Overview of Network on Chip Architecture

Bharati B. Sayankar
Research Scholar,
Dept. of Electronics Engineering
G.H. Raisoni College Of Engineering
Nagpur,India

S.S. Limaye, PhD.
Principal
JTI College of Engineering
Nagpur,India

ABSTRACT

Network-on-Chip (NoC) has been proposed as an alternative to bus-based schemes to achieve high performance and scalability in System-on-Chip (SoC) design. Performance evaluation of On-Chip Interconnect (OCI) architectures is widely based on simulation which becomes computationally expensive, especially for large-scale NoCs. In this paper, we study the various NOC architectures i.e. Virtual Channel Router Design, Wormhole Router Design, Circuit Switched Router Design.

Keywords-Networks-on-Chip (NoCs), Systems-on-Chip (SoCs),throughput, latency.

1. INTRODUCTION

Networks-on-Chip (NoCs) are key components of the emerging Systems-on-Chip (SoCs). As SoCs grow in area, complexity and functionality, so do their communication requirements in terms of performance (latency and throughput) and number of interconnected components. Reducing NoC latency is crucial for SoC performance, since it is introduced to every communication pair within the SoC. Latency may become vital in the case of real-time SoCs. It may also play an especially important role in the case of processor units communicating with other processor units, local memory, shared memory or cache blocks.

2. BACKGROUND

In this section, we discuss the basic background for NoC architectures and provide a review of some related works. The function of an on-chip network is to deliver messages from source node to destination node, and there are many design alternatives to accomplish this job. Depending on the application requirements, how to choose a suitable network architecture remains an open problem for research. Here we discuss the network properties that need to be considered when devising an NoC architecture for specific applications.

1) Switching Technique

There are two major switching techniques: circuit switching and packet switching. Circuit switching establishes a link between source and destination node either virtually or physically before a message is being transferred. The link is held until all the data are transmitted. Major advantages of circuit switching are that there is no contention delay during message transmission and its behavior is more predictable, so circuit switching is usually employed when Quality of Service (QoS) is considered.

On the other hand, packet switching transfers messages on a per-hop basis. With packet switching, messages are divided into packets at the source node and then sent into a network. Packets move along a route determined by the routing algorithm and traverse through a series of network nodes and finally arrive at the destination node. Packet switching is utilized in most of NoC platforms because of its potential for providing simultaneous data communication between many source-destination pairs. It can be further classified into three classes: store and forward (SAF), virtual cut through (VCT), and wormhole switching. The most commonly used approach for an NoC architecture is wormhole switching because it only requires a buffer size of one transmission unit called flit so that the area cost of a router can be kept low. In contrast, SAF and VCT require a buffer size equivalent to the whole packet which prohibits their adoption.

2) Topology Development

Topology defines how nodes are placed and connected, affecting the bandwidth and latency of a network. Many different topologies have been proposed such as mesh, torus, mixed and custom topology.

3) Routing Policy

Routing is the mechanism responsible for determining the path that a packet traverses from the source node to the destination node. Routing algorithms such as deterministic and adaptive ones have been proposed. With deterministic routing, the path between source-destination pair is fixed, regardless of the current state of the network. On the other hand, an adaptive routing algorithm takes the network state into account when deciding a route,resulting in variation of the routing path with time. For example, it may choose an alternative path when congestion occurs. This explains why it has the potential of supporting more traffic for the same network topology.

3. ON-CHIP NETWORKS

The move to on-chip networks has several motivations which are similar to those that drove off-chip networks. The design characteristics of on-chip networks however differ in multiple ways.

A. Motivation

The latency and electrical behavior of long wires scale poorly with diminished feature sizes due to a smaller crosssectional area. On-chip buss speeds are at a distinct scaling disadvantage because they connect components spread across the chip. In addition, multi-drop busses require protocols to ensure exclusivity among the transmitters and suffer from poor electrical behavior as long wires on-chip begin to look more like transmission lines. On-chip networks enjoy a scaling advantage relative to busses since network wire lengths between adjacent routers can be kept short and uni-directional. On-chip networks also enable the pipelining of data and a much greater aggregate bandwidth than busses. Finally, design complexity can be reduced since the router

only needs to be designed once and replicated for use wherever needed.

B. Design Characteristics

Bandwidth: The bandwidth of off-chip networks is typically much lower than on-chip networks. Off-chip networks are constrained in bit width by the expense of each chip pin. On-chip networks' wires are constrained by the number of metal layers and pitch of on-chip wire routing, allowing onchip networks to have a much higher bandwidth than their off-chip counterparts. The greater bit-width allows the packet length of an on-chip network to be much shorter for the same amount of data, compared to an off-chip network. These differences affect the optimum choice of routing algorithm and network topology for on-chip networks.

Latency: In off-chip networks, a router on one chip will be connected by board traces to a router on another chip.

Differences in wire length and chip placements create significant clock and data skew between chips in the same network, therefore off-chip networks typically resynchronize data at each router. Synchronization adds two to three network cycles of latency per hop as a result. Off-chip networks run at a lower frequency than the rest of the chip, compounding the latency required for synchronization. Data must also be resynchronized upon arrival at the destination chip. By contrast, on-chip networks can be designed to have only one cycle per hop because synchronization is not needed. Single hop routing delays greatly decrease the end-to-end latency of the packets on the network relative to off-chip networks.

Timing: Off-chip networks typically are clocked at much lower frequencies than the processor's main clock because their timing is dominated by transmission line capacitances and the relative skews of off-chip interconnect. On the other hand, on-chip networks can be designed to be clocked by the main processor clock because the wire lengths are much shorter and the relative data skews are minimal. Keeping the frequency up places a strong constraint on how much logic may be placed on the router's critical path prior to launching the flit to the next router.

Area: Area is not a strong constraint for off-chip networks because there is typically only one off-chip router per chip. In on-chip networks, depending on the granularity of the network, the routers may take up a significant fraction of the total die area, constraining the area allowed for buffering and therefore affecting the number of virtual channels and the bit width of the network.

4. NETWORK-ON-CHIP ARCHITECTURE

Intercommunication requirements of SoCs made of hundreds of cores will not be feasible using shared bus or a hierarchy of buses due to their poor scalability with system size and their shared bandwidth among all the attached cores. Network-on-Chip (NoC) has been proposed as a promising replacement for buses and dedicated interconnections to solve the scalability and complexity problem . NoCs involve the design of network interfaces to access the on-chip network, the selection of the suitable protocols and topologies of switches to transport the data. The design goals for NoCs can be described as

i. Platform based design

ii. Separation between communication and computing resources

iii. Minimization in energy and area

A platform based design is essential for modular network.

Network can be made reusable by separating the communication infrastructure from computing resources. A lot of research is going on to develop appropriate network architectures to meet the requirements.

New flexible and configurable communication channel architectures need to be identified. These communication channels will not form dedicated buses as currently implemented on-chip, due to noise, scalability and speed constraints. Thus, the overall communication scheme will resemble more computer networking than traditional bus based design. Here we are proposing the PDN Network-on-Chip architecture which provides optimum bandwidth utilization and at the most two hops in the communication from any node to other node.

5. DESIGN OF NOC's

This Section provides a brief overview of the network architectures used. All the networks were based on a meshtopology with 5-input×5-output port routers, with 4 ports connecting to the neighbouring routers and the fifth one connecting to a local computation tile. All flits provided a 64-bit data payload size, with additional control bits necessary for the WH and SpecVC designs. The WH and SpecVC networks also utilised a static, dimension-ordered, XY routing scheme.

1. *Circuit Switched Router Design*

The CS router provided a very simple data-path, being composed only of a crossbar with registered outputs. Each output port is 64-bits wide, since no control data is necessary. To provide more flexibility, each 64-bit output port is split into four, 16-bit wide, lanes. Given the 5-port design, 20 input and output lanes therefore exist. Each input lane can then be connected to any output lane apart from the ones on the same side of the router (i.e. no flit U-turns are allowed), using the 16×20 crossbar. The configuration memory therefore provides a 20 entry capacity (1 for each output lane), with 5-bits per entry (4 address bits to identify an input lane and 1 valid bit). The splitting of a 64-bit flit into 16-bit units for transport over the network means that a serialising and deserialising unit is necessary at the tile interface of the router.

The completely static nature of the CS network means that a separate control network is necessary to provide all circuit set-up and tear-down functions. To model a scalable solution for this, a very simple packet switched network, roughly based on wormhole flow control was provided. All experiments then considered both the circuit-switched and packet-switched routers, to account for the necessary overhead of the packet-switched network. Figure 1 shows the complete structure of the CS router.

2. *Virtual Channel Router Design*

The SpecVC router provides for single cycle flit forwarding by utilising look ahead routing and speculative VC and crossbar allocation. A conventional input-queued architecture with 4 VCs per port and 4 flit deep buffers for each VC were also used. Associated head pointer and tail pointer registers then referred to the first and last item in the FIFO respectively. Each flit identifies its VC by using a one hot encoded 4-bit VC identifier. The look ahead routing scheme

means that each flit additionally carries a 5-bit, one hot encoded next port identifier used by the downstream router. Moreover, the architecture does not provide for a separate head flit and every flit therefore identifies its destination X and Y address (4-bits each) and carries an additional single bit to indicate whether it's a tail flit or not. Combining with the 64-bit data-path results in a total flit size of 82-bits. Both the VC and switch allocators are based on matrix arbiters and can allocate VCs and crossbar ports speculatively for the next clock cycle if necessary. Since both crossbar and link traversal are performed in a single clock cycle, in the best case, an incoming flit finds pre-allocated resources and can thus be forwarded to the next hop in a single clock cycle. State-holding elements indicate which VCs are currently being used, to ensure they are not re-considered for allocation.

A passing tail flit then de-allocates this resource. Additionally a simple stop-go flow control method is utilized to prevent buffer overflow, where a single buffer nearly full signal is output by each input VC to the corresponding upstream VC. Figure 1 shows the structure of this router.

3. *Wormhole Router Design*

The WH router can be considered to be a simplified version of the SpecVC router. To represent a comparable design to the SpecVC router the techniques of look ahead routing and combining the crossbar and inter-router link traversal into a single stage were again utilised. Removing speculation from any allocation then naturally resulted in a two stage pipeline where the switch allocator, again based on matrix arbiters, grants an output port in the first pipeline stage, with crossbar and link traversal occurring in the second stage. As in the SpecVC router, no separate head flit was included and thus, besides the data payload, each flit carries a 5-bit, one hot encoded next-port identifier, X and Y destination addresses (2-bits each) and 1-bit to identify tail flits. Combining with the 64-bit data-path results in a total flit size of 74-bits. In this simple implementation, an additional pipeline stage separation register is provided between the input FIFOs and the crossbar. For the crossbar traversal stage the flit at the head of the FIFO is loaded into this register, which drives it across the rest of the data-path. Since in a wormhole flow control mechanism allocated output ports are not shared amongst separate input ports, state holding elements identify which output ports are currently being used by which input port. Passing tail flits then de-allocate this resource. A similar stop-go flow control, as in the Spec VC design, is also used. Figure 1 shows the structure of this router.

6. CONCLUSION

Out of the three NOC Architecture Virtual Channel Router Design is the most efficient than XY and WH router architectures because of its features.

7. FUTURE WORK

An adaptive routing algorithm should be investigated to provide NOC with more flexibility. NoC needs to be evaluated in a complete system under various workloads and demands.

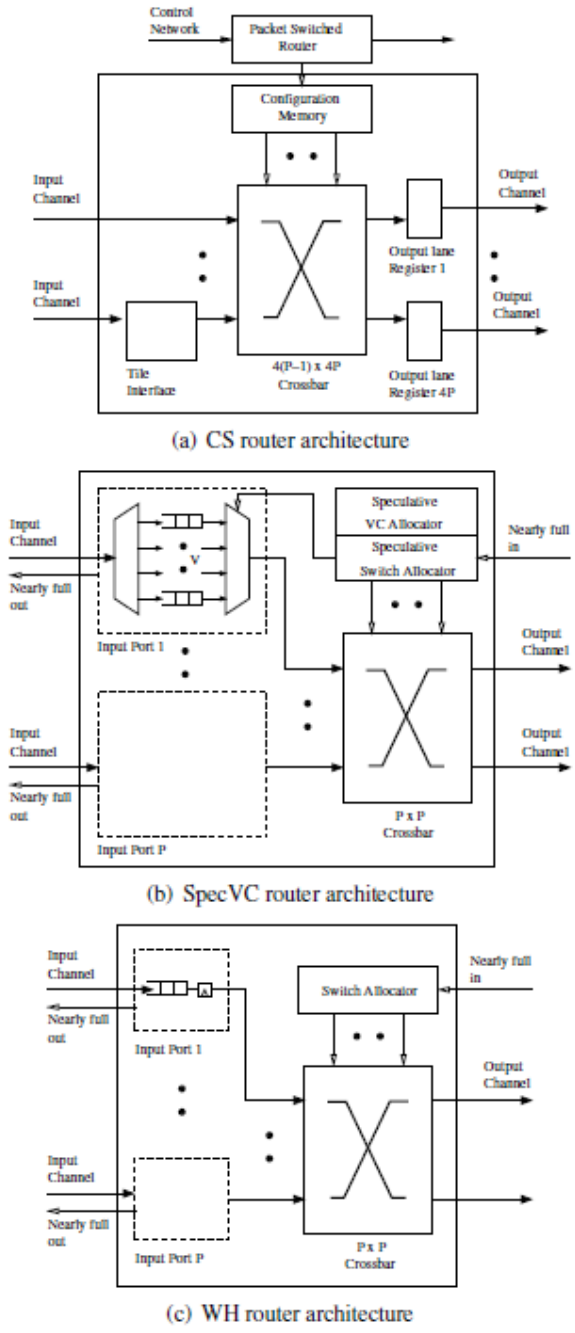

Figure 1. Router architectures used

8. REFERENCES

[1] Arnab Banerjee, Robert Mullins and Simon Moore, A Power and Energy Exploration of Network-on-Chip Architectures,Proceedings of the First International Symposium on Networks-on-Chip (NOCS'07)

[2] Mahendra Gaikwad, Rajendra Patrikar, Abhay Gandhi, Energy-aware Network-on-Chip architecture using Perfect Difference Network, 2010 IEEE

[3] Arnab Banerjee, Student Member, IEEE, Pascal T. Wolkotte, Member, IEEE, An Energy and Performance Exploration of Network-on-Chip Architectures, IEEE TRANSACTIONS ON VERY LARGE SCALE

INTEGRATION (VLSI) SYSTEMS, VOL. 17, NO. 3, MARCH 2009

[4] Chifeng Wang, Wen-Hsiang Hu, Nader Bagherzadeh, Area and Power-efficient Innovative Network-on-Chip Architecurte. 2010 IEEE

[5] George Michelogiannakis, Dionisios Pnevmatikatos, ManolisKa tevenis, Approaching Ideal NoC Latency with Preconfigured Routes, Copyright IEEE 2007 - to appear in Proceedings of NOCS 2007, Princeton, NJ, USA, May 7 -9, 2007