國立臺灣大學 資訊工程學系

碩士論文

晶片網路下之權重式錯誤偵測及架構

江忠桓 撰

98

6

國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

晶片網路下之權重式錯誤偵測及架構

Weighted Error Detection Method and NoC Architecture

江忠桓

Chung-Huang Jiang

指導教授：賴飛羆 博士

Advisor: Feipei Lai, Ph.D.

中華民國 98 年 6 月

June, 2009

國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

晶片網路下之權重式錯誤偵測及架構

Weighted Error Detection Method and NoC Architecture

江忠桓

Chung-Huang Jiang

指導教授：賴飛羆 博士

Advisor: Feipei Lai, Ph.D.

中華民國 98 年 6 月

June, 2009

誌謝

能夠順利完成碩士的學位，首先要特別感謝我的指導教授賴飛羆教授，在這兩年中，提供了我們豐富的研究資源與良好的環境，並授予我們許多專業的知識，以及在學業外的生活觀念及態度，並時常安排戶外的爬山及健行活動，讓我們在致力研究之餘，仍擁有健康的生活。

我還要感謝許多幫助過我的人，感謝東海電機系的蔡坤霖教授，經常與我們討論，指導我的研究方向，並適時的給予最佳的建議，對症下藥，使我在撰寫論文所碰到的問題，都能夠迎刃而解。

並感謝在實驗室的每位學長、同學、學弟。感謝志弘學長對實驗室的付出，還有建宇、翔仁、于揚、晏章、柏舜學長們在學業以及生活上的幫助，以及同學們小勝、育安、承浩一起過了這多采多姿的兩年。還有感謝各位學弟們，有你們的加入，我相信實驗室各方面都會更加茁壯。

最後，我要感謝我的家人，感謝我的爸媽，有他們辛苦的工作，才能讓我有無憂無慮的生活，他們對我的關心，使我這兩年來可以毫無後顧之憂的做研究。

摘要

單晶片系統(System-on-Chip)設計目前已經廣泛地運用在多媒體、電信通訊、消費性電子領域的電路設計。隨著製程不斷地縮小，越多的矽智財(IP)可以整合到單一晶片裡。各個矽智財之間的訊息交換將變成系統效能的瓶頸。一個基於封包交換傳輸方式的晶片網路(Network-on-chip)被提出來克服解決系統晶片連結的問題。在現今的晶片網路設計中服務品質(QoS)成為一個主要的設計目標，其中容錯性為服務品質中的一個部分。在之前的所設計過的錯誤偵測方法以及架構中在功率消耗以及延遲上並不符合需求。在此篇論文中，我們提出一個具權重式的錯誤偵測以及架構來達成可容錯性。設計主旨為將較具關鍵性及重要性的資料封包部分，以較嚴密的方式檢查之；剩下的部分，用較簡單的方式檢查之。實驗結果顯示，在大部分的情況下，我們所提出的方式及架構，會有較佳的"功耗延遲”的乘積。

Abstract

System-on-chip design has been commonly used in modern circuit design in multimedia, telecommunications and consumer electronics domain. With the technology scaled down, more IP cores can be integrated into a single ship. The interconnection between IP becomes the performance bottleneck. Network-on-chip (NoC) which is packet switch based communication is proposed to overcome the SoC interconnection problem. In modern NoC design, QoS (quality of service) becomes the main issue. Fault tolerant is one part of QoS. In fact, the previous error mechanism and architecture in fault tolerant NoC design can not meet our demands, no matter power or latency. In this thesis, we proposed a weighted error detection and architecture to overcome fault tolerant issue. The concept is letting the significant and critical part be checked strictly and the other part will be checked slightly. Simulation and results demonstrate that it can reduce the latency power product better than previous architectures in most conditions.

CONTENTS

List of Figures and Tables

Chapter 1 Introduction

With technology scaling, more and more processing elements (PEs) could be integrated into a single chip. The conventional interconnection between PEs becomes the performance bottleneck under such situation. Network-on-chip (NoC) has been proposed as a feasible method to overcome that problem in the future.

1.1 System-on-Chip (SoC)

System-on-Chip (SoC) design is to combine several intellectual properties (IPs) which have different functionalities into an integrated circuit (IC). With deep sub-micron CMOS technology, modern IC engineers must handle more than 500M transistors in a single chip. That means the increasing of design complexity and longer time to market. SoCs which provide a solution to cut down the development cycle, reduce the design complexity, cost, power and increase performance have been commonly used in multimedia, telecommunications and consumer electronics domain. The typical SoC platforms include microprocessor, DSP core, SRAM, Peripherals like timer and counter and external interface like USB, Ethernet. Fig. 1-1 is an example of SoC platform. We can see that IPs on SoC connected by a standard interconnection architecture shared bus. With this kind of simple and low overhead bus protocol those PEs may communicate with each other.

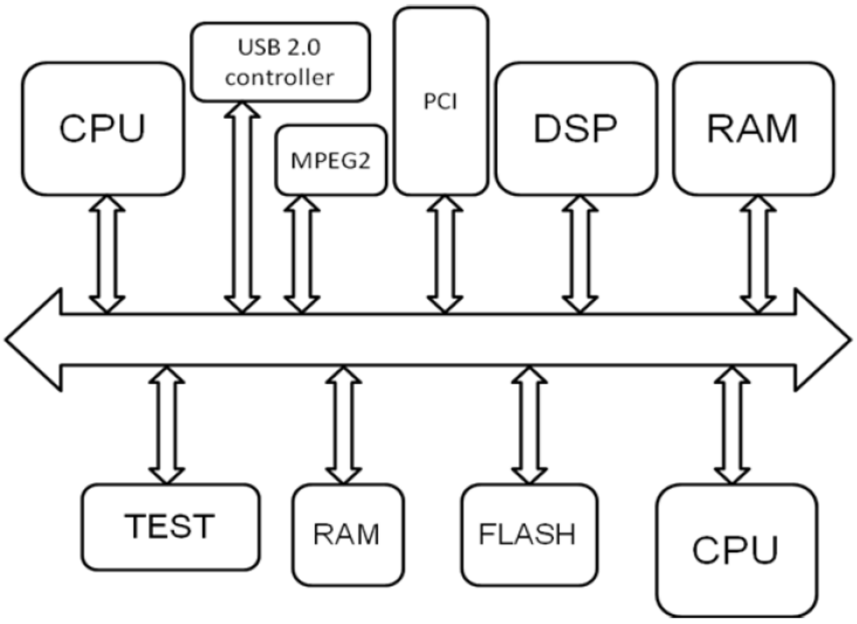


Fig. 1-1 Soc System

However, in this shared-medium architecture, all the devices need to contend for the shared bus. Only one PE can use the communication channel simultaneously. And we need to arbitrate the bus requests when several processors ask to use the bus at the same time. When more IPs connects to the bus, much complex centralized arbitration mechanism is needed. As a result, bus idle time will be longer than actually transfer time. It may restrict the shared-medium scalability and in fact that there are fewer than five processors and, rarely, more than 10 bus masters in current SoC design. [1] There will be tens to hundreds PEs delivering data in the future integrated systems. It would become a crucial performance bottleneck if we still use bus based interconnection.

1.2 Network-on-Chip (NoC)

Network-on-chip (NoC) has been proposed as a possible solution to overcome the above-mentioned problems. It borrows the packet switch concept from networking to on-chip communication. The fundamental concept of NoC is illustrated in Fig. 1-2. Each PE transfers data to other ones by intelligent switches. Those switches provide programmable connection between the input/output ports by the routing policy we decided. And we can choose any topology to connect the switches. Generally, the interfaces of these functional units are not designed for on-chip network. For this reason, we need a network interface (NI) between IPs and switches. There are two main functions of NI. One is to transform the data to the packet format. The other is to synchronize and communicate with switches.

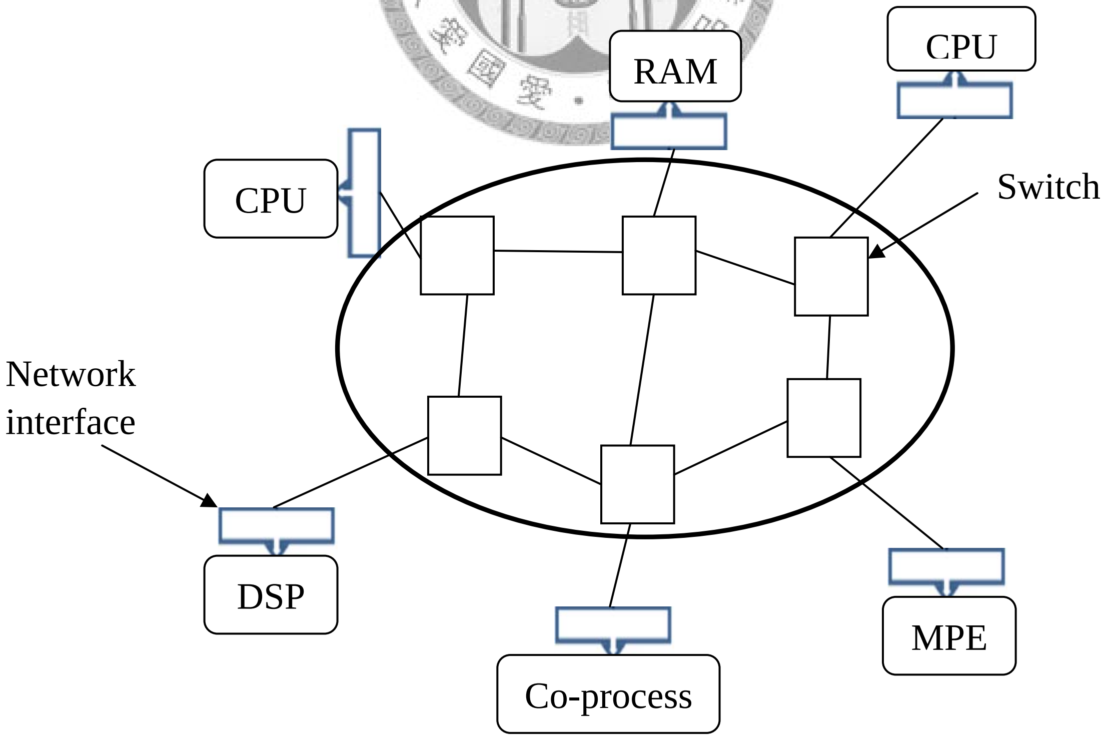


Fig. 1-2 Concept of network on chip architecture

Because of the different operating frequencies of every PEs in NoC and there will be contention in the network during heavy load traffic, we need some buffers queuing the packets in each ports. In the previous research, input buffer takes a significant portion of the silicon area of the NoC router. Not only the area but also the performance are highly related to the buffer size. [2]

More importantly, buffers are the largest part of leakage power consumption and significant dynamic power. Buffers consume approximately 64 percent leakage power of the router. [3] And the buffer energy will increase very fast as the packet flow throughput increases. [4] Nevertheless, the unpredictable network traffic contention may cause the buffer utilization unbalanced. That will waste the important resource of network.

1.3 Thesis organization

The rest of thesis is organized as follows: some backgrounds and related work of Network-on-chip are presented in chapter 2, which includes previous researches on NoC architecture design, NoC topology, routing algorithms, switching techniques, router architecture and router pipeline issues. In chapter 3, we propose our router architecture and our error detection method. Experimental environment and simulation results are presented in chapter 4. Finally, in chapter 5 we summarize the results and conclusion of this thesis.

Chapter 2 Related Work and Background

In this chapter, we will introduce several design concerns, including issues of NoC architecture which are related to our work.

2.1 NoC topology

Network-on-chip architecture consists of several switches, functional blocks and communication channels. This section, we will introduce how we arrange these nodes and connect with each other. Before we design NoC, we have to finish it since all of the following functions and techniques base on it. Such as routing algorithms, flow control, switch technique and router design. Here we will introduce some popular topologies for NoC architecture.

2.1.1 Mesh and torus type NoC topologies

This interconnection topology is presented by Kumar et al. in [5]. Since it is regular and could be implemented easily mesh type topology is the most popular topology. We show a 4 x 4 example of mesh topology in Fig. 2-1. The communications between IPs and switches and the ones between switches are communication channel which consists of two unidirectional links. Besides the switches at the edges, all of the switches have four ports connected to neighbor switches. The number of the switches is the same with the number of the IPs in this topology.

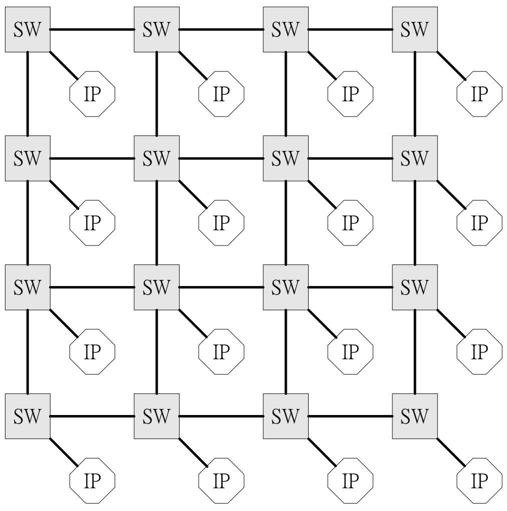


Fig. 2-1 Mesh topology example

Mesh topology has better space locality while transfer data to near nodes but when data transfer to the diagonal node, it will have high latency because of the hop counts. To overcome the drawback, the torus topology is proposed by Dally and Towles. This topology is shown in Fig. 2-2. The biggest difference between mesh and torus is torus topology add some extra links that connect the head and tail of the row and column. So no matter which switch in the system, it always has four links to other switches.

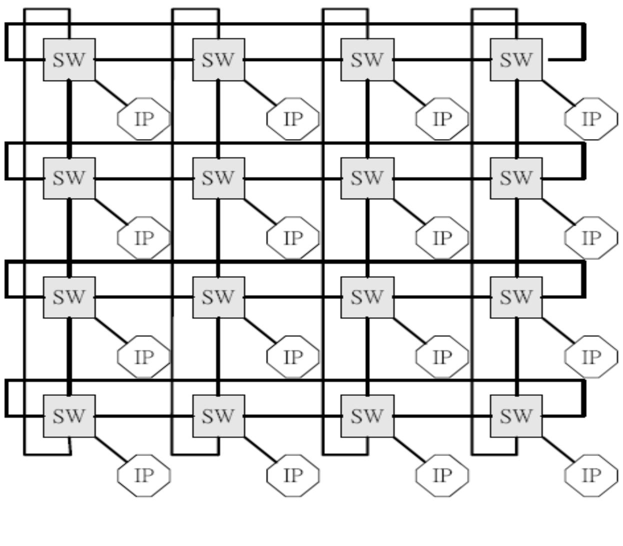


Fig. 2-2 Torus topology example

There is a topology that could solve long wrapping wires problem, is proposed in [6] called Folder torus. It is shown in Fig. 2-3. It rearranges the switches in the Folder torus topology such that the long end-around links are avoided by expense of doubling the length of links.

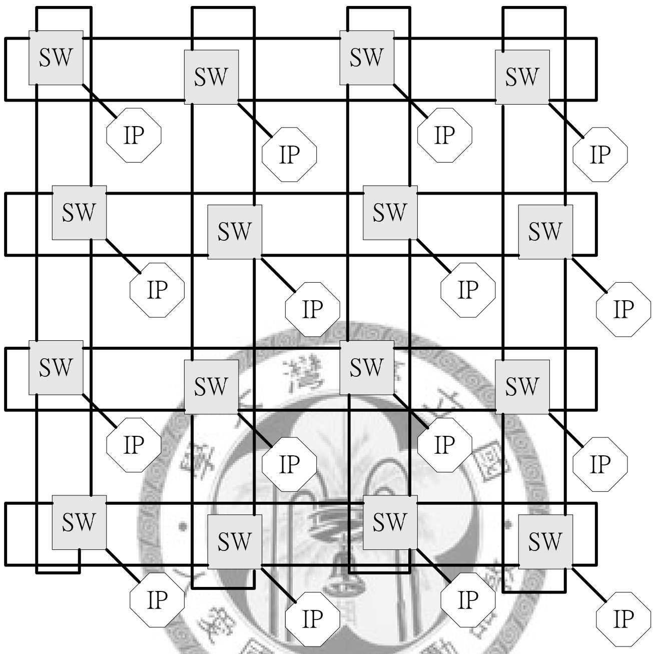


Fig. 2-3 Folder torus topology example

2.1.2 Tree type topologies

The tree type topologies are also commonly used in NoC. An example of balanced binary tree is shown in Fig. 2-4. The main difference between tree type and mesh type topology is every switch has two children nodes and leaf nodes are allowed to connect at most two IPs. The property of this topology is deadlock free since there would not be any cycle in a tree.

\

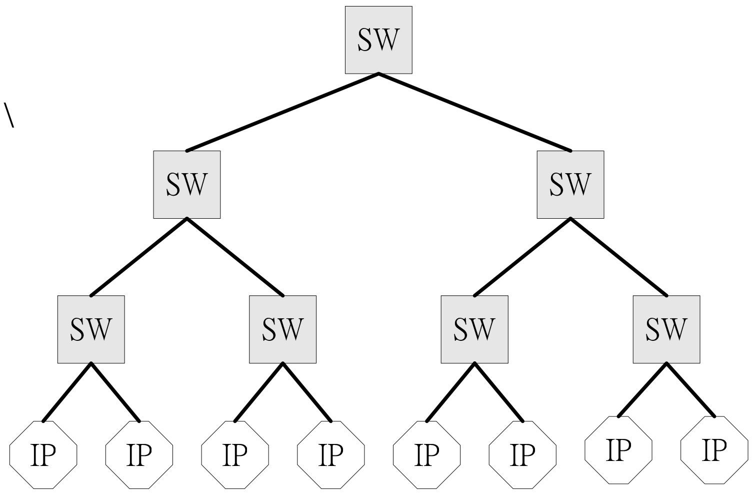


Fig. 2-4 Balanced binary tree example

Scalable, Programmable, Integrated Network (SPIN) is a packet switch interconnection architecture proposed by Guerrier and Greiner in 2000. [7]. It uses the fat tree model. It duplicates its roots to get larger bisection bandwidth, and has not only one path for pair of nodes. The layout of a fat-tree will become more difficult for larger nodes. The example in shown in Fig. 2-5.

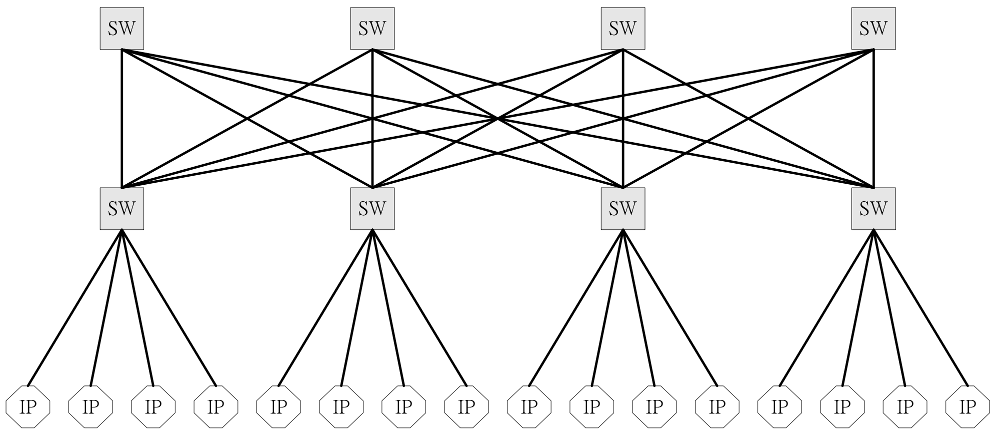


Fig. 2-5 SPIN type topology example

2.1.3 Ring type NoC topologies

The OCTAGON topology is proposed by Karim et al. on 2002. [8] Fig. 2-6 shows an example of OCTAGON architecture with 8 switches and 8 IPs connected by 12 bidirectional links. One of the attractive features in this topology is that only two-hop communication between any pair of nodes is needed. From this remark OCTAGON can be extended to multi-dimension topology easily. However, when the number of IPs grows, the interconnection leads to wiring problem and makes the switch more complex.

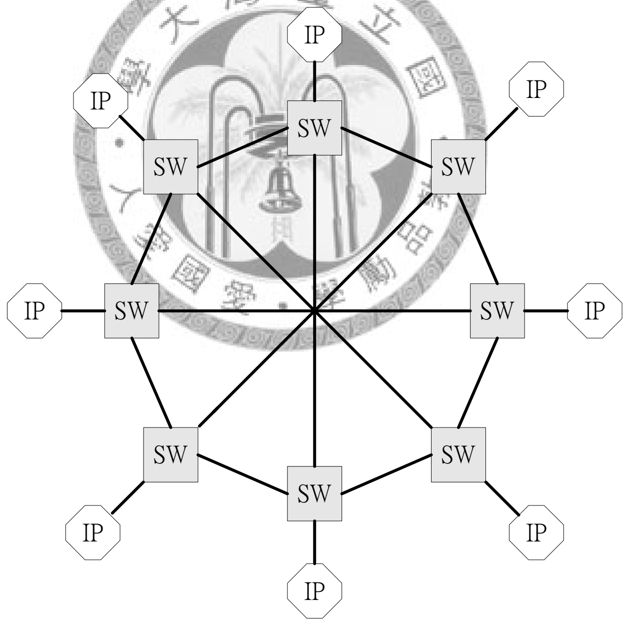


Fig. 2-6 OCTAGON type topology example

2.2 Switch techniques

Switching techniques determine how the packets flow through the routers and the granularity of data transfer. A good switching technique can release the network potential and allocate the network resources efficiently. Depending on the unit of data transferred in a cycle, if we transfer a packet per cycle, we have store-and-forward and virtual cut-through switching techniques; on the other hand, if we transfer a flit (flow control digit) per cycle, we have wormhole switching. Following we will introduce these methods in detail.

2.2.1 Store and forward

In store and forward technique, the entire packet is stored in the input buffer while a packet arrives at a switch. Then the packet is passed to the target switch that is recorded in the header information field when the next output channel is available and the target switch has available input buffer. But there is a drawback in this method, since the packet is allowed to send to the next switch only when the packet is completely received at switch, so the network latency will be high.

2.2.2 Virtual cut through

The difference between Virtual cut through method and store and forward method is that unlike store and forward method, virtual cut through method does not need waiting the entire packet to arrive current switch. The transmission is started right after the routing process is completed. By this mechanism, the network latency could be reduced. As same as store and forward method, it also needs each buffer with capacity of one or more packets.

2.2.3 Wormhole

Wormhole switching is quiet like virtual cut through. It differs from virtual cut through in the size of flow control unit. Wormhole switch reduces the flow control level form packet to flit.

2.3 Routing algorithm

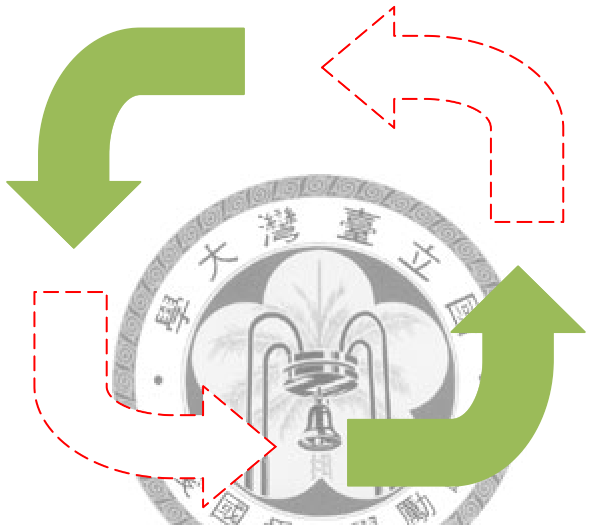
As soon as we decide the topology of our NoC design, because routing strategy is highly dependent on topology. It is important to choose the routing method. Routing algorithm includes the path from source node to destination node. Moreover, routing is one of the key factors to take good advantage of the network potential.

Because choosing routing algorithm for NoC is so important, we have to know some routing algorithm issues. Balancing the link load through the network is one of issues. If we balance the link load, the performance of the network will approximate ideal performance. But the other issue, keeping the traverse distance short conflicts to the previous design issue. If our routing method focuses on keeping the routing path as short as possible, it would cause the channel unbalanced in the network. This is the tradeoff between shorter pass and load balance which we need to consider. Last but not least, a significant topic about routing, is the routing algorithm deadlock-free or not? The deadlock would affect the performance and correctness of network and consequently we must examine this problem. In the light of the popularity of mesh topology in NoC, we introduce some famous deadlock-free routing algorithm for 2D mesh as follows.

2.3.1 XY routing

The policy of xy routing is to route the packet on the X axis first. After packet arrives the node that is at the same X axis with destination then route the packet on the Y axis. It could be used in mesh and avoid deadlock condition. The reason that xy routing could avoid deadlock condition is shown in Fig. 2-7. There are two forbidden turns so cyclic transfer would not appear. The disadvantage is that channel resource is not chosen fairly because xy routing method does not concern the traffic condition and just route the minimal path. But since the hardware implementation is very simple. It is often used in NoC design.

-



國立臺灣大學 · 愛國 · 愛學 · 勵

Fig. 2-7 The turns are solid is allow in xy routing and the turns with dotted line are forbidden

2.3.2 West first routing

On the contrary, west-first routing [9] is an adaptive routing algorithm and it provides non-minimal paths but also deadlock-free. The basic concept of this routing is that packets go west first and then those packets could adaptively go east, north or south. West-first routing restricts the turn to the west as depicted in Fig. 2-8 to overcome the deadlock problem.

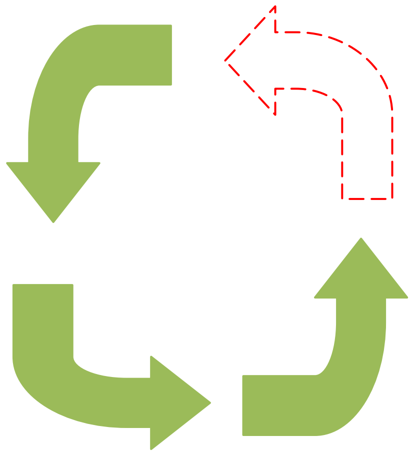


Fig. 2-8 The turns are solid is allow in west first routing and the turns with dotted line are forbidden

2.3.3 Odd-even turn model

The same as above, the odd-even turn model [10] is another adaptive routing method for 2D mesh. It can not guarantee the shortest path either, but it convinces of deadlock-free. The difference from resolving deadlock problem is that the restricted turns is related to where the packets located.

The following are the two rules of odd-even turn model:

Rule 1: Any packet is not allowed to take an EN turn at any nodes located in an even column, and it is not allowed to take an NW turn at any nodes located in an odd column.

Rule 2: Any packet is not allowed to take an ES turn at any nodes located in an even column, and it is not allowed to take an SW turn at any nodes located in an odd column.

2.4 Router architecture

A conventional router is composed of input/output channel, buffer, routing computation circuit, arbiter and crossbar. Fig. 2-9 shows the connection between these functional blocks. Input/output channels are the paths between the output ports of router and input ports of the neighbor router. Generally, there are five sets of input/output channels which connect to east, west, north, south and local different ways outers. Every direction has individual simple buffers, usually FIFO (First in, first out) registers, to queue the received packets or flits. Routing circuit decodes the header information and computes the path based on the routing algorithm which we decide in the previous design flow. Arbiter collects the determined output requests from routing computation circuit and arbitrates these requests when there are more than two input ports asking for the same output ports. Crossbar is configured by the grant signals from arbiter to transfer the packets from input buffers to the target output channels.

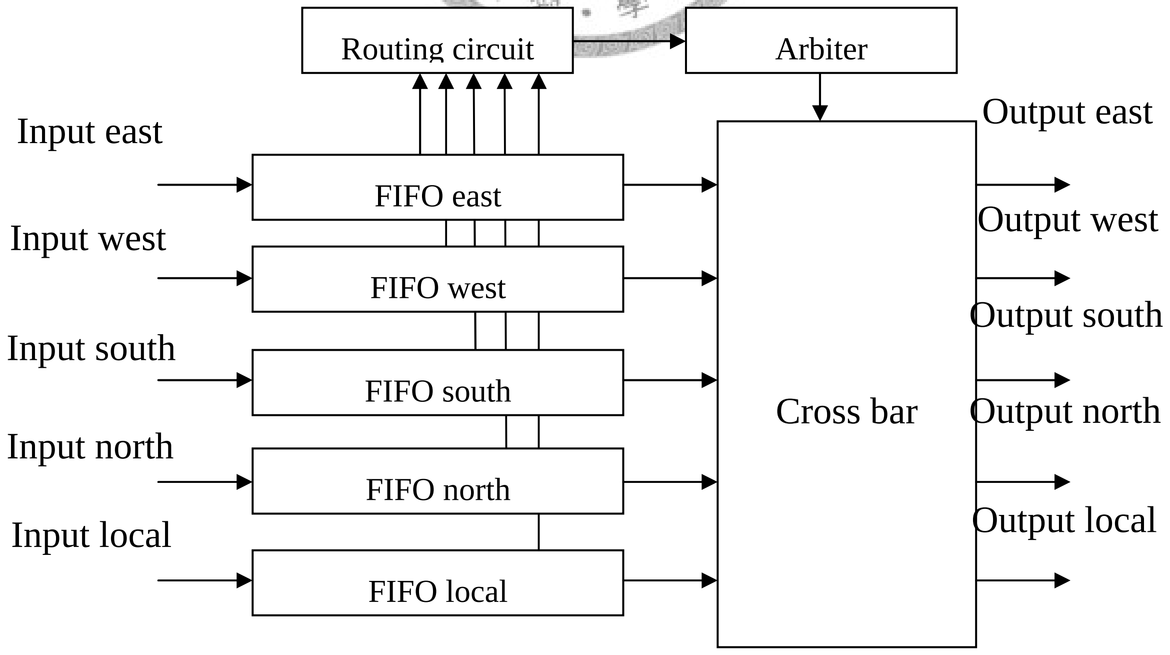


Fig. 2-9 Basic router architecture

2.5 Router pipeline

In this section we introduce the wormhole router pipeline stage proposed in [11]. Most NoC routers are pipelined at the flit level. Consider a packet divided into four flits, one head flit, two body flits and one tail flit and this packet traversing from east input channel to the north output channel. The head flit must pass through the stages of buffer write (BW), routing computation (RC), switch allocation (SA) and switch traversal (ST).When the head flit arrives at east input channel, storing this flit takes one cycle called BW. After that the routing circuit will decode its flit type, detect as a head flit and send the destination file saved in header to the routing logic. This stage called RC in the Fig. 2- 10 during cycle 2.

Routing logic will return the output port requests for each input packet, in this case the north output port during cycle 3. At the moment, the pipeline stage is going to SA. These requests for output ports will collect and send to arbiter. The arbiter arbitrates these requests and assigns available output ports to the requestors. In our example, the arbiter grants the north output channel to the east input channel and marks north channel as unavailable for other packets before the tail flit passes by.

After receiving these grant signals from the arbiter, the pipeline stage goes to ST. The crossbar sets the transfer paths by those grant signals and the head flit is read from the input FIFO to the input of crossbar. In our cases, the crossbar has connected the north output channel for the east input channel. The head flit traverses through the crossbar and sends to the next router during the cycle 4. When the following body flits and tail flit received by the buffer queue, the routing circuit checks their flit type which are not of head and need not go through the RC stage. Body flits and tail flit follow the path reserved by the head, in our example north. Fig. 2-10 illustrates these actions during cycle 5 to cycle 7. After the tail flit departs the crossbar, the preserved output channel can be released.

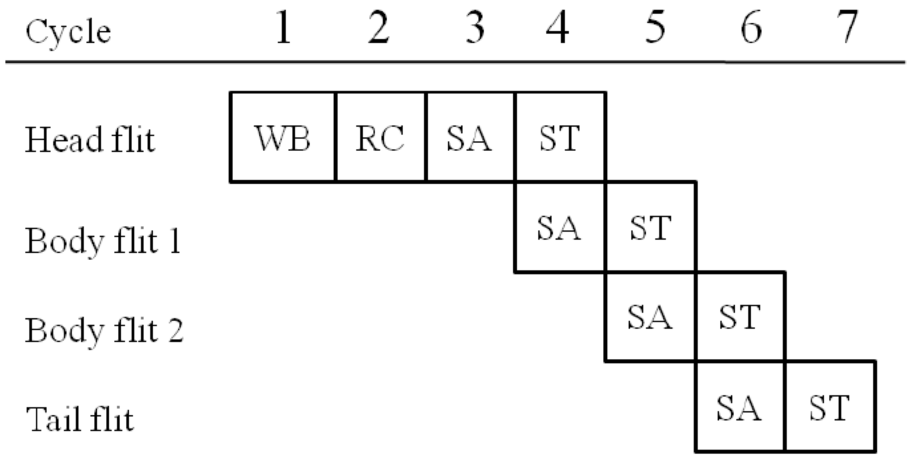


Fig. 2-10 The example of wormhole router pipeline stage

2.6 Virtual channels

Virtual channel flow control is a common used method for increasing the network throughput. Conventionally, a single FIFO buffer is associated with each channel in a NoC router. Sometimes, it may suffer from the head-of-line blocking, for example, a blocked packet *A* at the head of the queue will prevent the other innocent packet *B* in the same queue from progressing, just because packet *B* is not at the head of the queue. By providing multiple buffers (lanes) for each channel, they form virtual channels which still share bandwidth of the original channel (physical channel). As the above-mentioned example, the packet *B* can choose another virtual channel for progressing in the same situation. Additionally, virtual channels can be used for supporting quality of service or avoiding deadlock. However, the implementation of routers with virtual channels is more complex due to more control logic for virtual channel arbitration.

Moreover, Fig. 2-11 shows another example of how virtual channels relieve the blocking problem in wormhole routing. In (a) packet *A* can not use the idle channel between node 1 and node 2, because the channel is held by packet *B* on node 2 even though it is blocked and not using the bandwidth. In (b), with two virtual channels, packet *A* can acquire the second virtual channel on node 2, which associated with the same physical channel for packet *B*.

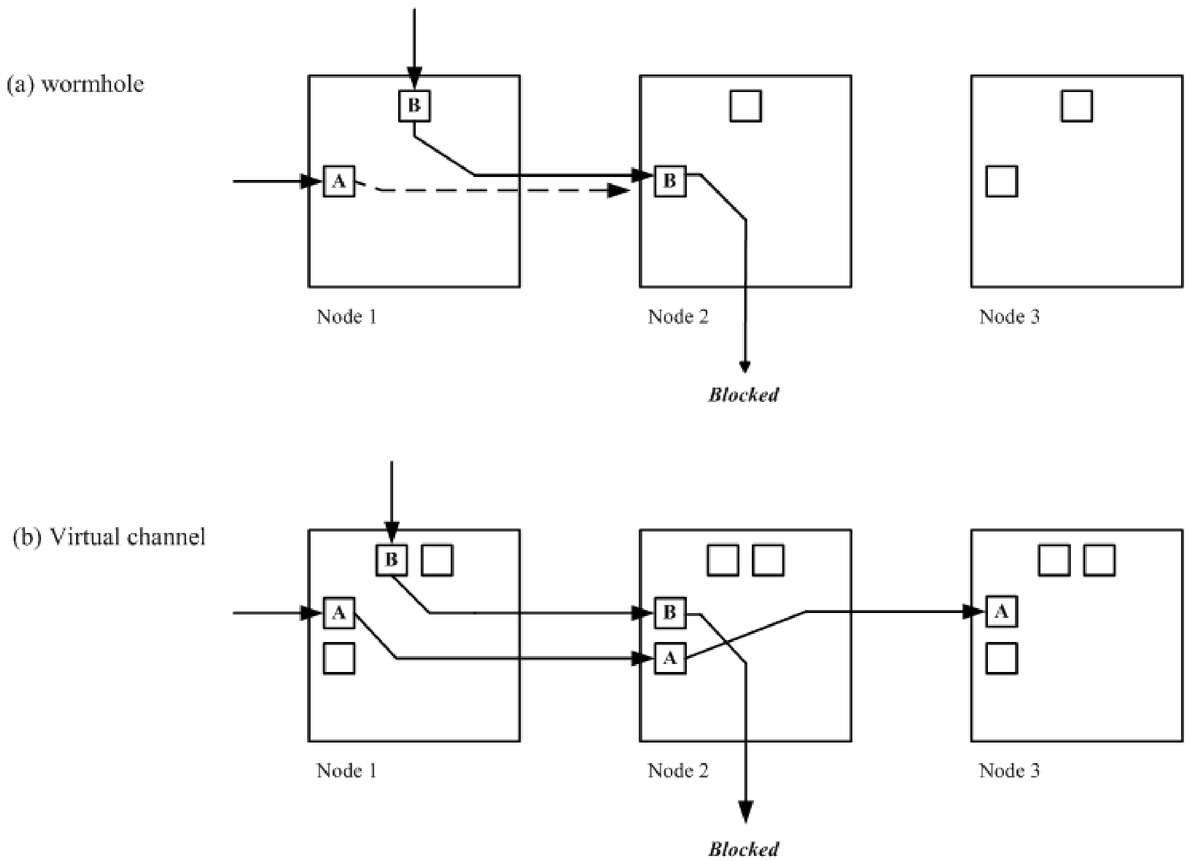


Fig. 2-11 Virtual channel examples

Chapter 3 Proposed Method for Error detection and proposed router architecture

In this chapter, we propose our error detection method and a new router architecture for our method. Our main idea is choosing the critical flits and gives them higher priority. When flits that have higher priority traversing at their routing path, error detection will enable at each node (router). When flit that has lower priority passing a node, error detection will not enable until it reaches the destination of its own.

The chapter is arranged as follows: we discuss the motivation which is the significance of error detection performance and design cost in a NoC router first. Second, we compare the related work in the error detection design with our approach. Continued are our router architecture and hybrid error detection in detail.

3.1 Motivation

When the chip size increases, the importance of error recovery mechanism also increases. At the same time, the feature size is shrinking too. This causes that the supply voltage decreases, and wire resource becomes much more sensitive by some noise signal than before. Crosstalk is the main producer to the noise affecting the interconnect transmission due to the supply voltage decreased. So the information and data transmission on the wire is not as reliable as before. Using error recovery mechanism could self-heal to move the data information or system from error state to correct state.

Different error recovery schemes cause different performance, area and power. So how we choose a suitable error recovery scheme is a main issue while we design a fault tolerance network on chip system.

Error recovery could be divided into two parts, each with distinct function. The first is error detection and the second part is error correction. In fact, retransmission mechanism is the conventional method to achieve error correction.

Choosing different error detection has a great effect upon the performance, power consumption and device area. So next we discuss this part and propose a new error detection mechanism.

3.2 Related work in fault tolerant router design

To ensure the correctness of data transition, error detection (ED) in the error recovery (ER) mechanism is needed. According to the error detection, two common types of error detection method are as below: Switch-to-Switch (S-S) [12] and End-to-End (E-E).[12]

However both kinds of error detection have some drawback. For example the S-S costs a large number of the area, and E-E has lower throughput and high power consumption when error rate becomes higher. So we propose a new error detection method and a new router architecture to fit our error detection method. Next, we will describe the advantage and disadvantage of the conventional router architecture in detail.

3.2.1 S-S error detection router

The characteristic of the S-S router is that the error could be detected immediately and retransmit the data right away after error is detected. The architecture of the S-S router is shown in Fig. 3-1. With this advantage, S-S router must have extra buffer that is reserved for ensuring retransmission mechanism could take place with flit in the reservation buffer. The reserved buffer is shown in Fig. 3-1 with red color. Another advantage of S-S router is flits would be checked every time when they reach a new hop at the trail of routing path. To achieve this, error detection circuit must be placed at each port. The decoder is filled with purple and coder is filled with yellow. Precisely, we have to put encoders at each output port to encode error information data except local output port and put decoders at each input port except local input port to decide whether the arrived packet is correct or not. In this way, there are four physical ports in a conventional router that means we will need four decoders and four encoders. In fact, these reservation buffers would be the increased part of area and power consumption of the whole router.

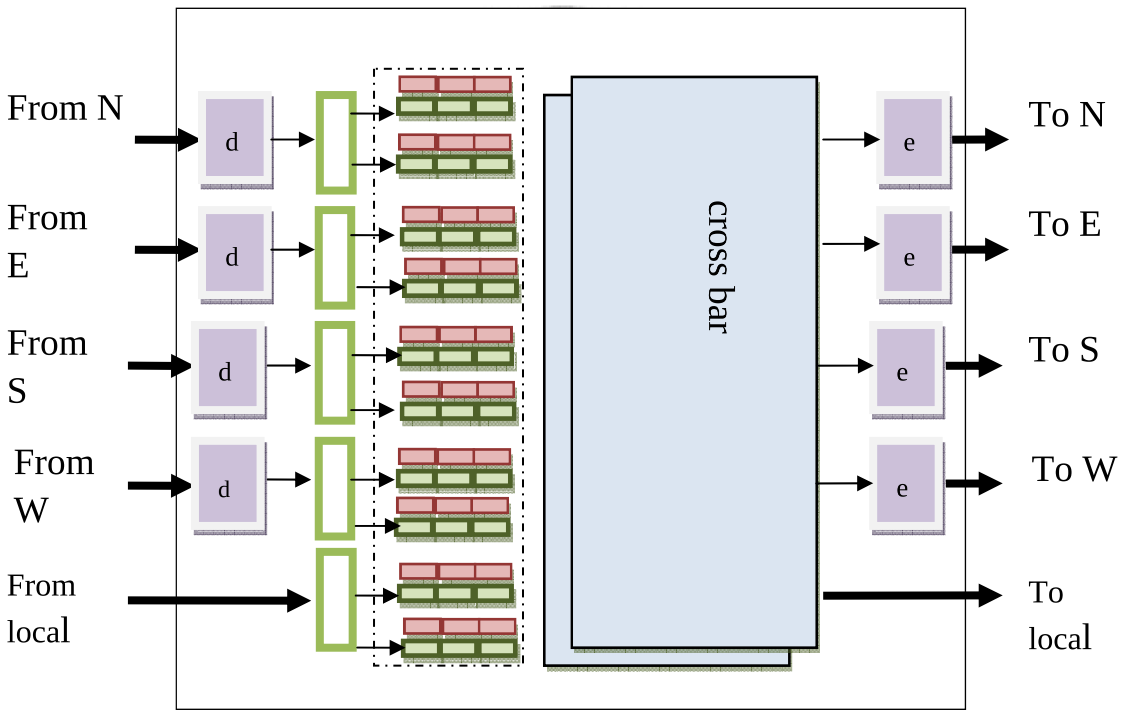
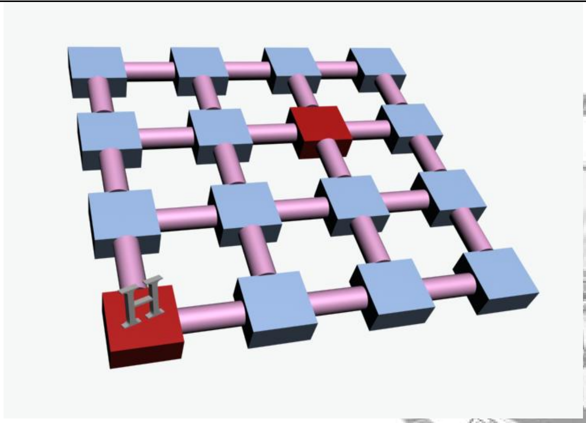
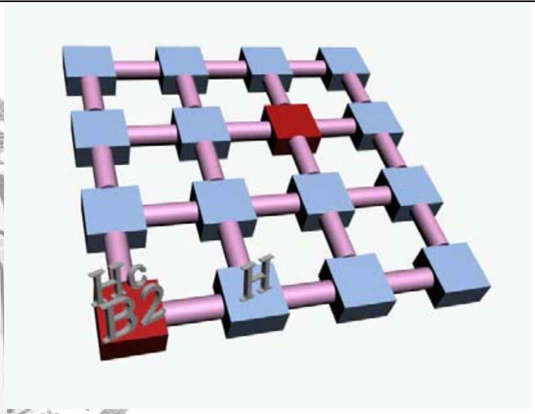
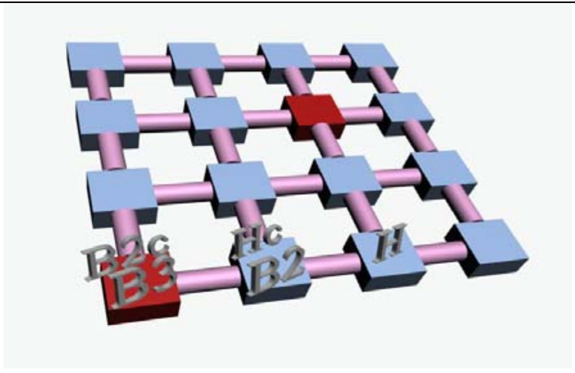
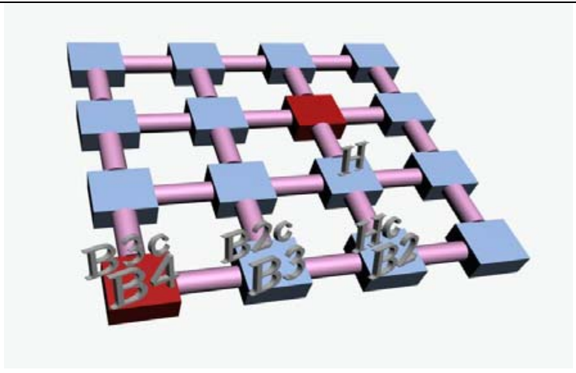


Fig. 3-1 S-S router architecture

In Fig. 3-2, we present the operation of S-S mechanism in a 4 by 4 network on chip system. In this figure, H means head flit. B means body flit and T means tail flit.

We use a 8 flits per packet format to demonstrate this method.

Assume a packet is traversing from source (0, 0) to destination (2, 2) and the second flit will bring an error at node (2, 1). As the S-S requirement, every flit must be copied while it passing a node. When B2 is going to (2, 1) from (2, 0), there is an error occurring. So node (2, 1) will drop the fault flit, b2 and sends an *nack* signal to node (2, 0) then node (2, 0) will send the copy of flit B2 to (2, 1) again. By this mechanism any fault occurred in any flit will be eliminated.

 **step 1**	 **step 2**
 **step 3**	 **step 4**

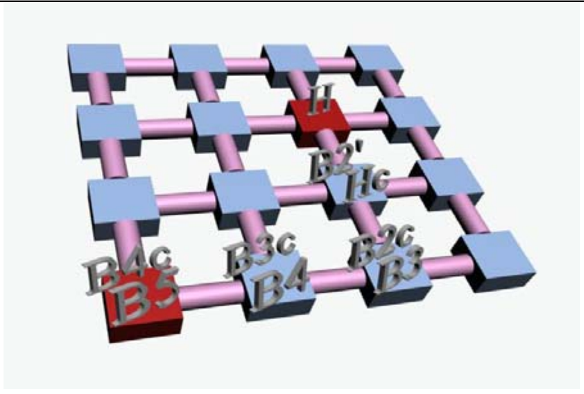
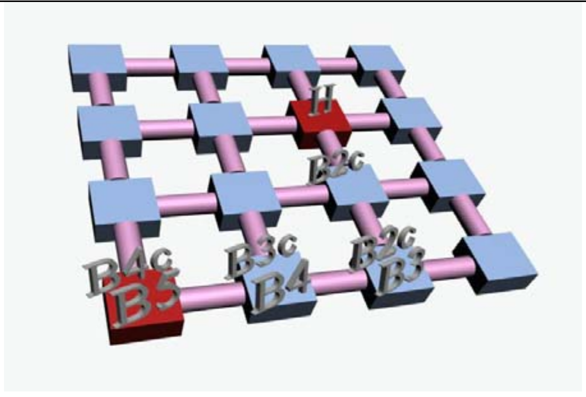
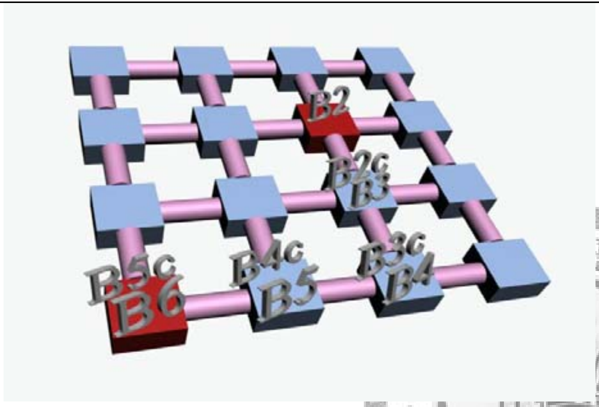
 **step 5**	 **step 6**
 **step 7**	

Fig. 3-2 S-S router transmit step

3.2.2 E-E error detection router

The characteristic of the E-E router is that packet need not do any error detection on the routing path until packet reaches the destination node. The architecture of E-E router is shown in Fig. 3-3. Since E-E router only detects error at the end it could put just one encoder (the yellow one) at the local input port and put one decoder (the purple one) local output port. Unlike S-S router, E-E router does not need extra buffer to store the temporary copy of data to achieve retransmission. Since the most area overhead of a router is composed of buffer, the E-E router is much smaller than the S-S router. At the same time, power consumption is smaller than the S-S router.

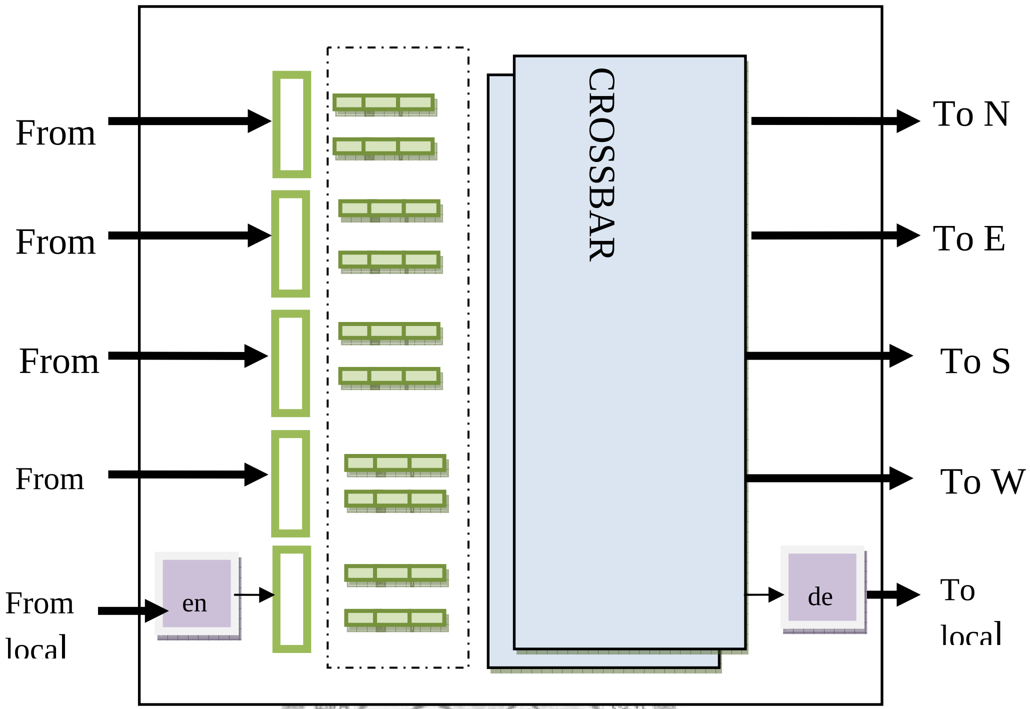
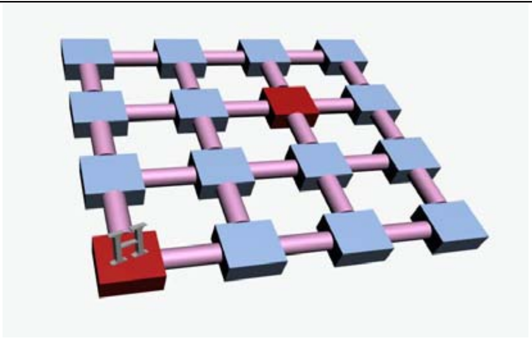
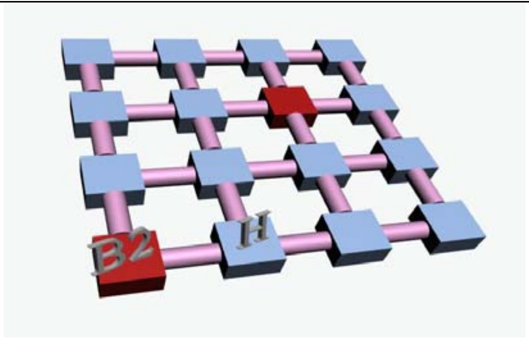
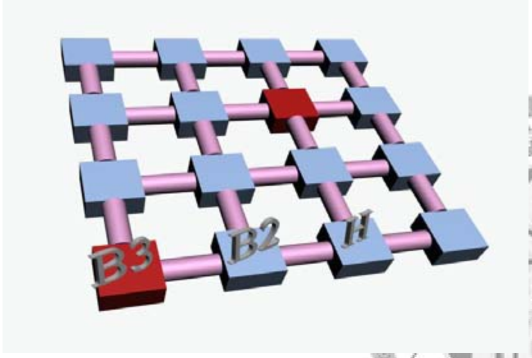
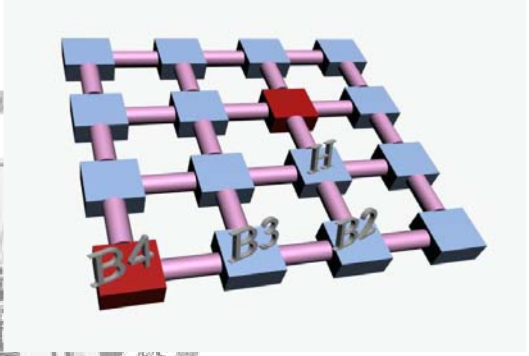
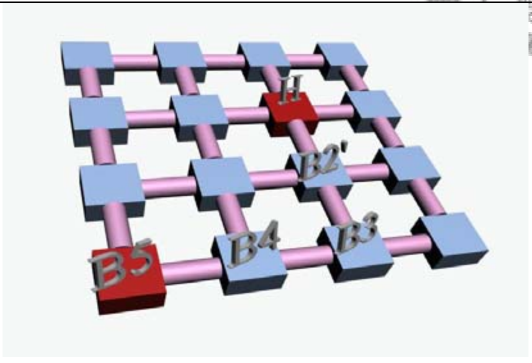
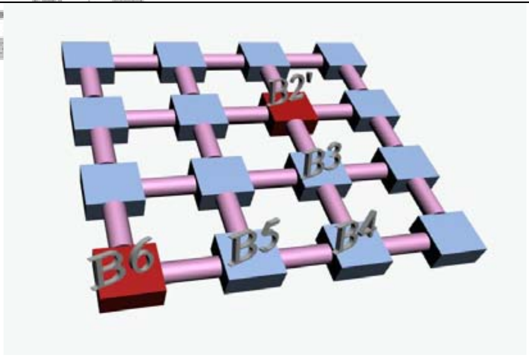


Fig. 3-3 E-E router architecture

In Fig. 3-4, we present the operation of E-E mechanism in a 4 by 4 network on chip system.

Assume a packet is traversing from source (0, 0) to destination (2, 2) and the second flit has an error at node (2, 1). While flit B2 is traversing from node (2, 0) to node (2, 1) there is an error occurred. Unlike S-S mechanism, node (2, 1) will operate to pass this faulty flit to its next hop, node (2, 2) still. When the faulty flit is getting to the destination the error detection will start. Since this packet is not correct, the destination node will send a retransmission signal to the source of the packet. In fact, this transmission is route as the normal data packet but we just give this retransmission signal packet another special format to let router recognize it.

While the source node (0, 0) receives the signal packet it will begin to resend the faulty part of the original packet. Here, the fault flit is B2 so the resent packet will consist of just three flits, head, B2 and tail.

 **step 1**	 **step 2**
 **step 3**	 **step 4**
 **step 5**	 **step 6**

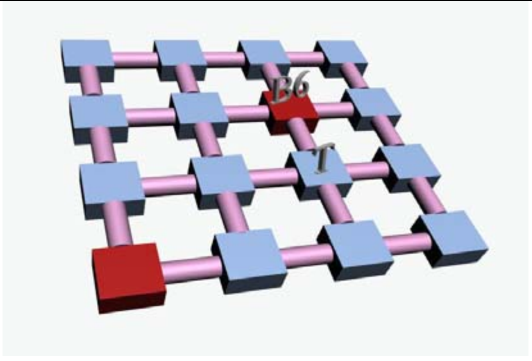
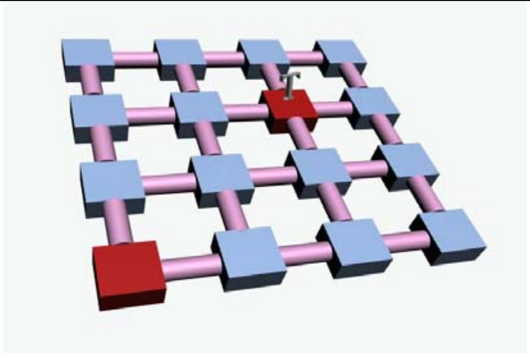
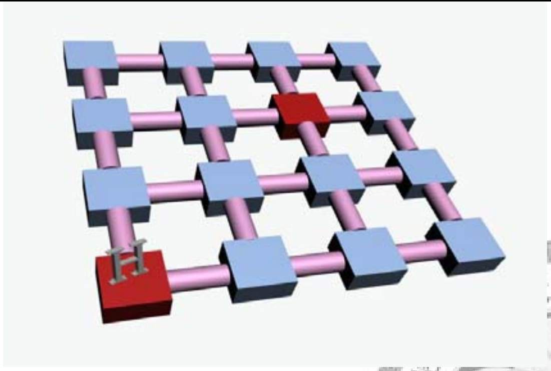
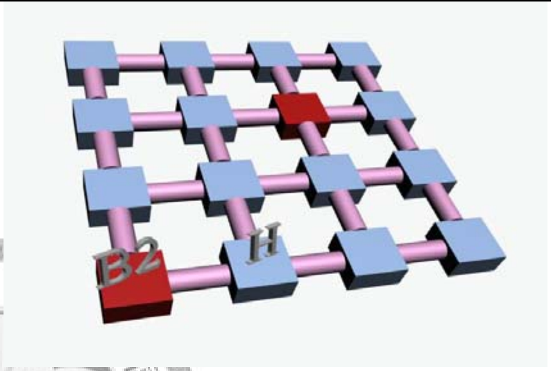
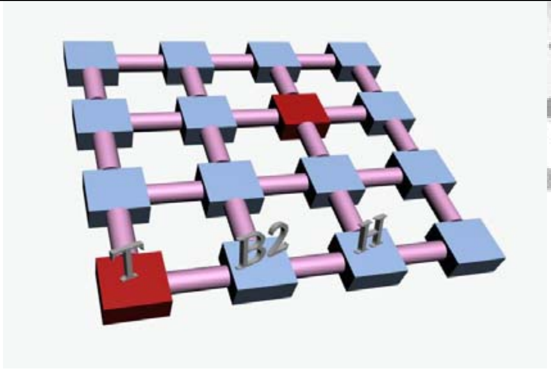
 **step 7**	 **step 8**
 **step 9**	 **step 10**
 **step 11**	

Fig. 3-4 E-E router transmit step

3.3 Proposed error detection

We provide a design method for NoC error detection with the advantage of both

S-S and E-E methods, which can reduce latency and router area. Our basic idea is trying to separate the importance of the flits that packet consists of. All the information that packet traversing on the NoC needs is in the head flit, including packet type, source/destination address, packet/flit size and routing path information. But in Fig. 3-6, only pure data and little packet information is in payload (body and tail) flit. By contrast, head flit is more important than payload flit for the NoC system. Thus we have to ensure the correctness of head flit to avoid error occurring to degrade the system performance.

So here we check head flit at each node, in this way we hope if there is any error at the head in each node we could detect it immediately and the other flit (payload flit) would not cost extra time and power to do error detection.

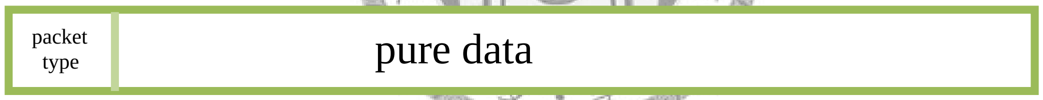


Fig. 3-5 Payload packet

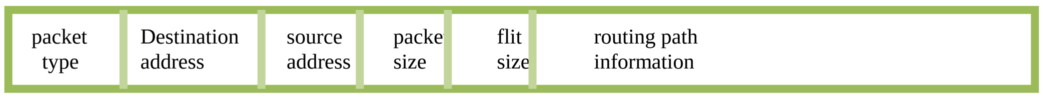


Fig. 3-6 Head packet

3.4 Proposed router architecture

Here, we are going to describe our fault tolerant router with error detection ability. Since our design principle is doing both S-S and E-E error detection at one system which using the suitable error detection method for different flit types. So we have to combine the advantage of these two methods.

Comparing S-S with E-E, the area of S-S router is much larger than E-E router because of the decoder/encoder. In UMC 0.18um, decoder/encoder area is about one quarter of whole router. It is quiet large. Unlike S-S router, E-E router only needs two extra coders/decoders. So consider area problem, E-E method will be the better choice.

Then consider the E-E router, there is a problem in the encoder and the decoder of the local in/output. The utilization rate of these two units is low. The utilization of distinct flit injection rate is shown in Fig. 3-7.

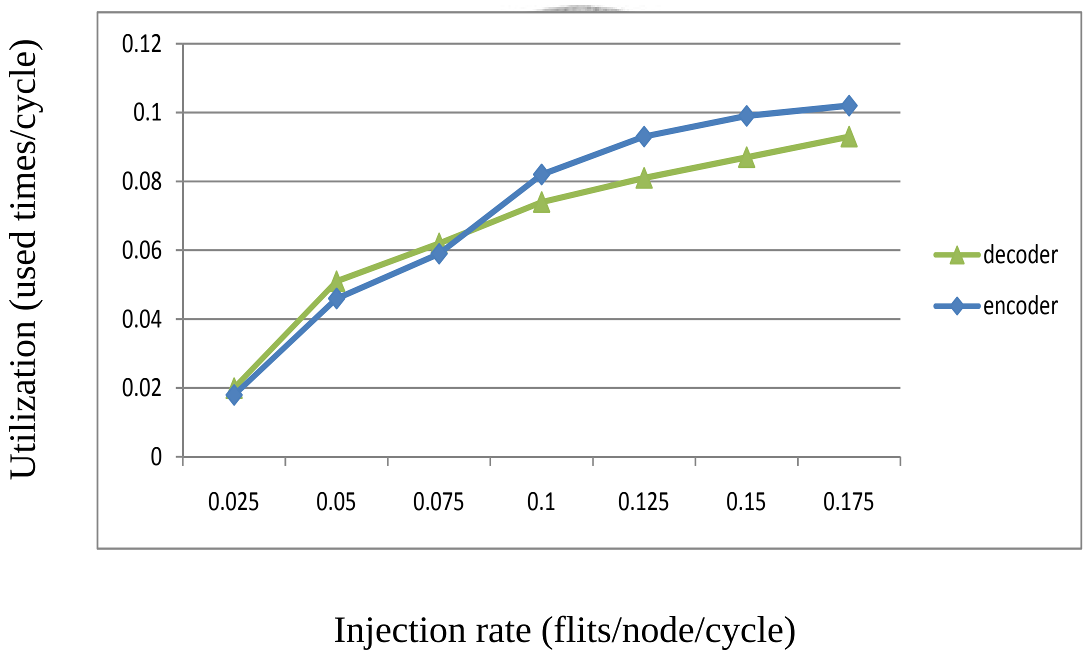


Fig. 3-7 Utilization of decoder and coder

As result, the utilization of these units is not effective. To improve this we propose a new router architecture. Since the design goal is to fully utilize the encoder and decoder. Our proposed router is modified from the E-E router. The reason that we do not follow the S-S type router architecture is the high power consumption and large area.

Considering the utilization of the coder and decoder, in Fig. 3-8, we build a conventional network-on-chip system with HDL [13-15]. At a 12 by 12 topology the throughput of the system is about 0.175, this means if the system injection rate is over 0.175 flit/cycle/node, this system will be jammed by the communication data packet.

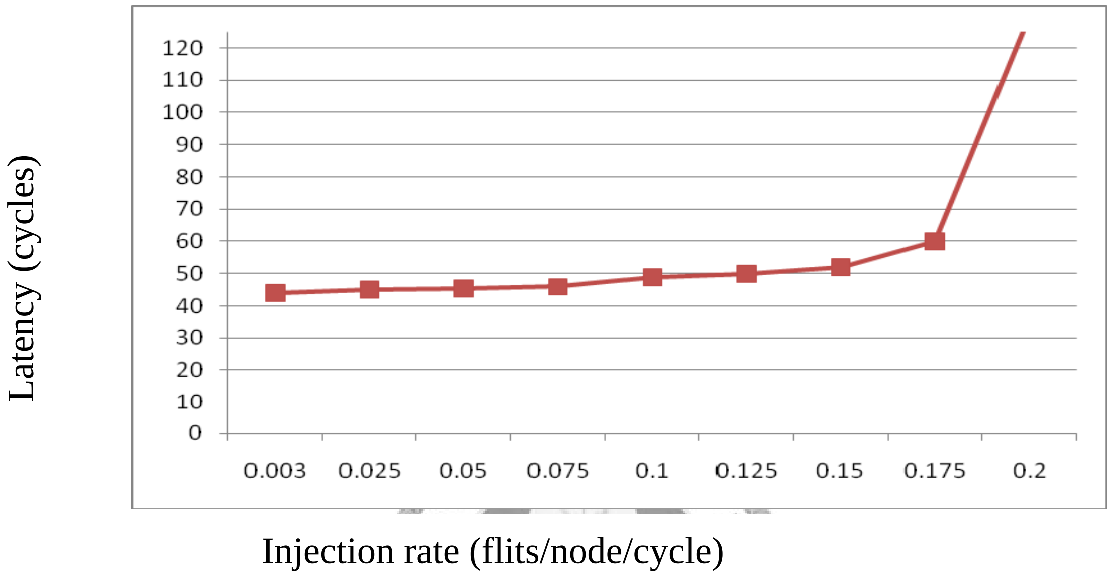


Fig. 3-8 injection rate V.S. latency

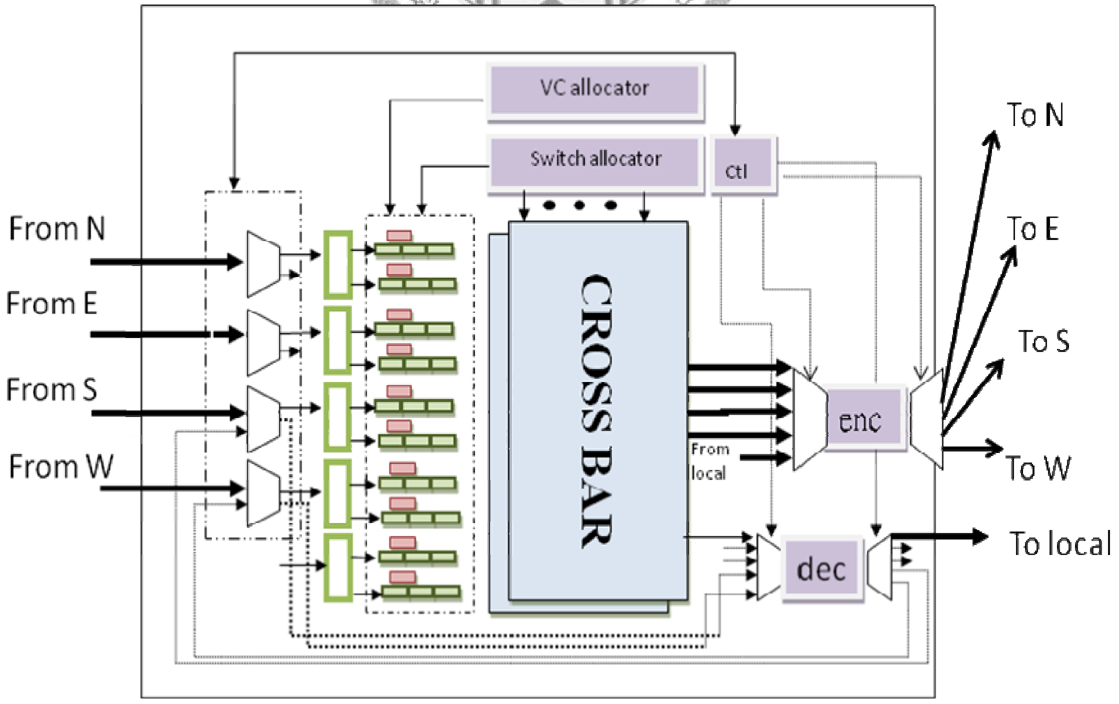


Fig. 3-9 proposed router architecture

Note that the meaning of 0.175 is while the system reaches the highest tolerable injection rate, the utilization of encoder and decoder is still low (encoder is about 0.11 decoder is about 0.095) so we could use these two units when they are not utilized by the injected packet or ejected packet. The router architecture is shown in Fig. 3-9.

As in our architecture, we only need one extra buffer to store the copy of the head flit rather than the whole virtual channel in S-S router. But the control circuits will be a little complicated to accomplish the flit control. Although the area of control circuits increases, the size of our design is still smaller than the S-S router because buffer is the major portion of the size of a router. Here, our proposed router is 37 percent smaller than the S-S type router and six percent bigger than tge E-E type router because of the additional buffer and the modified control circuit.

3.5 Additional buffer control method

Since we use an additional buffer to record the head flit to support retransmission while an error is detected at head flit. This buffer is just for head flit so when a head flit passes the error detection in next hop, this buffer is not needed any more. So how to use this additional buffer effectively is an important issue. Here we use reuse method to complete this goal. The mechanism is shown in Fig. 3-10.

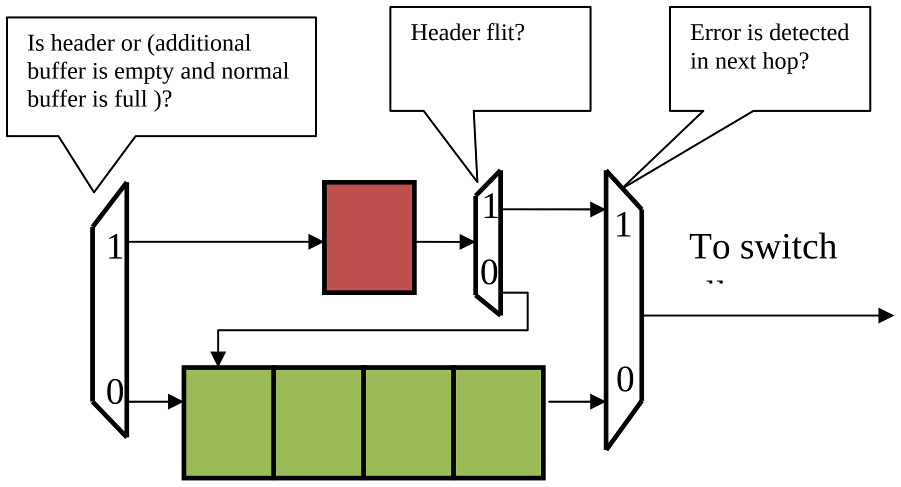

Is header or (additional buffer is empty and normal buffer is full)?
Header flit?
Error is detected in next hop?
1
0
1
0
1
0
To switch

Fig. 3-10 Additional buffer utilize method

國立臺灣大學 愛國 · 愛人 · 學勵 · 品敦

Chapter 4 Experimental result

In this chapter, we present some simulation results of comparing our proposed error detection method with the E-E and S-S methods.

4.1 simulation infrastructure

The router architecture we described in chapter 3 is implemented by verilog hardware description language (HDL) and synthesized by Synopsys Design Vision with UMC 0.18 um CMOS cell library.

Our packet format is shown in Fig. 4-1 and we use 34 bits to implement. The [33:32] bits means the packet type. 11 means head flit, 10 means body flit and 01means tail flit and bits [15:8] is the source location [7:0] stands for destination location. The entire experimental configuration environment is shown in Table 4-1.

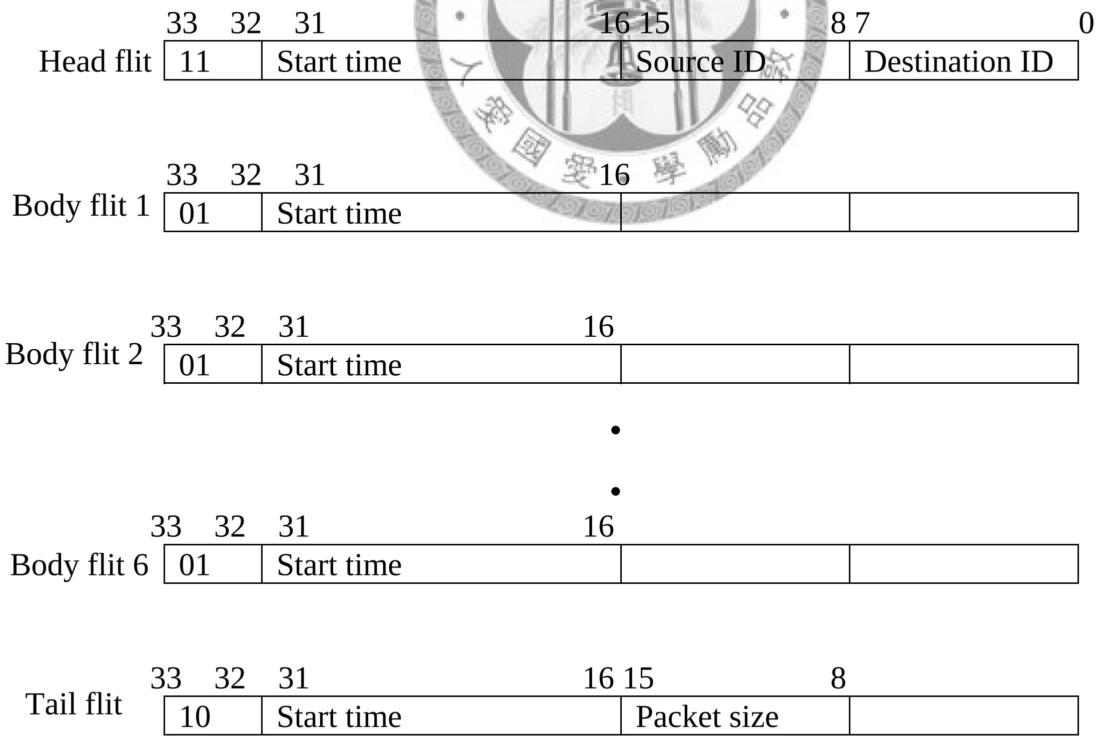


Fig. 4-1 Packet format in experiment

Table 4-1 Router configuration of simulation

Entry	S-S router	E-E router s	Proposed router
Topology	Mesh	Mesh	Mesh
Routing algorithm	XY routing	XY routing	XY routing
Switch Technique	Wormhole	Wormhole	Wormhole
Arbitration policy	Round robin	Round robin	Round robin
Flow control	On/off flow control	On/off flow control	On/off flow control
Data width	34 bits	34 bits	34 bits
Packet length	8 flits	8 flits	8 flits
Pipeline stage	4 stages	4 stages	4 stages
Operation frequency	200MHZ	200MHZ	200MHZ
Error detection method	Check sum	Check sum	Check sum
Number of error detection units	8	2	2

We investigated two of the most important network metrics for measuring performance, latency and throughput. And we design a new component for packet injection and ejection to replace the real hardware IP cores in this simulation.

For computing the latency and throughput of the traffic, we implement two extra modules, packet generation and packet ejection module. In packet generation module, we produce packet, according to the traffic type. Then put time information in the start time field of the flit. With packet ejection module, we collect the start time of the packet and count the amount of packets for computing the system throughput. The simulation environment is shown in Fig. 4-2.

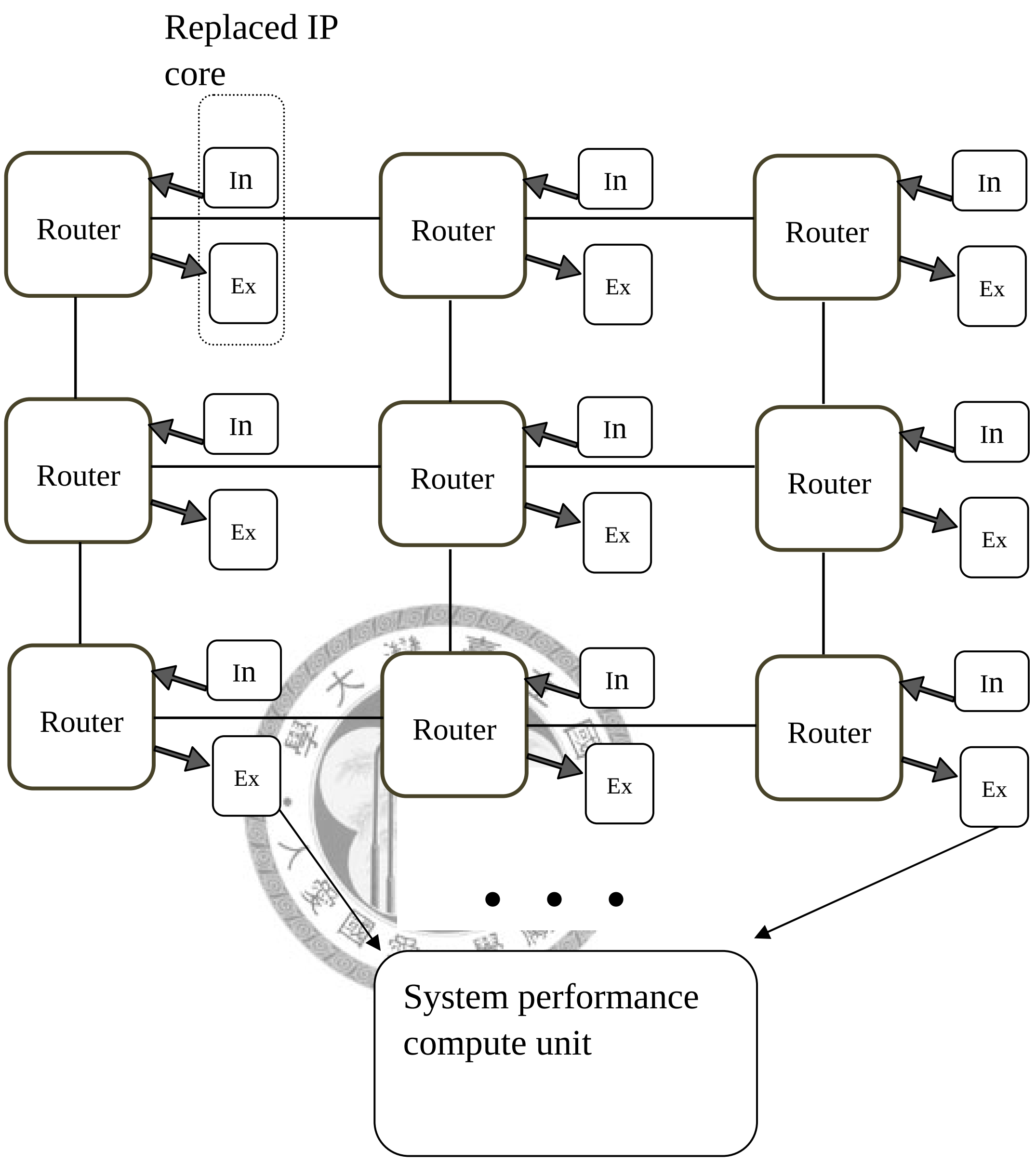


Fig. 4-2 Simulation environment

Because we focus on error correction, we put the error generation unit on each physical output ports to generate error. By tuning the error generator, we could get different error rates on each experiment. The schematic figure is shown in Fig. 4-3.

To ensure our experiment get result correctly, we assume the "flit type" field would not get any errors, otherwise our system will not work correctly in some condition. For example, with E-E router if a head flit gets a bit-flip error it will become an un-recognized flit. At this situation we could not route it, not to mention detecting it when it get to its destination.

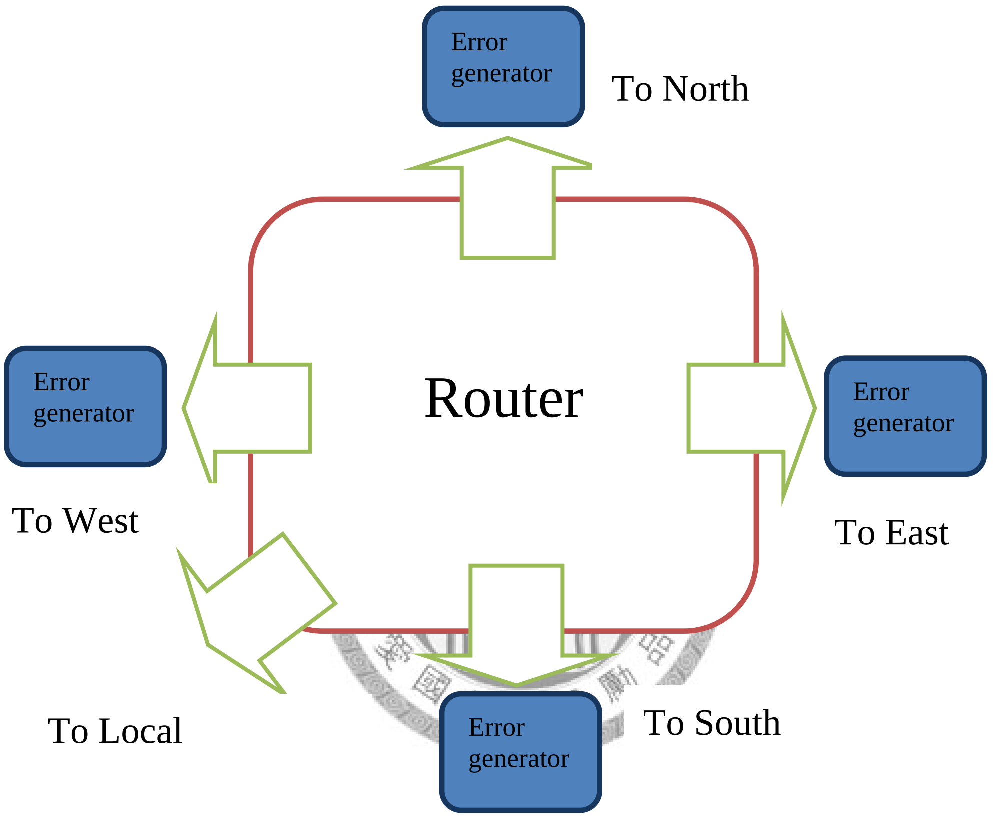


Fig. 4-3 Error generating model

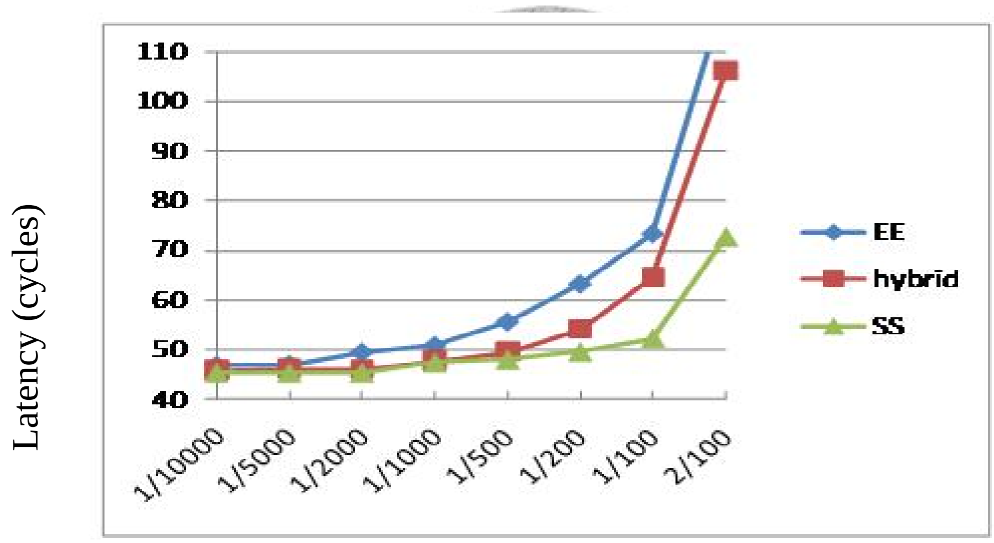
4.2 Experimental results

There are three parameters in our packet generation module. Injection rate, destination address and error rate. The injection rate is a constant rate of flits generation, and we queue the flit in source buffer until they are able to enter the network. We generate packets between random intervals. For experiment reason, the traffic type is not what we concern. So we use Uniform Distributed traffic that is the most common type of traffic in conventional NOC experiment. It means the destination node is selected randomly.

4.2.1 Area result

We choose UMC 0.18um CMOS technology library to synthesize our design with 200 MHz. The size of E-E router is 329625 um^2, the size of S-S router is 489730 um^2 and the size of our proposed router is 354960um^2.

4.2.2 Performance result



Error rate (bit error/flit/node)

Fig. 4-4 Performance comparison

Fig. 4-4 shows the average latency in clock cycles as a function of the error rate when using Uniformly Distributed traffic and the injection rate is set at 5 (flits/cycle/node). As the result, under low error rate for example 0.001%, different kinds of fault tolerant routers have almost thr same latency. This is because under low error rate, all of the three routers work as normal router that means it does not have to spend extra cycle to do retransmission at most situations. When the error rate becomes larger, S-S router will have lower latency because it checks every flit at every node and retransmits fault flit right away after an error is detected. So it will just cost channel resource slightly unlike E-E router.

Compared our proposed router, when error rate becomes larger, our proposed router gets better latency performance than E-E router because the error occurs in head flit would influence the latency greatly by wrong-routing. When packet is at wrong routing status, packet will occupy routing resource such as link and buffer. But S-S router still has better latency performance than both of E-E and our proposed router if we do not count the ratio of power consumption in.

In Fig. 4-5 we show the power consumption under different error rates when injection rate set at 5 (flit/cycle/node).

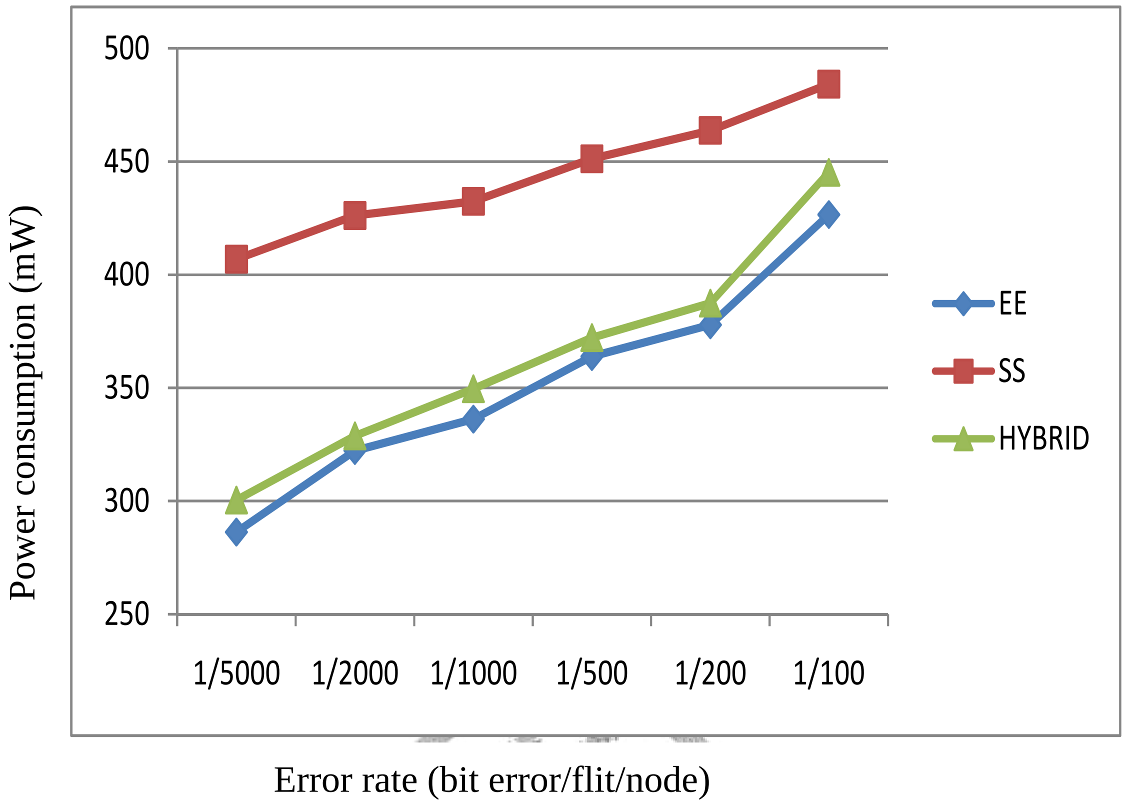
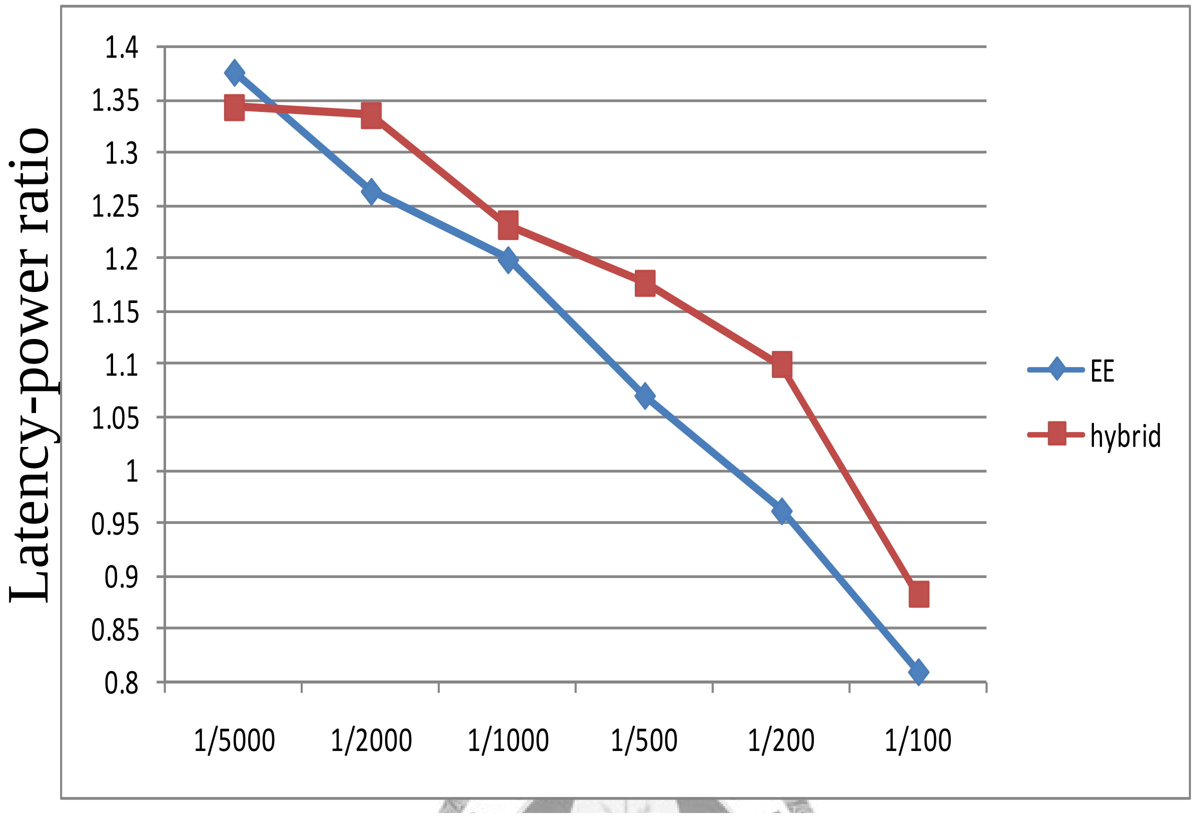


Fig. 4-5 Power consumption comparison

In Fig. 4-5, the result is from a 12 by 12 network on chip system shows that S-S type router consumes more power in all of the conditions with different error rates. The main reason is that S-S router needs additional buffer to maintain the correct copy of data. Even in conventional router, the power consumption of buffer is about 60% to 70% of entire router so S-S router needs additional power to maintain the function of redundant buffer.

While the error rate becomes larger for example at 0.01, our proposed router and E-E router will face a critical problem, network will be congested by the re-transmitted data packet and re-transmitted signal. These two factors will make the network overloaded while the original injection rate is at 5 (flit/cycle/node) so it will increase power consumption quickly.



Error rate (bit error/flit/node)

Fig. 4-6 Comparison of latency-power ratio

In Fig. 4-6, we show the exact system performance that counts the power consumption in. We choose S-S router as the criterion that means we count the ratio of E-E latency to S-S latency and the ratio of our router latency to S-S latency we call these values "latency_ratio" then count the ratio of E-E and our proposed router power consumption to S-S power consumption. We call these values power_ratio then divide latency_ratio by power_ratio. Then we got the performance improvement in power consumption against S-S router.

As we can see, in normal or lower error rate condition our router architecture has better latency-power ratio than S-S router, even better than E-E router. That means, although S-S architecture has better latency performance at any condition it also consumes most power at all the conditions. When we give the same weight to power and latency, S-S architecture is not the one we will choose. When there is no error occurred, our architecture will have better latency-power ratio than S-S router, about 1.4 times better. Then we compare our architecture with E-E architecture. Our architecture has better latency-power ratio when error rate is larger than 0.0002.

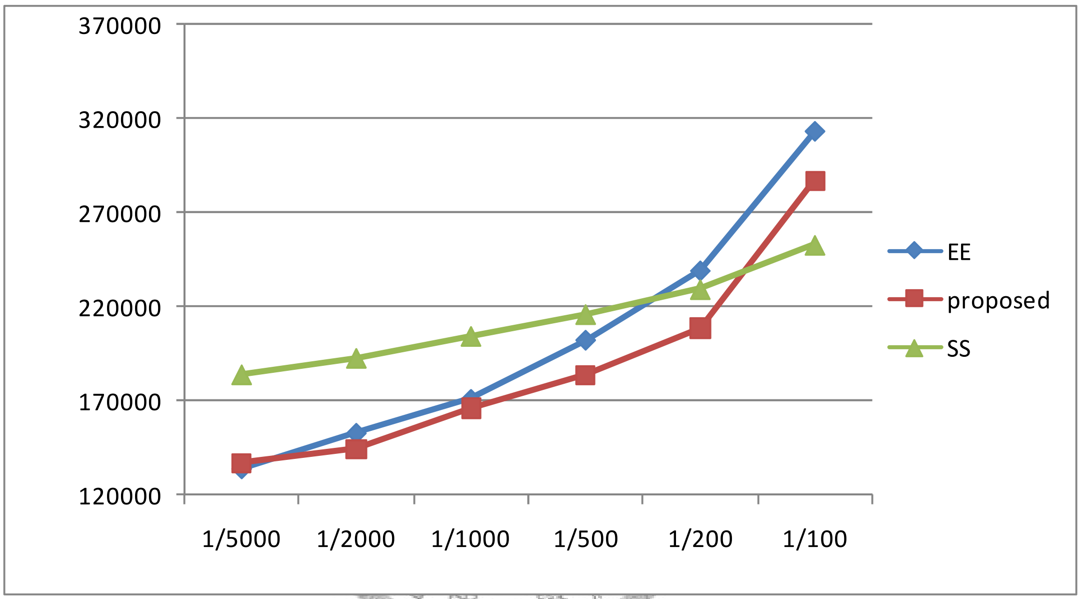


Fig. 4-7 Comparison of product of power and latency

In Fig. 4-7 we compare the product of latency and power. As we can see, the result is similar to the previous comparison. Our proposed architecture is better than S-S router in most low error rate conditions and better than E-E router since error rate is bigger than 1/5000.

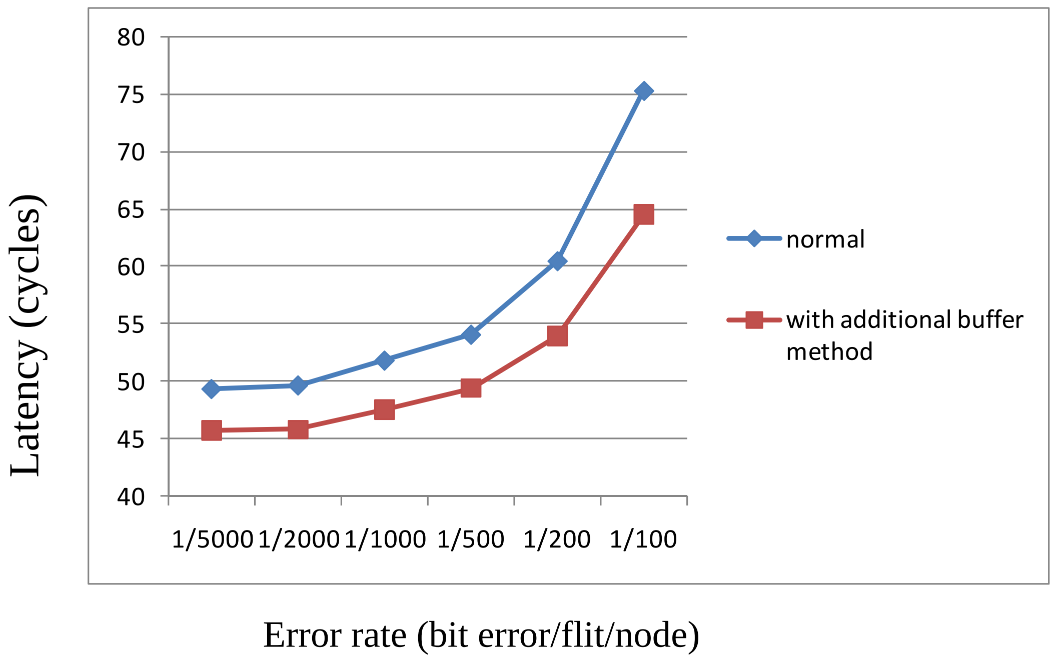


Fig. 4-8 Comparison between using reserved buffer and without using it

Fig. 4-8 shows the latency in clock cycle as the function of error rate when injection rate is 0.05 (flits/cycle/node). As we can see, using the reserved buffer efficiently the latency becomes about 0.15% lower and we could suffer higher error rate by increasing power consumption slightly.

Chapter 5 Conclusion

In this thesis, we have proposed a weighted error detection router architecture for fault tolerant network on chip design. By utilizing encoder and decoder more efficiently the latency-power ratio can be improved. We have also proposed a reserved buffer utilized method.

In the set of experiments we show that comparing with S-S router and E-E router, the performance-power improvement ratio can be increased to 1.4 when comparing with S-S router and 1.18 by comparing with E-E router. More importantly, our additional buffer control method could increase the performance 10% better than without using it. When the network size is larger, our method can get better results.

Reference

[1] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm, "*Computer,* vol. 35, pp. 70-78, 2002.

[2] H. Jingcao and R. Marculescu, "Application-specific buffer space allocation for etworks-on-chip router design," in *Computer Aided Design, 2004. ICCAD-200 IEEE/ACM International Conference on*, 2004, pp. 354-361.

[3] C. Xuning and P. Li-Shiuan, "Leakage power modeling and optimization in nterconnection networks," in *Low Power Electronics and Design, 2003. ISLPED'03. Proceedings of the 2003 International Symposium on*, 2003, pp. 90-95.

[4] T. T. Ye, L. Benini, and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers," in *Design Automation Conference, 2002.Proceedings. 39th*, 2002, pp. 524-529.

[5] S. Kumar, A. Jantsch, J. P. Soininen, M. A. F. M. Forsell, M. A. M. M. Millberg, J. A. O. J. Oberg, K. A. T. K. Tiensyrja, and A. A. H. A. Hemani, "A network on chip architecture and design methodology," in *VLSI, 2002. Proceedings. IEEE Computer Society Annual Symposium on*, 2002, pp. 105-112.

[6] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnectionnetworks," in *Design Automation Conference, 2001. Proceedings*, 2001, pp. 684-689.

[7] P. Guerrier and A. Greiner, "A generic architecture for on-chip packet-switched interconnections," in *Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings*, 2000, pp. 250-256.

[8] F. Karim, A. Nguyen, and S. Dey, "An interconnect architecture for networking

systems on chips," *Micro, IEEE,* vol. 22, pp. 36-45, 2002.

[9] C. J. Glass and L. M. Ni, "The Turn Model for Adaptive Routing," in *Computer Architecture, 1992. Proceedings., The 19th Annual International Symposium on,*1992, pp. 278-287.

[10] C. Ge-Ming, "The odd-even turn model for adaptive routing," *Parallel and Distributed Systems, IEEE Transactions on,* vol. 11, pp. 729-738, 2000.

[11] L. S. Peh and W. J. Dally, "A delay model and speculative architecture for pipelined routers," in *High-Performance Computer Architecture, 2001. HPCA. The Seventh International Symposium on*, 2001, pp. 255-266.

[13] L. S. Peh and W. J. Dally. "A delay model and speculative architecture for pipelined routers", in High-performance Computer Architecture, 2001 HPCA The Seventh International Symposium on, 2001, pp. 255-266

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

[15] R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," Proceedings. Computer Architecture, pp. 188-197, 2004.