Design and Implementation of Low-latency Networks-on-Chip

XIN, Ling

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

in

Electronic Engineering

The Chinese University of Hong Kong

August 2010

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Abstracts of thesis entitled:

Design and Implementation of Low-latency Networks-on-Chip

Submitted by **XIN, Ling**

for the degree of **Doctor of Philosophy**

in **Electronic Engineering**

at **The Chinese University of Hong Kong**

in July 2010

On-chip communication infrastructures are immensely important today. As silicon technology allows more than one billion of transistors in a single piece of silicon, the system-on-chip (**SoC**) circuits can contain already a large number of processing elements (**PE**s). Therefore, the Networks-on-Chip (**NoC**s) are a generally accepted concept to solve the problems such as the scalability and throughput limitation, and physical design problems inherent in dedicated links and shared buses. However, the state-of-the-art on-chip network suffers from latency overhead due to the additional network as compared with dedicated wire connection. According to the different application environments, there are different low-latency technologies for networks-on-chip. This thesis proposes some methods for low-latency NoCs design to relax the latency overhead, which include application-specific asynchronous buffer allocation, hardware multicast support, lookahead bypass scheme and short-circuit crossbar channel optimization.

Asynchronous circuits are usually applied for the communications between multiple clock-domain blocks in some SoCs. According to application-specific traffic, efficiently allocating reasonable buffers in an asynchronous NoC router can avoid the

waste or shortage of buffer resource. The method of application-specific asynchronous First-In-First-Out buffer allocation can reduce the silicon area and the power consumption to improve the network latency. According to given traffic patterns, the save of area buffer of our buffer-allocation method can be up to near 30% and the latency is reduced a little at same time.

Multicast is preferred in parallel computers. It is an inherent fault of network-on-chip as compared with competitor bus architecture. Software method is a conventional method to implement multicast, but there is a large overhead in latency. The latency overhead of a 4-flit multicast packet achieves 6~7 times as compared with tree-based or path-based hardware multicast. Hardware multicast support is necessary in these applications. A group-based hardware multicast method is described and estimated in this thesis. Quality of service is also introduced to speed up multicast packets.

Bypass schemes is efficient to reduce the average propagation cycles in NoCs. We propose novel lookahead bypass scheme to improve the network latency. The lookahead bypass router is implemented and evaluations of various configurations are compared, where the proposed architecture significantly improves the packet latency up to 32.1% over a baseline router. These prove that the router can reduce the average network latency and power consumption, and decreases the reliance on large buffers and virtual channels. Furthermore, the application-specific short-circuit channel is introduced to add some short cuts in a router to bypass the crossbar switch. It can provide additional internal channels to bypass the crossbar and increase the total probability of lookahead bypass. Therefore, the latency can be further reduced. And the throughput can be increased in some applications.

摘要

如今，片上通信架構是極其重要的。由於矽技術發展，使得十多億個晶體管可以集成在一塊矽片。系統級芯片（SoC）的電路已經可以包含大量的處理單元（PEs）的。因此，片上網絡（NoCs）是一種被普遍接受的概念來解決以下問題：可擴展性和吞吐量的限制，專用連線和共同的共享總線內在固有的物理設計問題。但是與專用線路互聯比較，由於實用了額外的網絡,片上網絡路由器必需付出延遲增加的代價。根據不同的應用環境，有不同的片上網絡的低延時技術。本論文為片上網絡設計提出了一些方法低延遲來減少延遲開銷，其中包括專用異步緩存分配，硬件多播(multicast)支持，超前旁路設計方案和短交換通路優化技術。

在一些多時鐘域的系統芯片，異步電路通常應用來通信。根據特定應用的流量，合理有效地分配異步片上網絡路由器的緩存大小，可避免浪費或短缺的緩衝資源。這種專用的異步先入先出緩存分配方法可以減少芯片面積和功耗，降低網絡延遲。根據給定的通信模式，我們的緩存分配方法在保持或略微降低延遲的同時，可節約緩衝內存面積高達近 30%。

並行計算機常常用到多播。而與競爭對手總線架構比較，廣播是片上網絡的一個固有缺陷。傳統上，實用軟件方法來實現多播，但有一個大的延遲開銷。與基於樹或基於路徑的硬件多播比較，4 片段(flit)長的多播數據包的延遲開銷達到 6～7 倍。由此可見，在一些應用中，硬件多播支持是有必要的。在本論文中描述和評估了 一個以通信組為基礎的硬件多播的方法。在路由器中，服務質量(QoS)被引入用來加快多播數據包。

旁路設計方案可以有效降低片上網絡的平均延遲。我們提出新的超前旁路設計方案以改善網絡延遲。我們實現了這個超前旁路路由器，並且比較了不同配置的結果。與基準路由器比較，該架構大大降低了數據包的平均延遲，最高可達 32.1%。這些證明了該路由器可以減少平均網絡延遲和功耗，並減少對虛擬通道大緩存的依賴。此外，我們提出專用的短交換通路優化，通過添加一些捷徑來繞過路由器的交換(crossbar switch)。它可以提供更多的內部通道繞過路由器的交換，并可以增加數據包超前旁路的概率。因此，可以進一步降低延遲。而在一些應用程序中，吞吐量限制也可以大大增加。

Contents

List of Figure

List of Table

Acknowledgement

I would like to express my thanks to all those who have supported me in finishing this thesis. Foremost, I express my deepest appreciation to my advisor, Professor CHOY Chiu-Sing, for his patient guidance throughout the course of my research. I have been benefited from his knowledge, wisdom, stimulating suggestions and encouragement.

Along with Professor Choy, I would like to thank Professor LEUNG Ka-Nang, Professor Gerald E. SOBELMAN, and Professor PUN Kong-Pang for their insightful suggestions and comments on my research and thesis. And I am pleased to express my thanks to laboratory technician, Mr. YEUNG Wing-Yee, who has helped me a lot for CAD tools and details of laboratory. I am also grateful for discussions and help of my research and study to other members in the VLSI and ASIC Laboratory, who are CHAN Chi-Fat, ZHANG Min, XU Ke, ZHENG Yan-Qi, KO Chi Tung, TANG Siu-Kei, SHI Wei-Wei, JIA Jing-Bin and so on. I am also thankful to my friend ZENG Hong, AI Yan-Qing, for his help during chip physical implementations. I am truly indebted to Dr. GRATZ Paul in the TRIPS team of the University of Texas at Austin for providing me the TRIPS OCN traffic traces of Minne-SPEC benchmarks.

Thanks to my family and friends of Shanghai to support me a lot. Thanks to my friends of university who have made my time at CUHK enjoyable. I have had many memorable moments besides my research, such as sports, movie, parties, and friendship.

CHAPTER 1. INTRODUCTION

1.1. System on Chip and Interconnection

Digital systems are pervasive in modern society. Systems on Chip (SoCs) are becoming increasingly complex and heterogeneous [12, 13]. One main characteristic of such a SoC is the seamless integration of numerous Intellectual Property (IP) cores performing different functions.

The chip density is growing very fast, which is the large challenge for each designer. The rise in scale and complexity of System-on-chip (SoC) designs is hampered by the difficulties in on-chip communication arising from poor bus scalability and poor interconnect signal quality. The on-chip interconnects are becoming a speed, power, and reliability bottleneck for more and more complex chips.

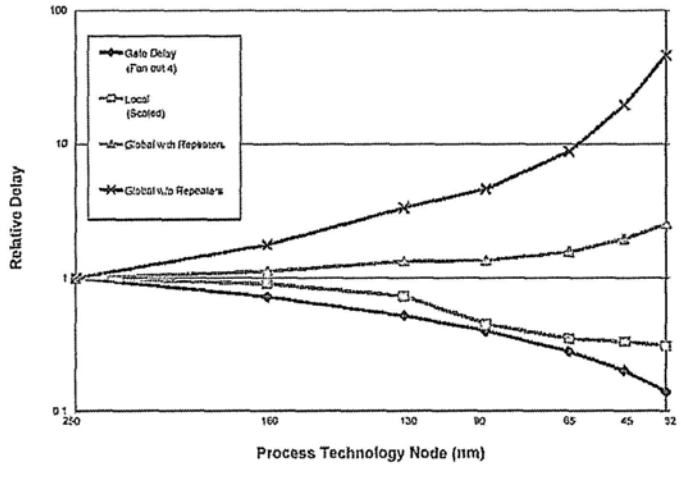


Figure 1.1 Delay for Local and Global Wiring versus Feature Size

The delay for global point-to-point connection is becoming larger, shown in Figure 1.1 [14]. In addition, the point-to-point connection requires a large number of wire resources in a chip, which make digital system's place & route more difficult. The

on-chip communication has become an active research area in the past few years.

Along many SoC interconnect schemes, bus architecture is widely used and well understood. Its limitations are 1) limited scalability, 2) long-wire loads and resistances, 3) Energy inefficiency, 4) Limited throughput. **Especially it is not naturally scalable.** Interconnection networks offer an attractive solution to this communication crisis. Networks on chips (NoCs) have emerged to be the best candidate to connect the many functional elements inside present and future SoCs [1, 10, 15, 16, 19]. They are historically used only in high-end supercomputer and telecom switches. However, they are becoming pervasive from large supercomputers to embedded System-on-chip. It can provide high-bandwidth, low-latency communication and a conveniently organized connection for many processing elements (**PE**s) in different applications.

1.2. Research Goal

Optimizing performance and cutting down cost are two hottest research topics in most designs. Network on chip has large advantage in throughput. However, due to the additional network, the latency overheads of routers are inevitable. Therefore, low latency is a major research issue in the design of NoCs.

Excellent communication protocol, flow control and router structure can all improve network throughput and packet latency. Asynchronous communication protocol has inherent advantage in low latency, low power, etc. [7]. Asynchronous on-chip network becomes a good choice to deliver messages among multi-clock domains. Conventional pipeline routers always cannot make the best use of each pipeline stage because of the unbalance between different stages and different flits. Improvement of pipeline is an interesting research topic. Distributed algorithms, parallelized computation and other

similar works require various types of communication [5, 6], such as one-to-all, all-to-one communication. Improvement of multicast support is eager in such applications.

The original contribution of this thesis includes several implementations of low-latency network-on-chip. Application-specific FIFO allocation algorithm is proposed to design a non-uniform-buffer network. It makes use of the property of asynchronous communication. And proposed hardware multicast support can implement fast broadcast/multicast to reduce average packet latency in some applications that require multicast communication. Moreover, lookahead bypass scheme is proposed to design a small propagation-latency router by performing allocation computation in advance. Further application-specific optimization of short-circuit crossbar channel, which is based on the lookahead bypass router, is implemented.

Other contributions include NoC design flow and NoC library. We make analysis of various traffic patterns. And the evaluation platform is constructed by a customized synthesizable NoC library. Traffic models implement the generation of message whose rule is referred as traffic pattern. And network framework and router models can implement different topologies and flow controls.

1.3. Thesis Outline

In chapter 2, we introduce Network on Chip, where various network topologies and communication protocols are described. Routers are primary components in a network on chip. Many researches focus on reduction of packet latency to achieve the goal of low-latency NoCs. And the design flow, evaluation platform and various traffic models are provided to design and evaluate on-chip networks.

The following chapters introduce several low-latency improvements of NoC router. In chapter3, the application-specific asynchronous FIFO allocation can reasonably allocate FIFOs of different size for input buffer per port according to detailed traffic flows of application. Chapter 4 provides a method of multicast for some special cases that need multicast or broadcast. NoC is not good at sending a message to multiple destinations. The method makes up this fault by introducing multicast scheme and additional control logic.

In chapter 5, lookahead bypass scheme is proposed. The novel bypass scheme needs to advance transmission of control signals for lookahead allocation of flits. It improves pipeline structure to reduce packet latency. Chapter 6 presents the evaluation of various configurations to validate lookahead bypass scheme. Moreover, further improvement of short-circuit path is proposed to improve the network performance according to application-specific traffic distribution. At last, we conclude the contribution and future works in chapter 8.

CHAPTER 2. NETWORK ON CHIP

Network-on-Chip, which has better scalability and compatibility, is regarded as a solution to replace the inferior bus communication. To meet challenges, we can borrow and adapt the concept of packet switched communication from computer networks to design on-chip networks. Packet switched communication, besides providing theoretically unlimited scalability, also provides possibility of standardization and reuse of communication infrastructure [35]. It is because the layered protocol of packet switched communication decouples computation (processing element) from communication (on-chip network). These features are crucial to chip designers to lower time to design and time to market new products.

In this new paradigm, the cores on the chip communicate among themselves by sending packets using a network of routers. The network is built using a set of identical routers connected in regular topology. The important issues for design of the network are router design, design of network access by cores and communication protocols.

2.1 Network Basic

2.1.1 Network topology

On-chip interconnection networks are composed of a set of shared router nodes and physical channels, and the topology of network refers to the static arrangement of these nodes and channels [2].

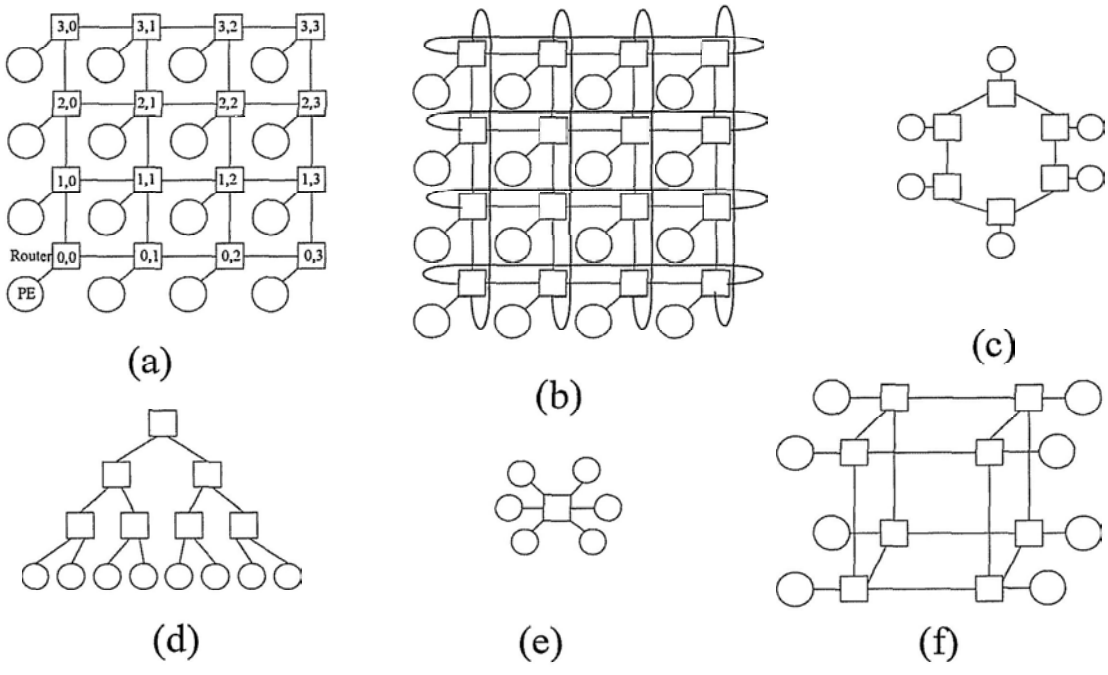


Figure 2.1 Network topology: (a)mesh, (b)tours (c)Ring, (d)Tee, (e)Star, (f)Hypercube

Figure 2.1 presents six types of topologies. For example, the network shown in Figure 2.1 (a) consists of 16 nodes, each of which has eight channels connecting (to or from) neighbors, and two in or out channels connecting local resource. Margin routers can skip some connections with outside. Otherwise, they connect with output pins.

Selecting network topology is the first step in designing a network-on-chip because routing strategy and flow-control method depend heavily on the topology. Then a router can be selected and the traversal of that route scheduled. The topology specifies not only the type of network (such as 2-D mesh), but also the detailed architecture, such as the radix of router, the width and bit-rate of each channel.

Selecting a good topology is based on its cost and performance. The cost is determined by the complexity of processing elements required to realize the network, and the density and length of interconnections. Most designers would try to match the topology of network to the data communication of application at hand. The benefits of customized topology are shown in [10]. Figure 2.2 shows an example of application-specific irregular interconnection, which includes NoC routers and

dedicated wires.

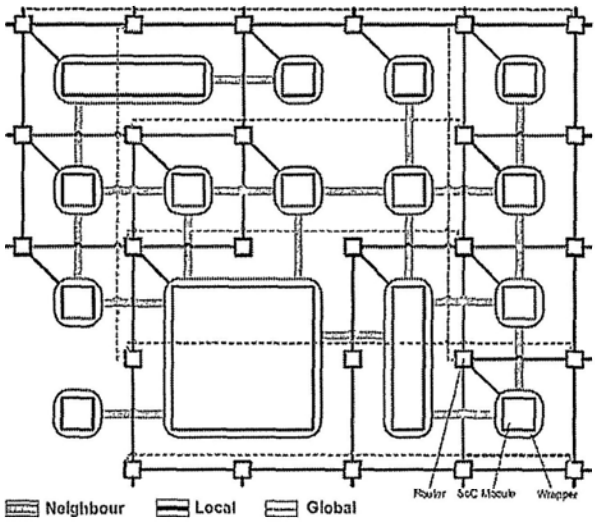


Figure 2.2 An example of application-specific interconnection including NoC router and dedicated wires

However, for a variety of reasons, a special purpose network is not a good idea sometimes. Due to dynamic load in balance in the application, the network load usually poorly balances. Finally, such networks are inflexible. If the user changes to use a different communication pattern, the network cannot be easily changed. A good general purposed network is more widely used than to design a network match to the application. Such 2-D mesh or 2-D tours network is a good generic topology. Whether an application-specific topology is applied is determined by your detailed problem.

2.1.2 NoC Communication Protocol

a) Protocol

A set of rules and methods are required for transfer of information from one resource to another resource in any system, which are generally referred as communication protocols. Whether dedicated wire connection or some other interconnections require the protocols to manager and control transfer of information. Any one unit can communicate with another if it has the interface to implement its protocol.

Hardware handshaking with request and acknowledgement signals is a simple example of control signal protocols for communication between two units connected through direct wires. Bus protocols provide rules for usage of shared communication wires for information exchange and for resolving conflicts among users, which allow reliable communication among many connected units. It specifies the timing relationship among various control and data signals and provides upper limits on the physical length and transfer bandwidth [5].

NoCs usually adopt packet switched networks. In a packet switched network, communication protocols determine how a resource is connected to the network as well as how the information flows from source to destination. The protocols used for packet switched communication are much more complex than bus protocols, and are generally partitioned and organized in many layers. The architecture defining the protocol layers is referred as a protocol stack.

b) Protocol space

7-layered OSI model is a general and famous protocol stack for communication in computer networks [4]. OSI reference model is commonly used as a reference to study NoC layered protocol stacks. There is a little difference between common NoC layered model and OSI model. The mapping of layer for NoCs is shown in Figure 2.3.

For designers of on-chip networks, there is a large protocol space for selection. The available choice provides a trade-off between performance and flexibility. On one extreme is the possibility of using protocols, which support only circuit switched networks for connecting cores. Such protocols can provide high performance but little

flexibility. On the other hand, very general protocols used for computer communication can provide a lot of flexibility but very low performance.

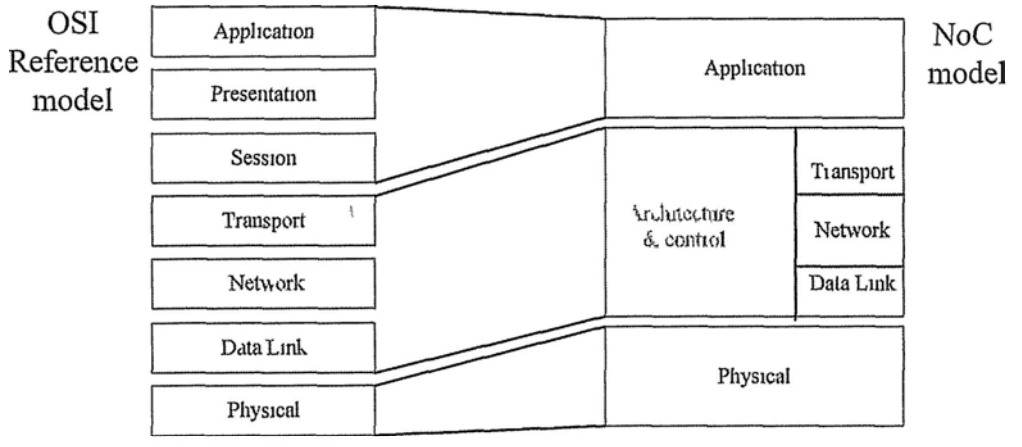
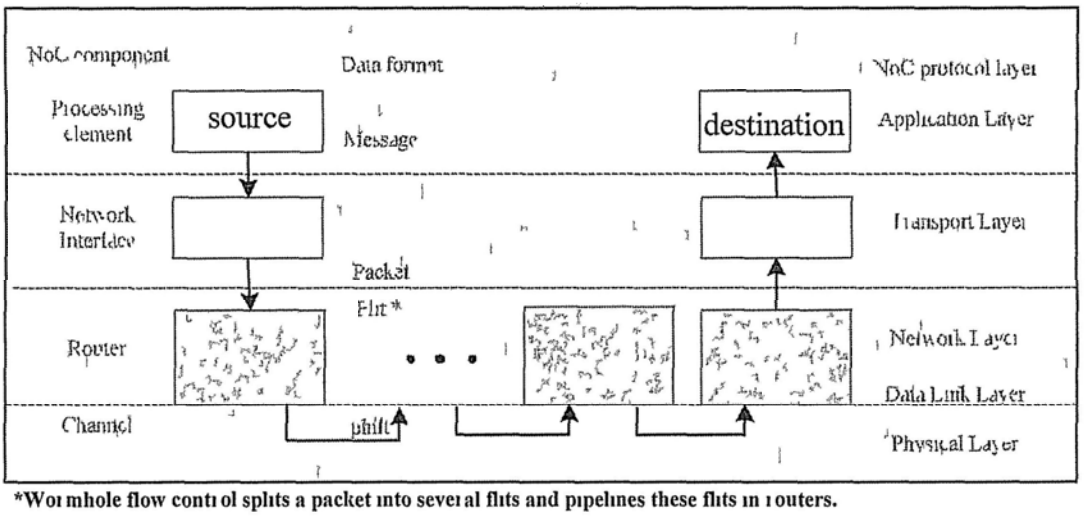


Figure 2.3 NOC layer mapping based on 7-layered OSI reference model

Because of the small physical dimensions of on-chip networks, it will be possible to assume smaller error rates and higher communication link bandwidths. Network-on-chip layered protocol and components used in each layer are shown in Figure 2.4, where the arrow lines present a message flow example from its source PE to its destination PE. Moreover, Figure 2.5 shows an example of data format of each layer.



*Wormhole flow control splits a packet into several flits and pipelines these flits in routers.

Figure 2.4 An example of network-on-chip layered protocol and components

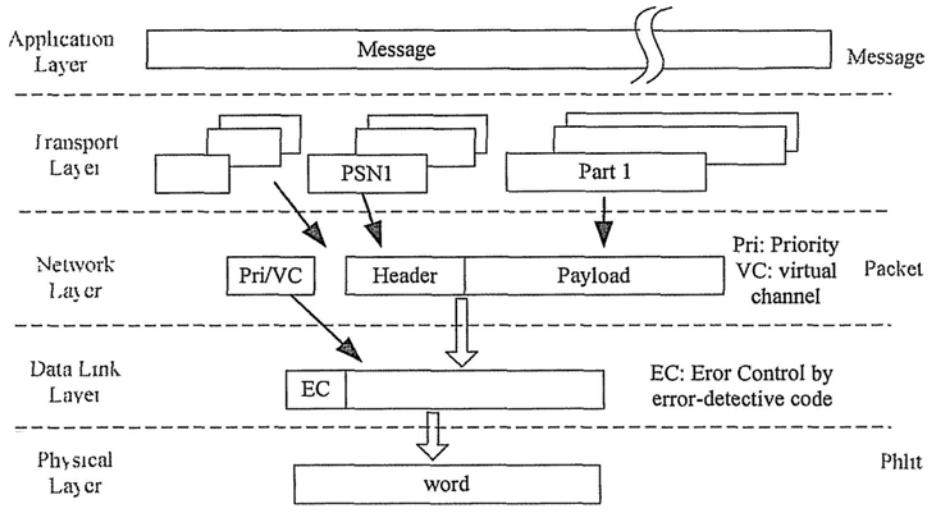


Figure 2.5 Data formats of various protocol layers in on-chip networks

- **Physical Layer:** This layer is concerned with physical characteristics of the physical channel used for connecting routers and resources with each other. It specifies voltage levels, length and width of wires, signal timings, number of wires connecting two units etc. The data link layer takes care of hardware synchronization also. It is noted that phit size and flit size are assumed equivalent, which equal to the width of **W**-bits physical data channel in our designs.
- **Data Link Layer**: This layer has the responsibility of reliability of transmission. Data link layer may also include data encoding or data rate management etc. This layer can encode various formats, such as CRC or comparison code, to error detect error or increase data rate. Of course, on-chip communication can skip error control because of the reliability of on-chip transmission in many applications.
- **Network Layer**: This layer is charge of delivering a packet from one element to another using the network of routers. Its responsibility includes taking routing decisions and allocating packets. In a virtual-channel wormhole network, a packet is broken up into flits across some virtual channel. This layer needs to provide the

additional service of flit management and virtual channel management.

- **Transport Layer**: This layer has the responsibility of establishing end-to-end connection and delivery of messages using the lower layers. Therefore its functionality includes packetization and of a message and conversion from a packet to flits at the source, de-packetization of packets into a message and assembly flits into a packet at the destination node and
- **Application Layer**: For on-chip communication, this layer is referred as upper three layers of OSI model. This layer controls the message injection and utilizes the received message. Its functionality includes message synchronization and management, conversion of data formats etc. Processing elements are used to implement this layer.

c) Flow control

Flow control is a synchronization protocol for transmitting and receiving a unit of information. The unit of flow control refers to that portion of the message whose transfer must be synchronized. This unit is defined as the smallest unit of information whose transfer is requested by the sender and acknowledged by the receiver.

For example, it is easy to think of messages in terms of fixed-length packets, shown in Figure 2.6. A packet is forwarded across a physical channel or from input ports of a router to output ports. In this example, the flow of information is managed and controlled at the level of an entire packet. The request/acknowledgment signaling is used to ensure successful transfer and the availability of buffer space at the receiver. Note that there is no restriction on when requests or acknowledgments are actually sent or received. Implementation efficiency governs the actual exchange of these control

signals (e.g., the use of block acknowledgments).

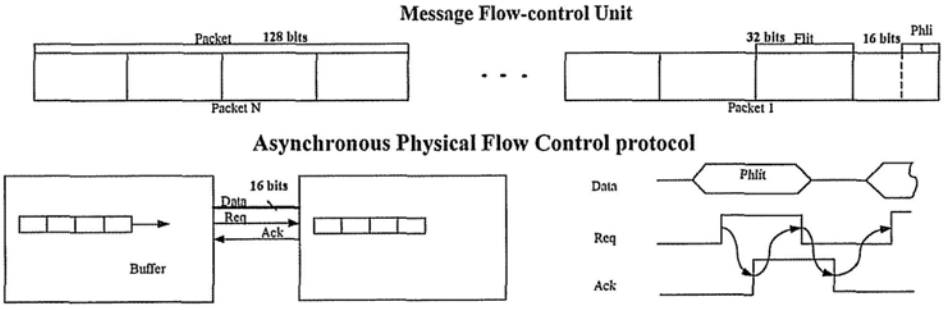


Figure 2.6 An example of asynchronous request/acknowledgment flow control

Flow control occurs at two levels. In the preceding example, message flow control occurs at the level of a packet. However, the transfer of a packet across a physical channel between two routers may take several steps or cycles, for example, the transfer of a 128-byte packet across a 16-bit data channel. The resulting multi-cycle transfers use physical channel flow control to forward a message flow-control unit across the physical link connecting routers.

Flow control also determines how resources of a network, like buffer space and channel bandwidth, are allocated to messages traversing the network. There are generally two categories of flow control strategies: circuit switching and packet switching.

In circuit switching, the complete message is transmitted after the circuit has been set up. At first, a physical path from the source to the destination is reserved prior to the transmission of the data. It could be realized by injecting the packet's routing header flit as the routing probe into the network at first. When the probe reaches the destination, a reserved path of routers has been set up and an acknowledgment would be transmitted back to the source. Circuit switching is generally advantageous when messages are infrequent and long, which is that the message transmission time is long as compared with the path setup time. Another advantage is that the transmission time is predicable after the path established. Designers can more precisely schedule their programs. The

disadvantage is that the physical path is reserved for the duration of the message and may block other messages.

The message also can be partitioned and transmitted as fixed-length packets. Each packet is individually routed from source to destination, such as the example in Figure 2.5. This technique is referred to as packet switching. The type of flow control includes three popular techniques: store-and forward, virtual cut-through, and wormhole [2].

- **Store-and forward (SAF) switching**: A packet is completely buffered at each intermediate node before it is forwarded to the next node. The header information is extracted by the intermediate router and used to determine the output link over which the packet is to be forwarded.
- **Virtual cut-through (VCT) switching**: The router can start forwarding the header and following data bytes as soon as routing decisions have been made and the output buffer is free. In fact, the message does not even have to be buffered at the output and can cut through to the input of the next router before the complete packet has been received at the current router. The message is effectively pipelined through successive routers.
- **Wormhole switching**: The need to buffer complete packets within a router can make it difficult to construct small, compact, and fast routers. In wormhole switching, message packets are also pipelined through the network. However, the buffer requirements within the routers are substantially reduced over the requirements for VCT switching. A message packet is broken up into flits. The flit is the unit of message flow control, and input and output buffers at a router are typically large enough to store a few flits.

The preceding switching techniques were described assuming that messages or parts

of messages were buffered at the input or output of each physical channel. Buffers are commonly operated as FIFO queues. Therefore, once a message occupies a buffer for a channel, no other message can access the physical channel, even if the message is blocked.

Furthermore, the **virtual channel flow control** associates several virtual channels with a single physical channel. It overcomes the blocking problems of wormhole flow control by allowing other packets to use the physical channel bandwidth that would otherwise be left idle when a packet blocks [20]. Therefore, when used in this manner these buffers are referred to as virtual lanes. Virtual channels can also be used to improve message latency and network throughput. By allowing messages to share a physical channel, messages can make progress rather than remain blocked. [22]

d) Buffer management flow control

Adding buffer to network-on-chip can result in significantly more efficient flow control. It is because buffers can decouple the allocation of adjacent routers. All of the flow control methods that use buffering need a means to communicate the availability of buffers at the downstream nodes. Then the upstream nodes can determine when a buffer is available to hold the next flit to be transmitted. Researchers have also proposed buffer-less switch for packet switched communication [23].

Buffering policies are extremely important to the design of the message layer. They are crucial to both correctness as well as performance. Three types of low-level flow control mechanisms are in common use today: credit-based, on/off, and ack/nack [5].

- **Credit-based flow control**: With the credit-based flow control, the upstream node keeps a count of the number of free flit buffer in each virtual channel downstream. If the count reaches zero, all of the downstream buffers are full and no further flits can be forwarded. Once the downstream router forwards a flit and frees the

associated buffer, it sends a credit to the upstream router. We assume the credit round-tip delay t_{crt}, which includes a round-trip wire delay and additional processing time at both ends. This corresponds to a bit rate of L_f / t_{crt}, where L_f is the length of a flit in bits. If there are **F**-flit buffers on this virtual channel, F flits could be sent before waiting for the credit, giving a throughput of F flits per t_{rt}. Thus, we see that to prevent low-level flow control from limiting throughput over a channel with bandwidth b we require:

$$F \geq t_{rt} b / L_f \qquad <2.1>$$

- **On/off flow control:** This flow control can greatly reduce the amount of upstream signaling in certain cases. The upstream state is a single control bit that represents whether the upstream node is permitted to send (**on**) or not (**off**). On/off flow control requires that the number of buffers be at least $\mathbf{t_{rt}b/L_f}$ to work at all. Twice this number of buffers is required to operate at full speed.
- **Ack/nack flow control:** This flow control reduces the minimum of this buffer vacancy time to zero and the average vacancy time to $\mathbf{t_{rt}/2}$. The upstream node sends flit whenever they become available. If downstream router has a buffer available, it accepts the flit and sends an acknowledgement. If no buffers are available, the downstream node drops the flit and sends a negative acknowledgement. Thus, buffers must be held for an additional $\mathbf{t_{rt}}$ waiting for an acknowledgment, making ack/nack flow control less efficient in its use of buffers than credit-based flow control.

2.1.3 Routing

There is usually more than one path of routers from the source to the destination in a on-chip network. One way of configuring path selection of routers in a particular

topology is referred as **routing**. A good routing algorithm can keep path lengths as short as possible, and balance load across the network channels to make the saturation throughput closer to the ideal. There are two main classes of routing algorithm: **static (deterministic) routing** and **dynamic (adaptive) routing** [9].

- **Static (deterministic) routing**:

The deterministic algorithms are simple. The static routing algorithms are used widely in small networks. The size of on-chip network cannot be too large because the limit of chip scale. In static routing systems, the routes are entered into the router at first, which adopts pre-computed routing tables sometimes. Routes through a data network are statically described by fixed paths of routers. The advantage of static routing is low delay and simple logic in routers. Moreover, static routing algorithms can implement deadlock-free network by the good pre-computed routing tables.

Despite its generally poor load balancing properties, **dimension-order routing** is widely used in mesh or and tours networks. A packet is routed one dimension at a time. Within each dimension, the packet travels until it reaches the same coordinate as the destination in that dimension. This routing approach is very simple to implement, which fits on-chip networks. And, it simplifies the problem of deadlock avoidance, which prevents any cycles of transmission between dimensions. However, deadlock can still occur within a dimension. In addition, dimension-order routing has no tolerance of faults. In our designs, X-Y routing is applied in 2-D mesh networks. It selects the OP depending on the location of the current node and destination. The packet will go through the X dimension at first until it arrives at the Y dimension of the destination.

- **Dynamic (adaptive) routing:**

Due to the dynamic load imbalance, the load balancing properties usually is not good

in static routing. The adaptive routing can construct routing tables automatically and allow the network to act nearly autonomously in avoiding network failures and blockages. It dominates the Internet. Moreover, the dynamic routing has better tolerance of the presence of faults in the network. If a link or a port fails, the entire system fails. However, if an algorithm can adapt to the failure, the system can continues to operate with a little loss in performance.

Several partially adaptive routing algorithms have been proposed. Partially adaptive routing algorithms represent a trade-off between flexibility and cost. They try to approach the flexibility of fully adaptive routing at the expense of a moderate increase in complexity with respect to deterministic routing. Most partially adaptive algorithms proposed up to now rely upon the absence of cyclic dependencies between channels to avoid deadlock. Some proposals aim at maximizing adaptivity without increasing the resources required to avoid deadlocks. Other proposals try to minimize the resources needed to achieve a given level of adaptivity.

2.2 NoC Router

The router is an important component of interconnection architectures for routing and delivering message from a source to a destination in networks. Network-on-chip routers can mostly determine the network performance and cost the most part of power consumption and silicon area of whole network.

2.2.1 Router architecture

The router has a functionality of transferring message from one of its input ports to any one/more of its output ports. The basic architecture of a network-on-chip router includes

a routing computation cell, a switch allocator, and a crossbar switch [34]. The three components can complete the basic task in a packet-switched network: to deliver the message from an input port to anther correct output port in a router.

Another common component buffer is used to cache the waiting network message in a network-on-chip router. It is possible that one network message need to wait the permission from the switch allocator if the allocation computation costs long time or the network message conflicts with another network message.

Moreover, the virtual channel technology is usually applied in most state-of-art routers. Network-on-chip router involves a virtual channel allocator to allocate a free virtual channel for the current packet. Figure 2.7 shows the basic architecture of a virtual-channel router.

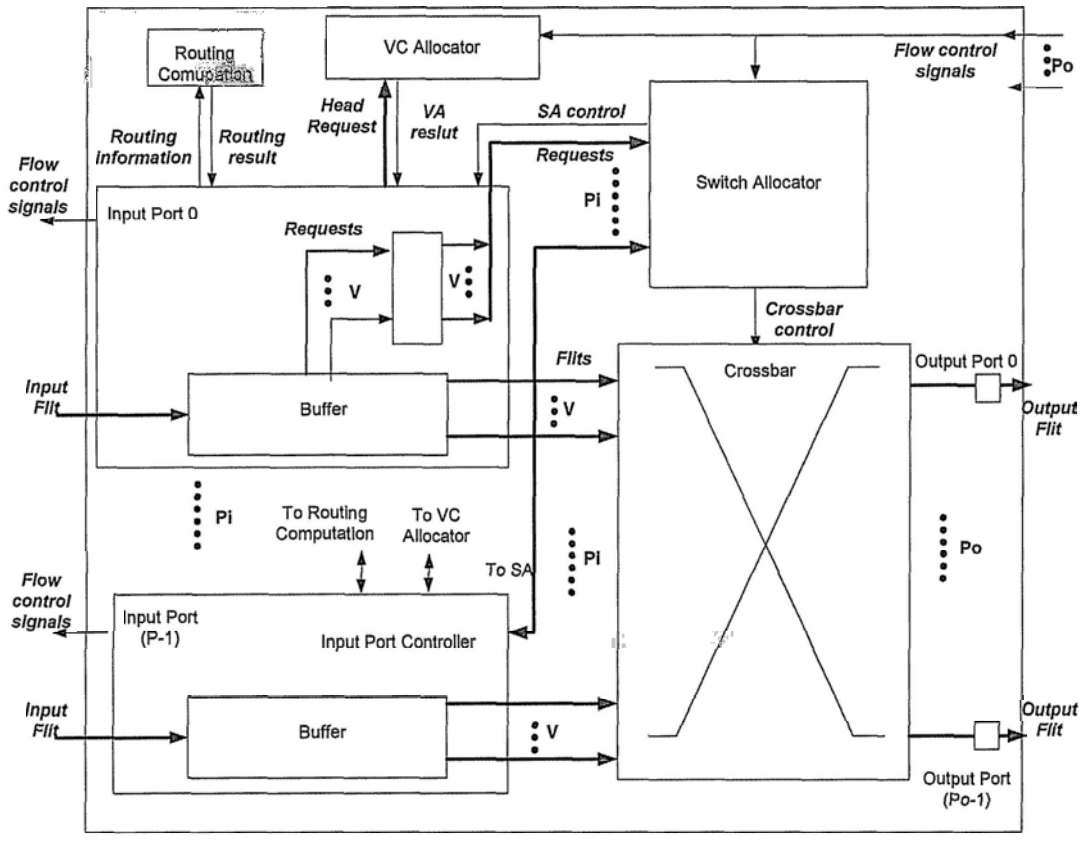


Figure 2.7. The architecture of virtual-channel router

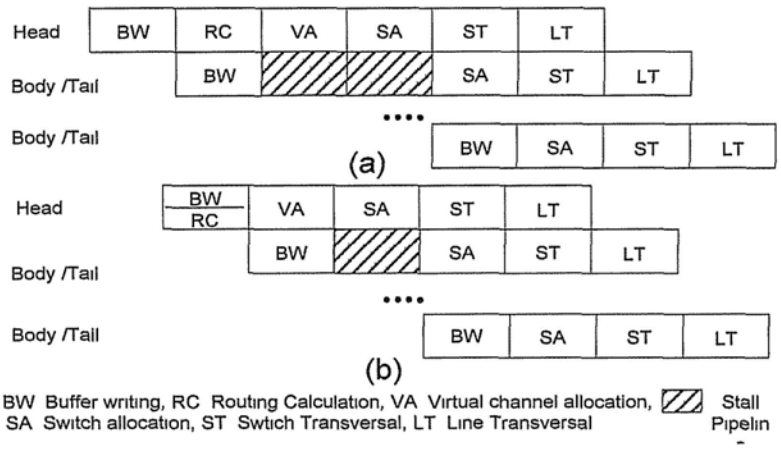


Figure 2.8 Examples of pipeline stage structures of virtual-channel routers

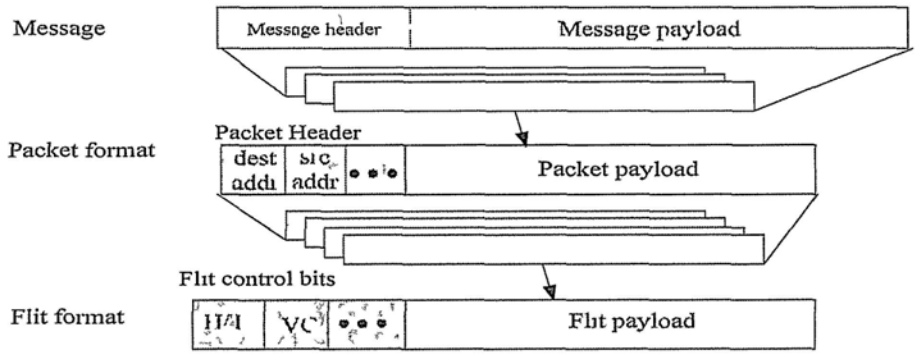


Figure 2.9 An example of packet format and flit format

Two examples of pipeline stage structures are shown in Figure 2.8. A head flit requires six stages because of additional virtual channel allocation and routing computation, such as (a). If static routing algorithm is adopted, the routing computation can be integrated into buffer write stage, such as (b). Figure 2.9 presents an example of packet format and flit format in the virtual-channel router.

2.2.2 The router components

A) Routing computation cell (RC)

It has responsibility of routing for each packet based on the routing information that is carried by network packets. In virtual-channel routers, only head flits require routing computation cell and the routing results involve storing in buffers for remaining flits of this packet. The format of routing results is one or more requests to correct output ports for some one input port according to adopted routing algorithm. Then, the input port

can submit its requests to virtual channel allocator and switch allocator.

If the on-chip network adopts static routing, the routing computation cell has a simple architecture and low operation latency. Although adaptive routing do avoid network failures and blockages better, the complex architecture is not suitable to on-chip interconnection networks.

B) Crossbar

The crossbar passes flits from source input ports to appropriate output ports. A router, which has **Pi** input channels and **Po** output channels, only need a $Pi \times Po$ crossbar to deliver flits. The number of input and output ports of a router is generally small in on-chip network. However, virtual-channel flow control largens the router's crossbar to $V \bullet Pi \times Po$. To **sim**plify the crossbar, **V**:1 multiplexer is introduced to switch the messages of **V** virtual channels at first. Then the output of the multiplexer connects one input port of a $Pi \times Po$ crossbar. For example, a mesh-network router has five input ports and five output ports. And almost NoC routers present the tour transmission. It is noted that the crossbar can ignore the internal path to the output with same orientation of input because U-turn transmission is prohibit in our network.

C) Switch allocator (SA)

Multiple packets may request the same neighbor router at same time in a router. It is necessary that a component be introduced to allocate the output physical resource to neighbor routers for all requests. Switch allocator is in charge of the allocation computation. The arbiters in the switch allocator have responsibility of providing a fair arbitration for these requests. However, the exact meaning fair can vary from application to application. Three usual types of fairness are weak fairness, strong fairness, and FIFO fairness [2]. Matrix arbiter and multi-way MUTEX arbiter, which both can provide fair arbitration, are used respectively in synchronous and

asynchronous routers in the thesis.

For a **Pi**-input-port **V**-virtual-channel router, there theoretically are Pi **"V*Pi**:1" arbiters in the switch allocator. The switch allocator is very complex and large. Fast arbitration policies are crucial to maintaining low flow-control latency through the switch. The implementation of switch allocator can be simplified by two-stage arbiters. In fact, tour transmission is prevented in our routers. Then, there are **P "V**:1" arbiters and **P "P-1**:1" arbiter. [31]

D) Buffer

Adding buffers can improve the performance of network [25, 32]. There are three types of buffers in a network-on-chip router: input buffer, output buffer and crossbar buffer. Figure 2.10 shows an example of these buffers.

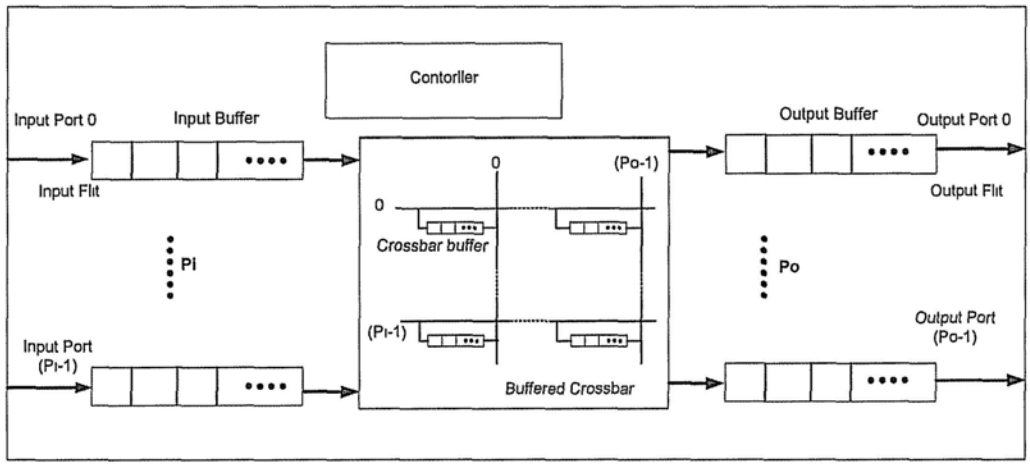


Figure 2.10 An example of input buffer, output buffer and crossbar buffer

A packet-switch router usually adding input queue or output queue to decouple the allocation of adjacent routers. Both input and output queue require relevant buffers to cache queue packets/flits. Moreover, the addition of buffers at the crosspoints to reduce head-of-line (HOL) blocking and hence increase throughput is referred to as buffered crossbar switch architecture. This scheme of placing buffers at the crosspoints instead of pure input buffering reduces the HOL blocking problem caused

by output contention. With buffer memory placed at each crosspoint, the scheduler can maximize total throughput and keep many output ports as busy as possible by storing the input packets in buffer memory [24]. Of course, the improvement requires more silicon area of buffer.

E) Virtual channel allocator (**VA**)

This component is responsible for allocating the virtual channels of downstream routers. A head flit usually needs to cost additional one cycle to be allocated a virtual channel in a conventional router. The use of virtual channels, while reducing header-blocking delays in the network, makes the controllers more complex by requiring more arbitration in SA and more complex flow control mechanisms in VA. Increasing the number of virtual channels also has a direct impact on router performance through their effect on the achievable hardware cycle time of the router. The controllers now become more complex since they must support arbitration between multiple virtual channels, and this arbitration function can be on the critical path for delay. This increase affects all messages through the routers. Speculative-pipeline router can parallelize switch allocation with virtual-channel allocation, which is discussed in section 2.3.

2.3 Low-latency Optimization of Routers

The function of network-on-chip is to deliver the message for units connecting to the network. The average latency of delivering message from sources to destinations is an import performance of network. As compared to dedicated connections, packets of NoCs need to take up additional latency and energy at each router (writing and reading

buffer, route and allocation computation, switch traversal). The latency is modeled as:

$$T_{latency} = T_{injection} + T_{network} + T_{drain} = T_{injection} + T_{drain} + \sum_{i}^{path} \left(T_{router}^{i} + T_{wire}^{i} \right) \quad <2.2>$$

The number of router, which passes a given flit, is referred as traversal distance **D** of this flit. It is noted that there are **D-1** long interconnection wires between these routers. Flit is the transmission unit. Assume that the latency of all router and interconnection wires is uniform for the same flit, which are $\mathbf{T_{router}}$ and $\mathbf{T_{wire}}$ respectively. The minimum latency of network part of a flit is:

$$T_{flit} = D \times T_{router} + (D-1) \times T_{wire} \quad <2.3>$$

Thus, the minimum latency of network part for a packet is:

$$T_{packet}^{network} = D \times T_{\text{header}} + (D-1) \times T_{wire} + \max(P_r, P_w)(L/W) \quad <2.4>$$

In Equation 2.4, **L/W** is the number of flit-packet size, $\mathbf{P_r}$ and $\mathbf{P_w}$ are the cycle time of router and wire, $\mathbf{T_{header}}$ is the latency of a router for head flit. The traversal distance **D** is fixed after the network topology is determined. $\mathbf{T_{wire}}$ is concerned with the physical specifications such as the length, width, voltage, etc. Reducing the header latency of a router or shortening the cycle time is possible choices to shorten the minimum latency of a packet.

Considering the waiting time $\mathbf{T_{confilict}}$ derived from conflict between different packets, the actual latency of a packet from source processing element to destination is:

$$T_{Latency} = \sum_{i \in path} \left(T_{conflict}^{i} \right) + T_{packet}^{network} + T_{inj} + T_{drain} \quad <2.5>$$

$T_{conflict}$ is determined by the flow control and the tangible traffic content. It becomes the most of actual latency when the network load is high. Of course, the minimum latency of a packet determines the average packet latency when network load is low.

Network-on-chip routers bring a large part of power consumption, network latency, and area cost. Flow control is implemented by routers too. Therefore optimizing

designs of routers is a very important method to optimize an on-chip network. The following introduces several classes of optimizations.

- Multiple-stage pipeline

Multiple-stage pipeline structure breaks up a whole switch operation into several pipeline stages to increase the frequency. Thus, multi-stage pipeline routers have a great advantage in throughput over single-stage routers. It is noted that there is no advantage in minimum flit latency. More pipeline stages mean more waste in latency because of the unbalanced pipeline stages and additional flip-flops. However, the shortening clock period can reduce the interval time between flits. Moreover, the multi-stage pipeline shortens the waiting time of conflict, which will happen only in SA or VA stage in multi-stage routers. Thus, a multiple-stage router can improve both network throughput and packet latency. Of course, a bad-design pipeline will introduce a mass of additional latency and energy waste.

- Parallelization of functions

Many researchers have proposed router architectures that reduce the latency along the critical path by parallelizing some functions, thereby achieving high-throughput low-latency on-chip networks [1, 11].

A) Routing computation module

In wormhole routers, the routing computation module is only used by head flits. If the routing cell can be removed from critical path by paralleling with other function, the minimum packet latency can be reduced.

The implementation of routing computation is simple if static routing is adopted. The delay of routing is short enough to integrate the routing into other pipeline stages, such as buffer writing stage in Figure 2.8 (b).

B) Virtual channel allocator

Virtual-channel allocation can be parallelized with switch allocation in one cycle [31, 60]. Thus, SA's arbiters and VA's arbiters are combined, shown in Figure 2.11. In other words, a packet can obtain virtual channel of downstream router only after it wins the arbitration of switch allocation. The function of virtual channel allocation is also simplified to the management of free virtual channels. The router architecture is speculative because there are the possible mismatches of virtual channel allocation, which is referred as speculative-pipeline router. The additional cycle of VA stage is saved in header latency.

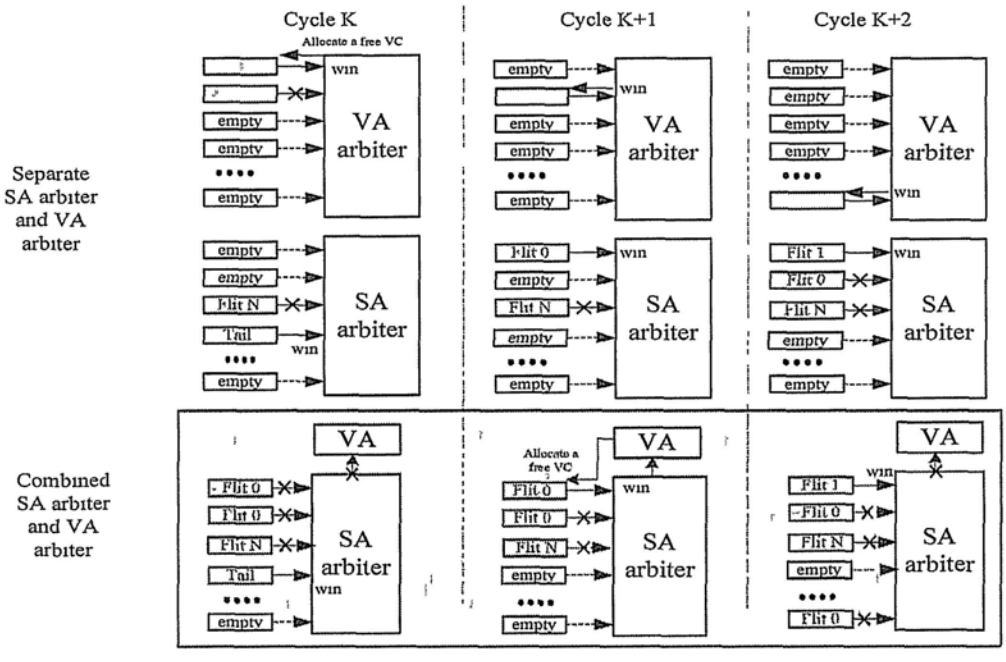


Figure 2.11 The comparison between separate and combined SA and VA arbiter

➢ Quality of service

Some applications, especially real-time applications, expect predictable quality of service (QoS) in embedded systems [17, 18].

A QoS-based design style alleviates the demanding task for the system designer. Systems, especially SOCs, have limited resources, and they must be shared and managed as the application unfolds its dynamic behavior. Resource management depends on two phases: negotiation to obtain resources, followed by a steady state in which allocated resources are used. As applications become more dynamic, the first

phase, renegotiation, will become more frequent. Thus, users can be given a predictable QoS by giving resource management and QoS a prominent place in system design. **In essence, offering a QoS requires a commitment.**

Although guaranteed services, which offer commitment, have many advantages over so-called best-effort services, which offer no commitment, we show that their combination is beneficial. A service is guaranteed if a commitment is given, and best effort otherwise. This holds for individual services, not their ensemble. For example, data transport may be uncorrupted (commitment to correctness), and lossless (commitment to delivery), and without throughput guarantees (best-effort throughput, i.e. no commitment to a completion bound). Moreover, a given service can be offered both with and without commitment to flexibly use the available resources. A combination of best-effort and guaranteed services gives the advantages of guaranteed services to only part of the system, but the available resources are used more efficiently.

- Multicast/Broadcast

In parallel computers, there are many studies about collective communication, including multiple one-to-one, one-to-all, all-to-one, and all-to-all communication [5]. In some applications, many communications are required to transmit from one source to multiply destinations. For example, a chip of distributed processing system needs to synchronize their caches for the coherence. Routing schemes differ in their delivery semantics:

- ✓ Unicast delivers a message to a single specified node;
- ✓ Broadcast delivers a message to all nodes in the network;
- ✓ Multicast delivers a message to a group of nodes that have expressed interest in receiving the message;
- ✓ Anycast delivers a message to any one out of a group of nodes, typically the

one nearest to the source.

Traditional bus architecture is not naturally scalable for higher bandwidth and more diverse clock frequencies. However, a bus is very efficient in broadcast communication since all elements are directly connected to the bus. NoC architecture offers more throughput and better signal quality, but it does not support broadcast very well. Most NoC routers only can apply a software operation to implement the multicast, whose efficiency is very poor. It is implemented by sending a copy of the message from the source elements to every destination or to a subset of destinations. Hardware multicast of NoC is necessary in these applications.

- Asynchronous communication

Most digital circuits designed and fabricated today are synchronous. All components share a common and discrete notion of time, as defined by a clock signal distributed through the circuit. In asynchronous circuits, there is no common and discrete time. Instead, the circuits use handshaking between their components in order to perform the synchronization, communication, and sequencing of operations.

The advantages of asynchronous circuits are: low power consumption, high operation speed, less emission of electro-magnetic noise, robustness towards variations in supply voltage, temperature, and fabrication process parameters, better compensability and modularity, and no clock distribution and clock skew problems [7]. In addition, many state-of-art SoCs adopt multiple clock domains. The global communication is difficult to synchronize different clock domains in the SoCs. Asynchronous protocol is suitable to multi-clock-domain system.

But asynchronous design is not yet a well-established and widely-used design methodology. Asynchronous communication details are discussed in APPENDIX A.

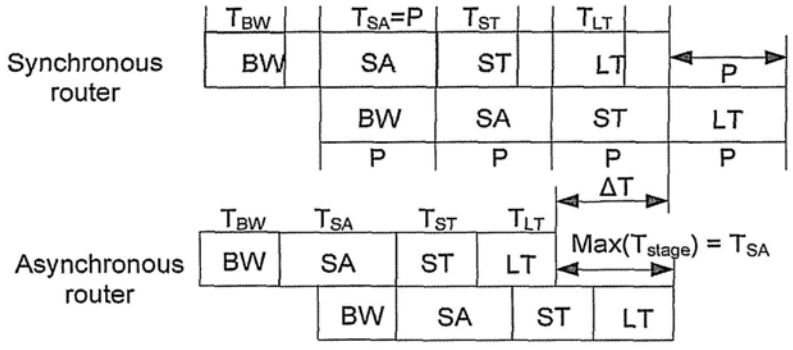


Figure 2.12 Propagation latency of asynchrony router and synchronous router

In a multi-stage pipeline router, the operation time of each stage is hardly balanced. In Figure 2.12, the gray part presents the practical operation time of each stage. The synchronous router's propagation latency is the times of period that is determined by the maximum operation time of four stages. However, the minimum propagation latency of asynchronous router is the sum of all stages' operation times. The improvement is obvious that asynchronous circuit is adopted in NoC.

Thus, asynchronous on-chip networks can provide low-power and low-latency communication as compared to synchronous networks.

➢ Bypass technology

Bypass technology is used widely in on-chip networks. In many cases, routers can bypass several components of the routers based on the detail of usage of router and current flit. The color lines are various bypass schemes in Figure 2.13.

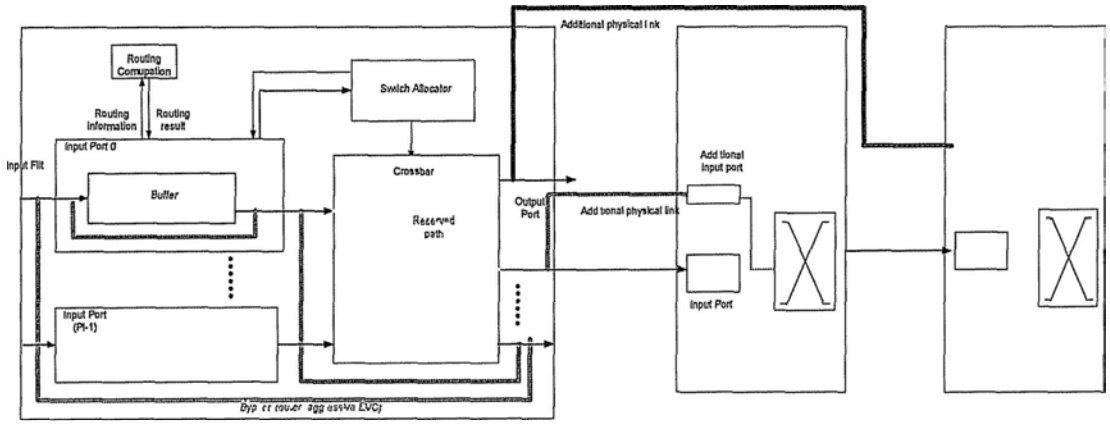


Figure 2.13 Various bypass schemes

A) Bypass buffer

Bypassing buffer operation is easy to be adopted usually. If an input FIFO is empty, an incoming flit will avoid writing the main memory of FIFO and bypass to the output registers of FIFO directly, such as the pink line. It can reduce the power consumption due to reduce the operation of writing and reading a FIFO's main memory. But it adds a little overhead in judgment and management.

Further, if a router can provides lookahead signals such as routing result for the neighboring routers. Some operations such as switch allocation and virtual channel allocation can be executed in advance in its neighboring routers. It is discussed in CHAPTER 5.

B) Bypass crossbar

Crossbar provides multiple paths from different input ports to different output ports. Sometimes there are a large part of packets pass from a given input port to a given output port. If we can set up a special connection between them, such as the orange line, these packets will go through the special path and bypass crossbar. In CHAPTER 7, we propose a design to bypass crossbar by short-circuit crossbar channels.

C) Bypass switch allocation

Circuit switching set up a path from source to destination in advance by route probe, such as the yellow line. The method of circuit switching can be combined with packet switching. A router can set up a reserved path and occupy the output port during transmitting a packet. Thus, a flit can bypass switch allocation if there has been a reserved data path for this packet or for other packets of a special packet group.

D) Express virtual channel

The bypass path can be integrated in the current router similar to an additional express virtual channel (EVC) [30]. There are different modes of EVC. The red line of Figure

2.13 is aggressive mode of EVC. The entire router can be bypassed if the pervious router has a special bypass path to the downstream router. The EVC optimization needs to add virtual channel overhead for the express virtual channels. However, EVC does not add physical channel bandwidth and cannot resolve the jam of channel.

E) Extra physical links

It also can be introduced by the additional physical links between the upstream and downstream routers, which is the blue lines in Figure 2.13. The extra physical links can provide larger channel bandwidth and extra connect of customized topology. Better performance is proved by extra physical links. However, it requires the overhead of physical wires in chip and more complex router because of increasing radix

2.4 Design Flow and evaluation platform

2.4.1 Design Flow

The design flow of on-chip network is shown in Figure 2.14. The design flow is based on a customized NoCLib including reconfigurable evaluation platform and various synthesizable network models, whose detail is discussed in APPENDIX B. Initially based on the given application specification, the spatial bandwidth distribution and temporal distribution of traffic are defined. According to the requirement of communication, the network configuration (such as topology) and router configuration (such as flow control) is determined.

The chosen network can then be optimized with more accurate traffic patters defined by a behavior/TLM model or traffic emulator. One can go on to the detail of router design and application model. With knowledge gained from preceding step, one can avoid the design iteration due to the compatibility problem between model and network, or software and hardware. If necessary, co-simulating the framework with embedded the

RTL model, gate-level model or post-layout model is possible.

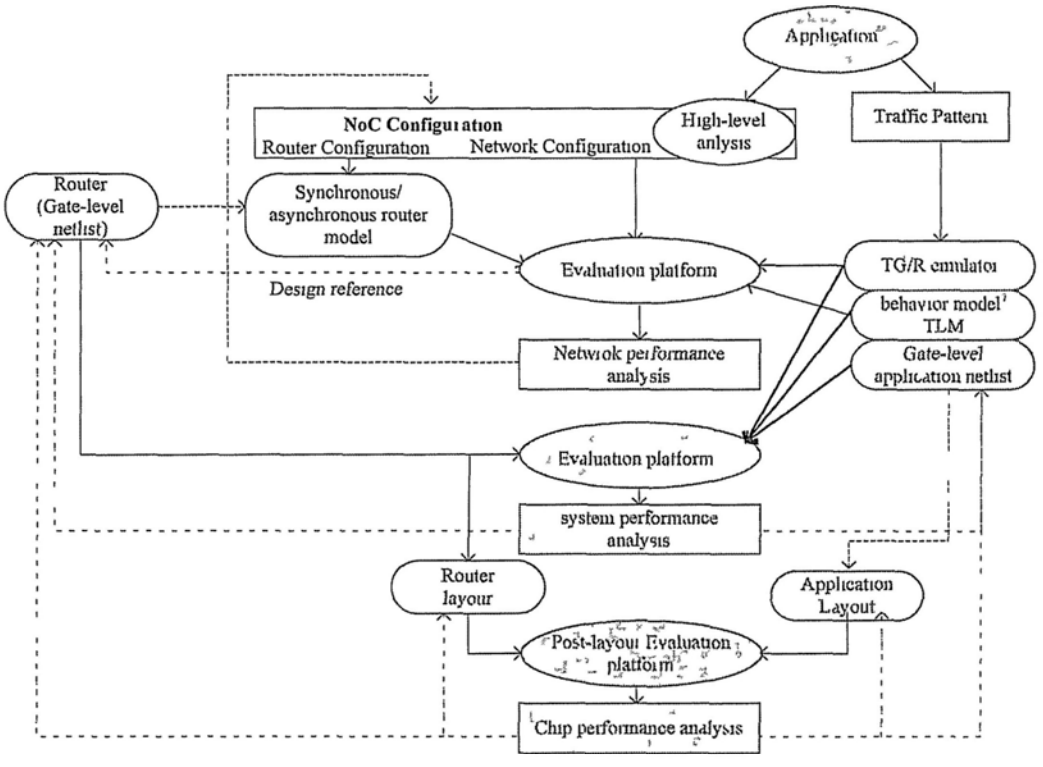


Figure 2.14 NoC design flow

Many traditional works [46], on performance evaluation provide identical analytical models of routers for limited traffic conditions. The assumption of identical router network may not be appropriate for NoC designs. Thus, the NoC architecture can be customized for each specific application to achieve optimal performance, power consumption, and area cost tradeoff, [47]. It is noted that many characteristics of asynchronous circuits are different with synchronous circuits. The evaluation platform must be modified for asynchronous networks. [49]

2.4.2 Traffic Models

NoCs are used to deliver message in systems. The message flows are determined by resources. The traffic of resources involves the detail of the application, which the system executes, such as in Figure 2.15.

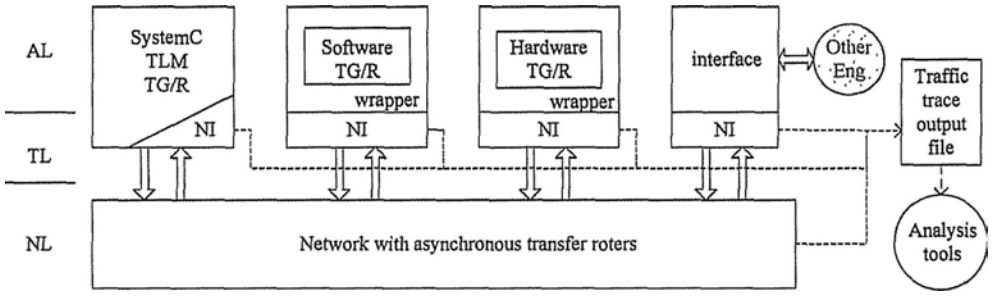


Figure 2.15 The various traffic modules for NoC design

Real applications are the best resources that can evaluate the most accurate simulation results. A function module is proposed to generate the real-application traffic pattern, which is usually described by some system-level languages such as C++, System-C, Java, and so on. A compatible wrapper must be used for a particular type of model. Some models may require a timing controller and others may require conversion of data structure or even the language conversion. Network interface (NI) takes charge in injection of packets to network [40, 42, 43]. Some existing evaluation platforms drop packets to guarantee bandwidth [36, 37, 38]. An infinite queue is applied in (NI) to estimate the ideal waiting delay in queue in other works [39, 41].

Moreover, the traffic emulator is provided to replace complex function modules for quick generation of pseudo-message, which emulates the rule of communication packets in different applications. The rule includes spatial distribution and temporal distribution, which is referred as traffic pattern.

Uniform pattern is a commonly-used benchmark in network. All resources have uniform spatial bandwidth distribution and uniform temporal distribution whatever the network topology is. It means that resources should communicate statistically the equal number of packets with each other resource in the same length of period. The simplest implementation of uniform pattern is that a node sends a packet to each other node in turn after a regular interval of time. The second method is that the destination node and injection time of packets are controlled by a pseudo-random function. The distribution of pseudo-random numbers is statistically even. The second type of

uniform traffic pattern can better evaluate the performance of network-on-chip because it can emulate the case of traffic burst sometimes based on the different pseudo-random numbers in different nodes.

Of course, the communication case of completely uniform pattern is a rare occurrence in real applications. Some other traffic patterns are introduced. **Matrix transpose** is a type of matrix operation, which could happen in some applications of multi-processor chips. And the traffic trace of an audio-video benchmark [26] is regarded as multi-media system (**MMS**) class of traffic pattern. All modules are mapped to 16 nodes of a 4x4 mesh network. **OCN traces** refer to the traffic traces of a multi-processor network [28]. We only use the traffic trace of memory network part, which is a 10x4 mesh network. Two cases (gzip and equake) of traffic bandwidth in OCN memory network are used to evaluate the performance of network-on-chip.

In addition, a statistical traffic model [29] is introduced. We first define the traffic acceptance probability ***p***, which is used to model the various traffic cases. When one source node is to send a packet, any other node in the network consumes that packet with acceptance probability p if the packet arrives. N_i is the node whose distance is *i*. The probability P of having a hop count h greater than d is hence:

$$P_{h>d} = (1-p)^{\sum_{i=1}^{d} N_i} \qquad <2.6>$$

Here, the temporal distribution parameter **H** and spatial injection distribution σ apply uniform distribution to simplify the traffic model. Thus, the type of traffic model is referred as **p-model** traffic pattern.

2.4.3 Traffic Spatial Distribution

Figure 2.16 presents the traffic bandwidth between each two nodes, which is based on

the record of communication flow in original benchmark.

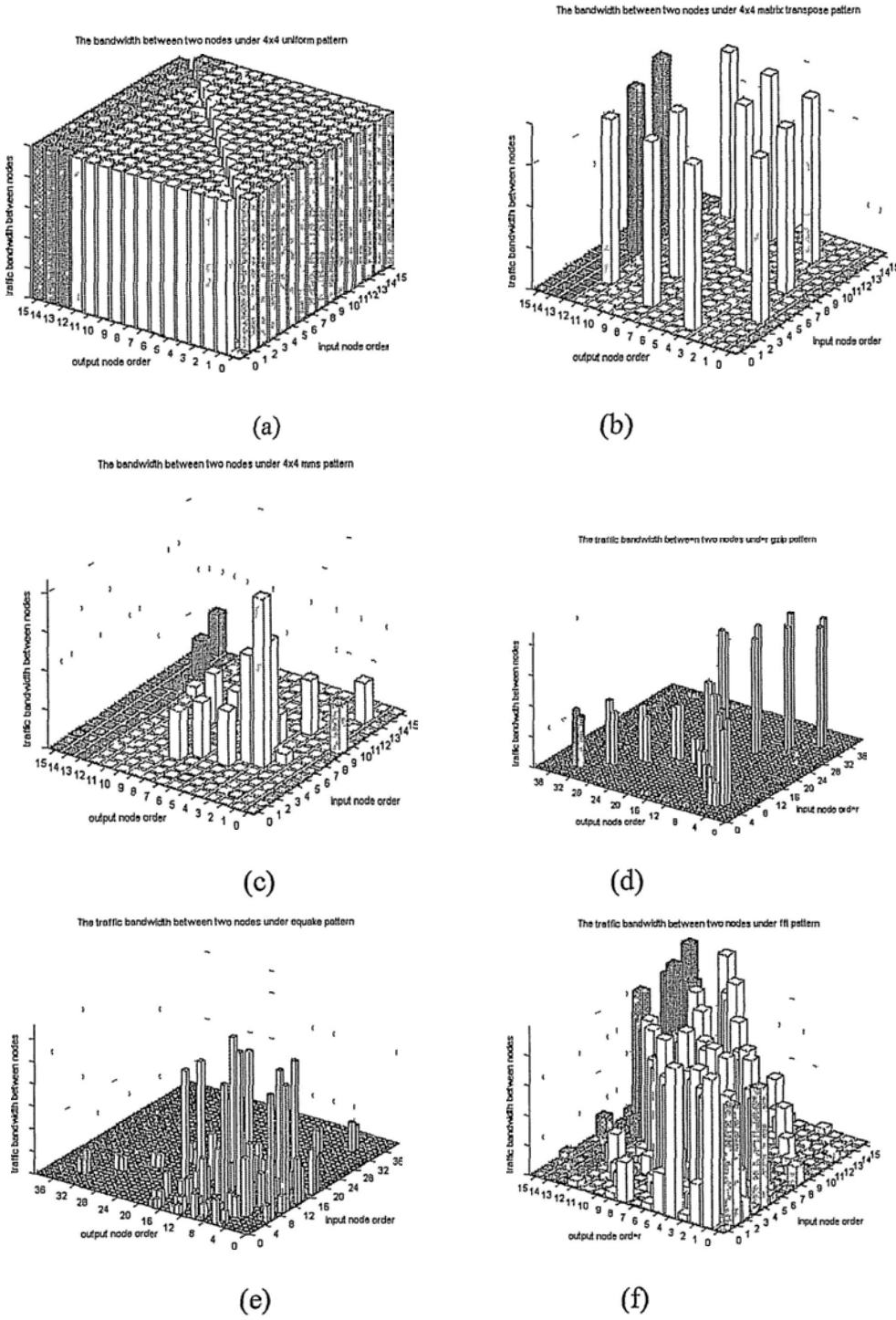
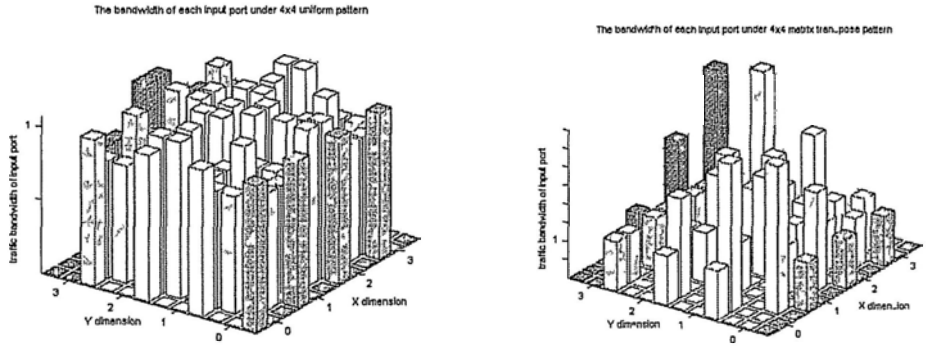


Figure 2 16 The traffic bandwidth between nodes of various traffic patterns (a) 4x4 matrix transpose, (b)MMS trace, (c) OCN gzip trace, (d) OCN equake trance, (e)4x4 p-model fft, and (f) 6x6 p-model jpeg

Uniform pattern is shown in Figure 2.16 (a) and matrix pattern in (b). And (c ~ e) present three traffic traces: MMS traffic trace, OCN gzip and equake traces. Figure

2.16 (f) presents p-model fft case in a 4x4 mesh network. It is known that the destination nodes of packets are all nearby the source according to the given mesh topology because p is 0.422.

Based on given spatial bandwidth distribution and routing algorithm, the tangible traffic flow of each port/channel in NoC can be estimated beforehand. Figure 2.17 shows the channel bandwidth analysis of various traffic patterns based on input flow of each input port, when X-Y routing algorithm is applied. The unit is the average injection bandwidth. The input bandwidth (spatial port/channel bandwidth distribution) of uniform traffic pattern is a little unbalanced although it has balanced spatial bandwidth distribution. And bandwidths of channels between two routers may be larger than injection bandwidth from local because the traffic flows from different sources can overlap in some channels if packets need to traverse multiple routers. And bandwidths of all channels in p-model fft case are no more than injection bandwidth because of its small average hop-count. The port bandwidth distributions of MMS and OCN traces are much unbalanced because of their unbalanced spatial bandwidth and injection distribution.



(a) (b)

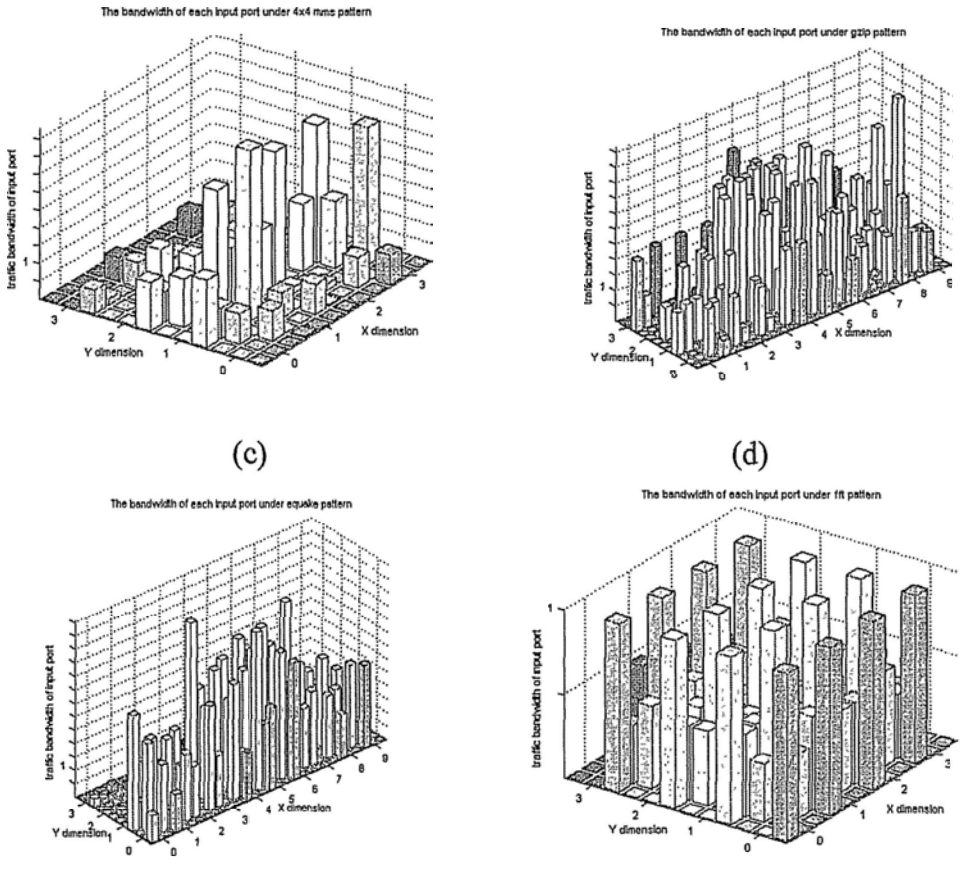


Figure 2.17 The bandwidth analysis of input flow in each input port of various traffic patterns: (a) 4x4 uniform, (b) matrix transpose, (c) MMS trace, (d) OCN gzip trace, (e) OCN equake trace (f) 4x4 p-model fft

Table 2.1 The average traversal distance of hop count (spatial hop distribution)

Traffic pattern	Network size	Average hop counter
Uniform	4x4	3.67
	6x6	5.00
	8x8	6.32
MMS	4x4	3.30
Matrix Transpose	4x4	4.33
	6x6	5.66
	8x8	6.99
p-model jpeg	4x4	3.98
	6x6	3.80
p-model fft	4x4	2.24
	6x6	2.19
OCN equake	10x4	4.17
OCN gzip	10x4	5.00

It is noted that a packet's traversal distance of hop counter is the number of routers that the packet go through, which is listed in Table 2.1 for various traffic patterns in

various sizes of networks.

2.4.4 Router Models

The high-level analysis is based on matlab or C language to evaluate the application communication, which takes charge of determining network and router configuration initially. Then, an evaluation platform is required to estimate the performance of network, which is coded by system-verilog language for the good compatibility. Different router models and traffic model are invoked in the platform for different level evaluation.

- Synchronous cycle-accurate model

Synchronous cycle-accurate model is used for fast simulation to evaluate synchronous network-on-chip. It is between high-level model and post model (post-synthesis and post layout). All signals are controlled by the clock and we assume that all computations of combinational logic can complete in one clock cycle.

The cycle-accurate router model can implement different NoC flow control protocols and can evaluate most performances of on-chip network, such as average packet latency, throughput. And the detailed traffic case and situation of each flit's transmission can be analyzed by tracing some internal status. It can takes charge of the comparison of various network environments and various synchronous routers.

- Asynchronous model

Asynchronous network cannot be simulated by cycle-accurate model because there is no a global clock signal. The evaluation of asynchronous networks requires the asynchronous router model, which is based on the handshake communication. It is more complex than cycle-accurate platform. The delay of different component needs to be modeled to implement the evaluation of network performance. For example, the

long wire is modeled in Figure 2.18 to estimate the wire delay between two routers. All component delays are pre-estimated and set up. The latency of each micro-pipeline stage in asynchronous router model can be computed by these values.

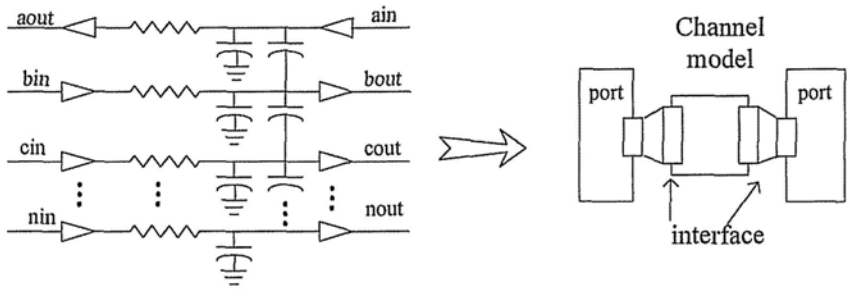


Figure 2.18 Long wire model estimates the wire delay

➢ Post model

Post model includes post-synthesis and post-layout model. They concern the practical technology library.

Post-synthesis simulation is executed based on the gate-level netlist. The gate-level netlist, which can be generated by the design compiler or the manual design, introduces the estimated gate delay and network delay to obtain more accurate evaluation results. And we can get the power consumption and silicon area according to the gate-level netlist. More accurate result is based the post-layout simulation, which is based on the whole system's layout. It is very close to the result of real chip. Of course, to evaluate the cycle-accurate latency, the post-layout simulation is not necessary.

CHAPTER 3. Application-specific Asynchronous Buffer Allocation

Asynchronous communication is useful in low-power low-latency field because of the advantage of asynchronous circuit. The asynchronous communication protocol and basic asynchronous components are discussed in APPENDIX A.

3.1 Asynchronous NoC router

The topology and some communication protocols of asynchronous network-on-chip are similar to those of synchronous network-on-chip.

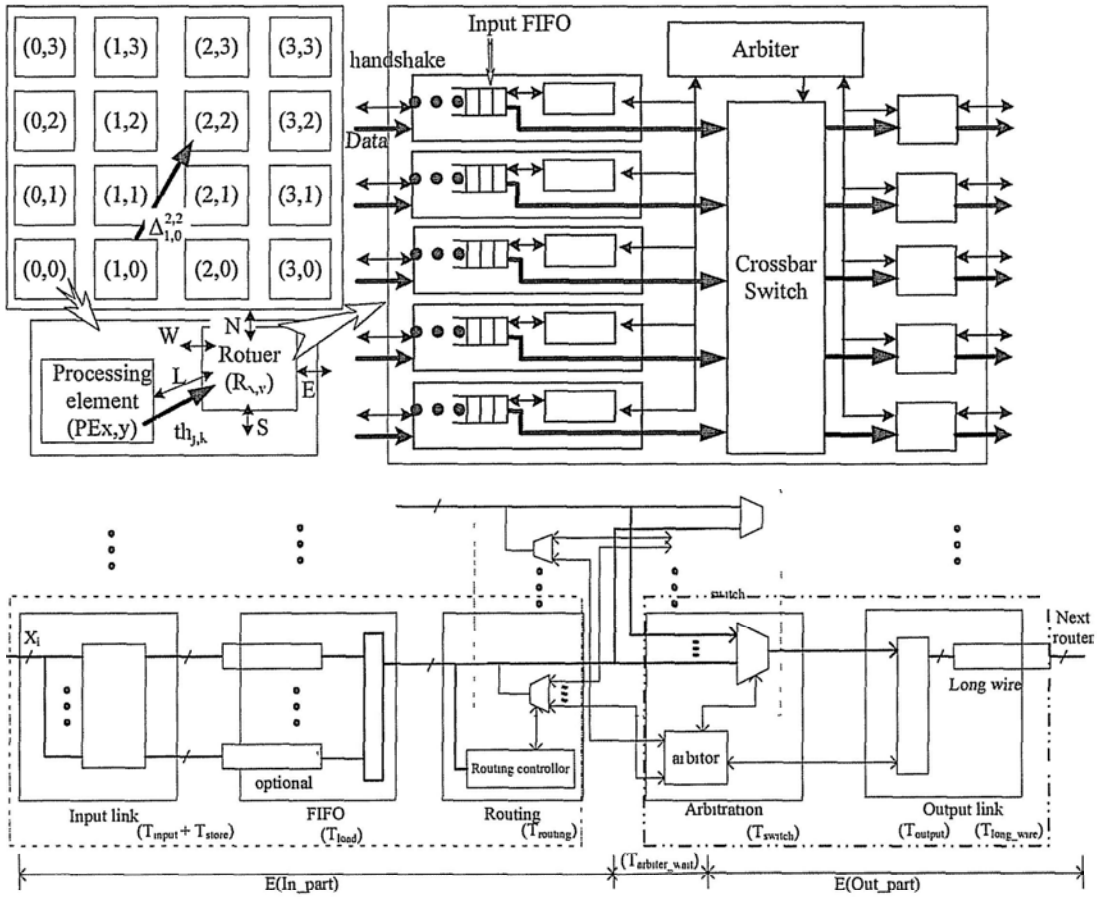


Figure 3.1 An example of asynchronous network and router architecture

The major difference is there is no global clock to synchronize different routers and different pipeline stages. The router is implemented by asynchronous circuits. An example of asynchronous network and router architecture is shown in Figure 3.1.

Such as Figure 2.12, the minimum propagation latency of asynchronous router is the sum of all stages. Thus, the average header latency is the sum of minimum propagation latency and the waiting time because of conflict:

$$T_{router} = T_{in} + T_{buffer} + T_{route} + T_{switch} + T_{output} + T_{conflict}(X) \quad <3.1>$$

$$T = \sum_{path} T^{j}_{conflict}(X) + T_j + T_{wire}(i,j) \quad <3.2>$$

Reducing the latency of any one stage can improve the total latency of a flit. Of course, reduce the latency of the largest stage can increase the throughput better.

3.2 FIFO analyze

In both synchronous and asynchronous router, the buffers will contribute a large part of the total area cost. In [27, 48], the authors show the advantage of customized buffer allocation. We always use a First-In-First-Out buffer (FIFO) composed of a number of registers as data buffer in a router. In a synchronous router, the minimum latency of a flit through the FIFO is at least one clock cycle. It is usually larger than the minimum latency of asynchronous FIFO. And it is noted that the size and architecture will more obviously influence the asynchronous NoC packet latency because of the property of asynchronous circuits.

The basic asynchronous FIFO architecture is Muller pipeline (or Muller distributor) [7], which is a chain of latches as shown in Figure 3.2. It supports 4-phase bundled-data protocol. The main advantage of the structure is simple control logic. But the power and latency of a chain-latch FIFO obviously will increase in networks on chip when the FIFO depth increases.

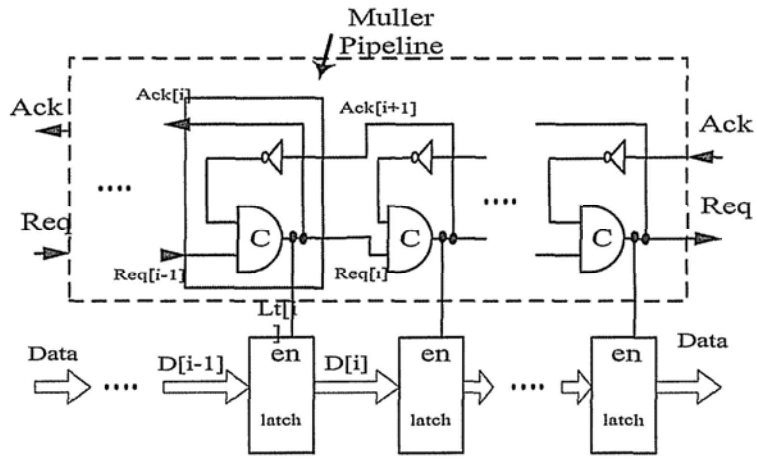


Figure 3.2 Muller pipeline FIFO

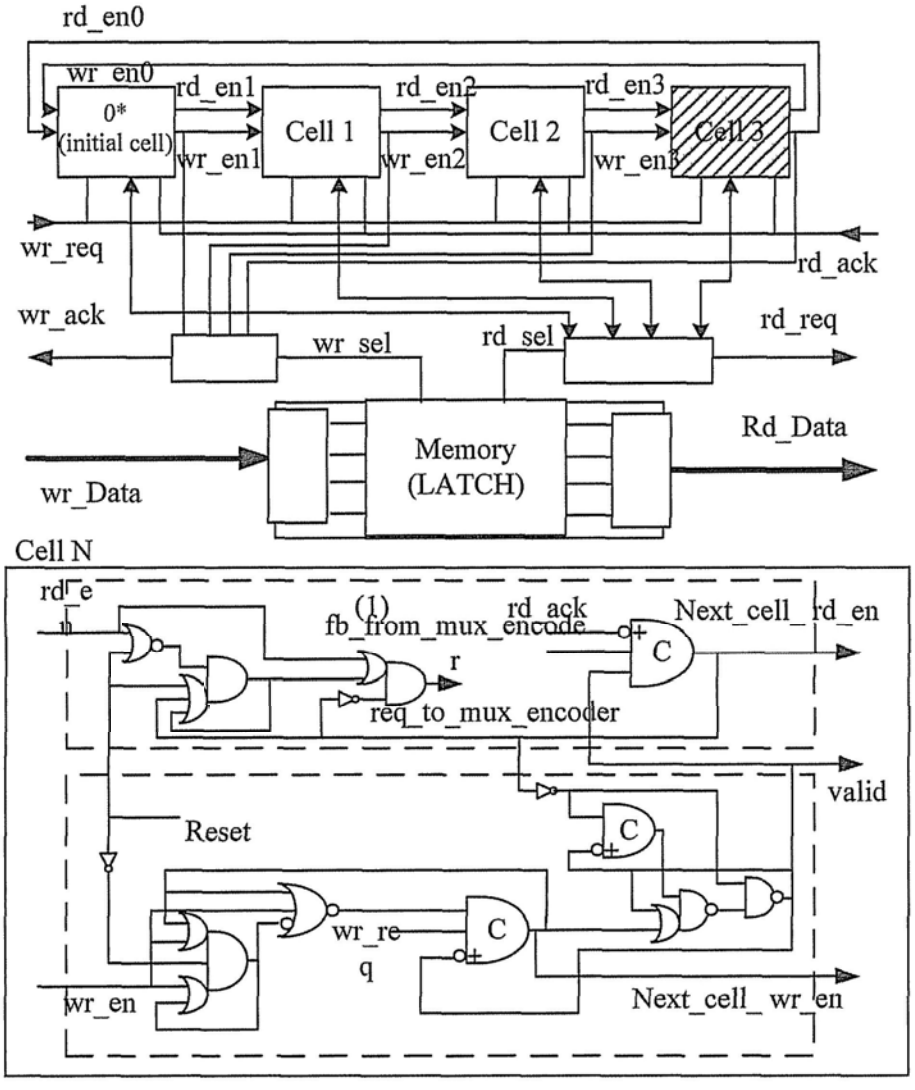


Figure 3.3 Circular FIFO and its control cell

The optimized circular structure FIFO using token ring has lower power consumption and a smaller latency [32]. There are two tokens around the ring: one allows writing

data into the FIFO and one that allows reading data out (push out), shown in and Figure 3.3. Whatever type the FIFO adopts, the area and latency vary with the FIFO depth. The area is nearly linear function of depth. The latency also increase with the growth of depth.

Table 3.1 The different FIFO architectures

FIFO type	Asynchronous		Synchronous
	Muller	Circular	Circular
Area (in nand2)	654	880	1502
Min Propagation Latency	0.6ns	0.65 ns	1 clock cycle
Min Cycle	1.8ns	1.68ns	1 clock cycle

Three types of 4-flit-deep 34-bit-wide, the flit size in our demo design, FIFOs are compared, as shown in Table 3.1, based the UMC 0.13um standard cell library. Performances are compared in terms of area, minimum propagation latency while the FIFO is empty and minimum cycle time based on an ideal environment. Although a circular synchronous FIFO also avoid the waste cycle through the Flip-flop chains, it must apply the same clock as the whole router and results in the loss of latency. And asynchronous FIFOs have overwhelming advantages comparing with synchronous FIFOs in area, latency, and so on.

Figure 3.4 shows that the area and minimum cycles of FIFOs of different depths. The circular's FIFO depth determines the scale of the input bus and output bus. From this figure we can know that the minimum cycle and latency of a circular FIFO will also increase as the depth increases.

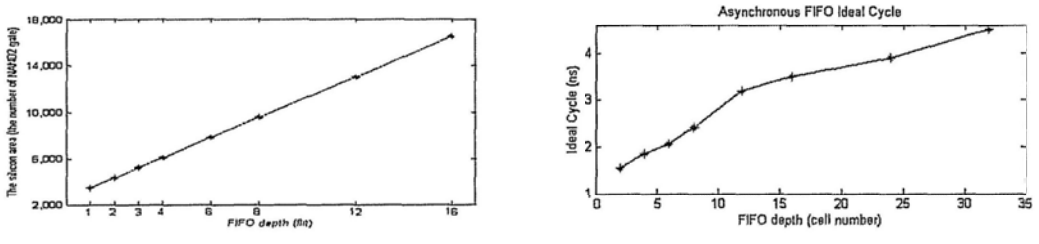


Figure 3.4 The area of router and minimum cycle of asynchronous FIFO with various depths

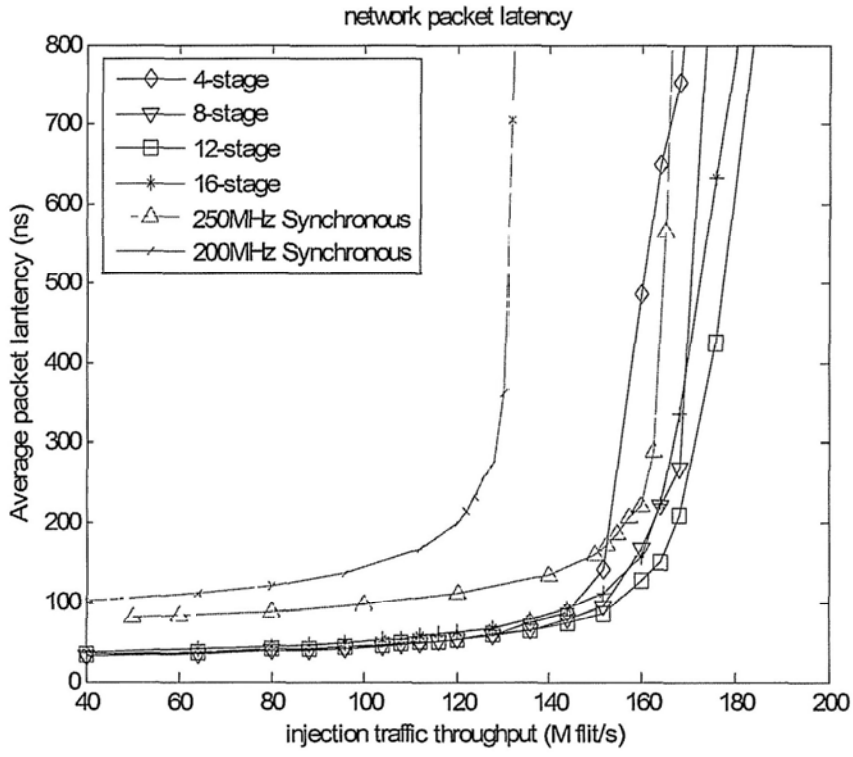
The influence of using large FIFOs as input buffers would be uncertain for overall

network performance. The gain from increasing depth can be counteracted by the loss from the increasing propagation delay. It is necessary to evaluate the network performance of different buffer depths in a NoC system.

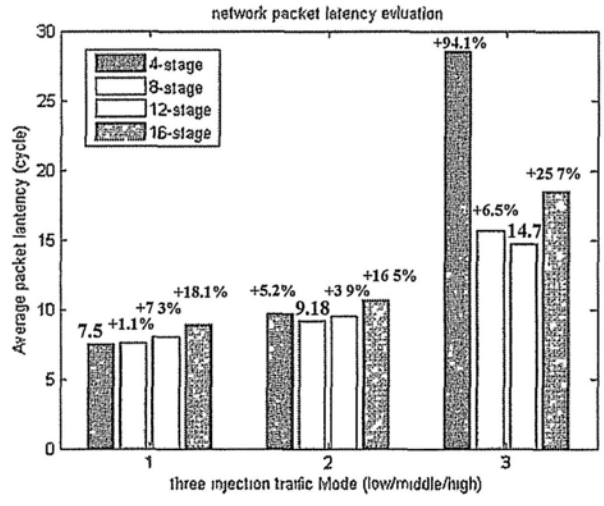
The asynchronous router model used in the evaluation platform is based on the timing information of the UMC 0.18um standard cell library. An asynchronous router uses the 4-phase bundled-data push protocol. It is the general asynchronous handshake protocol and is familiar to most digital designers.

In this section, the evaluation platform applies a uniform distribution and the packet size is four flits. The synchronous network-on-chip with 8-flit-depth FIFO is used as a reference. The router has four stages of pipeline: FIFO write, routing and arbiter, switch transport, and wire transport. The minimum clock cycle of routers is 5ns based on the synthesis result. The clock cycle of all processing elements applies the 5ns both in synchronous and asynchronous NoCs. Figure 3.5 shows the packet latency evaluation results of the routers with various buffer sizes. The asynchronous NoCs apply different FIFO depths from four flit stages to sixteen flit stages, which are uniformly allocated. The simulation shows that asynchronous NoC has obvious advantage as compared with synchronous NoC even while the entire synchronous NoC system applies 250Hz clock frequency (asynchronous NoC's PE only applies 200MHz). In fact, 250MHz has exceeded the maximum operation frequency of synchronous NoC router. And increasing the clock frequency of PEs can reduce handshake time and increase the utilization ratio, which can further improve the latency and throughput because asynchronous routers are compatible to different frequency input data.

And the comparison of average latencies under three chosen injection throughputs: low, medium and high (80, 112 and 152 Mflit/s*node) are individually shown in Figure 3.5 (b).



(a)



(b)

Figure 3.5 The packet latency of various buffer sizes based on traditional buffer allocation when the asynchronous network size is 4x4, traffic is uniform

These data prove that the allocation of a larger FIFO does not mean better performance in asynchronous routers. While the workload of communication bandwidth is given, the router has itself the most suitable FIFO depth to achieve both the better performance and the less cost in asynchronous NoC. Some conclusions are noted. The routers that allocate identical 16-flit-stage FIFOs get worse (larger) average

latency than the routers that allocate identical 12-flit FIFOs even under all traffic throughputs in simulation. It means that the allocated resource is surplus. The performance of 12-flit FIFOs is the best configuration while the throughput is high and become a little worse while the communication throughput decreases. And 4-flit FIFOs and 8-flit FIFOs are suitable for low and medium throughputs respectively.

Further considering the communication flow of each connection between the routers, the optimization for the FIFO depth of each port in NoC routers is necessary according to the different traffic patterns of the given application. The application-specific buffer allocation becomes very important for asynchronous routers because of not only the limit of resource such as for synchronous routers [33] but also obtaining the best network performance that is special for asynchronous routers.

3.3 Allocation algorithm

At first we need to describe the detail of network traffic to implement the application-specific buffer allocation based on given application communication. For each processing element (**PE**) at tile (x,y) shown in Figure 3.6, which is described as $PE_{x,y}$. The injection rate of packets is described by the location inject probability $th_{x,y}$. The parameter $\Delta_{x,y}^{x',y'}$ models the probability of a packet from $PE_{x,y}$ to $PE_{x',y'}$. Traffic spatial distribution characteristic of a network is determined by these parameters. And the parameter SP is the size of the packet (the number of flits per packet). $Rp_{x,y}^{x',y'}$ means a routing path through which a packet is transported from $PE_{x,y}$ to $PE_{x'y'}$. It is determined by the routing algorithm. $\mu_{j,k,d}^{d'}$ means the probability of a packet from d direction to d' direction in $R_{j,k}$; d has five possibilities, N, E, S, W and L. The routing

algorithm will determine which direction to take also.

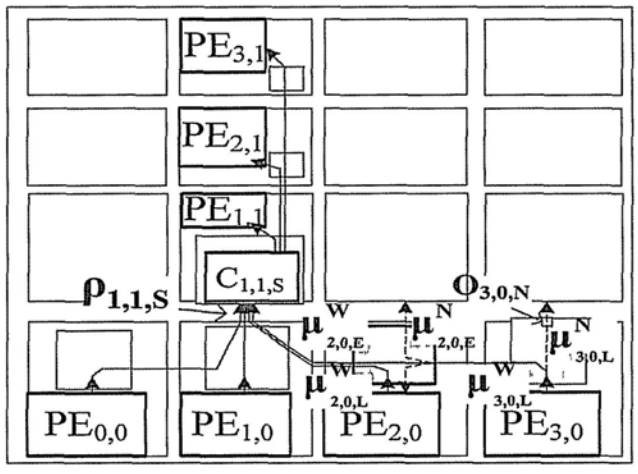


Figure 3.6 *The sketch of traffic parameter*

And $C_{x,y,d}$ is used to indicate the d direction channel in a router, $R_{x,y}$, along a routing path. Through a system-level evaluation the network communication data can be obtained to model the network traffic. Figure 3.6 shows an example of some traffic parameters. In this figure, $PE_{0,0} \sim PE_{3,0}$ are four source communication elements. $C_{1,1,S}$ means the south input channel of the router $R_{1,1}$. $O_{3,0,N}$ means the north output port of the router $R_{3,0}$. The lines describe all traffics from $PE_{0,0} \sim PE_{3,0}$ and through the input channel $C_{1,1,S}$. The parameter $\mu^{N}_{3,0,L}$ and parameter $\mu^{N}_{3,0,L}$ describe two possible transition directions of all packets from the local input port which is connected to $PE_{3,0}$. The example of Figure 3.7 shows all probabilities from south input port to each output port at tile (1,1). It is also determined by the routing algorithm.

Each application has a fundamental requirement of communication bandwidth (B) from the on-chip network. And average packet latency (L) is used as the metric for NoC communication performance. The aim of buffer allocation is to achieve the best possible performance. Also fewer buffering resource (shallower FIFO) means smaller silicon area (A).

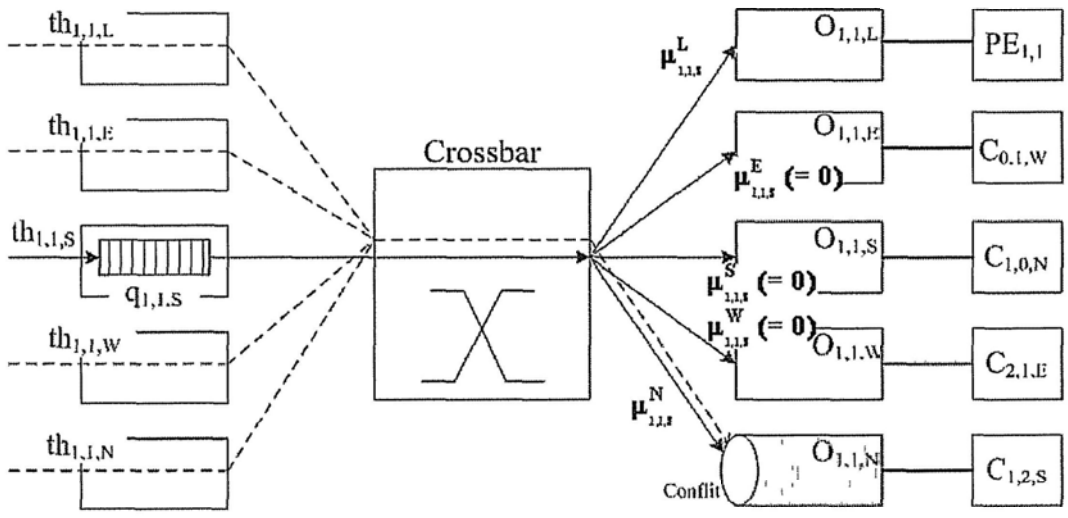


Figure 3.7 The queue model (it is based on dimesion-order routing packet mobility)

For one input channel, a packet arrival rate is an important parameter to allocate the buffers. $\rho_{x,y,dir}$ denotes the packet arrival rate at channel Cx,y,dir which can be formulated as follows: Given $\mu^{d'}_{j,k,d}$, $th_{j,k}$, and $Rp^{x',y'}_{x,y}$,

$$\rho_{x,y,dir} = \sum_{\forall k}\sum_{\forall j}\left(th_{j,k} \bullet \prod_{(d,d')\in Rp^{x,y}_{j,k}} \mu^{d'}_{j,k,d} \right) \qquad <3.3>$$

According to different routing arithmetic, the packet arrival rate is various. While routers apply deterministic dimension-order routing arithmetic, even if the PEs send the identical-distribution packets the traffic rate is uneven for each channel, as shown in Figure 2.17. We can see the difference of input channel arrival rates between different input ports of all routers.

In Equation 3.4, $\beta_{x,y,d'}$ is the probability of output d' is occupied by other input channels. $\lambda_{x,y,dir}$ describes the usage state of the next output direction dir router's FIFO. The parameter can consider the influence if the buffer of next router becomes full.

$$FR_{x,y,dir} = \sum_{\forall d'} \left[AP_{x,y,dir} \bullet \rho_{x,y,dir} \bullet \mu^{d'}_{x,y,dir} \bullet \beta^{dir}_{x,y,d'} \bullet \lambda_{x,y,d'} \right] \text{ , in which}$$

$$\beta^{dir}_{x,y,d'} \approx \left(\sum_{\forall D \neq dir} \left(\rho_{x,y,D} \mu^{d'}_{x,y,D} \right) + 2 \bullet \sum_{D1 \neq D2 \neq dir} \left(\rho_{x,y,D1} \mu^{d'}_{x,y,D1} \rho_{x,y,D2} \mu^{d'}_{x,y,D2} \right) \right) \quad <3.4>$$

At first iteration, we simplify to assume that the full states of all buffers are few. The default value is 1 for each output channel between routers and 0.8 for each output channel to a local PE because the local connection is near. We may need to amend λ for each output based on the state of the application's communication. It is noted that the λ parameter is influenced greatly in next iteration if too small of a buffer size is allocated in the current iteration, which leads to more possibility of FIFO full. AP denotes the influence of the packet size. We can get $AP = \sum_{n=1}^{SP} n$ based on the architecture of our platform. If virtual channels are introduced, the AP decreases because of decreasing probability of head blocking. The Figure 3.8 shows an example of FIFO desired ($FR_{x,y,dir}$) in each channel.

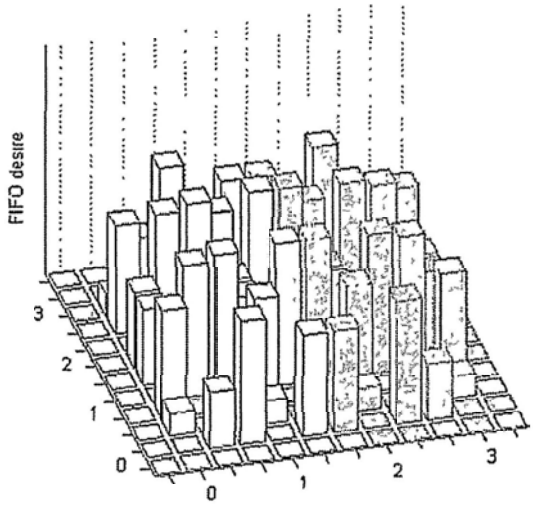


Figure 3.8 The desired degree of buffer in each port under uniform traffic pattern

The basic idea of our buffer allocation method is that reasonable buffering resources,

which are based on a buffer library being built in advance, need be allocated to each input channel to fit the desired degree of FIFO ($FR_{x,y,dir}$). It is noted that the channel which has large arrival rate is not necessarily the channel which needs more FIFO space. The degree of desiring FIFO space is mainly decided by the probability of packet blocking. The final buffer queue depth qx,y,dir is derived by FRx,y,dir. The allocated function $q_{x,y,dir} = q_{base} + f\left(FR_{x,y,dir}\right)$ is not linear because the large buffer size will introduce the additional loss of latency. It is the reason that the asynchronous timing library is required in our buffer allocation method. And the allocated function of FIFO depths can be modified to meet the different requirements of different optimization aims, such as best performance or enough performance with less area cost.

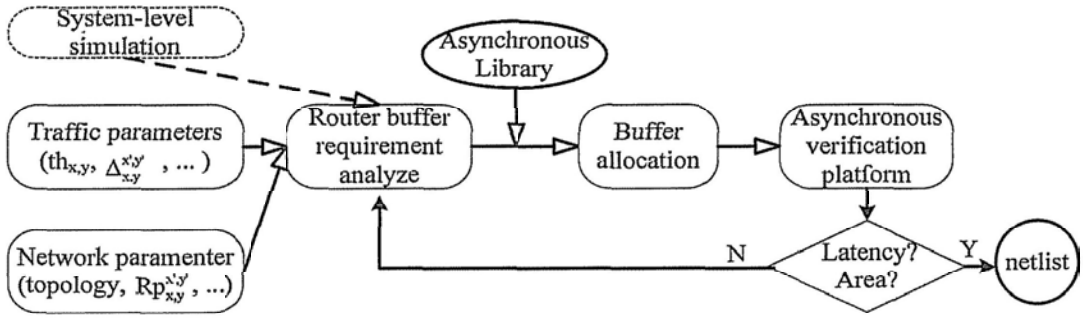


Figure 3.9 Buffer allocation flow for asynchronous NoC

Because smaller buffers are applied than traditional identical buffer-allocation, the shorter latency and the less cost could be achieved. Our buffer allocation method can choose the suitable resource to offer good throughput bandwidth according to the traffic through each buffer.

The buffer allocation flow is shown in Figure 3.9. According to the detailed application and NoC topology and protocol, we can get the traffic parameters and the network parameters. By the above formulas, we will analyze the buffer requirement. And

because the asynchronous circuit has no clock, an accurate asynchronous timing model library needs to be introduced. Then the optimization aim is decided. If the performance needs to be improved, the optimization aim of application-specific buffer allocation is to choose the suitable FIFO depth to achieve the best performance. And when the area cost is of more concern, the optimization aim becomes to choose the least buffer to meet the requirement of average latency. The buffer allocation result should be acceptable. If we need a better result, the parameters and allocation equation of the next iteration are optimized by the verification result of the last iteration. We need to modify the parameter λ based on the verification result and modify the function to choose a more suitable FIFO depth. The application-specific buffer allocation routers require further verification of the real network performance by post-synthesis simulation.

We choose some types of traffic to evaluate our application-specific buffer allocation (A-SBA) compared to the traditional identical buffer allocation (IBA) for asynchronous NoC. To simplify the test, they all apply the uniform temporal distribution. It is noted that the application-specific communication should be a reasonable throughput region for the network. It means that the traffic throughput is not too large and not too small.

The first traffic pattern is uniform distribution communication in a 4x4 mesh network. To further evaluate our buffer allocation of asynchronous router, a one-hotspot uniform distribution is applied. Under the one-hotspot pattern, one node has more distribution probability to receive packets than others do. The probability of packets from other PEs to hotspot is two times as much as the probability to other PEs, which is a uniform distribution. The model simulates the state of existence of a hotspot. In this section, we

choose one edge node $PE_{0,0}$ (hot1) or one centric node $PE_{1,1}$ (hot2) as the hotspot Moreover, the p-model traffic pattern, discussed in section 2.4.3, is adopted. P is 0.898 and 0.228, respectively, in the gzip case and the mpeg2 case.

Table 3.2 The spatial hop distribution of traffic patterns used in this chapter

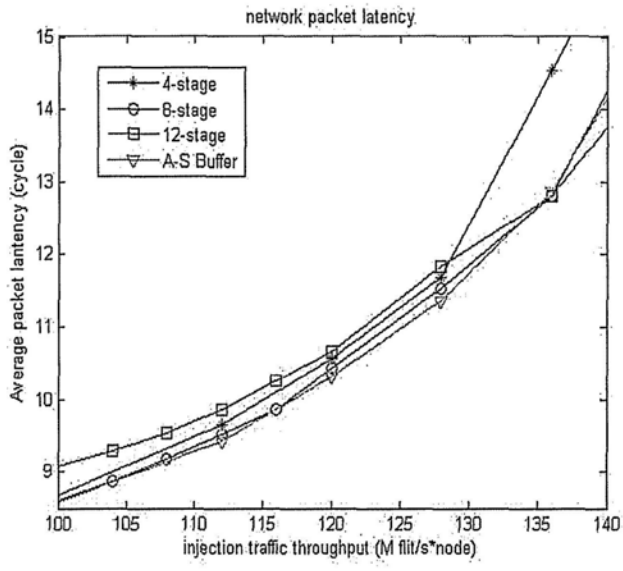
	Spatial hop count distribution (%)						Average hop^
	Hop1	Hop2	Hop3	Hop4	Hop5	Hop6	
Uniform	20.0	28.3	26.7	16.7	6.67	1.67	2.667
gzip	79.2	18.0	2.48	0.25	0.05	0.01	1.2395
mepg2	23.0	23.4	15.9	14.3	16	7.32	2.9862
Hot1	19.53	27.73	26.56	16.80	7.03	1.95	2.6872
Hot2	20.31	28.91	26.56	16.02	6.25	1.56	2.625

^The number of path router (Traversal distance) = average hop + 1

The channel arrival rate for uniform-distribution traffic has been shown in Figure 2.17 and the FIFO desired degree of each channel is shown in Figure 3.8. Here the buffer allocation configuration is optimized to achieve the best performance based on the requirement of the 120 Mflit/s*node communication bandwidth which is about 80% of the saturation throughput in the 4x4 mesh network. Then we can compare the performance of the new NoC composed of the A-SBA routers to the IBA asynchronous NoC, shown in Figure 3.10. The left figure (a) shows the average packet latency from 100 ~ 140 Mflit/s*node. The figure (b) shows the comparison of latencies at 120 Mflit/s*node communication traffic and silicon area for different buffer allocations.

For IBA routers, the best network performance is achieved as 10.42 cycles (52.1ns) while the routers apply the 8-flit FIFO. And the network performance of the routers that apply our A-SBA can reduce 1% latency with less cost. And we can see the new buffer allocation can achieve the best performance in a wide region around the optimization bandwidth not only at one point of 120 Mflit/ s*node. Also the comparison of FIFO core memory areas shows our new buffer allocation can save large area. It is noted that the

area does not include the buffers connected to the NoC external port and the local PE port. It only accumulates the FIFO between routers (48 channels). The A-SBA can save 25% of the area compared to 8-flit-FIFO routers which can achieve the best performance with traditional buffer allocation.



(a)

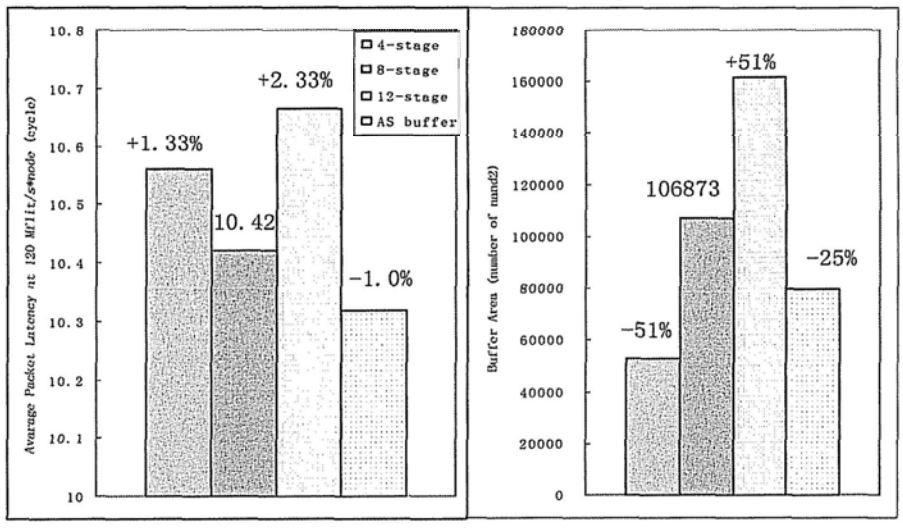


Figure 3.10 (a) Average packet latency, (b) latency comparison under 120Mflit/s*node injection bandwidth and area comparison when packet size is 4-flit for different buffer allocations

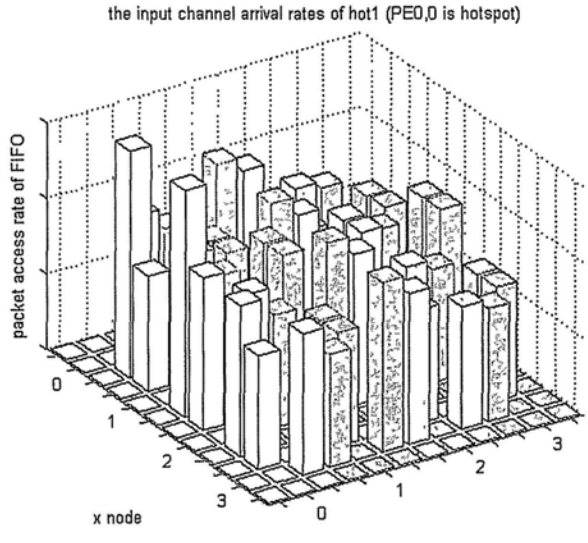
Table 3.3 The comparison of latency and area when packet size is 4-flit, injection bandwidth 120Mflit/s*node

Buffer type	latency(cycle)	Ratio
4-flit	23.4044	118.19%

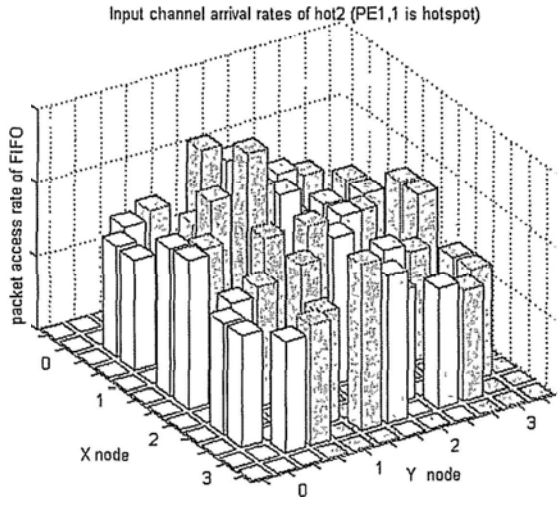
8-flit	19.8025	1
12-flit	19.9218	100.602%
A-S buffer	19.55	98.7%

And if we modify the packet size from 4 flits to 8 flits, the result is shown in Table 3.3. Our A-S buffer NoC has 1.24% performance improvement as compared with 8-flit FIFO router which has best performance in all traditional buffer allocation NoCs. Because the 8-flit packets desired degree more FIFO while the parameter AP increases, the routers need to be allocated more buffers and the area saving becomes 22% compared to the best one, 8-flit-FIFO NoC.

Under one-hotspot uniform traffic condition, the parameters are different from uniform distribution communication. New buffer allocation aim is to optimize the performance of average packet latency at the injection throughput of 120Mflit/s*node and the packet size is also 4 flits. At first, the channel arrival rates are calculated as shown in Figure 3.11. The large change of arrival probability of the hotspot has effects to all channel arrival rates. Then according to the rate and the mobility μ of each channel, the desired degree of FIFO is derived and buffer sizes are allocated in each channel. The choices of FIFOs are 3~10 flits. The performance comparison is shown in Table 3.4.



(a) Hop1 - PE(0,0)



(b) Hop2 – PE(1,1)

Figure 3.11 The bandwidth distribution of each input channel under two hotspot traffic patterns

Table 3.4 The comparison between UNoC and BANoC under application-special distribution patterns

	Type	Hot1	Hot2	gzip	Mpeg2
Average Packet Latency (cycle)	IBA NoC	8-flit	6-flit	4-flit	8-fit
		11.394	11.056	5.31771	6.9844
	A-SBA NoC	11.2	10.88	5.30	6.9168
Performance improvement		1.7%	1.59%	0.33%	0.96%
Area saving		29.7%	14.5%	2.8%	8.2%
The traffic ratio (hot/clod to av.)		4.32/ 0.45	4.93/ 0.54	1.05 /0.93	1.30/ 0.72

In this table, all average packet latencies are calculated under the throughput of 120 Mflit/s*node. The traffic ratio indicates the degree of uneven packet traffic. It includes two values that are the ratio of busiest channel to the average and the ratio of the most vacant channel to the average. All IBA NoCs have the best performance as compared with A-SBA NoC for a given application and throughput. The result of A-SBA NoC saves 29.7% silicon area compared with 8-flit-FIFO IBA NoC in the hot1 pattern and 14.5% area compared with 6-flit-FIFO IBA NoC in the hot2 application. And the latency can be reduced by 1.7% and 1.59% respectively. It is noted that the result is only through one or two loops of iteration. Better performance may be achieved after more iterations. Further buffer allocations are verified for different application-specific spatial-distribution traffic patterns (local and global, presented in section 5.1).

Evaluations of these NoCs on our simulation platform are also shown in Table 3.4.

From these simulation results, one can conclude that our A-SBA NoC can get the better network performance of packet latency by less cost for different traffic applications. The communication flow of the whole network is more uneven, the advantage of the new buffer allocation is larger. The excess resource can be avoided and the best network performances can be achieved.

Also even if the application which has average communication throughput in each channel or we apply the traditional buffer allocation, the new allocation flow is also useful to quickly determine the suitable buffer size according the design specification. It can reduce the region of choice based on the buffer analysis.

3.4 Summary

We concentrate on the high-performance and low-cost buffer allocation method for asynchronous router design. The common idea was identical increment of buffers for all routers to offer higher throughput and to reduce the latency. But this method is not good enough for a given application. And the increment of buffer needs more silicon area and maybe counterproductive with overlarge FIFOs in asynchronous routers. We therefore propose an uneven-buffer-allocation router based the tangible communication distribution of the application instead of traditional uniform buffer routers. According to the property of asynchronous circuits, in asynchronous NoCs the new buffer allocation method can obtain the advantage of lower silicon area cost and better network performance at same time.

Our buffer allocation method is mainly dependant on the degree of desired buffers in each channel. These parameters can be obtained by calculation or simulation. We give the formulas to quickly estimate the usage of FIFOs in the routers. A comparison

between the new buffer allocation routers and traditional routers verifies the method. All evaluations of our new application-specification buffer allocations for different traffic patterns demonstrate that the new buffer allocation method is indeed useful for asynchronous NoCs.

The more accurate formula is required for variable temporary distributions. The timing distribution is an important parameter to allocate the final FIFO queue depth. The more uneven the temporary distribution is, the larger FIFOs are required in routers.

CHAPTER 4. Multicast

We know that a bus is very efficient in broadcast communication since all elements are directly connected to the bus. Although broadcast communication naturally means high power consumption with so many redundant transfers, many applications still need this type of communication, such as transferring global states, requesting for vacant resources, managing and configuring the network, implementing cache coherency protocols, etc. Some efficient methods facilitated by hardware are necessary to improve broadcast traffic in NoC [52].

Multicast operating in the hardware approach delivers messages to all parties of the group through a path or tree of nodes, each of which forwards messages to one or more outgoing links. A message may be replicated at intermediate nodes and forwarded along multiple outgoing links towards the set of destinations. Adopting Quality-of-Service (QoS) routing will provide additional support to efficient multicast. In a mix of unicast and multicast traffic, QoS can guarantee a certain capacity of multicast traffic by giving a higher priority to multicast messages.

4.1 Multicast and unicast

In parallel computers, there are many studies about collective communication, including multiple one-to-one, one-to-all, all-to-one, and all-to-all communication [5]. According to delivery semantics, there are two types:

- ✓ unicast delivers a message to a single specified node;
- ✓ multicast delivers a message to a group of nodes that have expressed interest in receiving the message; broadcast is the special multicast where the group of nodes are all nodes

And multicasting [53, 54, 56], an important part of these studies, can be supported by either a software or a hardware approach.

```
P5:                    P0:          P1:          ......    P15:
Send(msg1,P15)
Send(msg1,P14)         ......       ......                 ......
Send(msg1,P11)         ......       ......                 ......
.....                  recv(msg1,P5) recv(msg1,P5)         recv(msg1,P5)
Send(msg,P4)           ......       ......                 ......
Send(msg,P1)
```

Figure 4.1 An example of software implementation of broadcast

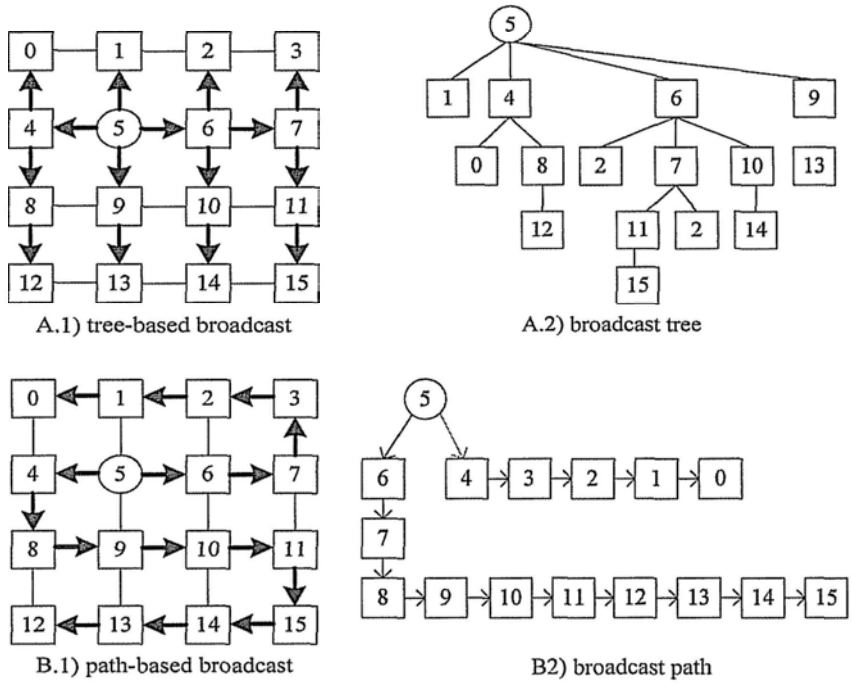


Figure 4.2 Hardware implementation of broadcast (a) tree-based (b)path-based

Most NoC routers only can apply a software operation to implement the multicast. It is implemented by sending a copy of the message from the source element to every destination or to a subset of destinations, as in Figure 4.1. As NoCs are usually performance conscious, we propose a hardware implementation in place of software. With the tree-based multicast, messages are sent through a predefined tree path of nodes. With a path-based multicast, the transfer of messages does not branch out at intermediate nodes. It is noted that a path-based broadcast operation will send a message through all nodes, which can be very long and causes large latency. In order to reduce

the length of path, the set of destinations can be divided into multiple disjointed subsets. Figure 4.2 shows the examples of tree-based broadcast and path-based broadcasts in a 4X4 mesh network.

4.2 Hardware Multicast

4.2.1 Protocol

We use customized algorithm and routers to implement the QoS-aware multicast communication in a best-effort network. The source can communicate with a set of destinations by the predefined group information with diversified algorithms. The scheme has the best efficiency. The detailed communication protocol is described.

The packet consists of three types of flits: a header flit with routing address and routing command, body flits and a tail flit indicating end-of-packet (EOP). Each flit contains bits indicating its service-level (SL) and type, as in Figure 4.3. Within a flit, the SL bit indicates whether the packet is unicast or multicast service-level. The flit type bits are header bit and tail bit.

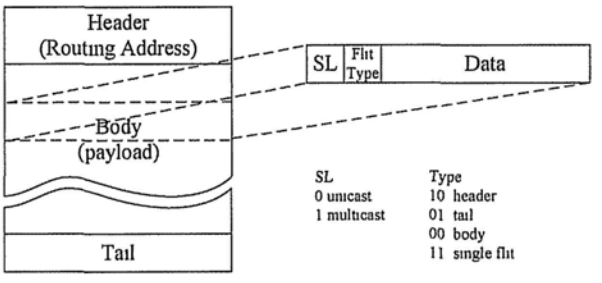


Figure 4.3 Data structure for multicast

With unicast communication in the wormhole network, the routing is performed only when the header flit of a packet becomes the earliest-arrived flit of one IP. The format of unicast header flit is shown in Figure 4.4. The "src addr" is optional, which is the NoC address of the processing element. The "des addr" is necessary, which is the NoC address of the destination.

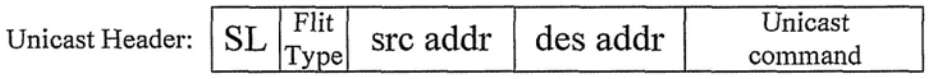


Figure 4.4 Unicast head flit format

Due to difference in requirement as compared to unicast, a multicast header flit format is modified as shown in Figure 4.5.

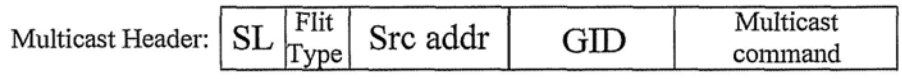


Figure 4.5 Multicast head flit format

The main modification is the "GID". The "GID" is the multicast group identity number, which is unique for each multicast group. Some GID ranges can be reserved for the broadcast on chip. And the predefined group of multicasting will determine the routing result in each routing block. Then the multicast packet is transmitted to all members of a multicast group according to the "GID".

When the number of group members is small, the format can be expanded to support multiple arbitrary destinations whose addresses replace the "GID". But the total number of destinations is constrained because one header flit can only accommodate a finite number of addresses based on the bandwidth. It needs routing algorithm support too

4.2.2 The router architecture

All multicast packets' requirements are managed by the additional multicast control block. Figure 4.6 shows the hardware-multicast router architecture. It adopts QoS, where there is multicast service level (SL) and unicast SL. The multicast routing cell takes charge in the implementation of the multicast algorithm, such as tree broadcast, path broadcast or anyone group. It uses GID information embedded in the head flit to search a multicast lookup table that is pre-computed according to user's demand. Then the result, which is multicast request, is sent to multicast switch allocator for arbitration between other multicast requests from other ports.

The multicast packets have higher priority than all unicast packets. If an output port is

required by a multicast packet, the port blocks unicast to apply this port. The free physical output channel can also be used by a unicast packet. And the multicast enable signal can bypass the control of the multicast control cell. Then the multicast SL channel can serve any unicast packet normally if the router has no the multicast communication.

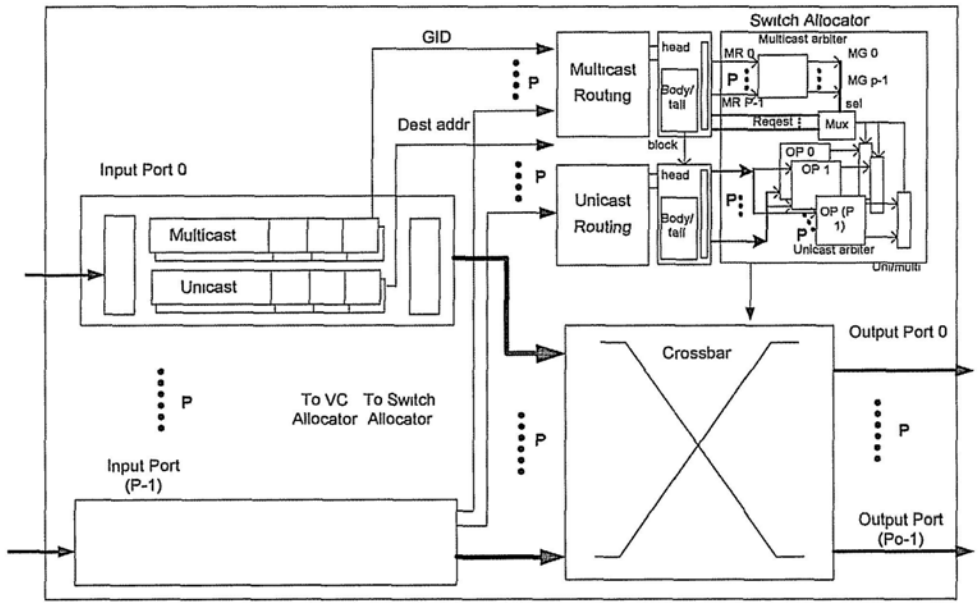


Figure 4.6 The block sketch of router supporting multicast

A synchronous multicast router just increases the multicast routing and arbiter part as compared with a unicast router. An asynchronous multicast router is similar to the synchronous version. There are several special-designed asynchronous components, such as shown in Figure 4.7.

To avoid deadlock in a switch like the example in Figure 4.10, the router only allows non-conflicted multicast traffic. To simplify the design, usually only one multicast packet can go through the router.

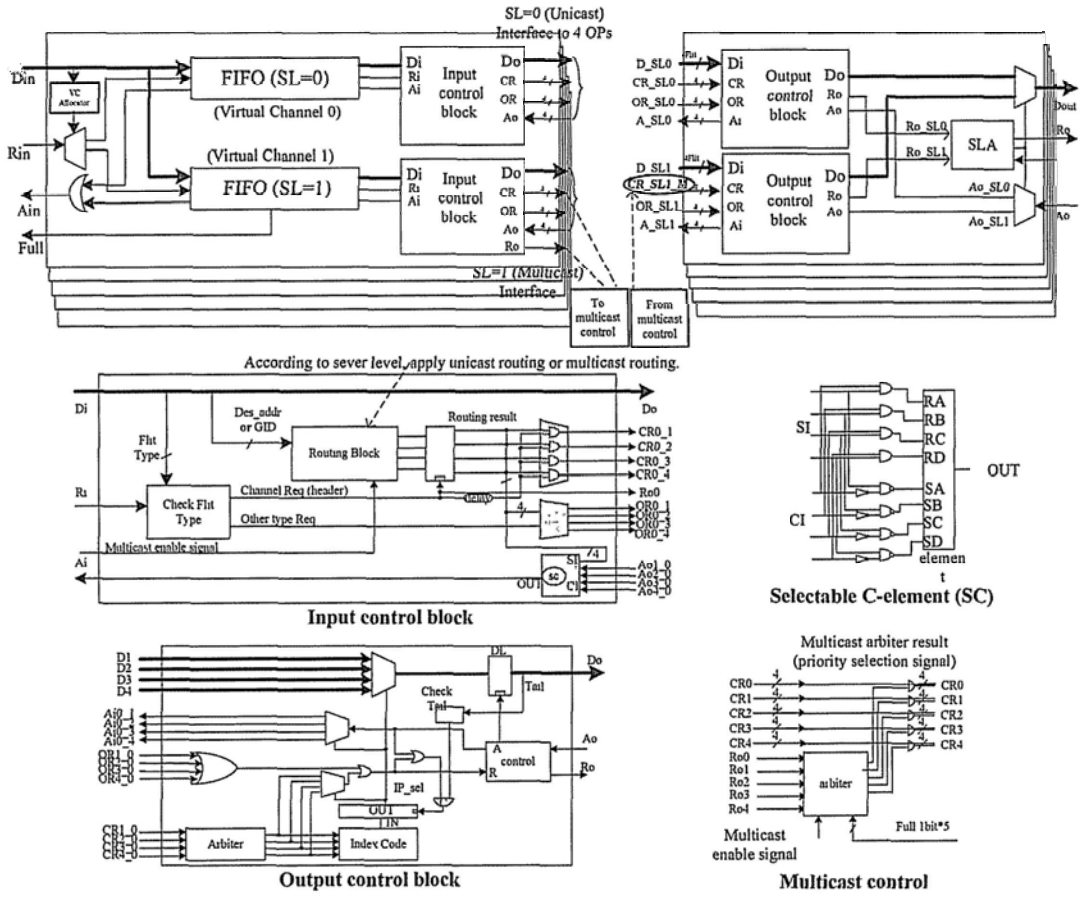


Figure 4.7 An implementation of asynchronous multicast router

The multicast enable register (MRE), providing the enable signal, determines whether the router will support multicast service. And there is a GID Routing Lookup Table (GRLT). It is noted that each port has a set of routing information for itself. Different multicast groups need to set their pre-computed routing result in the GRLTs of all member of a multicast group and all path routers connecting the members.

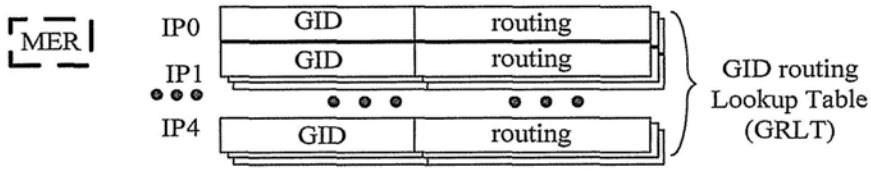


Figure 4.8 Multicast routing table and enable register

During unicast, the routing cell applies the XY routing algorithm. During multicast, the routing cell performs the routing according the predefined GRLT. "Routing" indicates the routing result with the corresponding GID. They determine which

orientation outputs are selected. According to these routing results, the multicasting packet with special GID can be sent to a local element or duplicated to other routes. The multicasting communication consists of two phases: 1) Group establishment: The multicast group source needs to setup the group by setting the GID register and multicast routing registers of the destination node and path nodes. According to the requirement, the nodes also can send back responses after they setup the registers. And the source can use the existing GID if the setting is appropriate for itself. 2) Multicasting communication: After setting up the group, the master sends the packets to each group destination according to the defined path by routing registers.

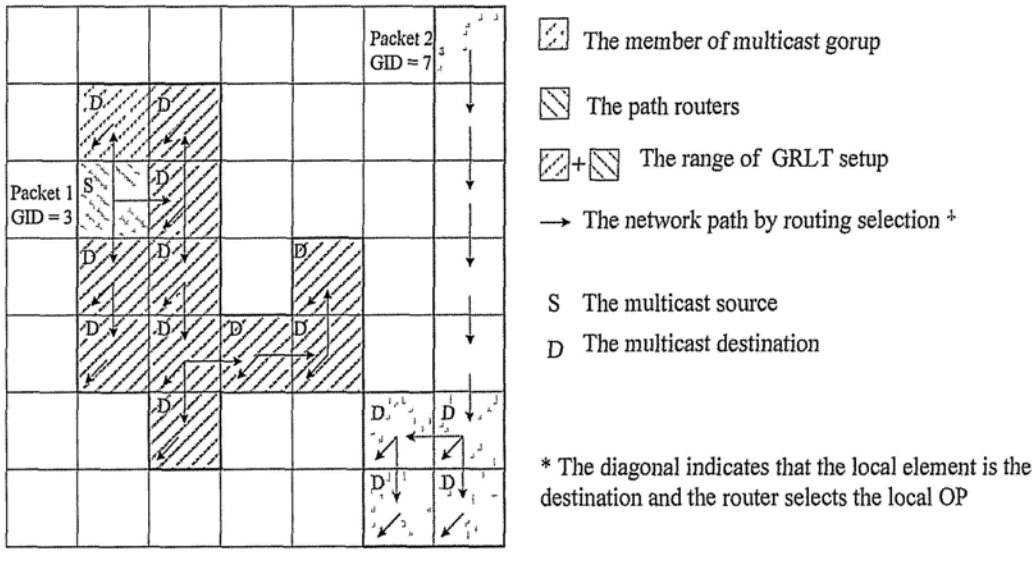


Figure 4.9 Multicasting transmission according to routing

As the packet 1 is shown in Figure 4.9, we can define all destinations as a group. Multicasting is performed only along these group members. The packet 2 of Figure 4.9 is through many path nodes whose local processing elements are not the multicasting destinations. These path nodes only send the packet to the next node through the router like unicast, determined by the GRLT. By default, we can define the GRLT for some

special aims in advance. For example, GID0 is defined as broadcast. The routing scheme provides enough agility and good routing performance. We can apply either tree-based routing algorithm or a path-based routing algorithm. It is up to the setting of GRLT

4.2.3 Deadlock avoidance

A deadlock occurs when some packets cannot advance toward their destination because they are waiting for some event that will not happen. Examples of multicast deadlock are shown in Figure 4.10.

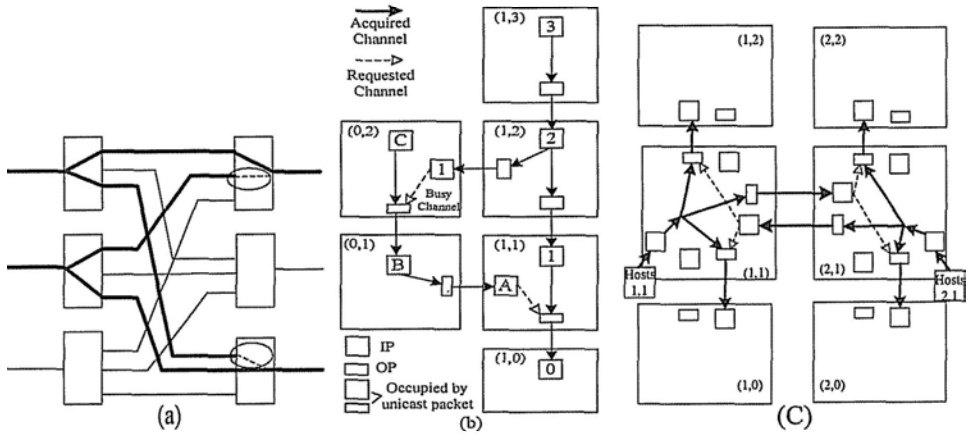


Figure 4.10 Examples of deadlock

Deadlock in a switch is shown in Figure 4.10 (a). Figure 4.10 (b) is the interlocking state in which two multicast packets are requesting the routers that are already occupied by each other. Then two packets will not release the channel if the input buffers have been full in (1,1) and (2,1). Another deadlock happens, as in (c), while any one of the requested channels is busy by other blocked unicast packet because the multicast will be on hold if any branch is blocked.

The case of Figure 4.10 (a) has been overcome by a multicast arbiter. And because the

router makes use of the QoS, the multicast packet has priority over the unicast to avoid the case depicted in Figure 4.10 (c). It means unicast packets cannot block any multicast packet. To avoid the inter deadlock like Figure 4.10 (b), a large enough input buffer to save at least one whole multicast packet is the sufficient condition. It means a multicast FIFO of IP must be larger than the maximum size of one multicast packet to avoid this type of deadlock. The second sufficient condition is the sane multicasting routing setting. We must remove the cyclic resource dependency for each GID routing table. And the system must restrict the number of multicasting sources. It is suitable that no more than two sources send packets to one element as a multicasting destination at the same time. For example, a system can allocate a global broadcast master and several regional multicast sources. But while there are too many sources to send the multicast packets with high throughput at same time in a given area, it gives rise to the potential risk to block the traffic. The source can release its multicasting packets through unicast channel to free the channel.

4.2.4 Evaluation

Multicast is requested in many NoC applications. Particularly, a shared-memory multiple-processors (SM-MP) design needs multicast communication for cache coherency and whole-chip processors state control. There are two elements with multicast right on chip. One is the main control processor. Another is the shared memory that will broadcast cache update packets to each element on chip. Our asynchronous router is very suitable for this class of applications.

The proposed asynchronous NoC router with multicast support is compared with the

asynchronous router without multicast support. The router is implemented by AMS 0.35um technology library and the synthesizable netlist is easy to be implemented by other more advanced technologies. The 1-SL asynchronous router with 8-flit buffer is about 1800 equivalent gates (2-input nand gate) and 2-SL router is about 4300 equivalent gates. The multicast router is about 4700 gates, which only introduces a tiny cost to support multicast. It is mainly used for the multicast arbiter and GRLT (here only add two sets of register for each input port). But because the depth of FIFO is not confirmed, the FIFO area is not included here.

And the maximum data cycles of 1-SL router are about 15.8ns and 11.8ns respectively for header flit and other flits. And the cycle of unicast communication in the routers, which have disabled the multicast support, is about 14ns. If multicast support is enabled, the maximum data cycles of both unicast and multicast are about 16ns for header flit and 12ns for other flits. So the penalty of maximum data cycle performance for multicast support is very small. And the cycle of synchronous multi-pipeline router is about 12ns. While the asynchronous routers and synchronous routers exhibit similar performance, in a multi-clock domain application the synchronous type will introduce the additional penalty of multiple synchronization latencies and clock skew from clock tree distribution. Because there are few NoC routers supporting hardware multicast implementation, we only compare the software implementation with the proposed multicast router respectively with the tree-based broadcast setting and the path-based broadcast setting.

The broadcast latency and load examples are shown in Figure 4.11, in which all unicast traffics are ignored and 1-flit, 2-flit and 4-flit packets are broadcasted. Because the broadcast source can use the default broadcast GRLT setting, the setup delay also can be ignored. The latency is defined as the time from a host (such as in Figure 4.2) sending

the whole packet to the destination node, in the units of processing element cycles (16ns). Also assume that software broadcast sends the nearest destination last. The network load is the average number of existing flits per cycle in the network.

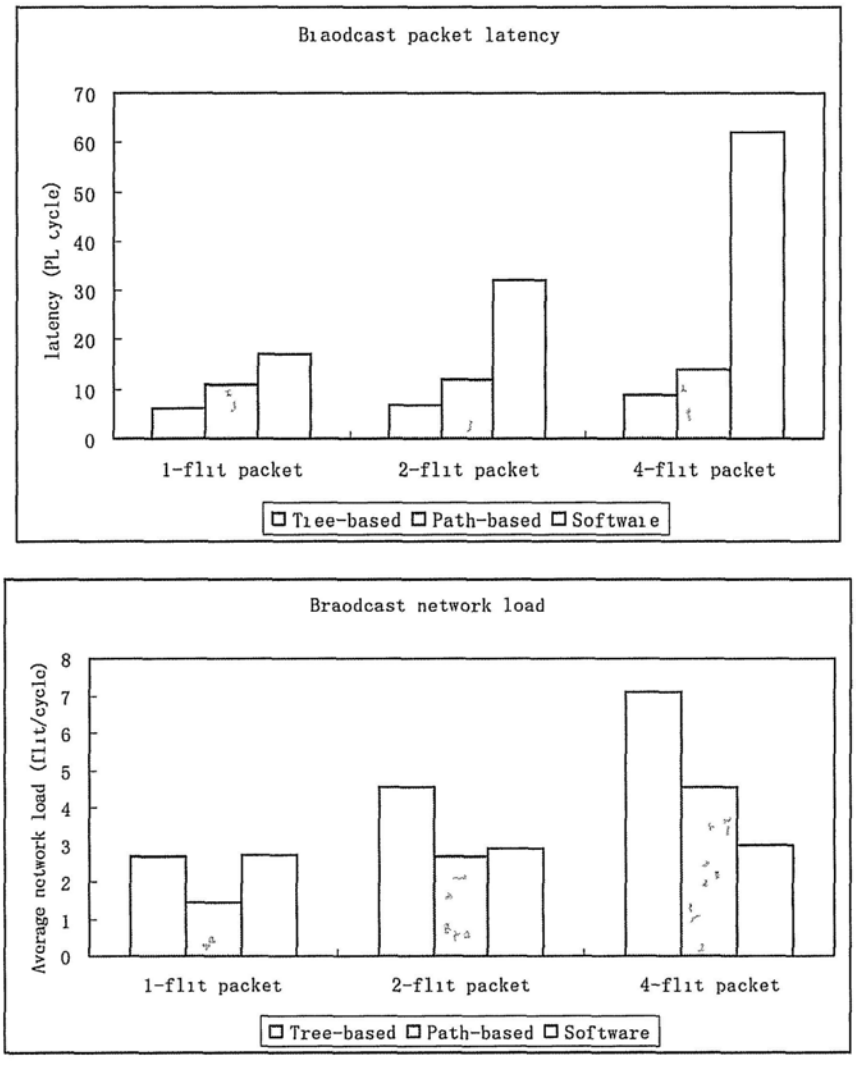


Figure 4.11 The latency and network load of broadcast example.

The hardware multicast scheme can significantly improve the broadcast latency with tiny extra logic for the routing block and multicast control. Whatever type broadcast (tree-based or path-based) the network implements in router's GRLT, the increment of performance is significant. In addition, the tree-based scheme is better than the path-based scheme.

Especially with increasing packet size, the hardware multicast's advantage in the

latency is obvious over the other implementation. But one must note the impact on the network load since the packet can be broadcasted in such short time. The tree-based broadcast algorithm occupies more network resources than the path-based algorithm. The increasing network blocks unicast packets to increase unicast latency. It is the trade-off between the performance and the increasing network load. The multicasting GRRS setting is up to the analysis of traffic pattern for real application.

Table 4.1 The average latency of various multicasts (PE cycle)

	Packet type	Software	Tree-based	Path-based
Broadcast mode1	Unicast	11.98	12.03	11.99
	Broadcast	62	9	14
	Total	12.29	12.011	12.004
Broadcast mode2	Unicast	12.98	12.43	12.39
	Broadcast	62.69	9.87	14.57
	Total	13.29	12.38	12.45

In broadcast mode1, only 10% of the packets of PE5 are broadcast packets in software implementation. In mode2, 10% of the packets of PE5 and PE10 are broadcast packets in software implementation. The injection rate of packets is 0.2 flit/cycle*node. The unicast packets adopt a uniform pattern.

It is noted that increasing broadcast packets will increase the total network load, although the average injection rate stays the same. Table 4.1 shows the simulation results. The average packet latency of software implementation is largest because its broadcast latency is much larger than the hardware implementations'.

4.3 Summary

We have presented an innovative group-based multicasting scheme for a wormhole switching Network on Chip. The protocol will be modified for multicast. The router supports different multicast modes, which can be defined to build an arbitrary multicast group. The router also can be switched into a traditional QoS router if the NoC has no

multicast mode traffic temporarily to improve the unicast traffic.

The improvement of broadcast performance is significant. And QoS is employed, so the multicast traffic does not show an obvious impact on the performance of unicast traffic while the network multicasting load is low.

CHAPTER 5. Lookahead Bypass

5.1 Bypass scheme

Bypass schemes are very useful to improve the network latency in a NoC by bypassing some steps of delivering a message. It has been introduced in chapter 2. Different bypass schemes have different advantages and disadvantages.

Bypassing some actions of a buffer is a good idea because of its simple architecture of implementation and small overhead. Many researchers proposed their improvements with buffer bypass.

To bypass other steps and components, such as switch allocator or the whole router, the router must be in charge of all computation of determination, and require to process and transmit more information from neighbors. Then the NoC router becomes a more complex design and has worse scalability.

In this section, a type of no-load bypass scheme is introduced in detail [58, 59, 61]. Other special bypass schemes, which are especially designed for some given application environments and not very general, so they are not discussed here.

If a flit can complete in an arbiter at the first pipeline stage in a multi-pipeline router, the router's propagation delay can be decreased. Then the flit is rapidly delivered and is removed to drain the buffer. The no-load bypass scheme simplifies the buffer write stage. It is useful to reduce average packet latency and to reduce the total buffering requirement.

The bypass scheme depends on lookahead routing to obtain a request quickly [57]. If a NoC router provides advanced routing results (requests to output port) for downstream routers, the downstream routers can execute switch allocation and virtual channel allocation at once according to the advanced requests. It means that the router removes

route computation from the critical path by calculating packet routes one hop in advance.

While a NoC adopts static routing, providing advanced requests can be easily achieved based on the routing information embedded in packets. Especially in a worm-hole mesh NoC, the routing information, which is generally the format of the destination address embedded in head flit, can be quickly used to generate the routing request according to the router's position in the network or its extended routing table.

Not only lookahead routing but also vacant input buffer is a necessary condition of success of bypass. If the buffer has stored some flits blocked by other flits, the incoming flit cannot leap over these flits. It needs to be written into the buffer and go through the regular pipeline.

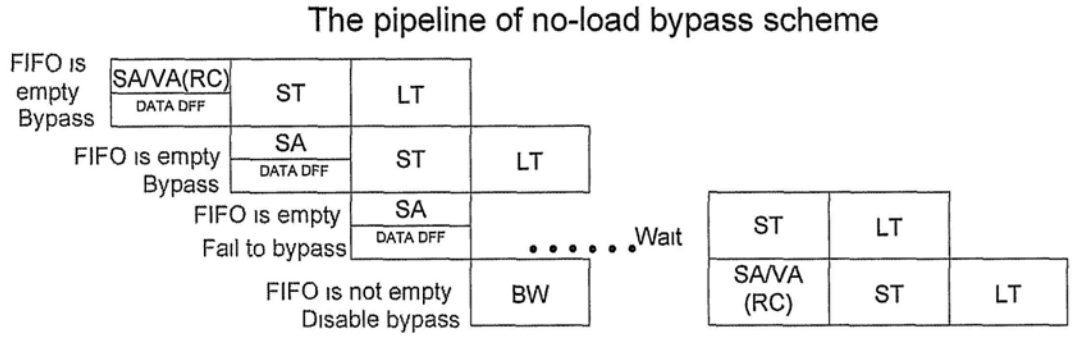


Figure 5.1 The pipeline of a no-load bypass router

The pipeline of a no-load bypass router is shown in Figure 5.1, which shows two cases of bypass success and bypass failure. In the figure, the input buffer is no-load (vacant) during the first two cycles. The request of an incoming head flit bypasses the buffer write stage and is submitted directly to virtual channel allocator (VA) and switch allocator (SA). In the same cycle, the routing information in the head flit is used to compute the routing request for a downstream router.

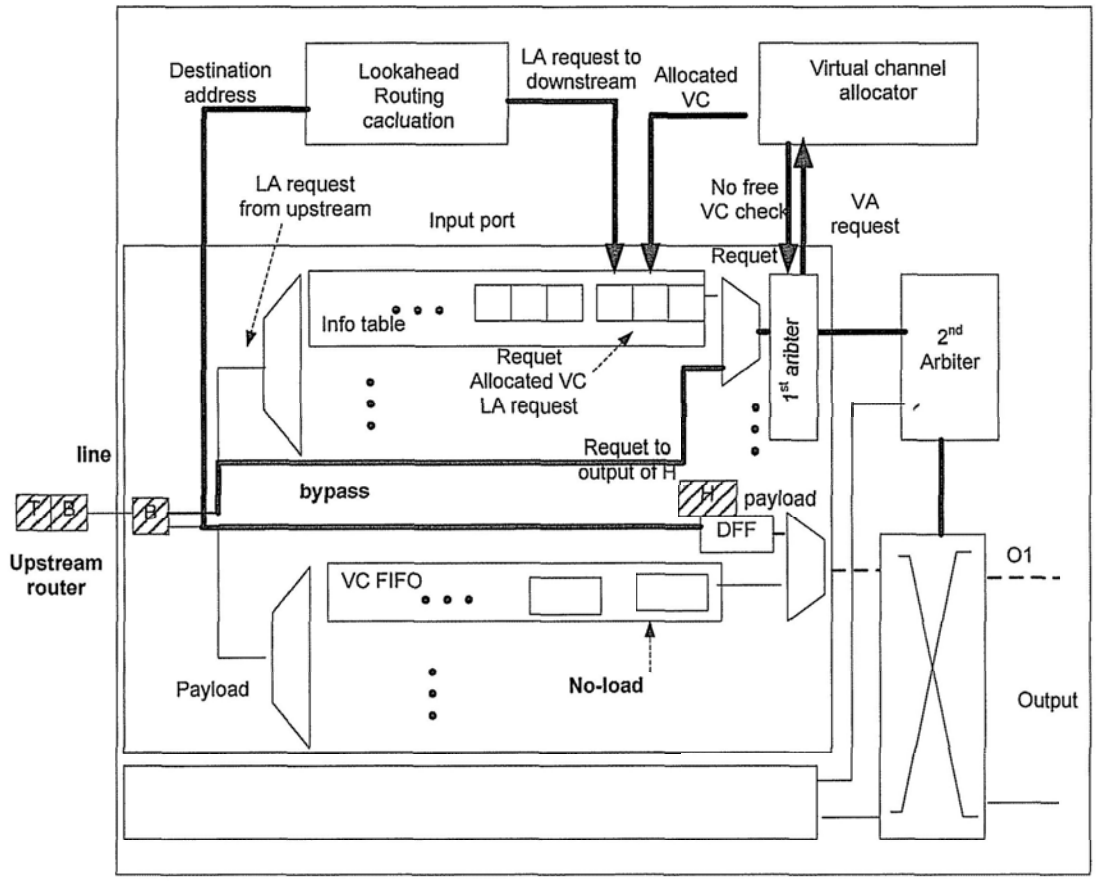


Figure 5.2 Architecture of no-load bypass router

The architecture of the no-load bypass router is shown in Figure 5.2. The bold line is associated with the stage of SA/VA(RC). And the dashed line is associated with the stage ST. The lookahead request of the head flit (H) from the upstream router can contest with other requests in the arbiter when the VC FIFO is no-load.

"DATA DFF" is a substitution of a new data-cache stage for an old buffer-write stage while the input buffer is no-load. It is implemented by the additional register and bypass path to the register. And it can make use of the advantage of no-load bypass to reduce the power consumption of writing and reading the large memory of the input buffer.

If VA allocates a new virtual channel and SA reserves the crossbar switch for the head flit, the cached flit traverses the crossbar switch in next cycle, followed by link traversal (LT). The following body or tail flit can also request the SA at once while it

arrives. And the arbitrating result of switch allocator determines whether the flit can be transferred to the output port.

If SA or VA of the current flit fails, such as the third flit in Figure 5.1, the bypass pipeline fails and the input buffer is not no-load again for the next flits. The following flits are written into the buffer (FIFO) and wait to go SA/VA until the stored flits are emptied and the input buffer becomes no-load again.

The no-load bypass can save latency and the energy associated with the buffer memory if bypass succeeds. Resource contention has a significant effect on the average packet latency of the networks because the no-load bypass scheme is effective only when the downstream router has no load in its input port buffer. As traffic increases, the input port memory is more likely to be occupied. The probability of bypass to occur will be severely reduced.

The optimized router, adopting the no-load bypass scheme, is general and is simply implemented before the application traffic is specific. It allows the NoC to have good scalability and compatibility. The lookahead bypass scheme will be introduced in the next section. The proposed lookahead bypass router is a general design, too.

5.2 Lookahead bypass router

It is known that lookahead routing enables processing SA immediately and bypassing one pipeline-stage. What can be improved in NoC if there are more lookahead signals?

5.2.1 Interconnection signals

At first, we analyze all connection signals between two routers. The interconnection wires between two NoC routers to deliver flits can be divided into two types: **control signals** and **payload data**. These wires are in charge of delivering the message flit

between two routers.

The payload data is the major part of a packet, shown in Figure 5.3. It takes the original message from the source element, which is only used by the destination element. The payload should not be involved in the computation of routers of routing path. In wormhole routers, the payload data is broken up into several pieces in flits.

The control signals are all real-time signals that are required by the computation (such as routing, allocating) in routers. The control signals include virtual channel information, head/tail flit information, route information, flow-control information, and so on. It is noted that the route information can only be required once by each packet. It usually exists in the head flit. The popular format of route information is the address of the message's destination in a NoC network. In some cases, the route information includes the lookahead routing result from the upstream router, which can remove the RC pipeline stage of a head flit.

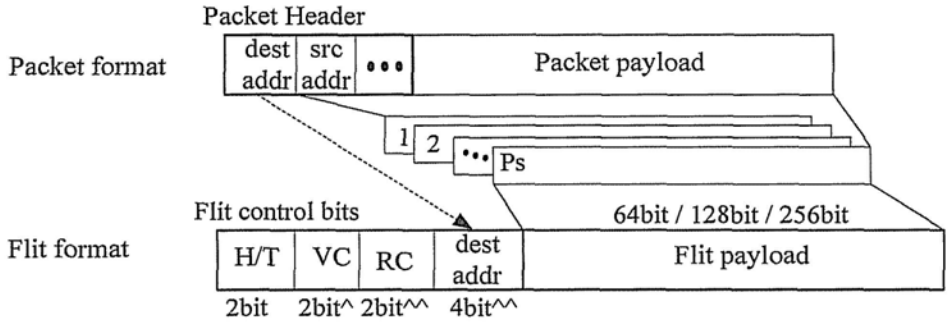
In NoCs, the width of control signals is usually much less than the number of signal lines for the payload data. The width of payload in a flit may be 64bit or even more bits and the control signals are 11bits in a 4-VC 4x4 mesh network, shown in the table 5.1. Therefore, the flit format is different in a lookahead bypass router, shown in Figure 5.3. The flit valid bit is not a part of the flit, which is used for control. More power and latency are required to transmit the payload data than to transmit the control signals.

Table 5.1 An example of interconnection in a 4-VC 4x4 mesh network

Payload data	Control signals					Credit
64/128 bits (or more)	5 bits + 2its* + 4bts**					4 bits
	Valid	Head/Tail	VC	LA Route info.	Address**	
	1 bit	2bit	2 bits	2bits*	4 bits	

* If the router support lookahead route, there are 2-bits wires for lookahead routing result

** In many routers, address information is delivered by the wires for the payload data



^For a 4-virtual-channel router, there is 2-bit vc information
^^For a 4x4 network, there is 4-bit destination address and 2-bit RC information

Figure 5.3 The format of packet and flit in a lookahead bypass router

It is noted that the **buffer-management information** (such as credit or stop signal) is not a part of flits. It is the feedback signals of buffers' information, which is returned to the upstream unit separately. So the buffer-management flow-control information is different from other interconnection wires and needs to be processed specially.

5.2.2 Pipeline

It is known that payload data does not directly involve the computations of routers while the control signals are the required information for the computations of the routers. In fact, the payload data are held idle in the input buffer until free resources are allocated to accept them.

A router can complete all computations and allocate the resources for a flit only after it has obtained all the necessary control signals from its upstream router. Since the allocations must be ahead of the switch traversal of payload data in NoC routers, the control signals from the upstream nodes are actually requested earlier than the payload data. Similarly, if the processing of control signals can be finished early, the control signals are available to traverse the link and reach the downstream node earlier.

As soon as the upstream node's lookahead control signals arrive, the router can execute immediately its allocation computation. This step is just like the process of the

no-load bypass scheme. And in the same cycle, the generation and traversal of all control signals for the downstream router is also parallelized to execute. The lookahead controlling traversal (LCT) delivers the allocated VC, routing results, destination address (only for head flits), and head/tail information in advance.

Figure 5.4 shows the pipeline of lookahead bypass. In the figure, the flits are transferred from "router ***i-1***" to "router ***i***". Each flit is divided into two parts: control signals (ctrl part) and payload data (data part). The two parts are not delivered in the same cycle.

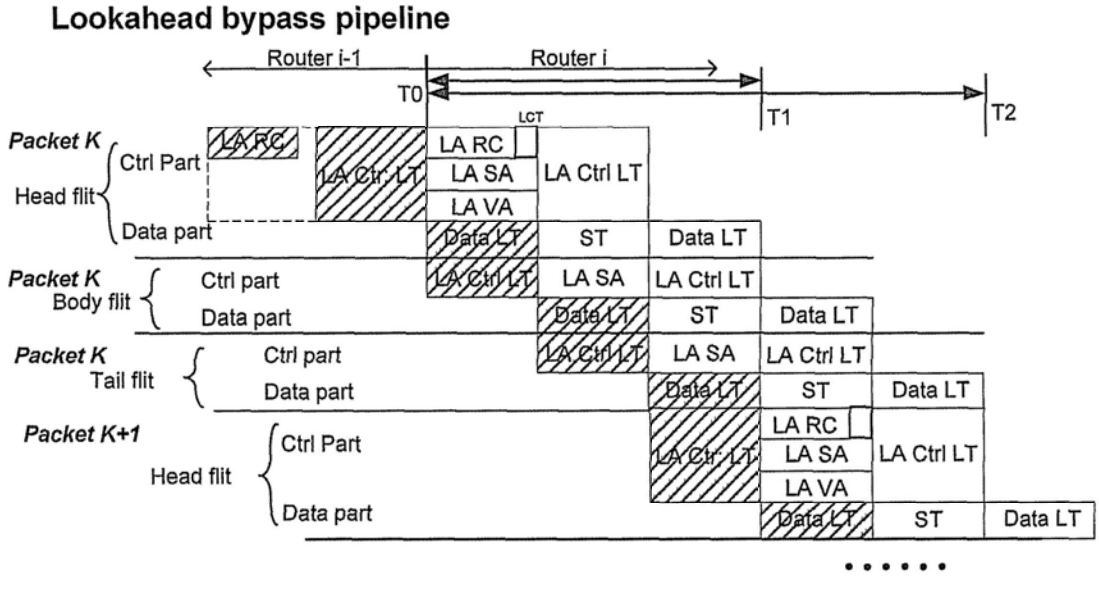


Figure 5.4 Lookahead bypass pipeline

The "router ***i***" can execute immediately switch allocation and virtual channel allocation for the head flit when it receives control signals. Because the flit's payload is still at the line transversal stage at the same time, two allocations are referred to as lookahead switch allocation (LA SA) and lookahead virtual channel allocation (LA VA). LA SA and LA VA are one cycle earlier than when the flit's major payload part arrives. The lookahead routing computation (LA RC) for downstream router (router i+1) is also completed according to the destination address which is the part of control signals in this cycle.

Then a flit's lookahead routing result, destination address, and head/tail bits traverse the router from one input port to another output port according to the arbitration of the switch allocation. And the lookahead control signals transversal (LCT) is speculative because the flit may fail to obtain the necessary resource. If the flit succeeds, the allocated VC and other control signals from LCT are ready to be transmitted ahead at the "lookahead control signals line transversal" (LA ctrl LT) stage of the next cycle.

The process of body flit or tail flit is similar to the process of head flit. But the body and tail flits can utilize the head flit's VA and RC results. So they don't need virtual channel allocation and routing computations. Of course, there are no destination address and routing result to be transmitted.

A flit's control signals cost only one cycle to traverse a router if the flit succeeds to obtain necessary resource to implement lookahead bypass. When a flit succeeds lookahead bypass, the flit can skip to access the FIFO completely. The lookahead bypass reduces the propagation latency and reduces the requirement of a large buffer. Maximum benefit can be gained if sequential bypass opportunity could be exploited all, such as in the example in Figure 5.4.

Based on Equation 2.4, the cycle of minimal network latency becomes $T_{network} = 2 \times D + (L/W)$. If the overhead of the critical path is small enough to implement lookahead bypass, the lookahead bypass can remarkably improve the average packet latency in a NoC.

In the no-load bypass scheme, the router only can enable bypass when the input buffer (FIFOs) is vacant (on-load). And the lookahead router can still enable lookahead bypass if the input buffer has stored the data because all incoming flits need be checked by "LA processing". Only when there is no free virtual channel or the proceeding flits of the packet have been stored in FIFOs, the request is masked to

contest the resource and the control signals are also stored in FIFOs. Otherwise, the incoming request attempts the lookahead bypass.

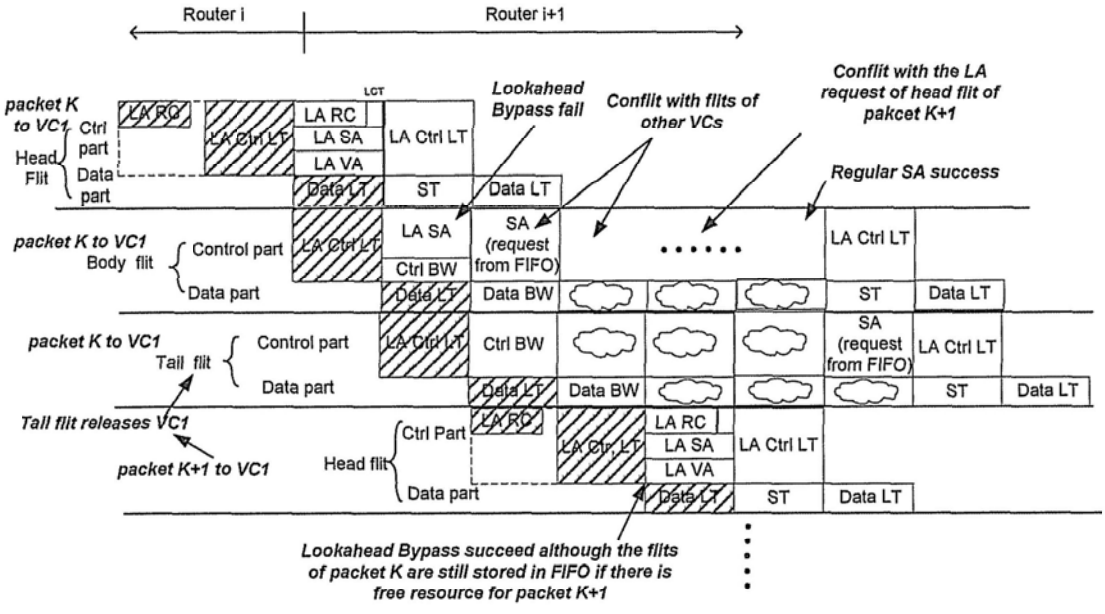


Figure 5.5 Pipeline example of Lookahead bypass if FIFOs are not no-load

Figure 5.5 shows an example when FIFOs are not vacant. Some flits of packet K have been stored in FIFOs of VC1 because there is a conflict with other packets. Packet K+1 also uses VC1 and the head flit is delivered after the tail flit of packet K. The new request goes through the LA processing module and succeeds to obtain the crossbar's input and output resources in the arbiters. The flit of a new packet can jump the remaining flits of packet K and implements lookahead bypass.

The wormhole flow control requires keeping the sequence of flits of a packet. All flits of a packet must be delivered in order. If one flit of a packet fails lookahead bypass, the requests of remaining flits should been masked by "LA failed marks" and the flits are stored into FIFOs.

But if all flits which are stored in FIFOs have been exhausted, the new flit can attempt lookahead bypass again. In the controller of the lookahead bypass router, there is the

logic to implement this re-enter bypass mechanism. The logic checks whether the stored flits are all delivered to downstream router. If there is nothing remaining in FIFOs, the new request is not stored in FIFOs but goes through the lookahead processing module. And some controlling information, such as routing result and virtual channel for the downstream router, is still submitted by the normal module.

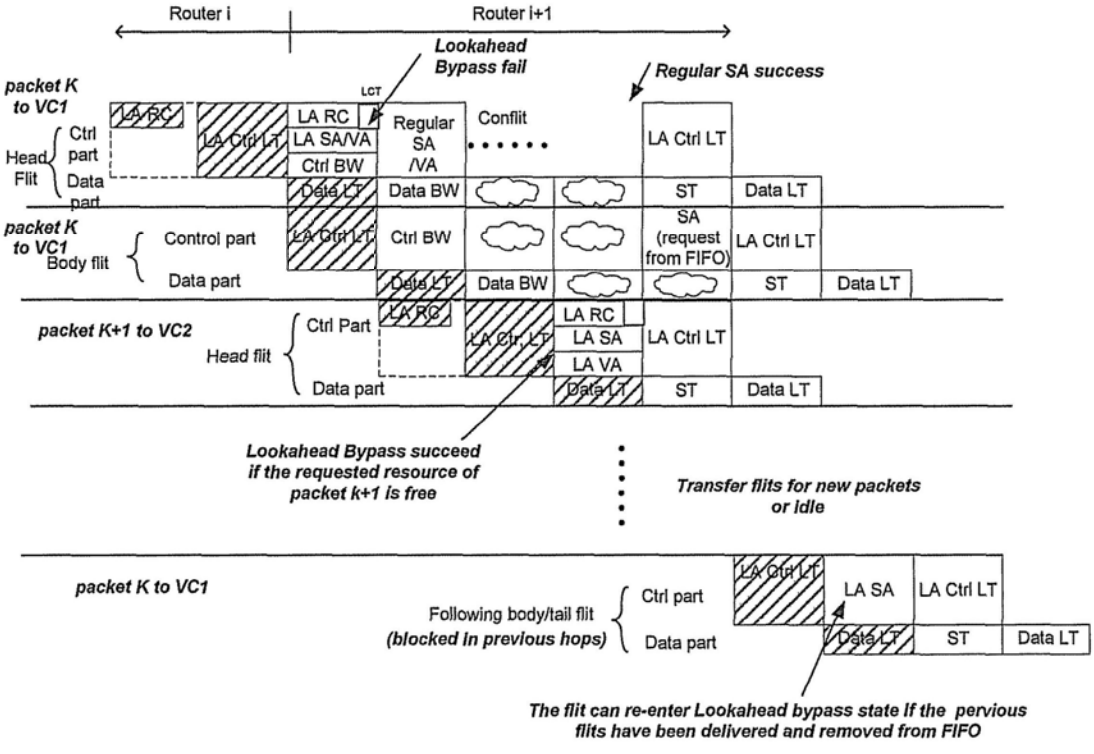


Figure 5.6 Re-enter lookahead bypass state

Figure 5.6 shows an example in which a packet (packet K) re-enters the bypass state. The first two flits of "Packet K" are delivered from "Router i:" to "Router i+1" and fail lookahead bypass. The third flit is blocked and other packets are transferred to Router i+1. Before the third flit arrives, the first two flits succeed to obtain the resource in the switch allocator through the normal request path and are delivered to the downstream router. Then, the third flit of packet K can attempt lookahead bypass to re-enter the lookahead bypass state.

5.2.3 Architecture

Lookahead bypass routers need to handle two types of flits: lookahead flits and regular (non-lookahead) flits. To implement the lookahead bypass, a router needs to introduce the additional lookahead logic to process and transport the lookahead control signals. To impair the critical path as little as possible, most of the introduced lookahead modules should be in parallel with the original processing modules. Then the overhead of the implementation of lookahead bypass could be acceptable. Figure 5.7 is the block diagram of lookahead bypass router.

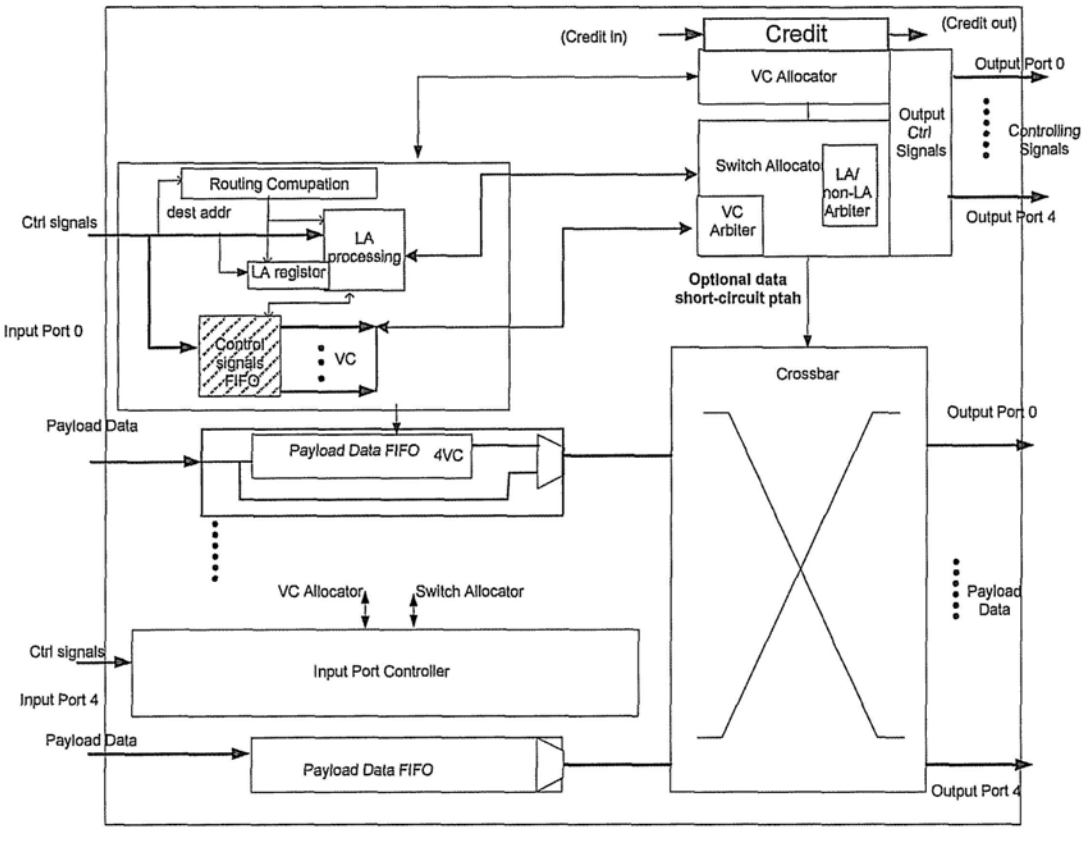


Figure 5.7 Lookahead bypass router

There are two types of modules in a router, namely normal modules and lookahead modules. The normal modules have the same function as the modules in speculative-pipeline routers. They include input buffers, routing computation modules, a crossbar

switch, and virtual channel allocator. Of course, they need to be modified to meet the requirements of a lookahead router. The lookahead modules, which are the gray blocks in the figure, are introduced to process lookahead request. All modules of lookahead logic include lookahead-control-signal registers, lookahead request processors, lookahead/non-lookahead arbiters, and lookahead-control-signal multiplexers (substitution of a part of switch).

- Routing computation modules, virtual channel allocator, and crossbar

In lookahead bypass routers, the route information includes the destination address and lookahead routing result. Only head flits have these control signals of route information. So the connection lines of route information are idle if the current flit is a body or tail flit. The routing computation module operates and transmits the lookahead routing result to the downstream router only for head flits.

The virtual channel allocator is similar to the speculative-pipeline router's. It only provides free virtual channels of downstream routers. The flit which obtains the switch resource consumes a free virtual channel. And the virtual channel allocator will provide another free virtual channel.

The crossbar switch in a lookahead router only delivers the payload part of flits. The results of switch allocators determine which flits from input ports can be transferred to the output ports. The other part of flits (control signals) is delivered by the special multiplexers ahead of the payload part.

- Input buffers and lookahead-control-signal registers

If a head flit succeeds to apply a vacant virtual channel and resources of switch to

lookahead bypass, its control signals from upstream router like head/tail bits and control information from local router like downstream router's request and allocated VC are stored in the lookahead control register (LA register) of the current VC. The following flits (body flit or tail flit) can use the information to submit the request to the arbiters of the switch allocator, which all use a matrix arbiter architecture.

The input buffers are divided into two parts. Payload data FIFO stores the payload part of a flit. Control signals FIFO stores control signals and control information of flits which fail lookahead bypass. These flits are referred to as regular flits because they do not bypass and submit their requests from FIFO, similar to regular routers.

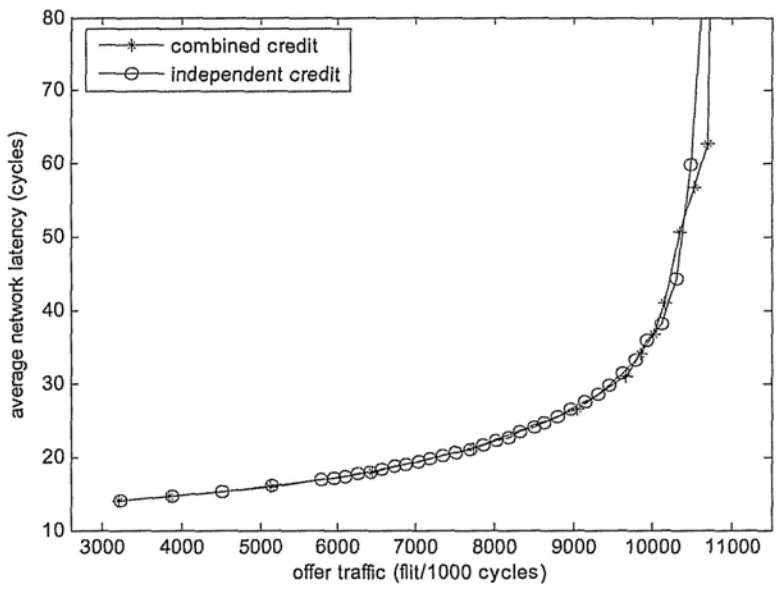
- Credit-based flow control

The lookahead bypass router's flit-level flow control applies the credit flow control protocol. The credit information is a special type of interconnection signals between routers. It is independent of the message flit and does not need to advance to be delivered. In general, the number of virtual channels determines the width of the credit wires.

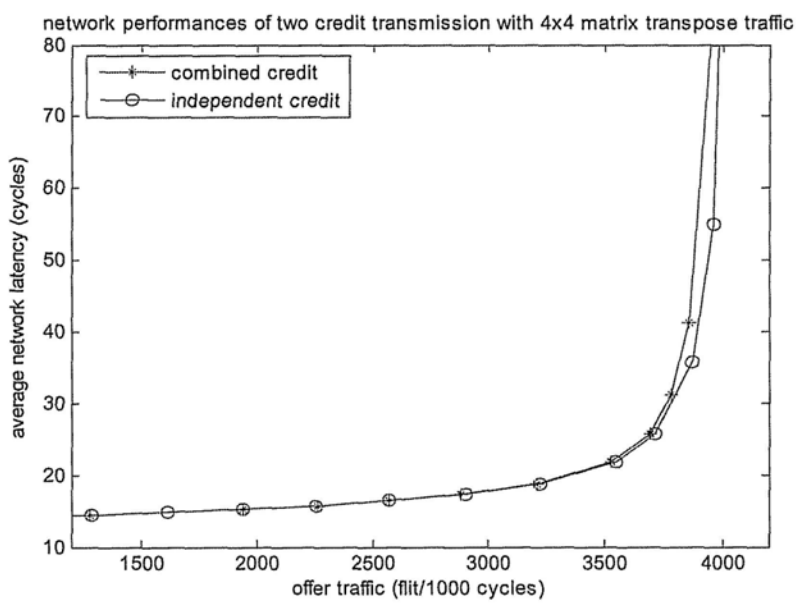
If the wire layout resource is limited, combined credit connection is an approach. The lookahead bypass router needs to increase wires to transfer the destination address independently. Because only head flit has the destination address, these wires can be utilized to deliver both credit and route information (destination address and lookahead routing result).

Figure 5.8 shows the comparison between two modes of 4x4 networks with combined credit mode or independent credit mode. The curves of combined credit and of

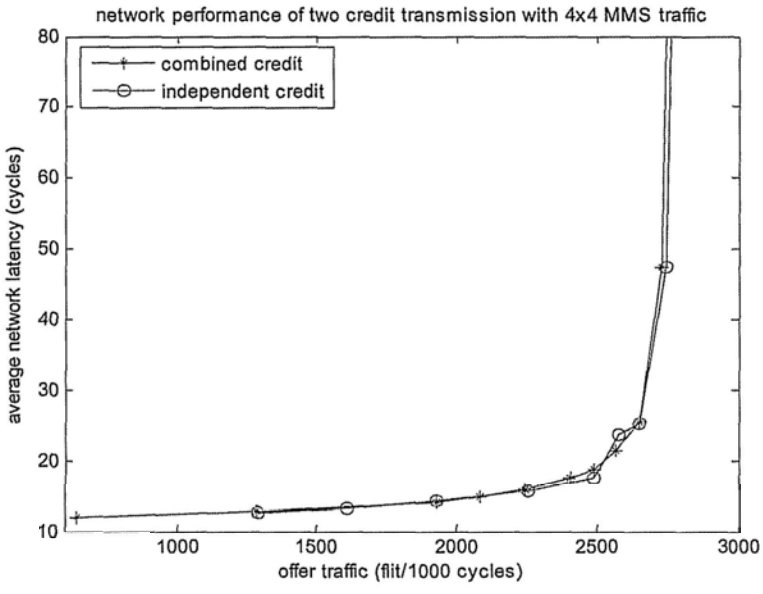
independent credit nearly overlap whatever the traffic pattern is. It means that the combined credit approach impairs the performance very little because there is enough credit (eight flits) in the original buffer to wait the credit information back and the head flit is only occupied in 1/4 of all flits.



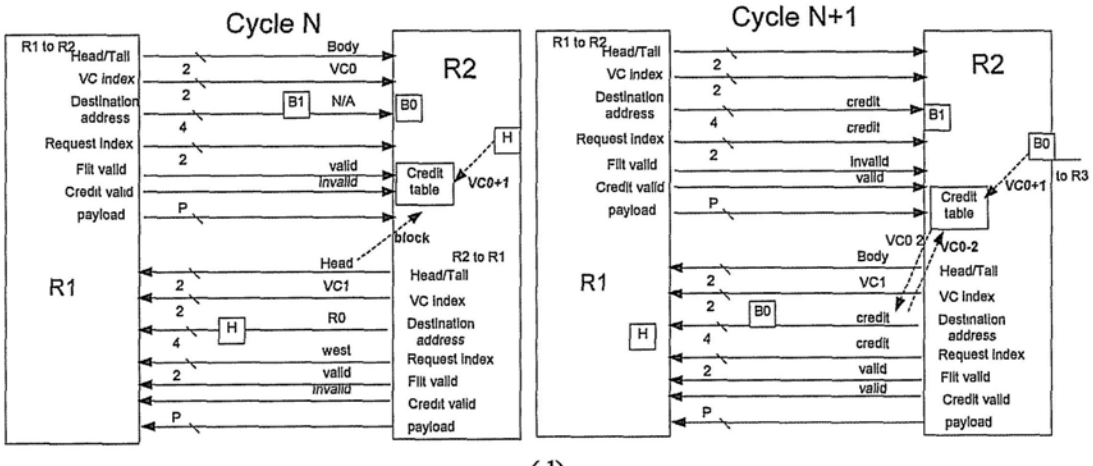
(a)



(b)



(c)



(d)

Figure 5.8 The comparison between two modes of credit connection when network size is 4x4, VC is 4, buffer size is 8 and traffic pattern is (a) uniform, (b) matrix transpose, (c) mms trace. (d) is an example of combined credit mode

➢ Switch allocator and lookahead-control-signal multiplexers

Because the flits of all types need to send lookahead control signals to downstream router ahead of the payload data, the traversal of lookahead control signals to the downstream router should be integrated into the pipeline stage of switch allocation whichever type the flit is.

To speed up the traversal of control signals, two-stage speculative control signal multiplexers are used in the switch allocator. If a flit wins the first-stage arbiter which arbitrates the requests from the same input port, its control signals will be transferred by the first-stage multiplexer at once. Because the first-stage multiplexers do not wait for the results of second-stage arbiters, its control signals output can be blocked by the requests from other input ports or lookahead flits.

Because the payload of lookahead flits go through the crossbar switch, the lookahead requests also need to contest the input resource of the crossbar. There are two implementations of lookahead controller with different lookahead/ non-lookahead arbiters, which are shown in Figure 5.9 and Figure 5.10.

The first one is shown in Figure 5.9. It arbitrates lookahead requests and the winner of the requests that are from the same orientation's control signals FIFO. And the lookahead request has higher priority than the non-lookahead (normal) requests. Then the arbitration result contests the resource of the output port with the results from other orientations. It is noted that the transmission of control signals is executed based on the arbitration result as soon as they are obtained.

The second one in Figure 5.10 is that all lookahead requests are submitted to the required output port to contest the resource with other lookahead requests of different input ports. And the two-stage arbiter is still used for normal (non-lookahead) requests. Then the winner of lookahead requests arbitrators with the winner of normal requests and the lookahead requests have higher priority than the normal requests.

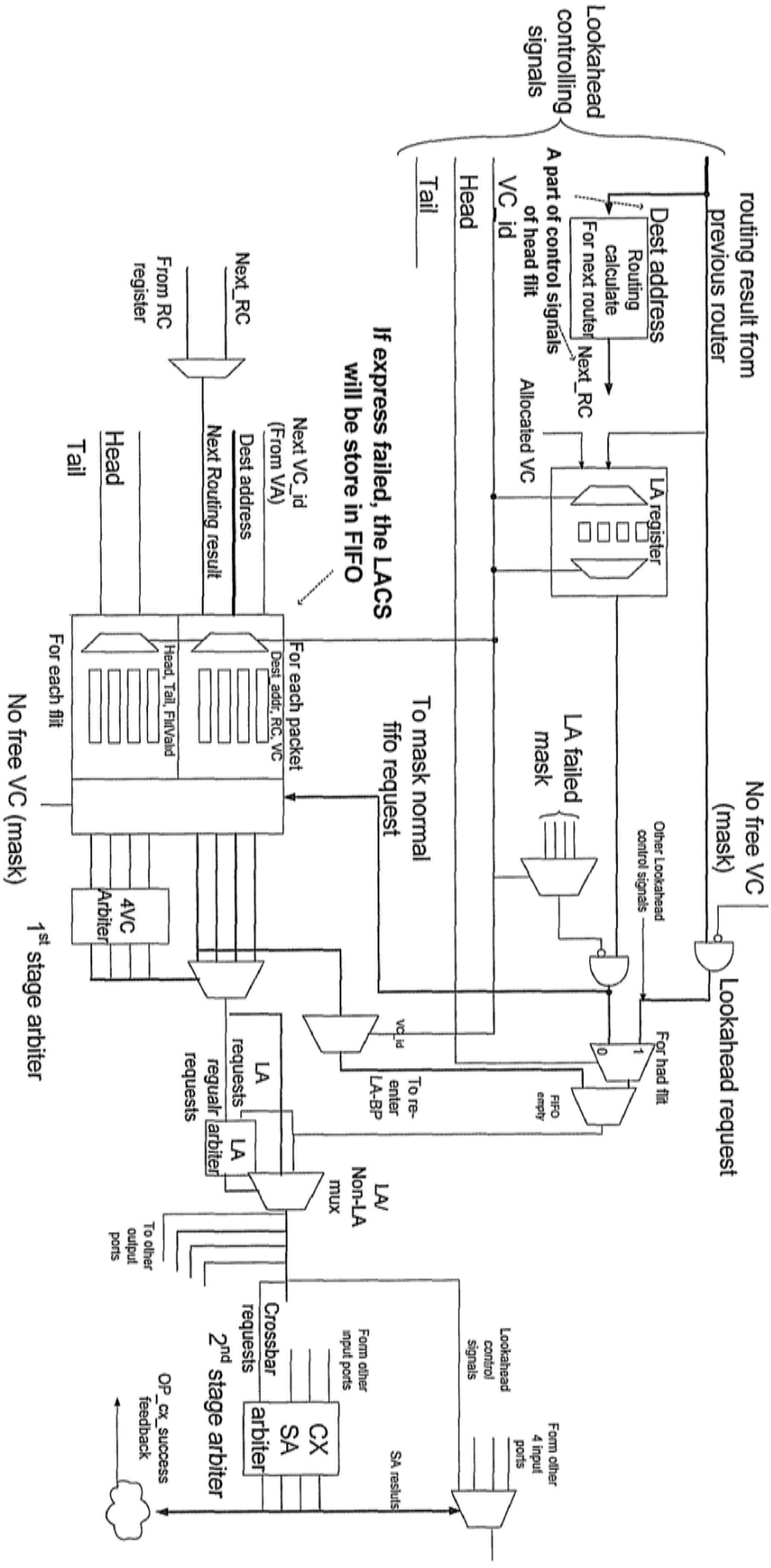


Figure 5.9 The first controller of lookahead bypass router

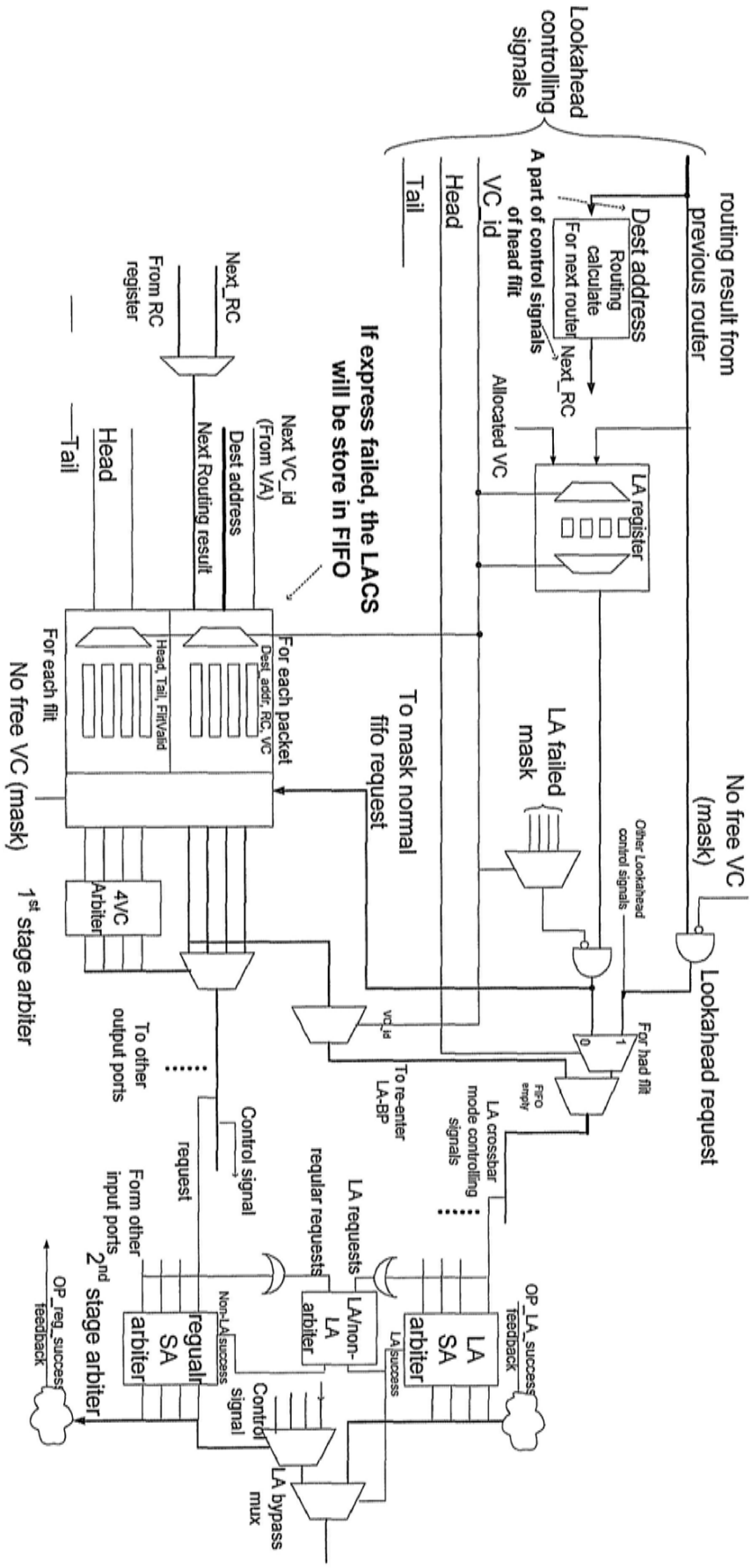


Figure 5.10 The second controller of lookahead bypass router

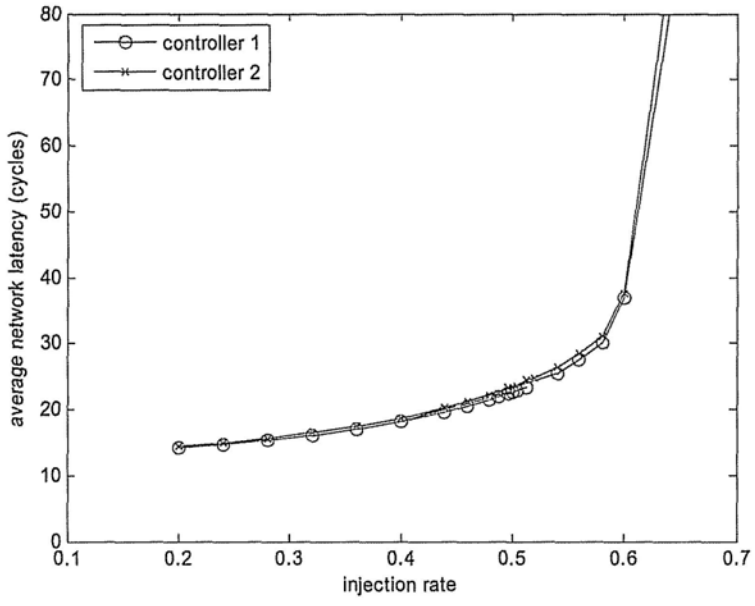


Figure 5.11 The comparison between two types of switch allocators when network size is 4x4, VC is 4, buffer size is 8, and traffic pattern is uniform

The two switch allocators can both implement the controller of the lookahead router. The latency comparison between the two switch allocators is shown in Figure 5.11. It is found that the results are very close. There is only 1.98% (0.28 cycle) difference between the two modes under 0.2 flit/cycle*node network load and 1.78% (0.67 cycle) difference under 0.6 flit/cycle*node load. The first one slightly outperforms the second one.

5.3 Implementation

The critical path of a NoC router is in the switch allocation/virtual channel allocation pipeline stage. Figure 5.12 (a) shows the critical path for generic SA/VA in speculative-pipeline routers [60], which is regarded as the baseline router. The lower line, allocating a free VC, is available for head flits. Figure 5.12 (b) shows the critical path in lookahead bypass routers. The lower line is also available for head flits. The

bottom line that processes lookahead requests is in parallel with the middle line that processes the non-lookahead requests. As compared with the baseline, the lookahead bypass router increases several gates for additional state control and LA/non-LA arbitration that is a 2-to-1 aribter in the step of 2nd stage SA request generation.

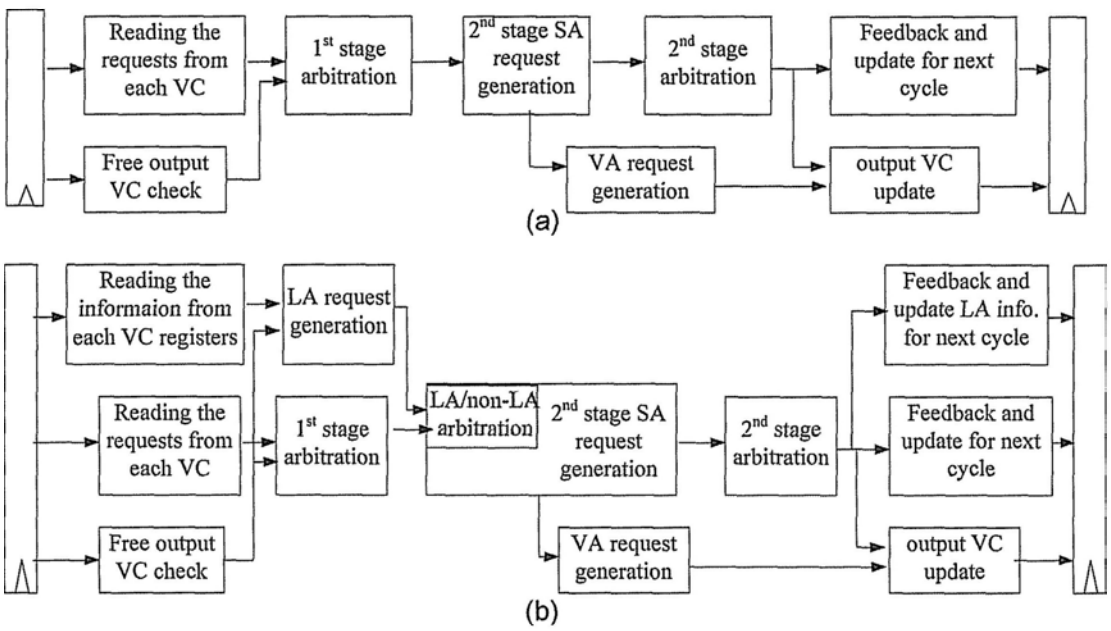


Figure 5.12 Critical path of SA/VA pipeline stage in (a) baseline, (b) lookahead bypss router

In the worst case, the critical path of a lookahead bypass router only adds near 0.2ns in delay (the gray block in Figure 5.12) with UMC 130nm technology library. This is negligibly small in comparison with the clock period of a NoC router. The lookahead bypass router can achieve the target of 250MHz clock frequency.

Table 5.2 The maximum frequency and area of various routers

	Baseline max frequency	LA max frequency
P=4,V=4,F=4	357MHz	347MHz
P=5,V=3,F=4;	359MHz	344MHz
P=5,V=4,F=4;	315MHz	307MHz
P=5,V=4,F=8;	310MHz	303MHz
P=5,V=8,F=8	260MHz	260MHz

Table 5.2 shows the maximum operation frequency, which includes 0.6ns clock uncertain time and library setup time. Of course, the maximum frequency is rarely adopted because network frequency is not only determined by router's maximum operation frequency. In a real application, the communication bandwidth requirement and the working frequency of processing elements is the major determinant.

- The silicon area evaluation

The lookahead bypass controlling logic is larger than the baseline controlling logic because the lookahead bypass scheme requires additional logic to allocate lookahead bypass flits. The port number, the virtual channel number and buffer size per channel all define the silicon area of the payload FIFO. However, the buffer size per channel can only influence the size of the controlling signals FIFO in a controller, which is only a small part of a router's controller. And it is irrelevant to the area of switch allocator and virtual channel allocator.

The buffers take up most of the silicon area of a router. If the router uses 4 VCs per port, 8 flits per VC, 128 bits payload per flit, the total payload buffer size is about 20k bits. The buffers take up most of the silicon area of a router if a router uses a large buffer for each virtual channel. The lookahead bypass improvement increases acceptable area. It is about equivalent to 5155 2-NAND gates (6.18%) overhead as compared with the baseline if the target frequency is 250MHz. And it includes the newly added FIFO of destination address, which occupied the payload FIFO in the baseline router. Other configurations are listed in Table 5.3.

Table 5.3 The silicon area of 128bit-payload routers (um^2) at maximum frequency

Configuration	Target Frequency	Payload FIFO + In/Out Payload Register	LA total	Baseline total
P=5, V=4, F=8	300MHz	~696000+60000	926228	846622
P=5, V=4, F=8	250MHz	~696000+60000	885392	833840
P=5, V=4, F=4	250MHz	~345000+60000	475224	426981
P=5, V=3, F=4	250MHz	~258000+60000	386224	347995

➢ The physical implementation and test board

The chip implements a router to verify the function of our lookahead bypass router with the UMC 0.13um technology, shown in Figure 5.13. It can switch its function among three types: the lookahead bypass router, the no-load bypass router and the speculation router, which is the rectangle part in the figure. Thus, we can quickly compare the three types of routers. The traffic generator logics are controlled by multiple configuration registers. The detailed traffic flow of five ports, which is summarized from the simulation platform according to one router in a NoC, tests the performance of the router. Then, traffic receiver logics take charge of counting the packets and calculating the average latency of each output port.

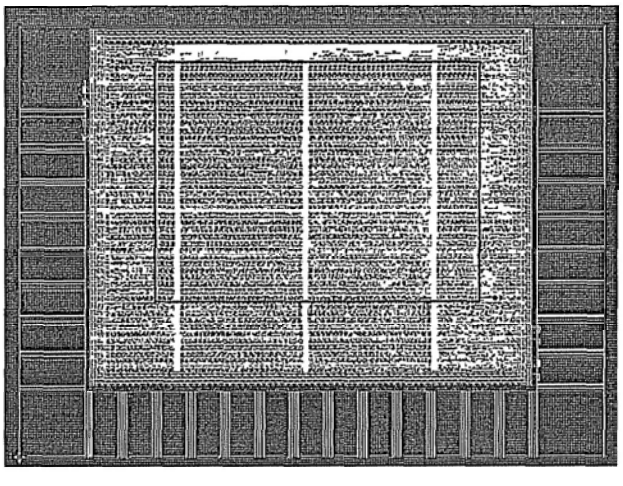


Figure 5.13 The layout of test chip

The test system uses Agilent 16702B logic analyzer to input the test bench into the configuration registers and read the result from the result registers that store the result of average latency calculation, shown in Figure 5.14.

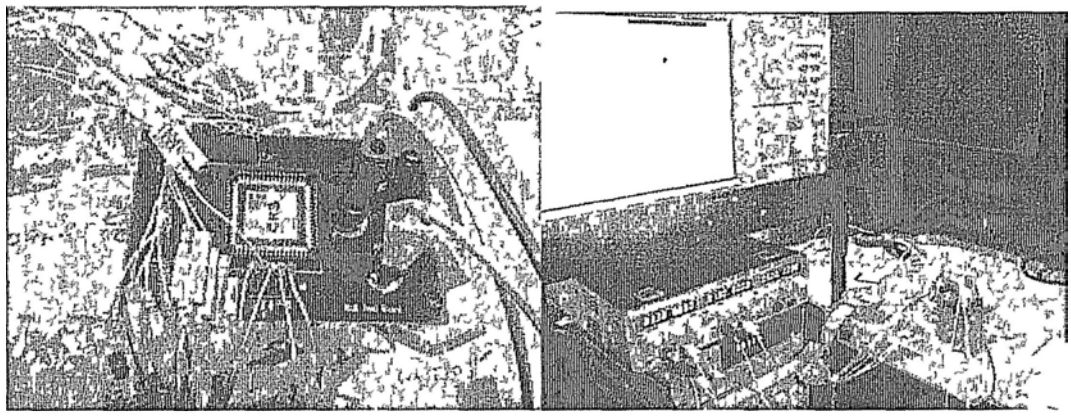


Figure 5.14 Test board and system

The test system verifies the function of our lookahead router. And the testing results of latency are same as the simulation results.

5.4 Deadlock

If a packet's first flits succeed and the succeeding flits fail bypass, the remaining flits are stored in FIFOs. The remaining flits may deadlock with the packets which have been stored in the VC's FIFOs if the old packets have not obtained the virtual-channel resource of the downstream router.

It means that the new packet occupied the virtual-channel resource. But it is also locked by the old packets in FIFOs. The old packets occupied the FIFOs' output resource but are locked because of no virtual-channel resource. Figure 5.15 shows an example of inter-lock between packet K and packet K+1. If all virtual channels of a downstream router's port are inter-locked like the example, the deadlock happens.

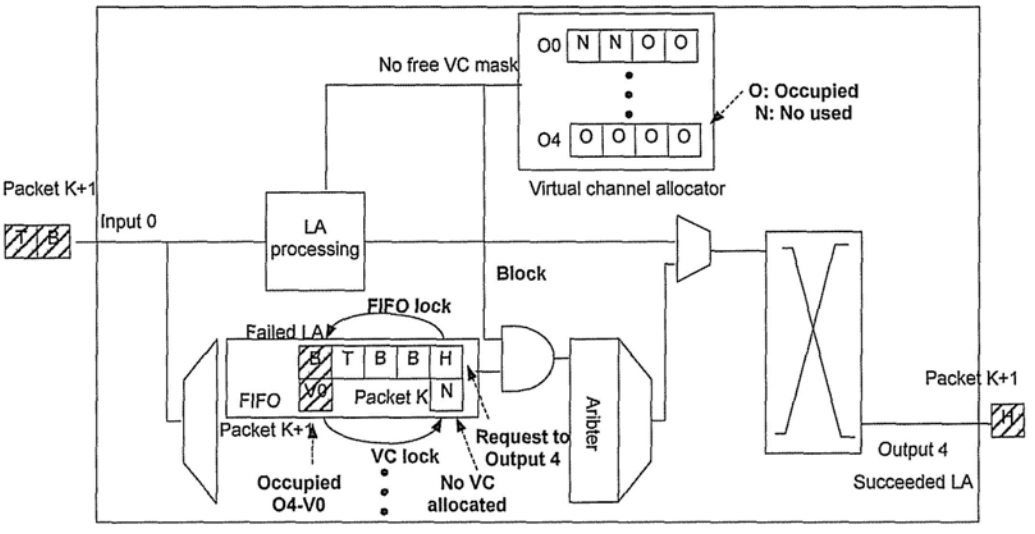


Figure 5.15 An example of inter-lock

There are three solutions. The first one is to store the remaining flits of the new packet in the first part of FIFOs. It releases the FIFOs' output resource which is occupied by old packets. But the architecture of such FIFOs is very complex.

The second one is the reserved virtual channel for non-lookahead (normal) requests. All lookahead requests can only use a part of virtual channel in downstream routers. And the normal requests can use all virtual channels. There is at least one reserved virtual channel for normal requests in an input port. Then, lookahead bypass flits can not exhaust the virtual channels and deadlock with the normal flits. It requires a more complex virtual channel allocator and wastes one channel for lookahead packets.

The last solution is to check the content of the input buffer. If there is still any head flit in some virtual channel's FIFO, the router will disable bypass for new packets of this virtual channel. In this case, the remaining flits of new packet can not deadlock with the flits which have been stored in the VC's FIFOs because these flits all have obtained the virtual channels of downstream routers. This last solution is the simplest

implementation. It only increases the AND gate to check all head bits in FIFOs. If there are valid head bits, the incoming packet disables an attempt to bypass.

5.5 Summary

The lookahead bypass router utilizes the lookahead controlling signals to compute and allocate a flit before its payload data part has been received. The lookahead allocation makes it possible that many flits can bypass the input buffer and are directly delivered to the required output port throughput a crossbar.

The new pipeline implements a small propagation delay of a flit in a lookahead bypass router. The lookahead allocation logic is in parallel with the regular allocation logic to implement the lookahead bypass router. Thus, the critical path only inserts a little logic, which has a small influence over the maximum operating frequency of a router. The overhead area of the controller is acceptable.

CHAPTER 6. Lookahead Bypass Evaluation

Network performances are obtained by the simulation platform modeled in System Verilog. Our NoC library includes various types of routers which are described in section 2.4. The speculative-pipeline router is referred as the baseline. The no-load bypass router is used to compare different bypass schemes. In addition, the EVC router that is improved from the baseline router and the conventional five-pipeline router are listed as references. The basic network parameters are listed in Table 6.1.

Table 6.1 Basic network parameters of evaluations

Topology	mesh network (4x4, 6x6, 8x8, 10x4)
Flow control	Virtual channel
Buffer management	Credit-based flow control
Routing algorithm	XY
Pipeline	Synchronous pipeline
Router radix	5 (four orientations and local)
Buffer architecture	Input buffer (V VCs per port, F flits per VC)
Packet length	P flits
Flits size	128 bits (payload) + control signals
Technology	UMC 0.13um technology library
Tile size	1mm x 1mm

The network performances of various routers in our NoC library are compared with various virtual-channel numbers, buffer sizes, traffic patterns, and network scales to validate and estimate our lookahead bypass scheme.

6.1 The influence of virtual channel and buffer

In this section, all traffics use a four-flit packet size in a 4x4 mesh network. The average packet latency is compared to estimate the influence of various

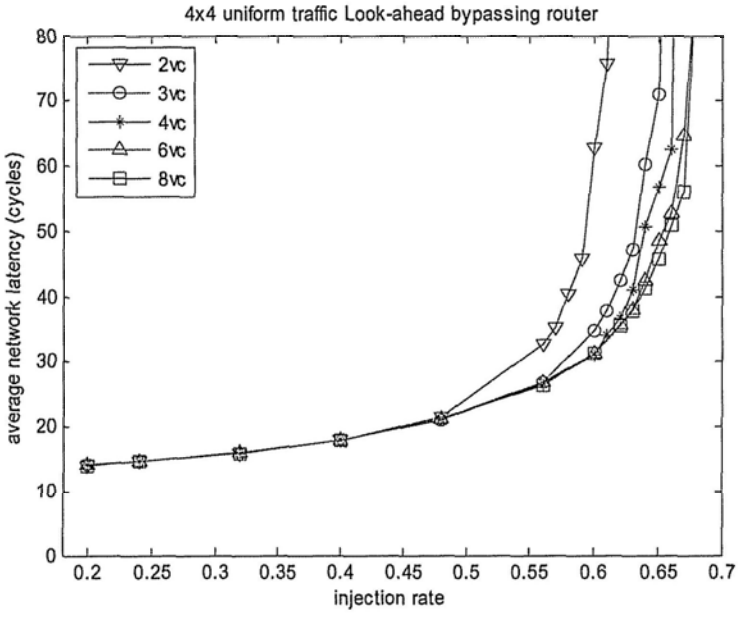
virtual-channel numbers, and buffer sizes.

6.1.1 Virtual channel

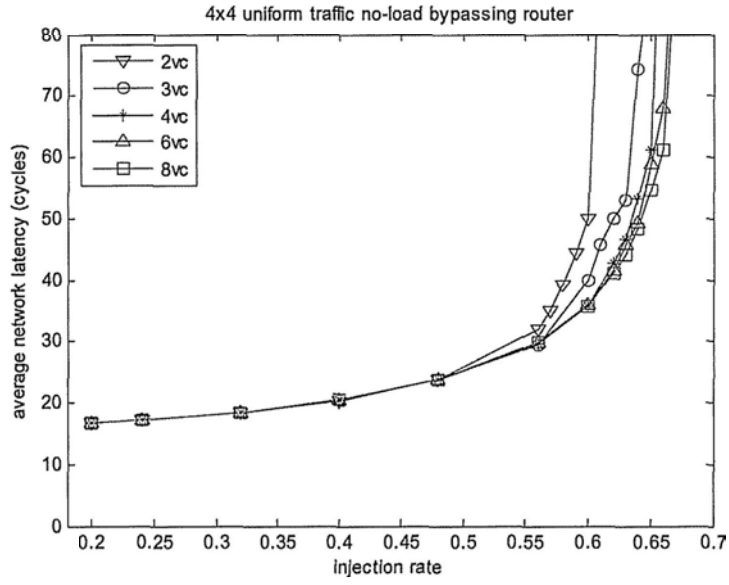
The virtual channels can improve the throughput and packet latency remarkably. For virtual channel flow control, each virtual channel requires its own buffer to cache flits. It is noted that channels' buffers cost the largest part of the silicon area, power consumption and have the greatest affect on the network performance.

Increasing the number of virtual channels can improve the performance. However, having too many virtual channel would cost a very large silicon area for each channel's buffer and require a very complex controller because switch allocator and virtual channels allocator need to handle more requests for virtual channels.

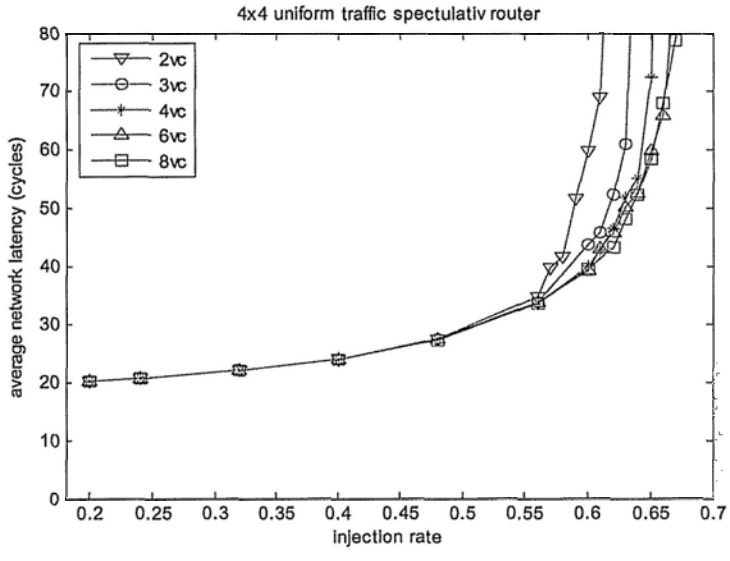
The new bypassing schemes will also influence the efficiency of virtual-channel flow control that is different from generic routers. Thus, a reasonable number V of virtual channels is the very important network parameter.



(a)



(b)

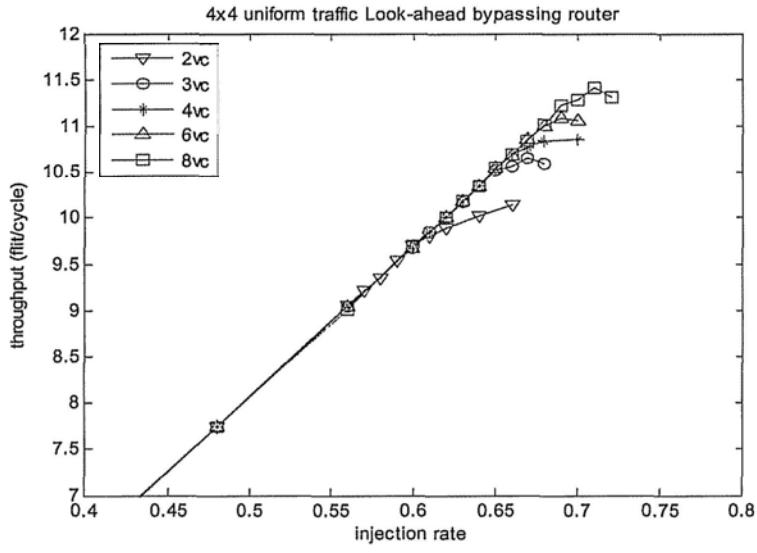


(c)

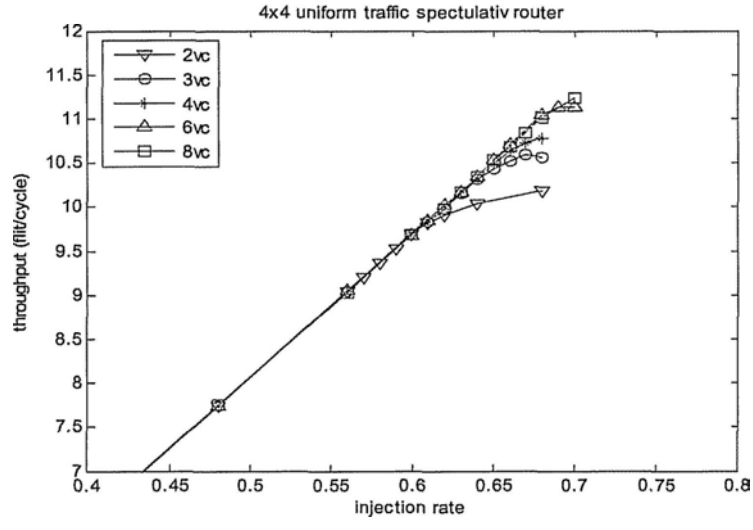
Figure 6.1 Average latencies of various virtual-channel numbers when buffer size is 8, traffic pattern is uniform, and router is (a) Lookahead bypass router, (b) No-load bypass router, (c) Baseline (speculative-pipeline) router.

At first, the uniform traffic pattern is evaluated when each virtual channel of all routers uses an eight-flit input buffer. The average packet latency as a function of the processing elements' average injection rate is plotted in Figure 6.1 for various virtual-channel numbers. The curves of various VCs overlap when the injection rate (offered traffic load) is low in all types of routers. It means that the effect of virtual

channel flow control can be ignored because the routers always keep idle. The virtual channel flow control becomes valuable only when the injection rate is high.



(a)



(b)

Figure 6.2 Throughput of various virtual-channel numbers as a function of injection rate when buffer size is 8, traffic pattern is uniform, router is (a) lookahead bypass router, (b) baseline router.

Figure 6.2 (a) shows the throughput of lookahead routers for various virtual-channel numbers. The virtual channel flow control is useful to improve the saturation throughput because it overcomes the head blocking problem, mentioned in the discussion in section 2.3. Of course, the average packet latency becomes unacceptable when the network reaches the saturation throughput. Thus, the saturation throughput

does not make any sense for practical applications. In a real application, the working load should be lower than the saturation throughput to assure that its average packet latency acceptable. Here we need to pay more attention to a network's average packet latency of the working traffic load, rather than its saturation throughput.

The router, which has more virtual channels, can get better average latency when the traffic reaches to a certain degree. The curves of average latencies, where routers have more than three virtual channels, are fairly close in Figure 6.1 even if the network has high traffic load.

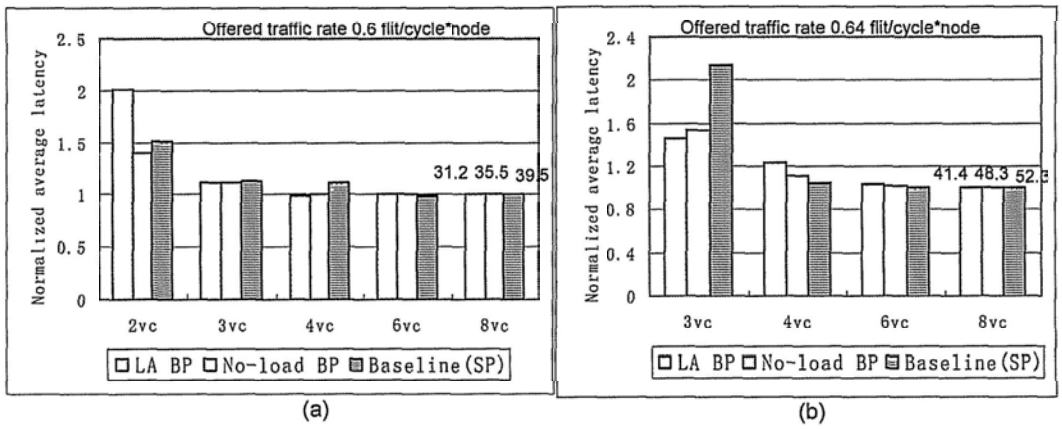


Figure 6.3 Normalized average packet latency as a function of VCs when buffer size is 8, traffic patter is uniform and throughput is near saturation

The effect of VCs to improve average packet latency is compared to the normalized average latency of various VCs in three types of routers when the throughput is near saturation, as shown in Figure 6.3. The injection rates are (a) 0.6 flit/node*cycle and (b) 0.64 flit/node*cycle respectively (9.6 and 10.24 flit/cycle network offer traffic load).

The comparison in Figure 6.3 (a) indicates that the lookahead router is very sensitive to a lack of virtual channels. This is because there are interlocks in our lookahead router, which is shown in Figure 5.15. A lookahead bypass router with a small number of virtual channels decreases the utilization ratio of physical channel because of

interlock problems. It makes the advantage of lookahead bypass to become impaired.

Because virtual-channel flow control can resolve a part of the head blocking problems, the utilization ratio of physical channel is improved when the network has high traffic load with a uniform traffic pattern. It means that the additional cost of virtual channels can earn the gain of improvement of network performance.

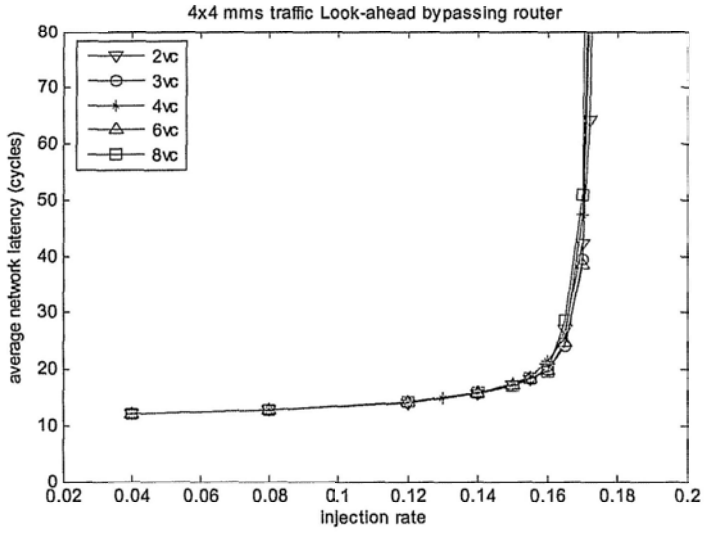
However, the lookahead router is more insensitive to the increasing virtual channels as compared to other two types of routers when the virtual channels reach a certain number (more than three virtual channels). It is because lookahead bypass can bypass more flits to reduce the occupied time in buffers of each virtual channel.

It indicates that having too many virtual channels will waste resources because the corresponding improvement is tiny. Choosing a reasonable number of virtual channels is necessary to balance between cost and performance.

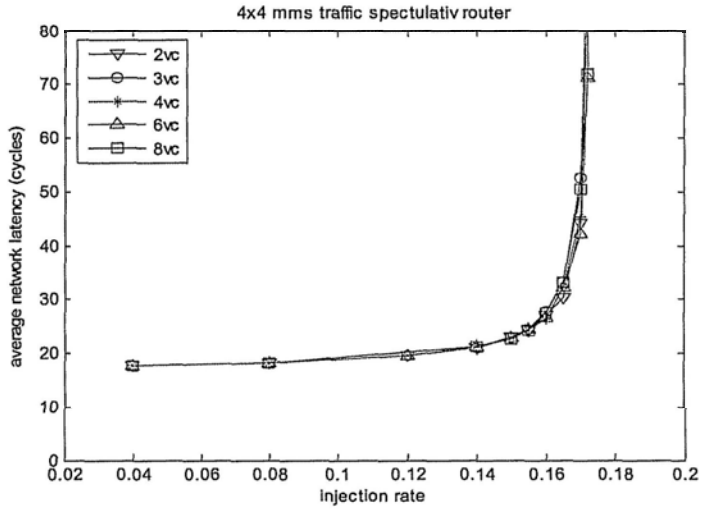
However, if traffic flows are mostly distributed between several pairs of processing elements, such as in the MMS traffic pattern, the advantage of virtual channels is less useful. Bottlenecks will appear in the path routers among these hot elements whereas the traffic load of the whole network is not high (only about 0.16 flit/node*cycle with the MMS traffic pattern in Figure 6.4). Here, the head blocking problem can be ignored.

All curves of average latencies of various virtual channels are nearly the same for both the lookahead bypass router and the baseline (i.e. the speculative-pipeline router). The role of virtual channels is useless as soon as the network has several very hot paths. It means that the effect of increasing the number of virtual channel is small if a

network-on-chip delivers the message that is similar to the MMS traffic pattern.



(a)



(b)

Figure 6.4 Average latencies of various virtual-channel numbers when buffer size is 8, traffic pattern is MMS, and router is (a) Lookahead bypass router, (b) Speculative-pipeline router

The total buffer sizes of various virtual channels are different in Figure 6.3 and Figure 6.4. The router that has more virtual channels costs larger total buffer size because the router has the same buffer size per channel. There is a little unfair in the comparison. We fairly compare the average latencies of routers with various virtual channels, which cost the almost same size of total buffer by various buffer sizes per channel in Figure 6.5. The buffer size per port is 32 flits, 64 flits, respectively.

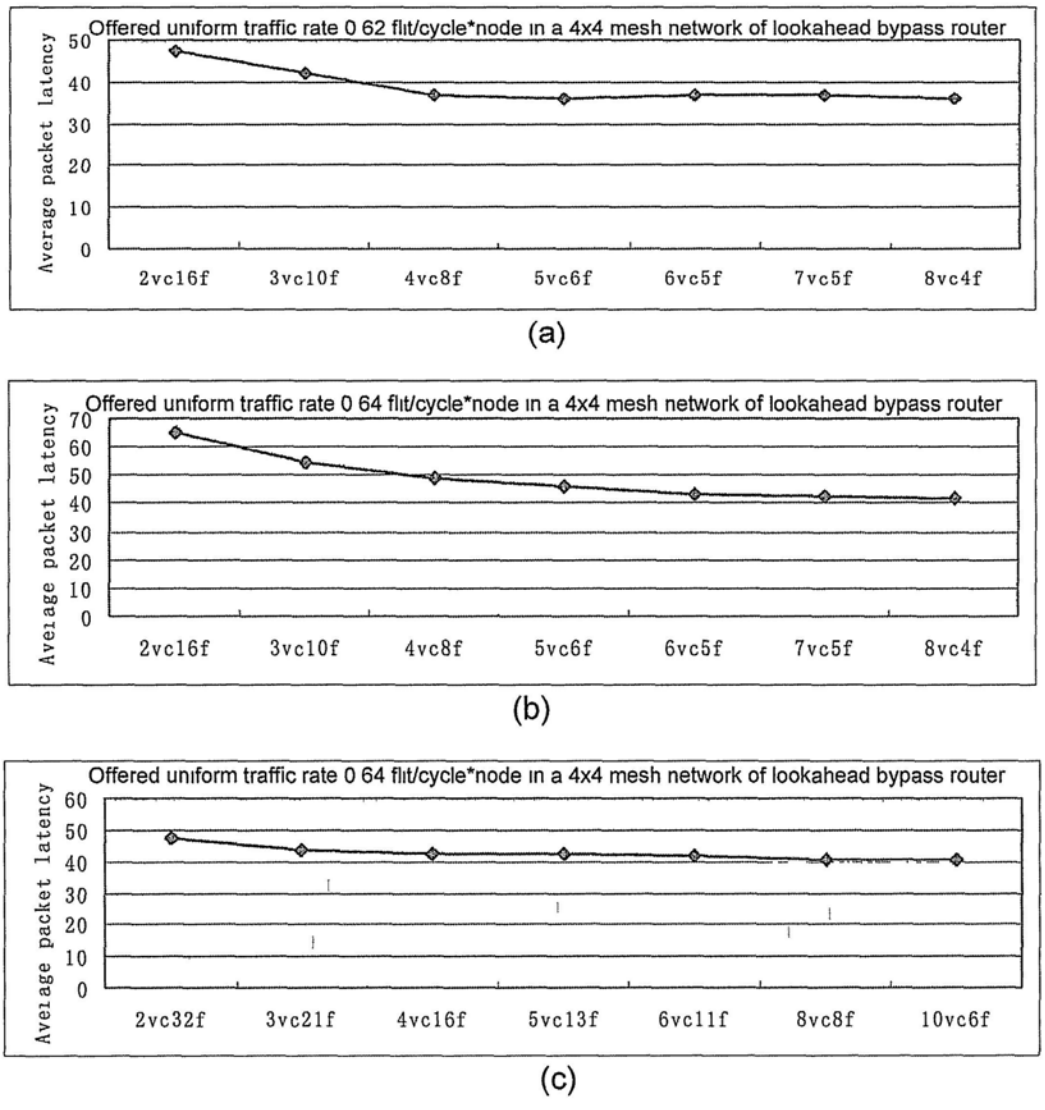


Figure 6 5 Average packet latency as a function of VCs when traffic patter is uniform, throughput is near saturation and total buffer size (a), (b) is about 32 flits, (c) is 64 flits

Based on Figure 6.5 (a), the latencies of 2-vc and 3-vc routers are about 33% and 16% more than 8-vc router's latency while the traffic is 0.62 flit/cycle*node in the uniform traffic pattern. With the increase of the traffic load, the virtual-channel flow control is playing a greater role. The data becomes about 40% and 20% if the traffic load increases to 0.64 flit/cycle*node in Figure 6.5 (b) because a router with a small number of virtual channels reaches the saturation point under a lower traffic load.

It is noted that a large buffer size can slightly compensate the latency for a smaller

number of virtual channels because a large buffer can increase the saturation throughput, as shown in Figure 6.5 (c).

The conclusion is similar to the previous comparison. Increasing buffer size per channel cannot make up for the loss due to a lack of virtual channels. However, a controller with too many virtual channels is very complex and gains little in performance. Thus, using too many or too little virtual channels is not a good choice for lookahead bypass routers.

Depending on all of above evaluations, the implementation of four ~ six virtual channels is a suitable choice range for lookahead bypass routers. In addition, the choice is also a reasonable number for the no-load bypass router and the baseline.

Thus, all routers in the following simulations use four virtual channels per port to achieve a reasonable balance between the cost and performance.

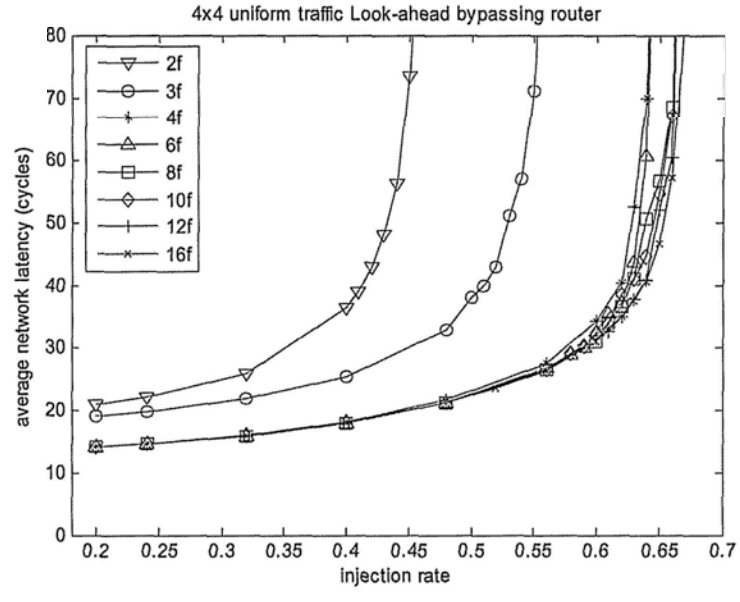
6.1.2 Buffer size

Adding enough buffers to our networks results in a significantly more efficient flow control because a large buffer can decouple the allocation of adjacent channels completely. However, the improvement approaches zero after the buffer size reaches a certain degree. Different flow controls and architectures have different requirements of buffer. Here we further discuss the influence of various buffer sizes in 4-VC routers.

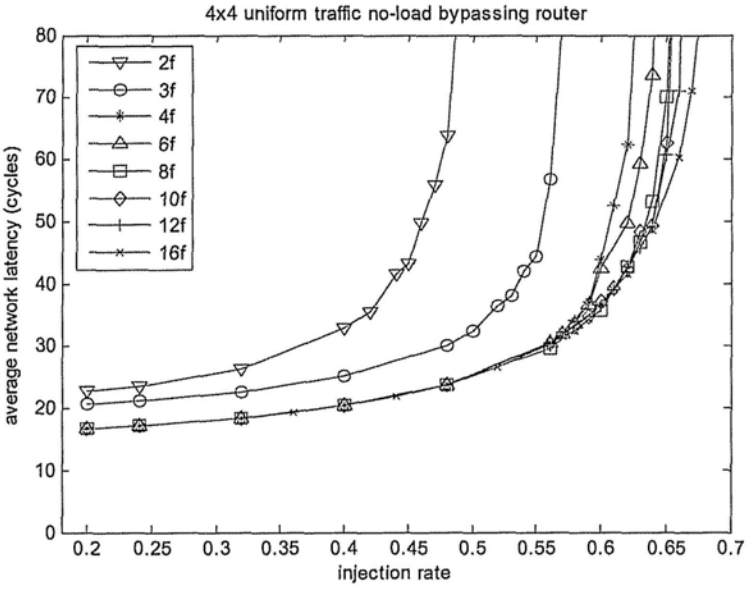
Figure 6.6 shows the average latency of various buffer sizes of the input FIFO in three types of routers. In general, if a router bypasses more flits, the fewer number of buffers are occupied. The latency of our lookahead router is better than that of no-load bypass

routers with same buffer size. Moreover, no-load bypass router's latency is better than the baseline's latency.

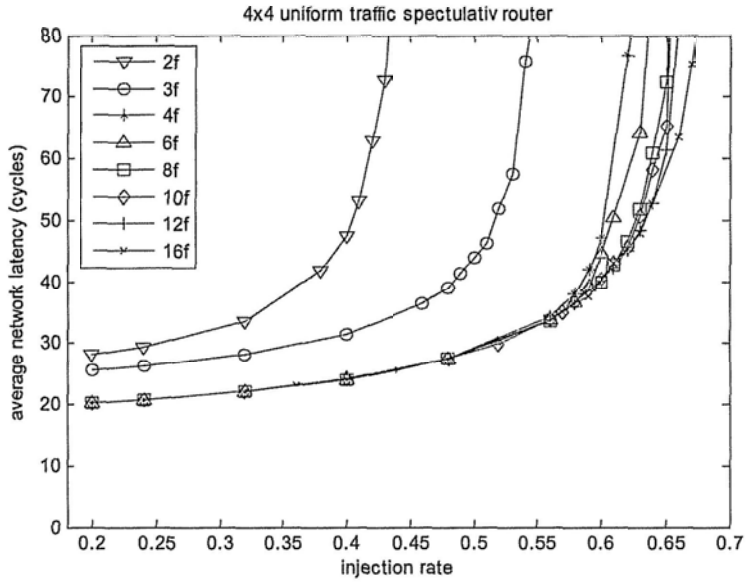
Of course, the requirement of buffers is also changed while the bypass scheme is introduced because bypass scheme can remit some actions of writing and reading buffer, which reduces buffers' utilization ratio.



(a)



(b)



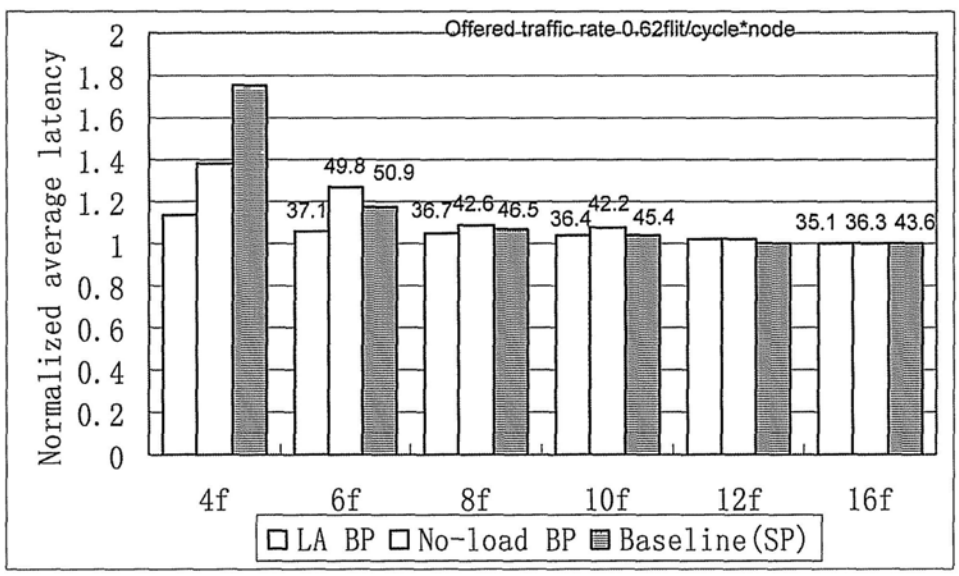
(c)

Figure 6.6 Average latencies for various buffer sizes when traffic pattern is uniform, and router is (a) Lookahead bypass router, (b) No-load bypass router, (c) Speculative-pipeline router.

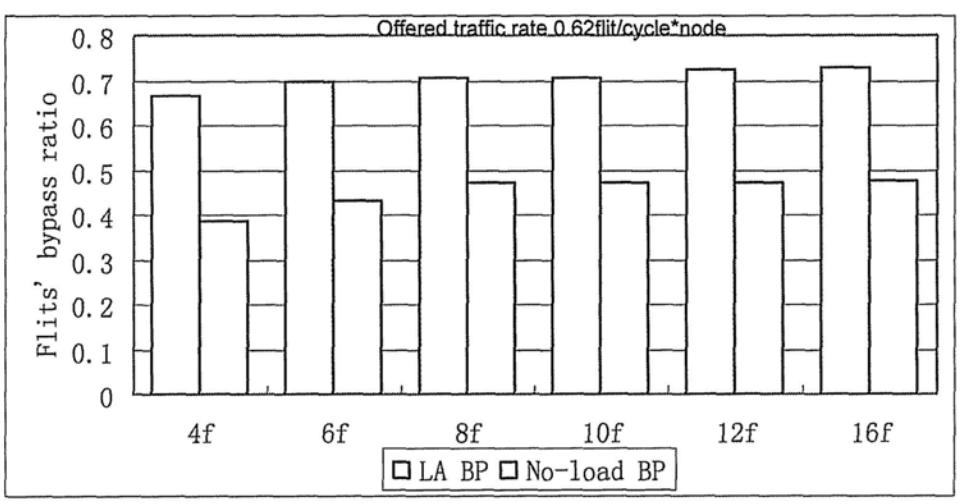
As a result, it cuts down the need of buffers. A router that supports bypass scheme needs fewer input buffers as compared to a router that does not. The latencies of all buffer sizes of lookahead bypass routers are better than others router's latencies whatever the traffic loads are. In other words, a router can have the same performance depended on fewer number of buffers if bypass opportunity increases. It is noted that the average packet latencies obviously increase whatever the traffic pattern is if the input buffer size is less than four flits (e.g. the curves of 2f and 3f in Figure 6.6).

It is mainly because the credit round-trip delay t_{crt} is four cycles in our implementations. Thus, buffer size F should satisfy $F \geq \frac{t_{crt} b}{L_f}$, refer to the discussion in section 2.1.2. Only the bandwidth limit of one flit per cycle can guarantee the continuous traffic flow. Too small buffer size seriously impairs the performance of routers. It means the router can be operated at full speed only if buffer size is no less than four flits.

The curves of the routers that have no less than four flits per virtual channel are nearly overlap when the network traffic load is not high in Figure 6.6. Only when the traffic load is near the saturation throughput, there is obvious difference in the routers of various buffer sizes. The larger buffer size means the lesser average latency and the higher saturation throughput. It proves that increasing buffer size can improve the network performance in all routers, shown in Figure 6.6 (a ~ c).



(a)



(b)

Figure 6.7 Normalized average packet latency and bypass ratio as a function of buffer sizes when traffic patter is uniform and throughput is near saturation.

Nevertheless, large buffer size cannot resolve the bottlenecks of physical channel. Oversize input buffer costs too large silicon area and high access power consumption. However, the improvement of oversize buffer is useless to all types of routers.

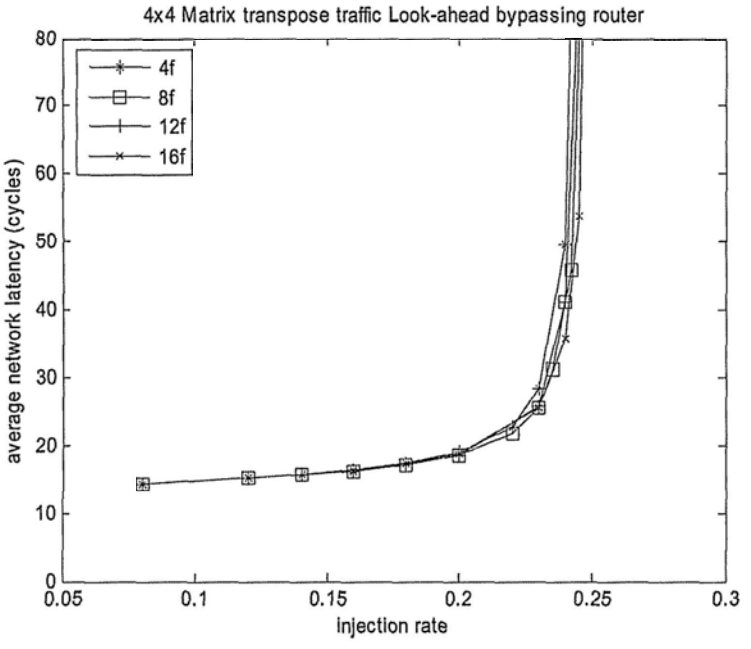
The trend of various buffers for three types of routers is compared by the normalized average latency of various buffer sizes in Figure 6.7 (a). The inject rate of traffic is about 0.62flit/node*cycle (about 9.92flit/cycle network load).

Since routers with bypass scheme attempt to use less buffers, large buffer size will provide little improvement in network latency. The increment of buffer size can give less help to decrease average packet latency in lookahead bypass routers than in other routers.

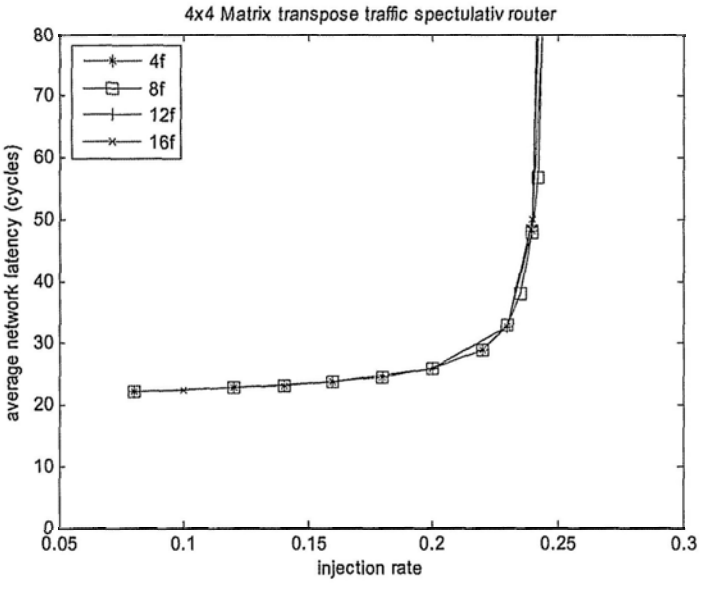
The lookahead bypass routers with 16-flit input buffers only improve 13% average packet latency than the lookahead bypass routers with 4-flit input buffers. The improvement of no-load bypass routers and baseline routers become 38% and 75% respectively. It is obvious that the lookahead bypass router need less buffer size than other two routers.

After the buffer size increases to more than four flits in lookahead bypass routers, the latency is very close to the 16-flit router's. There is only 5.5% or 4.5% difference between 6-flit or 8-flit router and 16-flit router.

In addition, the corresponding bypass ratios of lookahead bypass router and no-load bypass router are shown in Figure 6.7 (b). The growth of bypass ratio arise from lookahead bypass scheme is remarkable, which can cut the number of buffer access down.



(a)



(b)

Figure 6.8 Average latencies for various buffer sizes when traffic pattern is matrix transpose and router is (a) Lookahead bypass router (b) Speculative-pipeline router.

We have known that large buffer size cannot resolve the bottlenecks of physical channel. Thus, for an uneven traffic distribution such as MMS or matrix-transpose traffic pattern, large buffers hardly improve the average packet latency and saturation throughput. The estimation result of various buffer sizes is shown in Figure 6.8.

Whatever the router is, all curves are very close. Of course, the lookahead bypass router has obviously advantage in the latency over the baseline with matrix-transpose traffic pattern, which is similar to with uniform traffic pattern. The results substantiate this claim that our lookahead bypass routers can resolve the short of buffer. It means that our router can achieve better performance with less input buffers.

The requirement of buffer size in lookahead bypass routers is only four flits or more. To fairly compare the performances of different routers, eight-flit buffer per virtual channel is a reasonable choice, which can balance between cost and performance for all types of routers. Thus, all simulations of next section will apply eight-flit buffer to estimate the performance in various network environment.

6.2 The evaluation of various network environments

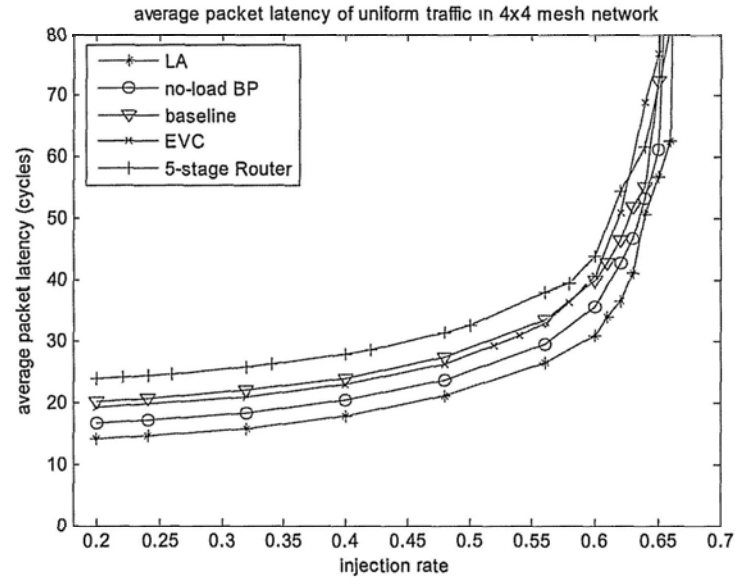
According to the analysis of evaluation results in the above sections, a suitable choice of virtual channel and buffer size can be determined. In this section, all routers apply four virtual channels per port and eight-flit buffer per channel.

6.2.1 Various traffic patterns

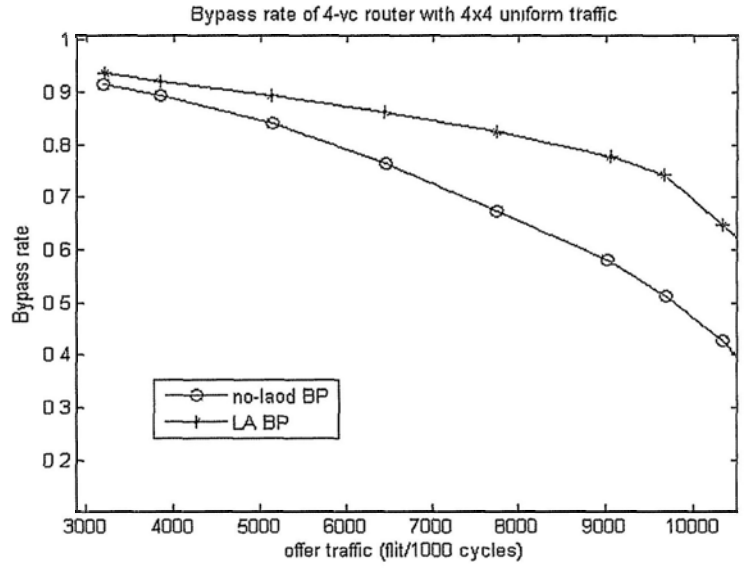
Various network environments (traffic patterns, and network scales) are simulated and analyzed to evaluate the lookahead bypass scheme. We compare the performances of several types of routers.

Figure 6.9 presents average packet latency and bypass ratio with uniform traffic pattern in a 4x4 mesh network. Table 6.2 shows average packet latencies of various

routers and the improvement as compared with baseline router (speculative-pipeline router) under three traffic loads.



(a)



(b)

Figure 6.9 (a) Average packet latency, and (b) Bypass ratio of various routers when network size is 4x4, packet size is 4 and traffic pattern is uniform.

Table 6.2 The improvement of average packet latency when traffic pattern is uniform

Traffic load (flit/node*cycle)	Baseline	LA	No-load BP	EVC	5-stage router
Low: 0.20	20.3054	14.0511	16.6255	19.28580	24.0211
	N/A	-6.2543	-3.6799	-1.0196	+3.7157
Medium: 0.40	24.06016	17.93311	20.42553	23.65381	27.89144

	N/A	- 6.12705	- 3.63463	-0.40635	+ 3.83128
Very High: 0.64	57.5823	41.9252	48.9638	68.85141	63.5963
	N/A	-15.6571	-8.6815	+11.26911	+6.014

The conventional 5-stage router has the largest packet latency, which will be not discussed thereinafter. The proposed lookahead bypass router can achieve the smallest average packet latency every time. The lookahead bypass router provides much improvement as compared with the baseline. There is 6.25 cycle (30.8%) reduction in latency when injection rate is low (0.2 flit/node*cycle). The latency is near the ideal performance when network load is low because the confliction between two flits is rare in a router. Thus, the improvement is mostly derived from the advantage of lookahead bypass router's pipeline architecture.

$$\Delta Latency = \sum_{Network}^{Packet} \left(T_{cell} - T'_{cell}\right) \Big/ N_{Packet} \qquad <6.1>$$

The difference of average packet latency is the accumulative average of all number of packets (N_{Packet}) through the network from the source to the destination. Equation 6.1 gives an ideal model for reduction in latency. Then applying statistics results the probability of bypass (P_{bypass}) and the packets' average traversal distance of hop count (D), we can get a model to calculate the latency difference from pipeline structure between the lookahead bypass router and the baseline, according to Equation 2.4.

$$\Delta Latency = \Delta T_{NI} + D \bullet \left[T_{router} - \left(T_{router} \bullet \left(1 - P_{LA}\right) + T_{LA} \bullet P_{LA} \right) \right] \qquad <6.2>$$

The difference of minimum propagation delay between lookahead bypass routers (T_{LA}) is two cycles and baseline (T_{router}) is four cycles. The propagation delay of a lookahead bypass router keeps T_{router} if a flit fails to bypass. Because the payload is late one cycle after the control requirement, the network interface of lookahead router cost one more

cycle to receive all flits of a packet. If the traffic load is 0.2 flit/node*cycle uniform traffic pattern in a 4x4 mesh network, P_{bypass} is 0.936 and H is 3.665. We calculate that the theoretical improvement from the pipeline advantage is 5.86 cycle, which is very close to the simulation result.

Of course, with the injection rate increases, the lookahead bypass router still outperforms the baseline. It is because the other effect of our lookahead bypass scheme is gathering strength with the traffic load grows. The difference of average packet latency is derived from the difference of flow control that leads the difference of utilization ratio. The proposed architecture can shorten the time of occupying router resources to increase router's utilization efficiency. It means that the reduction in latency is not only from small propagation delay but also from more efficient allocation of physical channel and buffer. Thus, the new router gains more improvement in packet latency when the throughput increases and goes near the saturation point.

With the growth of injection rate, the conflictions happen more often in a network. The bypass ratio drops so that the improvement in latency from bypassing pipeline drops. Of course, the increasing traffic flows are more sensitive to the size of free buffer space. With the increasing traffic flow, free buffers, which lookahead bypass routers provide, play a greater role to make up the lost from the drop of bypass ratio. When network load increases to 6.4 flit/cycle (0.4 flit/cycle*node injection rate), the improvement in latency of lookahead bypass router still keep about 6.125 cycle.

However, the bypass schemes cannot really resolve the bottleneck of physical

channels. The traffic jams of busy paths hot up very quickly when network approaches the saturation point. It makes the improvement of lookahead bypass scheme inconspicuous. When the network load is very close to saturation, the average packet latency is so much high that the working throughput point is worthless for a real application. Therefore, we only compare the working point that is not but near the saturation throughput.

That the reduction achieves 13.1 cycles in average latency when the network has a high load (0.6 flit/node*cycle). The reduction of latency increases to 15.75 cycles if injection rate is 0.64 flit/node*cycle. The raise of improvement in latency profits from the slight improvement in the saturation throughput of lookahead bypass router. The baseline's saturation point is smaller than lookahead bypass router's. Moreover, larger injection rates make results of all routers worthless.

Lookahead bypass has a significant advantage over no-load bypass too. The reduction is 2.58 cycles (15.5%) as compared with the no-load bypass router under low injection rate. Although the probability of lookahead bypass (P_{LA}) and the probability of no-load bypass ($P_{BP} = 0.917$) is very close when injection rate is low, which is shown in Figure 6.9 (b), the improved pipeline stage of lookahead bypass router can still reduce the propagation delay of a router.

$$\Delta Latency = \Delta T_{NI} + H \bullet \left\{ \left[P_{BP} \bullet T_{BP} + (1 - P_{BP}) T_{router} \right] - \left[P_{LA} \bullet T_{LA} - (1 - P_{LA}) T_{router} \right] \right\} \quad <6.3>$$

Applying the equation 6.3, the ideal latency difference from the pipeline architecture is about 2.5 cycles, which is near the latency improvement when the traffic load is low.

The network works under high load, the bypass ratio drops a little. The latency

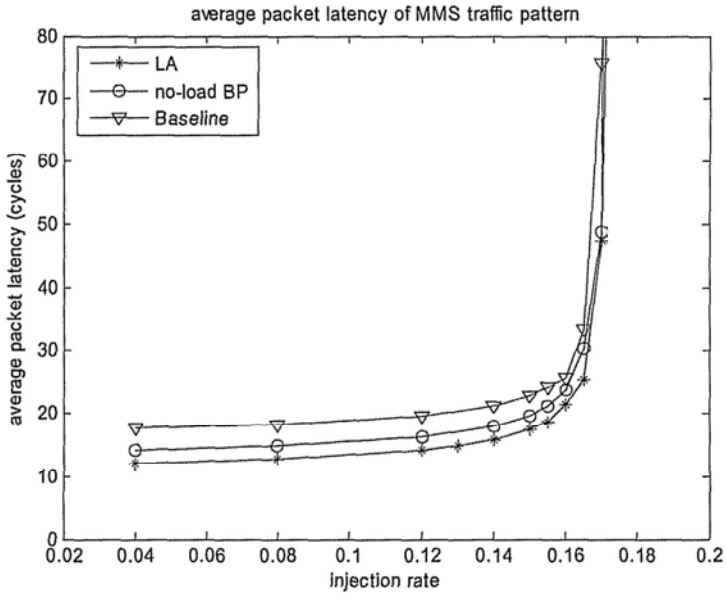
reduction in our lookahead bypass, which is derived from pipeline and flow-control, increases to 5.68 cycles (15.3%) and 7.04 cycles (14.4%) respectively under 0.6 and 0.64 flit/node*cycle injection rates.

Here, we specially list the latencies of EVC improvement as a reference. When the network load is low, the pipeline advantage of EVC is not obvious under uniform traffic pattern. EVC router only can bypass a little part of packets in some routers although the propagation delay is one cycle in the EVC bypass routers. There is only 1.019 cycle reduction in average packet latency as compared with the baseline router. Because the EVC architecture reserves a part of virtual channels only to deliver EVC packets in end of EVC paths, it could counteract the improvement from EVC bypass in the performance of latency when the traffic load increases. EVC router has only 0.406 cycles less than the baseline in the latency if the injection rate is 0.4 flit/cycle*node. Moreover, EVC router has a bad result (11.26 cycle incensement) if the injection rate is 0.64 flit/cycle*node. Of course, all data are worse than lookahead bypass router, even and the no-load bypass router.

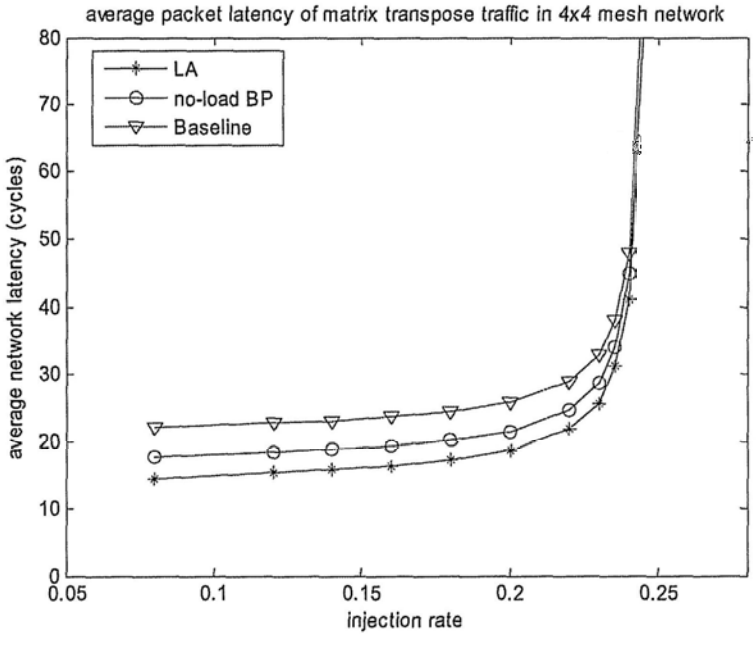
Based on the simulation result, the conclusion is that the lookahead bypass router has large advantage in packet latency under uniform traffic pattern in a 4x4 mesh network every time.

We further analyze some traffic patterns to evaluate the value of lookahead bypass scheme with different spatial distributions and traffic flows. Figure 6.10 presents the simulation results. The traffic patterns include MMS, matrix transpose, fft and jpeg case of p-model traffic.

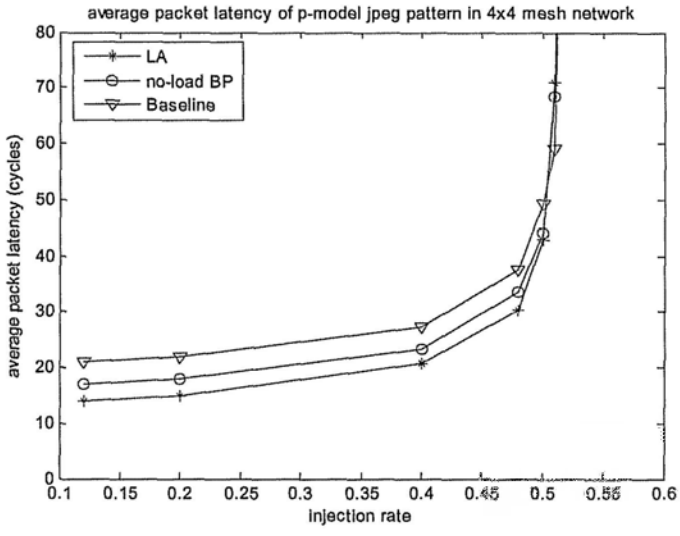
We know that the distribution of MMS and matrix transpose traffic patterns are very uneven, which concentrate in several nodes. It is different from uniform and p-model traffic patterns. When bottleneck nodes are saturation, the other nodes always keep idle in MMS and matrix transpose traffic patterns.



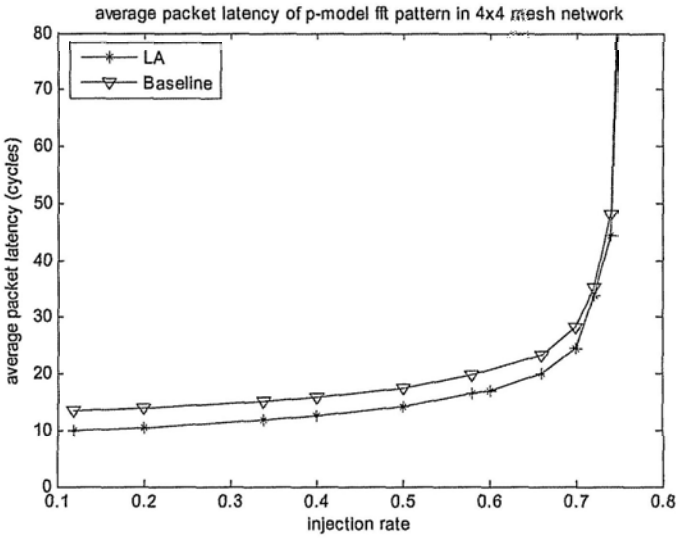
(a)



(b)



(c)



(d)

Figure 6.10 Average packet latency when the network size is 4x4, packet size is 4, and traffic pattern is (a) MMS, (b) matrix transpose, (c) p-model jpeg, (d) p-model fft.

Table 6.3 The average packet latency

Traffic load (flit/node*cycle)	Baseline	LA	No-load BP
MMS: 0.12	19.46969	14.08721	16.18282
MMS: 0.17	75.32384	47.33547	48.72091
Matrix transpose: 0.12.	22.71782	15.21920	18.39129
Matrix transpose::0.24	47.83877	41.16405	44.80782
p-model jpeg:0.12	20.91372	13.97708	16.92352
p-model jpeg:0.51	49.18007	42.84029	44.13292
p-model fft:0.12	13.51287	10.02852	11.27277
p-model fft:0.74	48.05013	44.32588	46.95218

It makes their saturation throughputs of whole network much lower than the traffic patterns' with uniform distribution. The bottlenecks become much heavy when the injection rate only increases a little.

We list the details in Table 6.3. For MMS traffic pattern, there is 5.38 cycles reduction in latency when injection rate is 0.12 flit/node*cycle. Moreover, the new architecture can improve saturation throughput slightly. The improvement of lookahead bypass router becomes larger (27.9 cycle) when injection rate is 0.17 flit/node*cycle. It is because the working point has been very near the baseline's saturation throughput. The lookahead bypass router's average latency also quickly reaches 210.234 cycle when injection rate only increase to 0.175 flit/node*cycle, which is worthless for a real application.

Under matrix transpose (MT) traffic pattern, network bottlenecks are two corner routers that are at left top and the right bottom. In addition, most packets go through the two corner routers. When load is low, the reduction in latency is 7.5 cycle. But when load is near saturation throughput, the traffic jam in two corner routers leads most packets need to wait. The latency of MT traffic pattern increases more quickly with the growth of injection rate than of uniform traffic. Accompanying the injection grows from 0.235 to 0.24 flit/cycle*node, the average packet latency increase rapidly from 31.23 to 41.16 cycle.

Of course, there still is 6.67 cycle reduction in latency before the network reaches saturation (0.24 flit/cycle*node). Higher injection rate leads that average packet latencies of all routers are more than one hundred cycles.

It has been mentioned that the p-model traffic patterns have more balanced distribution in different orientations in a mesh network than uniform traffic pattern. Each port has a similar traffic flow. It makes that the working efficiency of baseline has been very high. Thus, the part of improvement in latency, which is derived from low buffer usage ratio, under p-model patterns is smaller than under other un-balanced mesh-spatial-distribution traffic patterns, including uniform traffic pattern. Sometimes, this type of improvement only can make up the lost in latency from the reduction of bypass ratio under p-model patterns.

The result of JPEG case of p-model is shown in Figure 6.10 (c). There is 6.937 cycle reduction in latency of lookahead bypass router when injection rate is low because packets' average traversal distance is about 3.98 hop count. In addition, the reduction in latency becomes 6.34 cycle when injection rate is 0.51 flit/cycle*node because the bypass ratio quickly drops to 65.27%. The drop of bypass ratio more quickly in jpeg case because each packet goes through more routers so that more conflictions happen.

Figure 6.10 (d) shows the result of FFT case of p-model, whose process elements have high relativity that results in a small average traversal distance (2.24 hop count). Under low injection rate (0.12 flit/cycle*node), the reduction of lookahead bypass is only 3.484 cycle as compared with the baseline. Under high injection rate (0.74 flit/cycle*node), there is 3.724 cycle reduction in latency. Here, the bypass ratio decreases from 96.7% to 71.8% when the injection rate increases from 0.12 to 0.74 flit/cycle*node. Thus, the reduction of latency is still a large part of tribute to the pipeline advantage of lookahead bypass scheme even in the high load case.

These results of various traffic patterns prove that the lookahead bypass scheme can play a great role to reduce the average packet latency in different network environment before the network reach its saturation. However, the lookahead bypass scheme only improves the saturation a little. Although the bypass scheme can bypass some flits to reduce propagation delay and power consumption, it cannot improve the traffic flow density in any ports. The overlarge flit density in bottleneck ports would impede the continuous incensement of network throughput.

Table 6.4 Network improvement per hop in latency of various traffic patterns when network size is 4x4, and traffic load is 0.12 flit/node*cycle.

Traffic pattern	Net Latency reduction*	Average hop	Bypass ratio	reduction per hop
Uniform	7.3096	3.67	0.963	1.992
MMS	6.382	3.29	0.954	1.939
Matrix transpose	8.499	4.34	0.959	1.958
p-model jpeg	7.937	3.98	0.962	1.994
p-model fft	4.484	2.24	0.967	2.002
p-model gzip	4.031	2.00	0.950	2.016

* The latency reduction is the difference between lookahead router's latency and baseline's latency that removes the influence of network interface. Reduction = Latency (baseline) – [Latency(LA) – 1].

The average traversal distance of packets in different traffic patterns are different. To fairly evaluate the lookahead bypass router in different traffic patterns, we compare the effect of lookahead bypass scheme in various traffic patterns. Table 6.4 lists the network reduction in latency per hop when traffic load is 0.12 flit/node*cycle. The traffic load is so low that packets can go through all routers without any waiting. The reduction per hop should be same in theory.

The five traffic patterns have similar reduction per hop that is about 2.0 cycle. The reduction per hop of the traffic with uniform distribution is a little larger than those traffics with uneven distribution. In addition, the small average traversal distance

slightly benefits the improvement in latency.

6.2.2 Various networks

Some environment parameters vary with the networks. A larger network implies more source processing elements can send their message to more destinations at same time. In some applications, it means that the average traversal distance of packets increases so that the average packet latency rises. If the location mapping of processing elements is good enough, some applications can maintain the average traversal distance of packets.

More simulation results of various network scales are used to evaluate the lookahead bypass scheme. The networks include an OCN memory network, a 6x6 network, and an 8x8 network.

Uniform traffic patterns, matrix-transpose traffic pattern, a p-model traffic pattern, and two cases of OCN traffic traces are chosen. MMS traffic pattern is ignored because it is only mapped in a 4x4 mesh network. The traffic patterns in different network sizes should be different but have some similar characteristics of traffic flows. All simulation results are shown in Figure 6.11.

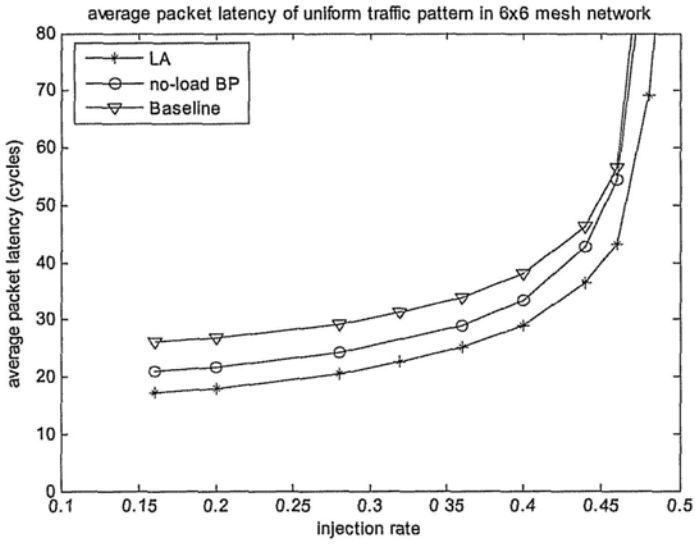
Matrix-transpose traffic pattern would have a larger average traversal distance of packets in a larger network. It makes the jam worse because more packets from the increased processing elements need to pass the corner hot points.

Although uniform traffic pattern has a more even traffic flows than matrix-transpose traffic, one processing element send uniform packets to each other processing element.

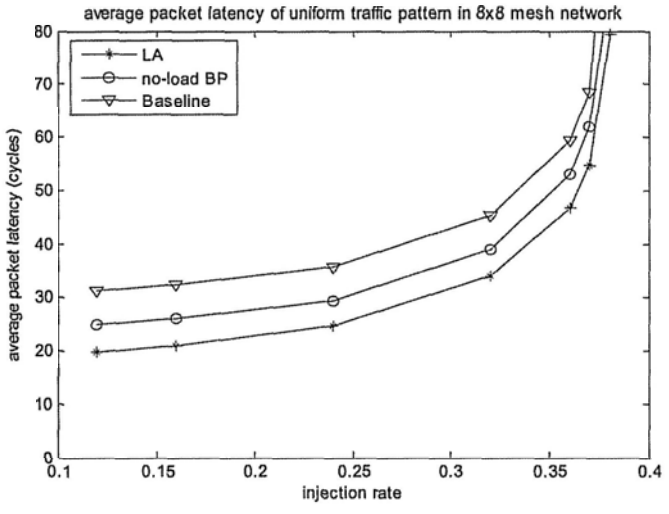
Thus, its spatial distribution changes and its average traversal distance increases with the growth of network scale.

In additional, p-model traffic patterns are designed by the relativity of processing elements, which can hold their average traversal distance. It means the increasing network size only has a little impact on the average latency.

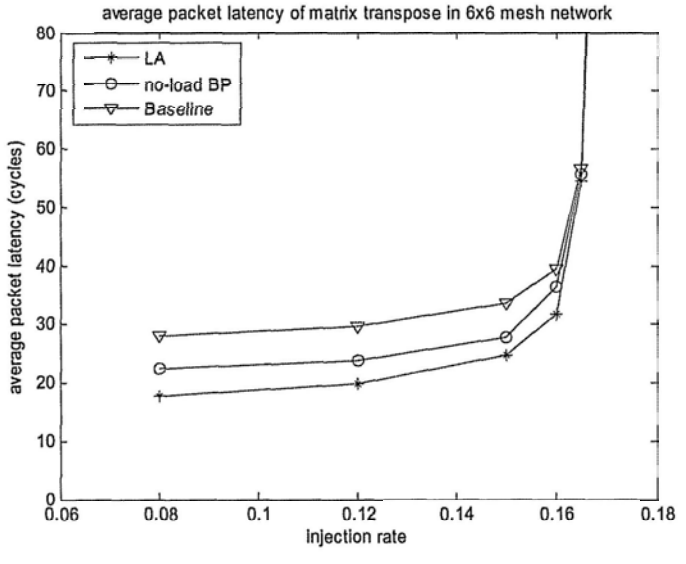
The traffic traces of OCN memory network work in a 10x4 mesh network. The spatial distribution of mesh network is also uneven in most cases. Here we choose equake and gzip cases to evaluate our lookahead bypass router. The 10x4 network is different from a square network such as 4x4, 6x6, and 8x8.



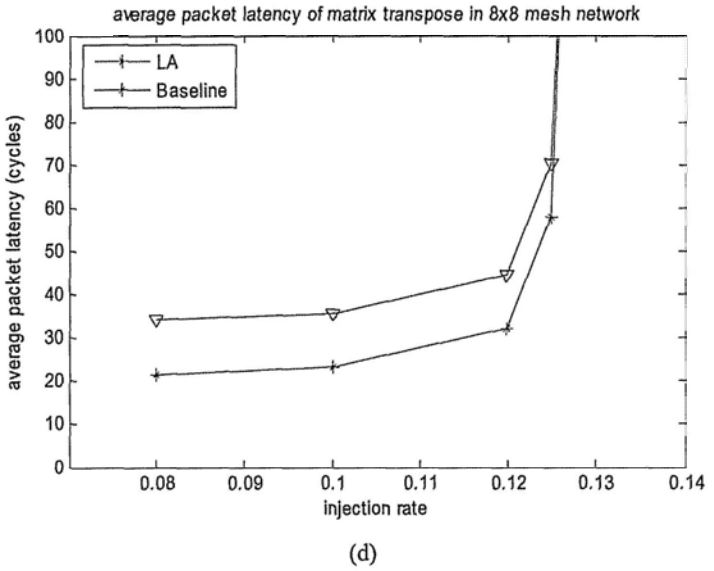
(a)


average packet latency of uniform traffic pattern in 8x8 mesh network
LA
no-load BP
Baseline
average packet latency (cycles)
injection rate
0
10
20
30
40
50
60
70
80
0.1
0.15
0.2
0.25
0.3
0.35
0.4

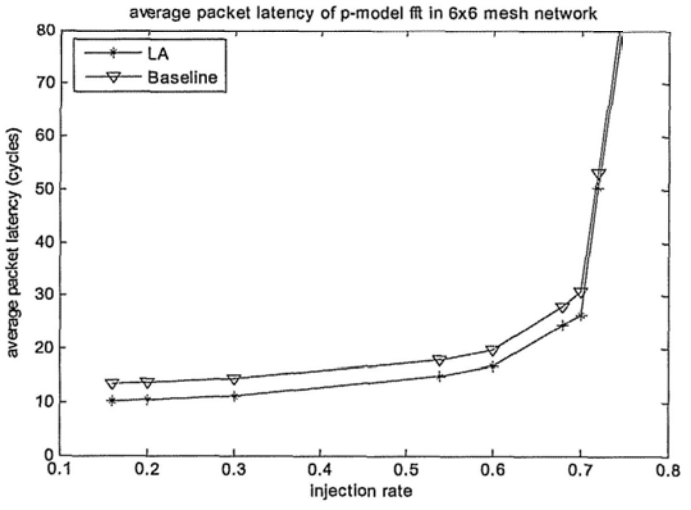
(b)


average packet latency of matrix transpose in 6x6 mesh network
LA
no-load BP
Baseline
average packet latency (cycles)
injection rate
0
10
20
30
40
50
60
70
80
0.06
0.08
0.1
0.12
0.14
0.16
0.18

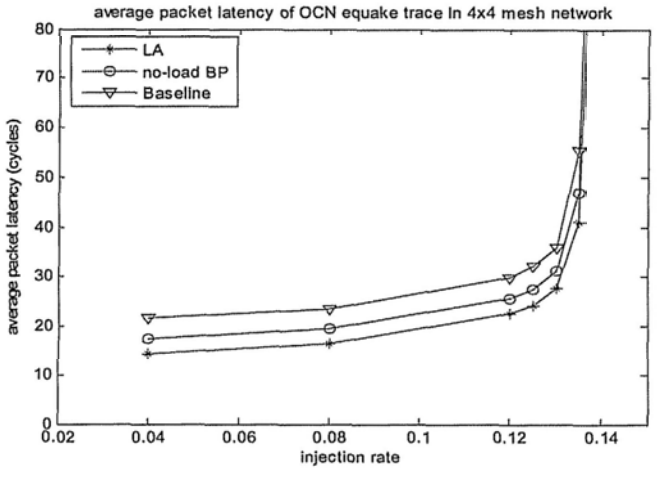
(c)


average packet latency of matrix transpose in 8x8 mesh network
LA
Baseline
average packet latency (cycles)
injection rate
0
10
20
30
40
50
60
70
80
90
100
0.08
0.09
0.1
0.11
0.12
0.13
0.14

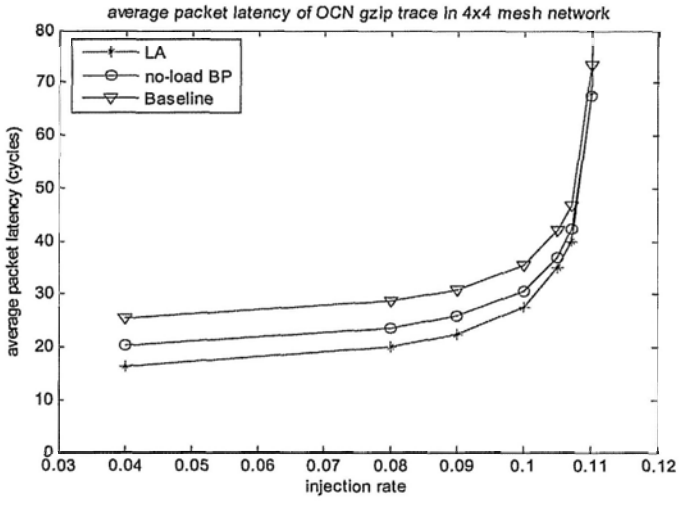
(d)



(e)



(f)



(g)

Figure 6.11 Average packet latency when, packet size is 4, the network size and traffic pattern are (a) 6x6 uniform, (b) 8x8 uniform, (c) 6x6 matrix transpose, (d) 8x8 matrix transpose, (e) 6x6 p-model fft case , (f) OCN memory network gzip traffic trace ,(g) OCN memory network equake traffic trace.

Table 6.5 The network performance of various network sizes when traffic pattern is uniform

Net	Latency (cycle)^	Reduction (cycle)^^	Average hop	Reduction per hop (cycle/node)	Saturation Injection rate	Saturation throughput	ST per hop (flit /cycle*node)
4x4	14.051	7.250	3.67	1.9755	<0.7	10.8	0.675
6x6	17.938	9.783	5.00	1.9566	<0.5	18	0.5
8x8	22.661	12.129	6.32	1.9192	<0.4	25.6	0.4

^Injection rate is 0.2 flit/cycle*node; ^^Reduction = Latency (baseline) – [Latency(LA) – 1].

At first, we evaluate the uniform traffic pattern in 4x4, 6x6, and 8x8 mesh networks. Table 6.6 lists the detailed comparison of network performance of various networks sizes. Because a processing element needs to send packets to each other processing element, packets would go through more routers from source to destination. It means more packets go through center hot routers. It is noted that some ports has a low workload in the network when the busiest ports have to deliver flits every cycle.

The maximum valid injection rate before saturation drops obviously from about 0.7 to 0.4 flit/cycle*node while the network size rises from 4x4 to 8x8. Although the global saturation throughput of whole network increases, the local saturation throughput of one processing element falls because of the worse traffic jams.

Of course, the average packet latency increases because packets need to pass more routers. The improvement per hop in latency from lookahead bypass routers can keep near 2 cycle because high bypass ratio in all network sizes. There is a little drop of the improvement per hop in latency with the average traversal distance increases. It is similar to the conclusion of various traffic patterns in a 4x4 mesh network.

Table 6.6 The latency reduction of various network sizes when traffic pattern is matrix transpose

Net	Latency (cycle)^	Reduction (cycle)^^	Distance(hop)	Reduction per hop	Byapss ratio	Saturation
4x4	14.47476	8.6332	4.343	1.9878	0.9741	>3.0
6x6	17.76198	11.138	5.665	1.9661	0.9641	>6.2

8x8	32.2303	13.1865	6.992	1.8859	0.9576	>8.2

^Injection rate is 0.08 flit/cycle*node; ^^Reduction = Latency (baseline) – [Latency(LA) – 1].

The traffic flows in hot routers under matrix-transpose traffic pattern increase proportionately to network size. The maximum traffic throughput of a port is 4 times, 6 times and 8 times injection rate (3 times, 5 time, and 7 times injection flow of a PE) respectively in 4x4, 6x6, and 8x8 networks. Thus, there is a great range in the traffic flows of different routers under the matrix transpose traffic. Some routers still are provided with enough free bandwidth when some routers are congested. The growth of saturation throughput of whole network becomes worthless after some busiest routers reach saturation.

The average traversal distance increases from 4.343 to 6.992. It results that the average packet latency rises very quickly. Moreover, the reduction in latency per hop and bypass ratio of lookahead bypass router drops a little because the more traffic flows congregate in several routers in a larger network.

Table 6.7 The network performance of various network sizes when traffic pattern is p-model fft

Net	Latency (cycle)^	Reduction (cycle)^^	Average hop	Reduction per hop (cycle/node)	Saturation Injection rate	Saturation throughput	ST per hop (flit /cycle*node)
4x4	10.502	4.4571	2.2408	1.989	<0.8	12.6	0.7875
6x6	10.399	4.3541	2.1912	1.987	<0.8	28.6	0.7944
8x8	10.240	4.3574	2.1910	1.988	<0.8	50.9	0.7953

^Injection rate is 0.2 flit/cycle*node; ^^Reduction = Latency (baseline) – [Latency(LA) – 1].

The type of traffics, similar to the p-model traffic, is different from the uniform and matrix transpose traffics. The destination of a packet is based on a uniform spatial distribution of parameter $\boldsymbol{p}$. The increasing network scale does not increase the average traversal distance of packets and even reduces it a little. From the results, we can know that the lookahead bypass router can keep the similar average packet latency

under a same injection rate for whatever the network size is. The improvement in latency, saturation injection rate, and throughput per hop of various network sizes all are nearly same. Of course, the whole network throughput can increase in direct proportion to the number of nodes.

Table 6.8 The network performance of various network sizes when traffic pattern is OCN

Traffic	Injection rate	Latency (cycle)	Reduction (cycle)^	Average hop	Bypass ratio	Reduction per hop (cycle/node)
OCN gzip	0.04	16.367	9.871	5.00	0.956	1.974
/	0.10	27.571	8.937	/	0.7942	1.787
OCN equake	0.04	14.351	8.300	4.16	0.963	1.995
/	0.125	27.722	9.244	/	0.830	2.222

^Reduction = Latency (baseline) – [Latency(LA) – 1].

The OCN memory network is a 10x4 mesh network, which is not a square. The gzip and equake traffic traces are simulated to evaluate the lookahead bypass router. The improvement of latency is marked both in two traffic traces.

The lookahead bypass scheme is more effective to OCN equake traffic trace. The more reduction in latency is provided by lookahead bypass router during the network is close to the saturation throughput.

The effect of our lookahead bypass scheme is proved in various networks and average traversal distance of packets. The average traversal distance of packets is an important factor in the average latency. However, it is only a minor factor to the improvement of the lookahead bypass router. The lookahead bypass router effectively works in different network environments.

6.3 The power evaluation

Equation 6.4 models the average energy that a network delivering a flit from a source

PE to a destination PE. The total network energy consumption for delivering message can be saved, if the average energy consumed by each router falls. Here, physical links between two routers and the network interfaces, which are in charge of injecting and receiving packets, are not considered.

$$\overline{E_{Flit}} = \sum_{Network}^{Flit} (E_{cell}) \Big/ N_{Flit} = E_{inj} + E_{recv} + (H-1) \times \overline{E_{Link}} + H \times \overline{E_{router}} \qquad <6.4>$$

If a flit succeeds to lookahead bypass, the payload data can directly go through the crossbar to the output port without accessing buffers. The power consumption of accessing FIFO can be saved, which is a large part of total power consumption in a router. The overhead of lookahead-bypass control logic's power consumption is enough small. Thus, the energy consumed by a lookahead bypass flit is less than the energy consumed by a regular flit. Equation 6.5 calculates the average energy consumption according to the bypass ratio.

$$\overline{E_{router}} = P_{LA} \times \overline{E_{LA}} + (1 - P_{LA}) \times \overline{E_{regular}} \qquad <6.5>$$

The saved energy varies with the network load because the network load influences the bypass ratio in a lookahead bypass network.

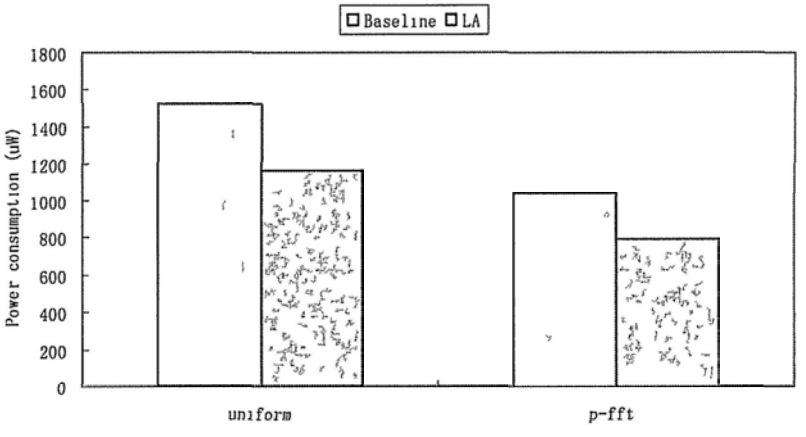


Figure 6.12 The power consumption of various traffics and injection rates when network is 4x4, packet is 4, virtual cannel number is 4 and buffer size is 8 flits.

For a router with 128-bit payload, the power of accessing FIFO accounts about 30%. Thus, more bypass ratio mean more power consumption save. The lookahead bypass network can save 23% of energy as compared to the baseline network even under high-load uniform traffic pattern. The network load can determine the power consumption because the traffic bandwidth and the bypass ratio are changed when the network load changes. Figure 6.12 lists the power consumption. Moreover, considering power consumption of link and network interface in NoC, the saving ratio of total power should shrink.

6.4 Summary

As compared with the baseline router, the lookahead bypass router is more sensitive to the lack of virtual channel. But too many virtual channels are also worthless to the lookahead bypass scheme. About four virtual channels per port is a good choice.

Moreover, the lookahead bypass router requires less input buffer because the bypass flits can save the regular flits enough buffer space in a lookahead router. When input buffers are no less that four flits per channel, the lookahead router can achieve a good performance. If the size of input buffers is more than eight flits per channel in a lookahead bypass router, the additional improvement in performance can be ignored.

The simulation results prove the marked effect of our lookahead bypass scheme. Whatever the traffic patterns and network scales, the proposed lookahead bypass router all outperforms the baseline router and no-load bypass router. There is a great advantage in pipeline stage for a lookahead bypass flit. The minimum propagation

can improve the saturation throughput a little by the less occupying time in a router. Of course, the lookahead bypass scheme cannot improve the traffic flow in a physical port. When a port achieves its limit, the congestion would spread over the whole network.

Although the additional lookahead logic brings about a little overhead in power consumption, bypassing buffers can save much power by avoiding operations of accessing buffer. Because the power consumption of accessing buffer is a large part of total power, the lookahead bypass router can save the operation power as compared with many routers.

CHAPTER 7. Short-Circuit Crossbar Channel

7.1 Internal traffic flows

We know that the different spatial distributions generate different traffic flows. Under some traffic patterns, there are several routers are much busier than the others in the whole network. We evaluate three patterns in a 4x4 mesh network: uniform, MMS and matrix transpose, whose spatial distributions have been discussed in section 2.4.3.

Table 7.1 Internal traffic flow distribution of left bottom four routers under 4x4 uniform pattern

R(1,0)	Local	North	East	South	West
Local	0	2/15	0.8	1/15	0
North	8/15	0	0	8/15	0
East	0.2	0.4	0	0.2	0
South	4/15	8/15	0	0	0
West	0	0	0	0	0

R(1,1)	Local	North	East	South	West
Local	0	2/15	8/15	1/15	4/15
North	8/15	0	0	8/15	0
East	2/15	4/15	0	2/15	8/15
South	4/15	8/15	0	0	0
West	1/15	2/15	8/15	1/15	0

R(0.0)	Local	North	East	South	West
Local	0	0.2	0.8	0	0
North	0.8	0	0	0	0
East	0.2	0.6	0	0	0
South	0	0	0	0	0
West	0	0	0	0	0

R(0,1):	Local	North	East	South	West
Local	0	0.2	8/15	0	4/15
North	0.8	0	0	0	0
East	2/15	0.4	0	0	8/15
South	0	0	0	0	0
West	1/15	0.2	8/15	0	0

The internal traffic flow distribution is shown in Table 7.1, where the flows from an input port/channel (left column) to an output port/channel (top line) are normalized to injection rate. Because uniform traffic pattern is symmetry, the table only lists four routers in left bottom of network. Each PE sends and receives the same flow. However, the internal traffic flows is uneven. The busiest input channels and the busiest output channel have 16/15 flows respectively in R(0,1), R(1,0), R(1,1). The flows of busiest

paths have 0.8 transport flow respectively in R(0,0), R(0,1), R(1,0).

Table 7.2 The internal traffic flow distribution under MMS traffic pattern

R0,0	Local	North	East	South	West
Local	0	0	0.0012	0	0
North	0.8445	0	0	0	0
East	0.0020	0.8475	0	0	0
South	0	0	0	0	0
West	0	0	0	0	0

R0,1:	Local	North	East	South	West
Local	0	0	0.0350	0	0.0020
North	0.7032	0	0	0	0
East	0	0	0	0	0.8475
South	0	0	0	0	0
West	0.0006	0	0.0006	0	0

R0,2	Local	North	East	South	West
Local	0	0	0.0195	0	0
North	0.1918	0	0	0	0
East	0	0	0	0	0.8475
South	0	0	0	0	0
West	0.0190	0.0159	0.0006	0	0

R0,3	Local	North	East	South	West
Local	0	0	0	0	0.8475
North	3.8555	0	0	0	0
East	0	0	0	0	0
South	0	0	0	0	0
West	0	0.0202	0	0	0

R1,0	Local	North	East	South	West
Local	0	0	1.8926	0	0
North	0.5118	0	0	0.8445	0
East	3.7535	0	0	0	0
South	0.0049	0.8426	0	0	0
West	0	0	0	0	0

R1,1	Local	North	East	South	West
Local	0	0	2.9092	0	1.8720
North	0	0	0	0.7032	0
East	0	0	0	0	1.8815
South	0	0	0	0	0
West	0.9463	0	0.9463	0	0

R1,2	Local	North	East	South	West
Local	0	0	0	0.1918	0
North	0.1759	0	0	0	0
East	0	0	0	0	1.8815
South	0.0159	0	0	0	0
West	0	0	3.8555	0	0

R1,3	Local	North	East	South	West
Local	0	0	0	0	1.8815
North	0	0	0	0.9463	0
East	0	0	0	0	0
South	0	0.0202	0	0	0
West	0.9463	0	0	2.9092	0

R2,0	Local	North	East	South	West
Local	0	0	0.8426	0.5069	0
North	0	0	0	0.0020	0
East	0.5983	0.1758	0	0.8475	0
South	0.8426	0	0	0	0
West	0	0	0	0	0

R2,1	Local	North	East	South	West
Local	0	0	0	0	1.3494
North	0	0	0	0.7032	0
East	0	0.1758	0	0	0.2721
South	0	0	0	0	0
West	0.8426	0	0	0	0

R2,2	Local	North	East	South	West
Local	0	0	0.9463	0	0.3515
North	0	0	0	0.1759	0
East	0	0	0	0	0.0963
South	0	0	0	0	0
West	0	0	0	0	0

R2,3	Local	North	East	South	West
Local	0	0	0	0	0.0963

North	0	0	0	0	0
East	0	0	0	0	0
South	0.0036	0.0166	0	0	0
West	0	0	0	0.9463	0

R3,0	Local	North	East	South	West
Local	0	0	0.6702	0	0
North	0	0	0	0	0
East	0	0	0	0.0020	0
South	0.1758	0	0	0	0
West	0	0	0	0	0

R3,1	Local	North	East	South	West
Local	0	0	0	0.7032	0
North	0	0	0	0	0
East	0	0	0	0	0.0020
South	0.1758	0	0	0	0
West	0.6702	0	0	0	0

R3,2	Local	North	East	South	West
Local	0	0	0	0.1759	0
North	0	0	0	0	0
East	0.7036	0	0	0	0.0020
South	0	0	0	0	0
West	0	0	0	0	0

R3,3	Local	North	East	South	West
Local	0	0	0	0	0.7056
North	0	0	0	0	0
East	0	0	0	0	0
South	0.0166	0	0	0	0
West	0	0	0	0	0

The detailed traffic flows of MMS traffic pattern are all listed in Table 7.2, whose traffic distribution is not symmetry. The traffic flows list the normalized results, too. From the table, we can know that the internal flows of each router in MMS pattern are more uneven than in uniform pattern.

At first, the injection flow and the drain flow of each PE are different. The flows of each channel and the transport flows of each branch are very different. The local port/channel of R(1,0) is the busiest output port/channel, which is 4.27 times injection rate. The north channel of R(0,3) west channel of R(1,2) and west channel of R(1,3) is the busiest input channel, which is 3.8555 times injection rate. The branch from the north port to the local port and the branch from the west port to the east port in R(1,2) have the most transport flows, which is 3.8555 times injection rate. It means that the traffic flow distribution of MMS pattern is very uneven in the network.

There is only 12 PEs send message to the network. A source PE has only one destination PE under matrix-transpose traffic pattern. The traffic flow distribution is

simply, which is shown in Figure 7.1. In this case, the bottleneck is two corners. Six routers' eight input or output channels have traffic flow of 4 times injection rate in the network.

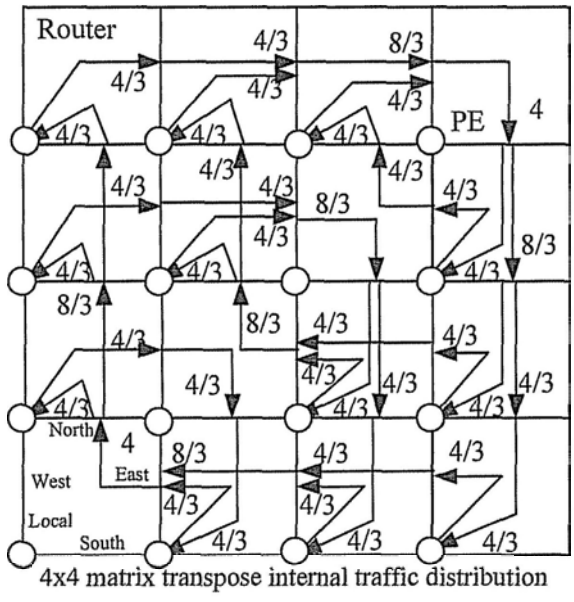


Figure 7.1 The internal traffic distribution under 4x4 matrix-transpose traffic pattern

7.2 Short-circuit crossbar channel

According to the analysis of internal traffic flows, we can know that there are some hot branches, which consume the most part of output channel's bandwidth, under some applications. If these hot branches can be optimized, the packets through these branches can ease the blocking problem, which should benefit network performance. We adds a special internal channel overflying the crossbar to deliver packets from an input to an output, which is referred as short-circuit crossbar channel. In a lookahead bypass router, the short-circuit crossbar channel can reduce the usage of crossbar and buffer, which can reduce the power consumption. Figure 7.2 presents an example of lookahead bypass router with short-circuit crossbar channel, where the dashed line denotes the optional short-circuit part. [55]

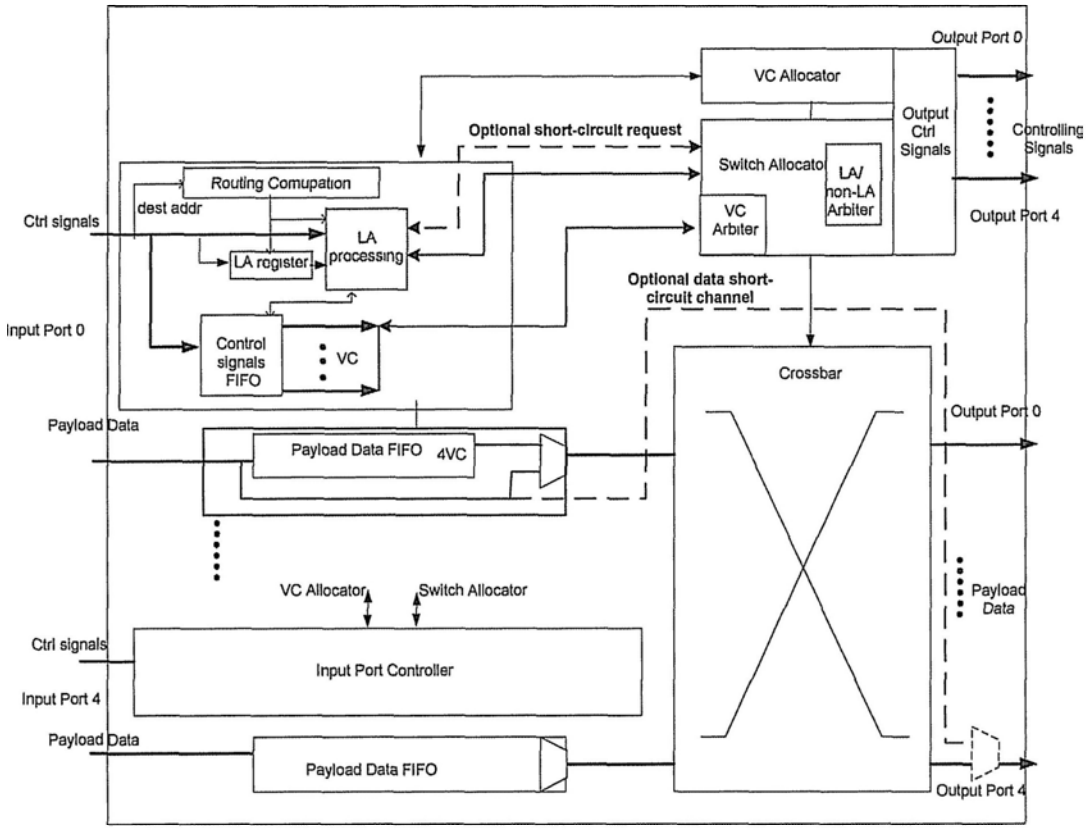


Figure 7.2 An example of lookahead bypass router with short-circuit crossbar channel

The modified controller is shown in Figure 7.3. Besides data short-circuit channels for payload, additional arbiter and control short-circuit channels are introduced for requests, which is referred as the short-circuit request. Only a nor2 gate delay is added into timing critical path, which is about 0.1ns. The overhead is little enough.

A request, whose corresponding payload goes through the crossbar, is referred as the crossbar request. The short-circuit request has higher priority than the crossbar request. Because short-circuit requests do not arbiter with crossbar requests, an input port can submit two valid requests, respectively, from input control buffer and lookahead control register at same time. If the two requests are submitted to two different output ports, they can reserve two output resources at same time. It means that the input port can deliver two flits' payload respectively through the crossbar (for a crossbar request) and the short-circuit crossbar channel (for a short-circuit request) in next cycle. The momentary bandwidth of this input channel can be double.

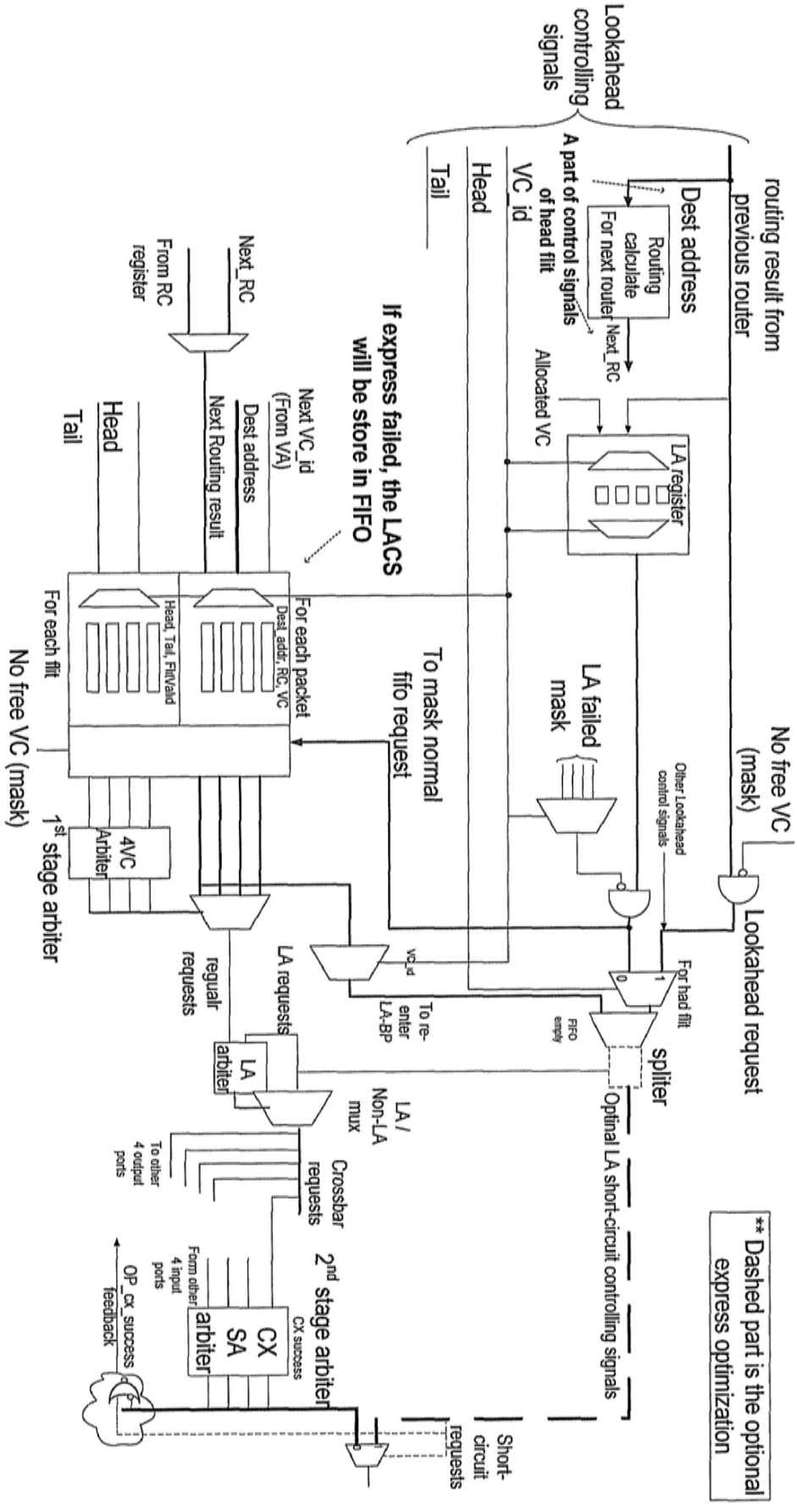


Figure 7.3 The lookahead controller with short-circuit channel

7.3 Allocation and Evaluation

To simplify the structure, any one output port can only connect to a short-circuit channel from an input port. It can avoid the use of an arbiter among short-circuit requests.

The allocation of short-circuit channel (SC) is based on the analysis of internal traffic distribution of all branches beforehand. The application-specific optimization is necessary to match short-circuit paths with the detailed traffic distribution.

The analysis is referred as $\boldsymbol{B_{m,n}}$, which denotes the bandwidth of branch from input $\boldsymbol{m}$ to output $\boldsymbol{n}$. LB_n is the maximum of all branch to input n, and M_n is the branch order:

$$\text{Given } (LB_n, M_n) = \max_{m=1}^{m \le p_i} \left(B_{m,n}\right) \text{ , If } LB_n > B_{Th}, \text{ set up}$$

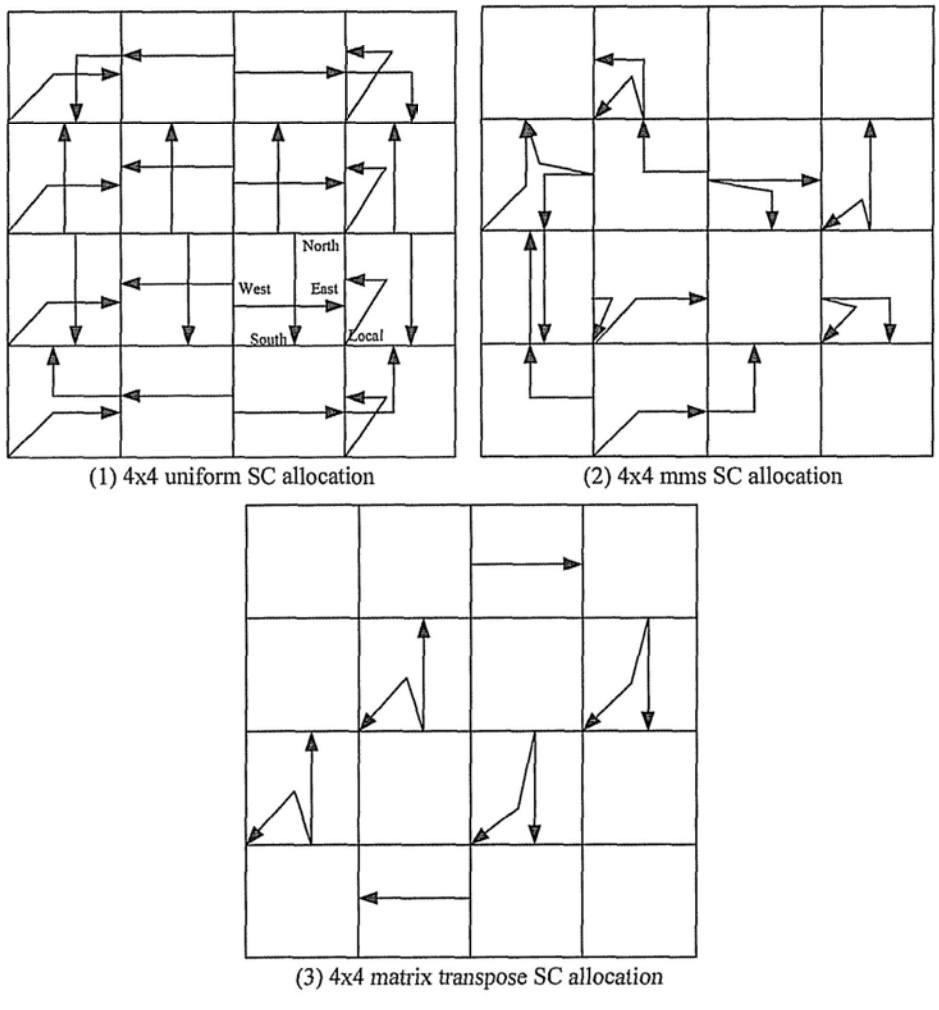
$$\text{a short-circuit channel } SC_n^{M_n} \text{ from input } Mn \text{ to output n} \quad . \quad <7.1>$$

The existence of threshold B_{Th} can avoid the improper SC impacts packets from other inputs, which account most part of bandwidth of this output. Here we define B_{Th} as: $B_{Th} = \left(\sum_{m=1}^{m \le p_i} B_{m,n}\right) \times 0.6$. It means that only if the bandwidth of branch with maximum flow accounts more than 60% bandwidth sum of all branch flow, the SC can be set up. In addition, it hardly improves latency if the maximum branch flow is sole from *m* port or to *n* port. Thus, this SC is optional according to your optimization target.

The example of short-circuit channel allocation is router(1,1) of uniform pattern in a 4x4 mesh network. 50% of the packets at the east input port are sent to the west output port, which account for 74% of the traffic flows of all packets to the west output port. Because more than half of the traffic flows at the west output port are from the east input port, SC should be provided for traffic from the east port to the west port. Another SC is implemented from the north port to the south port likewise. It is noted

that other output ports do not have any input, which can contribute more than half of the traffic. To these output ports, adding express path is not beneficial because too many express paths can work against network latency.

The internal traffic distributions of three patterns have different features. Uniform has uniform spatial bandwidth distribution, shown in Figure 2.16. Its spatial port/channel bandwidth distribution is near even. Although some routers deliver fewer packets because of empty port connections at the network margin, all routers have no any overlarge flow paths.



(1) 4x4 uniform SC allocation

(2) 4x4 mms SC allocation

(3) 4x4 matrix transpose SC allocation

Figure 7.4 Application-specific short-circuit allocation

The spatial bandwidth and injection distributions of MMS trace are very unbalanced.

The analysis of internal traffic distribution of routers presents that the difference of flows between different internal paths to an output may be very large. For example, in router(1,0), north, south, and west input channels respectively have 0.5118, 3.7535, and 0.0049 traffic flow to local port. The flow of path from south to local occupies 88% bandwidth of local and is 7.33 and 766 times as the flow from north and west.

The number of matrix transpose pattern's traffic flow is small because the destination of packets is oneflod. There are scarcely any output ports with more than two internal flows. And the bottleneck in router(0,0) or router (3,3) is one single path, which can not be resolved by adding some SCs.

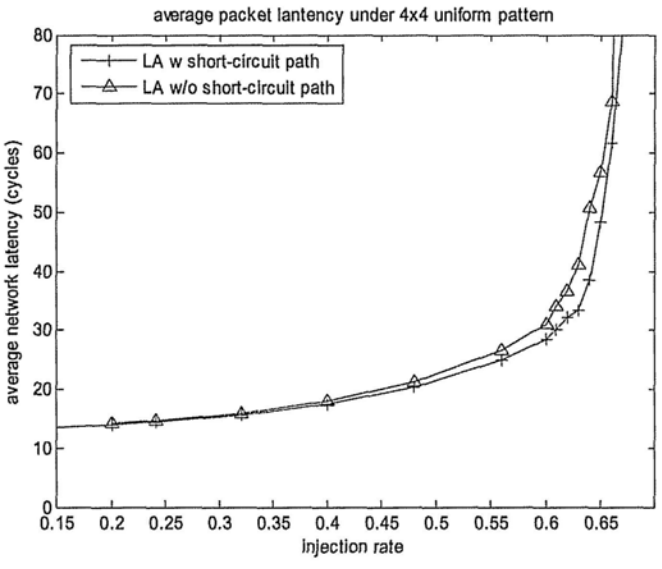
Figure 7.4 presents the detailed allocation of short-circuit channel for three traffic patterns according to the allocation method described above. To evaluate the performance of application-specific short-circuit allocation, the evaluation platform applies the parameter of Table 7.3.

Table 7.3 Basic network parameters of evaluations

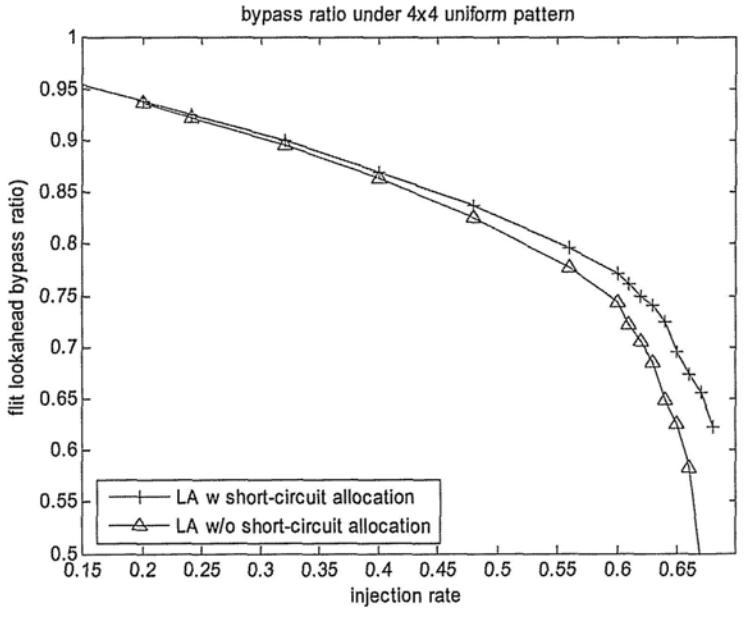
Topology	4x4 mesh network
Flow control	Virtual channel
Buffer management	Credit-based flow control
Routing algorithm	X-Y
Pipeline	Lookahead pipeline
Router radix	5
Buffer architecture	4 VCs per port, 8 flits per VC)
Packet length	4 flits
Flits size	128 bits (payload) + control signals

For those traffic patterns similar to uniform traffic, their traffic flows are symmetric and busy everywhere. It means that many bottleneck ports spread over the network when the global throughput is close to saturation. The application-specific short-circuit path can

hardly ease the global congestions because the physical resources in most will have been exhausted. Figure 7.5 shows the simulation result of average packet latency and bypass ratio.



(a)



(b)

Figure 7.5 (a) Average packet latency, and (b) bypass ratio of short-circuit improvement when traffic pattern is uniform.

The introduction of short-circuit can reduce latency and bypass ratio with the incensement of network load. When injection rate is 0.6 flit/cycle*node, the latency

reduces 8.76% (2.72 cycle) and bypass ratio increases from 0.744 to 0.771. The performance verging saturation throughput is improved more. When injection rate is 0.64 flit/cycle*node, the latency reduces 23.8% (12.1 cycle). Of course, the valid saturation throughput, which means the latency is acceptable under the throughput, also increase a little, about from 0.66 to 0.67 flit/cycle*node.

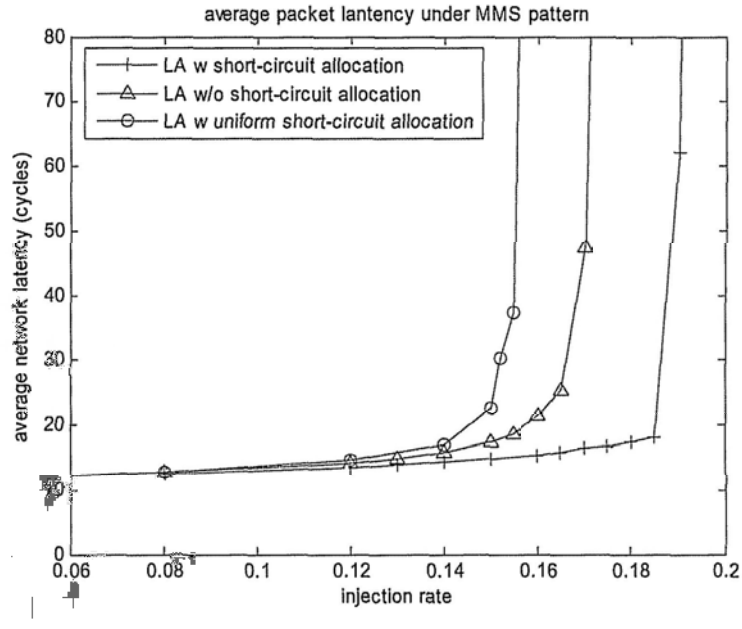
It is noted that traffics that are similar to MMS traffic pattern are different from uniform traffic. Traffic flows will be mainly distributed between several pairs of processing elements. Then only a few ports will become bottlenecks in the context of the whole network. Even if these ports are saturated, there are still spare physical resources in other ports.

Short-circuit channel optimization likes flyovers that can increase the bypass ratio to the hot branches and keep the traffic flows to the free branches. Especially when throughput is on the verge of saturation under MMS traffic pattern, the short-circuit channels can make better use of redundant resources by double momentary bandwidth. The simulation result is shown in Figure 7.6.

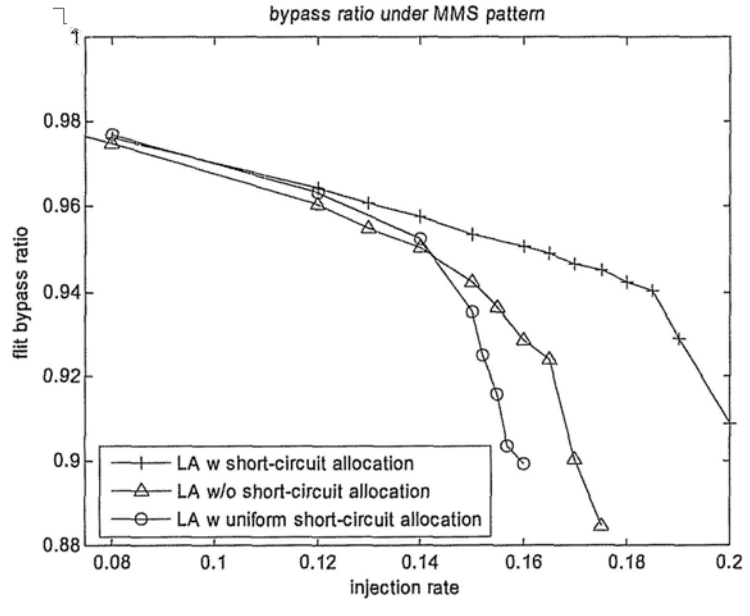
The valid saturation throughput of routers with the short-circuit optimization can be improved by 11% from about 0.17 to 0.19 flit/node*cycle. And when the network is moderate (0.15 flit/node*cycle), the latency still can be improved by up to 15.6% because the customized short-circuit channels can provide additional data channel to other free ports for the bottleneck ports.

It is mentioned above that network performance is not always improved with more short-circuit channels. More short-circuit channels can increase the total bypass ratio under low network load and can increase short-circuit bypass ratio. However, an inappropriate short-circuit channel will impact traffics from other input ports and will increase latency uncunningly.

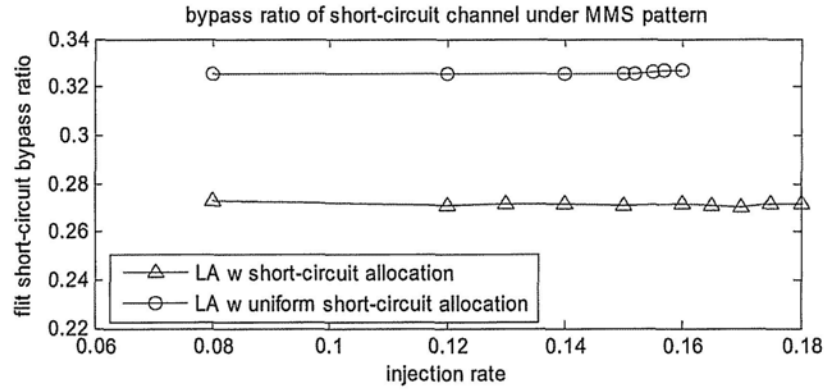
As shown in Figure 7.7 (b) and (c), the short-circuit optimized network-on-chip, which has wrongly customized for uniform but more short-circuit channels, can provide poorer latency and saturation throughput than even generic lookahead routers.



(a)



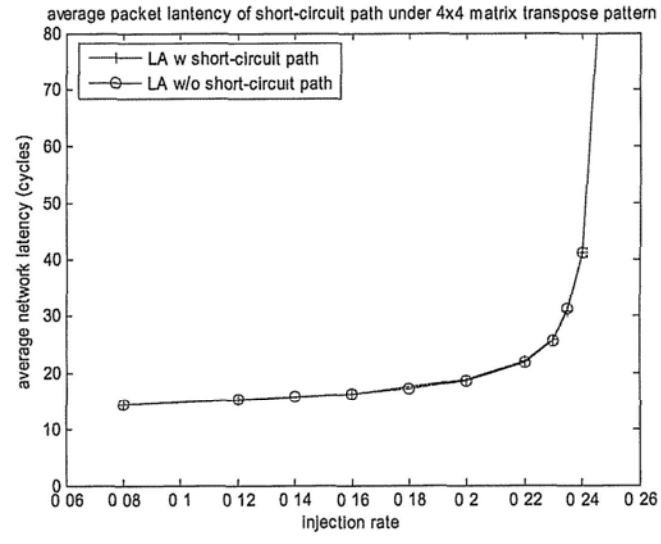
(b)



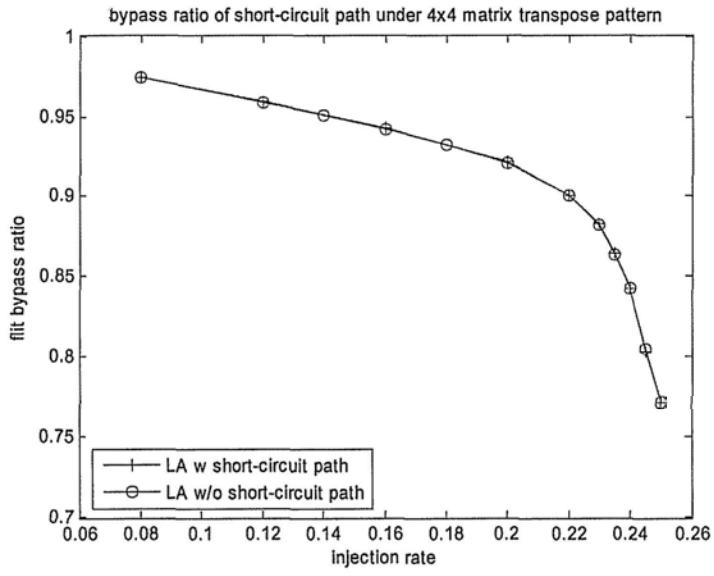
(c)

Figure 7.6 (a) Average packet latency, and (b) lookahead bypass ratio (c) bypass ratio of short-circuit improvement when traffic pattern is MMS.

At last, we research the type of traffic patterns similar to matrix transpose. The traffic flows focus on some branches. There are only a few ports will become bottlenecks in the context of the whole network, such as MMS traces. However, the bottleneck routers have a small number of traffic branches. There is only one traffic branch in R(0,0) and R(3,3) although the two routers have the largest bandwidth of traffic branch. The short-circuit optimization cannot play a role in this case. The curves of generic lookahead router and short-circuit lookahead router always nearly overlap, shown in Figure 7.7. There is only 1.4% (0.45 cycle) reduction in latency and 0.02 increment in bypass ratio when injection rate is 0.235 flit/cycle*node.



(a)



(b)

Figure 7.7 Average packet latency, and (b) bypass ratio of short-circuit improvement when traffic pattern is matrix transpose.

Thus, the optimized allocation of short-circuit channels must be done in according to the expected traffic pattern. This asks for detailed traffic flows to be evaluated early. And in networks supporting multiple applications, the short-circuit channel optimization is required to fit the traffic flows of all applications. It means only the common short-circuit channel can be set up in the network. Otherwise, it can be shut down or to be reconfigured.

7.4 Summary

The bandwidth of internal traffic flows is unbalanced. Optimizing the most of the flows through busy branch can improve the network performance. Short-circuit crossbar channel optimization only costs a little overhead in timing critical path and silicon area to provide the improvement in port-to-port bandwidth by the additional data path.

Evaluation results prove that a good short-circuit channel allocation can reduce the

average packet latency and increase bypass ratio. Thus, it also reduces the power consumption by the increasing bypass ratio and the bypass of crossbar.

Of course, more short-circuit crossbar channel means more flits can bypass crossbar. However, a poor short-circuit channel allocation impacts the saturation through and increase average packet latency under the high network load. Application-specific optimization is necessary in short-circuit crossbar channel allocation.

CHAPTER 8. Conclusion

8.1 Contributions

This thesis focuses on the design and implementation of low-latency network-on-chip. To reduce average packet latency, the major method is optimizing the router architecture because routers cost much more latency, power, and area than links in most on-chip networks. This thesis investigates and proposes several improvements of low-latency network-on-chip according to different requirements.

A flow design is proposed and the relevant synthesizable NoC library is implemented at first. An evaluation platform is used to obtain the network performance in different cases by various traffic models and router models of our NoC library.

Although asynchronous circuit is not wildly used and not well supported, asynchronous network-on-chip has advantage in cost. Because of the characteristic of asynchronous circuit, reducing the latency of any one stage can reduce the total latency of a flit. There is a trade-off with the variation of buffer size because increasing asynchronous FIFO would increase minimum buffer propagation latency and improve the flow control efficiency.

Application-specific asynchronous FIFO allocation algorithm can balance the minimum propagation latency and the efficiency under high network load. It means less area cost of buffer can obtain the better or equivalent performance if the buffer size of each port is allocated according to the detailed application specification. The

evaluation results prove the effect of application-specific asynchronous FIFO allocation algorithm under UMC 0.13um COMS technology.

Broadcast/multicast is an inherent disadvantage for network-on-chip over for bus architecture, which is wildly used in many applications, such as parallel computation and multi-processor applications. Although software method can implement multicast, the latency overhead is too large. Our hardware multicast design realizes the fast broadcast/multicast. It is based on quality-of-service routers with the additional group-cast routing and arbitration modules. It can play a great role if there are many multicast packets in system.

Moreover, novel synchronous router architecture is realized for low-latency target, which is referred as lookahead bypass router. The micro-architecture of lookahead bypass router is optimized to provide two-cycle minimum router propagation latency. High bypass ratio of buffer can improve flow-control efficiency for low latency under high network load. At the same time, it can abatement the buffer size to cut down silicon area and power consumption. The implementation with UMC 0.13um COMS technology is used to evaluate the performance. The simulation results prove the advantage in latency. And according to many results of various parameter configurations, the preferable one can be provided.

In addition, short-circuit crossbar channel optimization, which is based on the lookahead bypass router can provide better network performance in some applications. The optimization is a sort of low-latency implementation that has been discussed in the beginning of this thesis. The port-to-port bandwidth can be better utilized by the

additional short-circuit crossbar channels. Increasing the momentary port bandwidth can make up the loss in efficiency because of allocation and conflict. However, the too many casually-allocated short-circuit crossbar channels can be adverse because the short-circuit packets can block too much crossbar packets. Thus, application-specific analysis must be made to allocate short-circuit crossbar channels according to the traffic flow distribution.

8.2 Future works

There are many interesting topics for further research inspired by the work described in this thesis.

- Since the advantage of asynchronous network is great, we can build better asynchronous router architecture. 2-phase bundled-data communication protocol can be realized in the router to cut down the redundancy time of 4-phase bundled-data handshake. And more flow control protocols can be introduced to provide higher efficiency.
- There are tradeoffs between flexibility and setup time of multicast path, between flexibility and deadlock limit. We can research multicast strategy to improve the network performance for multi-processor application. Combining bypass scheme and other switching technology is an interesting research topic. Although packet switching is wildly used in network, other switching technologies have their advantage in latency for some applications. Hardware multicast can combine circuit switching to reserve the crossbar path for multicast packets.

- Our lookahead bypass router can cut down the cost of power consumption and silicon area. Further research of low-power and low-area lookahead router of various flow controls is necessary since the low power is in great demand by numerous mobile devices. Customized design of crossbar and FIFO can further reduce the power consumption.
- Short-circuit crossbar channel is an attempt to combine other bypass schemes to our lookahead bypass router. More optimizations can be implemented to increase the range of bypass and reduce latency, such as express paths between input channel and output channel or customized physical channels.

APPENDIX A. Asynchronous Communication Protocol and Basic Asynchronous Circuit

Asynchronous circuits require handshake signal to synchronize two components. The two wires per bit used in dual-rail protocols can be seen as a one-hot encoding of that bit and often it is useful to extend to 1-of-n encodings in control logic and higher-radix data encodings. If the focus is on communication rather than the computation, m-of-n encoding may be of relevant. The solution space can be expressed as the cross product of a number of options including [7]:

{2-phase, 4-phase} x {bundled-data, dual-rail, 1-of-n, m-of-n, ...} x {push, pull}

The choice of protocol affects the circuit implementation characteristics, such as area, speed, power, robustness, etc. Figure A.1 presents two protocols used in this thesis, which are 2-phase and 4-phase bundled-data push protocols.

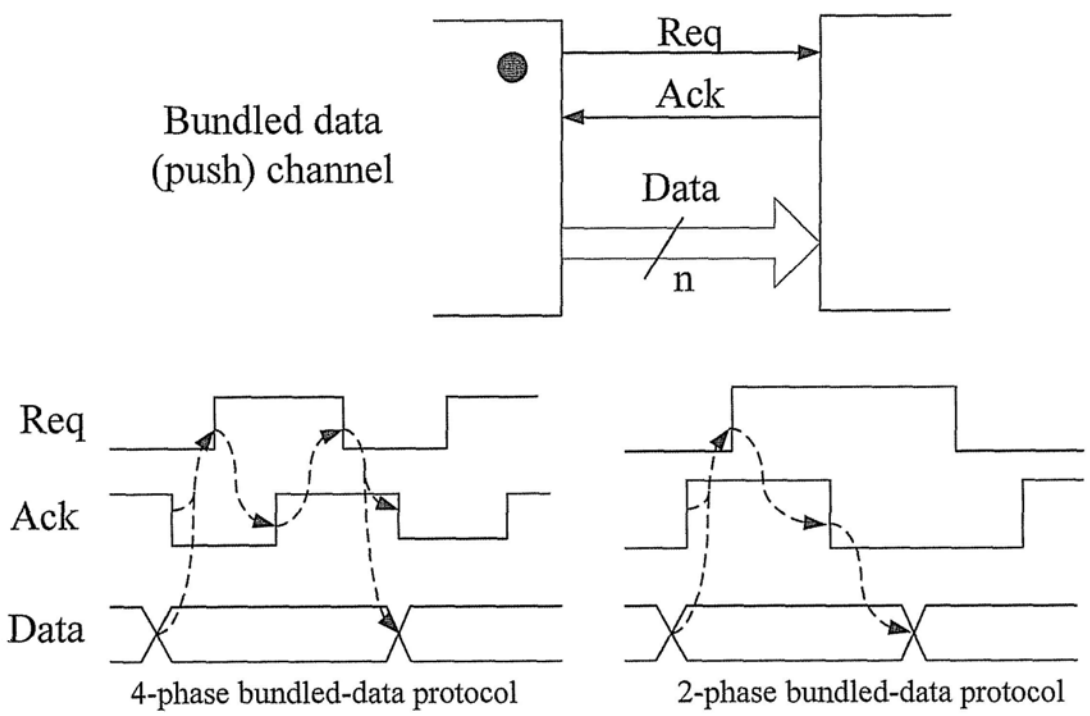


Figure A.1 Asynchronous communication protocol

The COMS-level and gate-level implementations of several basic asynchronous components are shown in Figure A.2, Figure A.3, and Figure A.4. It includes C-element, MUTEX, and muller C-element.

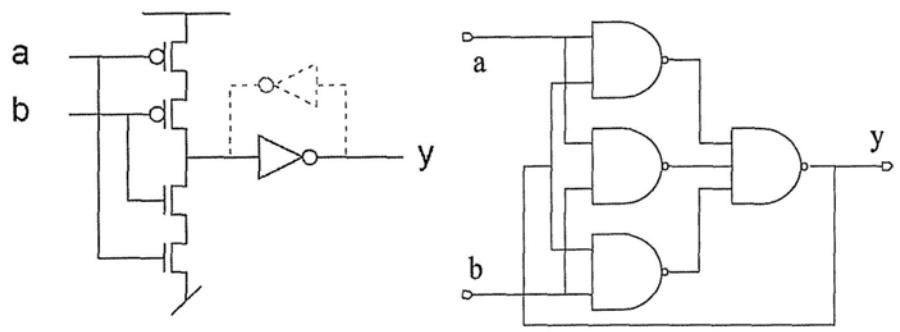


Figure A.2 The implementation of C-element

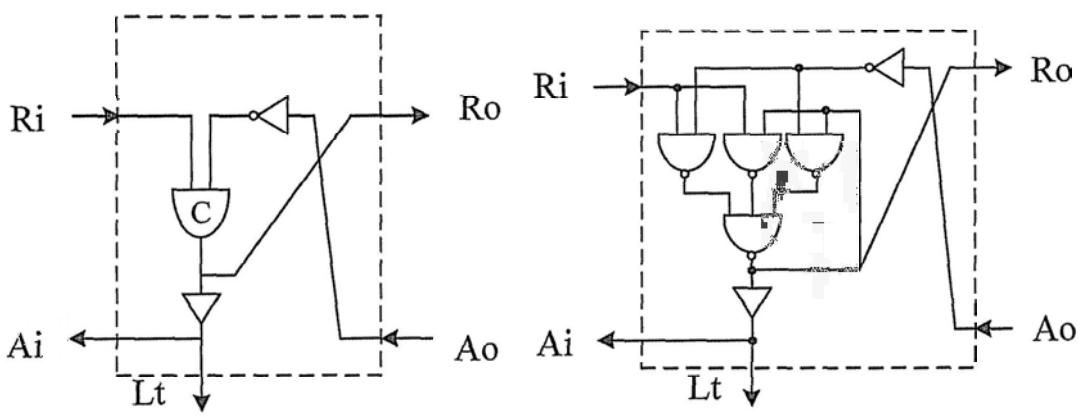


Figure A.3 The implementation example of muller C-element for micropipeline. Lt is a local clock to control a latch

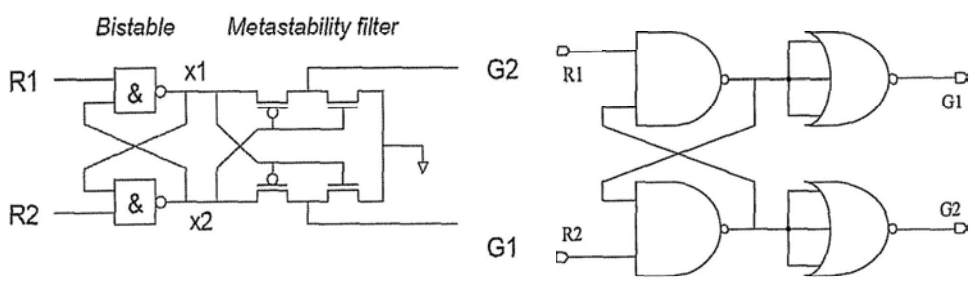


Figure A.4 The implementation of MUTEX

Although the COMS-level performance is better than gate-level performance, the gate-level implementation can be easily realized by the standard gates of digital library.

In this thesis, we use UMC 0.13um technology library. The results are good enough.

- C-element (0.13um load: invx1): Tpr ≈ 0.063ns Tpf ≈ 0.072ns
- MUTEX : No conflicting case: Tp ≈ 0.113ns

  Conflicting case: Tp ≈ 0.202ns

The router requires arbitration in allocation, which can be realized by MUTEX cell. Figure A.5 presents a possible implement of asynchronous 2-input arbiter by MUTEX cell and standard gates.

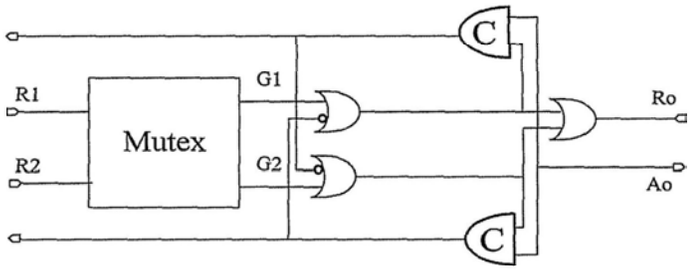
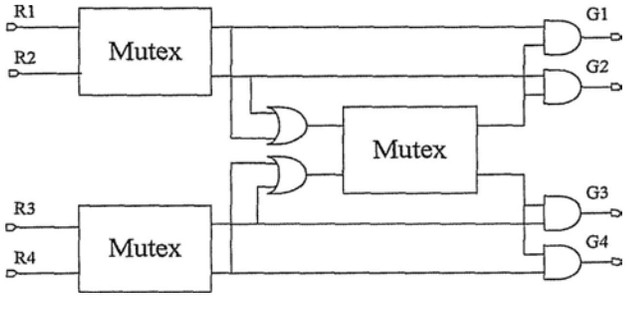
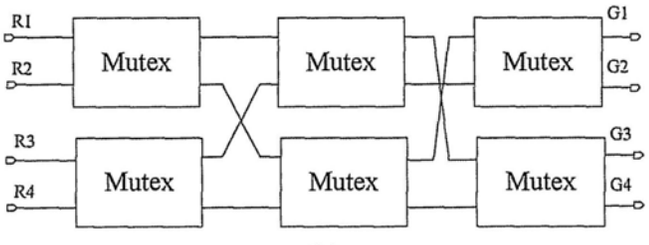


Figure A.5 The asynchronous arbiter



(a)



(b)

Figure A.6 The request part of asynchronous 4-input arbiter

In NoC, there are many n-input arbiters, such as the arbitration of multiple input requests. If network applies mesh topology, the router requires 4-input arbiters. Two implementations of cascaded tree arbiter and multi-way MUTEX are shown in Figure

A.6 [62, 63].

Cascaded multi-way arbiters are based on a tree topology, in which the front end requests arbitrate in adjacent pairs, and then new requests are generated on their behalf. The new requests propagate to the next level of the tree and arbitrate with their neighbor in the same fashion.

A two-way MUTEX is also used for constructing arbitrating combinations on the 2-of-n basis. It is speed-independent. For n=4 one would need 6 two-way MUTEXes.

APPENDIX B. A Reconfigurable Synthesizable NoCs Library

A reconfigurable synthesizable NoCs library is referred as NoCLib. It includes evaluation platform, synchronous router model and asynchronous router model. The evaluation platform constructs a network topology, generates test traffic pattern, and controls and checks the injection and drain of packets. Figure B.1 shows the usage of NoCLib.

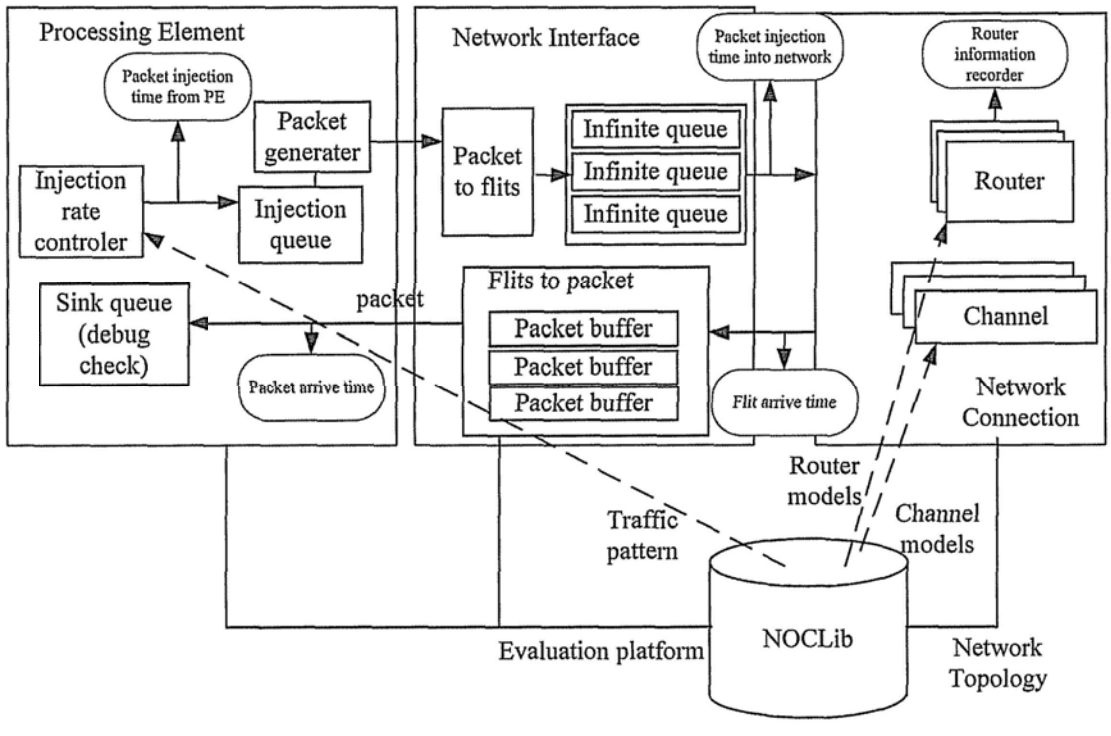


Figure B.1 The usage of NoCs library

- NoCLib supports a wide range of pipeline architectures of synchronous router. It includes synchronous 5-stage-pipeline router, speculative-pipeline router, no-load bypass pipeline router, and lookahead-pipeline router. In additional, asynchronous micro-pipeline architecture is implemented in NOCLib.

- It supports a wide range of customized configurations, which is realized by the global parameters listed in Table B.1. It is flexible to evaluate the network performance of different configurations.

Table B.1 Reconfigurable parameter in NoCs library

Parameter	Description
Network topology parameter	
NETWORK_X	The number of rows in a mesh network
X_ADDR_BITS	The number of bits used to describe routing information of X dimension. A 4x8 mesh network requires 2 bits to presents the poison of X dimension.
NETWORK_Y	The number of columns in a mesh network
Y_ADDR_BITS	The number of bits used to describe routing information of Y dimension. A 4x8 mesh network requires 3 bits to presents the poison of Y dimension.
Link parameter	
CHANNEL_DATA_WIDTH	Width of payload data in a flit. The width of channel is the sum of width of payload data and width of all control signals that is determined by other parameter
channel_num_reg	The number of pipeline stage of link between routers. If the link is just the wire connection, the parameter is one.
channel_latency	In synchronous router, the value should cloud be 0, which means that we assure the latency is less than the clock period,. For asynchronous network, the value denotes the latency of the link
Basic router parameter	
ROUTER_RADIX	Specifies the maximum number of router ports (input/output bi-direction port). At most five ports are supported for mesh network.
ROUTE_NUM_REQUEST	Specifies the maximum number of requests used in the router. If the U-turn transmission is forbidden, the value usually is **ROUTER_RADIX** minus one.
ROUTE_REQUEST_BITS	The number of bits used to describe the maximum number of requests. It is also the

	width of routing result.
ROUTER_NUM_VCS	Specifies the maximum number of virtual channel (including both unicast and multicast version) in router.
VC_INDEX_BITS	The number of bits used to describe the maximum number of virtual channel
FIFO_DEPTH	Specifies uniform depth of input buffer. If application-specific allocation is used, it specifies the maximum depth. The detailed depth of buffer of each port determined by an additional parameter PORT_fifo_detph
PORT_fifo_detph	Describes the depth of buffer of each port
FIFO_BITS	The number of bits used to describe the uniform depth of input buffer.
LOCAL	The number denotes local port in mesh router
NORTH	The number denotes north port in mesh router
EAST	The number denotes east port in mesh router
SOUTH	The number denotes south port in mesh router
WEST	The number denotes west port in mesh router
ASYN_DELAY_TABLE	The pre-computed delay of critical components in asynchronous router is used for behavior netlist. This is not usually used because the accuracy.
Customized configuration parameter	
LA_ENABLE	If define **LA_ENABLE,** lookahead bypass scheme is supported in router.
EX_ENABLE	If define **EX_ENABLE**, the short-circuit crossbar channel is supported. The detailed allocation is determined by IP_express_path_config and OP_express_path_config.
IP_express_path_config	To set up the short-circuit channel
OP_express_path_config	To set up the short-circuit channel
COMB_CREDIT_ENABLE	If define it, lookahead bypass router will combine credit information with destination address and lookahead routing result.
CREDIT_FLOW_CONTROL	If defined, it denotes the router support credit flow control. Otherwise, full-stop flow control is adopted.
QOS	Enable QoS. The virtual channel allocation is ignored when **QOS** enables. The SL bits embedded in flit determines which virtual channel is used.
MULTICAST	Use multicast support. It requires switch on

	QOS option. The number of service level is determined by **ROUTER_NUM_VCS**.
MULTICAST_NUM_VCS	It determines how much virtual channels are used for multicast. It must be less than **ROUTER_NUM_VCS.**
GRLT_FILE	The default content of GRLT
Test and Data format parameter	
MAX_PACKET_SIZE	Specifies size (in flits) of packets. Currently, only fixed-length packets are supported.
PACKET_INDEX_BITS	The number of bits used to describe the size of packets.
DEBUG	If **DEBUG** is defined, the payload data is embedded a set of debug information. The width of payload is the sum of debug part and **CHANNEL_DATA_WIDTH.**
INJ_RATE_NUM	The base of injection rate. The injection rate of each PE is the product of the base and their value from spatial injection distribution file
LOOKUP_TABLE_FILE	The spatial bandwidth distribution file
INJ_RATE_TABLE_FILE	The spatial injection distribution file
MEASUREMENT_P	The minimum number of packets used to measure the network performance.

REFERENCES

1. "Networks on chip", A.Jantsch, H.Tenhunen, Kluwer Academic Publishers, 2003

2. Willam James Dally, Brian Towles, "Principles and practices of interconnection networks", Morgan Kaufmanm Publishers, 2004

3. Luca Benini and Giovanni De Micheli, "Networks on Chips: Technology and Tools", Morgan Kaufmanm Publishers, 2006

4. James Irvine and David Harle, "Data Communications and Networks: An Engineering Approach", John Wiley & Sons Publishers, 2002.

5. Jos´ e Duato, Sudhakar Yalamanchili, Lionel M. Ni, "Interconnection Networks: An Engineering Approach", John Wiley & Sons Publishers, 2003.

6. Nancy A.Lynch, etc., "Distributed Algorithms", Morgan Kaufmanm Publishers, 1996

7. Jens Sparso and Steve Furber, "Principles of Asynchronous Circuit Design", Kluwer Academic Publishers, 2001

8. C. Piguet, "Low-power Electronics Design", CRC Press, 2005

9. Lawence Landweber, "Network routing: algorithms, protocols, and architectures", Morgan Kaufmanm Publishers, 2007

10. U.Y.Ogras and R.Marculescu, "'It's a small world after all': NoC performance optimization via long-rang link insertion",IEEE trans. VLSI syst., Vol.14, no.1,pp.693-706, Jul.2006

11. Chrysostomos Nicopoulos, Vijaykrishnan Narayanan, Chita R Das, "Network-on-

Chip Architectures: A Holistic Design Exploration", Springer Publishers, 2009

12. Ahmed Hemani, Axel Jantsch, Shashi Kumar, Adam Postula, Johnny ¨Oberg, Mikael Millberg, and Dan Lindqvist. Network on chip: An architecture for billion transistor era. In Proceeding of the IEEE NorChip Conference, November 2000

13. Luca Benini and Giovanni De Micheti, " Network on Chips: A New SoC Paradigm" , IEEE Computer, Jan. 2002, pages 70-78

14. M. Sgroi et. al., "Addressing the System-on-a-chip Interconnect woes through communication based design", Proc. of the Design Automation Conference, June 2001

15. D. Wingard, "MicroNetwork based integration for SoCs" , Proc. of the Design Automation Conference, June 2001

16. William J. Dally and Brian Towles, "Route Packets, Not Wires: On-Chip interconnection Networks" , Proc. of the Design Automation Conference, June 2001

17. G. Apostolopoulos et al. "Quality of service based routing: a performance perspective", In SIGCOMM '98, pages 17–28, 1998

18. E. Bolotin et al. "QNoC: QoS architecture and design process for network on chip", The Journal of Systems Architecture, Dec. 2003

19. R Thid, M Millberg, A Jantsch, "Evaluating NoC communication backbones with simulation", Proceedings of the IEEE Norchip, 2003

20. WJ Dally, "Virtual-channel flow control", IEEE Transactions on Parallel and Distributed systems, 1992, pp194-205

21. R Mullins, A West, S Moore, "Low-latency virtual-channel routers for on-chip networks", In Proceedings of the 31th ACM SIGARCH Computer Architecture, 2004

22. J. H. Kim and A. A. Chien, "Evaluation of wormhole-routed networks under hybrid traffic loads," Proceedings of the 26th Hawaii International Conference on System Sciences, pp. 276 - 285, January 1993.

23. [2.xx3] Erland Nilsson, "Design and Analysis of a hot potato switch in Network on Chip", Master thesis, Laboratory of Electronics and Computer Systems, Department of Microelectronics and Information Technology, Royal Institute of Technology, Stockholm, July 2002.

24. [2.xx4] Chrysos, M. Katevenis: "Weighted Fairness in Buffered Crossbar Scheduling", Proceedings of the IEEE Workshop on High Performance Switching and Routing (HPSR 2003), Torino, Italy, June 2003, pp. 17-22

25. Melanie L. Fulgham, Lawrence Snyder, "A comparison of input and output driven routers", on Euro-Par'96 Parallel Processing, Lyon,France,1996, pp195-204

26. J Hu, R Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints", In Proceedings of Design Automation Conference, 2003

27. Jingcao Hu; Ogras, U.Y.; Marculescu, R, "System-Level Buffer Allocation for Application-Specific Networks-on-Chip Router Design", ICCAD 2006, vol.5, pp.2919 – 2933, 2006

28. Gratz, P.; Changkyu Kim; McDonald, R.; Keckler, S.W.; Burger, D "Implementation and Evaluation of On-Chip Network Architectures", on ICCD 2006, pp.477-484.

29. Vassos Soteriou, Hangsheng Wang, and Li-Shiuan Peh, "A Statistical Traffic

Model for On-Chip Interconnection Networks",on14th IEEE International Symposium on Modeling, Analysis, and Simulation, Monterey, 2006

30. A Kumar, LS Peh, P Kundu, NK Jha, "Express virtual channels: Towards the ideal interconnection fabric", Proceedings of the 34th ACM SIGARCH Computer Architecture, 2007

31. Li-Shiuan Peh, William J. Dally , "A Delay Model and Speculative Architecture for Pipelined Routers", In Proceedings of the 7th International Symposium on High-Performance Computer Architecture, Jan., 2001, pp. 255-266.

32. H.Zimmer, S.Zink, Hollstein., M.Glesner, "Buffer-Architecture Exploration for Routers in a Hierarchical Network-on-Chip", Parallel and Distributed Processing Symposium, 2005. Proceedings. Pp.4.

33. Jingcao Hu; Ogras, U.Y.; Marculescu, R., "System-Level Buffer Allocation for Application-Specific Networks-on-Chip Router Design", IEEE Trans Computer-Aided Design of Integrated Circuits and Systems,Vol. 25, Issue 12, Dec. 2006 Pp.:2919 – 2933

34. Edwin Rijpkema, Kees Goossens, and Paul Wielage, "A Router Architecture for Network on Silicon" , Proceedings of Progress 2001, Workshop on Embedded Systems.

35. Pierre Guerrier and Alain Greiner. "A generic architecture for onchip packet-switched interconnections", In Proceedings of Design, Automation and test in Europe, pages 250 - 256, 2000.

36. E. Rijpkema et al. "Trade offs in the design of a router with both guaranteed and

best-effort services for networks on chip", In DATE, 2003.

37. Mikael Millberg, Erland Nilsson, Rikard Thid, and Axel Jantsch. Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip. In Proceedings of the Design Automation and Test Europe Conference (DATE) 2004.

38. Zhonghai Lu, Rikard Thid, Mikael Millberg, Erland Nisson and Axel Jantsch, "NNSE: Nostrum Network-on-Chip simulaiton Enviroment", Swedish System-on-Chip, 2005

39. Santiago Gonzalez Pestana, Edwin Rijpkema, Andrei Radulescu, Kees Goossens and Om Prakash Gangwal, "Cost-Performance Trade-offs in Networks on Chip: A Simulation-Based Approach",In Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, 2004 (DATE'04)

40 A. Radulescu et al. An efficient on-chip network interface offering guaranteed services, shared-memory abstraction, and flexible network programming. In DATE, 2004.

41. L. Xin, C.S. Choy, "A Network-on-Chip simulation environment supporting asynchronous router", IEEE Proceedings on ASIC (ASICON) 2007.Sep.2007, 6-41

42. M. Forsell. "Advanced simulation environment for shared memory network-on-chips", In 20th IEEE Norchip Conference, 2002.

43 G. Varatkar et al. "Traffic analysis for on-chip networks design of multimedia applications", In DAC, 2002.

44. Youngbok Kim and Yong-Bin Kim, "An Asynchronous NoC Router Architecture

Supporting Quality-of-Service", Fifth Annual Boston Area Architecture Workshop, 2007

45. D. Rostislav, V. Vishnyakov, E. Friedman, R. Ginosar, "An asynchronous router for multiple service levels networks on chip" IEEE Proceedings on Asynchronous Circuits and Systems (ASYNC) , 2005, Pp.44 – 53

46. Jeremy Chan, Sri Parameswaran, "NoCEE: Engergy Macro-Model Extraction Methodology for Network on Chip Router", Computer-Aided Design, 2005. ICCAD-2005. IEEE/ACM International Conference, pp.254-259

47. R.P.Dick, "Embedded System Synthesis Benchmarks Suites (e3s). available on web: http://www.ece.northwestern.edu/~dickrp/e3s/

48. [3.10] V. Chandra, A. Xu, H. Schmit, L. Pileggi., "An interconnect channel design methodology for high performance integrated circuits" Proceedings on Design, Automation and Test in Europe Conference and Exhibition, 2004. Vol.2, Feb. 2004 Pp.:1138 - 1143

49. L. Xin, C.S. Choy, "A Network-on-Chip simulation environment supporting asynchronous router", IEEE Proceedings on ASIC (ASICON) 2007.Sep.2007, 6-41

50. [3.12] A.Agarwal, "Limits on interconnection network performance", IEEE Trans. Parallel Distributed System, Vol.2, no.4, pp.398-412, 1991

51 [3.13] V. Soteriou, Hangsheng Wang, L. Peh, "A Statistical Traffic Model for On-Chip Interconnection Networks", IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Sep.2006,Pp.104 – 116

52. L Xin, CS Choy, "An Asynchronous Router with Multicast Support in NoC", In Proceedings of the Fifth IASTED International Conference,2007

53. MPMalumbers, J.Duato, and J.Torrellas, An efficient implementation of tree-based multicat routing for distributed shard-memory multiprocessor, In proceding of the 8th IEEE Symposiuum on parallel and distributed Processing, 1996

54. John Byers, Michael Luby, Michael Mitzenmacher, A digital fountain approach to asynchronous reliable multicast, IEEE Journal on Selected Areas in Communications, 2002

55. L Xin, CS Choy, "A low-latency NoC router with lookahead controlling pipeline", In Proceedings of ISCAS 2010

56. X.Lin, P.K.McKinley, and L,M.Ni, Deadlock-free multicast wormhole routing in 2-D mesh multicomputers, IEEE Transactions on Parrallel and Distributed Systems, Aug.1994

57. M. Galles, "Scalable pipelined interconnect for distributed endpoint routing: The SGI SPIDER chip." in HotI, Aug. 1996.

58. P. Gratz, C. Kim, R. McDonald, S. W. Keckler, and D. Burger, "Implementation and evaluation of on-chip network architectures," in ICCD, Oct. 2006.

59. Amit Kumar, Partha Kundu, Arvind P. Singh, Li-Shiuan Peh and Niraj K. Jha, "A 4.6Tbits/s 3.6GHz Single-cycle NoC Router with a Novel Switch Allocator in 65nm CMOS"

60. R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," in ISCA, June 2004.

61. S. S. Mukherjee, P. Bannon, S. Lang, A. Spink, and D. Webb, "The Alpha 21364 network architecture," IEEE Micro, vol. 22, no. 1, pp. 26–35, Jan./Feb. 2002

62. Mitrani, A Yakovlev, "Tree arbiter with nearest-neighbour scheduling", in computer and information sciences' 98, pp.83-92

63. David J. Kinniment, "Synchronization and Arbitration in Digital Systems", John Wiley and Sons, Feb 2008