

The Research and Implementation of Router for Packet Connect Circuit Network-on-chip

Bochen Ye
Hefei University of Technology

Abstract—In the past decades, the scale of integrated circuits has increased rapidly, and billions of transistors can be integrated on a single silicon wafer. In the face of such a huge system on chip (SoC), the traditional bus structure has been difficult to meet the communication needs. In this case, the network-on-chip technology with high scalability and low power consumption is born, and the switching mechanism is crucial for the network-on-chip.

In the traditional wormhole switching mechanism, each node has high latency and may generate deadlock, but the packet connect circuit protocol can lock the link to send data once the link is successfully established, and each node only needs one cycle delay, which is suitable for high bandwidth and low latency communication without generating deadlock, so it can fully meet the needs of multi-core or multicore on-chip networks. It is suitable for high bandwidth and low latency communication without deadlock, so it can fully meet the data transmission requirements of multi-core or multicore on-chip networks in real time.

This thesis designs a single-node router for on-chip networks based on packet connect circuit mechanism, using Verilog to design input and output state machines, arbitration modules, address decoding modules, priority modules, and cross-switching modules. Two on-chip network platforms are constructed according to two different topologies, and packet sending and receiving simulations are performed on the on-chip network to verify the platform. Finally, the on-chip network is verified at board level and implemented in hardware on a Xilinx Artix-7 series FPGA development board, which communicates with the PC side through the UART serial port, receives the serial data on the PC and adjusts the order to display it.

Index Terms—Network on Chip; Packet connect circuit; Routing; FPGA; Double Ring

I. INTRODUCTION

A. Motivation

Over the past few decades, the integrated circuit industry has developed rapidly. Due to the increasing demand for high performance, chips have evolved towards higher integration and lower power consumption. Traditional bus-based interconnect architectures cannot support simultaneous communication among multiple users, resulting in low utilization of time resources[1][2]. For multi-core or many-core systems, this performance level is far from adequate to meet current demands. Figure 1 illustrates the traditional bus architecture.

To address the communication issues among multiple cores, researchers have proposed a new architecture known as Network on Chip (NoC). The core idea of NoC is to introduce the concepts of computer network communication into chip design, replacing the traditional bus model with routing algorithms and switching techniques used in wide area networks.

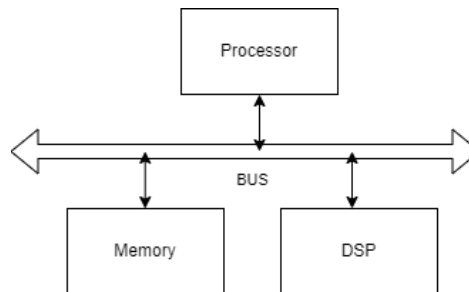


Fig. 1. traditional bus architecture

This approach addresses the shortcomings of the bus architecture's interconnect technology[3]. NoC offers high throughput, good spatial scalability, and stronger parallel communication capabilities, making it a novel communication method for System on Chip (SoC).

B. Related Work

Since the early 21st century, the design of multi-core systems based on network communication has been pioneered by internationally renowned research institutions such as the Royal Institute of Technology (KTH) in Sweden, Stanford University, and the Massachusetts Institute of Technology (MIT). In 2001, Hemani et al.[4] from KTH first proposed the concept of Network on Chip (NoC), which involves introducing the concept of computer communication networks into system-on-chip design. They used computer network design methodologies and routers to connect various modules on the chip for information exchange. The SoCBUS network on chip, designed by Linköping University in Sweden[5], employs the Packet Connected Circuit (PCC) protocol for switching[5]. The PCC protocol initially uses packet switching to send a head packet from the source node to the destination node. Once the link is successfully established, the path is locked, creating a dedicated circuit-switched channel for data transfer between the source and destination nodes. If the head packet fails to find a path, it will be retransmitted until successful.

In China, significant research achievements have been made in the field of NoC architecture and communication protocols by institutions such as Tsinghua University, Hefei University of Technology, Xidian University, the Institute of Microelectronics of the Chinese Academy of Sciences, Harbin Institute of Technology, and Beihang University. For wormhole switching, many current studies introduce virtual channels or

virtual networks. Zhang Zhe and colleagues from Beihang University[6] designed a configurable router for network on chip based on the 2D-Mesh structure. This router employs wormhole switching, incorporates virtual channels into the design, and provides configurable options, effectively mitigating network congestion and enhancing performance. Song Yukun et al. from Hefei University of Technology[7] proposed a packet-switched router supporting virtual circuit switching, integrating the concept of virtual channels. This meets the needs of short packet transmission, broadens the application range, and significantly improves system performance.

Research on NoC is a crucial field, with products becoming increasingly complex and diverse. Currently, NoC research faces three major challenges. Firstly, achieving low power consumption in NoC is critical. As the number of transistors increases, power consumption becomes a significant constraint, severely limiting overall performance improvements. Although recent progress in low-power research has been substantial, there remains a gap compared to the ideal interconnect power consumption[8]. Secondly, NoC needs to surpass traditional interconnects, including 3D stacked NoCs and optical interconnects. The transition to three-dimensional integration, whether moving from 130nm to 90nm or from 65nm to 28nm, merely postpones the demise of Moore’s Law. Therefore, a concept of “More than Moore” is needed, with 3D stacking being a promising technology[9]. Lastly, NoC fault tolerance is vital due to the potential for unexpected errors arising from the chip’s operational environment. To ensure the reliability of on-chip interconnects, fault tolerance research is ongoing, exploring various coding and architectural solutions. However, current primary fault tolerance research focuses mainly on fault-tolerant routing algorithms[10].

C. Structure and Content of the Thesis

This thesis focuses on the research and implementation of packet-switched network-on-chip routers. The structure of the thesis is as follows:

- 1) Introduction. This chapter introduces the research background of the design and the current state of research both domestically and internationally.
- 2) Fundamentals of Network on Chip. This chapter covers the basic knowledge of Network on Chip (NoC), including common NoC topologies, switching mechanisms, routing algorithms, arbitration algorithms, as well as issues such as deadlock and livelock.
- 3) Design of a Packet-Switched Network on Chip Router. This chapter presents the design of a packet-switched NoC single-node router, describing its basic structure. It also explains how to build a 2D-Mesh NoC platform and a dual-ring NoC platform using these single-node routers.
- 4) Experimental Verification of the NoC Platform. This chapter verifies the feasibility of the designed NoC platform through experiments. Various scenarios are considered, including single-flow transmission and reception experiments, two-flow cross transmission and reception

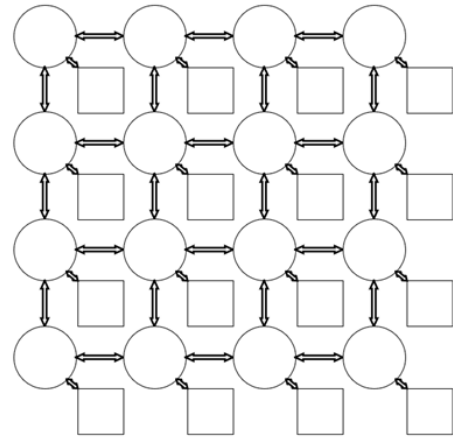


Fig. 2. 2D-Mesh Topology

experiments, and multi-flow conflict experiments. Finally, the entire packet-switched NoC is implemented on an FPGA, and data reception is verified via a PC to confirm the success of the implementation.

- 5) Summary and Outlook. This chapter summarizes the results presented in the thesis, analyzes and reflects on existing problems, and proposes improvement plans for future research.

II. BACKGROUND OF NOC

A. Network on Chip Topologies

Topology is a concept in mathematics that abstracts physical objects into points that have no relation to the shape, size, or attributes of the actual objects. The connections between these objects are abstracted into line segments connecting these points. This highly abstracted representation, using simple points and line segments to form a graph that only includes the positions of the objects, is a method used to facilitate the study of relationships among objects.

NoC topology refers to the interconnection methods among various resource points in a Network on Chip (NoC). It typically determines the network’s routing strategies, arbitration methods, and the distribution of IP modules[11]. Common NoC topologies include the following: 2D-Mesh Structure (2D-Mesh), Torus Structure, Fat-Tree Structure, Ring Structure. These topologies influence the overall performance, efficiency, and scalability of the NoC by defining how data is routed between different nodes and how resources are allocated and managed within the network.

1) *2D-Mesh Topology*: The 2D mesh structure is the most widely used and simplest form of structure. As shown in Figure 2, each routing node in this structure is connected to local resource nodes (IP cores), and except for edge nodes, each node is connected to nodes in four directions.

For the 2D mesh structure, its advantages lie in scalability and regularity, making it conducive to efficient wiring. Therefore, many studies on Network on Chip (NoC) are based on the 2D mesh structure. For instance, the Nostrum system

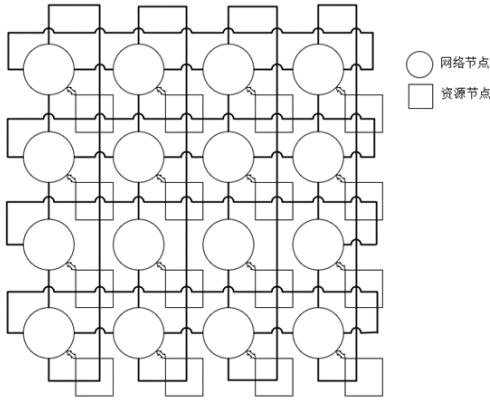


Fig. 3. Torus Topology

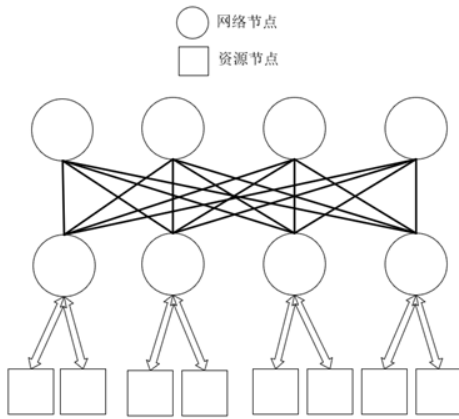


Fig. 4. Fat-Tree Topology

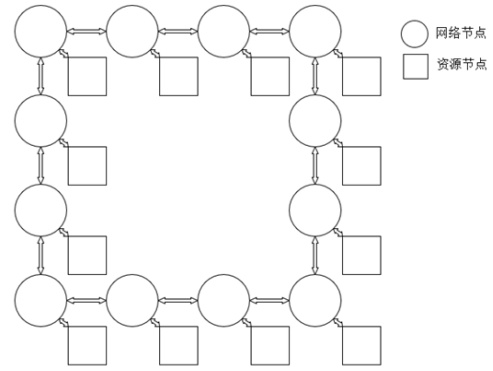


Fig. 5. Ring Topology

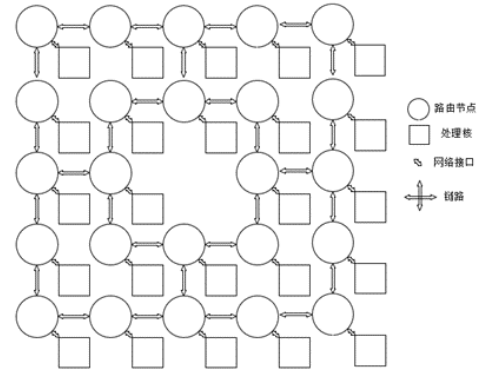


Fig. 6. Double Ring Topology

developed at the Royal Institute of Technology in Sweden utilizes a Mesh structure[12].

Therefore, the topology structure utilized in this study is the 2D mesh structure, with a node distribution of 4 rows by 6 columns, totaling 24 nodes.

2) *Torus Topology*: As the size of Mesh networks increases, both the network's topology diameter and average distance also significantly increase, thereby directly impacting network performance. The 2D torus structure is similar to the 2D mesh but with edge nodes connecting to each other in a wrap-around manner. Its structure is illustrated in Figure 3. The Torus structure offers more regular and symmetric routing allocation and internal structure, a smaller network diameter, reduced latency, and enhanced network communication capabilities[13].

3) *Fat-Tree Topology*: As shown in Figure 4, in a Fat-Tree structure, each parent node has multiple child nodes, and each node is connected to multiple nodes in the next stage. Each child node connects to multiple modules.

4) *Ring Topology*: The ring structure is also a simple topology where all routing nodes are connected sequentially in a circular manner. The advantages of a ring structure lie in its simplicity, small power consumption, and area efficiency,

while still maintaining certain performance capabilities [14]. As shown in Figure 5.

5) *Double Ring Topology*: As shown in Figure 7, the Double Ring (DR) structure is a special type of ring structure characterized by two concentric rings. These two rings are interconnected by "bridge" links to facilitate communication between them. Inspired by urban highway systems, this structure allows traffic to switch to the other ring via bridges in case of congestion on one ring, ensuring continuous data transmission.

B. Network on Chip Switching Mechanisms

The switching mechanism refers to the way data is exchanged between routing nodes in a Network on Chip (NoC). Current NoC switching mechanisms are broadly categorized into two types: connection-oriented switching and connectionless switching. Connection-oriented switching includes circuit switching, while connectionless switching includes packet switching, virtual channel switching, and wormhole switching [15]. The switching mechanisms are illustrated in Figure 7.

1) *Circuit Switching*: Circuit switching is a connection-oriented switching method characterized by dedicated bandwidth along the established path. Once a connection is set up, data can directly transmit along the locked path to the

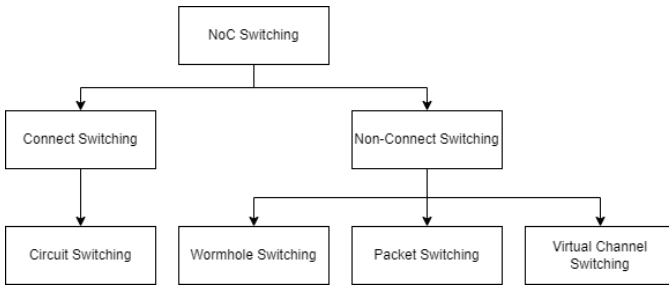


Fig. 7. switching mechanisms

destination node without needing route selection. At the end of communication, a signal releases the locked path.

Circuit switching ensures quality of service for data communication and can transmit large amounts of data over long distances without loss. However, setting up a circuit switch involves significant time, thereby increasing packet latency.

2) *Wormhole Switching*: Wormhole switching is currently one of the most prevalent switching mechanisms. In wormhole switching, data is segmented into many small flits, each of which moves through the network following the head flit like a pipeline. When congestion occurs, flits can be temporarily buffered at the current node, so each routing node in wormhole switching has buffering resources. Adopting wormhole switching consumes more area and can lead to deadlock during severe congestion.

3) *Packet Switching*: Packet switching involves dividing data into several fixed-length packets, each of which is stored in the buffer of routing nodes before being forwarded. Therefore, packet switching requires significant area and results in relatively high latency.

4) *Virtual Channel Flow Control*: In traditional wormhole switching, head-of-line blocking is a common issue. To address this problem, virtual channel switching employs multiple virtual channels instead of a single deep FIFO. This approach parallelizes the blocked serial data, allowing subsequent data to bypass the congested routing node and continue transmission downstream.

C. Network on Chip Routing Algorithms

A routing algorithm is responsible for selecting the path data packets take from a source node to a destination node. The choice of routing algorithm involves balancing multiple factors: ensuring packets reach their destination, preventing deadlock and livelock, and ideally selecting the shortest path while distributing load evenly. Additionally, simpler routing algorithms are preferred for easier implementation.

Routing algorithms are generally classified into two main categories: deterministic and adaptive. Deterministic routing algorithms determine the path from source to destination based solely on the coordinates of the source and destination nodes. While less flexible, they are easier to implement and more stable. Adaptive routing algorithms, on the other hand, dynamically adjust the routing path based on real-time network conditions, providing greater flexibility.

Arbitration	advantages	disadvantages
Round-Robin	Fairness	Not high-throughput
priority	Considers actual traffic	Low-priority not starved
combined	Fairness and quality assurance	No

TABLE I
ADVANTAGES AND DISADVANTAGES OF THREE ARBITRATION ALGORITHMS

Deterministic algorithms offer a unique path once the coordinates of the source and destination nodes are known, making them predictable but less adaptable to changing network conditions. Adaptive algorithms, by contrast, adjust routes based on current network conditions, offering greater flexibility but requiring more computational overhead to determine optimal paths dynamically.

D. Router Arbitration Algorithms

Regarding data requests coming from different directions but heading towards the same output direction, an arbiter is needed to select which data request successfully transmits. The implementation of arbiter algorithms is closely related to the structure of on-chip network routers, with different router structures corresponding to different complexity levels of arbiter structures. Based on the performance characteristics of arbiter algorithms, we categorize them into round-robin arbitration, priority arbitration, and combined round-robin and priority arbitration [16]. The advantages and disadvantages of these three arbiter algorithms are shown in the table I.

E. Deadlock and Livelock Issues

In on-chip networks, when data packets occupy network resources, subsequent packets may have to wait for the release of resources by preceding packets, resulting in a situation where several conflicting tasks in the Network-on-Chip (NoC) deadlock each other and cannot proceed, known as deadlock [17]. Deadlock in on-chip networks occurs due to insufficient resource allocation or competition for resources, such as contention for links or insufficient cache units.

In contrast to deadlock, livelock in NoCs involves packets circling around their target addresses without reaching them, continuously attempting transmission through the network but failing to reach their destinations. When on-chip networks employ dynamic routing algorithms, livelock issues are likely to occur.

F. Chapter Summary

This chapter provides an introduction to the fundamentals of Network-on-Chip (NoC). It begins by introducing several commonly used on-chip network topologies. Next, it discusses the classification of current on-chip network switching mechanisms and routing algorithms. Following this, it explores the advantages and disadvantages of three arbitration algorithms. Finally, it elaborates on the common issues of deadlock and livelock in NoCs.

III. DESIGN OF PACKET CONNECT CIRCUIT NETWORK ON CHIP ROUTER

A. Packet Connect Circuit (PCC)

In the previous chapter, several on-chip network switching mechanisms were introduced, among which packet switching simplifies router structures but consumes more area due to the use of caches. Circuit switching ensures the quantity and quality of communication but reduces link utilization. Therefore, researchers at Lin Xueping University have proposed a new switching mechanism that combines the advantages of both packet switching and circuit switching, called "Packet-Connection Circuit" switching, abbreviated as PCC [5].

1) *Introduction to PCC Mechanism:* In this thesis, the Packet-Connection Circuit (PCC) switching mechanism is adopted, characterized by establishing links through the transmission of request packets, while data transmission occurs in a circuit-switching manner.

Each data transmission using the PCC mechanism involves four stages [7]:

- **Link Request Stage:** The source routing node initiates a request packet, also known as a header packet, into the network. Decoders in routers make routing decisions based on address information in the header packet using routing algorithms. When a header packet successfully passes through a router node, it pre-locks the traversed link, making it unavailable for other header packets. If congestion occurs and the header packet is blocked, the current node sends a failure signal back to the source node. The source node retries sending the header packet until successful. Upon successful arrival at the destination node, the process moves to the next stage.
- **Link Establishment Stage:** Upon receiving the header packet, the destination node sends an acknowledgment signal back. This signal travels back to the source node along the pre-locked path, transitioning the pre-locked state to a locked state. Arrival of this acknowledgment signal at the source node indicates successful link establishment and allows progression to the next stage.
- **Data Transfer Stage:** Once the link is established, the source node begins sending data streams. Each data stream consists of packets or flits (flow control digits), with varying numbers per stream.
- **Link Release Stage:** After data transmission completes, the destination node sends a release signal to unlock the previously locked link.

The flowchart depicting these four stages is shown in Figure 8.

2) *Analysis of PCC Advantages:* Packet-Connection Circuit (PCC) switching combines the features of both packet switching and circuit switching mechanisms. This hybrid approach allows for dynamic allocation of resource channels during the link request and establishment stages, while utilizing dedicated channels during the data transfer stage to achieve high-speed transmission of large volumes of data [18].

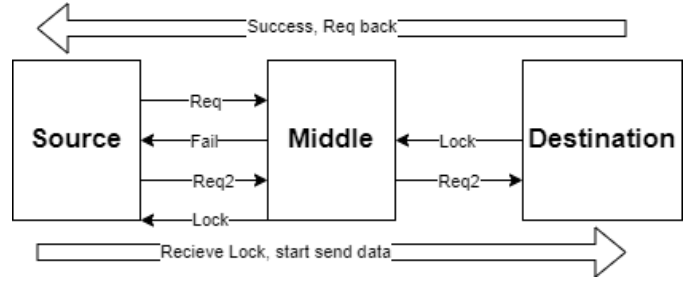


Fig. 8. PCC mechanism

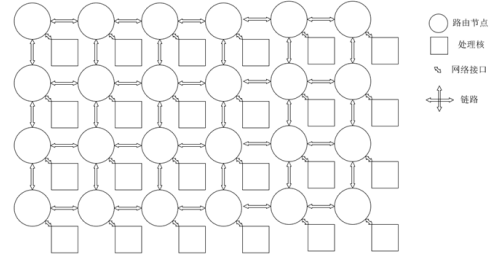


Fig. 9. 4x6 2D-Mesh structure

Regarding deadlock issues, PCC switching prioritizes releasing links over indefinite waiting if a request signal is denied. Therefore, PCC switching does not lead to deadlock situations [19].

In summary, compared to pure packet switching, PCC switching significantly reduces hardware implementation area and can effectively reduce latency during transmission. Compared to circuit switching, PCC switching improves link utilization efficiency by not permanently occupying links.

B. Design of PCC Network on Chip Router Based on 2D-Mesh Topology

1) *Overall Design:* This paper first designs a 4x6 2D-Mesh structure of a PCC on-chip network, depicted in Figure 9. The entire on-chip network consists of routing nodes, processing cores, network interfaces, and channels. In the 2D-Mesh structure, all routing nodes have a consistent layout, featuring links in the north, south, east, and west directions, enabling communication with adjacent nodes in these directions. Each routing node hosts a processing core for information processing. For simplicity in this design, each processing core comprises only a data sending module and a data receiving module. The interface through which the processing core communicates with the routing node is called the network interface, responsible for connecting the processing core to the network. The lines connecting two routing nodes are referred to as channels, which facilitate data communication between nodes within the network.

In this paper, the design employs the classic XY routing algorithm and a priority-based arbitration algorithm on a 2D-Mesh structure. To facilitate future scalability, the Network-on-Chip (NoC) adopts a reusable modular design approach. This means that individual routing nodes are interconnected

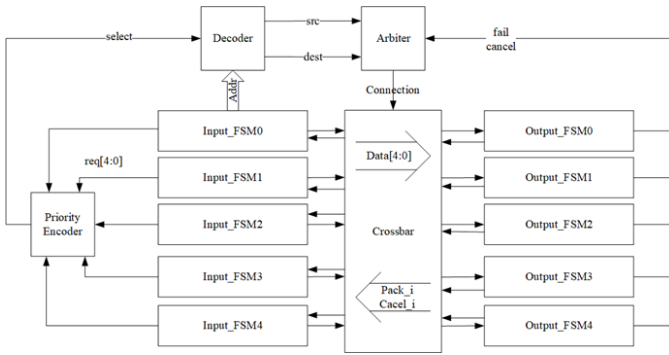


Fig. 10. 2D-Mesh Router architecture

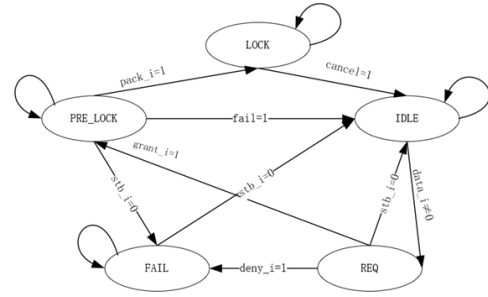


Fig. 11. input state machine

to form the entire 2D-Mesh on-chip network. This modular design approach significantly reduces the time and economic costs associated with chip design.

2) *Single Node Router Design*: A routing node serves as the fundamental unit of the on-chip network and should incorporate the following functionalities:

- **Request Establishment**: When an input channel generates a request, it applies to the arbiter to establish a link.
- **Address Analysis**: During the link establishment phase, the decoder intercepts the destination address and current address from the header packet. By comparing these addresses using routing algorithms, it determines the next direction (selecting the corresponding output channel).
- **Arbitration Function**: In scenarios where multiple requests simultaneously apply to establish links, the priority arbiter sorts them based on predefined priorities. The request with the highest priority is granted the link establishment first.
- **Link Revocation and Feedback Signals**: When the destination node successfully receives all data, it propagates a link cancellation signal upstream. In case of congestion, failure signals propagate upstream to indicate unsuccessful transmission.

Based on these functionalities, each routing node should include the following components: Five Input Finite State Machines, Five Output Finite State Machines, Priority Module, Arbiter Module, Decoder Module, Crossbar Switch Module. The architecture is depicted in Figure 10. This architecture enables efficient handling of requests, address analysis, arbitration, and feedback signaling within the on-chip network, ensuring reliable and scalable communication.

- **Design of Input Finite State Machine Module**: Each routing node has five input state modules, numbered 0 to 4, representing five directions: local, east, south, west, and north. In this design, the input state machine has 5 operational states: Idle, Request, Pre-lock, Lock, and Fail. The state machine remains in the Idle state if no request is received. Upon receiving a request to establish a link, it transitions to the Request state. The direction in the Request state with the highest priority receives the permission signal

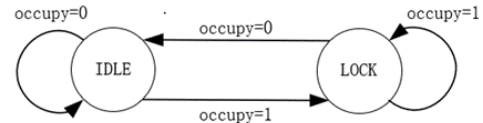


Fig. 12. output state machine

from the arbiter, causing the state machine to transition to the Pre-lock state. Upon successful establishment of all links and receipt of feedback acknowledgment signals, it enters the Lock state. If no success signal is received during the Request, Pre-lock, or Lock states, the machine transitions to the Fail state. Upon detection of any failure or link cancellation signal in any state, the machine returns to the Idle state. The state transition diagram for the input state machine is shown in Figure 11.

- **Design of Output Finite State Machine Module** Similar to the input state machine, each routing node also has five output state modules, corresponding to the same directions for bidirectional data transmission. Thus, each routing node can engage in bidirectional data transfer with four directions.

The output state machine has only two states: Idle and Busy. When the output channel is idle and selected by the arbiter for transmission, the output state machine transitions to the Busy state. During the link release phase, when the arbiter releases the channel, the state machine transitions back to the Idle state. In all other cases, the state machine remains in its current state.

The state transition diagram for the output state machine is illustrated in Figure 12.

- **Priority Module** When multiple requests from input state machines reach the priority module, the priority module sorts them based on predefined priorities. The request with the highest priority direction will be executed first. The priorities for each direction are outlined in Table II.
- **Arbiter Module** The arbiter module receives input signals from the decoder and provides output signals to three modules: It sends a grant or deny signal to the input state machines

Direction	Local	East	South	West	North
Priority	5	4	3	2	1

TABLE II
PRIORITY OF ALL DIRECTION

to indicate whether a request for link establishment has been successful. It issues an occupy signal to the output state machine modules to indicate the occupation of an output channel. It sends a connection signal to the crossbar switch module to lock the output channel with the corresponding input channel. The arbiter module plays a crucial role in managing and coordinating the establishment and release of links within the routing node of the on-chip network.

- Decoder Module

The core of the entire routing node is the decoder module because it executes the entire routing algorithm. The decoder module determines which output channel direction to select based on the comparison between the destination address and the current address. Once the arbiter confirms the channel is idle, it completely locks that channel.

For the classic XY routing algorithm, each address is divided into two coordinates (X, Y). The decoder module follows these steps:

X-direction determination: Compare the X-coordinate of the destination address with the current node's X-coordinate. If the destination X-coordinate is greater than the current X-coordinate, the target direction is east; otherwise, it is west.

Y-direction determination: If the X-coordinates are equal, compare the Y-coordinate of the destination address with the current node's Y-coordinate. If the destination Y-coordinate is greater than the current Y-coordinate, the target direction is north; otherwise, it is south.

Destination reached: If both X and Y coordinates are equal, it indicates that the destination node has been reached. The packet then passes through the network interface into the processing core.

The decoder module plays a critical role in directing packets through the on-chip network efficiently according to the routing algorithm specified, ensuring proper delivery to the intended destination.

- Crossover switch module

The crossbar module is a combinational logic module, which will transmit the corresponding input data information to the output channel according to the link signal of the arbiter, and also feed back the control signal of link establishment and cancellation.

C. Design of PCC Network on Chip Based on Dual-Ring Topology

1) Overall Design: Inspired by urban traffic lines in 2004, the University of Yenching in Sweden put forward the R2NoC structure of the loop [20]. On this basis, this paper reduced the three-layer loop to a Double Ring (DR), and adopted

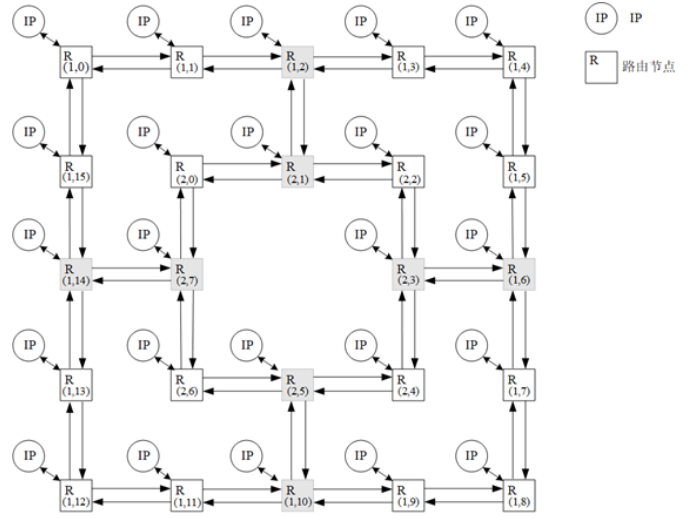


Fig. 13. DR NoC Architecture

packet circuit switching instead of wormhole switching. The 24-node double ring structure designed in this paper is shown in Figure 13.

For DR structure, it is divided into two rings, the outer ring coordinate is 1 and the inner ring coordinate is 2. There are two kinds of routing node structures in DR structure: the first is normal routing node, which has only two channels except the local network interface, and there are a large number of them on the ring. The other is a bridge routing node, which acts as a channel node for exchanging data between the inner and outer loops. When a loop is blocked, data transmission across the loop can be carried out through bridge nodes. Therefore, the two kinds of single-node routers should be designed separately.

Because of the change of topological structure, the whole routing algorithm can't be routed according to the XY routing algorithm in 2D-Mesh structure, so this paper adopts the static ring routing method designed by ourselves, and the arbitration still adopts priority arbitration.

2) Single Node Router Design:

1) Normal routing node

A normal routing node has three input and output ports. The No.0 input state machine and output state machine are local communication channels, the No.1 input state machine and output state machine enter clockwise (CLKW), and the No.2 input state machine and output state machine enter counterclockwise (ATCLKW). The architecture is shown in Figure 14.

Because the nodes on the same ring are connected end to end, the No.2 output state machine of the previous node is connected with the No.1 good input state machine of the current node, thus forming a two-layer loop of inner ring and outer ring. There are two channels in each loop, one for clockwise transmission and the other for counterclockwise transmission, which do not affect each other.

Compared with 2D-Mesh router, the number of input

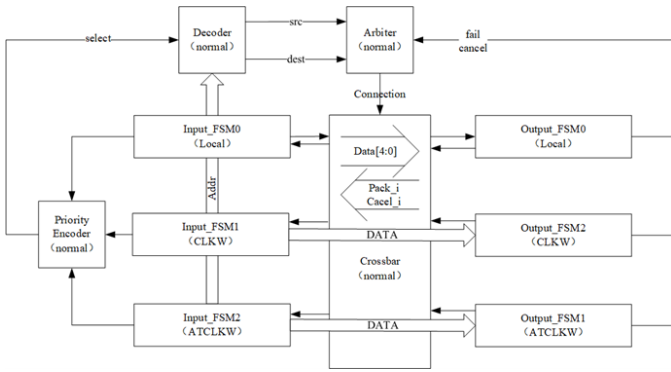


Fig. 14. DR normal router architecture

Outer node	Mapping node	Inner node	Mapping node
2	0,1,2,3	1	0,1
6	4,5,6,7	3	2,3
10	8,9,10,11	5	4,5
14	12,13,14,15	7	6,7

TABLE III

ROUTING BRIDGE MAPPING RELATIONSHIP

and output state machine modules in DR router is reduced to three, and the priority module, arbitration module, address decoding module and crossbar module are all improved for DR router. Among them, the address decoding module is the most improved, because the routing algorithm in this module is improved from XY routing algorithm to static ring routing method suitable for DR structure.

The static ring routing algorithm first maps the normal nodes on the inner and outer rings to the corresponding bridge routing nodes, as shown in Table III.

The address decoding module will first judge whether to transmit in the same ring or across rings according to the address of the current node and the address of the destination node. If the transmission is in the same ring, the first routing node judges whether the distance is clockwise or counterclockwise, and then the rest nodes output directly from another port without judging the direction. If it is cross-ring transmission, the bridge node is selected according to the bridge mapping, and then the same-ring transmission is carried out to the bridge routing node.

2) Bridge routing node

Bridge routing nodes are precious resources in the DR on-chip network, with only 8 bridge nodes available for cross-ring communication in the 24-node DR-NOc. The structure of bridge routing nodes is depicted in Figure 15. As shown in Figure 15, nodes 2, 6, 10, and 14 on the outer ring use bridge node routers, corresponding to nodes 1, 3, 5, and 7 on the inner ring, which also use bridge node routers. Their respective input and output state machines (number 3) are interconnected to form cross-ring bridges.

Due to the additional input and output state machines

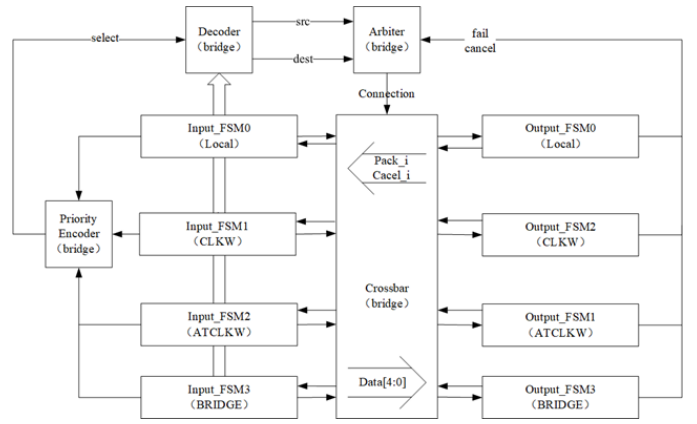


Fig. 15. DR Bridge router architecture

in bridge routing nodes compared to regular routing nodes, modifications have been made to the priority module, arbiter, and crossbar switch module. In regular routing nodes, local priority is highest, followed by clockwise (CLKW) direction priority which is higher than counterclockwise (ATCLKW) direction. However, in bridge routing nodes, the priority of bridge channels is lower than local priority but higher than the clockwise direction. This means that when a head packet is at a bridge routing node and needs to cross to the other ring, it will prioritize crossing the bridge for transmission.

D. Chapter Summary

This chapter first introduces the packet circuit switching mechanism used in this graduation project, and analyzes the advantages of packet circuit switching compared with the other two switching mechanisms and the timing of packet circuit switching in the simulation. Then, a PCC network-on-chip based on 2D-Mesh structure is designed, and the internal structure of the single-node router and the design of each module are described in detail. Then, a PCC network-on-chip based on a new DR(Double Ring) structure is proposed, which has two different routing nodes, and the design of the two routing nodes is introduced respectively to prepare for the experiment in the next chapter.

IV. SIMULATION AND VERIFICATION OF PACKET-SWITCHED NETWORK ON CHIP

A. Simulation of Single Node Router Submodules

• 2D-Mesh PCC-NoC

- The output state machine operates similarly to the input state machine. As shown in Figure 16, upon receiving the $pack_i$ signal, which indicates the successful establishment of the link, it transitions to the lock state. After the data transmission is complete, the $cancel_o$ signal is sent to cancel the link.
- As shown in Figure 17, the arbiter receives two data inputs, src and $dest$, where src indicates the input channel number and $dest$ indicates the output

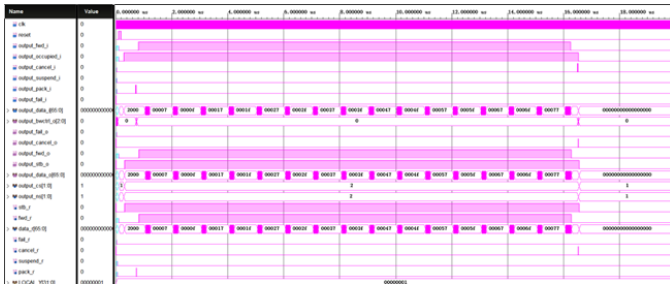


Fig. 16. Output state machine simulation timing

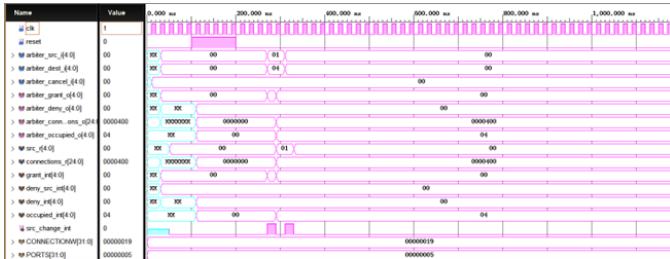


Fig. 17. Arbiter simulation timing

channel number. The arbiter checks the status of the corresponding output channel to determine if it is idle. If the output channel is idle, the arbiter sends a grant signal to the output state machine and sets the corresponding bit in the connection signal to 1.

- As shown in Figure 18, the address decoder module first analyzes the head packet to obtain the source and destination addresses. Based on the XY routing algorithm, it first determines whether the X coordinates are the same. Depending on the relative positions, it assigns the value 1 to the dest signal to indicate the direction of the output.
- As shown in Figure 19, the crossbar switch module routes the data from the corresponding input state machine to the output state machine module for transmission based on the connection signal received from the arbiter.

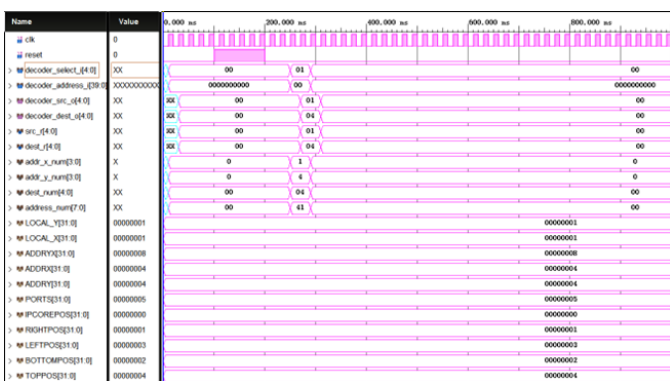


Fig. 18. Decoder simulation timing

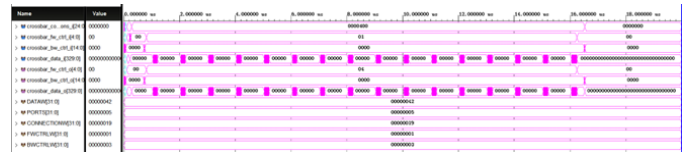


Fig. 19. Crossbar simulation timing

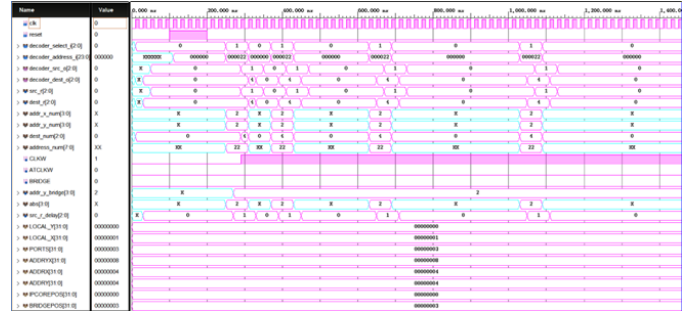


Fig. 20. Decoder of Normal Router simulation timing

• DR PCC-NoC

The most critical component in the DR-structured NoC is the address decoder module, so the primary focus is on simulating this module. Since the DR-structured NoC has two different router architectures, each address decoder module is simulated separately.

- Normal Router

A data flow is set up to be sent from node (1,0) to node (2,2). The simulation waveform of the address decoder module at node (1,0) is shown in Figure 20. First, the destination address (2,2) is analyzed, and then, based on the current node address (1,0), it is determined that the next step is to reach node (1,2). This node will use the bridge to enter node (2,1), so the initial transmission should proceed clockwise within the same ring.

- Bridge Router As shown in Figure 21, the simulation of the address decoder module at the bridge node (1,2) confirms that this node is the bridge node, so it raises the BRIDGE signal and outputs the data across the bridge to the other ring. According to Figure 17, for the head packet set to flow from node (1,0) to node (2,2), it must pass through node (1,2), entering from input state machine 1 and exiting through output state machine 3. The overall timing sequence of the bridge node is depicted in Figure 22.

B. Hardware Design and Simulation Experiments

This chapter conducts packet transmission and reception simulation experiments on the two types of packet connection circuit (PCC) on-chip networks designed previously. Before detailing the specific experimental procedures, the experimental environment and the hardware and software platforms used in this work are introduced: All simulation experiments were

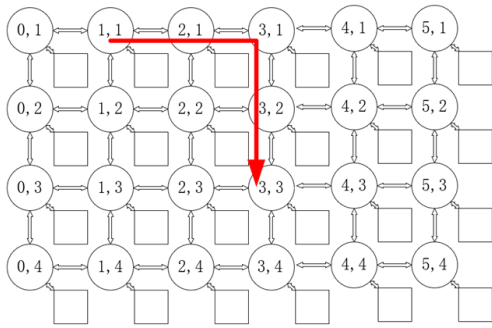


Fig. 26. Mesh Routing

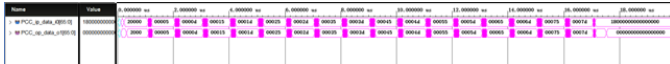


Fig. 27. Timing Simulation of (1,1) Node

pletes the entire routing process at the source node. Please refer to simulation waveform in Figure 27 for details.

At node (2,1), similar to node (1,1), data continues to be transmitted through output state machine 1. Upon reaching node (3,1), according to the routing algorithm, since the destination address has the same X coordinate as the current address, it checks the Y coordinate. Therefore, it should proceed southward, transmitting data through output state machine 2, as shown in simulation waveform in Figure 28.

At node (3,2), similar to node (3,1), data continues to be transmitted through output state machine 2. Upon reaching node (3,3), according to the routing algorithm, since the destination address has the same X and Y coordinates as the current address, it determines arrival at the destination node. The header packet is outputted from output state machine 0 to the data receiving module, followed by sending an acknowledgment signal. All other data is transmitted along the path of the header packet to the data receiving module. The waveform is depicted in Figure 29.

b) DR PCC-NoC

For the 24-node DR structure on-chip network,



Fig. 28. Timing Simulation of (3,1) Node



Fig. 29. Timing Simulation of (3,3) Node

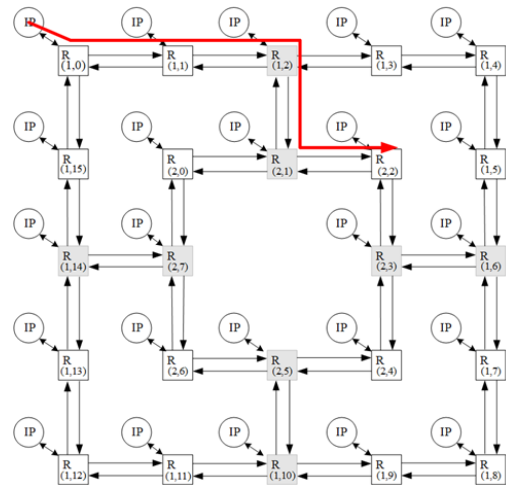


Fig. 30. DR NoC Routing



Fig. 31. Timing Simulation of (1,0) Node

this study sets up a data flow from (1,0) to (2,2), as illustrated in Figure 30. The configuration of the data sending module remains the same as in previous experiments, mounted on node (1,0) with the destination address set to (2,2).

At node (1,0), the data sending module transmits the header packet. Input state machine 0 (local) detects the header packet, and since there is only one request, it has the highest priority. The header packet enters the address decoding module, which retrieves the destination address coordinates (2,2). It is determined that the X-coordinate of the destination address differs from the current node's X-coordinate, necessitating a cross-ring transfer. According to the mapping, it is identified that the transfer needs to proceed from node (1,2) across to node (2,1). Therefore, the data should be directed clockwise from output state machine 2. The arbiter checks if output state machine 2 is idle, and the crossbar switches the data to output state machine 2. This completes the entire routing process at the source node. Refer to simulation waveform in Figure 31 for details.

The (1,1) node is the same as the (1,0) node, and data is always transmitted from the direction of the output state machine 2. Until the (1,2) node, according to the routing algorithm, it needs to cross the bridge to reach the bridge node at this time, so the data should be transmitted through the output state machine 3 (bridge). Similarly, the input state machine 3 of the (2,1) node will receive the transmitted data and transmit it in the same ring

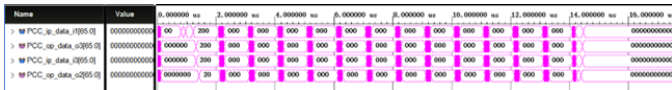


Fig. 32. Timing Simulation of (2,1) and (1,2) Node



Fig. 35. Receiver Timing Simulation of (1,2) Node

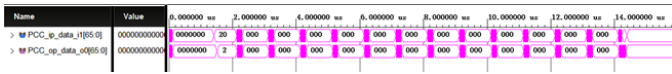


Fig. 33. Timing Simulation of (2,2) Node

again. The simulation waveform is shown in Figure 32.

Until the (2,2) node, according to the routing algorithm, the destination address at this time is the same as the current address in X and Y coordinates, and it is judged that the destination node has been reached. The head packet is output from that output state machine 0 to the data receive module, and the response signal fed back by the data receiving module is receive, and all other data are transmitted to the data receiving module along the head packet path. The waveform is shown in Figure 4.19.

4) Simulation Experiment of Two-stream Cross-sending and Receiving

According to the single stream receiving and sending experiment in the previous section, it has been proved that both NOCs can operate normally, but when there is more than one data stream in the network on chip, it is the time to really test the performance of the network on chip. According to the principle of packet circuit switching, we can easily find a phenomenon: after the first packet 1 has locked an output direction of the routing node, the first packet 2 entering from other directions cannot be output from the locked direction for this routing node, as shown in Figure 34.

In a 2D-Mesh NoC, a data stream from (1,1) to (1,3) and a data stream from (0,2) to (1,3) are set, and the simulation results are shown in Figure 35. The node (1,2) receives the input requests from the west and the north, and gives priority to the transmission from the west through priority judgment. Because the output directions of the two streams are the same, the second stream sends back a failure signal and waits for the end of the

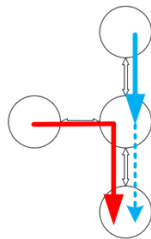


Fig. 34. Conflict situation of two streams

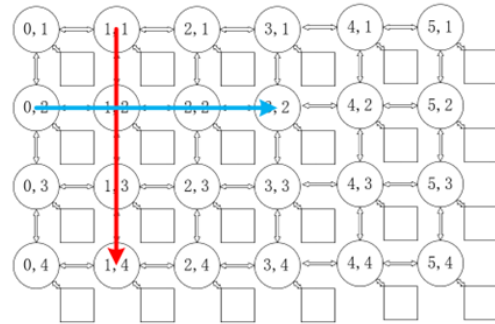


Fig. 36. Two cross flow in Mesh

transmission of the first stream.

If the output channel selected by the second stream is not occupied, then the two streams can efficiently use the same node for transmission. In the 2D-Mesh NoC, the data streams (1,1) to (1,4) and (0,2) to (3,2) are set, and the data paths are shown in Figure 36. In the NoC with DR structure, the data streams from (1,0) to (1,3) and (1,3) to (1,0) are set, and the data paths are shown in Figure 37.

After Vivado simulation experiment, the waveform is shown in Figure 38 below. It can be seen that (1,2) node of mesh structure has two input channels and both output channels have header and data transmission, which proves that (1,2) node can pass through two unrelated streams at the same time. As shown in Figure 39, the DR structure can also take two data streams in opposite directions at the same time.

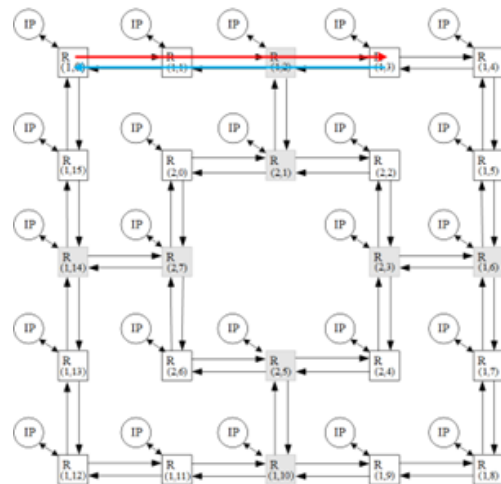


Fig. 37. Two flow in DR

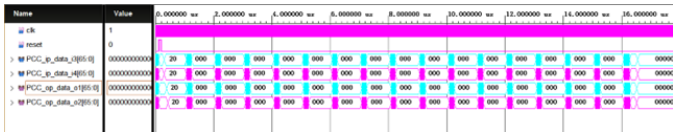


Fig. 38. Timing Simulation of (1,2) Node

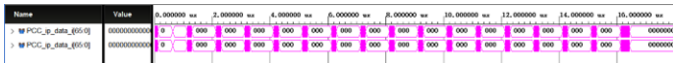


Fig. 39. Two data stream in DR

5) Multi-stream conflict simulation experiment

a) 2D-Mesh PCC-NoC

With the increasing number of data streams in NoC, the network-on-chip congestion will become more and more serious, and the communication efficiency may be reduced. In the multi-stream collision experiment, this paper sets four streams to run in the network-on-chip platform at the same time, and observes the congestion by simulating the waveform.

In a 2D-Mesh structure NoC, the first stream is (0,1) to (3,2), the second stream is (1,1) to (3,3), the third stream is (5,2) to (2,2), and the last stream is (0,4) to (3,2). The data route of the four streams is shown in Figure 40.

In the diagram, red represents the first flow, blue the second flow, green the third flow, and yellow the fourth flow. In the Mesh structure, flow conflicts primarily include co-directional and crossing transmissions. The four configured flows satisfy all conflicts: the first and second flows conflict in the same direction, the second and third flows in crossing transmission, and the second and fourth flows in reverse transmission.

In the Vivado simulation, the waveforms of the four flows are shown in Figure 41. From the waveforms, it can be seen that all four flows correctly reach their destination nodes. The second (blue) and third (green) flows almost reach the destination node si-

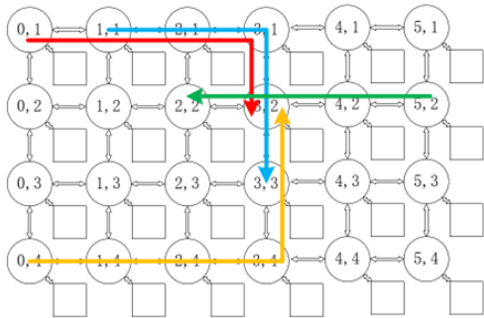


Fig. 40. 4 data flow

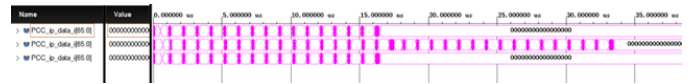


Fig. 41. Timing simulation of 4 data flow

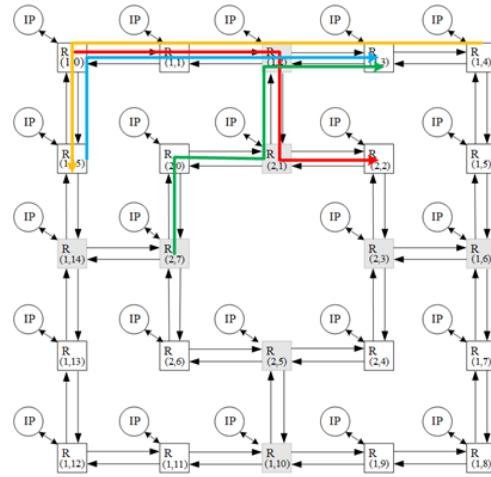


Fig. 42. 4 dataflow in DR

multaneously, followed by the fourth flow (yellow) as the second to arrive at the destination node. Due to congestion caused by the blue flow, the first flow (red) waits until the blue flow completes transmission before proceeding, ultimately arriving at the destination node last.

b) DR PCC-NoC

In the DR-structured NoC, the first flow is from (1,0) to (2,2), the second flow is from (1,15) to (1,3), the third flow is from (2,7) to (1,3), and the final flow is from (1,4) to (1,15). The data paths of these four flows are shown in Figure 42.

The red color in the diagram represents the first flow, blue represents the second flow, green represents the third flow, and yellow represents the fourth flow. There is a co-directional conflict between the first and second flows, a bridge conflict between the first and third flows, and a crossing transmission between the second and fourth flows. The transmission waveforms of these four flows in Vivado simulation can be seen in Figure 43.

The waveform chart shows that all four flows correctly reach their destination nodes. The second flow (blue) arrives at the destination node first, followed by the fourth flow (yellow) as the second to arrive. Due to congestion caused by the blue



Fig. 43. Timing simulation of DR

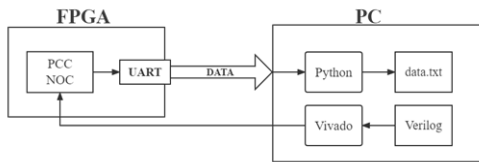


Fig. 44. Implementation and verification system structure

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 10.314 ns	Worst Hold Slack (WHS): 0.096 ns	Worst Pulse Width Slack (WPWS): 9.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 12007	Total Number of Endpoints: 12007	Total Number of Endpoints: 8502

All user specified timing constraints are met.

Fig. 45. 2D-Mesh NoC timing report

flow blocking the channel, the first flow (red) and the third flow (green) wait until the blue flow completes its transmission before proceeding.

C. FPGA-Based Hardware Implementation and Verification

1) *System Overview:* This paper ultimately conducts hardware synthesis and implementation using Vivado on Xilinx's Artix 7 series XC7A35T FPGA chip for two types of packet-switched networks designed. To verify the successful operation of the NoC on FPGA, the paper connects a UART serial port to the data reception module of the destination routing node, transmitting received data to a PC via the UART serial port. For ease of observation, Python software on the PC is used for data processing, ultimately generating a TXT file. The entire system architecture is shown in Figure 44.

2) *Performance Evaluation:* In VIVADO, NoC is integrated to generate a bit stream file, and specific performance parameters can be obtained, including timing information, hardware resource consumption and power consumption information.

1) 2D-Mesh PCC-NoC

In Vivado, using Tcl commands can list all clocks and their parameters. The overall timing parameters are shown in Figure 45.

After synthesis, which also yields the hardware resource utilization of the system. The system uses a total of 15,153 LUTs and 8,451 D flip-flops, accounting for 72.85% and 20.31% of the total available on the development board, respectively. The specific details of the overall resource consumption of the system are shown in Table IV.

Resource	Utilization	Available	Utilization %
LUT	15153	20800	72.85
FF	8451	41600	20.31
BRAM	1	50	2.00
IO	3	250	1.20
BUFG	1	32	3.13

TABLE IV
2D-MESH NoC RESOURCE CONSUMPTION TABLE

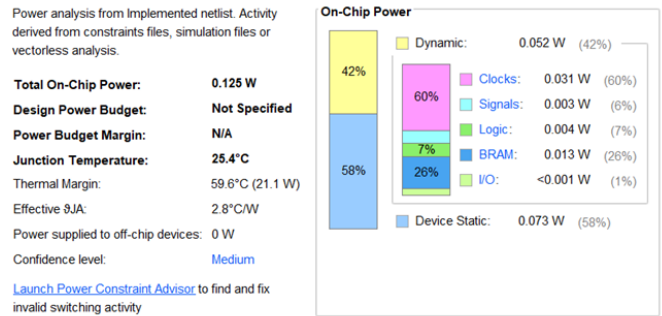


Fig. 46. 2D-Mesh NoC power consumption report

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 12.366 ns	Worst Hold Slack (WHS): 0.066 ns	Worst Pulse Width Slack (WPWS): 9.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 8558	Total Number of Endpoints: 8558	Total Number of Endpoints: 6810

All user specified timing constraints are met.

Fig. 47. DR NoC timing report

The total power consumption of the system is 0.125W, with a node temperature of 25.4 degrees Celsius. The total dynamic power is 0.052W, accounting for 42% of the total power consumption, which is reasonable. Figure 46 shows the specific power consumption details

2) DR PCC-NoC

The overall timing parameters of PCC-NoC with DR structure are shown in Figure 47 below:

After synthesis, the hardware area resource usage of the system will also be obtained. This system uses LUT8553 LUTs and 6743 D flip-flops, accounting for 41.12% and 16.21% of the total development board respectively. The specific content of the overall resource consumption of the system is shown in Table V.

The total power consumption of the system is 0.117W, the node temperature is 25.3 degrees Celsius, and the total dynamic power consumption is 0.043W, accounting for 37% of the total power consumption. The overall power consumption is reasonable. Figure 48 below shows the specific power consumption:

3) *Board level verification:* This graduation project uses the Daffodil Pro development board from DFRobot, equipped with Xilinx's Artix 7 chip. The computer connects to the board's UART serial port, and the bitstream file is programmed into the FPGA using a JTAG downloader, as shown in Figure 49.

Resource	Utilization	Available	Utilization %
LUT	8553	20800	41.12
FF	6743	41600	16.21
BRAM	31	50	62.00
IO	3	250	1.20
BUFG	1	32	3.13

TABLE V
DR NoC RESOURCE CONSUMPTION TABLE

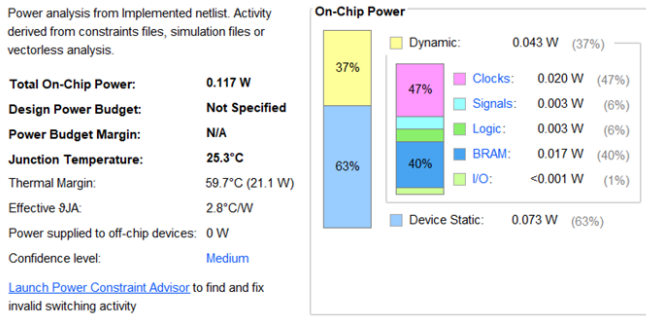


Fig. 48. DR NoC power consumption report

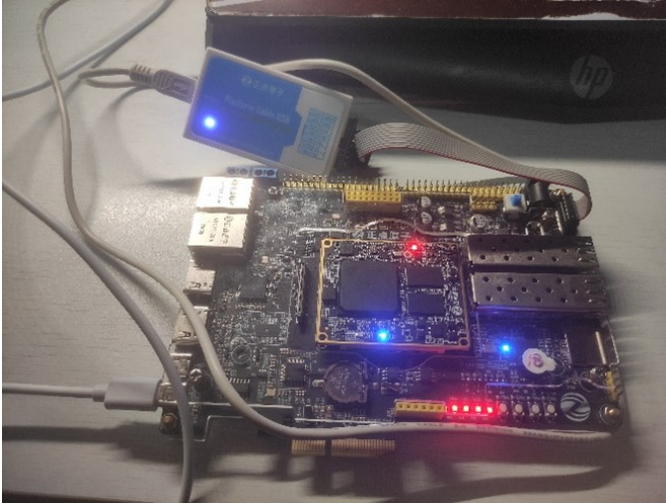


Fig. 49. FPGA physical test

At the PC end, Python receives the initial data, processes it, and stores the processed data into a TXT file, as shown in Figure 50.

D. Comparative Analysis of Two NoC Structures

Based on the experimental data collected from simulation and implementation in the previous sections, a comparative analysis is conducted on two different NoC structures. Firstly, latency analysis involves recording the total time consumed by the platform from packet establishment to completion of data

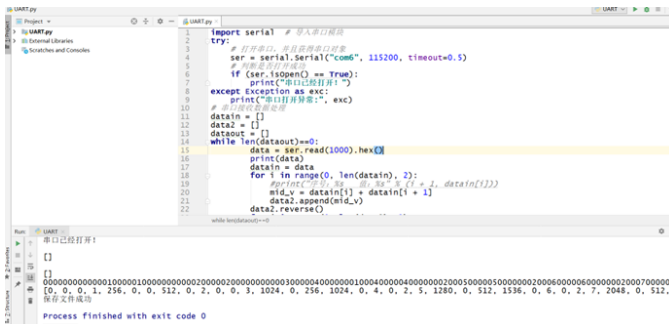


Fig. 50. Python data processor

	Lab1	Lab2	Lab3
2D-Mesh NoC	15.82 us	15.78 us	32.8 us
DR-NoC	15.94 us	15.7 us	30.6 us

TABLE VI
CAPTION

transmission in the three experiments in Section 1, as shown in Table 4.3.

From the analysis in Table 4.3, it can be observed that in the single-flow experiment, the 2D-Mesh structure is faster by 0.76%. In the two-flow experiment, the DR structure is faster by 0.51%, and in the multi-flow conflict experiment, the DR structure is faster by 6.71%. Overall, there is no significant difference in speed between the two, with the DR structure slightly faster when multiple flows are present. However, the number of flows in this experiment is too small, and differences may arise with more flows.

In terms of hardware resource consumption, the DR structure's NoC uses 31.73% fewer Lookup Tables (LUTs) and 4.1% fewer Flip-Flops (FFs) compared to the 2D-Mesh structure's NoC. Consequently, the DR structure significantly reduces resource consumption and hardware area.

In terms of power consumption, the total power consumption of the DR structure's NoC is 6.4% lower than that of the 2D-Mesh structure's NoC, with a 17.3% reduction in static power. Thus, the DR structure's NoC exhibits lower power consumption.

In conclusion, while both NoC structures exhibit similar transmission times, the DR structure shows significant advantages in terms of reduced hardware resources and power consumption. Therefore, DR structure has certain advantages over the 2D-Mesh structure.

E. Chapter Summary

This chapter primarily conducts simulation experiments and implementation verification of two designed on-chip networks. Firstly, each submodule of the single-node router is simulated. Then, through three experiments involving single-flow, two-crossing flows, and multiple conflicting flows, the NoC's functionality is validated to ensure it aligns with the design. Simultaneously, simulation data from both platforms are recorded for comparison purposes. Subsequently, the two NoCs are synthesized and implemented in Vivado, and data transmission to a PC via UART is used to verify successful operation. Finally, the quality of the two structures is analyzed through Vivado-generated timing reports, hardware resource utilization reports, and power consumption reports.

V. SUMMARY AND OUTLOOK

A. Summary

With the increasing performance and demand of Network on Chip (NoC), NoC has become the best alternative to bus communication. The switching mechanism of the router and the topology formed by the NoC routers significantly impact performance. This paper conducts research using a packet-switched circuit-based NoC, achieving the following results:

- Investigated the basic knowledge of NoC, introducing its topology, switching mechanisms, routing algorithms, arbitration algorithms, and issues of deadlock and livelock.
- Designed a single-node router using Verilog based on the packet-switched circuit method, including input/output state machines, arbiter modules, priority modules, address decoding modules, and crossbar switch modules. A 2D-Mesh structure NoC with 24 routing nodes was constructed. Then, the single-node router structure and routing algorithm were improved to form a Double Ring (DR) structure NoC, also with 24 routing nodes.
- Simulated and verified the sub-modules of the routers for both structures. Three experiments were conducted on both NoCs to verify normal data transmission and record delay times. The NoCs of both structures were synthesized using Vivado and verified on FPGA boards. The comparison of the Vivado hardware resource consumption report and power consumption report demonstrated the advantages of the DR structure.

B. Outlook

This paper primarily investigates the structure and topology of Network on Chip (NoC) routers based on packet-switched circuitry. However, several shortcomings requiring improvement were identified during the research process:

- During the link establishment phase of packet-switched circuitry, inevitable head-of-line blocking occurs as multiple header packets contend for routing paths within the platform, significantly reducing link utilization. Additionally, the fixed priority arbitration mechanism employed in this study may result in lower priority paths never establishing links. Introducing round-robin arbitration could mitigate this issue.
- Both NoC structures studied in this paper employ the most common static routing algorithms, lacking adaptability and fault tolerance. For the DR structure, it may be beneficial to develop specialized dynamic routing algorithms to better leverage the characteristics of DR and enhance NoC performance.
- The conclusions drawn from the three experiments conducted in this paper may not be fully accurate due to the limited sample size. It is recommended to conduct more experiments, adjust injection rates, calculate average packet latency, and employ other methods to derive more precise conclusions.

ACKNOWLEDGMENT

First, I would like to thank Professor Li Zhenmin for giving me the opportunity to participate in the research project of the research group, allowing me to apply what I have learned and identify my shortcomings in my final year. Professor Li provided valuable advice on many important issues that I found difficult to determine, helping me to choose the correct and suitable direction for my research.

Secondly, the research process of this thesis utilized much of the knowledge learned during my university courses. I would

like to express my sincere gratitude to all the professors who taught me during my university years, as the foundational knowledge gained from these courses greatly supported this graduation project.

I would like to thank senior student Ma Yuqing and my fellow group member Tan Xiao for their enthusiastic assistance when I encountered problems during my graduation project. I am also grateful to my roommates Luo Wen, Zhao Jingyi, Gao Yifan, and Li Yue for providing a fun and enjoyable time, helping me to relieve much of the pressure from my studies, and for the friendship we have built. Lastly, I would like to thank my parents for their upbringing, allowing me to grow up in a way that I enjoy.

REFERENCES

- [1] Zhonghai Lu and Axel Jantsch. "TDM virtual-circuit configuration for network-on-chip". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 16.8 (2008), pp. 1021–1034.
- [2] Ouyang Yiming, Dong Shaozhou, and Liang Huaguo. "Design and Simulation of NoC Routing Algorithm Based on 2D Mesh". In: *Computer engineering(China)* 35.22 (2009), pp. 227–229.
- [3] Srinivasan Murali et al. "A Method for Routing Packets Across Multiple Paths in NoCs with In-Order Delivery and Fault-Tolerance Guarantees". In: *VLSI DeSign* 2007.1 (2007), p. 037627.
- [4] Ahmed Hemani et al. "Network on chip: An architecture for billion transistor era". In: *Proceeding of the IEEE NorChip Conference*. Vol. 31. 20. sn. 2000, p. 0.
- [5] Daniel Wiklund and Dake Liu. "SoCBUS: Switched network on chip for hard real time embedded systems". In: *Proceedings International Parallel and Distributed Processing Symposium*. IEEE. 2003, 8–pp.
- [6] Zhang Zhe, Gao Xiaopeng, and Long Xiang. "A High Performance round-robin Arbitrator for Virtual Channel Routers". In: *journal of beijing university of aeronautics and astronautics(china)* 33.06 (2007), pp. 743–747.
- [7] Hou Ning, He Wei, and Song Yukun. "Packet connection circuit router supporting virtual circuit mechanism". In: *Journal of Hefei University of Technology: Natural Science Edition(china)* 38.4 (2015), pp. 485–489.
- [8] Wang Zhaoliang. "Fault-tolerant and low-power codec design based on NoC". MA thesis. Hefei University of Technology, 2016.
- [9] Ma Qingyong. "Research on Network on Chip (NoC) Interconnection Technology". MA thesis. University of Electronic Science and Technology of China (UESTC), 2013.
- [10] Zheng Yang. "Design of Fault-tolerant Adaptive Routing Algorithm for Network on Chip". MA thesis. Southeast China University, 2018.
- [11] Pan Pan. "NoC design based on 2D-Mesh topology". MA thesis. Anhui Univeristy, 2012.

- [12] Mikael Millberg et al. “The Nostrum backbone-a communication protocol stack for networks on chip”. In: *17th International Conference on VLSI Design. Proceedings*. IEEE. 2004, pp. 693–696.
- [13] Song yukun, Qian qingsong, and Zhang duoli. “Two-port NoC model and performance analysis of Torus topology”. In: *journal of electronic measurement and instrument(China)* 3 (2017), pp. 361–368.
- [14] Research on ring network-on-chip architecture based on wormhole routing. MA thesis. Zhejiang University, 2010.
- [15] Leon Wang et al. “Research on Network-on-Chip Switch System”. In: *china integrated circuit* 16.12 (2007), pp. 22–27.
- [16] Hao Xiaojie, Huaxi Gu, and Shang Junhui. “Research on Scheduling Algorithms of Network on Chip”. In: *china integrated circuit* 20.4 (2011), pp. 48–55.
- [17] Fu ZhiZhou. “Design and Research of Network on Chip (NoC) Switching Structure”. MA thesis. University of Electronic Science and Technology of China (UESTC), 2011.
- [18] Li Li et al. “Back-off-to-turn routing algorithm for network-on-chip based on packet-circuit switching”. In: *Journal of Electronics and Information Technology(China)* 33.11 (2011), pp. 2759–2763.
- [19] Yang Xin. “Research on the design of NoC routing technology”. MA thesis. Hefei University of Technology, 2016.
- [20] Henrik Samuelsson and Shashi Kumar. “Ring road NoC architecture”. In: *Proceedings Norchip Conference, 2004*. IEEE. 2004, pp. 16–19.
- [21] Li Miao. “Analysis and Optimization of Multi-path Routing NoC Reorganization Cache”. MA thesis. Hefei University of Technology, 2015.