Title:

Implementation of High Performance Buffer Manager for an Advanced Input-Queued Switch Fabric

Authors:

Gab Joong Jeong
Kyongju University, Hyohyundong, Kyongju, 780-712 KOREA
+82-54-770-5325
gjjeong@kyongju.ac.kr

Jung-Hee Lee
IP Switching Team, ETRI, 161 Gajeongdong Yuseonggu, Daejeon, 305-350 KOREA
+82-42-860-6747
hilee@etri.re.kr

Bhum-Cheol Lee
IP Switching Team, ETRI, 161 Gajeongdong Yuseonggu, Daejeon, 305-350 KOREA
+82-42-860-5671
bclee@etri.re.kr

Contact Author:

Gab Joong Jeong
Kyongju University, Hyohyundong, Kyongju, 780-712 KOREA
+82-54-770-5325 (FAX: +82-54-748-2895)
gjjeong@kyongju.ac.kr

Paper Category Number:

(First) Digital Circuits
(Second) Communication Circuits

Name of the Speaker:

Gab Joong Jeong

Implementation of High Performance Buffer Manager for an Advanced Input-Queued Switch Fabric

Gab Joong Jeong
Kyongju University
Hyohyundong, Kyongju
780-712 KOREA
+82-54-770-5325,
gjjeong@kyongju.ac.kr

Jung-Hee Lee
IP Switching Team, ETRI
161 Gajeongdong Yuseonggu
Daejeon, 305-350 KOREA
+82-42-860-6747
hilee@etri.re.kr

Bhum-Cheol Lee
IP Switching Team, ETRI
161 Gajeongdong Yuseonggu
Daejeon, 305-350 KOREA
+82-42-860-5671
bclee@etri.re.kr

ABSTRACT
In this paper, we describe the implementation of high performance buffer manager that is used in an advanced input-queued switch fabric. It provides the architecture of buffer manager and illustrates the design of ingress and egress buffer for the high-speed switch fabric, which uses a multi-gigabit serial crossbar structure. The designed buffer manager provides wire-speed cell/packet routing with low cost and tolerates the transmission pipeline latency of request and grant data. The buffer manager is implemented in a FPGA chip and supports the speed of OC-48c, 2.5Gbps per port.

Keywords
ATM, Latency, Pipeline, Switch Fabric, Wire-speed Routing.

1. INTRODUCTION
Input-queued packet switches have been extensively studied in recent years to resolve the output contention problem that occurs when multiple packets destined for the same output arrive simultaneously [1-4]. The loser of an output contention has to wait for the next arbitration in an input buffer. The input buffer needs to communicate request and grant data with an arbiter, which determines the winners for the next packet transmission. The transmission time of request and grant data is more serious when the switch system uses a high-speed serial transmission technology as point-to-point communication line between an input buffer and the central arbiter for the implementation of a large distributed switching system.

The data transmission latency should be considered in a pipelined operation of buffer manager and the high-speed logic design of an arbiter. The transmission latency can also occur in cell reformatting to transmit arbitration information if the switch system uses the in-band transmission of the request and grant data. Since our research focuses on solving the problem described above, we propose a new way to improve the performance of an input-queued switch system and to allow the data transmission latency between the input buffers and the central arbiter. Our approach uses the pipelined operation of all functions: the request data transmission, the arbitration, and the grant data transmission. The switch system of our scheme uses the full time slot for each operation.

In this paper, we introduce the design and implementation of a high performance buffer manager for ingress and egress buffer of an advanced input-queued ATM switch, which has transmission latency of arbitration information and a pipelined approach to improve the throughput [5]. The proposed buffer manager as an input buffer of an input-queued ATM switch provides wire-speed pipelined queue management that has dynamically allocated virtual output queues [6]. The dynamic allocation of queues minimizes the amount of packet memory in an input buffer. The designed buffer manager adopts a novel method to manage the transmission latency between input buffers and a central arbiter, which is based on request shifting. It has been implemented in FPGAs.

2. ADVANCED SWITCH ARCHITECTURE
The proposed switch architecture is depicted in Figure 1. For the pipelined operation of data transmission and arbitration, we add the request first-in-first-out (FIFO) buffer for each virtual output queue (VOQ) in each input buffer. The RF buffer stores a request bit of a cell. We then insert a request FIFO (RF) buffer at a central arbiter, which stores the request data, for the VOQ of each input buffer. The switch architecture allows for the transmitting of consecutive request data for each consecutive new head cell of each VOQ at each input buffer, and it allows for the transmitting of the consecutive grant data for each input buffer, despite the existence of transmission latency.

When a VOQ has one or more queued cells, the queue is considered to generate a request signal in the current time slot. Then, the request signal is generated only if the first element of the RF buffer of the queue is not occupied by a previous request. When the first element of the RF buffer is occupied, there are no new requests for the queued cells, as there are, for example, in a blocked state caused by a backpressure. If the request signal of the queue is generated, the length of the queue is decreased by one in the RF buffer controller and the buffer manager stores the request signal of the last element of the RF buffer controller, after shifting each stored request data in the RF buffer.

When the data requested is validated, the grant signal is set to high, and the input buffer transmits an output cell using the head cell address in the VOQ selected by the output port number, which is transmitted from the central arbiter with the grant signal. While the selected output cell is read from the cell buffer and transmitted to the crossbar switch, the address of the outgoing cell is restored to the idle address queue, and the earliest request

signal in the selected RF buffer is cleared. A simple leading-one detection circuit is used to search the earliest request signal in the RF buffer.

The central arbiter receives the request signals of VOQs from each input buffer. Then, the request signal is stored in the last element of the RF buffer in the central arbiter, after shifting each stored data in the selected RF buffer only if the received request signal of each VOQ is high. The input request vector of the arbitration logic block is generated in a two-dimensional request matrix, using the stored request data in the RF buffer. Each request element of the two-dimensional request matrix is from each RF buffer. Only if the RF buffer of each VOQ has at least one request signal, is the request vector element of the VOQ of the two-dimensional request matrix assigned as high, and is considered by the arbitration logic as the request of the VOQ. The arbitration logic arbitrates all requests from the input buffers, and it generates the grant data for each input port. According to the grant data, only if the output port of an input buffer is selected, is the earliest request signal in the RF buffer of the central arbiter set to low before the next arbitration.

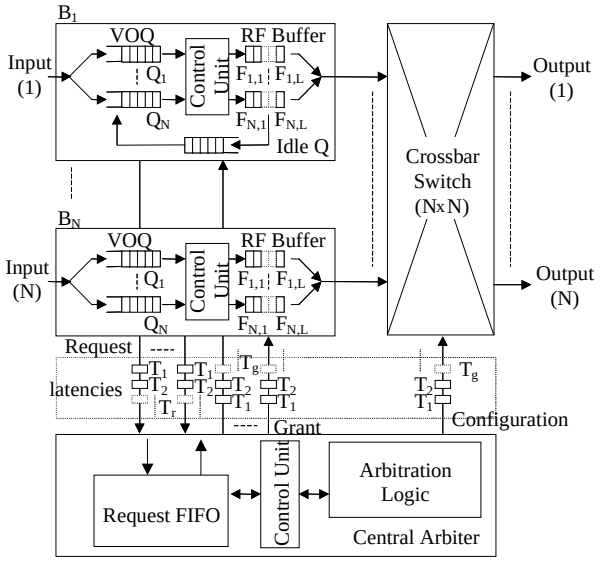

Fig. 1 Advanced input-queued switch architecture.

3. PROPOSED BUFFER MANAGER

The buffer manager in an advanced input-queued ATM switch controls VOQs at each input port of the switch and output queues for traffic management. Figure 2 illustrates the pipelined buffer manager for ingress and egress buffers. We explain mainly the architecture of the buffer manager for the input queue management. The operations are divided into incoming cell writing, outgoing cell reading, policing, and request control for arbitration latency. Each operation has to be connected in a pipelined manner for wire-speed routing with low cost. The input queue manager consists of a VOQ and idle queue (IDQ) modules, write pointer manager (WPM), read pointer manager (RPM), and request first-in-first-out controller (RFC).

The RFC has a request first-in-first-out (FIFO) register for each VOQ and communicates with a central arbiter. It controls request generation and deletion according to the VOQ and request FIFO status. The RFC generates a valid request signal for each VOQ that has one or more queued cells, when the first element of the request FIFO register of the VOQ is not occupied with a valid request. It then stores the request signal in the request FIFO register after shifting existing data in the request FIFO, and decreases the VOQ length by 1. The VOQ length in the RFC is smaller than the VOQ length in the policing module (PM) by the number of valid request signals in the request FIFO. If the VOQ length in the RFC is zero and the first element of the request FIFO of the VOQ is not occupied by a valid request, the RFC generates an invalid request signal and just shifts the existing data in the request FIFO. When a granted output port number arrives at the RFC from the central arbiter, the RFC deletes the oldest request signal in the request FIFO of the granted output port, and it propagates the granted output port number to the outgoing cell reader (OCR). Figure 3 shows the pending request control algorithm in each RFC of the buffer manager.

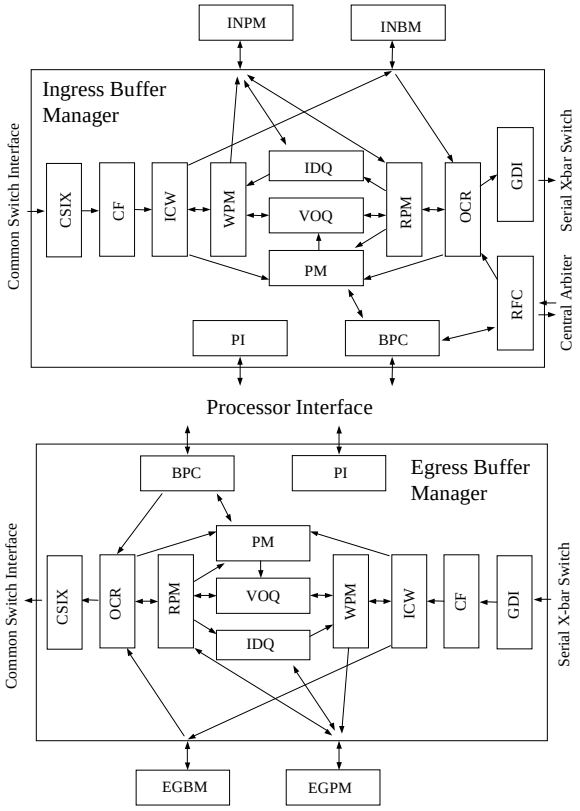

Fig. 2 Pipelined ingress/egress buffer manager.

The OCR propagates the granted output port number to the RPM and PM, and reads an outgoing cell from the ingress buffer memory (INBM) using an output cell address that comes from the RPM. The RPM updates the currently selected port queue of the VOQ module with the next address extracted from the next cell pointer information in the ingress pointer memory (INPM). The RPM stores the current outgoing cell address in the next destination port queue of multicast leaves if the outgoing cell is a multicast cell. A new arriving cell is stored in the INBM by the incoming cell writer (ICW) after receiving a new cell writing address from the WPM. The WPM updates the destination output port queue of the current incoming cell in the VOQ module with the new cell writing address. The RFC aggregates the new cell

arriving and multicast cell stitching information from the PM through the backpressure controller (BPC). Figure 4 shows entire data paths to manage all VOQs and an idle queue that fully share the INPM, a single dual-port synchronous SRAM.

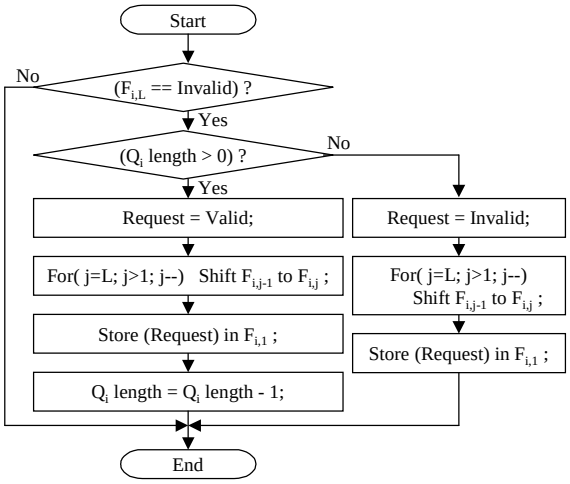

Fig. 3 Pending request control algorithm in the request FIFO controller (RFC).

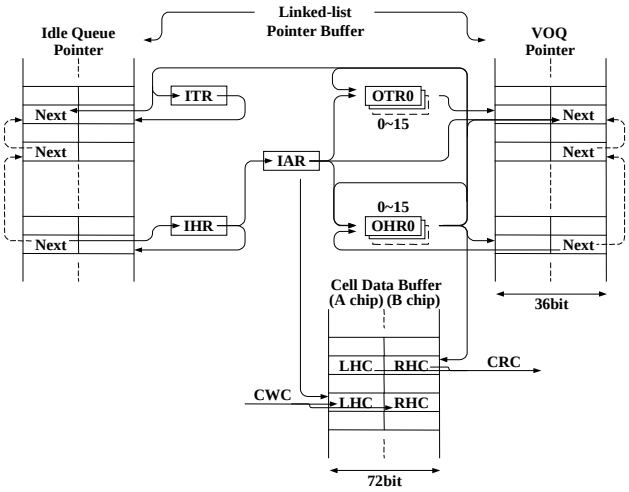

Fig. 4 Entire data path for the pointer movement with output leaf multicasting.

4. IMPLEMENTATION

Every small block of the buffer manager is pipelined by pipeline register and the operation is independent of each adjacent block. We illustrate the micro-operation of the read pointer manager (RPM) to show how to operate the blocks each other. The RPM block eliminates a head cell address in the VOQ selected by the central arbiter for each cell slot and adds the read cell address in the idle queue or it attaches the read cell address in the next destination output queue if the outgoing cell is a multicast cell. The operation steps of the RPM are shown below.

The first cell slot:

Step a) takes a granted output port identification from the OCR.

Step b) gives the ID of step (a) to VOQ block.

Step c) takes a head cell address from the VOQ block.

Step d) gives the cell address of step (c) to the OCR and INPM.

Step e) takes the next cell address and multicast bitmap from the INPM.

Step f) gives the next cell address of step (e) to the VOQ block and the cell address of step (c) to the IDQ if it is not a multicast cell.

The second cell slot:

Step g) gives the next destination output leaf ID to the PM if the read cell in step (f) is a multicast cell.

Step h) gives the next destination output leaf ID and the cell address in step (c) to VOQ block.

Step i) takes the tail address of the queued cell in the selected destination in step (h) from the VOQ block.

Step j) gives the tail address in step (i) and the cell address in step (c) to INPM for queuing the next multicast output cell.

Step k) gives the cell address in step (c) and the updated multicast bitmap of the cell to INPM for eliminating current destination output port in its output leaf.

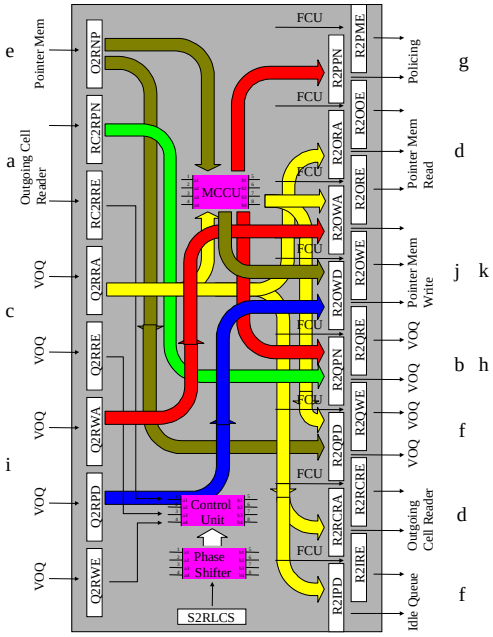

Fig. 5 Pipelined process steps in the RPM block.

Since the stepwise operation of the RPM described above occurs at every cell slot, the reading of a current outgoing cell through the first 6 steps is processed with the writing of a multicast cell in the next output leaf of the previous output cell through the last 5 steps in a single cell slot. Every stepwise operation of all blocks in the buffer manager is scheduled its own pipeline clock to evade any collision of pipelined processes. Figure 5 illustrates the RPM block and the pipelined process steps. Main pipeline stages consist of four stages for processing and one stage for synchronizing, and each pipeline stage has two clock cycles for

72-bit pointer data processing using the data bus width of 36-bit between the buffer manager and the external pointer memory chip. Figure 6 shows the timing diagram of the INPM access and the four pipeline stages for data processing. The arrows in the timing diagram show the concatenation operation.

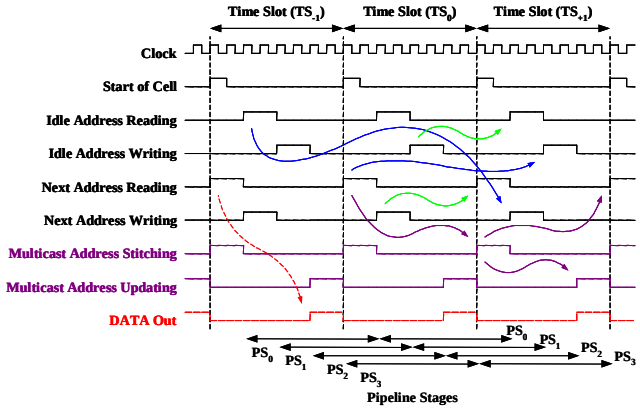

Fig. 6 Pointer memory r/w timing and the movement of data.

The proposed buffer manager has been designed and implemented in a field programmable gate array (FPGA). Table 1 summarizes the characteristics of the designed buffer manager, which has been verified in a core ATM switch system that has a multi-gigabit serial crossbar structure, a 62.5MHz operating frequency, and OC-48c speed per port. The designed buffer manager has been used as an egress cell buffer manager as well as an ingress cell buffer manager. It supports common switch interface (CSIX) format to communicate with an ingress and egress port processor. Figure 7 shows the photograph of the ingress and egress buffer board using the proposed buffer manager, which is implemented in a high performance backbone switch of the Internet.

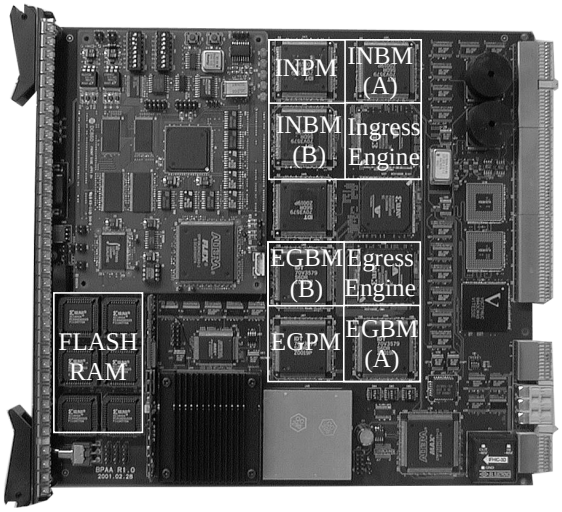

Fig. 7 Photograph of the designed ingress and egress buffer board.

Table 1. Characteristics of the ingress buffer manager

FPGA device	XCV1000E-6	INBM data bus width	72bit
package	900pin FGA	buffered cell size	640bit
I/O	LVTTL	INPM data bus width	36bit
used I/O	629pin	managing pointer packet size	64bit
estimated power dissipation	13.2W	experimental operating freq.	62.5MHz
equivalent gate count	380K	experimental port speed	OC-48c (2.5G)
CLB usage	40%	experimental switch size	16x16
max. operating freq.	77MHz	experimental one cell time	160ns
max. bit rate per port	5Gbit/s	aggregated switch throughput	40Gbit/s

5. CONCLUSION

In this paper, we proposed and implemented a new architecture of the VOQ buffer manager for an advanced input-queued ATM switch. The proposed buffer manager provides wire-speed routing at low cost using a pipelined buffer management. In the design of the proposed VOQ manager, we used a novel request shifting method to manage the transmission latency of request and grant data between input buffers and central arbiter. The wire-speed pipelined VOQ management has been implemented in a FPGA and has been tested for all functions in an internet backbone switch system which has 16x16 switch size, 2.5Gbit/s per port speed, and 40Gbit/s aggregated switching capacity.

6. ACKNOWLEDGMENTS

The authors would like to gratefully acknowledge the contributions of the members of the IP Switching Team, Electronics and Telecommunications Research Institute, especially J. M. Sohn, E. C. Choi, and B. S. Kim.

7. REFERENCES

[1] H. Obara, S. Okamoto, and Y. Hamazumi, "Input and output queueing ATM switch architecture with spatial and temporal slot reservation control," *Electron. Lett.*, vol. 28, no. 1, pp. 22-24, Jan. 1992.

[2] N. McKewon, P. Varaiya, and J. Walrand, "Scheduling cells in an input queued switch," *Electron. Lett.*, vol. 29, no. 25, pp. 2174-2175, 1993.

[3] R. O. LaMaire and D. N. Serpanos, "Two-dimensional round-robin schedulers for packet switches with multiple input queues," *IEEE/ACM Trans. Networking*, vol. 2, no. 5, pp. 471-482, 1994.

[4] N. McKeown, “The iSLIP scheduling algorithm for input-queued switches,” *IEEE/ACM Trans. Networking*, vol. 7, no. 2, pp. 188-201, 1999.

[5] G. J. Jeong, J. H. Lee, and B. C. Lee, “An advanced input-queued ATM switch with a pipelined approach to arbitration,” in *Proc. IEEE GLOBECOM*, pp.496-499, Nov. 2000.

[6] G. J. Jeong, J. H. Lee, and B. C. Lee, "Design of pipelined routing engine for input-queued ATM switches," *Electron. Lett.*, vol. 37, no. 2, pp. 137-138, Jan. 2001.