Petabit Switch Fabric Design

Bhavana Chaurasia
Yale Chen
Ian Juch
Surabhi Kumar
Jay Mistry

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2015-116
http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-116.html

May 15, 2015

Acknowledgement

Special thanks to our advisors Elad Alon and Vladimir Stojanovic. Also many thanks to the graduate students at BWRC who helped us tremendously with the tools setup for our project: Brian Zimmer, Steven Bailey, Nathan Narevsky, and Krishna Settaluri.

University of California, Berkeley College of Engineering

MASTER OF ENGINEERING - SPRING 2015

Electrical Engineering and Computer Science

Integrated Circuits

Petabit Switch Fabric Design

Bhavana Chaurasia

This **Masters Project Paper** fulfills the Master of Engineering degree requirement.

Approved by:

1. Capstone Project Advisor #1:

Signature: _____ Date _____

Print Name/Department: Elad Alon/EECS

2. Capstone Project Advisor #2:

Signature: _____ Date _____

Print Name/Department: Vladimir Stojanovic/EECS

Acknowledgements

Special thanks to our advisors Elad Alon and Vladimir Stojanovic. Also many thanks to the graduate students at BWRC who helped us tremendously with the tools setup for our project: Brian Zimmer, Steven Bailey, Nathan Narevsky, and Krishna Settaluri.

Table of Contents

Shared Team Paper

Individual Paper

1. Problem statement

The current trend in the computing industry is to offer more performance by leveraging more processing cores. Because we have run into some physical limits on how fast we can make a single processor run, the industry is now finding ways to utilize more cores running in parallel to increase computing speeds. Looking beyond the four and eight core systems we see in commercially available computers today, the natural progression is to scale this up to hundreds or thousands of processing units (Clark, 2011). All of those processing units working together cohesively at this scale requires a great deal of communication. Furthermore, these processors need to talk not only to each other, but also to any number of other resources like external memories or graphics processors. Being able to move bits around the chip efficiently and quickly therefore becomes one of the limiting factors in the performance of such a system.

To enable this communication, most of today's multi-core systems use interconnection networks. While there are many different ways to design these networks, network latency, the time it takes to communicate between network endpoints, becomes directly dependent on the number of router hops (Daly, 2004). The number of router hops depends upon the total number of endpoint devices as well as the number of ports available on each router—the router's radix. With higher radix routers, we can connect more endpoint devices with fewer total hops. Our project is thus to explore the design space for a high radix router, which will reduce the latency of the interconnect networks and thus enable more efficient communication. Given an initial design based on the work of Stanford graduate student Daniel Becker, we will be exploring how changing different parameters affects the performance of the overall router design in terms of chip area, power consumed, data transmission rates, and transmission delays. We hope to use this data to draw conclusions about the optimal configurations for a high-radix router, and to justify

our conclusions with data. The researchers at Berkeley Wireless Research Center (BWRC) will consider the results of our analysis as they try to construct future high performance systems.

2. Industry and market trends

I. INTRODUCTION

With current trends in cloud computing, big data analytics, and the Internet of Things, the need for distributed computation is growing rapidly. One promising solution that modern computers employ is the use of large routers or switches to move data between multiple cores and memories. The goal of our Petabit Switch Fabric capstone project is to explore the design tradeoffs of such network switch architectures in order to scale this mode of communication to much larger magnitudes. We aim to examine the viability of using these designs for a petabit interconnect between large clusters of separate microprocessors and memories. High bandwidth switches will allow distributed multicore computing to scale in the future. Given a prototype, we will be studying power, area, and bandwidth tradeoffs. By analyzing the performances of these parameters, we will eventually map a Pareto optimal curve of the design space. The results of the project will provide valuable data for future research related to developing network switch designs. As we consider how to commercialize this project, it becomes useful to understand the market that we will be entering. In this paper, we will use Porter's Five Forces as a framework to determine our market strategy (Porter, 1979).

II. TRENDS

First, we will explore some of the trends in the semiconductor and computing industries that motivate our project. One of the most important trends in technology is the shift toward cloud computing in both the consumer and enterprise markets. On the enterprise side, we are observing an increasing number of companies opting to rent computing and storage resources from companies such as Amazon AWS or Google Compute Engine, instead of purchasing and managing their own servers (Economist, 2009). The benefits of this are multi-fold. Customers gain increased flexibility because they can easily scale the amount of computing resources they require based on varying workloads. These companies also benefit from decreased costs because they can leverage Amazon's or Google's expertise in maintaining a high degree of reliability. We are seeing that these benefits make outsourcing computing needs not only standard practice for startups, but also an attractive option for large, established companies because the benefits often outweigh the switching costs.

As warehouse scale computing consolidates into a few major players, the economic incentive for these companies to build their own specialized servers increases. Rather than purchasing from traditional server manufactures such as IBM or Hewlett-Packard, companies like Google or Facebook are now operating at a scale where it is advantageous for them to design their own servers (Economist, 2013). Custom built hardware and servers allow them to optimize systems for their particular workloads. In conjunction with the outsourcing and consolidation of computing resources, these internet giants could potentially become the primary producers of server hardware, and thus become one of our most important target customers as we bring our switch to market.

On the consumer side, we have seen a rapid rise in internet data traffic in recent years. Smartphones and increasing data speeds allow people to consume more data than ever. Based on market research in the UK, fifty percent of mobile device users access cloud services on a weekly basis (Hulkower, 2012). The number of mobile internet connections is also growing at an annual rate of 36.8% (Kahn, 2014:7). Data usage is growing exponentially as an increasing number of users consumes increasing amounts of data. Moreover, the Internet of Things (IoT) is expected to produce massive new amounts of traffic as data is collected from sensors embedded in everyday objects. This growth in both data production and consumption will drive a strong demand for more robust networking infrastructure to deliver this data quickly and reliably. This will present a rapidly growing market opportunity in the next decade (Hoover's, 2015). Overall, the general trends in the market suggest a great opportunity for commercializing our product.

As the IoT, mobile internet, and cloud computing trends progress, they will all drive greater demand for more efficient data centers and the networking infrastructure to support further growth. Concurrently, the pace of advances in semiconductor fabrication technology has historically driven rapid performance and cost improvements every year. However, these gains have already slowed down significantly in recent years, and are expected to further stagnate over the next decade. We are rapidly approaching the physical limits of current semiconductor technology. As a result, we observe a large shift from single core computing to parallel systems with many distributed processing units. With no new semiconductor technology on the immediate horizon, these trends should continue for the foreseeable future.

III. INDUSTRY AND COMPETITIVE LANDSCAPE

Next, we will examine our industry and competitive landscape. The semiconductor industry is comprised of companies that manufacture integrated circuits for electronic devices such as computers and mobile phones. This is a very large industry, consisting of technology giants such as Intel and Samsung, with an annual revenue of eighty billion dollars in the United States alone (Ulama, 2014:19). Globally, the industry revenue growth was a relatively modest 4.8% in 2013 (Forbes, 2014). However, as cloud computing becomes more prevalent, we expect that the need for better hardware for data centers will continue to rise, and the growth of this sector will likely outpace the overall growth of the semiconductor industry.

Although the sector is growing rapidly and the demand for networking infrastructure is high, competition is fierce in both telecommunications and warehouse scale computing. There are many well established networking device companies such as Juniper Networks, Cisco, and Hewlett-Packard. Large semiconductor companies such as Broadcom and Mellanox, along with smaller startups such as Arteris and Sonics, are also designing integrated switches and network on chips (NoC).

Specifically, one of our most direct competitors is Broadcom. In September of 2014, Broadcom announced the StrataXGS Tomahawk™ Series (Broadcom, 2014). This product line is targeted towards Ethernet switches for cloud-scale networks. It promises to deliver 3.2 terabit-per-second bandwidths. This new chip will allow data centers to vastly improve data transfer rates while maintaining the same chip footprint (Broadcom, 2014). It is designed to be a direct replacement for current top-of-rack as well as end-of-row network switches. This means that the switching costs are extremely low, and it will be very easy for customers to upgrade their existing hardware. Another key feature that Broadcom is offering is packaged software that will

give operators the ability to control their networks for varying workloads (Broadcom, 2014). The Software Defined Network (SDN) is proprietary software customized for the Tomahawk family of devices. This software might be a key feature that differentiates Broadcom's product from other competitors.

We distinguish ourselves from these companies by targeting a very focused niche market. For example, Sonics has found its niche in developing a network on chip targeted towards the mobile market. Their product specializes in connecting different components such as cameras, touch screens, and other sensors to the processor. We find our niche in fulfilling a need for a high speed high radix switch in the warehouse scale computing market. Data centers of the future will be more power hungry and will operate at much faster rates (Hulkower, 2012). Therefore, our product aims to build more robust systems by minimizing power consumption while maximizing performance.

The semiconductor industry already competes heavily on the basis of price, and as performance gains level off, we expect this competition to increase (Ulama, 2015, p. 27). As a new entrant, we want to avoid competing on price with a distinguished product. As previously mentioned, our switch product is meant to enable efficient communication between collections of processors in data centers. However, it also has potential applications in networking infrastructure. Given the strong price competition within the industry, we would want to focus on one or the other in order to bring a differentiated product to market.

Another force to consider is the threat of substitutes, and we will now examine two distinct potential substitutes: Apache Hadoop and quantum computing. Apache Hadoop is an open source software framework developed by the Apache Software Foundation. This framework is a tool used to process big data. Hadoop works by breaking a larger problem down

into smaller blocks and distributing the computation amongst a large number of nodes. This allows very large computations to be completed more quickly by splitting the work amongst many processors. The product's success is evidenced by its widespread adoption in the current market. Almost every major company that deals with big data, including Google, Amazon, and Facebook, uses the Hadoop framework.

Hadoop, however, comes with a number of problems. Hadoop is a software solution that shifts the complexity of doing parallel computations from hardware to software. In order to use this framework, users must develop custom code and write their programs in such a way that Hadoop understands how to interpret them. A high throughput and low latency switch will eliminate this extra overhead because it is purely a hardware solution. The complexity of having multiple processors and distributed computing will be hidden and abstracted away from the end user. Hadoop is a software solution, so you still need physical switch hardware to use Hadoop, but future improvements to Hadoop or similar frameworks could potentially mitigate the need for the type of high-radix switch which we are building.

The other substitute we will look at is quantum computing. Quantum computing is a potential competing technology because it provides a different solution for obtaining better computing performance. In theory, quantum computers are fundamentally different in the way that they compute and store information, so they will not need to rely as heavily on communication compared to conventional processors. However, it is unclear whether practical implementations of quantum computers will ever be able to reach this ideal. Currently, only one company - D-Wave - has shown promising results in multiple trials, but, their claims are disputed by many scientists (Deangelis, 2014). Additionally, we expect our solution to be much more compatible with existing software and programming paradigms compared to quantum

computers, which are hypothesized to be very good for running only certain classes of applications. Therefore, switching costs are expected to be much higher with quantum computers. Because quantum computing is such a potentially disruptive technology, it is important to consider and be aware of advancements in this field.

IV. MARKET

Next, we will examine two different methods of commercializing our product: selling our design as intellectual property (IP), or selling a standalone chip. Many hardware designs are written in a hardware description language such as Verilog. This code describes circuits as logical functions. Using VLSI (Very Large Scale Integration) and EDA (Electronic Design Automation) tools, a Verilog design can be converted into standard cells and manufactured into a silicon chip by foundries. If we were to license our IP, a customer would be able to purchase our switch and integrate it into the Verilog code of their own design.

Some key customers for licensing our IP are microprocessor producers. The big players in this space are Intel, AMD, NVIDIA, and ARM. Intel owns the largest share of microprocessor manufacturing, and it possesses a total market share of 18% in semiconductor manufacturing (Ulama, 2014:30). Microprocessors represent 76% of Intel's total revenue, making it the largest potential customer in the microprocessor space (Ulama, 2014:30). AMD owns 1.4% of the total market share, making it a weaker buyer (Ulama, 2014:31). While Intel represents a very strong force as a buyer because of its power and size, they are still an attractive customer. If our IP is integrated into their design, we will have a significant share in the market.

Another potential market is EDA companies themselves. We can license our product to EDA companies who can include our IP as a part of their libraries. This can potentially create a

very strong distribution channel because all chip producers use these EDA tools to design and manufacture their products. Currently, EDA is a $2.1 billion industry, with Synopsys (34.7%) and Cadence (18.3%) representing 53% of the total market share (Boyland, 2014:20). Having our switch in one of these EDA libraries would result in immediate recognition of our product by a large percentage of the market.

Another option for going to market would be selling a standalone product. This means that we will design a chip, send our design to foundries to manufacture it, and finally sell it to companies who will then integrate the chip into their products. This contrasts with licensing our design to other semiconductor companies. Licensing our design would allow our customers to directly embed our IP into their own chips. One downside of manufacturing our own chip is the high cost. Barriers to entry in this industry are high and increasing, due to the high cost of production facilities and low negotiation powers of smaller companies (Ulama, 2014:28). Selling a standalone chip versus licensing an IP also targets two very different customers—companies who buy parts and integrate them, or companies who manufacturer and sell integrated circuits.

The main application of our product is in warehouse scale computing. The growth in cloud computing and media delivered over the internet means that demand for servers will see considerable growth (Ulama, 2014:8). High-speed high-radix switches will be essential in the future for distributed computing to scale (Binkert, 2012:100). In a data center, thousands of servers work together to perform computations and move data. Our product can be integrated in network routers connecting these servers together. Companies such as Cisco and Juniper, who supply networking routers, are our potential buyers. They purchase chips and use them to build systems that are sold to data centers. Our product can also be integrated directly inside the servers themselves. Major companies producing these servers include Oracle, Dell, and Hewlett-

Packard. These companies design and sell custom servers to meet the needs of data centers. As the number of processing units and memories increase in each of these servers, a high-radix switch is needed to allow efficient communication between all of these subsystems.

In order to enter the market strategically, we need to consider our positioning. The market share of the four largest players in the networking equipment industry—our target customers—has fallen by 5.2% over the past five years (Kahn, 2014:20). The competition is steadily increasing, and the barriers to entry are currently high but decreasing (Kahn, 2014:22). With the influx of specialist companies offering integrated circuits, new companies can take advantage of this breakdown in vertical integration (Kahn, 2014:22). This means that the industry may expect to see a rise in new competitors in the near future. With the increase in competition among the buyers, their power is expected to decrease. Thus, if we have a desirable technology, we may be in a strong position to make sales. Competition in server manufacturing is also high and increasing with low barriers of entry (Ulama, 2014:22). This competitive field in both networking equipment and data center servers is advantageous for us because these companies are all looking for any competitive edge to outperform each other. A technology that will give one of these companies an advantage would be very valuable.

In order to create a chip, we will need to pay a foundry to manufacture our product. Unfortunately, although there is healthy competition among the top companies in the semiconductor manufacturing industry, prices have remained relatively stable because of high manufacturing costs and low margins (Ulama, 2014:24). Because custom and unique tools are required for producing every chip, there are very high fixed costs associated with manufacturing a design. Unless we need to produce very large volumes of our product, the power of the foundries, our suppliers, is very strong. The barriers of entry for this industry are extremely high,

and we don't expect to see much new competition soon. EDA tools developed by companies such as Synopsys and Cadence are also required to create and develop our product. As discussed in previous sections, these two companies represent more than half of the market share. As a result, small startups have weak negotiation power. Both our suppliers, foundries who manufacture chips and EDA companies that provide tools to design chips, possess very strong power largely in the form of fixed costs.

V. CONCLUSION

In this paper, we have thoroughly examined a set of relevant trends in the market and, using Porter's Five Forces as a framework, conducted an analysis of the semiconductor industry and our target market. We have concluded that our project will provide a solution for a very important problem, and is well positioned to capitalize on projected industry trends in the near future. We have proposed and analyzed two different market approaches - IP licensing and selling discrete chips - and weighed the pros and cons of each. We have surveyed the competitive landscape by looking at industry behaviors and researching a few key competitors, as well as thinking about potential substitutes. With all of this in mind, we can carefully tailor our market approach in a way that leverages our understanding of the bigger picture surrounding our technology.

3. IP Strategy

Distributed computing is rapidly growing due to demand for high performance computation. Today, computers have multiple cores to divide and solve complex computational problems. In the near future, they will have many more cores which will need to work in unison. In this project, we are designing a high-radix router which will serve as an interconnect between processor cores and memory arrays in data centers. Our project addresses the problem of transferring large amounts of data between processors and memories to achieve high speed computation. It is a part of ongoing research in Berkeley Wireless Research Center (BWRC) for building hardware for next generation data centers.

The router we are designing is unique among other routers available today in several ways. First, it is a high-radix router which means it can be used to direct traffic to and from a large number of endpoints. Second, the router can support very high bandwidth. We have designed such a high-performing router by proposing a novel system architecture based on a few key design decisions from the results of our design space exploration. These design decisions differentiate our router from existing designs in the commercial and research domains, and would form the core of our patent application.

If we are successful in implementing our proposed design changes, then the router design can qualify for a patent. We would apply for a utility patent since the router will produce a useful tangible result like increased bandwidth. One of our marketing strategies is to sell the router as a standalone chip, which means we will be mass producing the router from a chip foundry. This makes it an article of manufacture, another quality of a utility patent. In addition to qualifying for one of the patent categories, our router can be considered novel invention since

it is a high radix router with up to 256 ports. This is much higher than any others that we have come across during our literature review.

Patenting our novel design will give us a huge competitive advantage because we would be the first to develop a petabit bandwidth router. In general, the semiconductor industry is highly litigiousness because of rapid change in the technology each year. Many lawsuits are filed every year between rivals like Broadcom, Qualcomm, and Samsung. Furthermore, many of these companies have very deep pockets, along the motivation and resources to rigorously protect their patent portfolio. Therefore, before commercializing our technology, we must to exercise careful scrutiny to ensure we do not infringe on anyone else's patents. In this environment, it also becomes necessary for us to hold our own patents, both to keep others from copying our technology and to prevent them from coming after us with lawsuits. However, as a small startup, we would have to weigh any sort of legal action very carefully, as we would likely not have sufficient funding to carry out protracted legal battles.

The primary risk of choosing not to patent our novel router architecture would be forfeiting the legal protections that a patent grants. As a small company starting out, we would not provide much value as to our customers beyond our technological advantage. Without a patent, we risk allowing a much larger company to copy our technology. Combined with their vast resources, this could effectively put us out of business. While we might not actually be able to defend our patent, having one would at least deter others from blatantly copying us.

Something else to consider here would be how easy we think it would be for our technology to be reverse engineered. Since our project is conducted in a research setting under BWRC, any major breakthroughs would most likely be published and peer reviewed, rather than kept

as a trade secret. Furthermore, since our technology would be based on a novel architecture rather than an implementation detail, others would almost certainly be able to engineer their own solutions based on our architecture, depending on how much we decide to publish. Thus, without a patent, we would have no way of controlling or profiting from our technology.

A potential secondary risk of not patenting might be that we would be passing on the chance to attract potential investors. In addition to the legal protection described above, holding a patent could have the additional effect of demonstrating strength to investors in multiple ways. First, the patent would differentiate us from our competitors; it gives us a sustainable, legally enforceable competitive advantage. Second, the patent would signal a high level of expertise to investors; it can signal that we are truly experts in our particular domain. Finally, the patent could provide assurances to investors that other companies will not be able to patent something similar and attempt to come after us for infringement. With all of this in mind, we would most definitely want to obtain a patent for our novel technology. Practically, the extent of legal protection we might receive remains questionable given our limited financial resources, but a patent still grants us many other advantages which could provide a huge boost to a company in its early stages. From this preliminary analysis, the benefits far outweigh to costs, and we would thus want to pursue a patent as soon as possible. We will conduct a thorough patent search with assistance from a patent attorney to make sure our invention has not previously been patented and does not infringe on any existing patents.

4. Technical contribution

I. PROJECT OVERVIEW

In light of continuous increase of microprocessor cores, there is a growing need for efficient on-chip interconnect. But as the number of cores are increasing interconnect complexity is increasing. It has been proposed that High Radix (port) routers which will lead to reduction of hop counts in interconnect networks may help in reducing latency and power. (Kim at el 2005:1) In this project we aim to explore design space for high radix routers. This will help to deduce the best architecture for network router. Our objective is to figure out the complexities related to the design of high-radix routers. We have implemented different design algorithms for the designing of the on-chip network routers and checked the design on three performance metrics: area, power and timing.

At the beginning of the project we were provided with a Stanford graduate's thesis and parameterized Verilog code, which became the starting point for our project. We build the infrastructure of our project over the provided Verilog code.

Designing a router follows standard Digital Integrated Circuit design flow. For optimum circuit design, tools required need to be configured with appropriate settings. As I have discussed the design space exploration and tools setup requirement, it generated tasks which were divided among the team members. The tasks were divided among team members according to their interest and competency. Although in course of time no one has worked only on specific tasks but here I am discussing the major tasks done by each member of our team. Ian Juch did the initial tool setup for running the whole design flow. (Juch, 2015) This enabled us to explore the design space for router design. Since we scaled the design to higher ports, tools initial setting were not appropriate and run time was very large. Hence Yale Chen and Jay Mistry helped him

to optimize the run time. Yale started looking for hierarchical coding, which is another way of implementing design and this helped in reducing run-time. (Chen, 2015) Jay and Ian investigated the design manuals for extracting different parameters which can increase the efficiency of the tool. (Mistry, 2015) Due to tools run time issue, we narrowed our design exploration to 64 ports design.

In initial 64 port design, more than 40% of the chip area was occupied by input buffers, and as SRAMs are known for its density, we proposed to implement SRAM as input buffers. SRAMs have synchronous read while current design uses flip-flops based register which have asynchronous read, this increases the complexity of implementation. This task was accomplished by Surabhi and Ian. (Kumar, 2015)

While other team members were working on tool setup and implementation of SRAM, I worked towards understanding the design and Verilog code and forming the link between the two. In the literature review every team member studied about one of the router design block and later we discussed each-others learning to integrate all the blocks and check how it is implemented in the design. The code we have been provided was parameterized code, hence while understanding the code, I focused on parameters implications and how changing each of them will change the router design. Later I started working on 64 port network router design. I analyzed the design and studied the critical path to identify the bottlenecks in the design. I explored the design space by implementing different algorithms for router components.

II. KNOWLEDGE DOMAINS

ROUTER DESIGN BLOCKS: GENERAL OVERVIEW

Routers primary function is to route the incoming packets from one of the input ports to one of the output ports. Router contains following main computational blocks which governs the path for the flow of flits (packets are divided into flits). Figure 1 represents the block diagram of a router. Following explanation has been extracted from Daniel Becker's thesis. (Becker 2012:20-23)

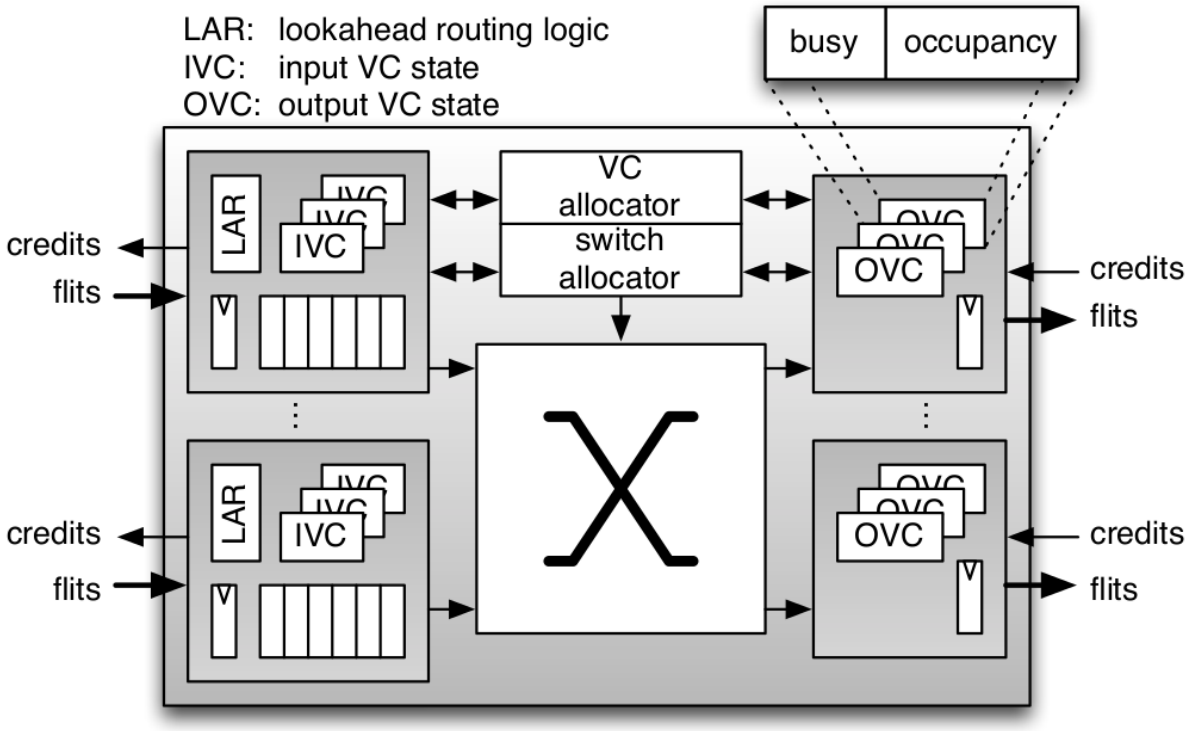


Figure 1. Router Microarchitecture (Becker, 2012)

Input Buffers: Once the packet enters the router, it needs to be stored in buffers. All the router computation occur after the packets are stored in buffer. These input packets are divided into set of flits. Hence the width of buffer is according to the flit size. There are dedicated input buffers

for every port in the router, hence with increase in number of ports, size of buffer increases linearly. In high radix routers buffer size becomes critical as it occupies around 40% of chip.

Route computation: The first flit contains routing information for whole packet. On the basis of flit's routing information and networks routing algorithm, the routing logic selects a suitable output port and a set of candidate output VCs. Lookahead routing is the commonly used algorithm for NoC routers. Complexity of route computation increases linearly with increase in number of ports.

VC allocation: Before the packet traverse from input port to output port, it needs to be ensured that a virtual channel (VC) is available at the destination output port. VC allocation allocates output VC to input packets and make sure that access to output VC is exclusive. As all the routing information is present on head flit, it will take part in VC allocation and once the VC is allocated, all the flits of a packet follow the head flit. With the increase in number of ports, allocator complexity increases quadratically which is critical in terms of timing.

Switch allocation: Once output port and VC for input ports is decided, flits take part in switch allocation. Switch allocation is responsible for providing a crossbar link between input and output port. It ensures that at a given time no two input ports receives grant for the same output port. It also provides time-slots to the flits at input buffer for having crossbar connections. Similar to VC allocator, complexity and delay of switch allocator increases quadratically with number of ports.

Switch traversal: If a request is granted by Switch allocator, it will generate a grant signal which is used as a select signal for crossbar switches. Hence after receiving a grant, in next cycle flits are transferred from input port to the output port. In crossbar switches, each input port can communicate to any output port, thus routing in the crossbar switches is a major concern.

According to hierarchical simulation results, most of the tool run-time is due to crossbar switches. (Chen, 2015) Also the area and routing on crossbar increases quadratically with number of ports.

SWITCH ALLOCATORS

Switch allocator creates matching between input active VCs request and crossbar connection to respective output port. (Becker, 2012:82-83) The quality of generated matching decides the latency and throughput of the router. Crossbar connection is made available only if receiving router connected to output port have sufficient space available for new packet. Hence it checks for the input-output port connection and tracks it for the course of time so that it can restrict the connection to output port when it is about to be full. Switch allocator generates a grant signal which is used to setup registers which controls crossbar connection.

There are different Switch allocation techniques available where throughput and matching varies with the complexity of design.

Separable input-first implementation (Becker and Dally, 2009:1-12): In this scheme, all the active VCs at every input port compete to win the grant. This is done by using V-input arbitration where V is the number of active VCs. This is called input level arbitrations. Later P-input arbitration occurs at every output port which decides the input-output port connection. The grant generated after two level of arbitration is stored in register which is used to generate select signal for Crossbar link.

Separable output-first implementation (Becker and Dally, 2009:1-12): In this scheme, all the requests from input go to output port for P-input arbitration. Since input port can win arbitration

at two output ports, all the winning VCs again participate in arbitration to select one VC for each input port. In this case grant generated from output port arbiter can't be directly used for crossbar connection because of above discussed case. Hence grant generation time is same as Separable input-first allocation scheme.

Wavefront-based switch allocator (Becker and Dally, 2009:1-12): All the request from active VC participate in arbitration. However there is no need for second level of arbitration because in the first level arbitration it is ensured that only one VC per input will win the grant for packet traversal. Hence grant signal generated can be used for enabling crossbar link.

Combined VC and Switch Allocation (Becker, 2012:91-92)**:** In this scheme, instead of input active VCs, head flits request the switch allocator for granting crossbar link. Once the flit wins a switch allocation, Output Virtual channel is assigned to input VCs. Because the throughput of the router doesn't depend upon the number of VCs, this scheme can prove quite useful in reducing VC allocation logic. Thus, it might help in the reduction of critical path of the system. It might also help in reduction of chip area and complexity of the system. We will discuss the results of the implementation of this scheme in later section.

TREE ARBITERS

As other monolithic arbiters has been discussed in Yale's paper (Chen, 2015), I am here discussing the another arbiter type which explores the potential of hierarchical coding. A tree arbiter is hierarchial organization of smaller arbiters. (Song at el, 2008) Here mxn-input arbiter is fragmented into m independent groups of n-input arbiters. An m-input arbiter is used to select one of the input groups and the output is used to mask the outputs generated from individual n-input arbiters. Thus arbitration happens between n-inputs instead of mxn inputs. The critical path

of arbiter is proportional to number of ports thus reducing number of port per arbiter will help in reduing critical path. Also the size of the arbiter increases exponentially with number of input ports, thus this implementation can provide significat saving in area. Results of the tree arbiter implementation will be discussed in a later section.

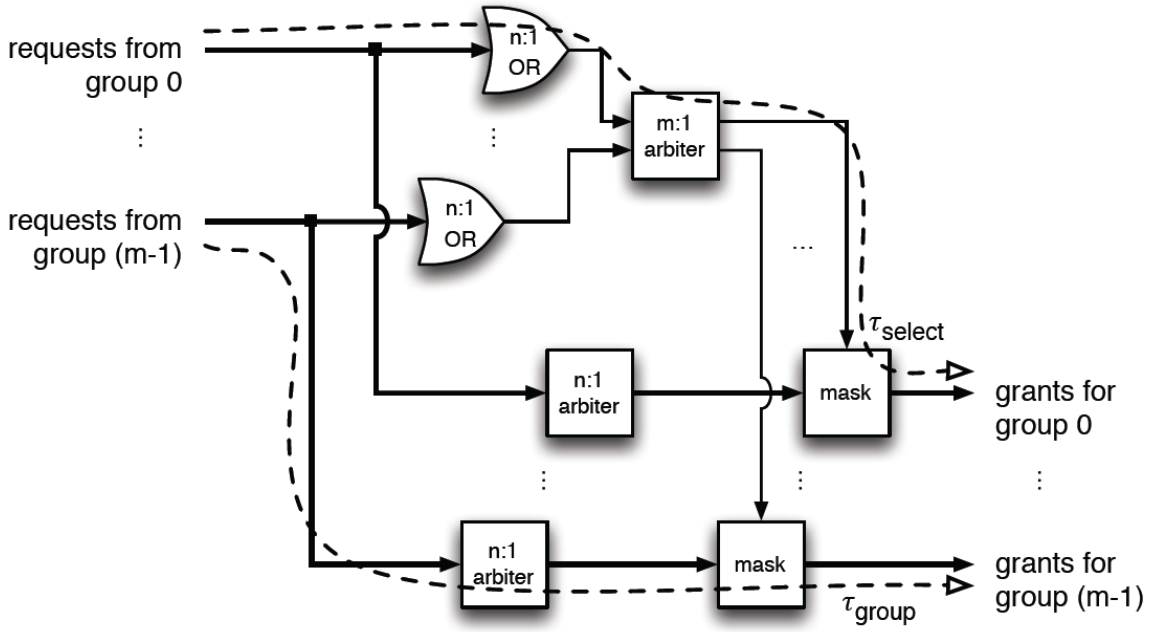


Figure 2. Tree Arbiters (Becker, 2012)

III. METHODS AND MATERIALS

Network router design is an Application Specification Integrated Circuit (ASIC) design. ASIC design follows a standard design methodology which includes four steps: Verilog code and validation, logic synthesis, floor planning, and place-and-routing (PnR).

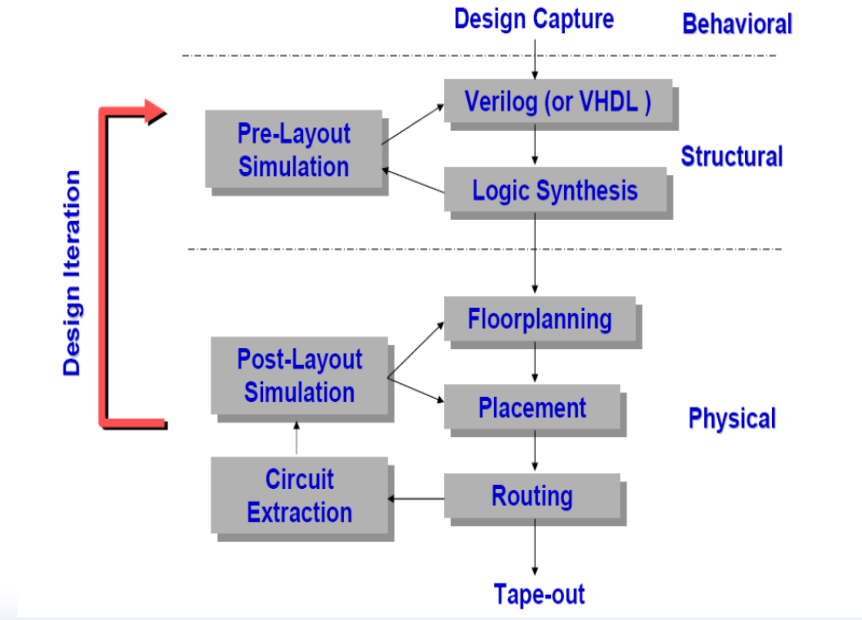
Verilog code and Validation: First task is abstract level implementation of our router design. This is done by writing synthesizable behavioral Verilog code. Next step is to validate the

behavior of Verilog code. This can done by creating testbench which generates random input signals and feed it to the design and monitor the output. We used VCS as a Verilog simulator.

Logic Synthesis: Second, logic synthesis procedure converts behavioral code into a circuit made of logic gates. Logic synthesis uses a standard cell library which contains implementation of basic logic gates like Nor, Nand etc. It also contains macro cell like adders, multipliers or SRAMs. Logic synthesis estimates the delay, area, and power of the design but it is not accurate as routing information of the signals are missing. Design Compiler tool is used for the logic synthesis.

Floor-Planning: Third, Floorplan is used to define the coordinates of the different design blocks in the chip. Floor-planning is generally used in hierarchical approach of integrated circuit design (explained in Yale's paper). There are certain constraints which need to be followed while designing a chip, floor-planning helps in achieving them. The design tool used for floor-planning is ICC-PAR.

Place-and-Route: Fourth, we sought to achieve design functionality by finding an optimal connection between different blocks. Place-and-route is one of the most essential step of the design automation flow. Final layout of the design is obtained after this step. To achieve certain chip performance, placement optimization, clock tree synthesis, and routing optimization needs to be done. All these steps become critical with increase in the complexity of the design. It has been observed that routing become very critical with increase in number of connections. Run time for the tool also increases exponentially. Tool used for PnR is ICC-PAR.



The ASIC design flow

IV. RESULTS AND DISCUSSION

As discussed in above sections, in this project we aim to explore design space for 64 port network routers. Our objective is to improve the performance of the system. Performance is measured in terms of throughput. Along with throughput other main metrics for router designing are area and power consumptions. We aim to deduce the best schemes for the implementation of input buffers, switch and virtual channel allocators and cross-bar switches, which can help in the improvement of performance of the system.

Throughput:

Throughput is the amount of bits router can transfer from input to output port per second. This is the preliminary performance metric of the router.

Throughput = Number of ports * Flit_width * clk_frequency.

According to the above equation, to achieve maximum throughput there are three parameters that needs to be maximized. Figure-3 shows that for 5 port design with increase in flit width Clk period remains almost constant but area increase linearly from 64 to 128 flit size and exponentially from 128 to 256 flit size. Also increasing the flit size or number of ports is same in terms of routing. But with increase in number of ports, number of hop count in interconnect network decreases which will increase the overall throughput of the network. Figure-4 shows the trend of Cycle time of the router with increase in router radix with flit width of 64. Cycle time increases from 5 to 32 but remain constant from 32 to 64, thus there will be substantial improvement in throughput by using 64 radix routers. Hence we fixed the flit size to 64 and radix of the router as 64. Thus to improve throughput, we need to increase operating Clock frequency.

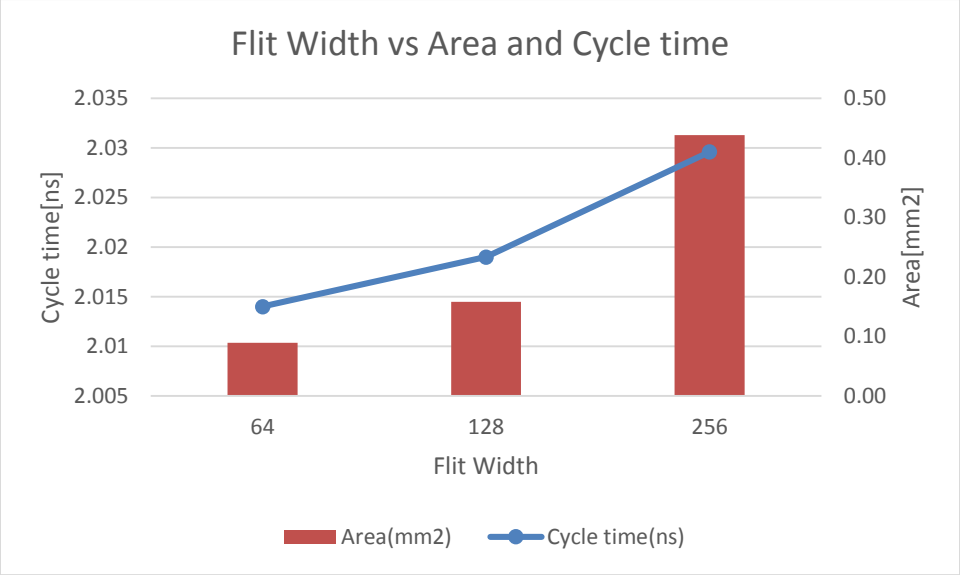


Figure 3

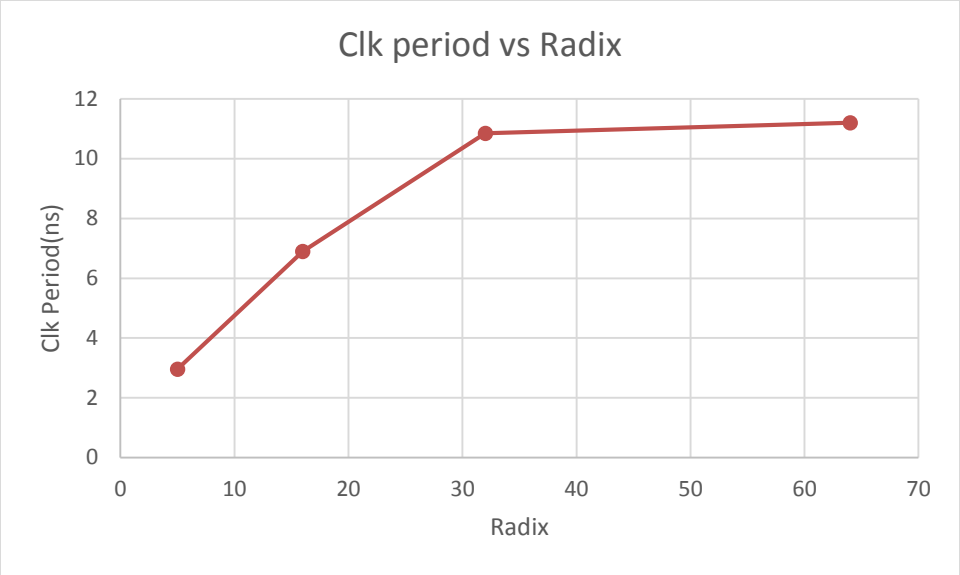


Figure 4

Along with cycle time, it was important to study the area progression with router radix. This will provide insights for the improvement of network performance. Figure-5 compares the area of different blocks with increase in radix of the router. Input buffers and crossbar switches consumes the majority of the chip area. Hence to reduce the area, we proposed the implementation of SRAMs as input buffers. Results has been discussed in Surabhi's paper. (Kumar, 2015) For Crossbar switches, different scheme has been implemented and results are reported in Figure-6. Results shows that distributive Mux is the most area efficient scheme for crossbar switch implementation but it has substantially increased the critical path of the design. To study the cause for latency in distributive mux based design, I investigated the layout and checked the placement of router blocks. Figure-7 and Figure-8 shows the layout of 64 port router with Mux based and distributive mux based crossbar implementation respectively. Yellow, Green and Red represents the allocator, crossbar and input buffer respectively. The placement of crossbar and allocator is interchanged in both designs, this could be the cause of increase in

critical path. Hence by changing the floorplan, we might reduce critical path for distributive Mux crossbar based router.

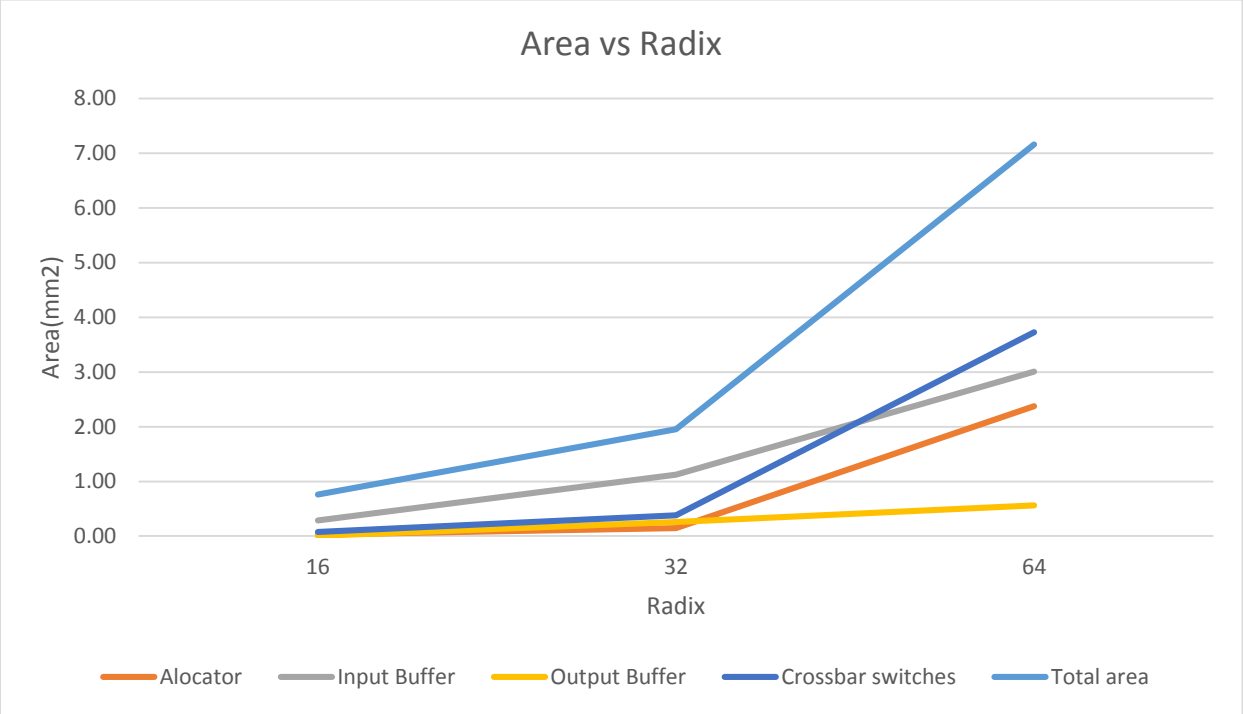


Figure 5

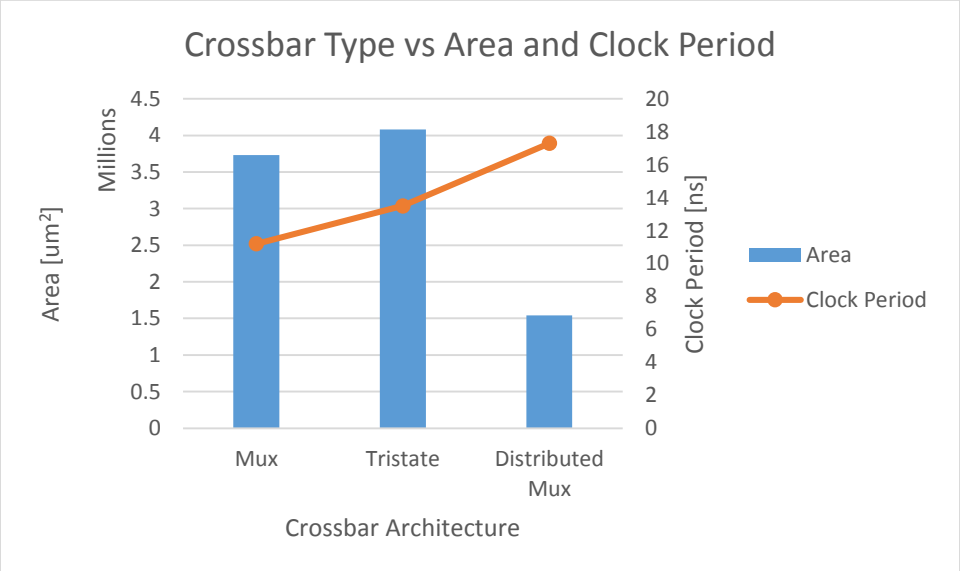


Figure 6

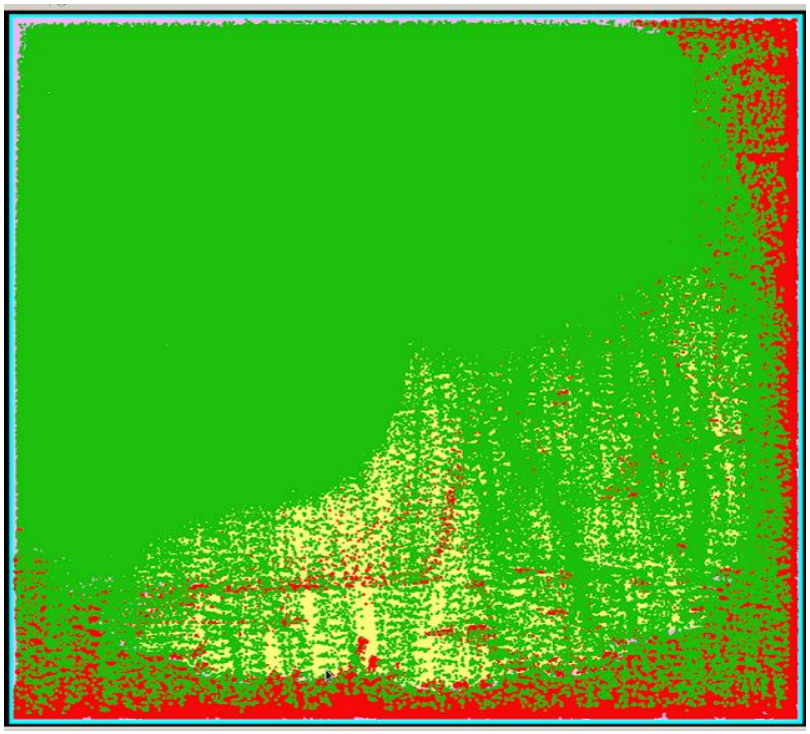


Figure 7. Mux Crossbar based router design

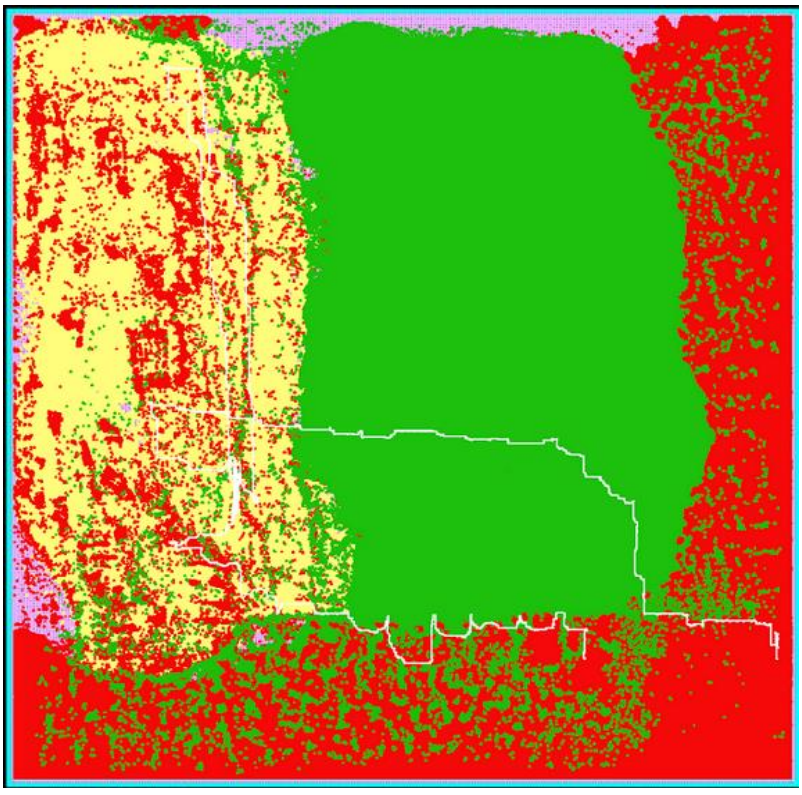


Figure 8. Distributive Mux Crossbar based router design

Critical Path:

Allocator possess the critical path in the router. By choosing Combined allocator over Separate allocator we improved cycle time. (Figure-9) It is also better in terms of area and power.

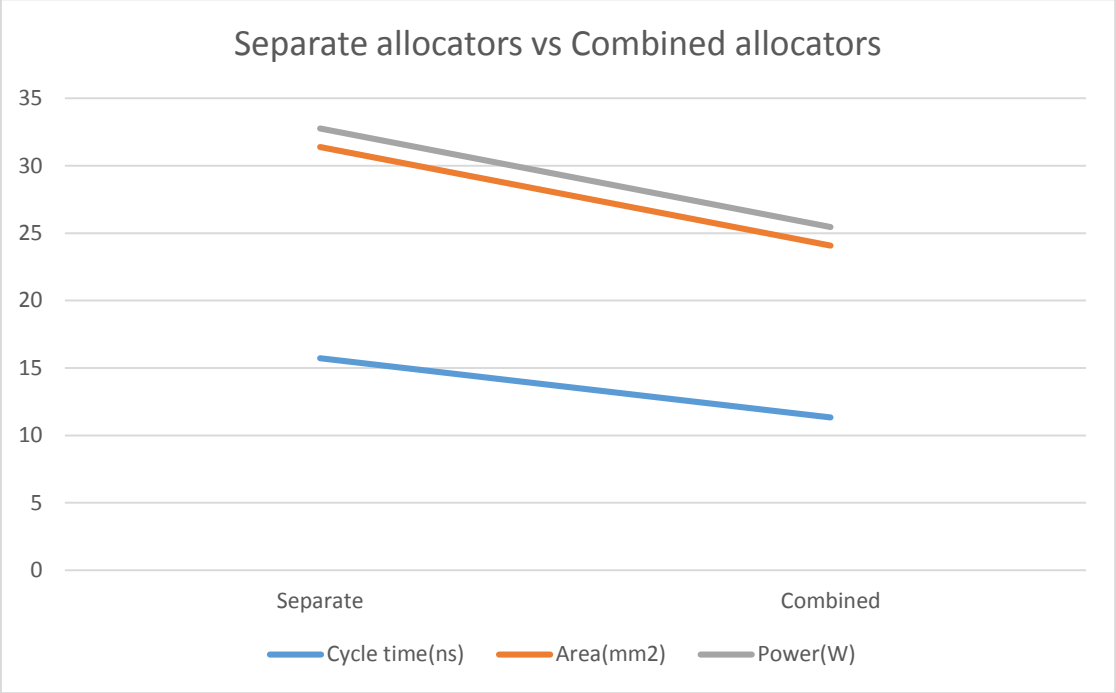


Figure 9

Further I investigated the allocator path and found that major block in allocator is arbiter. Gnt signal which is generated for switch traversal is basically the output of arbiter, and with the increase in number of ports delay increases for the Gnt generation. Hence I synthesized different arbitration schemes and studied area vs cycle time for these arbiters. Figure-10 and Figure-11 shows the area vs cycle time and power consumption vs cycle time comparison for 32 port arbiter design respectively. Result shows that in case of monolithic arbitration scheme Round-robin algorithm is the best. Area penalty in case of Matrix arbiter makes it unfavorable to use in router. Later similar results has been collected for 64 port arbiters. (Figure-12) I implemented

tree arbiters on Verilog and synthesized the code. I have implemented two versions of tree arbiters, one in which basic arbiter is Round-Robin (RR) and another has Matrix. Results shows that tree arbiters achieve faster cycle time with minimal area overhead, although for relaxed cycle time monolithic RR is appropriate. The Matrix arbiter has better matching but due to area and power overhead it is less popular. By using matrix arbiter as a building block in tree arbiter we can achieve better matching with less overhead in power and area. In conclusion, RR based tree arbiter is best in terms of area with improved speed while Matrix based tree arbiter scheme has best timing and area has been reduced significantly.

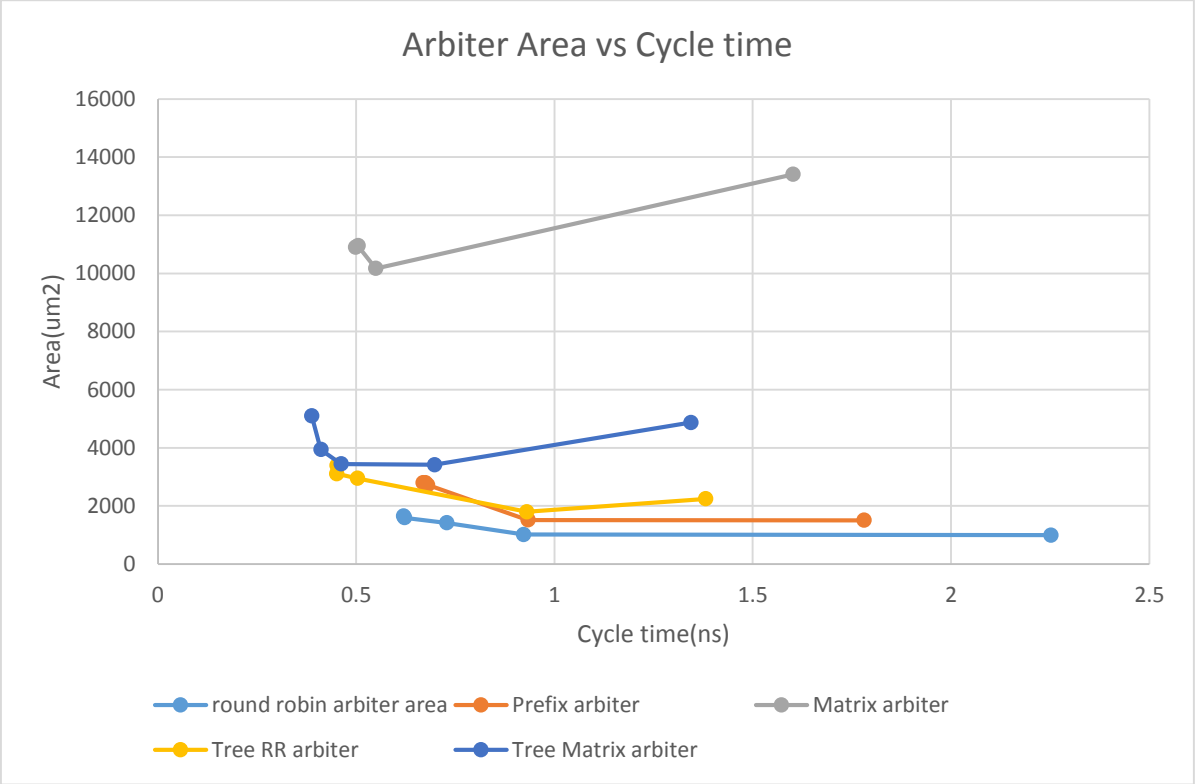


Figure 10

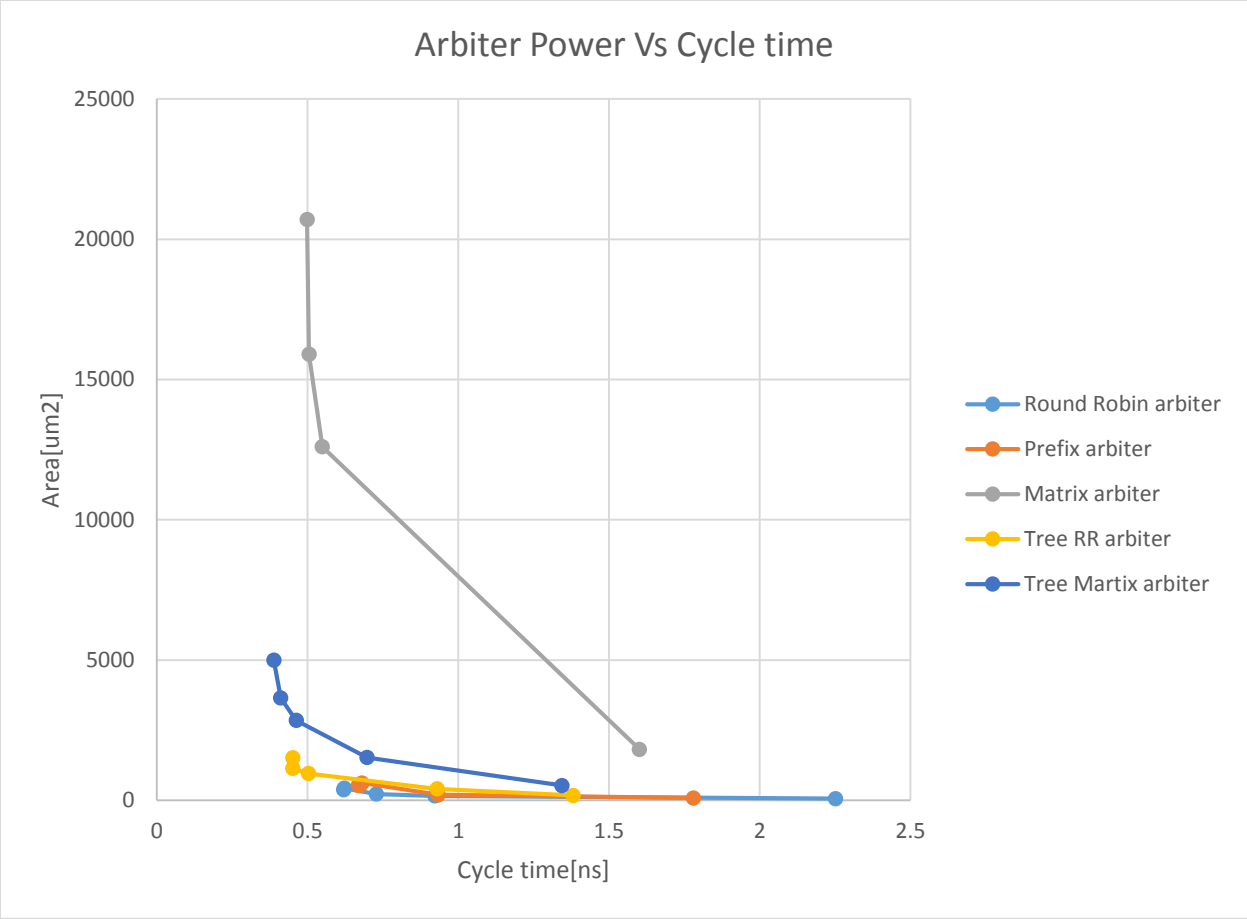

Arbiter Power Vs Cycle time
25000
20000
15000
10000
5000
0
Area[um2]
0
0.5
1
1.5
2
2.5
Cycle time[ns]
Round Robin arbiter
Prefix arbiter
Matrix arbiter
Tree RR arbiter
Tree Martix arbiter

Figure 11

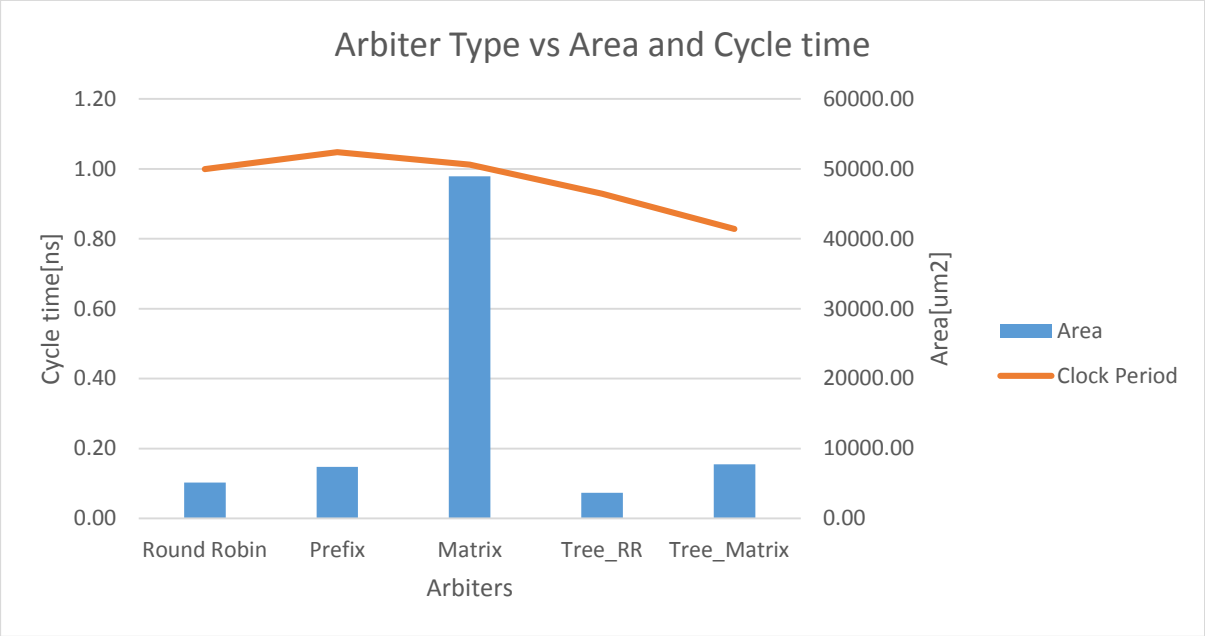


Figure 12

5. Concluding Reflections

In the course of time, we have explored the design space for 64 port router design. We are able to achieve throughput of 0.37 terabit/sec of router with area of 12 sq. mm. While working on the design, we figured out that while designing high radix routers, signal routing is the major concern. Either it is Crossbar module, Allocators or Arbiters, the speed is limited due to the routing congestions. Buffers/Inverters consumes around 50% of chip area and 25% is due to interconnects. Thus to improve the router performance, major architectural changes which reduce the signal routing are required. Distributive Mux scheme for crossbar switches needs to be explored for the improvement. As arbiters are the critical for performance improvement, research is required for better arbitration schemes. Tree arbiter implementation has been done which

shows significant improvement over conventional schemes, but due to high run time of 64-port design, we were not able to synthesis whole router integrated with tree arbiters. Thus in future work, integration of tree arbiter on router microarchitecture can be performed.

With the work we have done, future project group can continue the project by cloning our design directory. As the major concern we faced in this project was tool run-time, we worked a lot on the improvement of it. With hierarchal design approach, someone can advance design exploration for high radix routers.

Apart from technical learning, this project helped us to learn collaboration and team work. While working in a team of five members, I learnt the importance of knowledge sharing. As all of us were from different undergraduate background, this brought multiple technical skills on our project. We helped each other to achieve our project goal. Although we were not able to achieve the goals we anticipated at the beginning of our project, we have made a substantial progress towards it. I hope the work done by our team will enable future project groups to achieve the performance targets for the 64 port router design.

Works Cited

1. Becker, Daniel. “Efficient microarchitecture for network-on-chip routers”. Doctoral dissertation submitted to Stanford University. August 2012. http://purl.stanford.edu/wr368td5072

2. Daly, William, and Brian Towles. *Principles and Practices of Interconnection Networks.* San Francisco: Morgan Kaufmann Publishers, 2004.

3. Clark, Don. “Startup has big plans for tiny chip technology”. *Wall Street Journal*. 3 May 2011. Accessed 5 April 2015

4. Binkert, N.; Davis, A.; Jouppi, N.; McLaren, M.; Muralimanohar, N.; Schreiber, R.; Ahn, Jung-Ho, "Optical high radix switch design," *Micro, IEEE* , vol.32, no.3, pp.100,109, May-June 2012.

5. Boyland, Kevin. 2013 IBISWorld Industry Report OD4540: Electronic Design Automation Software Developers in the US. http://www.ibis.com, accessed February 13, 2015

6. Broadcom. Broadcom Delivers Industry's First High-Density 25/100 Gigabit Ethernet Switch for Cloud-Scale Networks. Press Release. N.p., 24 Sept. 2014. Web. 1 Mar. 2015. <http://www.broadcom.com/press/release.php?id=s872349>.

7. “Computing: Battle of the Clouds”. *The Economist*. 15 October 2009. http://www.economist.com/node/14644393?zid=291&ah=906e69ad01d2ee51960100b7fa502595

8. Deangelis, Stephen F. “Closing in on Quantum Computing”. *Wired*. 16 October 2014. http://www.wired.com/2014/10/quantum-computing-close/

9. Handy, Jim. 2014 Semiconductors A Crazy Industry. http://www.forbes.com/sites/jimhandy/2014/02/11/semiconductorsacrazyindustry2/, accessed February 16, 2015

10. Hoover’s. “Semiconductor and other electronic component manufacturing: Sales Quick Report”. http://subscriber.hoovers.com/H/industry360/printReport.html?industryId=1859&reportType=sales, accessed February 11, 2015

11. Hulkower Billy, “ Consumer Cloud Computing- Issues in the market”, Mintel (2012)

12. Kahn, Sarah. 2014 IBISWorld Industry Report 33421: Telecommunication Networking Equipment Manufacturing in the US. http://www.ibis.com, accessed February 10, 2015

13. Kahn, Sarah. 2014 IBISWorld Industry Report 51721: Wireless Telecommunications Carriers in the US. http://www.ibis.com, accessed February 26, 2015

14. Porter, Michael. “The Five Competitive Forces That Shape Strategy”. *Harvard Business Review*. January 2008.

15. “The Server Market: Shifting Sands”. *The Economist*. 1 June 2013. http://www.economist.com/news/business/21578678-upheaval-less-visible-end-computer-industry-shifting-sands

16. Ulama, Darryle. 2014 IBISWorld Industry Report 33441a: Semiconductor & Circuit Manufacturing in the US. http://www.ibis.com, accessed February 10, 2015

17. Ulama, Darryle. 2014 IBISWorld Industry Report 33329a: Semiconductor Machinery Manufacturing in the US. http://www.ibis.com, accessed February 10, 2015

18. Ulama, Darryle. 2014 IBISWorld Industry Report 33411a: Computer Manufacturing in the US. http://www.ibis.com, accessed February 10, 2015

19. John Kim, William J. Dally, Brian Towles, Amit K. Gupta. “Microarchitecture of a High-Radix Router”. Proceedings of the 32nd annual international symposium on Computer Architecture, 2005: 420-431. Washington, DC. 2005

20. D. Becker and William J. Dally. “Allocator Implementations for Network-on-Chip Routers”. Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, 2009: 1-12. Portland, OR. 14-20 Nov. 2009

21. Juch, Ian. “Technical Contribution – Petabit Swtich Fabric Design”. Capstone Project Report. Berkeley, CA. 2015

22. Mistry, Jay. “Technical Contribution – Petabit Swtich Fabric Design”. Capstone Project Report. Berkeley, CA. 2015

23. Chen, Yale. “Technical Contribution – Petabit Swtich Fabric Design”. Capstone Project Report. Berkeley, CA. 2015

24. Kumar, Surabhi. “Technical Contribution – Petabit Swtich Fabric Design”. Capstone Project Report. Berkeley, CA. 2015

25. Zhaohui Song, Guangsheng Mal and Dalei Song, “Low power circuits for NoC-Based SoC Design”. 9th International Conference on Solid-State and Integrated-Circuit Technology, 2008: 2180 - 2183. Beijing. 20-23 Oct. 2008