

Increasing the Throughput of an Adaptive Router in Network-on-Chip (NoC)

Seung Eun Lee and Nader Bagherzadeh

Dept. of Electrical Engineering and Computer Science, University of California, Irvine
Irvine, CA, 92697, USA

seunglee@uci.edu, nader@uci.edu

ABSTRACT

In this paper, we propose a simple and efficient mechanism to increase the throughput of an adaptive router in *Network-on-Chip (NoC)*. One of the most serious disadvantages of fully adaptive wormhole routers is its performance degradation due to the routing decision time. The key idea to overcome this shortcoming is the use of different clocks in a head flit and body flits, because the body flits can be forwarded immediately and the *FIFO* usually operates faster than route decision logic in an adaptive router. The major contributions of this paper are: 1) a proposal of a simple and efficient mechanism to improve the performance of fully adaptive wormhole routers, 2) a quantitative evaluation of the proposed mechanism showing that the proposed one can support higher throughput than a conventional one, and 3) an evaluation of hardware overhead for the proposed router. In summary, the proposed clock boosting mechanism enhances the throughput of the original adaptive router by increasing the accepted load and decreasing the average latency in the region of effective bandwidth.

Categories and Subject Descriptors

C.1.2 [Processor Architectures]: Multiple Data Streaming Architecture (MultiProcessor)—*Interconnection architecture*

General Terms

Algorithms, Performance, Design, Verification

Keywords

Network-on-Chip (NoC), Interconnection Network, Wormhole Routing, Adaptive Router, Chip-Multiprocessor

1. INTRODUCTION

Recently, on-chip interconnection network, *Network-on-Chip (NoC)*, using packet switching technique has been pro-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CODES+ISSS'06, October 22–25, 2006, Seoul, Korea.

Copyright 2006 ACM 1-59593-370-0/06/0010 ...\$5.00.

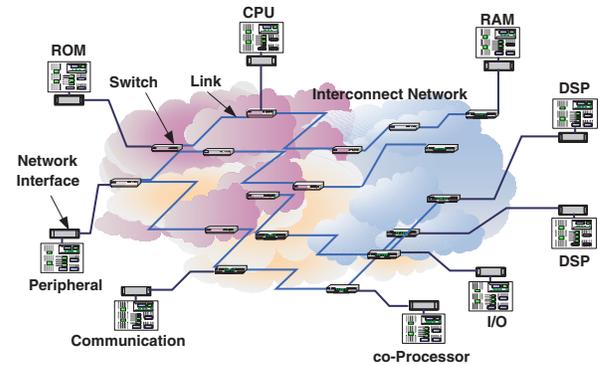


Figure 1: Example of Network-on-Chip architecture

posed to accommodate the trend for *System-on-Chip (SoC)* integration [1, 2]. An example of the *NoC* architecture is shown in Figure 1. The basic idea is the use of packet switching technique that has been extensively used for computer networks. *NoC* is a promising solution to the communication challenges of on-chip interconnection, featuring the requirements of the next-generation multiprocessor systems.

In this paper, we propose a simple and efficient mechanism to increase the throughput of an adaptive router in *NoC*. One of the most serious disadvantages of fully adaptive wormhole routers is its performance degradation due to the route decision time. To overcome this weak point, we propose the use of a faster clock during the service of the body flit. We will show that router throughput can be enhanced with respect to the conventional adaptive routers while keeping the same functionality. To the best of our knowledge, the idea of increasing throughput with boosting clock has not been addressed before. In order to show the feasibility of our mechanism, a detailed hardware implementation will be presented and evaluated.

The major contributions of this paper are: 1) a proposal of a simple and efficient mechanism to improve the performance of fully adaptive wormhole routers, 2) a quantitative evaluation of the proposed mechanism showing that the proposed router can support higher throughput than a conventional one, and 3) an evaluation of hardware overhead for the proposed router.

The rest of this paper is organized as follows. Section 2 will present the motivation for this research. An informal and intuitive description of the proposed mechanism will be presented in Section 3. Section 4 will describe the base-

line router that is used to evaluate the performance of the proposed mechanism in detail. The implementation of the proposed mechanism and the experimental results will be presented in Sections 5 and 6, respectively. Finally, conclusions are drawn in Section 7.

2. MOTIVATION

Recently, a number of researches have been working on network architectures appropriate for on-chip environments. Different routing algorithms, such as deterministic, oblivious and adaptive routing algorithms have been proposed. Many researchers used deterministic or oblivious algorithms such as *DOR* [3], *ROMM* [4], and *OITURN* [5] for simplicity and ease of analysis. Some researchers have developed better performance routing algorithms even using adaptive routing algorithms [6, 7, 8, 9, 10, 11, 12]. Recently there have been several implementation related works using deterministic routing algorithms as well as adaptive routing ones [13, 14].

The design of a high performance router for on-chip interconnection has tight resource constraints such as router’s area, power, and speed. The adaptive routing has been proposed as a method of improving network utilization by using information about the network state to select a path among alternative paths to deliver a packet, potentially reducing network latency. However, while a good adaptive routing algorithm can balance network occupancy and enhance its maximum throughput, it also suffers from the expensive cost in terms of additional sophisticated logic and performance degradation due to the routing decision time.

Wormhole flow control has increasingly been advocated as a means of reducing latency. It reduces latency by routing a packet as soon as its head flit arrives at a node. The routing of a head flit enables the switches to establish the path and body flits are simply forwarded along the path as a pipelined fashion. However, wormhole flow control has some disadvantages. A dominant one is that a router stops a packet when its head flit is blocked. When the link requested by a head flit is busy, the head flit could not advance and the remainder of the flits is also stopped holding the buffers and channels along the path that has already formed leading to significant link congestion.

One of the most serious disadvantages of an adaptive wormhole router is the performance degradation due to the routing decision time because routing flexibility requires additional resources. Our goal is to design a fast fully adaptive router which will provide more throughput than current wormhole routers. Thus, we wish to design a wormhole router competitive at medium and high loads by increasing network throughput.

In this paper, we will investigate the use of multiple clocks to implement an adaptive wormhole router. We will show how this technique reduces latency and greatly increases throughput of an adaptive router. We will also demonstrate the viability of our proposal by providing real experimental results.

3. CLOCK BOOSTING ROUTER

In this section, we propose a mechanism to enhance the throughput of an adaptive wormhole router. An informal and intuitive description of the proposed mechanism will be presented.

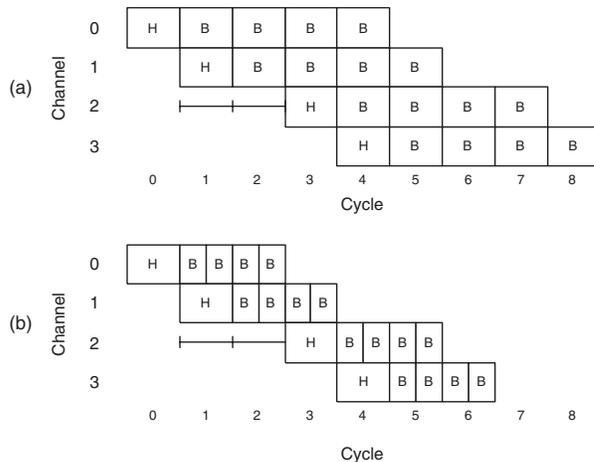


Figure 2: Time-space diagram showing wormhole flow control sending a 5-flit packet over 4 channels. (a) The conventional wormhole flow control. (b) The proposed mechanism.

3.1 Mechanism

In an adaptive wormhole router, the routing decision time for the head flit is the critical path of an adaptive router restricting the operating frequency of overall router. While the head flit requires support of sophisticated logic increasing the decision path, body flits can continue advancing along the reserved path that is already established by the head flit.

The key idea of this paper is use of different clocks in a head flit and body flits, because the body flits can be forwarded immediately without any computation and the *FIFO* usually operates much faster than route decision logic in an adaptive router. Figure 2 explains the mechanism for the proposed router. The conventional wormhole flow control uses same clock to advance the head flit and body flits. Generally, the operating frequency is defined by the routing decision logic for the head flit. The proposed mechanism uses a faster clock to advance the body flits. It reduces the average latency of a packet as well as the contention by compacting the effective length of body flits in the time domain.

3.2 Analysis of the Mechanism

The performance of an interconnection network can be described by latency versus offered traffic curves. Although latency versus offered traffic curves give the most accurate view of the ultimate performance of an interconnection network, they do not have simple, closed form expressions and are generally found by discrete-event simulation [15].

Zero-load latency gives a lower bound on the average latency of a packet through the network by assuming that a packet never contends for network resources with other packets. The zero-load latency of a packet using wormhole routing is

$$T_0 = Ht_r + L/b \quad (1)$$

ignoring wire latency. The first term reflects the time required for the head flit to traverse the network, with an average hop count of H and a delay of t_r through a single

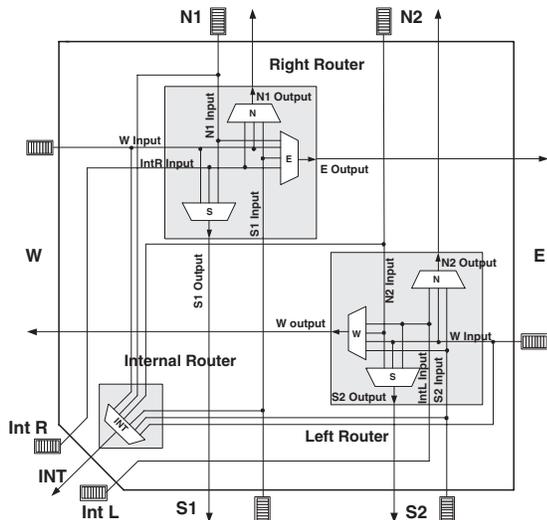


Figure 3: Adaptive router architecture.

router. The second term is serialization latency, the time for a packet of length L to cross a channel with bandwidth b . The proposed mechanism reduces the serialization latency by boosting clock frequency during the body flit transfer. The boosting of clock frequency results in increasing the bandwidth and reduces the zero-load latency. To evaluate the upper bound of average latency between the proposed mechanism and the original one, average latency versus offered traffic curves are obtained by the discrete event simulation in Section 6, demonstrating the performance improvement with the proposed mechanism.

4. BASELINE ADAPTIVE ROUTER

We propose an adaptive routing algorithm and architecture for a flexible on-chip interconnection. This technique uses a wormhole switching technique with a deadlock- and livelock- free algorithm for 2D-mesh topology. The proposed router demonstrated near-optimal performance with comparison to *O1TURN* [5] in terms of average latency. Moreover the bandwidth and the total area overhead of the router enabled the router as a feasible alternative to existing routers for *NoC*. In this paper, we use the adaptive router as a baseline router and improve the throughput by adopting the proposed clock boosting mechanism.

4.1 Overview

We assume the network topology in 2D-mesh which is $N \times M$ routers. The overall block diagram of a single router is shown in Figure 3. There is an *FIFO* queue per each input channel and each output port has an associated arbiter to choose the proper packet among the given incoming packet from each candidate input port. We consider a router with seven interfaces, suitable for a 2D-mesh with an additional interface to an integrated processing element. We assume that a packet coming through an input port does not loop back, thus each input port is connected to four output ports.

The router is composed of three architectural blocks: *Right Router*, *Left Router*, and *Internal Router*. The *Right Router*

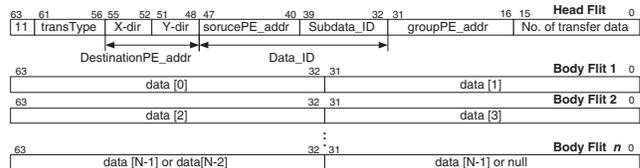


Figure 4: Packet format.

serves the port set $\{W\text{-in}, N1, E\text{-out}, S1\}$. On the contrary, the *Left Router* serves the port set $\{E\text{-in}, N2, W\text{-out}, S2\}$. The *Internal Router* supports the additional interface to an integrated processing element. The separated routing paths for vertical direction and unidirectional path for horizontal direction allow the network avoid cycles in its channel-dependency-graph, resulting a dead lock-free operation [7, 8]. Also by choosing the shortest path in routing, a livelock free operation is guaranteed.

4.2 Packet Format

The Figure 4 shows the packet format. Each packet has the *DestPE_addr* field to indicate the destination node in the head flit. The address of the destination node is represented by relative distance of horizontal and vertical direction. A positive value represents southern and eastern direction in vertical and horizontal direction, respectively. Each relative distance is signed magnitude value, i.e. *MSB* of each *X-dir* and *Y-dir* field represents its sign and the rest of bits represent its magnitude. For instance, if *destPE_addr* has $0x91$ in hexadecimal format, it represents that the destination node is located at western 1 hop and southern 1 hop from the current node. Its vectorized representation in $X - Y$ coordinate is $(-1, 1)$. The head flit also has the *No_of_Flits* field to represent the number of body flits followed by the head flit.

4.3 Priority

For outgoing channel allocation, the router applies a fixed priority scheme to the incoming packets that have reached the corresponding node simultaneously. For each outgoing channel, the possible incoming channels have a descending order of priority in a clock-wise direction. Similarly, each output port has a descending order of priority in clock-wise direction from *N1*. The incoming packets and output port of an internal router have the lowest priority.

4.4 Router Architect

The detail block diagram of the *Right Router* is shown in Figure 5. The routers for each output port are placed according to their priority level from the highest router (*N1*) to the lowest router (*S1*). Each incoming packet is directed to the *Header Parsing Unit (HPU)* per each output port. The *HPU* generates a set of possible incoming packets, which could be routed to the corresponding output port, in order of the input priority level by looking up the destination address in the head flit.

When the output port is available (by referencing *FULL* signal), the router chooses the input packet for the corresponding output port among the set of routable incoming entries provided by the *HPU*. If two or more packets arrive simultaneously, the arbiter will choose a packet according to their priority.

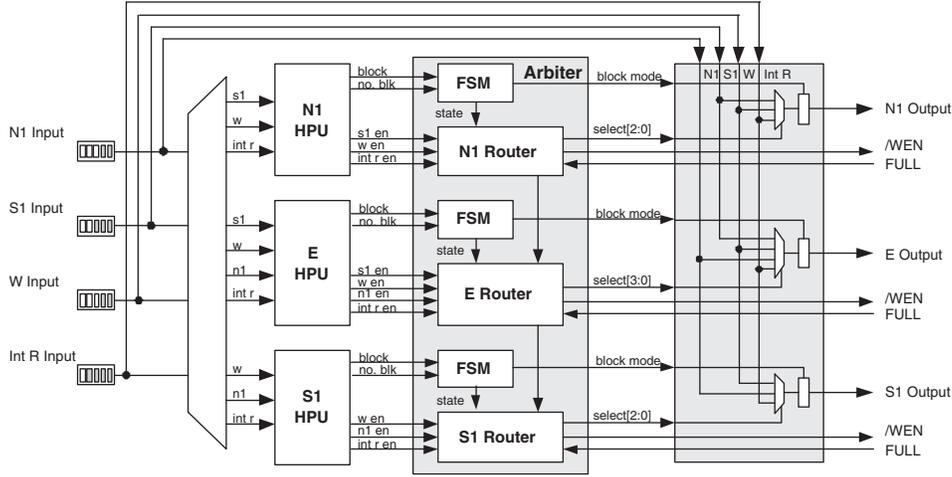


Figure 5: Micro-architecture of the *Right/Left* router.

If one output port is granted to one head flit, it reserves the output port until all body flits are forwarded. To support this feature, our router has a *Finite State Machine (FSM)* in each output port, which stores the state of the corresponding router. Also it counts the number of forwarded flits in order to record the remaining number of the body flits that need to be forwarded. Therefore, each incoming packet has its own path in order to forward the body flits to its selected output port. The body flits arriving along with the router output without passing through the *Arbiter* unit.

After completing the decision path for the highest router, the lower prioritized router references the decision result of the highest router and disables the incoming port, which was served by the former router, from the set of possible incoming packets for the router. In this manner, the arbitration is done by propagating the routing decision from the highest router to the lowest router. Therefore all incoming packets could be advanced in their output port at once.

Finally, the multiplexer unit maps corresponding incoming packet to the output port by referencing the *SELECT* signal. It also updates the destination address in the head flit in order to complete proper transmission; (1) Head flit: decrease the corresponding destination address, (2) Body flits: bypass the incoming flits.

In order to achieve high performance, all routing decisions are made within one clock cycle. So, an incoming flit can advance in one clock cycle if the corresponding output port is free.

5. CLOCK BOOSTING ADAPTIVE ROUTER DESIGN

This section describes the implementation of the proposed mechanism based on the previously mentioned router. Figure 6 shows the block diagram of modified router. The hardware cost of the proposed mechanism is a multiplexer in each channel for switching the clock domain and use of a *FIFO* supporting dual clock operation. The clock multiplexer is in charge of switching clock domain according to the service flit and it also could drive the corresponding load with appropriate fan out. In this research, we use a multiple clock

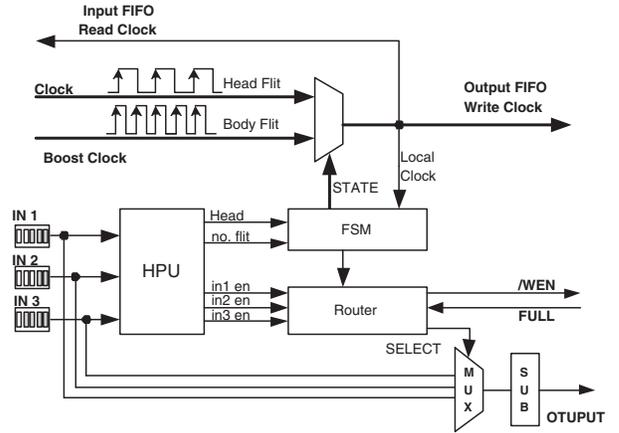


Figure 6: Micro-architecture of the modified router.

scheme to boost clock speed for the body flits in order to simplify the implementation. As a test case we use two times faster clock (2x) and four times faster clock (4x) since the original router and the FIFO show the operating frequency up to 423MHz and 1.8GHz, respectively.

To ensure constancy of clock phase during clock domain changes between original clock (1x) and boosting clock (2x and 4x), we add a constraint in body flit size. Therefore, in case of using 2x boosting clock, the length of body flits is limited to a multiple of two and in case of using 4x boosting clock, the length of body flits is limited to a multiple of four.

6. EXPERIMENTAL RESULTS

We use the previously proposed adaptive router as a baseline and make performance comparison in different boosting clock frequencies such as two-times (2x) and four-times (4x) faster clock.

6.1 Evaluation methodology

In order to evaluate the performance of the proposed clock boosting mechanism, we have developed the router written

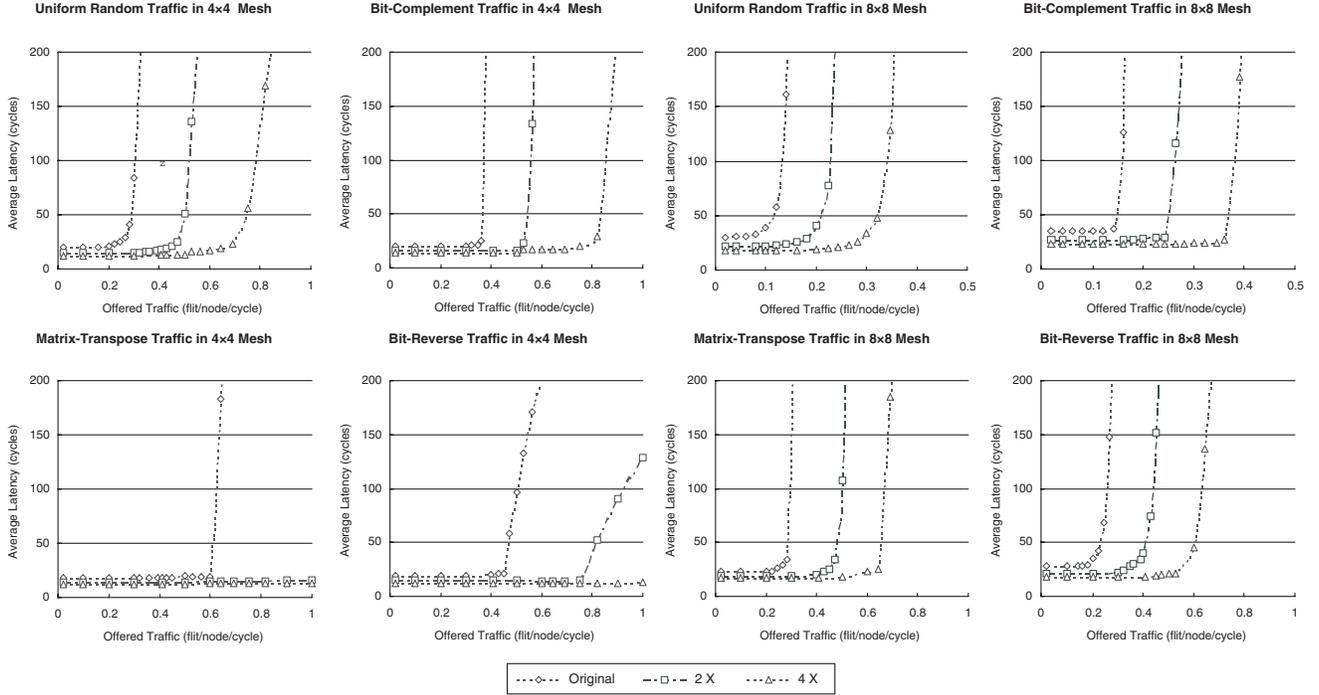


Figure 7: Average latency versus offered traffic curve.

in *VerilogTM HDL*. For the measurement of throughput and adjusting incoming traffic, we adopted a standard interconnection network measurement setup [15] where the packet generation is placed in front of an infinite depth source queue and an input timing of each packet is measured whenever it is generated. Without infinite depth source queue at source packet generator, the measured latency does not apply to real network environments such as delays caused by packet contention and network congestion. In this simulation, the number of body flits in a packet was fixed to 8 even though the defined packet format supports various sized flits. And the depth of a *FIFO* between each link is fixed to 8 in this simulation. In order to achieve fair comparison between the original router and the proposed router, we assume that the operating frequency of each processing element is the same as the operating frequency of the original router. So, the input packet is injected in each cycle of the processing element's operation frequency during the simulation of the boosting router featuring the same traffic load with the original router.

6.2 Simulation results

The simulation is completed using four different traffic patterns such as uniform random, bit-complement, matrix transpose, and bit-reverse traffics in 4×4 , and 8×8 mesh networks (see Figure 7). Each graph represents offered traffic (flit/node/cycle) in X -axis and average latency (cycles) in Y -axis.

For the 4×4 network, the adaptive router with 4x boosting clock makes the phase transition region shift from 0.26 flit/cycle to 0.7 flit/cycle with uniform random traffic, from 0.36 flit/cycle to 0.8 flit/cycle with bit-complement traffic, from 0.6 flit/cycle to 1.0 flit/cycle with matrix transpose traffic, and from 0.45 flit/cycle to 1.0 flit/cycle with bit-

reverse traffic. Similar improvements have been obtained for the 8×8 mesh network. In this case, the phase transition region shifts from 0.12 flit/cycle to 0.32 flit/cycle with uniform random traffic, from 0.14 flit/cycle to 0.36 flit/cycle with bit complement traffic, from 0.28 flit/cycle to 0.64 flit/cycle with matrix transpose traffic, and from 0.18 flit/cycle to 0.54 flit/cycle with bit-reverse traffic. The critical traffic load grows with the boosting clock frequency thus utilizing the link bandwidth that is wasted in the original router.

Table 1 describes the zero-load latency calculated by equation (1) and average latency with 0.02 flit/cycle offered traffic obtained by simulation in case of uniform random traffic pattern. The difference between ideal zero-load latency and simulation results is because of the contention in network and the input injection assumptions. While we inject the flits with 0.02 flit/cycle, it could result in contention situation. For instance, in the case of 4×4 network, total of 32 flits are traveling and in the case of 8×8 , total of 128 flits are traveling in the network while each node injects the flit with the rate of 2% flit/cycle. In this simulation we assume that the operating frequency of each processing element is the same as operating frequency of the router (1x). So, in case of boosting operation, the input body flit could not be injected in the incoming port yet adding latency. Although there exists small difference between them, the experimental results show the superiority of the proposed mechanism in the range of low data rate.

6.3 Physical Implementation

A logic description of our router's component has been obtained using the synthesis tools from the *SynopsysTM* using *TSMCTM 90nm* technology. While the results are obtained by simulation, it provides us with the physical design characteristics of the proposed routers.

Table 1: Zero-load Latency and Average Latency with 2% flit/cycle Offered Traffic (Cycles) with Uniform Random Traffic

Boosting Router	4 × 4 mesh network		8 × 8 mesh network	
	Zero	2%	Zero	2%
Original (1x)	15.549	19.54	20.1288	30.33
Boost (2x)	11.549	14.72	16.1288	21.71
Boost (4x)	9.549	12.28	14.1288	17.44

Table 2: Physical Characteristics of the Router

	Router 1x	Router 2x	Router 4x
Voltage	1.0V	1.0V	1.0V
Frequency	423MHz	421MHz	421MHz
Area	17,537 μm^2	17,821 μm^2	17,830 μm^2
Dynamic Power	2.4680mW	3.3024mW	5.7279mW
Leakage Power	171.938 μW	175.434 μW	175.686 μW

Table 2 summarizes the physical characteristics of routers. The additional area cost in the clock boosting router is basically due to the clock multiplexer for each channel. Table 3 describes the physical characteristics of the original FIFO and modified FIFO that supports dual clock operation (read/ write). If it were integrated within NoC using the same technology, the total area overhead imposed by router would be relatively negligible.

In summary, the simulation results demonstrate that the proposed clock boosting mechanism enhances the throughput of the original adaptive router by increasing the accepted load and decreasing the average latency in the region of effective bandwidth.

7. CONCLUSIONS

In this paper, we have proposed a simple and efficient mechanism to increase the throughput of an adaptive router in NoC. One of the most serious disadvantages of fully adaptive wormhole routers is its performance degradation due to the route decision time. The key idea to overcome this shortcoming is the use of different clocks in a head flit and body flits, because the body flits can be forwarded immediately and the FIFO usually operates faster than route decision logic in adaptive router.

The proposed mechanism is implemented for the adaptive wormhole router and the performance evaluation was completed using simulation. The simulation results demonstrated the performance enhancements in terms of maximum accepted traffic and the average latency in the range of accepted traffic. Moreover the proposed router was synthesized using 90nm technology and proved the feasibility of

Table 3: Physical Characteristics of the FIFO with depth 8

	FIFO Single Clock	FIFO Dual Clock
Voltage	1.0V	1.0V
Frequency	1.81GHz	2.32GHz/Write 1.68GHz/Read
Area	17,428 μm^2	14,412 μm^2
Dynamic Power	10.1191mW	10.1116mW
Leakage Power	160.770 μW	160.342 μW

this mechanism for NoC design. This mechanism could be applicable for conventional adaptive routers with wormhole flow control.

8. REFERENCES

- [1] W. J. Dally and B. Towles. Route Packets, Not Wires: On-chip Interconnection Networks. In *Proc. of the 38th Annual conference on Design Automation (DAC'01)*, pages 684–689, June 2001.
- [2] L. Benini and G. D. Micheli. Networks on Chips: A New SoC Paradigm. *IEEE Computer*, 35(1):70–78, 2002.
- [3] H. Sullivan and T. R. Bashkow. A Large Scale, Homogeneous Fully Distributed Parallel Machine. In *Proc. of the 4th annual ACM Symp. on Computer Architecture (ISCA'77)*, pages 105–117, 1977.
- [4] T. Nesson and S. L. Johnsson. ROMM Routing on Mesh and Torus Networks. In *Proc. of the 7th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'95)*, pages 275–287, 1995.
- [5] D. Seo, A. Ali, W. Lim and N. Rafique. Near Optimal Worst-case Throughput Routing for Two-Dimensional Mesh Networks. In *Proc. of the 32nd Annual Int'l Symposium on Computer Architecture (ISCA'05)*, pages 432–443, 2005.
- [6] J. Hu and R. Marculescu. DyAD: Smart Routing for Networks-on-Chip. In *Proc. of the 41st Annual conference on Design Automation (DAC'04)*, pages 260–263, 2004.
- [7] W. Dally and C. L. Seitz. Deadlock-free Message Routing in Multiprocessor Interconnection Networks. *IEEE Trans. on Computer.*, C-36(5):547–553, 1987.
- [8] J. Duato. A New Theory of Deadlock-free Adaptive Routing in Wormhole Networks. *IEEE Trans. on Parallel and Distributed Systems*, 4(12):1320–1331, 1993.
- [9] G. Chiu. The Odd-Even Turn Model for Adaptive Routing. *IEEE Trans. on Parallel and Distributed Systems*, 11(7):729–738, 2000.
- [10] C. J. Glass and L. M. Ni. Adaptive Routing in Mesh-connected Networks. In *Proc of the 12th Int'l Conference on Distributed Computing Systems (ICDCS'92)*, pages 12–19, June 1992.
- [11] C. J. Glass and L. M. Ni. The Turn Model for Adaptive Routing. *Journal of the ACM*, 41(5):874–902, 1994.
- [12] V. Puente, et al. Adaptive Bubble Router: A Design to Improve Performance in Torus Networks. In *Proc. of the 28th Int'l Conference on Parallel Processing (ICPP'99)*, pages 58–67, Sept. 1999.
- [13] S. J. Lee, K. LEE and H. J. Yoo. Analysis and Implementation of Practical, Cost-effective Networks on Chips. *IEEE Design & Test of Computers*, 22(5):422–433, 2005.
- [14] K. Goossens, J. Dielissen and A. Radulescu. Æthereal Network on Chip: Concepts, Architectures, and Implementations. *IEEE Design & Test of Computers*, 22(5):414–421, 2005.
- [15] W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufman, San Francisco, 2004.