ASYNCHRONOUS BYPASS CHANNELS IMPROVING PERFORMANCE IN MULTI-SYNCHRONOUS NETWORK-ON-CHIPS

A Thesis

by

TUSHAR NAVEEN KUMAR JAIN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2010

Major Subject: Computer Engineering

ASYNCHRONOUS BYPASS CHANNELS IMPROVING PERFORMANCE IN MULTI-SYNCHRONOUS NETWORK-ON-CHIPS

A Thesis

by

TUSHAR NAVEEN KUMAR JAIN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Co-Chairs of Committee,	Alex Sprintson
	Paul V. Gratz
Committee Member,	Eun Jung Kim
Head of Department,	Costas N. Georghiades

August 2010

Major Subject: Computer Engineering

ABSTRACT

Asynchronous Bypass Channels

Improving Performance in Multi-synchronous Network-on-chips. (August 2010)

Tushar Naveen Kumar Jain, B.Tech, Indian Institute of Technology, Roorkee

Co-Chairs of Advisory Committee: Dr. Alex Sprintson
Dr. Paul Gratz

Dr. Paul V. Gratz Network-on-Chip (NoC) designs have emerged as a replacement for traditional shared-bus designs for on-chip communications. As with all current VLSI design, however, reducing power consumption in NoCs is a critical challenge. One approach to reduce power is to dynamically scale the voltage and frequency of each network node or groups of nodes (DVFS). Another approach to reduce power consumption is to replace the balanced clock tree with a globally-asynchronous, locally-synchronous (GALS) clocking scheme. NoCs implemented with either of these schemes, however, tend to have high latencies as packets must be synchronized at the intermediate nodes between source and destination. In this work, we propose a novel router microarchitecture which offers superior performance versus typical synchronizing router designs. Our approach features Asynchronous Bypass Channels (ABCs) at intermediate nodes thus avoiding synchronization delay. We also propose a new network topology and routing algorithm that leverage the advantages of the bypass channel offered by our router design. Our experiments show that our design improves the performance of a conventional synchronizing design with similar resources by up to 26% at low loads and increases saturation throughput by up to 50%.

To my family and friends

ACKNOWLEDGMENTS

I would like to thank my co-advisors and committee co-chairs, Dr. Alex Sprintson and Dr. Paul V. Gratz, for their constant support and guidance in my research. Dr. Sprintson, with his expert guidance, was instrumental in getting me started on an initial study which paved the way for my thesis topic. Dr. Gratz, with his expertise in NoC, helped me in quickly ramping up on this topic and always kept me focused on the right step to carry my work forward. In Dr. Sprintson and Dr. Gratz, I found my ideal advisors. Without their expert advice and numerous insightful comments over the past year, it would have been virtually impossible to finish this body of work. As I proceed in my career, I hope to take the teachings and experiences accumulated from working with them.

I would also like to acknowledge my other committee member, Dr. Eun Jung Kim, for her support throughout the course of this research. Special thanks to Dr. Gwan Choi, who along with Dr. Gratz, gave several insightful comments as the co-author of the initial work on ABCs.

Thanks also go to all my friends, colleagues and CAMSIN group members for their encouragement and intellectual discussions which helped me shape this work.

Finally, thanks to my department faculty and staff, external faculty and campus employers for making my time at Texas A&M University a great experience.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION

With the continued advance of Moore's Law, ever increasing transistor densities are yielding ever greater numbers of processing elements (PEs) on chip, resulting in the current proliferation of chip-multiprocessor (CMP) and system-on-chip (SoC) designs. As the numbers of PEs increases, the communication between those components is becoming ever more critical. Networks-on-Chip (NoCs) have recently emerged as a scalable alternative to the bus-based and ad-hoc interconnect seen in past designs. NoCs leverage the design techniques of multi-hop interconnection networks developed for the large scale multiprocessor computers of the 1990s, to provide low-latency, scalable communication between PEs on chip.

Although Moore's law continues to provide greater transistor densities, current generation VLSI presents several challenges going forward. In particular, balancing performance against the energy and power consumption in a given design is a challenge. One increasingly commonly used method to achieve this balance is dynamic voltage and frequency scaling (DVFS) [1]. In CMP and SoC designs which support DVFS, the voltage and frequency of individual PE nodes, or sets of PE nodes is dynamically managed to reduce overall power and energy consumption while maintaining enough performance to meet application demands [2].

Another common approach to reduce the power and energy consumption in CMPs and SoCs is the use of a globally-asynchronous, locally synchronous (GALS) clocking scheme [3]. In traditional VLSI design, a single synchronized clock is maintained throughout the entire chip. These synchronized architectures require fully balanced clock distribution trees to ensure minimal clock skew between communicating components. Fully balanced clock distribution trees, however, in addition to

This thesis follows the style of *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*.

requiring additional effort during the design stage also consume a significant portion of the total chip power which can be as high as 30% of the total power consumed by the chip [3]. On-chip power consumption can be reduced by replacing the balanced clock tree with a GALS clocking scheme which only guarantees minimal clock skew within the local processing element.

PEs in both DVFS and GALS systems therefore are either mesochronous – of the same frequency but unrelated phase, or plesiochronous – of differing frequency and phase, to one-another. Communication between nodes which do not have the same frequency and phase typically requires synchronization into the receiver's clock domain in order to avoid metastability problems caused by violations of setup or hold time of the flip-flops when latching data from a different clock domain [4]. A well established approach to a avoiding metastability involves adding multiple stages of latching on incoming signals to reduce the likelihood of a metastable event. While this approach allows reliable communication across clock domains, it comes at the cost of an added two or three cycles of latency for every clock domain traversal. As the number of CMP and SoC PE nodes increases, the number of clock domains which data must traverse between source and destination will increase, thereby increasing the effective latency of that communication. This latency has been shown to directly impact the system performance and the overall cost [5].

The objective of the thesis is to present a network architecture which offers low network latency solutions. With low communication latency as our objective, we propose a new router microarchitecture for mesochronous and plesiochronous NoCs which avoids synchronization delay at the intermediate nodes via an asynchronous by-pass channel (ABC) around these nodes. We also propose a novel 2-D grid based topology and routing algorithm to complement our router design.

Thus the main contributions of this work are:

- We present the ABC router design, a novel router design which offers a highly skew tolerant solution to DVFS and GALS NoCs.
- ABC routers offer a low latency, asynchronous bypass path through the intermediate routers, avoiding synchronization delay and latching at intermediate nodes.
- We also present an optimized network topology and routing algorithm to leverage the advantages of ABC routers.

The rest of the work is organized as follows. This chapter further presents a background on NoCs and the motivation for ABCs. Chapter II gives a detailed description of the network. Chapter III proposes an efficient topology for the interconnected networks. In Chapter IV, we present the experimental results and compare the performance of our router against a comparable baseline router. Chapter V discusses the related work. Chapter VI concludes the work.

I.1. Network on Chips

One of the major problems in future SOC designs arises from non-scalable delays in global wires[6]. Global wires carry signals across a chip and typically do not scale in length with technology scaling[7]. Though gate delays scale down with technology, global wire delays typically increase exponentially or linearly by inserting repeaters. Even after repeater insertion[7], the delay may exceed the limit of one or multiple clock cycles. In ultra-deep submicron processes, it is claimed that 80 percent or more of the delay of critical paths will be due to interconnects[8]. As a result, many large designs use as hoc FIFO buffers to synchronously propagate data over large distances to overcome this problem.

The most frequently used on-chip interconnect architecture is an arbitrated bus, where all communication devices share the same transmission medium. Advantages of such shared-bus architectures are simple topology, low area and extensibility. However, for a long bus line, the intrinsic parasitic resistance and capacitance can be quite high. Moreover, every additional IP block added to the bus adds to this parasitic capacitance, in turn increasing propagation delay. As the bus length and/or the number of IP blocks increases, the associated bit transfer delay over the bus becomes large and will eventually exceeds the targeted clock period. This places a limit on the number of IP blocks that can be connected to a bus and thereby limits the system scalability[9].

One solution to deal with this problem is to use a network-centric approach, where, communication between IPs happens in the form of packets. A common characteristic of such architectures is that the IP blocks communicate with each other using intelligent switches or routers. As such, these switches dubbed infrastructure IPs (I2Ps)[10] provide a robust data transfer medium for the functional IP modules.

I.2. Dynamic Voltage Frequency Scaling

One active area of work on NoC has focused on dynamically varying operating voltage and frequency levels to achieve a balance between power and performance[11]. This technique, referred to as DVFS, is used quite often in SoC designs[12]. Dynamic voltage and frequency scaling (DVFS) was introduced in the 90s to dramatically reduce power consumption in large digital systems by varying both voltage and frequency of the system with respect to changing workloads[13].

Alternative techniques using voltage/frequency islands (VFIs) for IP blocks are used in achieving fine grain system-level power management. Use of VFIs in the NoC context can provide better power-performance tradeoffs than single voltage, single clock frequency case as it benefits from the natural partitioning and mapping of applications onto the NoC platform. Despite the huge potential for energy savings with VFIs, the NoC design methodologies considered so far are limited to a single voltage-clock domain[14][15]. Studies that do consider multiple VFIs assume that each module/core in the design belongs to a different island and different islands are connected by point-to-point (P2P) links[16].

I.3. Motivation behind Asynchronous Bypass Channels

The intuition behind the ABC design is that a packet should not be required to latch at the intermediate nodes within an NoC. At a minimum, intermediate node latching results in one cycle of overhead for synchronous designs and at least two cycles of overhead for mesochronous and plesiochronous designs which require synchronization. If intermediate node latching can be avoided, a significant fraction of the packet latency overhead can be removed. This is because, in doing so, we could synchronize the control signals in the respective clock domains only when necessary and avoid synchronizing the data signals.

In the absence of network congestion, a packet should not, in-fact, require latching at intermediate nodes. While at high congestion, when multiple packets are vying for the same buffer FIFO and output port resources, the packet must be latched at the intermediate nodes as it waits for the contention to clear. At low congestion levels, however there is no such constraint.

As we will show, the ABC design offers an asynchronous bypass path which bypasses the buffer FIFOs as well as synchronization at the intermediate nodes. Thus the packets can virtually fly through the network without getting latched at the intermediate nodes, reaching there destination with a delay approaching the wire delay of the connecting links.

CHAPTER II

NETWORK OPERATION

On its path through the network a packet traverses three node classes, the source node, intermediate nodes and the destination node. In a typical GALS or DVFS NoC each of these nodes may operate in its own clock domain. The goal of the ABC router design is to reduce or remove latching and synchronization at the intermediate nodes. Furthermore, the routers should independently and dynamically determine the availability of the bypass channels without the overheads of allocation or setup and tear-down. In this section we discuss the design and operation of the ABC router to show how these goals were met.

II.1. ABC Network Architecture

As illustrated in the packet propagation path shown Figure II.1, we observe that packets traversing a given intermediate node router are most likely to travel straight, that is from input port on one side of the router to output port on the opposite side. In a typical 2D Mesh network, a minimum length path between any two nodes may be found which has at most one turn. We propose to leverage this observation by biasing our ABC routers towards the straight path over the turn path. ABC routers are pre-disposed to allow the asynchronous propagation of packets along the straight path, as long as conditions allow.

In 2D Mesh networks, only a limited set of source-destination pairs require no turn during packet traversal, those pairs which are in the same column or row. We observe that it should be possible to gain a further benefit from the ABC router's straight path bias if the network topology provides a greater number source-destination pairs which do not require a turn. To this end, we propose a new class of network topol-

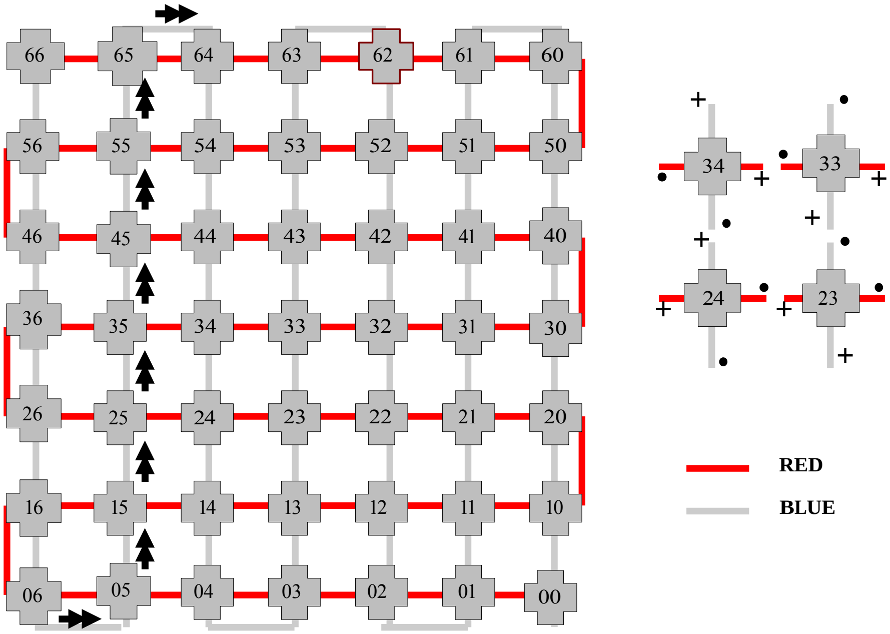
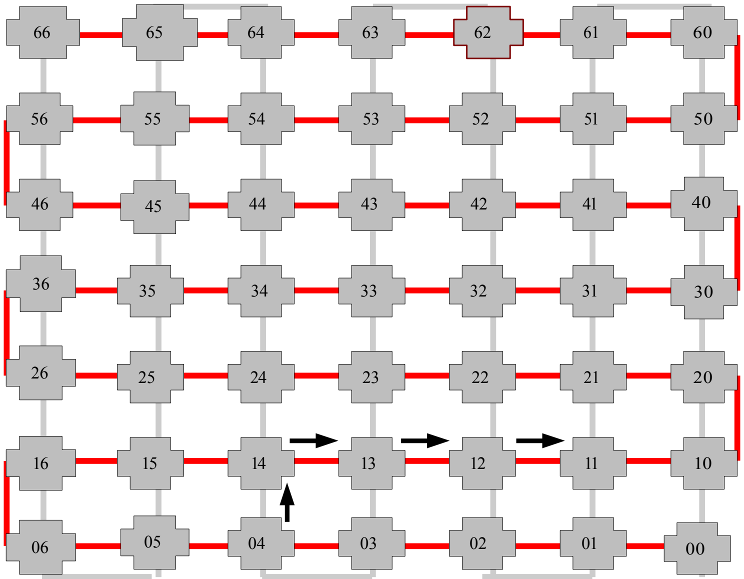


Fig. II.1.: An ABC network topology showing red and blue chains. Also depicts path traversed between (0,6) and (6,4).

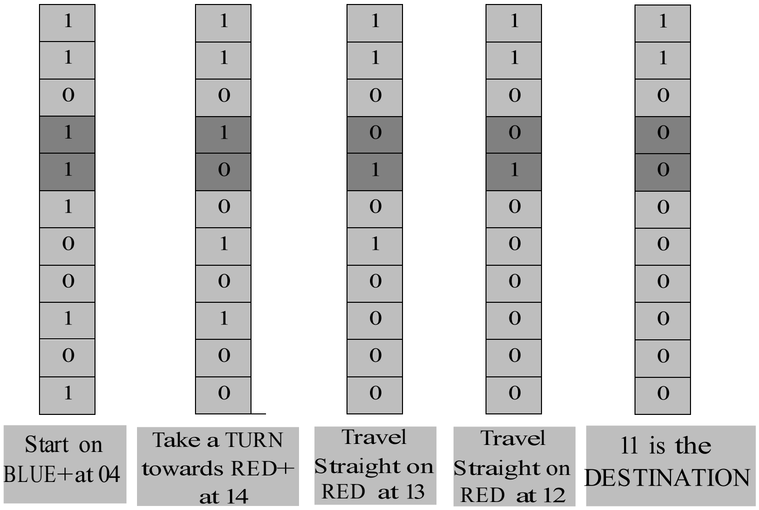
ogy, the double-chain topology, one of which is shown in Figure II.1. A double-chain topology is comprised of two disjoint but overlapping chains, each of which connects all network nodes. Double-chain topologies provide an advantage over 2D Mesh networks for ABC routers by providing two paths comprised solely of “straight-path” links between all source-destination pairs.

As the traditional 2D mesh terminology of North, South, East and West is less useful in a double-chain topology, we will label these chains the red chain and a blue chain respectively. As shown in Figure II.2, nodes within each chain is perceived to be ordered such that traversing a given chain in one direction is considered to be increasing (e.g. the red+ direction) and traversing that chain in the opposite direction



(a) Path of a packet generated at (0,4) for (1,1).

HEADER FLIT DURING TRAVERSAL FROM 04 to 11



Start on BLUE+at 04	Take a TURN towards RED+ at 14	Travel Straight on RED at 13	Travel Straight on RED at 12	11 is the DESTINATION
1	1	1	1	1
1	1	1	1	1
0	0	0	0	0
1	1	0	0	0
1	0	1	1	0
1	0	0	0	0
0	1	1	0	0
0	0	0	0	0
1	1	0	0	0
0	0	0	0	0
1	0	0	0	0

(b) Most significant bits of the header flit for the packet shown in Figure II.2(a) flowing through the network.

Fig. II.2.: Updating the header flit during flight.

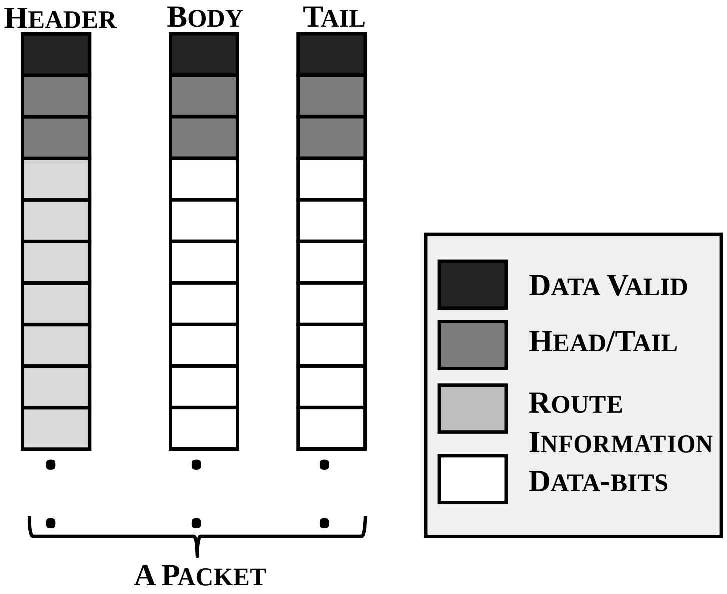


Fig. II.3.: Subdivision of packet into flits and different bit-fields inside a flit.

is considered to be decreasing (e.g. the red- direction) as shown in the Figure. II.2. Jumping from one chain to other is referred to as a turn. To avoid deadlock in the network, the chains themselves are ordered, and a jump is allowed only from the blue chain to the red chain and not the vice-versa.

II.2. Packet Structure

ABC packets are deterministically source routed, i.e. they have a full set of routing directions encoded into the header, as shown in Figure II.2. Source routing is used to ensure a minimal delay and skew induced at each hop. As the figure shows, the three most significant bits of each flit contain flit type (Header, Body or Tail) and a valid bit. This necessities the need to store the whole route information in the header flit at the source node itself. However, for the network on chip applications, the overhead is tolerable. The fourth and fifth most significant bits of the header flit encode the

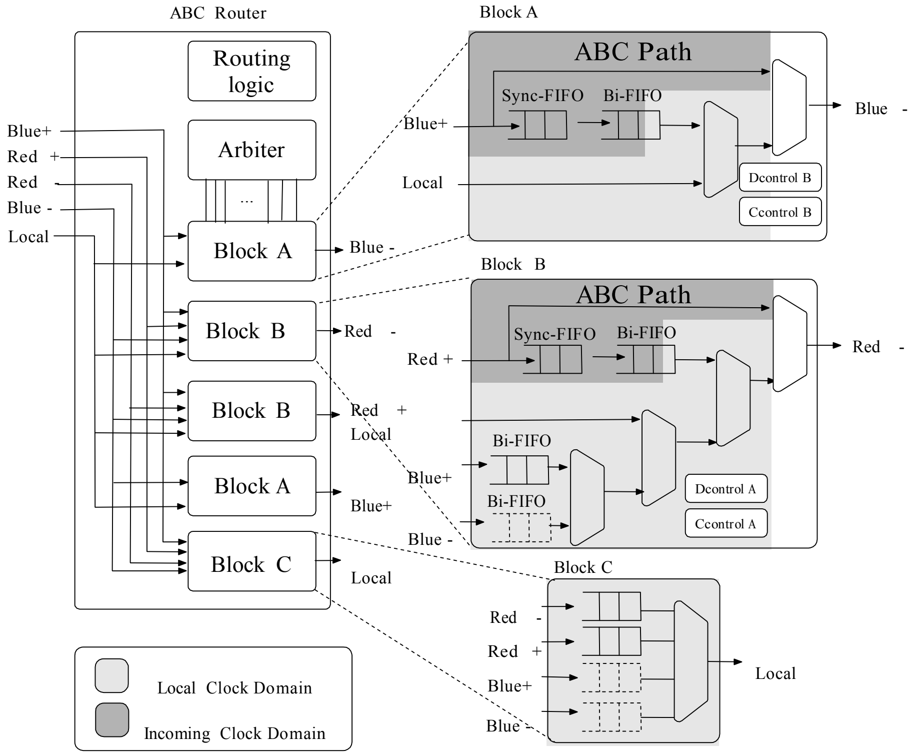


Fig. II.4.: ABC router microarchitecture design.

routing directions for the packet at that particular node. Figure II.3 depicts the bitwise break up of all the three kind of flits. After each hop, the current routing bits are shifted away until only a flag is left which marks arrival at the destination as depicted in Figure II.2. Since the static shift operation can be achieved without any logic, we are able to retrieve the next hop information at the node without impacting the skew between the flits and its clock signal.

II.3. Router Microarchitecture

Figure II.4 shows a high-level block diagram of the proposed ABC router. Similar to a typical 2D Mesh NoC router, the ABC router contains five input and output ports corresponding to the four neighboring directions and the local processing element (PE). All the data communications through the network are carried out in the form of packets, subdivided into flits, which are sent through the NoC using wormhole flow-control. The ABC router design is source-synchronous, hence each port's link is comprised of a clock bit along with data bits and an on/off flow control bit. Both the data and clock signals travel same link length and router logic, therefore they face nearly the same delay and their skew does not increase much for each hop traversed. Nevertheless, after place and route, if an unexpected skew is observed between the clock signal and the flits, transparent latches or buffers may be inserted to reduce it, at the cost of some added latency.

II.3.1. Router Datapath

As shown in Figure II.4, the ABC router contains five output unit blocks corresponding to each direction. These output unit blocks are broken into three types. Output block "A" connects the blue (+ or -) and local inputs to the opposite blue chain output, while output block "B" connects the all the possible inputs to the corresponding red output. Block "C" connects all possible inputs to the router's local PE. Block "A" has a smaller set of inputs because it is illegal to route from a red input to a blue output as discussed in Section II.1.

At the intermediate nodes, if a packet travels straight through the router, it is desirable that the packet be forwarded without latching. To this end, output blocks "A" and "B" both contain an "ABC Path" which the router is pre-disposed to use in the absence of congestion. When an incoming flit arrives, it is simultaneously latched into the synchronous FIFO (sync-FIFO) and propagated along the ABC path towards the output mux. In the absence of congestion, indicated by a lack of packets currently waiting for the output port and the presence of favorable downstream control flow, the flit is propagated through the output port mux via the ABC and the contents of the sync-FIFO are flushed away. Alternately, if the buffer FIFOs are not empty, indicating presence of older packets, the contents of the sync-FIFO are forwarded to the straight path bi-synchronous FIFOs [17] (bi-FIFOs) and the output port is fed by one of the bi-FIFOs. Thus, the sync-FIFO acts as a backup in case the flit was not able to successfully utilize ABC. Alternately, if the packet needs to turn at that particular node, the flits are latched into one of the turn path bi-FIFOs.

Although there are no buffer FIFOs along the ABC path, we employ bi-synchronous FIFOs (bi-FIFOs)[17] to store packets in the event of congestion. Bi-FIFOs allow writes and reads to be done in different clock domains, although they require an overhead of two to three clock cycles to avoid metastability. In the ABC routers, bi-FIFO writes are done in the incoming clock domain and reads are done in the local clock domain. Overall the complete router has 10 bi-FIFOs, two of which are shared between block “C” and the two block “B”s.

The path joining a given input port with the same color output port (e.g. red- to red+), is referred to as a “straight-path”. All other paths are referred to as turn paths as shown in Figure II.4. Each turn path is made up of a single bi-FIFO. The sync-FIFOs are standard FIFOs which are clocked in the incoming clock domain. In the current ABC router design, the ABC is only a bypass around the straight path FIFOs, as shown in Figure. II.4. We considered providing ABC paths for the turns as well, the approach was rejected however, due to exponential increase in FIFOs required to implement it.

ABC routers employ on/off flow control. In this scheme, when the number of free buffer FIFOs in a direction falls below a threshold value the node transmits an off signal to the upstream node connected to it through the on/off link. Similarly, an on signal is transmitted when the number of free buffer FIFOs exceeds the threshold.

This signal needs to be synchronized with respect to the outgoing clock of the upstream node. Thus, summing up the round trip latency, which is nearly 1.5 times the clock cycle as discussed in Section III and two cycles for synchronization, to ensure no flit gets dropped in the process, the depth of the FIFO must be overcommitted by at least four more than the maximum packet length.

II.3.2. Router Arbiter

As shown in Figure. II.4, the output channel is shared between ABC, the straight path bi-FIFO and turn path bi-FIFOs. To avoid livelock and ensure routing fairness the arbiter selects the ABC path in the ABC mode, and arbitrates among the bi-FIFOs according to the Round Robin algorithm. However, on transition from ABC mode to FIFO mode, the straight path bi-FIFO is given a priority.

As shown in Figure. II.4, the router arbiter is divided into two types of control modules for each output unit block, the clock control module (Ccontrol) and the data control module (Dcontrol). The Ccontrol module arbitrates the output clock while the Dcontrol module controls the output data. As the Ccontrol and Dcontrol modules receive inputs from two different clock domains, metastability must be addressed. We avoid all metastable conditions through control logic synchronization. Further detail on metastability avoidance is given in Section II.3.3.

On arrival of a head flit at the output, the router becomes active and remains in this state until a tail flit arrives, when it becomes idle. Arbitration occurs in the idle state, except if an off signal is received from the downstream node while the ABC is in use in which case the chosen path is switched to the straight path bi-FIFO.

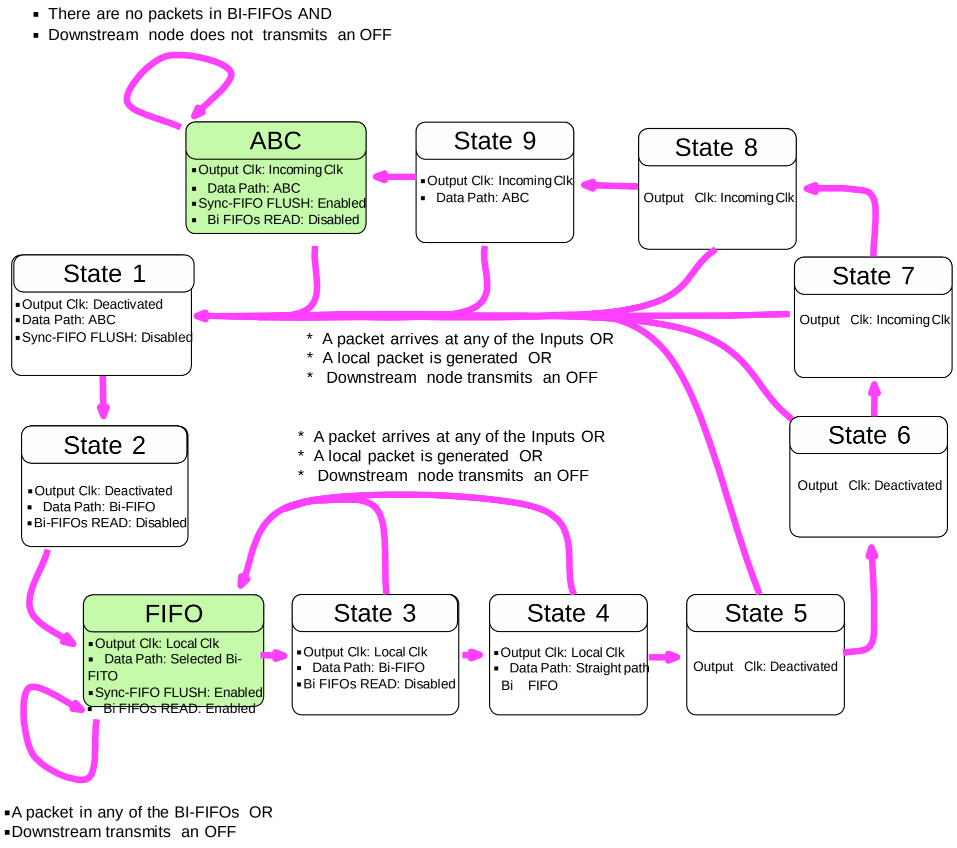

▪ There are no packets in BI-FIFOs AND
▪ Downstream node does not transmits an OFF
ABC
▪Output Clk: Incoming Clk
▪ Data Path: ABC
▪Sync-FIFO FLUSH: Enabled
▪ Bi FIFOs READ: Disabled
State 9
▪Output Clk: Incoming Clk
▪ Data Path: ABC
State 8
Output Clk: Incoming Clk
State 1
▪Output Clk: Deactivated
▪Data Path: ABC
▪Sync-FIFO FLUSH: Disabled
State 7
Output Clk: Incoming Clk
* A packet arrives at any of the Inputs OR
* A local packet is generated OR
* Downstream node transmits an OFF
State 2
▪Output Clk: Deactivated
▪ Data Path: Bi-FIFO
▪Bi-FIFOs READ: Disabled
* A packet arrives at any of the Inputs OR
* A local packet is generated OR
* Downstream node transmits an OFF
State 6
Output Clk: Deactivated
FIFO
▪Output Clk: Local Clk
▪ Data Path: Selected Bi-FITO
▪Sync-FIFO FLUSH: Enabled
▪ Bi FIFOs READ: Enabled
State 3
▪Output Clk: Local Clk
▪ Data Path: Bi-FIFO
▪Bi FIFOs READ: Disabled
State 4
▪Output Clk: Local Clk
▪ Data Path: Straight path Bi FIFO
State 5
Output Clk: Deactivated
▪A packet in any of the BI-FIFOs OR
▪Downstream transmits an OFF

Fig. II.5.: FSM depicting the working of the router.

II.3.3. Control Logic

The control logic in an ABC router can be represented as a Moore finite state machine (FSM). The various states of this FSM are displayed in Figure II.5.

The ABC router operates in one of two modes: ABC mode where flits are forwarded along with the upstream or the incoming clock, and FIFO mode where incoming packets are synchronized into the local clock domain and forwarded along with the local clock. If the outgoing flit was read out from any FIFO, the local clock at that particular node is transmitted as the outgoing clock and the output port is said to be in the FIFO mode. The state machine is then in the FIFO state shown in Figure II.5. If the outgoing flit came from the ABC, the incoming clock is forwarded as the outgoing clock and the output port is said to be in the ABC mode. This ensures that the output data flits are synchronous with the output clock thus avoiding potential metastability conditions downstream. The state machine is in the ABC state. The ABC state and the FIFO state are two stable states for this FSM. The state machine can stay in any other state for only one cycle of the local clock.

The trigger signal for transition from one mode to another is generated in the outgoing clock domain. This signal is then synchronized into the desired clock domain by standard, two-cycle synchronization. Once the trigger signal has been synchronized the transition to the new clock begin and only after a successful transition of the clock is the datapath switched, thus avoiding potential metastability conditions in the control logic.

ABC Mode to FIFO Mode: Transition from ABC mode to FIFO mode for a given output port occurs in the following two cases:

- If an off signal arrives from the downstream node.
- If there is a packet in the turn-path or a local packet has been generated.

As presented in the Figure. II.5, in both the cases, first the output clock is switched to the local clock (State 1). The datapath is then switched to one of the bi-FIFOs depending on which has a packet in it (State 2). The straight path bi-FIFO is always given preference just after the transition . In State FIFO the router begins transmission of the packets present in the bi-FIFO. If the router was idle before the switch and the straight-path bi-FIFO is empty, the datapath is switched to the turn path. This ensures, that all the flits of the packet pass through the router before a new packet is transmitted. During the normal operation in FIFO mode, the bi-FIFOs are selected by Round Robin algorithm.

The router then transmits the flits read from the selected bi-FIFO unless an off signal is received from the downstream node, which indicates the presence of congestion at the downstream node. Once the off signal is cleared, the transmission is restarted. The state machine enters into the FIFO state and transmits the local clock at the output port.

Altogether the transition from ABC mode to FIFO model takes three cycle to complete. This added latency is not seen by flits traversing the straight path because they are already latched into the sync-FIFO.

FIFO Mode to ABC Mode: The transition from FIFO mode to ABC mode for a given output port occurs when the FIFO mode is currently selected and there are no flits occupying any FIFO associated with this output port.

These transitions can be divided into 4 stages corresponding to States 3 to States 9 in Figure II.5:

- In the first stage, the read signals of all the bi-FIFOs are deactivated. This makes sure that no packet is read out during the clock transition (State 3).
- If all the turn buffer FIFOs are empty, the straight path is then selected as the output datapath, (State 4).
- In the third stage if the straight path bi-FIFO is empty and the downstream node is not transmitting an off signal, the output clock is switched to the incoming clock;(State 5 to State 8). It takes at least two cycles for the arrival of a packet at the input of the bi-FIFOs to translate in the reading clock domain, so ensure that no packet was latched into the bi-FIFO while the clock transition was taking place, this stage takes four cycles to complete.

- In the final stage, the datapath is switched to ABC State 9.

However, if a flit arrives on the straight-path in any of the above mentioned stages the transition is aborted and the datapath is switched back to the straight path bi-FIFO at a cost of three additional cycles.

CHAPTER III

TOPOLOGY AND ROUTING ALGORITHM

As discussed in Section II.1, we propose a new class of topologies, double-chain topologies, to leverage the advantages of our ABC router design. In particular, double-chain topologies take advantage of the lower latency of the straight-path over the turn-path in an ABC router. A double-chain topology is comprised of two disjoint but overlapping chains, each of which connects all network nodes. As the possible space of all possible double-chain topologies is large, in this section we discuss the selection of a double-chain topology and routing algorithm suitable for the 2D planar silicon substrate of current NoCs.

III.1. Topology Characteristics

To limit the state space of the network topologies that might be easily implemented in 2D planar silicon, we began with the following restrictions:

- The topology should not require more router ports than a typical 2D Mesh would require (five ports in total). Adhering to this restriction helps ensure that the design complexity and area of the ABC router remains similar to that of a 2D Mesh router.
- Each topology must consist of only two chains, each of which must pass through every network node. Trivial analysis shows that it is not possible to connect all nodes with more than two chains without increasing the number of router ports or subdividing the network into non-overlapping chains. Subdividing the network into more than two non-overlapping chains always performs worse than connecting the non-overlapping chains together.

- Each chain may only connect near neighbor nodes together in one hop. This restriction ensures that there are no “fly-over” links, links which pass through but do not connect to a given node. Fly-over links reduce the maximal wire density without adding to a node’s I/O bandwidth.

- The overall maximum bisection bandwidth of the topology must not be significantly more than that of a standard mesh topology. Strictly speaking, connecting all network nodes together in two chains will always require at least one more bisection link than a comparable 2D Mesh topology. We will only examine topologies which increase bisection wire density minimally and in our evaluation we will ensure packet flit counts are adjusted to reflect the slightly reduced available wire density.

Note that for the purposes of this discussion we will examine chains rather than rings, to simplify deadlock avoidance.

The intent of double-chain topologies is to increase the number of source-destination pairs which are connected by straight-paths, thereby reducing the average packet latency. In order to evaluate the relative merits of individual topologies we must understand the relative latencies of the straight-path versus the turn-path.

III.2. Router Traversal Latency

As shown in Figure II.4, when not experiencing congestion, packets traversing an ABC router may take either a straight-path ABC channel or be latched into a turn-path bi-FIFO. The latency of bi-FIFO is two cycles. Thus, combined with the latency of the control logic, there is a penalty of three cycles for a packet to make a turn.

Alternately packets which traverse the router on a straight-path, utilizing the ABCs, will only face the link delays. According to International Technology Road map for Semiconductors [18], assuming 22 nm technology, the global clock frequency would be around 3 GH z while the link delay should be between 700-1300 ps/mm. Thus, we will assume the link delay, combined with the small amount of combinatorial logic delay in each router to be approximately 75% of a clock cycle.

III.3. Routing Algorithm

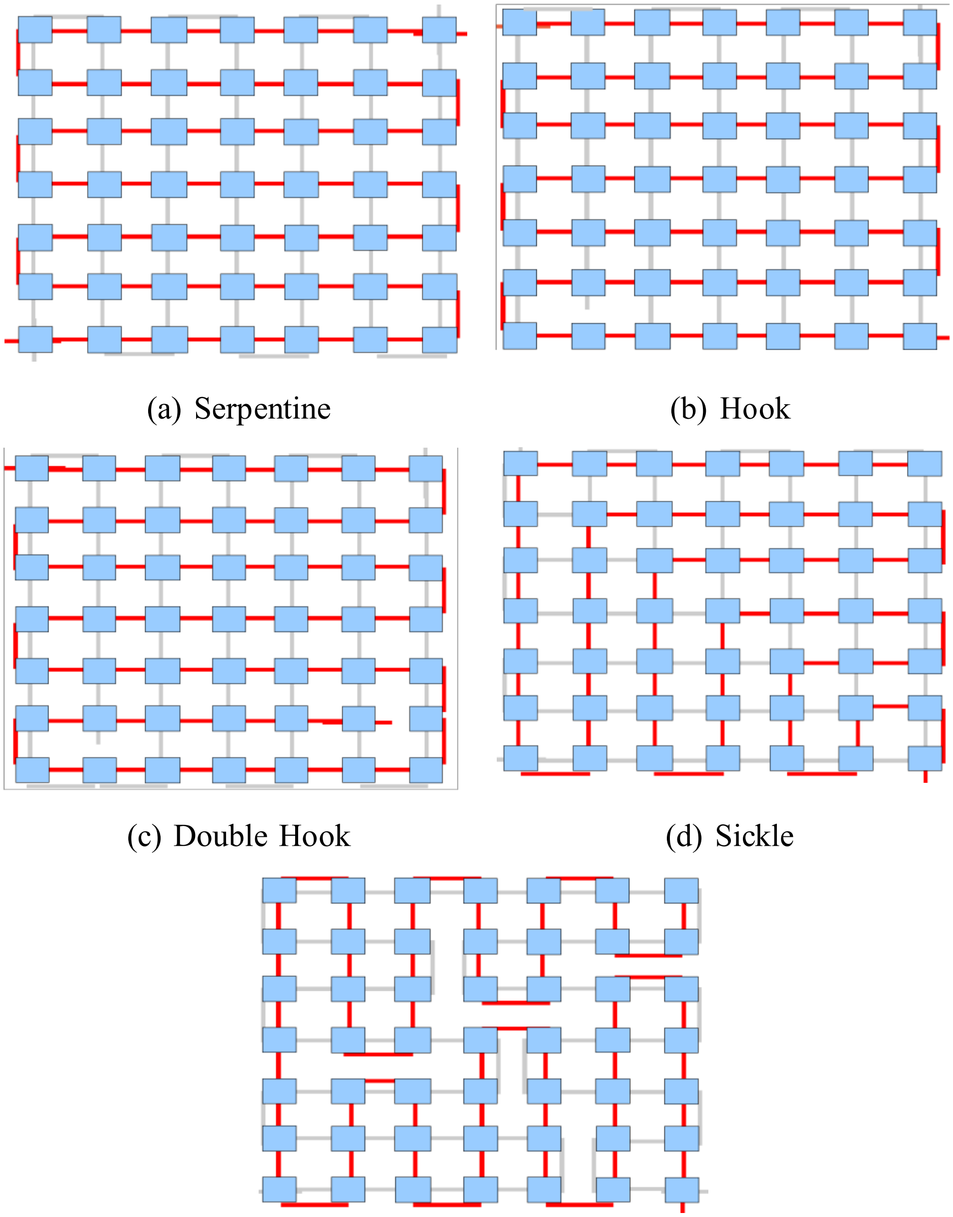
To complement the new, double-chain topologies we use with ABC routers, we developed a modified version of standard XY dimension order routing (DOR). Our routing algorithm leverages the observation that in any double chain network there are always three different routes between any two nodes: 1) following the straight-path on the blue chain, 2) following the straight-path on the red chain, or 3) following a DOR path starting on the blue chain and turning to the red chain. Our algorithm attempts to optimize packet route based upon assumptions of link and turn delays under no load. Suppose that each link between two adjacent nodes causes a delay of x clock cycles. A turn costs 3 cycles in addition to the link delay. If a flit turns it faces an additional delay of (3/x + 1) times the link delay. Thus, if the number of hops is less than or equal to (3/x + 1), traveling on ABC is a better choice than making a turn. When traveling along the straight path requires more hops than this value, turning leads to less delay in terms of clock cycles. Our routing algorithm selects from among the three legal paths between a source-destination pair based upon this estimation of latency under no-load.

Using the estimated router latencies discussed in Section III.2, yields a turn cost of (3/.75 + 1) = 5. Thus, for the given topologies, we remain on the same chain if the number of hops is less than or equal to five else we make a turn. As can be seen from the Figure. II.1, a packet generated at (0,6) will travel only through the blue chain to reach (6,4). However, a packet generated at (0,4) for (1,1) will take up the blue chain and then turn and travel through the red chain, Figure. II.2.

III.4. Topology Analysis

In this subsection we discuss our selection double-chain topology for a 7x7 network. Initially we ran a brute force search to find the optimum topology. We computed all the possible topologies that met the topology restrictions described in Section III.1 and evaluated them based upon average, zero-load latency for random traffic given the routing algorithm described in Section III.3. Using this technique we were able to find the optimum topologies for 4x4 and 5x5 network, however due to the exponential expansion of the solution set we were unable to compute the optimal topology for larger networks. Instead, we extrapolated the top five results obtained from the evaluation of 4x4 and 5x5 networks to up to a 7x7 by network and then compared the five resultant topologies to determine the best topology. As the search of both 4x4 and 5x5 networks produced the same five double-chain topologies, with the same order or precedence, we believe this method is likely to provide a network topology that is close to optimal for a 7x7 network.

Figure III.1 depicts the five topologies evaluated for an 7x7 2-D network. The evaluation results for these topologies are shown in Figure III.2. The figure shows both the average, as well as the worst case no-load, packet latency for every possible source-destination pair. As shown, the Serpentine topology performs the best with respect to both the average as well as the worst case latency. This is because it is the only topology among the five, in which each node is connected to all its physical neighbors through both the chains resulting in lower cost paths in comparison to the other topologies for all source and destination pairs. The Hook topology is also a



(a) Serpentine

(b) Hook

(c) Double Hook

(d) Sickle

(e) ’H’ Shape

Fig. III.1.: 2-D topologies for ABC router network.

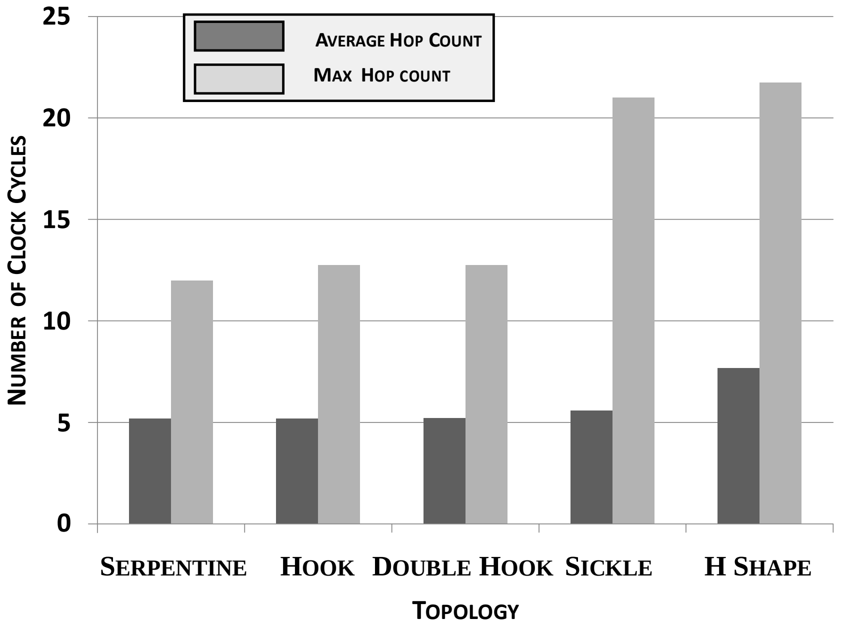


Fig. III.2.: Average and worst case hop count for the five topologies.

close contender. We found, however, that as the turn-path penalty decreases with respect to the straight path, the Serpentine topology performs better than the Hook topology. Therefore, we use the Serpentine topology for our implementation. These five topologies do not represent all possible topologies, and we plan to investigate an optimal ABC topology further in future work.

An important observation in the search of topologies is that as the network scales, the distance between source and destination pairs increases and the straight path starts losing its sheen against the turnpath. Thus with the increase in number of nodes, all the topologies perform equivalently. A similar trend would be followed with the increase in wire delays.

CHAPTER IV

EXPERIMENTS AND EVALUATION

In this section, we evaluate ABC routers experimentally to gain an understanding of their performance under different types of synthetic and realistic workloads. We compare ABC's performance against a baseline, synchronizing router design.

IV.1. Methodology

In these experiments, we evaluate a network of ABC routers connected in the 7x7 2-D Serpentine topology depicted in Figure. III.1(a) and implemented using the routing algorithm described in Section III.3. To capture all ABC router network performance results, a fully synthesizable Verilog implementation of the ABC network was designed and simulated. The baseline design consists of an 7x7, 2-D mesh network with XY DOR routing. In the baseline design packets are synchronized at every hop, incurring a delay of three clock cycles per hop. The baseline router has two, eight-flit-deep, virtual channels (VCs) per port, yielding approximately the same area overhead as the ABC router.

We evaluated three types of synthetic workloads: uniform random, bit complement, and transpose. In all cases packets were injected according to a uniform random process. Five experimental runs were completed for each workload and the mean and standard deviation of the results are shown. Packet length was varied randomly between two to five flits and the simulator was run for 1000 cycles of warm-up followed by 5000 monitored packets.

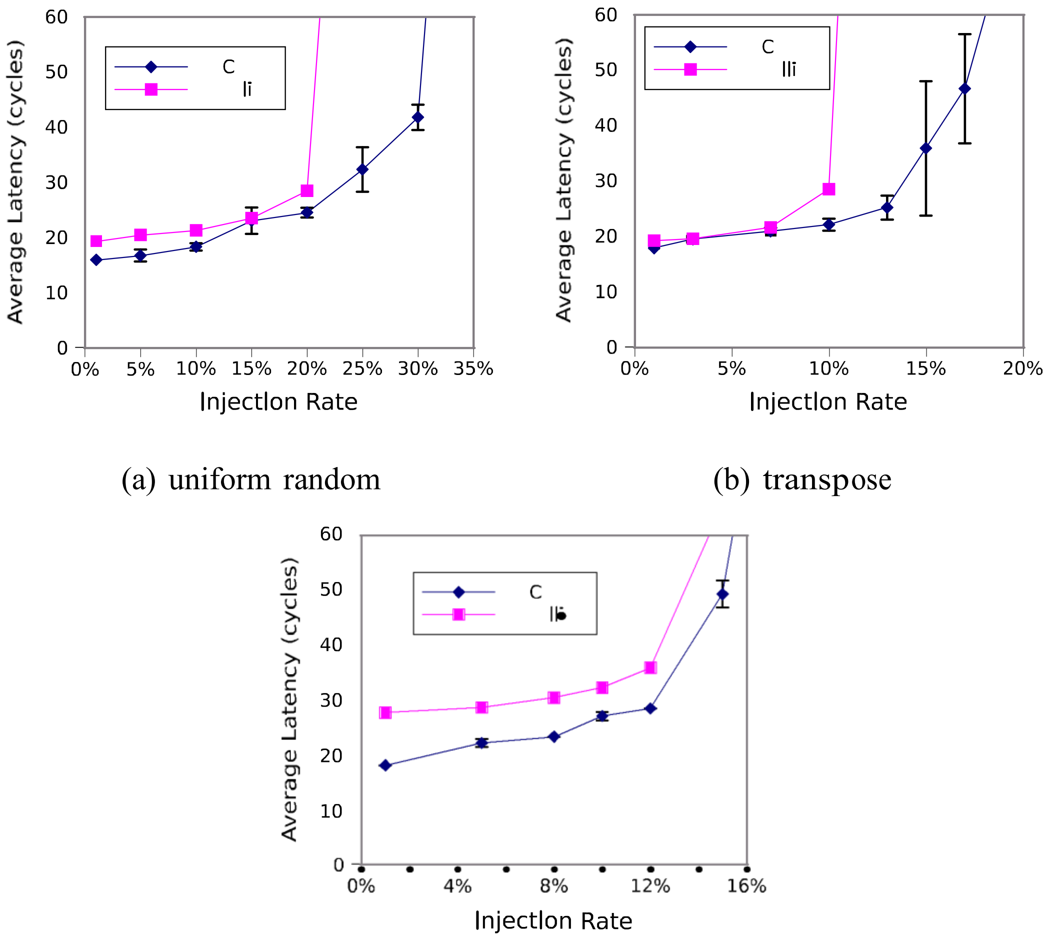
To determine the performance under a realistic workload the ABC network was also evaluated under network traces taken from the SPLASH-2 suite of benchmarks [19]. The traces were obtained from a forty-nine node, shared memory CMP system simulator, arranged in a 7x7 2-D mesh topology [20]. Due to the simulator performance limitations associated with full Verilog simulation of all 49 cores, we were forced to run only a small portion of the actual trace. To have an un-biased evaluation, we chose 500,000 cycles from the middle of each trace with a warm-up period of 10,000 cycles.

Finally, the Serpentine topology, shown in Figure. III.1(a), has one extra bisectional link than a standard 2D Mesh topology. Assuming equal bitwidths between the ABC and baseline topologies would give the ABC network 14.28% more bisection bandwidth. To approximate a uniform wire density across all bisections between the ABC and baseline designs, we decreased the number of flits in each baseline network packet by one relative the ABC design.

IV.2. Synthetic Traffic

Figure. IV.1 compares the latency performance of ABC router with the baseline for uniform random, transpose, and bit-complement workloads. In uniform random traffic, each source is equally likely to send a packet to each destination. In transpose traffic, node (x, y) sends packets to node (y, x). In bit-complement traffic, node (x, y) exchanges packets with node $(\tilde{x}, \tilde{y})$ where, $\tilde{x}$ is the one's complement of x. Among the three traffic patterns, random traffic uniformly balances load even for topologies and routing algorithms that normally have poor load balance. The other two patterns concentrate on individual source-destination pairs, thus stress the load balance of a topology and routing algorithm [21].

For uniform random traffic, shown in Figure IV.1(a), ABCs offer 20% improvement in no-load latency and saturation throughput is improved by 50% over the baseline design. The improvement in the saturation throughput occurs due to our



(a) uniform random

(b) transpose

(c) bit-complement

Fig. IV.1.: Latency performance against baseline for synthetic workloads.

topology's increased number of bisectional links compared to a standard mesh network.

Transpose traffic, in Figure IV.1(b), shows little improvement in the no-load latency. This is because, for transpose traffic, packets must always take a turn due to the arrangement of source-destination pairing. Saturation throughput, however, is greatly improved as compared to the baseline design.

For bit complement traffic, shown in Figure IV.1(c), ABCs offer an improvement

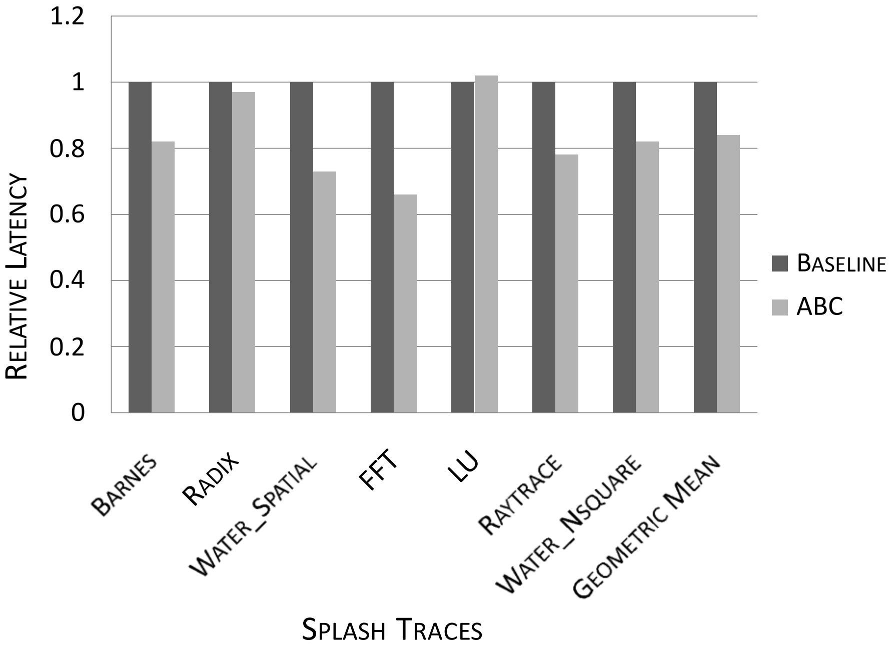


Fig. IV.2.: Relative performance against baseline for realistic workloads.

of 26% in no-load latency as well as 26% percent improvement in saturation throughput. In bit complement traffic, packets travel further than the other two synthetic patterns; therefore, ABC's benefits compound resulting in much better performance than baseline.

Generally, ABCs outperform baseline for all the workloads. An important feature common for all the three loads is the shape of the curve. In all the three graphs the shape of the curve for the ABC design is uneven, unlike the baseline. For example, for random traffic there is a distinct bump in latency at 15% traffic. This is caused due to the following reason. At around 8% traffic,congestion causes the routers to switch back and forth between FIFO and ABC modes which lead to an increase in the relative rate of increase of latency until the load becomes large enough that the routers get locked in FIFO mode, thus causing a fall in the rate and thus a bump is formed.

IV.3. Realistic Traffic

Figure. IV.2 presents the relative performance of the ABC network with respect to the baseline design for the SPLASH-2 traces. From the figure it can be observed that ABC outperforms the baseline design in all the traces except one, (LU), where baseline is marginally better than ABC. ABC shows an improvement of nearly 30-35% over the baseline design for FFT and Water-Spatial. The Geometric Mean for the relative performance is around 84.5% with respect to the baseline thus, on the average, the ABC network performed 15.5% better than the baseline design for the fraction of realistic traffic observed.

CHAPTER V

RELATED WORK

In this section we describe the connections between this work and other works which address performance improvements in asynchronous interconnection networks and new topologies to reduce hop count.

V.1. Asynchronous Interconnect

ABC network is a multi-synchronous architecture employing a mesochronous clocking scheme such that each node is clocked at the same frequency but can tolerate any amount of skew [22]. A critical aspect of mesochronous and plesiochronous interconnect design is the avoidance of metastable state due to a violation of setup or hold time of the flip-flops and registers present in the system. In the metastable state the output of the system is unpredictable. A well established approach for mesochronous NoCs was proposed by Panades and Greiner, in the form of an optimized bi-synchronous FIFO featuring low-latency and small footprint [17]. This FIFO adds a latency of two and three clock cycles to gain robustness against metastability.

Another solution, presented by Dally and Poulton, consists of delay-line synchronizers, using a variable delay on the data lines [23]. This delay avoids switching in the metastable window of the receiving registers. This solution requires careful, post-manufacturing tuning to ensure metastability is avoided. Variable delay lines also may make this solution undesirable as they are not always available in standard cell libraries.

Mangano et. al. proposed the skew insensitive link (SKIL) solution to address metastability in mesochronous interfaces [24]. SKIL supports arbitrarily skewed clock signals by relying on a two stage buffer FIFO structure by writing and reading flits into and from the buffer FIFO in a ping-pong fashion.

It should be noted that some of the above synchronization solutions could be used to augment our ABC approach, we plan to explore this in future work. In each of the above techniques, data must be synchronized at each intermediate node. Synchronization delay at every hop forms a major portion of the total latency and can be reduced by either lowering the hop-count or reducing the per-hop delay.

Another facet of GALS architecture is a completely asynchronous implementation. Asynchronous circuits exchange information using a handshake to explicitly indicate the validity and acceptance of data. This is in contrast to the multi-synchronous and synchronous design styles, which use a globally distributed clock signal to indicate moments of stability of the data. There are a number of alternative asynchronous design styles differing in how they indicate data validity. Some styles base the handshake upon a matched-delay path representing the slowest part of the data path, but these suffer from the same timing validation problems as synchronous design [25].

Examples of NoCs based on asynchronous circuit techniques are CHAIN [26], MANGO [27], ANoC [28], and QNoC [29].

The CHAIN network (CHip Area INterconnect) is implemented entirely using asynchronous, or clockless, circuit techniques. The network is based on narrow delay-insensitive highspeed links using one-of-four data encoding.

In order to address the design challenges of multi-application SoCs, CEA-LETI has proposed and developed ANOC [30], an Asynchronous Network-on-Chip architecture, where NOC nodes and links are implemented using Quasi-Delay-Insensitive asynchronous logic while the NoC functional units are implemented with independent clock domains using standard synchronous design methodologies.

While, these techniques offer high performance in terms of latency and reduced idle power, they involve asynchronous or non-standard cell modules in the router design. ABC on the other-hand can be implemented with standard cell design.

V.2. Low Hop Count Topologies

New topologies to lower the hop count have also been explored. Dally proposed express cubes which help in lowering the per-hop latency rather than the hop count [31]. An express cube is a k-ary n-cube, augmented by one or more physical links or express channels that allow nonlocal messages to bypass the intermediate nodes. Kim et al. and Grot et al. proposed topologies with higher radix routers, leading to lower hop counts but resulting in more complex router designs [32, 33].

Express virtual channels (EVCs) have also been proposed to reduce per-hop delay. EVCs virtualize the physical links in an express-cube, limiting wasted physical link bandwidth [34]. EVC-based flow control allows virtual bypassing of the packet at the intermediate nodes thus reducing the per-hop delay. This method is efficient for multi-hop packets but does not fare well for communications between neighboring routers.

In contrast to the prior work, the ABC router design targets both the per-hop latency and the hop count to reduce network latency. ABC routers, thus, perform well for both long-haul and short-haul packet communication. Also in contrast to EVCs and express cubes, ABCs are physical links present in the router offering an asynchronous bypass from the buffer FIFOs. Since ABCs are inherent to the router, they can be dynamically assigned at every hop, unlike EVCs which must be allocated upstream at specific nodes only. A packet takes an ABC whenever possible thereby avoiding FIFOs and synchronization delay, resulting in a lower per-hop latency.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

In this work we present the design, implementation, and evaluation of asynchronous bypass channel (ABC) routers. ABC routers aim to achieve lower latencies by eschewing synchronization and latching at the intermediate nodes between source and destination with a network. We present a detailed microarchitecture for an ABC router.

We also present a new class of network topologies and associated routing algorithm to complement the router design. This new class of topology, double-chain topologies, leverages the ABC router's bias towards the straight path over the turn path. Our experiments show that our ABC router, interconnected via a double-chain topology, produces significantly lower latencies compared to baseline.

This work introduces a framework for ABC router based networks. Our future work will focus on improving the performance of the network through lower latency synchronizing techniques in the turn path. We also plan to further investigate optimal topologies for ABC enhanced routers.As mentioned in Section II.3, skew reducing latches might be required if the links are physically long. Further study needs to be done in this direction.

The presented network offers the benefit of path diversity which could not be exploited by the static routing algorithm thus necessitates employing a dynamic algorithm. Also, dynamic congestion information may be used to inform the mode switch between the FIFO and ABC modes to reduced the thrashing we see in lightly congested network and improve performance.

REFERENCES

[1] P. Macken, M. Degrauwe, M. V. Paemel, and H. Oguey, “A voltage reduction technique for digital systems,” in IEEE International Solid-State Circuits Conference, Feb. 1990, pp. 238–239.

[2] W. Kim, M. Gupta, G. Wei, and D. Brooks, “System level analysis of fast, percore DVFS using on-chip switching regulators,” in International symposium on high-performance computer architecture, 2008.

[3] T. Meincke, A. Hemani, S. Kumar, P. Ellervee, J. Oberg, T. Olsson, P. Nilsson, D. Lindqvist, and H. Tenhunen, “Globally asynchronous locally synchronous architecture for large high-performance ASICs,” in The IEEE International Symposium on Circuits and Systems (ISCAS), vol. 2, 1999, pp. 512–515.

[4] M. J. S. Smith, Application-Specific Integrated Circuits. Boston, MA, 1997, Addison-Wesley Longman Publishers Inc.

[5] P. Gratz, C. Kim, R. Mcdonald, S. Keckler, and D. Burger, “Implementation and Evaluation of On-Chip Network Architectures,” in International Conference on Computer Design, 2006, pp. 477–484.

[6] P. Pande, C. Grecu, M. Jones et al., “Performance evaluation and design trade-offs for network-on-chip interconnect architectures,” IEEE Trans. Comput., vol. 54, no. 8, pp. 1025–1040, Sep., 2005.

[7] P. Kapur, G. Chandra, and K. Saraswat, “Power estimation in global interconnects and its reduction using a novel repeater optimization methodology,” in Proceedings- Design Automation Conference. pp. 461-466. 2002, 2002.

[8] D. Sylvester and K. Keutzer, “Impact of small process geometries on microarchitectures in systems on a chip,” Proceedings of the IEEE, vol. 89, no. 4, pp. 467–489, 2001.

[9] A. Shacham, K. Bergman, and L. Carloni, “Maximizing GFLOPS-per-Watt: High-bandwidth, low power photonic on-chip networks,” in P= ac2 Conference. Citeseer, 2006, pp. 12–21.

[10] Z. Yu, M. Meeuwsen, R. Apperson, O. Sattari, M. Lai, J. Webb, E. Work, T. Mohsenin, and B. Baas, “Architecture and evaluation of an asynchronous array of simple processors,” Journal of Signal Processing Systems, vol. 53, no. 3, pp. 243–259, 2008.

[11] T. Kuroda, K. Suzuki, S. Mita, T. Fujita, F. Yamane, F. Sano, A. Chiba, Y. Watanabe, K. Matsuda, T. Maeda et al., “Variable supply-voltage scheme for low-power high-speed CMOS digital design,” IEEE Journal of Solid-State Circuits, vol. 33, no. 3, 1998.

[12] T. Lin, C. Liu, S. Tseng, Y. Chu, and A. Wu, “Overview of ITRI PAC project-from VLIW DSP processor to multicore computing platform,” in Proc. IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT 2008), 2008, pp. 188–191.

[13] D. Marculescu and A. Iyer, “Power efficiency of voltage scaling in multiple clock, multiple voltage cores,” in iccad. IEEE/ACM, 2002, pp. 379–386.

[14] U. Ogras, R. Marculescu, P. Choudhary, and D. Marculescu, “Voltage-frequency island partitioning for GALS-based networks-on-chip,” in Proceedings of the 44th annual Design Automation Conference. ACM, 2007, p. 115.

[15] I. Panades, A. Greiner, and A. Sheibanyrad, “A low cost network-on-chip with guaranteed service well suited to the gals approach,” Proc. NANONET, 2006.

[16] U. Ogras, R. Marculescu, D. Marculescu, and E. Jung, “Design and management of voltage-frequency island partitioned networks-on-chip,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Special Section on Networks-on-Chip, 2009.

[17] I. Panades and A. Greiner, “Bi-Synchronous FIFO for Synchronous Circuit Communication Well Suited for Network-on-Chip in GALS Architectures,” in The First International Symposium on Networks-on-Chip (NOCS), 2007, pp. 83–94.

[18] A. Allan, D. Edenfeld, W. Joyner, A. Kahng, M. Rodgers, and Y. Zorian, “International Technology Roadmap for Semiconductors,” IEEE Comput, 2002.

[19] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, “The SPLASH-2 programs: characterization and methodological considerations,” SIGARCH Compututer Architecture News, vol. 23, no. 2, pp. 24–36, 1995.

[20] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha, “Express Virtual Channels: Towards the Ideal Interconnection Fabric,” in The 13th International Symposium on Computer Architecture (ISCA), 2007, pp. 150–161.

[21] W. Dally and B. Towles, Principles and practices of interconnection networks. Morgan Kaufmann, 2004.

[22] A. Sheibanyrad, A. Greiner, I. Miro-Panades, K. SoC, and L. de Paris, “Multi-synchronous and Fully Asynchronous NoCs for GALS Architectures (HTML),” IEEE Design and Test of Computers, vol. 25, no. 6.

[23] W. Dally and J. Poulton, Digital systems engineering. Cambridge Univ Pr, 1998.

[24] D. Mangano, R. Locatelli, A. Scandurra, C. Pistritto, M. Coppola, L. Fanucci, F. Vitullo, and D. Zandri, "Skew insensitive physical links for network on chip," in The 1st International Conference on Nano-Networks and Workshops (NanoNet), 2006, pp. 1–5.

[25] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," ACM Comput. Surv., vol. 38, no. 1, p. 1, 2006.

[26] J. Bainbridge and S. Furber, "CHAIN: A delay-insensitive chip area interconnect," IEEE MICRO, pp. 16–23, 2002.

[27] T. Bjerregaard and J. Sparso, "A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip," in Proceedings of the conference on Design, Automation and Test in Europe-Volume 2. IEEE Computer Society Washington, DC, USA, 2005, pp. 1226–1231.

[28] P. Vivet, D. Lattard, F. Clermidy, E. Beigne, C. Bernard, Y. Durand, J. Durupt, and D. Varreau, "FAUST, an Asynchronous Network-on-Chip based Architecture for Telecom Applications," Proc. 2007 Design, Automation and Test in Europe (DATE07), 2007.

[29] D. Rostislav, V. Vishnyakov, E. Friedman, and R. Ginosar, "An asynchronous router for multiple service levels networks on chip," in Proceedings of the 11th IEEE International Symposium on Asynchronous Circuits and Systems. IEEE Computer Society, 2005, pp. 44–53.

[30] Y. Thonnart, E. Beigné, and P. Vivet, "Design and Implementation of a GALS

Adapter for ANoC Based Architectures," in Proceedings of the 2009 15th IEEE Symposium on Asynchronous Circuits and Systems (async 2009)-Volume 00. IEEE Computer Society Washington, DC, USA, 2009, pp. 13–22.

[31] W. Dally, "Express cubes: Improving the performance of k-ary n-cube interconnection networks," IEEE Transactions on Computers, vol. 40, no. 9, pp. 1016–1023, 1991.

[32] J. Kim, W. Dally, B. Towles, and A. Gupta, "Microarchitecture of a high-radix router," ACM SIGARCH Computer Architecture News, vol. 33, no. 2, pp. 420–431, 2005.

[33] B. Grot, J. Hestness, S. Keckler, and O. Mutlu, "Express Cube Topologies for on-Chip Interconnects," in The 15th International Symposium on High Performance Computer Architecture (HPCA), Feb. 2009, pp. 163–174.

[34] A. Kumar, L. Peh, P. Kundu, and N. Jha, "Express virtual channels: towards the ideal interconnection fabric," in The 34th annual International Symposium on Computer Architecture (ISCA), 2007, pp. 150–161.

VITA

Name: Tushar Naveen Kumar Jain

Address:
c/o Department of Electrical and Computer Engineering,
WERC, Texas A&M University, College Station, TX 77843-3128, USA

Email Address: tnjiitr@gmail.com

Education:

B.Tech, Indian Institute of Technology, Roorkee (IIT R) 06/2007

M.S., Computer Engineering, Texas A&M University, 08/2010

Work Experience:

Engineer, R&D, Bajaj Auto Ltd., 06/2007- 07/2008

LinkedIn Profile: http://www.linkedin.com/pub/tushar-jain/13/816/663

The typist for this thesis was Tushar N K Jain.