Petabit Switch Fabric Design

Surabhi Kumar

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2015-143
http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-143.html

May 15, 2015

University of California, Berkeley College of Engineering

MASTER OF ENGINEERING - SPRING 2015

Electrical Engineering and Computer Science

Integrated Circuits

Petabit Switch Fabric Design

Surabhi Kumar

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

Table of Contents

1. 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.

2. 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.

Works Cited

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

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

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

3. Industry and Market Trends

1 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).

2 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.

3 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.

4 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.

5 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.

Works Cited

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.

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

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>.

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

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

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

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

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

Kahn, Sarah. 2014 IBISWorld Industry Report 33421: Telecommunication Networking

Equipment Manufacturing in the US. http://www.ibis.com, accessed February 10, 2015

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

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

“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

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

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

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

4. 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.

5. Technical Contributions

1 Project Overview and Context

In the project we aim to build a high radix router by doing design space exploration of a router's architecture. Taking a base design, we will be changing different components of the design like buffer, allocation schemes and arbitration schemes, and number of ports to build a high end router integrated circuit .We started the project with an open source Verilog code from Network on Chip research project at Concurrent VLSI Group at Stanford University. We have pushed the Verilog code through Synopsys design flow to get area, power and timing information of the chip and optimize the base design . To accomplish aforementioned task we divided our project into different parts.

One part of the project is to setup the tools for the project and to optimize the run time of the tools. . Large wire congestions in a high radix router over strain ICC Compiler, the placement and routing compiler of Synopsys, and it takes long time to complete routing for large designs . Hence for the project we needed to setup the tools such that routing is completed in a reasonable time (around one day).

The second part of the project is to study the design to identify the bottlenecks in terms of area, power and critical path. Synopsys compiler tools generate reports on different metrics of chip like area, power and timing which were used to identify key bottlenecks and optimize the design to reduce the bottlenecks.

The final part of the project is to replace synthesizable registers used in base design with Static Random Access(SRAM) for the buffers since area analysis showed that buffers will occupy a large part of the chip and using SRAM will optimize chip area since SRAM is a denser memory than an array of flip flops. The different parts have been divided between teammates

depending on the tasks that have to be completed for each part. Ian Juch, Jay Mistry and Yale Chen are working on setting up the tools. Bhavna Chaurasia is working on studying the design to reduce critical path. I have worked on integrating SRAM buffer in the base design.

2 Literature Review

A thorough literature review was performed to study the internal architecture of a router and current research and challenges in the field of router design. The main focus of literature review was Daniel Becker's dissertation thesis at Stanford University and book on interconnection networks by William Dally. The thesis is about building an efficient microarchitecture for network on chips routers. In the first part, Becker evaluates different Virtual channel and Switch allocation architectures for delay, power and area. In the second part of the thesis the author describes static and dynamic buffer (memory) management schemes and the network performance and cost of implementing the schemes. This section discusses microarchitecture of a router, buffer management schemes and their results in Becker's thesis and current research in buffer design for routers. To read Becker's results on allocators please refer to my teammate Jay Mistry' paper. Similarly to read about Arbiter, Switch Allocation and Virtual Channel Allocation refer to Yale Chen, Bhavana Chaurasia and Ian Juch's papers respectively.

1 Router Overview

Figure 1 shows the internal microarchitecture of a router. The main components inside a router are input ports, virtual channel allocator, switch allocator, cross bar and output ports. A flits path through router pipeline can be briefly outlined by following steps. Firstly, the head flit of a packer enters a router's input port and is assigned an output port by the routing algorithm. The head flit then progresses to next stage of pipeline where it waits unless it is assigned a virtual channel at output port with buffer space available. Virtual channel is the state of each packet

which is at input port of the router. Virtual channel state of a packet gets updated as flits of the packet travel the router pipeline. Once the flit is assigned a virtual channel it progresses to the switch allocation stage of the router where the allocator resolves conflicts between various input port VC's to traverse crossbar switch and reach destination output port. Switch allocation stage ensures that no two input ports receive a grant to traverse switch for same output port. In the final stage of the router pipeline, the flit traverses the pipeline and waits at output port until it can leave the router to go to the next networking device.

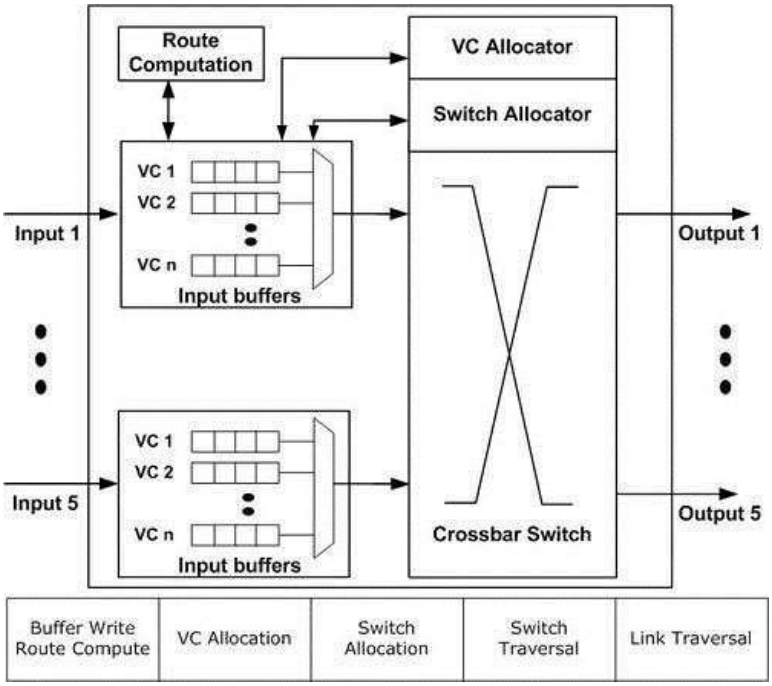

Figure 1: Microarchitecture of a router

.

2 Buffer Management schemes in Becker's work

In a router, buffer is divided between different virtual channels. Virtual channels have two advantages. One it enables flits from different plackets to bypass each other and improve channel utilization and hence channel throughput. In a router without virtual channels, flits from earliest arrived packet that are waiting on a resource block flits from latter arrived packets that are ready to traverse switch and go to output port. This is called Head of Line blocking. One analogy to understand benefit of virtual channel is that of dedicated traffic lanes at an intersection. Car drivers don't have to unnecessarily wait for heavy vehicle drivers to move in the traffic. Hence there is better flow of traffic (Becker). Virtual channels help achieve better throughput and prevent head of line blocking scenarios in network. Second advantage of virtual channel is that different flows of traffic can be constrained in different subset of network's VCs which enables implementation of deadlock avoidance schemes. Deadlock is another unwanted situation in a network in which packets are stuck because they are waiting on each other to release resources.

The depth and number of virtual channels a buffer is divided into greatly impacts network performance. Increasing number of virtual channel to reduce deadlock avoidance increases network performance, it also increases router cost as each VC requires control logic, state logic and additional buffers to meet capacity requirements. Number of VCs affects complexity of VC allocation and therefore router's cost and cycle time.

Depth of VC affects network latency under modest load. Under modest load, credit stall is the main bottleneck in a network. Credit stall is the time a flit waits for buffer space to be available at output VC buffer. Credit round trip delay is the time between successive credits

received for same buffer slot. The depth of VC buffer should ideally be such that it covers the credit round trip delay. The relation between VC depth and credit round trip delay is shown in equation 1.

$$F * L_f > t_{crt} * B \qquad \text{Equation1}$$

F is number of flits that can be stored in a VC

L_f is flit size

t_{crt} is credit round trip delay

B is bandwidth

Meeting condition in equation 1 does not lead to credit stall

The author discusses two buffer management schemes -static buffer management and dynamic buffer management. In static buffer management, total buffer per input port is divided equally amongst VCs. Static buffer management scheme uses low complexity logic. It is implemented as circular buffers with management logic comprising of a pair of index registers that point to buffer location which will be read from and written to next. Hence it has shorter critical path, however such a scheme leads to "under- utilization of buffer when network load is not evenly distributed across VCs. Moreover, in order to avoid credit stalls, each VC must be allocated enough buffer space to cover round trip delay which causes buffer space to scale linearly with number of VCs" (Becker).

Dynamic buffer management improves buffer utilization by allowing buffer space to be shared among VCs unevenly. Better buffer utilization is achieved by allowing the effective number and depth of VC to vary depending on network traffic. Dynamic buffer either improves performance for a given buffer size or gives comparable performance as a static buffer at a lower cost. Dynamic buffer is implemented as a variable length queue per VC. Based on the author's results, dynamic buffer has more logic overhead compared to static buffer (Becker).

To test buffer management schemes, the author models interconnect network traffic using simulator tool called BookSim developed at Concurrent VLSI group at Stanford as a part of Network on Chip research. The author tests network throughput using static and dynamic buffer management and presents tradeoffs of the two schemes. Firstly, the author states that at smaller buffer size, dynamic buffer management gives better throughput with reduced cost of buffers. Experimental results have indicated that when dynamic and static scheme have been set up to have same number of virtual channels, the difference in buffer cost is 11% while dynamic buffer management has performance advantage of 30%. There are diminishing returns to increase in performance for large sized buffers. This is because for large buffers, credit stalls are less and there is less increase in network performance. However, dynamic buffer has advantages over static buffer for large buffer sizes when number of virtual channels is large. For large number of virtual channels, individual depth of each static VC may not be large enough to avoid credit stalls and dynamic buffer will vary VC depth depending on demands of traffic to meet credit round trip condition.

Hence from the experimental results it can be concluded that dynamic buffer management scheme leads to higher throughput with more VCs but dynamic buffer has more logic overhead than static buffer.

3 Research in router buffer design

“A study of research in the field of router design has shown that buffers account for a large fraction of the overall area and power of NOC routers” (Ling et al;Becker). Hence a good buffer design is important to create an area and power efficient chip. Moreover, because of credit delay, buffers also play an important role in router performance. Buffer design is a widely pursued topic in research field of router design. In this section two papers are discussed which have achieved improvement in buffer utilization and performance in a router at two different levels- buffer management and buffer type used.

In the paper, Router with Centralized Buffer for Network on Chip, the authors Ling Wang, Jianwen Zhang, Xiaoqing Yang, and Dongxin Wen propose a centralized buffer for the router. However, the buffer slots for packets are dynamically allocated depending on traffic patterns. The centralized buffer architecture consist of Head decode unit, input pointer control logic and output pointer control logic which take in an incoming packet, store it in a central buffer and route it to a output port depending on address of head flit. This implementation is similar to Becker's dynamic buffer management scheme except the router in Wang 's paper uses buffer to route input packets to output ports and Becker's router is more sophisticated since it uses a crossbar switch. The author has reported an decrease in buffer use by fifty percent to increase buffer utilization.

State of the art in today's switches and routers are high bandwidth switches and Dong Lin's paper on Distributed Packet Buffers for high bandwidth switches and router discusses the advantages of using a hybrid SRAM and DRAM buffer over just using a purely SRAM or DRAM buffer for high bandwidth requirements. SRAM provides low access time but is less dense than DRAM .DRAM on the other hand is more denser , but its access speed is tooslow,

40 ns. In fact, DRAM access speed decreases by only ten percent every eighteen months and router line rate increases by 100 percent every month (Corbal et al). DRAM will be unable to provide the necessary bandwidth for high throughput routers. The author hence proposes a hybrid SRAM/DRAM buffer and implements and efficient algorithm that coordinates load balancing between the SRAM and DRAM. Author's work overcomes the challenges of time complexity and large SRAM size requirement of previous such designs.

3 Methods and Materials

Our goal for the project is to optimize the buffers in the base router design of Becker. Our study of the base design shows that for sixty four port router, buffers occupy 49.8 % of chip area, while allocator and crossbar occupy the rest (Chaurasia). Hence buffer resources must be utilized efficiently to design area efficient chip. In this section I will describe the methodologies adopted for buffer design.

1 Tools and Memory Compiler

Since we are building a router integrated circuit we have heavily used VLSI tools that are commonly used in chip design. The VLSI design flow consist of RTL simulation which is coding the digital circuit in behavioral Verilog, synthesis which is converting the behavioral code to gate level implementation and placement and routing which is placing the different standard cells in core area and routing wires to connect the standard cells. We had access to Synopsys Education license and used Synopsys VCS simulator, DC synthesis which is Synopsys synthesis compiler and ICC-PAR which is Synopsys routing and placement tool.

To generate SRAM blocks I used CACTI memory compiler. Cacti models memory

access time, cycle time, area, leakage and dynamic power of integrated cache. It takes in a set of input parameters like width and height of memory array, technology node, number of read and write ports etc in a configuration file and generates db and mw files which can be used with DC Synthesis and ICC-PAR respectively.

2 Types of Memory

There are many types of memories available that can be used to implement buffer. In the base design, memory array of flip flops is used to store incoming flit in each cycle since it is a small port design and flip flops can be easily synthesized without building a custom designed memories. Other alternatives for buffer can be a Static RAM which is volatile memory that uses inverters in feedback loop to store data. A master slave flip flop is implemented using twenty two transistors while a SRAM cell is usually implemented using only six (or eight transistors for dual port SRAM) which makes SRAM a very dense memory compared to register file.

In the paper 'Design planning for large SoC implementation at 40nm :Guaranteeing predictable schedule and first-pass silicon success, the author talks about synthesized area of register and SRAM. For 64x32 size memories, SRAM occupy significantly lesser area (10000 micrometer square) than registers (20000 micrometer square). In fact, area of synthesized register grows exponentially with increasing memory size while area of SRAM grows linearly as seen in figure 2. Hence SRAM will be a good choice of memory array to optimize area of a chip, especially for large sized buffers and should bring large savings in area of the chip and we decided to use SRAM for the buffers.

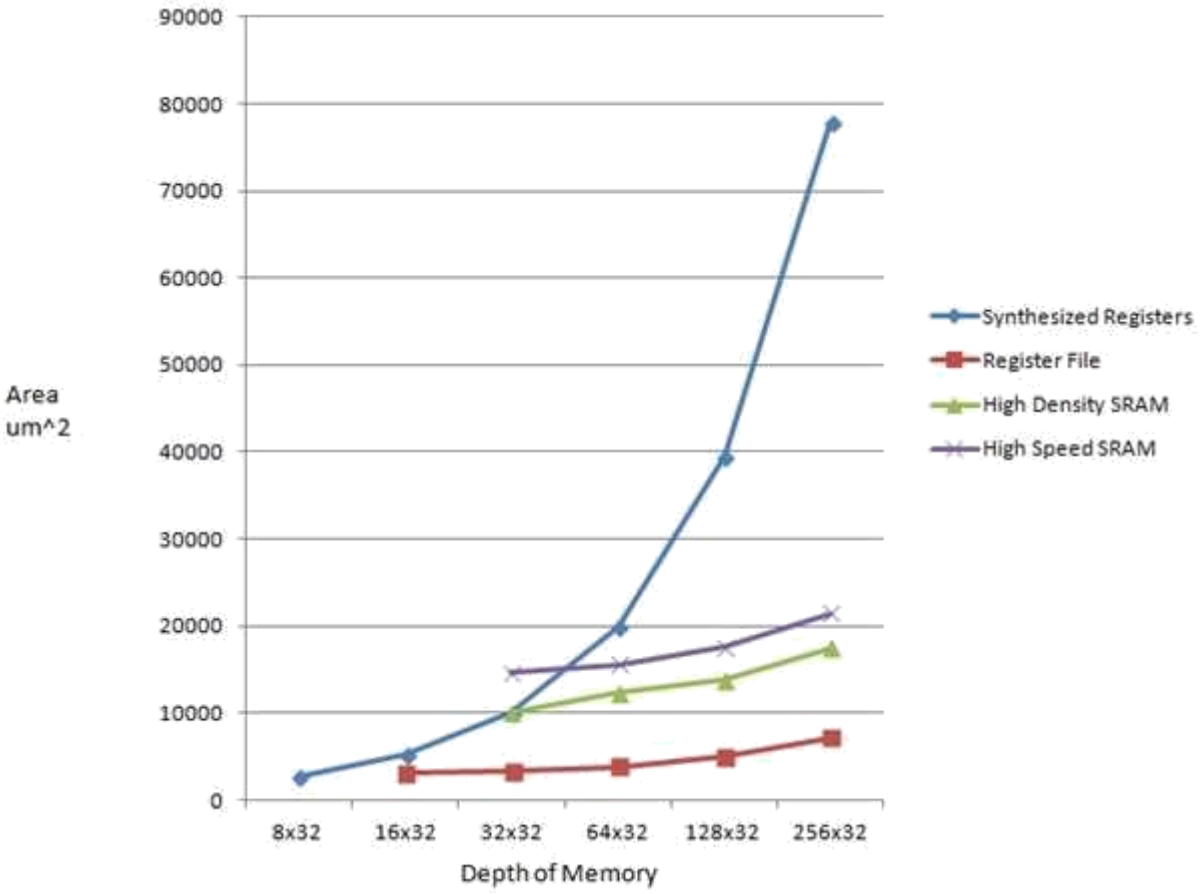

Figure 2: Plot of area vs depth of memory from Dasila's paper

While SRAM is denser memory, SRAM read operation is synchronous and a flip flop's read operation is asynchronous. There are asynchronous SRAMs available which read synchronously at a fast rate that it seems the read operation is asynchronous but they have limited performance in terms of speed. Most high end SRAMs are synchronous. Since we are designing buffer for high radix and high throughput router that will be reading in data at a high rate, we decided to use synchronous SRAM in the design.

3 SRAM synchronous read

One of the technical challenges in the project was to modify the RTL to account for the synchronous read of SRAM which caused the RTL test bench to fail since the memory module in

the base code used registers which were asynchronous read. To resolve this issue two step approach was taken. One was to bypass a register at the SRAM memory generator so that the address to SRAM comes before clock edge. However, bypassing the register was creating a combinational loop which created unknown (Xs in Verilog) values in many signals around the address generation logic. Once I verified that this was not a feasible solution, me and Ian Juch worked on another solution which was to add extra registers at signals before the combinational logic that has SRAM data as its input so that all the signals are synchronized. Ian Juch successfully incorporated the changes and discusses it in his paper.

4 SRAM Implementation

The register file used originally in the design has two read ports and one write port. However, there were only two ports SRAMs available from the memory compiler. Hence to account for the third port, I used two SRAM of size 16x64 per port to design memory module. Both SRAMs stored the incoming flit, and read out different addresses every clock cycle. Figure 3 shows the schematic of the SRAM design. Adding an extra SRAM to read out two flits every clock cycle may seem redundant however multiport SRAM can have large area because of the increased bitline, wordline and two extra read transistors(8T SRAM cell). Hence savings in area by using single large multi ported SRAM will not be significant.

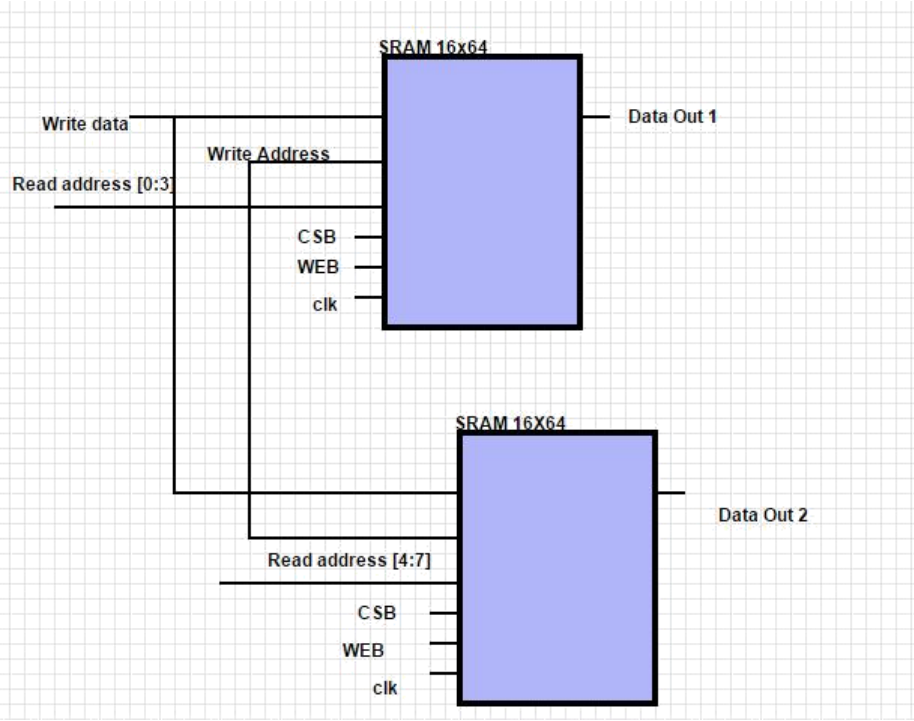

Figure 3: 16x64 SRAM implementation for two read and one write port

5 Small SRAM Arrays

In the paper, Energy Consumption Modeling and Optimization for SRAM, the author talks about different SRAM sizes for energy optimization and delay optimization. According to the paper, for two SRAMs of same size, SRAM with non-square dimensions that have more rows than number of columns will dissipate less energy and SRAM with a square dimension will have less delay and hence access time. Using the results of the paper, smaller SRAM size of 16x32 was generated using Cacti Memory compiler to create a big memory block of size 16x64. Using 16x32 dimension SRAM should reduce power dissipation and access time since number of columns is reduced. We had planned to use smaller SRAMs dimensions like 16x16 but we were limited by the SRAM sizes available by the Cacti compiler. The schematic of the memory

module I implemented to build larger SRAM from smaller SRAM is shown in Figure 4. The memory module of the router was then run through DC synthesis and ICC-PAR with two different SRAM sizes but same flit width to get area, delay and power estimate of the module.

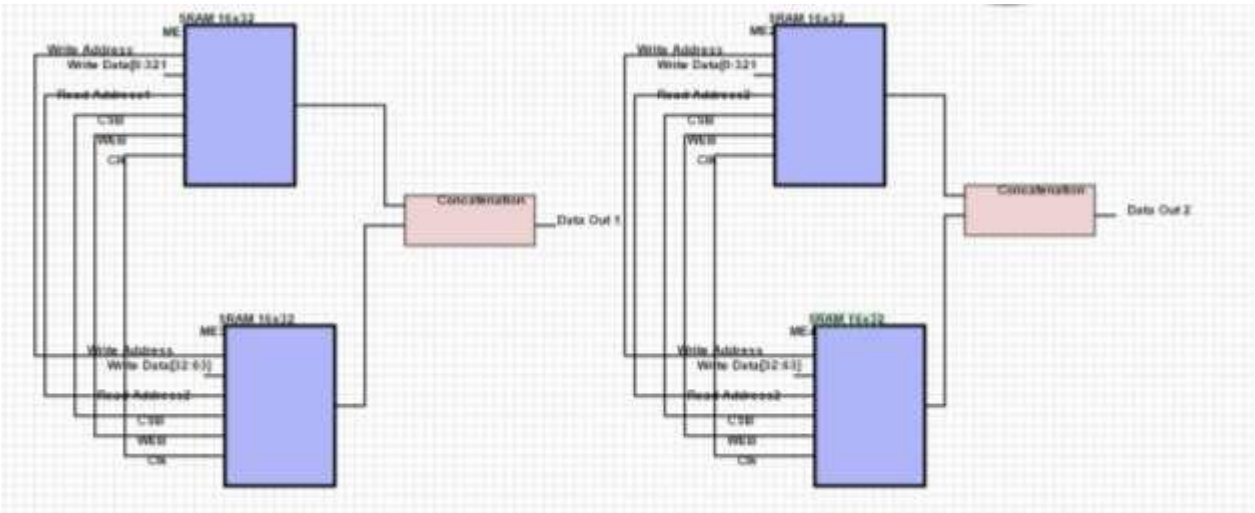

Figure 4: Creating larger memory from smaller SRAMs

4 Results and Discussion

Area, power and clock period results from sixty four port design are shown in Figure 5, Figure 6 and Figure 7. Each plot has four data points, two with flit width set to 32 bits and two with flit width set to sixty four bits. From figure 5, we can see that introducing SRAM buffers has saved area and increased critical path when flit width is sixty four bits. There is 14 % area reduction in 32 flit width router design by using SRAM and 13% area reduction in sixty four bit flit width router design. This is much less than the fifty percent area reduction we got with five port designs with SRAM's instantiated. Since number of ports is the only difference in the two designs this result can be attributed to the fact that synthesis compiler must have used more timing optimized blocks which were area less area efficient to meet the critical path of the design

and hence savings in area from SRAM may not have scaled up from five port design to sixty four port design.

There is 2.7 % increase and 2.7 % decrease in clock period in sixty four flit width design and thirty two flit width design respectively. Since the critical path in the design does not go through SRAM, these results does not indicate anything about SRAM buffer integration and may be caused by other factors like different placement of standard cells by the tools. There is also eleven percent power savings in sixty four flit width SRAM design and ten percent power savings in thirty two flit width SRAM design.

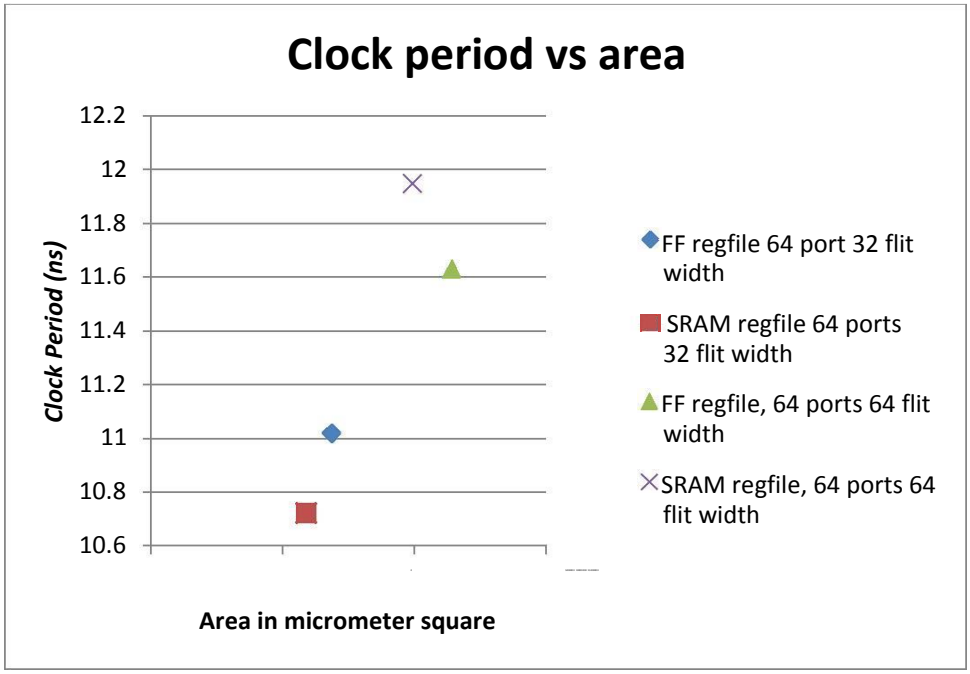

Figure 5

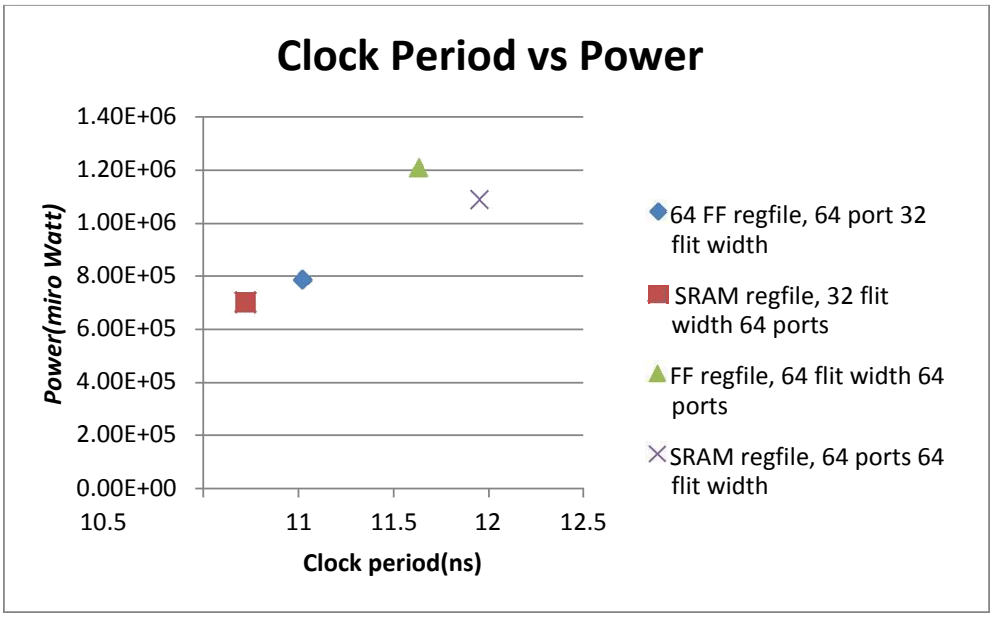

Figure 6

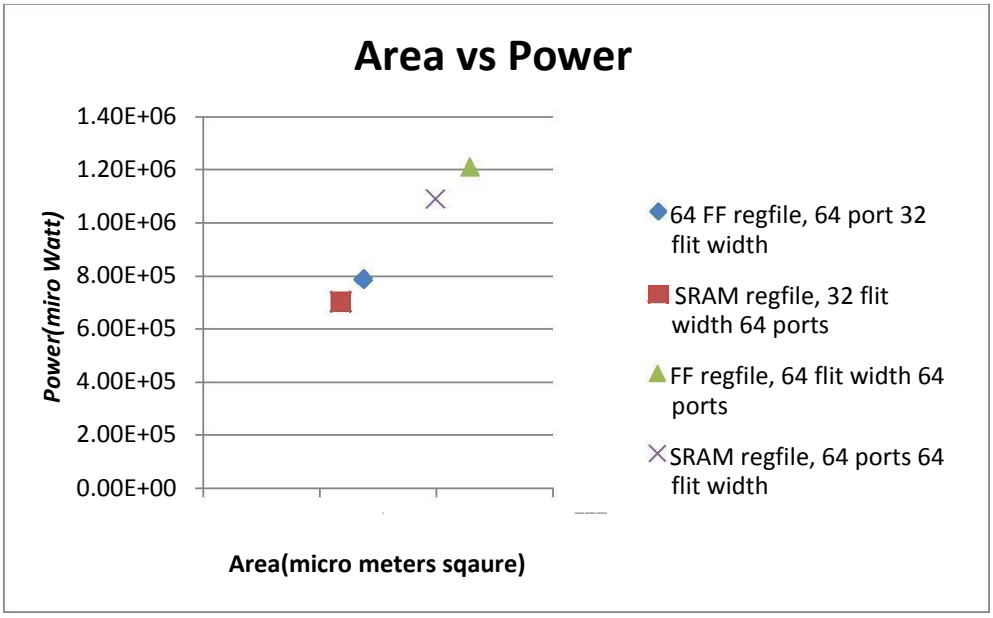

Figure 7

In addition to making the design area and power optimized, the team has achieved other significant results in the project. Firstly, we have reduced the run time of the tools significantly by optimizing the tools especially for our project which is a design with high wiring congestion caused by crossbar. We have also implemented Hierarchical place and route for smaller port designs which is expected to reduce the run time of tools by letting DC Synthesis and ICC-PAR to reuse modules from previous runs.

We have also identified the critical path in the design to pass through allocators and arbiters and have discovered that using Tree Arbiters can be useful in reducing the critical path. This is currently work in progress and can be considered as future work for further improving this project.

5 Conclusions

In conclusion , we have been successful in achieving many key milestones in the project even though we did not accomplish our original goal of building a high radix router with 256 ports. We have overcome many challenges of designing large digital circuits. Firstly, we accomplished a good understanding of David Becker's work by reading his thesis and surveying other literature in router design. Next we succeeded in setting up complex Synopsys VLSI tools and optimizing the tools to reduce run time because of high congestion in the design. We also implemented important changes in the router's microarchitecture to get a better design of router that has higher throughput and is more area and power efficient. In addition to tacking technical challenges in the project, we have also established the market and IP strategy that would be necessary in order to launch this router in the market. We have combined leadership and technical skills we have learnt in our program.

Based on team's results for future work I will recommend to estimate the credit round trip delay of the router to size the buffers accordingly, implementing tree arbiters for reducing critical path and using the current setup of Hierarchical place and route approach and extending it for sixty four port router.

6. Works Cited

1. Ling Wang, Jianwen Zhang, Xiaoqing Yang, and Dongxin Wen. Router with Centralized Buffer for Network-on-Chip. In Proceedings of the 19th Great Lakes Symposium on VLSI, pages 469–474, 2009.

2. Dasila Bhupesh, "Design Planning for large SoC implementation at 40 nm: Guaranteeing predictable schedule and first pass silicon success". *EDN Network*. Web. 7 May 2013

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

4. Evans Robert, Franzon Paul, "Energy Consumption Modeling and Optimization for SRAM's", *IEEE Journal of Solid State Circuits*, Vol 30, No.5 ,May 1995

5. Binkert, N.; Davis, A.; Jouppi, N.; McLaren, M.; Muralimanohar, N.; Schreiber, R.; Ahn,

6. Jung-Ho, "Optical high radix switch design," *Micro, IEEE* , vol.32, no.3, pp.100,109,

7. May-June 2012.

8. Boyland, Kevin. 2013 IBISWorld Industry Report OD4540: Electronic Design

9. Automation Software Developers in the US. http://www.ibis.com, accessed February 13, 2015

10. Broadcom. Broadcom Delivers Industry's First High-Density 25/100 Gigabit Ethernet

11. 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>.

12. “Computing: Battle of the Clouds”. *The Economist*. 15 October 2009.

13. http://www.economist.com/node/14644393?zid=291&ah=906e69ad01d2ee51960100b7fa502595

14. Deangelis, Stephen F. “Closing in on Quantum Computing”. *Wired*. 16 October 2014.
15. http://www.wired.com/2014/10/quantum-computing-close/
16. Handy, Jim. 2014 Semiconductors A Crazy Industry.
17. http://www.forbes.com/sites/jimhandy/2014/02/11/semiconductorsacrazyindustry2/,
18. accessed February 16, 2015
19. Hoover’s. “Semiconductor and other electronic component manufacturing: Sales Quick
20. Report”. http://subscriber.hoovers.com/H/industry360/printReport.html?industryId=1859 &reportType=sales, accessed February 11, 2015
21. Hulkower Billy, “ Consumer Cloud Computing- Issues in the market”, Mintel (2012)
22. Kahn, Sarah. 2014 IBISWorld Industry Report 33421: Telecommunication Networking
23. Equipment Manufacturing in the US. http://www.ibis.com, accessed February 10, 2015
24. Kahn, Sarah. 2014 IBISWorld Industry Report 51721: Wireless Telecommunications
25. Carriers in the US. http://www.ibis.com, accessed February 26, 2015
26. Porter, Michael. “The Five Competitive Forces That Shape Strategy”. *Harvard Business Review*. January 2008.
27. “The Server Market: Shifting Sands”. *The Economist*. 1 June 2013.
28. http://www.economist.com/news/business/21578678-upheaval-less-visible-end-computer industry-shifting-sands
29. Ulama, Darryle. 2014 IBISWorld Industry Report 33441a: Semiconductor & Circuit
30. Manufacturing in the US. http://www.ibis.com, accessed February 10, 2015
31. Ulama, Darryle. 2014 IBISWorld Industry Report 33329a: Semiconductor Machinery
32. Manufacturing in the US. http://www.ibis.com, accessed February 10, 2015
33. Ulama, Darryle. 2014 IBISWorld Industry Report 33411a: Computer Manufacturing in

the US. http://www.ibis.com, accessed February 10, 2015

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

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

36. Chaurasia Bhavana, "Technical Contributions-Petabit Switch Fabric",Masters of Engineering thesis submitted to UC Berkeley in 2015.
37. Juch Ian, "Technical Contributions-Petabit Switch Fabric",Masters of Engineering thesis submitted to UC Berkeley in 2015.
38. Chen Yale, "Technical Contributions-Petabit Switch Fabric",Masters of Engineering thesis submitted to UC Berkeley in 2015.
39. Mistry Jay, "Technical Contributions-Petabit Switch Fabric",Masters of Engineering thesis submitted to UC Berkeley in 2015.

40. J. Corbal, R. Espasa, and M. Valero, “Command Vector Memory Systems: High performance at Low Cost,” Proc. Int’l Conf. Parallel Architectures and Compilation Techniques, pp. 68-77, Oct. 1998.

41. D. Lin, M. Hamdi, and J. Muppala, “Distributed Packet Buffers for High-Bandwidth Switches and Routers,” IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 7, pp. 1178-1192, July 2012