Multi-bank based Switch Architecture with Flexible Scheduled Buffering of Packets

Takayuki Fujii†, Kazuhiko Kobayashi††, Tetsushi Koide†, Hans Juergen Mattausch† and Tetsuo Hironaka††

†Research Center for Nanodevices and Systems, Hiroshima University, 1-4-2, Kagamiyama, Higashi-Hiroshima, 739-8527, Japan
Phone: +81-82-424-6265, FAX: +81-82-424-3499, E-mail: {fujii, koide, hjm}@sxsys.hiroshima-u.ac.jp
††Graduate School of Information Sciences, Hiroshima City University, 3-4-1, Ozuka-Higashi, Asaminami-ku 731-3194, Japan
Phone: +81-82-830-1566, FAX: +81-82-830-1792, E-mail:zuhiko@csys.ce.hiroshima-cu.ac.jp, hironaka@ce.hiroshima-cu.ac.jp

1 Introduction

In recent years, the amount of traffic is increasing rapidly with the spread of broadband [1]. Therefore it is required to expand the data throughput of the whole network. Especially the network switches, which are the connection nodes of the network, need a strong reinforcement of their throughput capability. However, the improvement of the switch performance is difficult because blocking situations are easily generated within the existing switch structures in case of data contention at a given path in the switch fabric. The authors previously addressed the blocking problem which occurs in the switch fabric and the inefficient usage of the switch memory under biased network traffic. A switch architecture using a switch fabric which consists of a multi-port memory with 1-port banks was proposed as a solution and a special focus was given on optimizing the data transfer from the 1-port banks to the output interfaces of the switch [2]. In this paper, we investigate the achievable performance improvement by optimizing the data transfer from the input interfaces of the switch and by introducing 2 ports into the memory banks. The effectiveness of the proposed switch architecture is verified by simulation.

2 Multi-Bank based Switch Architecture

We use a bank structure multi-port memory with distributed crossbar as the shared memory [3, 4]. This architecture, called Hierarchical Multi-port memory Architecture (HMA), realizes high access bandwidth of the shared memory by using fully-parallel multiple ports, and also realizes high area efficiency. The proposed scheduling algorithm which is explained in the next section is used for the bank structure, so that it's possible to assign banks according to output port contention or biased data traffic. It becomes also possible to output packets which can't be transmitted from an input port to an output port by blocking in existing network switches. For example, as shown in Fig. 1, the traffic is biased toward output port 1. Several packets are transmitted from plural banks to output port 1 (ex: Banks 1 and M). By definition of the output order, we can reduce the blocking problem. In addition, in this case, the sub-packets of packet ID 7 are transmitted to Bank M, so the sub-packets of packet ID 9 are able to be transmitted to output port 5 through a different bank without blocking. Thus we can reinforce the throughput capability.

3 Proposed Algorithm with Dynamic Bank Scheduling

Three management informations are necessary to handle each sub-packet: the packet ID, the sub-packet ID and the output-port number. Since 2-port banks are used, scheduling of input and output interface of the shared memory can be performed simultaneously and independently. The proposed scheduling algorithm compares packet ID or output port number of the sub-packet to be transmitted with the equivalent data of the previously transmitted sub-packet. Details of the proposed algorithm are as follows.

Step.1 It relies on the principle that the complete traffic from a given input port to a given output port uses the bank, which had been initially scheduled, as long as this bank doesn't overflow. In Fig. 1(b) the sub-packets of packet ID 4 will be transmitted to Bank 1.

Step.2 In this step the new packet is scheduled to the bank to which the last packet with the same output port has been scheduled, as long as the respective bank is not full. In Fig. 1(b) the sub-packets of packet ID 5 or 6 will be transmitted to output port N or 1.

Step.3 In this step the scheduler searches for an empty bank, which is subsequently used for buffering the traffic from the input port to the respective output port. In Fig. 1(b) the sub-packets of packet ID 9 will be transmitted to output port 5 through an empty bank if there is no bank which stores already sub-packets for output port 5.

4 Simulation Evaluation of Proposed Switch Architecture

The reference architectures chosen for comparison are the existing switch architectures which use shared bus, crossbar, or multi-stage connection network (omega network [5]). The simulation model of the proposed architecture uses shared memory switch fabric and the scheduling algorithm described in Sections 2 and 3. Evaluation criteria are throughput, inner delay time, and packet-loss rate. Simulation conditions are shown in Tables I and II.

With respect to the output ports both balanced traffic (each output port has the same traffic) and unbalanced traffic (one output port has increased traffic) are evaluated. The degree of unbalance (or bias) for one of the output ports is expressed by a factor N (N=1-8). In addition to existing switch structures, a shared-memory architecture which uses 1 port banks is also included in the comparison. The scheduling method for the 1-port-bank-based shared-memory is the same as that described in Section 3. The parameters for the proposed bank-based shared-memory switch architecture are 64 banks and 54 [Kbit] bank capacity, as determined in a preliminary simulation of the optimal bank number.

The comparison results with the existing switch architectures for throughput and packet-loss rate are shown in Fig. 2(a) and 2(b), respectively.

The performance for the proposed architecture significantly exceeds that of the existing conventional switch architectures in every aspect shown in Fig. 2. This is considered to be the case because blocking by contention at the output ports, which is a big problem for conventional switch structures, is successful avoided due to an efficient use of the shared-memory banks. Moreover, when the traffic bias is increased, it turns out that the performance difference between 1-port and 2-port banks increases too. This is,

because the scheduling algorithm is carrying out control in units of the output ports and of the associated banks. Therefore, a data-traffic bias increases the access volume to specific banks. On the other hand, the use of 1 port banks, increases the blocking frequency because input and output interfaces are no longer independent. So it can be expected that performance difference between 1-port and 2-port banks will appear more clearly for strongly biased data traffic.

5 Conclusion

A switch architecture which uses a banked multi-port memory with 2-port banks was proposed as a solution for the blocking problem of existing switches. Simulation results of the proposed switch architecture verified an improvement in throughput under unbiased as well as biased traffic of about 120 % when compared with a switch using a crossbar switch fabric and equal total memory capacity. Future research work will mainly concentrate on the hardware verification of the proposed switch architecture with an LSI design.

REFERENCES

[1] Information and Communications in Japan 2003, Home Pages, http://www.johotsusintokei.soumu.go.jp/whitepaper/eng/WP2003/2003-index.html/, 2003.

[2] K. Kobayashi, et al., IPSJ Tech. Rep. 2004-ARC-156, pp. 37-42, 2004 (in Japanese).

[3] H. J. Mattausch, et al., IEICE Trans. Electron., Vol. E84-C, No.3, pp. 410-417, 2001.

[4] S. Fukae, et al., IEE Elect. Lett., Vol.40, No.2, 101-103, 2004.

[5] H. Amano, "Parallel Computer," Shokodo Co.,Ltd., 1996 (in Japanese).

TABLE I
SIMULATION CONDITION(EXCEPT FOR MEMORY CAPACITY).

|	
Simulation time	200000 [cycle]
Number of input/output ports	32 [port]
Input/Output line width	32 [bit/cycle]
Input load	100 [%]
Packet length	1 12000 [bit] at random
Delay time of routing	0 [cycle]
Wire width	32 [bit]

TABLE II
SIMULATION CONDITION(MEMORY CAPACITY).

Existing switches		
Input buffer capacity per port		
228[Kbit] × 32[port]	=	7296[Kbit]
Output buffer capacity per port		
12[Kbit] × 32[port]	=	384[Kbit]
		Total 7,680[Kbit]
Proposed switch		
Input buffer capacity per port		
120[Kbit] × 32[port]	=	3,840[Kbit]
Output buffer capacity per port		
12[Kbit] × 32[port]	=	384[Kbit]
Total bank capacity	=	3,456[Kbit]
		Total 7,680[Kbit]
(Bank capacity per bank is the value which divides the total bank capacity by the number of banks.)		
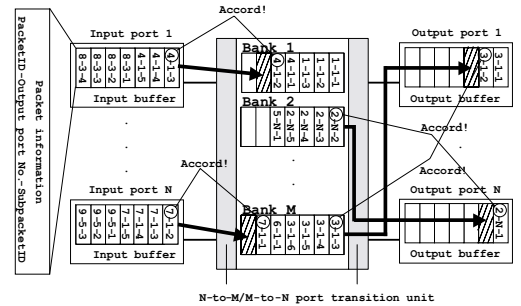
(a) State of proposed switch architecture at time t.
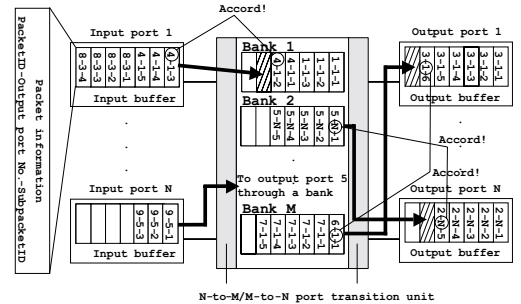
(b) State of proposed switch architecture at time t+4.

Fig. 1. Proposed multi-bank based switch architecture, which shows packets which couldn't been transmitted can output to output port 5 with dynamic bank scheduling. In (a), the sub-packets of packet ID 7 are transmitted to Bank M, so, in (b), the sub-packets of packet ID 9 are able to be transmitted to output port 5 through an empty bank.
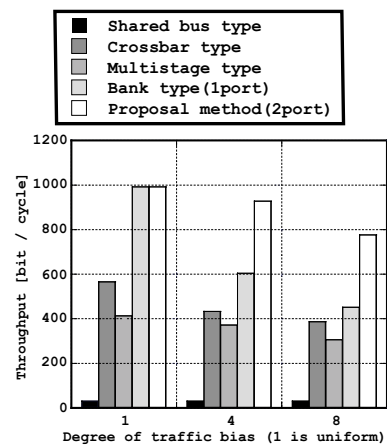
(a) Throughput.
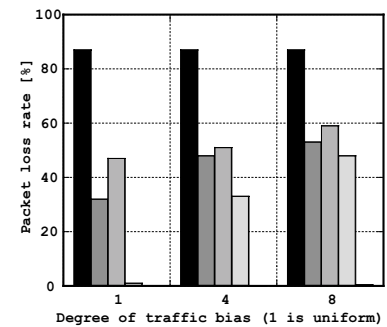
(b) Packet loss rate.

Fig. 2. Comparison of the proposed architecture with existing architectures.

Multi-bank based Switch Architecture with Flexible Scheduled Buffering of Packets

○Takayuki Fujii[1], Kazuhiko Kobayashi[2], Tetsusi Koide[1], Hans Juergen Mattausch[1], Tetsuo Hironaka[2]
[1] Research Center for Nanodevices and Systems, Hiroshima University
[2] Graduate School of Information Sciences, Hiroshima City University

NTIP
Hiroshima University
Nanoelectronics for Tera-Bit Information Processing
Research Center for Nanodevices and Systems, (RCNS)
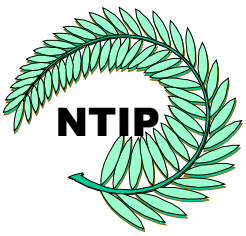System Design and Architecture Research Division

Necessity for improvement in the performance of network switches

Increase in the network traffic by the formation of broadband hardware and services

Necessity for improvement in performance of both network switches and connection lines

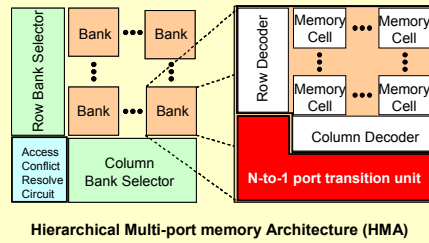### Possibilities for improvement of performance
- High-speed data transmission between switches
- Wide data transmission bandwidth within switches
- High-speed path determination within switches

Packet
Header (160Bit) Data (160~11840bit)

Problems of existing switches
- Blocking by competition at an output port
- Packet loss by inefficient use of memory

Input Buffer / Output Buffer
Switch Fabric
Blocking
Packet loss
①: Packet (Number in circle is port number to be outputted)

Proposal of a network switch structure using a bank-based multi-port memory

Hierarchical Multi-port memory Architecture (HMA)

Hierarchical Multi-port memory Architecture (HMA)
Row Bank Selector / Bank / Row Decoder / Memory Cell
Access Conflict Resolve Circuit / Column Bank Selector / Column Decoder / N-to-1 port transition unit

Input port / Output port
Bank1, Bank2, Bank3, Bank4 ... BankM
Port 1 ... Port N

N-to-M/M-to-N port transition unit
Composition figure of a proposal switch

Advantages using a multi-port memory
- We can access each port simultaneously, so that high access bandwidth is possible.
- Ports share the memory capacity, so that highly efficient use of memory is possible.

Problem
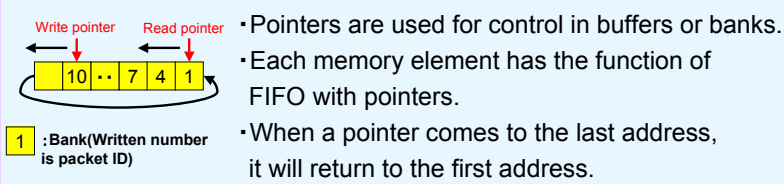- In the conventional N-port SRAM system, area is proportional to the 2nd power of the number of ports.

Hierarchical Multi-port memory Architecture (HMA)
- Regular 2-dimensional arrangement using 1 port memory banks, so HMA has comparatively small area.
- Activation of each bank only on demand, so HMA has comparatively low power consumption.

Further improvement in performance is aimed at by using a scheduling algorithm for bank usage.

Scheduling system

Control method of writing and reading of packets
Write pointer / Read pointer
10 ·· 7 4 1
1 : Bank(Written number is packet ID)
- Pointers are used for control in buffers or banks.
- Each memory element has the function of FIFO with pointers.
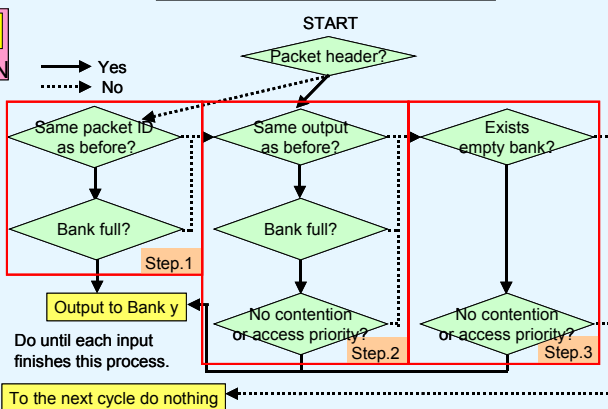- When a pointer comes to the last address, it will return to the first address.

Bank scheduling algorithm for packets

Banks are used as buffers separated from input / output, and can be used freely for intermediate storage.
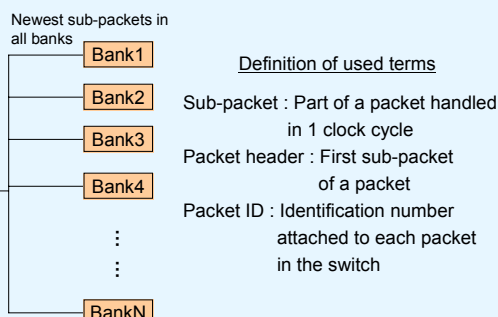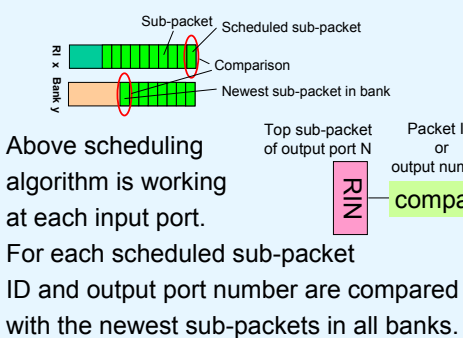
Input port / Output port
Bank1, Bank2, Bank3, Bank4 ... BankM
Port 1, Port 2 ... Port N

N-to-M/M-to-N port transition unit
Write from a RI → / Read to a FI →
Read request to a FI

Packets are divided into sub-packets (32bit).

Characteristics
- According to the data amount towards an output, the necessary bank capacity is assigned individually.
  → Blocking at input is prevented.
- An algorithm is defined for assigning further banks when a certain bank becomes full.
  → Efficiency of memory usage is raised.

Bank scheduling algorithm
START
Packet header? Yes / No
Same packet ID as before? → Bank full? → Output to Bank y Step.1
Same output as before? → Bank full? → No contention or access priority? Step.2
Exists empty bank? → No contention or access priority? Step.3
Do until each input finishes this process.
To the next cycle do nothing

Further explanation of bank scheduling algorithm
Sub-packet / Scheduled sub-packet
RI x Bank y / Comparison / Newest sub-packet in bank
Newest sub-packets in all banks
Bank1, Bank2, Bank3, Bank4 ... BankN
Top sub-packet of output port N
RIN / Packet ID or output number → compare

Above scheduling algorithm is working at each input port.
For each scheduled sub-packet ID and output port number are compared with the newest sub-packets in all banks.

Definition of used terms
Sub-packet : Part of a packet handled in 1 clock cycle
Packet header : First sub-packet of a packet
Packet ID : Identification number attached to each packet in the switch

Simulation

Simulation conditions

Simulation time	200000 [cycle]
Number of input / output ports	32 [port]
Input / output data width	32 [bit / cycle]
Input load	100 [%]
Packet length	1 ~ 12000 [bit] at random
Switch-internal data width	32 [bit]

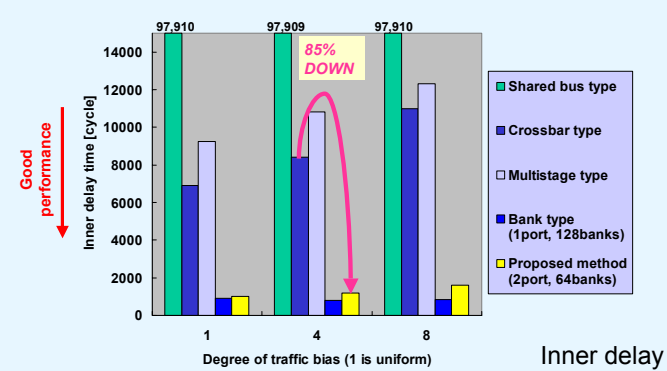Total memory capacity is 7,680 [Kbit] in the compared switch types.

Traffic biases towards one output port by a factor N (N = 1, 2, 4, 8) are analyzed.
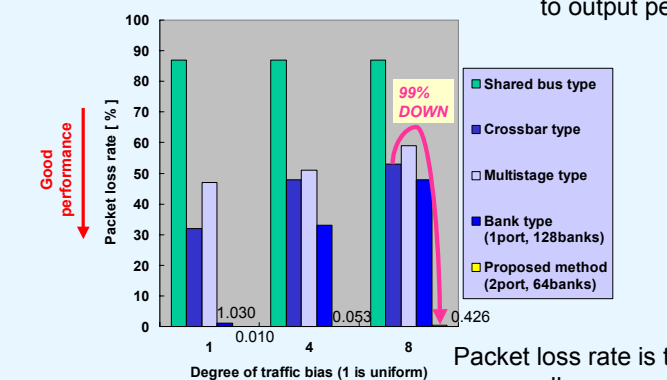
Simulation results

Throughput
- Shared bus type
- Crossbar type
- Multistage type
- Bank type (1port, 128banks)
- Proposed method (2port, 64banks)
- Limit of throughput

120% UP
Degree of traffic bias (1 is uniform)
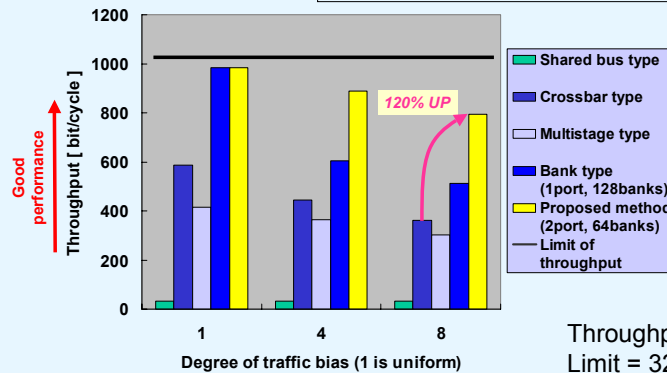
The throughput of the proposed method is higher than for any other method. The possible throughput increase is 120% in comparison to the crossbar type switch.

Throughput is bandwidth per cycle.
Limit = 32 * 32 = 1024 [bit/cycle]

Inner delay time
97,910   97,909   97,910
85% DOWN
- Shared bus type
- Crossbar type
- Multistage type
- Bank type (1port, 128banks)
- Proposed method (2port, 64banks)
Degree of traffic bias (1 is uniform)

The inner delay time of the proposed method is shorter than for all other methods, except for the bank type method with 1port. The possible inner delay time reduction is 85% in comparison to the crossbar type switch.

Inner delay time is the time from input to output per sub-packet.

Packet loss rate
99% DOWN
- Shared bus type
- Crossbar type
- Multistage type
- Bank type (1port, 128banks)
- Proposed method (2port, 64banks)
1.030 0.010 0.053 0.426
Degree of traffic bias (1 is uniform)

The packet loss rate of the proposed method is smaller than for any other method. The possible loss rate reduction is 99% in comparison to the crossbar type switch.

Packet loss rate is the rate number of lost packets among all processed packets.

Conclusion

Result
- A switch architecture using a banked multi-port memory is proposed.
- Simulation evaluation is performed, and the results below are obtained.
  In comparison to a crossbar type switch,
  120% of possible improvement in throughput,
  85% of possible improvement in delay time,
  and 99% of possible improvement in packet loss rate.

Future work
- Hardware implementation is investigated, and it's compared with other architectures.
- Examination of the banked multi-port memory network switch architecture and circuitry, implementing proposed scheduling algorithm.
- Basic performance evaluation by a test chip design.