International Journal of Engineering Science Invention Research & Development; Vol. I Issue IX March 2015
www.ijesird.com e-ISSN: 2349-6185

NoC
(Network on chip)
Victory Bidirectional Ring NoC Router

Purva Dave (Shrimali)
Sri Balaji College of Engineering and Technology, Jaipur (Rajasthan)
(Department of Electronics & Telecommunication (VLSI Design)
Rajasthan Technical University, Kota
purva.harsh@yahoo.com

***Abstract*- Designing ASICs for each new generation of backbone routers is a time intensive and fiscally draining process. In this paper we focus on the design of the Victory bidirectional ring network-on-chip (NoC) router, to be used as the building block of a NoC for multi-core processors. The router is a simple router that uses source routing, fixed-size 64-bit packets and a virtual cut-through style of switching. Additionally, it uses two virtual channels per physical channel to provide for deadlock-free routing in the ring. Since it uses virtual cut-through and packets are only 64 bits that means that all input and output buffers are 64 bits wide. The following sections detail the relevant aspects of the router architecture, specification and significance of NoC in VLSI design.**

***Keywords*- Systems-on-Chip, Multi-processor systems, Networks-on-Chip, Ring router, Clock (clk)**

I. INTRODUCTION

To meet the growing computation-intensive applications and the needs of low-power, high-performance systems, the number of computing resources in single-chip has enormously increased, because current VLSI technology can support such an extensive integration of transistors. By adding many computing resources such as CPU, DSP, specific IPs, etc to build a system in System-on-Chip, its interconnection between each other becomes another challenging issue. In most System-on-Chip applications, a shared bus interconnection which needs arbitration logic to serialize several bus access requests, is adopted to communicate with each integrated processing unit because of its low-cost and simple control characteristics. However, such shared bus interconnection has some limitation in its scalability because only one master at a time can utilize the bus which means all the bus accesses should be serialized by the arbitrator. Therefore, in such an environment where the number of bus requesters is large and their required bandwidth for interconnection is more than the current bus, some other interconnection methods should be considered. [6]

Such scalable bandwidth requirement can be satisfied by using on-chip packet-switched micro-network of interconnects, generally known as Network-on-Chip (NoC) architecture. The basic idea came from traditional large-scale multi-processors and distributed computing networks. The scalable and modular nature of NoCs and their support for efficient on-chip communication lead to NoC-based system implementations. Even though the current network technologies are well developed and their supporting features are excellent, their complicated configurations and implementation complexity make it hard to be adopted as an on-chip interconnection methodology. In order to meet typical SoCs or multi-core processing environment, basic module of network interconnection like switching logic, routing algorithm and its packet definition should be light-weighted to result in easily implemental solutions.

In communication between several cores in System-on-Chip (SoC) environment, some prevailing mechanisms for this purpose are several bus-based architectures and point-to-point communication methodologies. Emerging Network-on-Chip (NoC) designs consist of a number of interconnected heterogeneous devices (e.g. general or special purpose processors, embedded memories, application specific components, mixed-signal I/O cores) where communication is achieved by sending packets over a scalable interconnection network. [6]

The design of NoCs trades-off several important choices, such as topology selection, routing strategy selection and application mapping to network nodes. Developing a design methodology for NoC-based communication poses novel and exciting challenges to the EDA community. [4]

For simplicity and ease of use, the bus-based architectures are the most common. However, in bus-based architecture, it has fundamentally some limitation in bandwidth, i.e. while the number of components attached to the bus is increased, a physical capacitance on the bus wires grows and as a result its wiring delay grows even further. To overcome the fundamental limitation of scalability in bus-based architectures, some advanced bus architectures such as ARM, AMBA [2], Open Cores, System-on-Chip (SoC) interconnection [3], and IBM Core Connect [4], are adopted. The Figure 1 illustrates basic structure of ARM AMBA. As shown in Figure 1, most of advanced bus architectures adopt a hierarchical structure to obtain scalable communication throughput and partition communication domains into several group of communication layers depending on bandwidth requirement such as high-performance, low-performance and so on. [4]

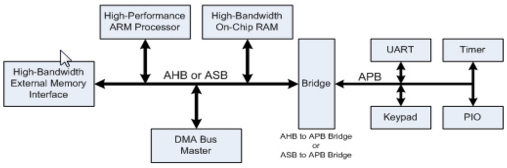

High-Performance ARM Processor
High-Bandwidth On-Chip RAM
High-Bandwidth External Memory Interface
AHB or ASB
DMA Bus Master
Bridge
AHB to APB Bridge or ASB to APB Bridge
APB
UART
Timer
Keypad
PIO

Figure 1: Typical ARM AMBA- bused system

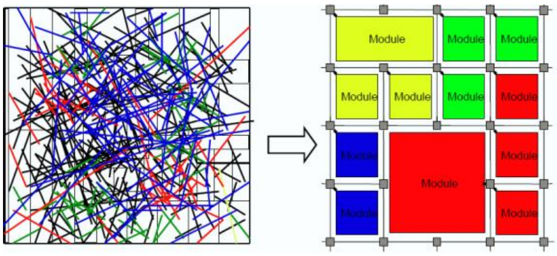


Figure 2: Dedicated signal wires to shared network (NoC)

II. NOC ESSENTIALS

1. Communication by packets of bits
2. Routing of packets through several hops, via switches
3. Parallelism
4. Efficient sharing of wires

A. *Advantages of NoC*:

- NoC eliminates ad-hoc global wire engineering
- NoC separates computation from communication
- NoC supports modularity and reuse of cores
- NoC is a platform for system integration, debugging and testing
- NoC is a scalable platform for billion-transistor chips
- Several driving forces behind it.

B. Original bus features:

- One transaction at a time
- Central Arbiter
- Limited bandwidth
- Synchronous
- Low cost

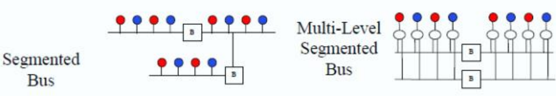


Figure 3: Segmented Bus and Improved version of segmented Bus

C. New features with NoC:

- Versatile bus architectures
- Pipelining capability
- Burst transfer
- Split transactions
- Overlapped arbitration
- Transaction pre-emption and resumption
- Transaction reordering…

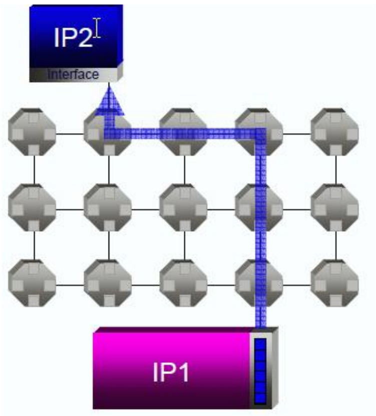


Figure 4: A type of switching to connect two IPs on chip (Wormhole switching)

So, NoC provides different kind of switching technique by which two IPs (Intellectual property) can communicate with each other along with external world.

In this type of switching small buffers and low latency is required next we will discuss example of NoC based ring router by which we can understand the concept of NoC (Network on chip).

III. Victory Bidirectional Ring NoC Router Architecture And Specification

This document specifies the structure and operation of the Victory bidirectional ring network-on-chip (NoC) router, to be used as the building block of a NoC for multi-core processors. The router is a simple router that uses source routing, fixed-size 64-bit packets and a virtual cut-through style of switching. Additionally, it uses two virtual channels per physical channel to provide for deadlock-free routing in the ring. Since it uses virtual cut-through and packets are only 64 bits, that means that all input and output buffers are 64 bits wide. The following sections detail the relevant aspects of the router architecture. [7]

A. External Interface (Signal) Descriptions

Refer to Figure 5 and Table 1, which shows all external signals for the router and describes the signals, respectively. Each unidirectional channel contains a 64-bit data portion and two control signals, send (s) and ready (r), for handshaking. There are three input channels: 1 for the processing element (pe), 1 for the clockwise (cw) direction, and 1 for the counter-clockwise (ccw) direction. Similarly there are three output channels with corresponding designations. The router is a clocked (synchronous) device, so there is also a clock input. The reset is assumed to be synchronous and asserted high. When asserted, the reset signal should initialize all state machines to their idle states and buffer statuses to empty. There is also a polarity signal output which simply indicates if the current clock cycle of the router is odd or even and is used to indicate which virtual channel is being forwarded internally for the current cycle, with the opposite virtual channel being forwarded externally for any clock cycle.

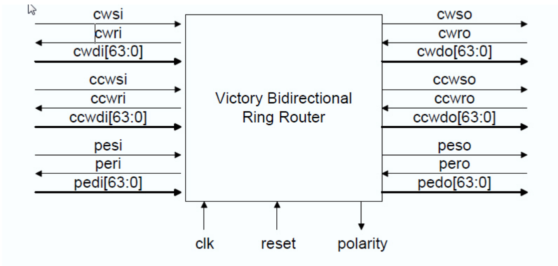


Figure 5: Victory Router External interface

Table I: Signal Name and Type for Victory Router I/O

Signal Name	Signal Type	Bit Width
cwsi	input	1
cwri	output	1
cwdi	input	64
ccwsi	input	1
ccwri	output	1
ccwdi	input	64
pesi	input	1
peri	output	1
pedi	input	64
cwso	output	1
cwro	input	1
cwdo	output	64
ccwso	output	1
ccwro	input	1
ccwdo	output	64
peso	output	1
pero	input	1
pedo	input	64
Clock	Input	1
Reset	input	1
Polarity	output	1

B. Packet Format/Header Processing

Each packet to be routed through a ring of Victory routers is of fixed length of 64 bits. While this is not a realistic packet size (real packets are much larger), most of the principles of NoCs will still be able to be demonstrated. Thus, the packet size equals the flit size equals the phit size equals the channel width. Given such a small packet size, the network can forward the entire packet from the output buffer of one router channel to the input buffer of the next router channel in one cycle. Similarly, assuming no contention, an entire packet can be forwarded from an input channel buffer to an output channel buffer within a router in one cycle. Refer to Figure 2 for a description of the packet format. The most significant 32 bits of the packet are used as header information.

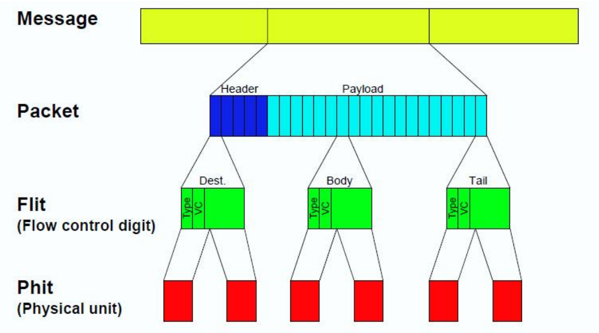
The most significant sixteen bits of this header information are used for routing purposes, with the most significant 2 bits used for the virtual channel polarity and direction, respectively, while the least significant 8-bits of the 16-bit routing field represent a binary hop count (the other 6 bits of the routing info are reserved for potential future use and can always be set to 0). The values for the routing header are set by the source node that injects the packet into the network. However, the hop count field will get decremented at each hop of a packet's traversal. For example, if a packet is to traverse 4 hops, the hop count field should contain 8'h04 when it is first injected on a pe input channel. As it traverses through the network, the hop count will be updated at each router in the following sequence: 8'h03, 8'h02, 8'h01, and finally 8'h00. With this scheme, it is fairly easy to perform routing (or address decoding) when a packet arrives at an input channel. The routing logic simply needs to be able to perform zero-detect on the hop-count field. More info on routing is given in the next section.

The direction bit indicates whether the packet should travel in the clockwise direction (value of 0) or the counter-clockwise direction (value of 1). It is only needed when a packet is first injected into the pe input of a router. The vc field indicates which virtual channel polarity this packet should use: 0 for even polarity and 1 for odd polarity. The 16-bit source field represents the identification number of a packet's source node, that is, the node which injected the packet into the network. Instructions for setting the value of this field will be given in future assignments [7].

Table II: Signal Name and Description for Victory Router I/O

Signal Name	Description
cwsi	Send handshaking signal for the clockwise input channel. When asserted, indicates channel data is a valid packet that should be latched at next rising clk edge into corresponding input channel buffer.
cwri	Ready handshaking signal for the clockwise input channel. When asserted, indicates corresponding input channel buffer is empty and can accept a new packet.
cwdi	Packet data for the clockwise input channel.
ccwsi	Send handshaking signal for the counter-clockwise input channel. When asserted, indicates channel data is a valid packet that should be latched at next rising clk edge into corresponding input channel buffer.
ccwri	Ready handshaking signal for the counter-clockwise input channel. When asserted, indicates corresponding input channel buffer is empty and can accept a new packet .
ccwdi	Packet data for the counter-clockwise input channel.
pesi	Send handshaking signal for the processing element input channel. When asserted, indicates channel data is a valid packet that should be latched at next rising clk edge into corresponding input channel buffer.
peri	Ready handshaking signal for the processing element input channel. When asserted, indicates corresponding input channel buffer is empty and can accept a new packet.
pedi	Packet data for the processing element input channel.
cwso	Send handshaking signal for the clockwise output channel. Asserted when associated output channel has packet data to send and cwro signal is asserted.
cwro	Ready handshaking signal for the

	clockwise output channel. When asserted, indicates neighboring router has space and can accept a new packet
cwdo	Packet data for the clockwise output channel.
ccwso	Send handshaking signal for the counter- clockwise output channel. Asserted when associated output channel has packet data to send and ccwro signal is asserted .
ccwro	Ready handshaking signal for the counter-clockwise output channel. When asserted, indicates neighbouring router has space and can accept a new packet.
ccwdo	Packet data for the counter-clockwise output channel
peso	Send handshaking signal for the processing element output channel. Asserted when associated output channel has packet data to send and pero signal is asserted.
pero	Ready handshaking signal for the processing element output channel. When asserted, indicates processing element has space and can accept a new packet.
pedo	Packet data for the processing element output channel.
Clock	Clock
Reset	Active-high synchronous reset
Polarity	Indicates if current clk cycle is even (0) or odd (1); defined as even during reset, toggles to odd at first rising clk edge after reset is negated, and toggles every cycle thereafter while reset is negated



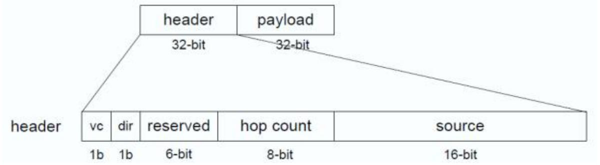


Figure 6: Victory Packet and Header Format

C. Routing/Switching

Recall that the Victory Router is a ring router using two virtual channels per physical channel. More info on how the virtual channels are multiplexed is given later, but for all practical purposes the virtual channels can be thought of simply as two sets of buffers sharing physical channels and control logic externally, and can even share some internal logic with the operation defined in this specification. Basically, one set of virtual channels is operational externally on even clock cycles and the other set of virtual channels is operational externally on odd clock cycles, and vice versa for internal forwarding from input buffers to output buffers. Regardless of virtual channel, when a packet first enters the network on a pe input channel of some router, the routing logic (or address decoder) will first inspect the direction bit to determine if the packet should be routed in the clockwise direction (direction bit equal to 0) or the counter-clockwise direction (direction bit equal to 1). By definition of this router, the hop count field must be non-zero at the time of injection on a pe input channel, i.e., a processing element cannot inject a zero-hop packet through the router to itself. For packets arriving at cw or ccw input channels, the value of the hop count field is inspected to determine whether to continue the packet in the network or not. If the hop count field value is 8'h00, the packet has arrived at its destination and should be routed to the pe output channel at the proper time [7]. Given this routing paradigm and the virtual channel multiplexing scheme, a representation of the internal components and the input channel to output channel switching can be readily determined and is depicted in Figure 7.

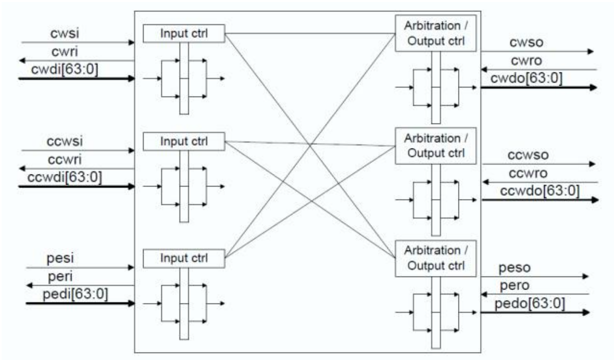


Figure 7: Victory Router internal components and switching

D. Layers of Abstraction in Network Modelling

1) Software layers includes

Ex.: O/S, application

2) Network and transport layers includes

Ex.: Network topology, Switching, Addressing, Routing, Quality of Service, Congestion control, end-to-end flow control.

3) Data link layer

Ex.: Flow control (handshake), Handling of contention, Correction of transmission errors

4) Physical layer

Ex.: Wires, drivers, receivers, repeaters, signalling, circuits [6]

IV. CONCLUSIONS

In this paper, we have presented the significance and requirement of the design of NoCs. Rather than simply enumerating relevant prior work, we have provided a discussion for comparison of BUS and NoC technology. The problems addressed in this work are focused at the architectural-level, while future work can cover the other levels of abstraction for different type of routers. We have also discussed the example of NoC based router (Victory Bidirectional Ring NoC Router) with Architecture and Specification; we can also perform ring router implementation using Verilog (HDL). Such a dynamic companion can transform the ideas of this paper into a valuable in depth analysis for other routers and NoC research.

TABLE II: Comparison of Bus system with NoC for communication purposes

NoC	BUS
Aggregate bandwidth grows	Bandwidth is limited, shared
Concurrent spatial reuse	No concurrency
Link speed unaffected by N	Speed goes down as N grows
Pipelining is built-in	Pipelining is tough
Distributed arbitration	Central arbitration
Separate abstraction layers	No layers of abstraction
However	
No performance guarantee	Fairly simple and familiar
Extra delay in routers Area and power overhead	Communication and
Modules need network interface Unfamiliar methodology	Computation are coupled

ACKNOWLEDGEMENT

I would like to thank **Kapil Kumawat Sir** (HOD E&TC, Sri Balaji college of Engineering, Jaipur) **Harsh Dave** (Technology Analyst and professional skills development member at Infosys Pvt. limited) **Ravi Payal** (Senior Project Engineer at CDAC, Noida) and our anonymous reviewers for providing useful comments on this paper. **Aastha Shrimali** (Scientist) for her feedback on both the writing and analysis related to research paper.

REFERENCES

It may be helpful to read papers about similar routers. While the references below describe such routers, it should be noted that there are differences between the Victory Router and the routers referenced below. In all such cases, this Victory Router architecture specification supersedes any other such info.

[1] http://www.isi.edu/~draper/papers/vlsi04.pdf
[2] http://*www.isi.edu/~draper/papers*/mwscas2000.pdf
[3] Avi Kolodny Technion – Israel *Institute of Technology ,International* Workshop on System Level Interconnect Prediction (SLIP), March 2007
[4] http://en.wikipedia.org/wiki/Network_on_a_chip
[5] A. Jantsch, H. Tenhunen (Eds.). *Networks-on-Chip. Kluwer*, 2003.
[6] http://gram.eng.uci.edu/comp.arch/lab/NoCOverview.htm
[7] EE 577B *Spring 2013 ,University of Southern California Dr. Jeff Draper,Victory chip multi-processors*, 2013.
[8] ITRS, International Technology Roadmap for Semiconductors 2004 Update, 2004.
[9] AMBA, Specification Rev. 2.0, http://www.arm.com, 1999.

[10] Specification for the: WISHBONE System-on-Chip (SoC*)*
Interconnection Architecture for Portable IP Cores, OpenCore, 2002.