Design of a Low Power NoC Router using Marching Memory Through type

Ryota Yasudo†, Takahiro Kagami†, Hideharu Amano†, Yasunobu Nakase,
Masashi Watanabe‡, Tsukasa Oishi‡, Toru Shimizu‡, and Tadao Nakamura†
†Keio University ‡ Renesas Electronics
Yokohama, Japan 223-8522 Tokyo, Japan 100-0004
E-mail: marching@am.ics.keio.ac.jp

***Abstract*—Power consumption of Network-on-Chip (NoC) is becoming more important in many core processors. Input buffers utilized in routers consume a significant part of the total power of NoCs. In order to reduce this power consumption, a novel power efficient memory called *Marching Memory Through type (MMTH)* is introduced. By connecting transparent latches in tandem, MMTH achieves high speed operation with a low power consumption. MMTH, however, requires a certain overhead at read operation, and hence we propose a latency reduction scheme based on the look-ahead routing. The proposed router was designed in Renesas's 40nm process and compared with a standard router using conventional register-based FIFOs in terms of the network performance, application performance, and power consumption. The result of evaluation shows that the proposed router reduces the power consumption by 42.4% on average at 2GHz and the expense of only 0.5-2.0% performance overhead.**

I. Introduction

NoC (Network-on-Chip) [1], [2], [3] is a key component of recent multi-core and many-core CMPs (Chip Multiprocessors) as well as heterogeneous SoCs (Systems-on-a-Chip) [4], and its design is a crucial factor for system performance and cost of the chip. Unlike the traditional bus connected systems, many IP (Intellectual Property) cores on a single chip can be connected through routers which transfer packets. It provides high communication bandwidth, parallelism, and scalability. Since these networks are facing tight delay requirements, prior designs and architecture studies are heavily performance-driven, aiming at lowering network latency.

Since the operational frequency of NoCs reaches a few GHz to achieve the low latency data transfer, the power consumption of NoC sometimes occupies a considerable part of the system; for example, in MIT 16-core RAW CMP [5], Intel 80-core Tera FLOPS processor [6] and Intel 48-core SCC [7], it occupies 36%, 28% and 10% of each total power, respectively. This compels us to review network microarchitecture from a power-driven perspective. Since input buffers provided in the router are the dominant part of the power consumption, introducing a low power FIFO concept is efficient to reduce the total power of the router. Here, a novel power efficient "through type" of the marching memory [8] is proposed and utilized in the router of an NoC.

A novel power efficient buffer memory called Marching Memory Through type (MMTH) is proposed for buffers in routers of NoCs. Marching Memory is a memory with high speed marching of data/information stored in the memory [8]. MMTH consists of transparent latches connected in tandem, and works as a FIFO with high operational frequency yet low consuming power. The problem of MMTH is that it requires some time delay of signals as a read latency. In order to reduce it, a new mechanism based on the look-ahead technique is proposed. A network latency overhead is reduced to only one clock cycle per packet.

The remaining part of this paper is organized as follows: Section II shows the motivation to this work and reviews related work briefly. Section III describes Marching Memory Through type, which is utilized in the input buffers. Section IV describes our proposed router microarchitecture. In Section V, evaluation results about power consumption and performance are reported. Finally, Section VI concludes this paper.

II. Motivation and Related Work

A. Input buffers in routers

Compared with off-chip networks, on-chip networks are cost sensitive, and hence buffers reduction is important [2]. As shown in Figure 1, common router provides several input ports each of which provides packet buffers. Since recent CMPs require several virtual channels for types of packets, a few set of buffers are required for each input port. Buffers are implemented with a set of registers or small 2-port memory. In order to follow recent CMPs operating with a few GHz clock, NoCs are also required to work with the same clock signal. Thus, high speed buffer with registers are utilized rather than 2-port memory.

It is reported that approximately 46% of the power and 15% of area of a router are occupied by the input buffers [9]. Moreover, a leakage power modeling [10] shows that the largest leakage power consumer in a router is input buffers, and the analysis of power consumption [11] clarifies dynamic power of the buffer is also high, and it increases rapidly as the traffic of packets increases. From the above, our design goal is to reduce the power consumption of buffers as well as maximizing the performance of network.

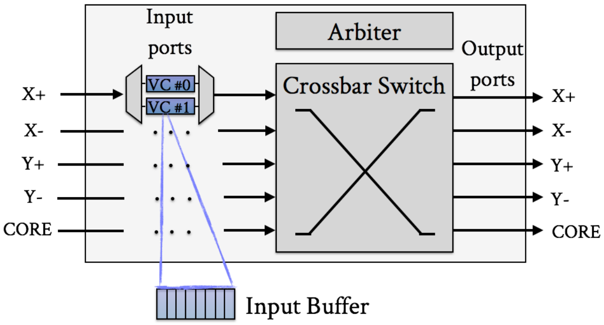


Fig. 1: Router microarchitecture

B. Related Work

Reduction of the power consumed in buffers is tried from various aspects. Buffer-less deflection routers which remove input buffers completely [12] are proposed to reduce router power. In buffer-less schemes, conflicting packets or flits are retransmitted by deflecting them to a free output port. By controlling which flits are deflected, a buffer-less deflection router can ensure that every traffic is eventually delivered. Its routing scheme is based on "hot-potato" routing [13], which is originally proposed for off-chip networks. Large power savings are reported compared with conventional buffered networks. For example, BLESS [12] and CHIPPER [14] reduce power by 39% and 55%, respectively. At high network load, however, deflection routing degrades performance because of frequent deflections caused by many conflicting packets.

More realistic optimization is to decrease the number of input buffers rather than removing them. A centralized buffer router [15] with elastic buffers on the link [16] is proposed to decouple the required buffer space per router. At heavy loads, the centralized buffer is used, and at light loads, it is power gated and bypassed. ViChaR [17] and Reconfigurable routers [18], where the buffer slots are dynamically allocated, are proposed to increase the buffer efficiency in a router. The depth of each buffer word can be reconfigured at run time according to the traffic pattern. Above sophisticated buffer management methods achieves efficient usage of the total buffer based on the observation of a traffic load and the current situation of the router. However, they introduce complexity to both the structure and control of buffers as well as a certain performance degradation.

Our approach is to reduce the power consumption of the buffer by introducing novel buffer memory without using complicated buffer structure and its management. A similar approach is taken by using MRAM technologies. A hybrid buffer design with STT-MRAM (Spin Torque Transfer Magnetic RAM) [19] is proposed to reduce the bottleneck through increasing throughput. Since STT-MRAM is a high-density memory, area budget can be used efficiently. This memory, however, requires long latency and high power consumption in write operations. Therefore a hybrid design of input buffers using both SRAM and STT-MRAM is proposed to hide the long write latency. A migration scheme between SRAM and STT-MRAM is required in this design. In fact, an incoming flit is first written into the SRAM buffer quickly and subsequently migrated to STT-MRAM slowly. Besides, this scheme needs to trigger the migration of a flit on the basis of the estimated network load per VC in the router so as to reduce wasteful power at low loads when STT-MRAM is unnecessary. This method needs both a hybrid design and a migration scheme as well as new process technology, while they are unnecessary for our proposal.

III. Marching Memory Through type

A. Marching Memory

Marching Memory (MM) [8] is invented as a novel memory device that integrates all memory including cache memory and register files into a single unit and can avoid the memory bottleneck [20] by accessing with the same clock cycle of the CPU. MM uses DRAM based memory cell technology but reorganizes the structure that consists of columns and rows of the DRAM. The basic idea is to create a memory structure wherein the data is scheduled to arrive at a fixed physical memory port for immediate use by the processor's functional units. Data are shifted to the CPU synchronized with the clock and come to the processor rather than the CPU searching randomly for the data. Since only data marched to the output port are accessed, high speed access without the energy for accessing bit lines of the DRAM using precharge and sensing can be done.

B. Concept of MMTH

MMTH (Marching Memory Through type) is a novel buffer memory based on an idea from MM that data march along the circuit. It is mainly designed for storing streaming data for media processing and buffers for communication including NoC instead of main memory or cache memory. The structure and target of MMTH are completely different from the original MM, however, it is an application example of MM concept. Although MM is an epoch-making invention, it needs a new DRAM based design technology, and can not be used in CMPs immediately. On the other hand, MMTH is designed so that it can be implemented with the current common CMOS technology unlike the original MM.

The clear distinction of MMTH is that a kind of asynchronous circuit is used. This circuit, which we assume a black box for the present, is sandwiched between input/output ports as shown in Figure 2. By going through the circuit, the written data is moved from the input port to the output port. After we write data to the input port, the data goes through the asynchronous circuit and we can read the data from the output port in order. MMTH is appropriate for input buffers in an NoC router because it is a memory of small capacity with low power at high speed whose data structure is a FIFO structure.

C. Behavior of MMTH

The detailed behavior of MMTH including the operating of an asynchronous circuit is described here. Basically, the

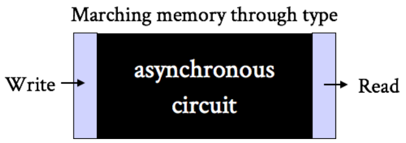
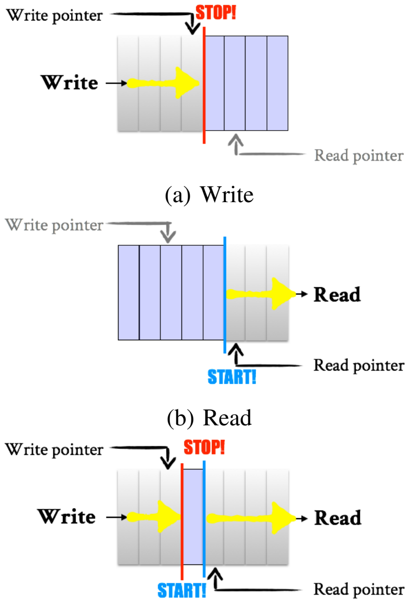


Fig. 2: Overview of Marching Memory Through type



(a) Write

(b) Read

(c) Write & Read

Fig. 3: Behavior of Marching Memory Through type

asynchronous circuit consists of tandem connected transparent latches. A set of latches whose size correspond to the size of a flit or word composes a column. Figure 3 illustrates the operation of MMTH. Each rectangle represents a column. W-pointer and R-pointer control the movement of the data and both pointers initially indicate the right most position.

For write operation, the written data are directly transferred to the position where W-pointer indicates with a clock cycle as shown in Figure 3a, and W-pointer moves one to the left at the next clock cycle. In other words, W-pointer points the terminal column of the transition and the data are written from the input port to the column pointed by W-pointer. For read operation, the data in the column pointed by R-pointer is transferred to the output port as shown in Figure 3b, and then R-pointer moves one to the left. That is to say, R-pointer indicates the starting point of the transition. Both write and read operation can be done in the same clock cycle as shown in Figure 3c. In this way, a FIFO concept is achieved by using an asynchronous circuit.

The positions of the two pointers also determine the state of MMTH. If W-pointer moves to the left most position, the memory becomes full and the data cannot be written anymore. The data pointed out by R-pointer can only be read out. On the contrary, when R-pointer indicates the same position as W-pointer, it becomes empty. When both W-pointer and R-pointer reach the left most position, the MMTH is reset by an external signal and both pointers return to the right most position. However, the read operation requires some delay for transferring data to the output port. The delay depends on the operational clock frequency and size of the memory. For example, when eight-depth MMTH works at 2GHz as we assume, one clock cycle is needed to read the data.

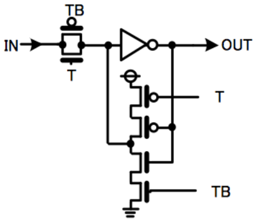


Fig. 4: Memory cell of MMTH

D. *Structure of MMTH*

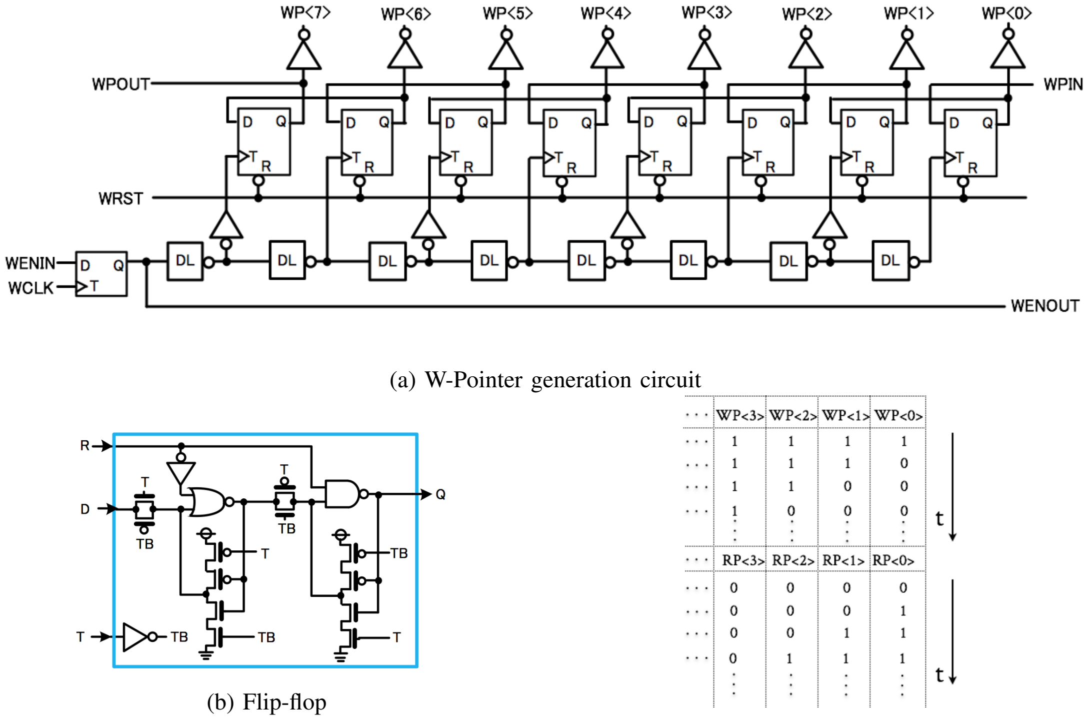
The memory cell of MMTH is composed of a transparent latch with a transmission gate as shown in Figure 4, where T and TB are a control signal and its inverted one, respectively. If T is asserted, the data goes through the memory cell, otherwise the data is stored in the cell. This memory cell structure reduces power and area, since a simple transparent latch is used and a local clock signal is unnecessary.

W-pointer and R-pointer are provided for each cell so as to control T. Figure 5 shows pointer generation circuits. The circuit shown in Figure 5a generates W-pointer. The diagram of Flip-flop used in the circuits is shown in Figure 5b. When WP$<i>$ is asserted, column$<i>$ lets an input data through. Initially, all WPs are asserted by a reset signal WRST, since a written data goes through all columns at first. DL, a delay element, makes a column-by-column wiring delay. When WENIN, a write enable signal, is asserted, WCLK moves at the same speed as the written data. First, when WCLK reaches the right most Flip-flop, WP<0> is negated. When the next data are written, thus, the WENIN is asserted, the WP<1> is negated after a certain delay as the data goes through. The WPs are negated from right to left with the same manner as shown in Figure 5c, and finally when WP<7> is negated, the buffer becomes full.

Almost the same structure is utilized for R-Pointer except the 1s and 0s are inverted as shown in Figure 5c. When WP<7> is negated and RP<7> is asserted, the reset signals: WRST and RRST are asserted to initialize the MMTH again. Note that both W-pointer and R-pointer require a considerable amount of hardware, they are shared by a buffer memory column with a certain bit-width, for example 64bits, here.

E. *Power Consumption related to Bit Change Rate*

If the frequency of read/write operations is the same, a standard FIFO consumes almost the constant power regardless of input data pattern, however MMTH has the unique characteristic that a power consumption depends also on the input data contents. This is caused by the behavior of MMTH



(a) W-Pointer generation circuit

(b) Flip-flop

···	WP<3>	WP<2>	WP<1>	WP<0>
···	1	1	1	1
···	1	1	1	0
···	1	1	0	0
···	1	0	0	0
	⋮	⋮	⋮	⋮

···	RP<3>	RP<2>	RP<1>	RP<0>
···	0	0	0	0
···	0	0	0	1
···	0	0	1	1
···	0	1	1	1
	⋮	⋮	⋮	⋮

(c) Transition of pointer

Fig. 5: Pointer generation circuits

mentioned above. Because the data is written to a number of columns from the input port to the column that W-pointer indicates, it is written on the top of preceding data necessarily. If the same bit is continuously written, almost no power except the controllers for W-pointer and R-pointer is required. Here the probability of bit change is called BCR (Bit Change Rate). The power consumption of MMTH linearly changes in response to BCR. Note that the position of pointers when data are written does not affect the power consumption because all data go through the same number of cells in total.

IV. The Router using MMTH

A. *Baseline Router using register-based FIFOs*

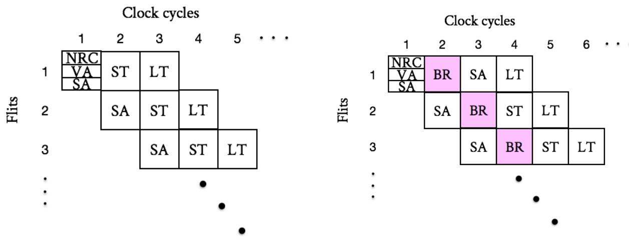
Figure 1 in Section I sketches a standard input-buffered virtual cut-through router for the 2-dimensional mesh network using virtual-channel flow control [21]. Here, this standard router structure is adopted as the baseline for our design. It provides five input and output physical channels (four for neighboring routers and one for the processor core), a 5 × 5 crossbar switch, and a round-robin arbiter that allocates a pair of output virtual and physical channels for each incoming packet. A crossbar switch consists of five 5-to-1 multiplexers, each of which is controlled by a select signal from the arbiter. At each input physical channel, two input buffers are organized as separate FIFO queues for each virtual channel. For these FIFO queues, the baseline router and the proposed router use the standard circular buffers composed of a bunch of flip-flops and MMTH respectively. Besides, an input physical channel has a routing computation unit and a multiplexer that selects only a single output from two virtual channels.

Generally, five steps, Routing Computation (RC), Virtual channel Allocation (VA), Switch Allocation (SA), Switch Traversal (ST) and Link Traversal (LT) are required to transfer a packet through a router. The RC and VA stages are required only for the header flit. The simplest implementation is making the pipeline which processes each step in a clock cycle with a dedicated stage. In NoCs, the basic five-stage pipeline is rarely used since it requires too large latency.

A speculative technique [22] and a look-ahead routing scheme [23] are introduced in order to reduce the number of pipeline stages in a router. Using the speculative technique, VA and SA are performed in parallel. Provided that the VA fails, SA will be ignored even if it succeeds. The look-ahead routing employs Next Routing Computation (NRC) instead of RC. In NRC stage, where the output port of a packet is computed, one hop in advance, and consequently, NRC and VA/SA can be executed in parallel. The first routing is computed beforehand by a source node. Thus, a low latency router with a 3-stage pipeline shown in Figure 6a, which is illustrated from a router's viewpoint, can be designed if standard register-based FIFO is used for the input buffer.

B. *Proposed Router using MMTH*

A router using MMTH, whose architecture is almost the same as the above baseline router, is designed in a similar fashion. However, since MMTH needs an extra clock for a read operation, Buffer Read (BR) stage is necessary for the



(a) A traditional router

(b) A router using MMTH

Fig. 6: Pipeline structures

TABLE I: A type of a flit

Type	Value
None	00
Header	01
Body	10
Pre-header	11

router with MMTH as shown in Figure 6b, and the latency of the packet transfer is stretched if we design naively a router with MMTH. Since the latency of an NoC is directly related to the pipeline depth in a router, it is a serious problem. Thus, a new router design inspired by Flit-Reservation Flow Control [24] for MMTH is proposed in order to avoid the extra clock delay.

Figure 7 illustrates the stage reduction by applying the look-ahead routing scheme from a header flit's viewpoint. The result of NRC is filled in a header flit in the baseline router. In the proposed router, however, the routing information for the next router computed in the NRC stage is filled in an additional temporary flit to transmit only the routing information including the result of NRC and the destination node. This flit is called pre-header flit in the sense that it is antecedent to a header flit. It is a proxy for header flit stored in the buffer with read latency rather than the reservation. As a result, a type of a flit, which is specified in the most significant two bits, is assumed as shown in Table I. The pre-header flit bypasses input buffers and is directly forwarded to the next router as shown in Figure 8. Since bypassing flit does not need BR stage, a clock earlier the pre-header flit arrives at the next router, and consequently, the next router can start VA, SA and NRC during the LT of the first header flit of the packet.

By using this design, the overhead becomes only one clock cycle in the destination router. Their respective latencies are formulated as follows:

$$L_{baseline} = 3H \tag{1}$$

$$L_{naive} = 4H \tag{2}$$

$$L_{proposed} = 3H + 1, \tag{3}$$

where H is the number of hops from source to destination. When H is 3 as shown in Figure 7, the latency is 9 cycles, 12 cycles and 10 cycles, respectively. It is important to note that the overhead of the proposed router (i.e. $L_{proposed} - L_{baseline}$) is always one clock cycle regardless of the number of hops.

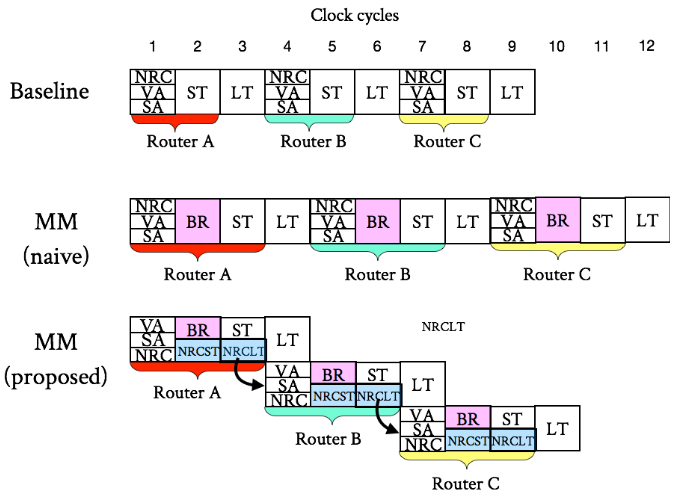


Fig. 7: The latency reduction applying the look-ahead technique

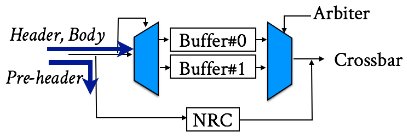


Fig. 8: The latency reduction scheme

This means, the influence of performance overhead is constant even if a number of hops increases. NoCs with 1024 nodes will become realistic around year 2020 [25], and if traffic between distant nodes are increased in such routers, the influence of the overhead may become small. We will see the influence on real applications in Section V.

Several additional external signals are also needed to control MMTH. A reset signal is important to use MMTH since the buffer reset is required when a packet is stored in the buffer. Since the whole flit of a packet is transmitted consecutively in a virtual cut-through router, a reset signal should just be asserted after finishing transmitting a packet. This does not affect performance because the duration of reset is only one clock cycle. In BR stage, an invalid signal is required to invalidate data, since the output during BR stage is not used. This signal is asserted in response to a stage of each virtual channel.

V. Evaluation

To understand the impact of our proposal, we evaluate the baseline router and our proposed router in terms of the performance and power consumption by using a full system simulator and RTL models.

A. Performance overhead

We develop three different router models in a platform, GEM5 full system simulator [26], to investigate the performance degradation by using MMTH in the router. The baseline router, the naive router with BR stage, and the proposed router

TABLE II: CMP System Configuration

System Parameters	Details
Processor	X86-64
# of processors	4
# of directories	4
# of L2 caches	16
L1 I/D cache size	32KB
L2 caches size	256KB
Coherence protocol	MOESI directory

TABLE III: NoC System Configuration in RTL models

System Parameters	Details
Clock frequency	2GHz
Topology	2D-Mesh
# of cores	4
# of VCs per input port	2
Buffer size	8flits
Routing	XY Routing
Arbiter type	Round-robin
Flit size	64bit
Packet size	1 header flit + 6 body flits
Traffic pattern	Uniform

are compared. The baseline router and the naive router have a three-stage pipeline and a four-stage pipeline respectively. The proposed router is based on a three-stage pipeline. However, an extra stage is created in the destination router as described in Section IV-B. We assume a many-core processor with 16 cores which are connected with 4×4 mesh by the above described router. Nine benchmark programs from NAS Parallel Benchmark (NPB) [27] are simulated. These programs are designed to help evaluate the performance of parallel supercomputers. Table II lists the detailed CMP configuration we use to run benchmarks.

First, network performance is evaluated. Figure 9 shows the average latency versus injected traffic under the three different synthetic traffic patterns (the uniform random traffic, bit-complement traffic and tornado traffic). These traces represent a mixture of benign and adversarial traffic patterns. All the simulations are performed for 10000 cycles. Although the naive router design with the BR extra stage stretches the latency of about 20%, the overhead in the latency of the improved design is only approximately 5%. In the case of the tornado traffic pattern, where each node sends packets $(\lceil k/2 \rceil - 1) \bmod k$ hops (k represents the size of the network in a dimension) to the right in the X dimension, the overhead of the naive router is even modest. This is because the number of hops in the tornado traffic tends to be small. When k is four as we assume, the number of hops is mostly only one. The saturation throughput which shows the bandwidth of the network is almost the same in all the three designs.

Secondly, the full system simulation are implemented. Figure 10 shows the execution result of full simulation. This graph also shows that the performance degradation of the improved version router is only 0.5% - 2.0%. A remarkable difference is observed in the case of CG (Conjugate Gradient), which consists of irregular memory accesses and communications. In the program, the overhead of the naive router and the proposed router is 10% and 2%, respectively.

B. Power consumption

We design the baseline router and the proposed router with Renesas's 40nm CMOS design technology to evaluate the power consumption of the router with MMTH. The router architecture is the same as shown in Figure 1. The width of a link is set to be 64bits, and four 16-bit width 8-depth FIFO units are used for a virtual channel. Table III specifies the detailed configuration of RTL models. We adopt the commonly used XY routing where packets are first routed in the X-dimension followed by the Y-dimension.

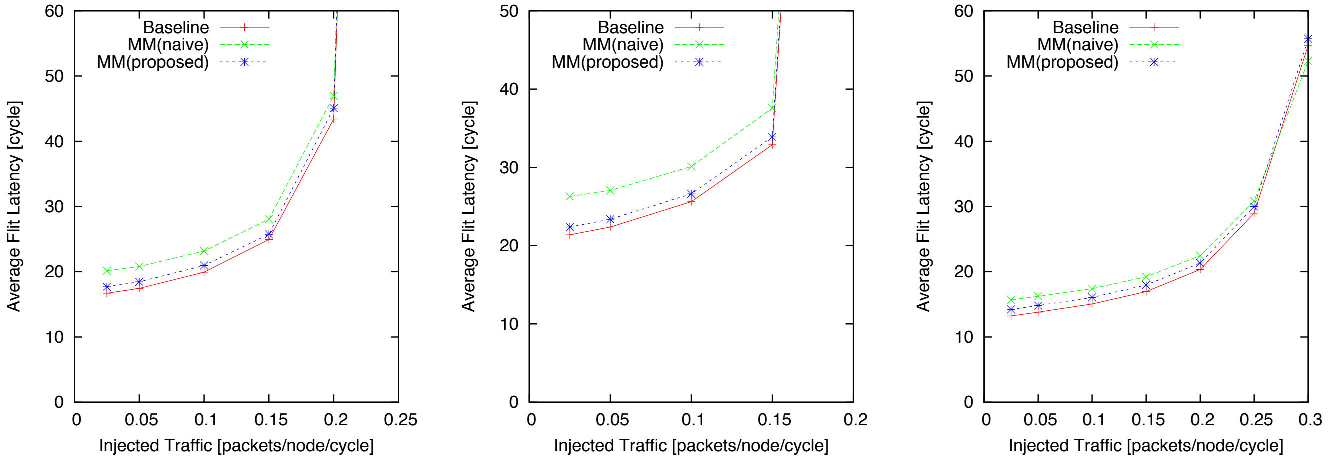
Firstly, we evaluate the power consumption using Apache's PowerArtist [28] on the basis of the RTL and Synopsys's Liberty library format [29]. Since PowerArtist does not take BCR into consideration, the result corresponds to the maximum power consumption (i.e. $BCR = 100\%$). The second bar in Figure 11 shows the maximum power of the proposed router when it works at 2GHz. For the comparison, we also evaluate the baseline router with the traditional register-based FIFO and the first bar in Figure 11 shows the result. Since the standard cells are designed for low energy consumption rather than high speed operation, the baseline router can only work at 800MHz. The dynamic power of the baseline router is scaled assuming that it works at 2GHz. The scaling is needed for fair comparison because MMTH is designed for high speed operation.

From the figure, it appears that the router using MMTH improves the power consumption by 28.8% in the aggregate. Although the power except for input buffers shown as "The others" in the figure increases by 13.3% owing to additional control signals and logic, the power of input buffers shown as "Input buffers" decreases by 46.5%. It indicates that the proposed router can reduce the power even if BCR is 100%.

Subsequently, we take BCR into consideration in application programs. To compute the BCR of benchmark programs, GEM5 full system simulator and NPB are used again. Specifically, the change between current bit and preceding bit is classified into four patterns (0 to 0, 0 to 1, 1 to 0, and 1 to 1) whenever a flit comes to a router, and each rate is computed. Figure 12 shows this result. A sum total of "0 to 1" and "1 to 0" shows BCR. From the figure, BCR is only 25.0% on average and "0 to 0" is especially frequent. It is expected that high-order bits include zeros plentifully. Considering BCR, the real power consumption is computed as follows:

$$P_x = P_{min} + \frac{x}{100}(P_{max} - P_{min}), \tag{4}$$

where P_x, P_{min} and P_{max} are powers when BCR is $x\%$, 0% and 100%, respectively. P_{min} contains only the power consumption for W-Pointer and R-Pointer, and are evaluated by the special tool for MMTH.



(a) Uniform random traffic

(b) Bit-complement traffic

(c) Tornado traffic

Fig. 9: The average flit latency vs. injected traffic

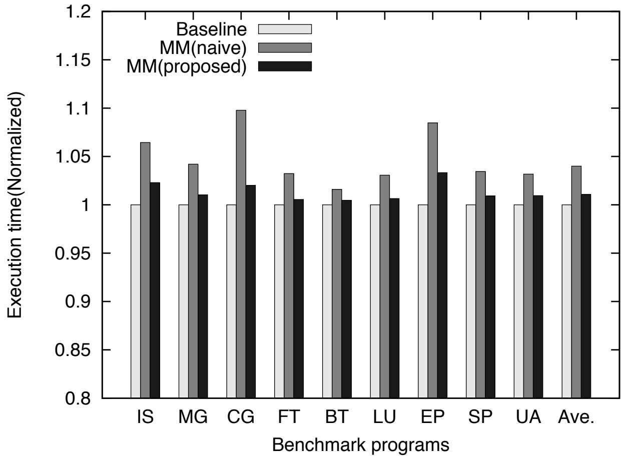


Fig. 10: Execution Time of Full System Simulation

By using P_{min} and P_{max}, we compute the real power consumption with the BCR in NPB. The results are shown in the last bars in Figure 11. As shown in the figure, the reduction ratio of power consumption is further increased to 42.4% on average. As for input buffers, the reduction ratio runs up to 68.4%. Note that this reduction is achieved only by utilizing MMTH. Provided that our approach can be combined with other power reduction technique, more significant reduction can be done.

VI. Conclusions

In this paper, a router using MMTH, which is a novel power efficient buffer memory, has been presented. It is an efficient approach for power reduction without reducing the number of buffers or using complicated scheme. We have compared a baseline router using traditional register-based FIFOs and our proposed router. The present study has demonstrated that the power consumption is associated with the bit change rate of the input data, and when NPB work on NoC, it is reduced by 42.4% on average at 2GHz compared with a traditional FIFO implementation. The performance degradation caused by the delay of the reading time can be mostly saved by the new scheme based on the look-ahead technique in the router. Our results show the execution time of full system simulations increases only 0.5%-2.0% and the saturation throughput is not degraded.

Acknowledgment

This work was supported by New Energy and Industrial Technology Development Organization (NEDO) under the Ministry of Economy Trade and Industry of Japan.

References

[1] L. Benini and G. D. Micheli, “Networks on Chips: A New SoC Paradigm,” *IEEE Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.

[2] W. J. Dally and B. Towles, “Route Packets, Not Wires: On-Chip Interconnection Networks,” in *Proceedings of the 38th Design Automation Conference (DAC)*, Jun. 2001, pp. 684–689.

[3] A. Hemani, A. Jantsch, S. Kumar, A. Postula, J. Oeberg, M. Millberg, and D. Lindqvist, “Network on a Chip: An architecture for billion transistor era,” in *Proceedings of the IEEE Norchip Conference*, Nov. 2000.

[4] M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey, and A. Sangiovanni-Vincentelli, “Addressing the System-on-a-Chip Interconnect Woes Through Communication-Based Design,” in *Proceedings of the 38th Design Automation Conference (DAC)*, Jun. 2001, pp. 667–672.

[5] J. S. Kim, M. B. Taylor, J. E. Miller, and D. Wentzlaff, “Energy Characterization of a Tiled Architecture Processor with On-Chip Networks,” in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*, Aug. 2003, pp. 424–427.

[6] Y. Hoskote, S. Vangal, A. Singh, and N. Borkar, “A 5-GHz Mesh Interconnect for a Teraflops Processor,” *IEEE Micro*, vol. 27, no. 5, pp. 51–61, Nov. 2007.

[7] S. Borkar, “NoCs: What’s the point?” in *NSF Workshop on Emerging Technologies for Interconnects (WETI)*, Feb. 2012.

[8] T. Nakamura and M. J. Flynn, “Marching Memory: designing computers to avoid the Memory Bottleneck,” in *Proceedings of the Sixth International Workshop on Unique Chips and Systems (UCAS)*, Dec. 2010, pp. 44–47.

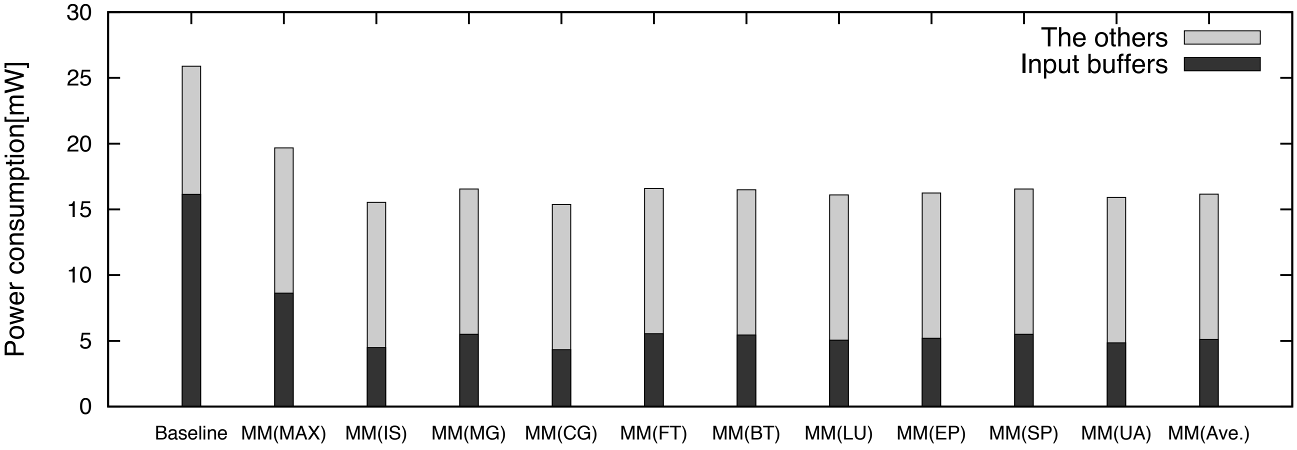


Fig. 11: Power Consumption

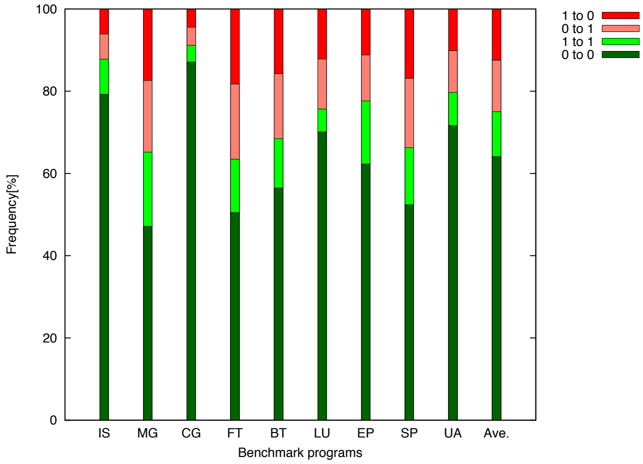


Fig. 12: Bit Change Rate

[9] P. Kundu, "On-Die Interconnects for Next Generation CMPs," in *Workshop on On- and Off-Chip Interconnection Networks for Multicore Systems*, Dec. 2006.

[10] X. Chen and L.-S. Peh, "Leakage power modeling and optimization in interconnection networks," in *Proceedings of International Symposium on Low Power Electronics and Design (ISLPED)*, Aug. 2003, pp. 90–95.

[11] T. T. Ye, L. Benini, and G. D. Micheli, "Analysis of power consumption on switch fabrics in network routers," in *Proceedings of the 39th Design Automation Conference (DAC)*, Jun. 2002, pp. 524–529.

[12] T. Moscibroda and O. Mutlu, "A Case for Bufferless Routing in On-Chip Networks," in *Proceedings of the 36th International Symposium on Computer Architecture (ISCA)*, Jun. 2009, pp. 196–207.

[13] P. Baran, "On distributed communications networks," *Communications systems, IEEE Transactions on*, no. 1, Mar. 1964.

[14] C. Fallin, C. Craik, and O. Mutlu, "CHIPPER: A low-complexity bufferless deflection router," in *Proceedings of the 17th IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb. 2011.

[15] S. M. Hassan and S. Yalamanchili, "Centralized Buffer Router: A Low Latency, Low Power Router for High Radix NOCs," in *Proceedings of the 7th ACM/IEEE International Symposium on Networks-on-Chip (NOCS)*, May 2013, pp. 118–125.

[16] G. Michelogiannakis and W. J. Dally, "Elastic Buffer Flow Control for On-Chip Networks," *Computers, IEEE Transactions on*, vol. 62, no. 2, pp. 295–309, Feb. 2013.

[17] C. A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M. S. Yousif, and C. R. Das, "ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers," in *Proceedings of the 39th annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Dec. 2006, pp. 333–346.

[18] D. Matos, C. Concatto, M. Kreutz, F. Kastensmidt, L. Carro, and A. Susin, "Reconfigurable Routers for Low Power and High Performance," *VLSI Systems, IEEE Transactions on*, vol. 19, no. 11, pp. 2045–2057, Nov. 2011.

[19] H. Jang, B. S. An, N. Kulkarni, K. H. Yum, and E. J. Kim, "A Hybrid Buffer Design with STT-MRAM for On-Chip Interconnects," in *Proceedings of the 6th ACM/IEEE International Symposium on Networks-on-Chip (NOCS)*, May 2012, pp. 193–200.

[20] M. J. Flynn, *Computer Architecture: Pipelined and parallel processor Design*. John & Bartlett Publications, 1995.

[21] W. J. Dally, "Virtual-channel flow control," *IEEE Trans. on Parallel and Distributed Systems*, vol. 3, no. 2, 1992.

[22] L.-S. Peh and W. J. Dally, "A Delay Model and Speculative Architecture for Pipelined Routers," in *Proceedings of the 7th International Symposium on High-Performance Computer Architecture (HPCA)*, Jan. 2001, pp. 255–266.

[23] W. J. Dally and B. P. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2004.

[24] L.-S. Peh and W. J. Dally, "Flit-Reservation Flow Control," in *Proceedings of the 6th International Symposium on High-Performance Computer Architecture (HPCA)*, Jan. 2000, pp. 73–84.

[25] M. Eggenberger and M. Radetzki, "Scalable Parallel Simulatio of Networks on Chip," in *Proceedings of the 7th ACM/IEEE International Symposium on Networks-on-Chip (NOCS)*, May 2013, pp. 1–8.

[26] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 Simulator," *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, pp. 1–7, May 2011.

[27] H. Jin, M. Frumkin, and J. Yan, "The OpenMP Implementation of NAS Parallel Benchmarks and Its Performance," in *NAS Technical Report NAS-99-011*, Oct. 1999.

[28] "Power artist: Apache design, inc." http://www.apache-da.com/products/powerartist/.

[29] "Liberty Library Modeling," http://www.synopsys.com/community/interoperability/pages/libertylibmodel.aspx.