-22.

[23] D. Hensgen , R. Finkel and U. Manber, “Two algorithms for barrier synchronization”, International Journal of Parallel Programming, Feb. 1988, v.17 n.1, p.1-17.