Exploiting Non-Uniform Access Time in Interconnect Sensitive Cache Partitioning

A Thesis Presented
by
Christopher Cowell

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

July 2003

Department of Electrical and Computer Engineering

Exploiting Non-Uniform Access Time in Interconnect Sensitive Cache Partitioning

A Thesis Proposal Presented

by

Christopher Cowell

Approved as to content and style by:

____

Wayne P. Burleson, Chair

____

Csaba A. Moritz, Member

____

Russell Tessier, Member

____

Seshu B. Desu, Department Head
Department of Computer and Electrical Engineering

Abstract

Exploiting Non-Uniform Access Time in

Interconnect Sensitive Cache Partitioning

July 2003

Christopher Cowell

B.S.E.C.E, Rochester Institute of Technology

M.S.E.C.E, University of Massachusetts Amherst

Directed by: Professor Wayne P. Burleson

Growing wire delay and clock rates limit the amount of cache accessible within a single cycle [3,13]. Cache architectures assume that each level in the cache hierarchy require a uniform access time. As microprocessor technology advance, architects must decide how to best utilize increased resources while accounting for growing wire delays and clock rates. Because on-chip communication is very costly [14], accessing different physical locations of the cache can return a range of hit time latencies. This lack of uniformity can be exploited to provide faster access to cache blocks physically closest to processing elements. More cache is being placed on the chip causing the access time of the closest cache bank to be much less than the access time of the farthest cache bank. Previous research leveraged such non-uniformity by migrating the most likely to be used cache sets into the closer cache banks. This research work focuses on the placement of the cache banks and the interconnection topology that allows each bank to communicate with one another and the processor core.

This research evaluates the performance gain of non-uniform cache architectures, interconnected in a hypercube network, through a detailed cache model, an Alpha 21364 floorplan model and an out-of-order processor simulator. The research methodology generates various cache organizations and timing information given a variety of cache requirements. The cache organization is then manually laid out on the physical floorplan where global wire lengths are manually extracted and modeled in HSpice to obtain the latency due to global wire delay. The generated hit/miss access times along with the global wire latency are simulated with Simplescalar [11] and the SPEC2000 benchmark suite [9]. Initial results compare an S-NUCA cache with a mesh network to a D-NUCA cache with a torus, mesh and hypercube interconnection topology and demonstrate a 43% performance improvement.

Table of Content

ABSTRACT

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1. INTRODUCTION

2. VLSI TREND

2.1 Wire Properties 10

2.1.1 Wire Resistance 10

2.1.2 Wire Capacitance 11

2.1.3 Wire Inductance 13

2.2 Wire Scaling Effect 15

2.2.1 On-chip Wiring Archtitectural Trend

2.2.2 Wire/Dielectric Projection

2.2.3 Scaling the Wiring Layers

2.2.4 Related Interconnect Researh

3. FLOORPLANNING MEMORY ELEMENTS

4. DYNAMIC NON-UNIFORM CACHE ARCHITECTURE COMPONENTS

4.1 Data Mapping 27

4.2 Bank Search 29

4.3 Data Promotion Scheme 30

5. SIMULATION ENVIRONMENT

5.1 Generating the Floorplan 33

5.2 Generating the Bank Timing 34

5.3 Global Wire Delay Table 35

5.4 Simplescalar Extended 36

6. DYNAMIC NON-UNIFORM CACHE ARCHITECTURE PERFORMANCE ANALYSIS

6.1 Technology Study 38

6.2 Topology Study 44

6.3 Data Promotion Study 47

7. CONCLUSION AND FUTURE WORKS

BIBLIOGRAPHY

List of Tables

1. International Technology Roadmap for Semiconductor: 2001 Interconnect Report 18

2. Near Term Scaling [ITRS2002] 19

3. Long Term Scaling [ITRS2002] 20

4. Cache and Chip Layout Assumptions 33

5. Cacti Parameters for 130nm floorplan 34

List of Figures

1. Cross-sectional Close-up of Copper Wire 11

2. Capacitance Models 12

3. Miller's Multiplication 13

5. Delay for Local and Global Wiring vs. Feature Size 16

6. On-chip Wiring Architecture 17

7. Cross Section of Hierarchical Scaling 20

8. Cache Organizations [1] 25

9. Mapping Schemes (a,b,c [1]) 28

10a. Incremental Bank Search 29

10b. Broadcast Bank Search 29

11a. Incremental Promotion 30

11b. Absolute Promotion 31

12. Simulation Flowchart 32

13. Global Wire Delay 35

14. Alpha 21364 (130nm): L2 Cache 2MB 38

15. D-NUCA2 Speedup over S-NUCA2 for 130nm: L2 Cache 2MB 38

16. Alpha 21364 Floorplan 90nm: L2 Cache 8MB 39

17. Hypercube Interconnection Scheme 40

19. Alpha 21364 Floorplan 65nm: L2 Cache 16MB 41

20. D-NUCA2 Speedup over S-NUCA2 for 65nm: L2 Cache 16MB 42

21. Cache Bank Contention for D-NUCA2 43

22. Bank Interconnection Topology Study 44

23. Interconnect Topology Routing Comparisons 45

Chapter 1.

Introduction

Although microprocessors have made serious breakthroughs over the past few decades, transistor scaling was key towards achieving new maximum clock rates and greater chip complexities. According to Moore's Law, the transistor's minimum size, have been shrinking at a rate of about 30% every three years and expects to maintain such development over the next decade [5].

In more detail, the International Technology Roadmap for Semiconductor (ITRS) demonstrates growth in the transistor count, wiring levels, implying an increasing interconnection density with each new technology. Accompanied with the increasing density are global line parasitics such as crosstalk and electromigration due to escalating aspect ratio (height/width) and a thinning wire width. Circuit techniques such as repeaters, boosters and current mode receivers can be used to accelerate signal transmission [?,?,?]. In conjunction with circuit innovation, insightful architectural intervention is necessary in attaining pioneering performance throughout the next computing era [?].

Microprocessors are becoming bigger in size, faster in clock rates, deeper in pipeline depth and more capable of issuing more instructions per cycle. A couple of key problems are the instruction supply and more specifically the data supply. Instruction supply can improve through out-of-order fetching, hybrid branch prediction and trace caches. Data supply improvement can occur through data speculation or larger on-chip caches. Data

speculation improves the data supply by predicting the usage of data and retrieving it early. Larger on-chip cache improves data throughput by placing more unique data closer to the processing elements. These architectural trends require an interconnection network capable of supporting a demanding communication.

Interconnect is the main problem in high performance processors and memory hierarchies [3,5,15]. Previous work demonstrated that very large uniform cache architectures are incapable of supporting a high performance processor [17]. For each technology shrink, a smaller percentage of the chip is reachable within a clock cycle [13]. In particular, slow interconnects is the main reason for stalling a fast processor when waiting for cache accesses. Cache latency will continue to attack performance as long as on-chip cache requires multiple access cycles due to wire delay. In order to cope with on-chip wire delay, cache architectures are evolving from uniform cache accesses to a system that supports non-uniform access.

The UCA cache (uniform cache architecture) is the traditional cache architecture that required the same clock cycles for all L1, L2 or L3 cache accesses. Processors with this cache configuration are generally simple and have requires low data throughput when compared to high performance processors. Next, the ML-UCA (Multi-Level Uniform Cache Architecture) is the notion of having multiple levels of cache, where the smaller cache is a subset of the larger cache structures. The S-NUCA-1 (Static Non-Uniform Cache Architecture) is the first non-uniform architecture that is evaluated. This cache organization requires direct wiring to each of the banks where each bank is assigned a

specific latency for every bank access. The second non-uniform cache architecture (S-NUCA-2) introduces network characteristics in cache architectures. The S-NUCA-2 represents a grid of networked cache banks that use shared busses to transmit data. Finally, the D-NUCA (Dynamic Non-Uniform Cache Architecture) is an upgrade to the S-NUCA-2 where the most frequently used data are stored in the banks closest to the processing core.

Global wire delay is a circuit, architecture [HPCA2002] and more specifically a floorplanning issue. Floorplanning is the process of module placement and the creation/modification of a supporting interconnection network. Floorplanning is tedious and repetitious throughout the entire design life of a processor or in this case an on-chip cache architecture. Non-uniform cache architectures require floorplanning to place each cache bank and route the necessary wire interconnections to the processor.

The remainder of the document will discuss VLSI trends in wire properties in Chapter 2 concatenated with a brief overview of floorplanning in Chapter 3. Chapter 4 describes D-NUCA and implemented enhancements followed by a description of the simulation environment in Chapter 5. The results of a chip scaling, interconnect topology, and data migration study will be analyzed in Chapter 6 and finally the conclusion and future works will be discussed in Chapter 7.

2. VLSI Trends

Historically, on-chip interconnect wires were considered to be secondary with the exception of performing high-precision or critical path analysis. The introduction of deep sub-micron semiconductor technologies causes wire delay to take on a new perspective. The parasitic effects introduced by the wires display a scaling behavior that differs from transistors, and tend to gain importance as device dimensions are reduced and circuit speed increases. This situation is aggravated by the fact that improvements in technology make the production of ever-growing die sizes economically feasible, resulting in an increase in the average length of an interconnect wire and in the related parasitic effects. A careful and in depth analysis of the role and the behavior of the interconnect wire in a semiconductor technology is therefore not only desirable, but is necessary.

2.1 Wire Properties

Wires have three important electrical characteristics: resistance, capacitance, and inductance. Wire delay and noise behavior can be well modeled using these three characteristics. Each parameter depends on the wire's geometry and its position relative to other surrounding structures (other wires or substrate). This section briefly describes each wire property.

2.1.1 Wire Resistance

Resistance is the measurement of a wire's ability to carry an electrical current. Aluminum and copper are the two metal interconnect materials that are common to industry. Aluminum, an easily fabricatable and cheap wire material have a resistivity of

3.3 mΩ-cm, while thin-film copper wires have a resistivity of 2.2 mΩ-cm. The unit length resistance is simply calculated as the material's conductance (provided by manufacturer) multiplied by the conductor's cross-sectional area. Industry has experimented with copper interconnect and has discovered significant improvements. Associated with a copper wire is a thin insulation barrier layer on three sides during production to prevent copper/aluminum from diffusing into surrounding oxide in Figure 1. The thin barrier consumes portion of the dimension creating an effective cross-sectional area. The barrier thickness for 0.18-μm generation is 16nm [3]. Given the simple relationship between resistance and geometry, it is the easiest wire parameter to calculate; the following equation assumes a conformal barrier layer whose thickness on the sides equals that on the bottom:

$$R_{Wire} = \frac{\rho}{(thickness - barrier)(width - barrier)}$$

Equation 1. Resistance Calculation

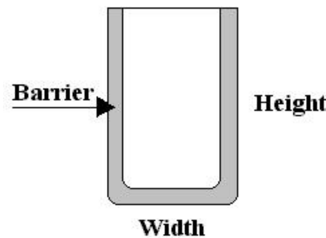

Figure 1. Cross-sectional Close-up of Copper Wire

2.1.2 Wire Capacitance

Capacitance is the amount of charge a wire can store. Although essential for digital logic, too much capacitance can also represent an undesirable time factor necessary for charge to be added or removed to change the electric potential on the wire. Many analytical

models approximate the capacitance of a wire over a plane; ones that are more accurate combine a bottom-plate term with a fringing term to account for field lines emerging from the edge and top of the wire. However, wires today are often taller than they are wide, and will grow even taller to reduce resistance as technologies scale. At minimum pitch, their horizontal capacitances are a significant and growing portion of the total. Capacitance is better modeled by a parallel-plate capacitor for each side of a wire, plus a constant term for fringing capacitance, as shown in Figure 2. [5]. The vertical and horizontal capacitors may have different relative dielectrics in technologies that use low-κ materials [6], this is discussed in more detail in the interconnect architecture section.

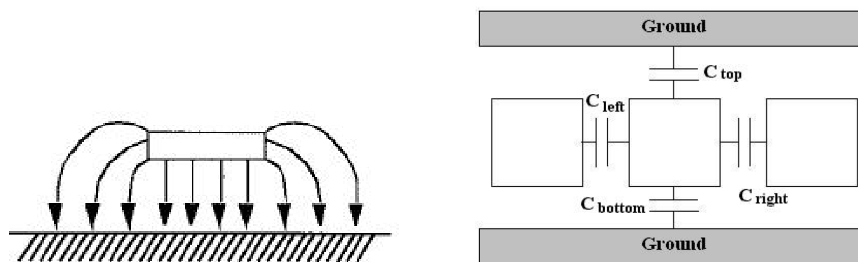

Figure 2. Capacitance Models

The plates for the top and bottom capacitors are typically modeled as being rounded, since they represent a collection of orthogonally routed conductors that when averaged over the length of the wire, maintain a constant voltage. The left and right capacitors have data dependent capacitances that can vary according to the neighbor's behavior. This effect, known as "Miller's Multiplication," can essentially half or double a signal's transmission given the circumstance in Fig 3.

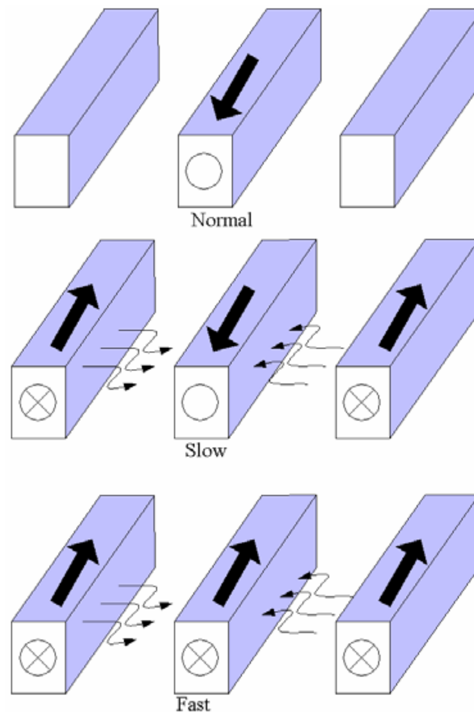

Figure 3. Miller's Multiplication

From the diagram, Miller's Multiplication can have a positive or negative effect on a signal depending upon the signal's timing and edge rate. The above case was simplified and assumed that all disturbances occurred simultaneously, causing immediate noise superposition to the main line. Because noise is time-independent, circuit designers use techniques such as *power shielding*, *swizzling* or *ground planes* to either minimize or fairly distribute crosstalk respectively.

2.1.3 Wire Inductance

Unlike resistance and capacitance, on-chip inductance cannot be modeled with a closed form equation. Determining the inductance is a complicated task mainly because of the unpredictability of the return path that completes the loop.

When current flows down an on-chip wire, the return currents may flow in nearby wires, parallel power supply buses, or even the substrate. Because return currents will flow in the paths of least impedance, the actual return paths may be determined by the frequency of the signal. For low frequencies, low resistance power buses, even if relatively far away, are low-impedance and return currents will use them, creating large loops and implying higher inductance. At high frequencies, long distant return paths have higher impedances, creating return currents through local, capacitively coupled wires, implying lower inductance. Modeling inductance can be a daunting task, but can be simplified with the following assumptions [Rabaey2001]:

- Inductive effects can be ignored if the resistance of the wire is substantial. This is the case for long Aluminum wires with a small cross-section or if the rise and fall times of the applied signals are slow.

- When the wires are short, the cross-section of the wire is large, or the interconnect material used has a low resistivity, a capacitance-only model can be used.

- Finally, when the separation between neighboring wires is large, or when the wires only run together for a short distance, inter-wire capacitance can be ignored, and all the parasitic capacitance can be modeled as capacitance to ground.

Interconnect architecture need to support larger and faster architectures. Given a faster clock rate, shrinking aspect ratios and a smaller minimum pitch, inductance is a growing concern that has circuit designer's making less of the above assumptions through out their simulations.

Wire resistance, capacitance and inductance are all wire properties that were once considered negligible but are now considered central to the future of processing. Wire and device scaling has reached a point where parasitic side effects become bottlenecks.

2.2 Wire Scaling Effect

The powerhouse behind the dramatic advancement of the integrated circuit technology over the past few decades has been the exponential scaling of the transistor. Transistor scaling followed Moore's Law at a reduction rate of 30% every three years and is expected to continue for at least another decade [?]. This will lead to over a billion transistors integrated on a single chip with an operating frequency of multiple GHz in the 65nm technology by the year 2007. With rapid feature size scaling, the circuit performance is increasingly determined by the wiring architecture rather than the devices as shown in Figure 5.

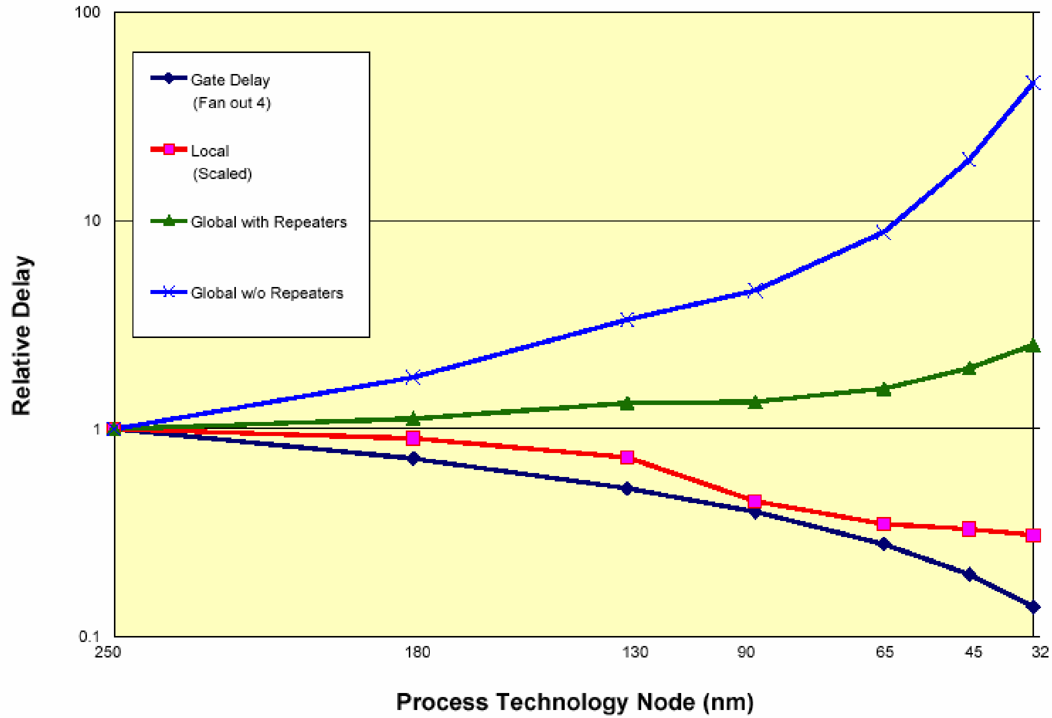

Figure 5. Delay for Local and Global Wiring vs. Feature Size [?]

Local, intermediate, and global wiring pitches are differentiated and highlighted as a hierarchical scaling methodology that has been broadly adopted. Implementation of copper and low κ materials allows scaling intermediate wiring levels and minimizes the impact on wiring delays. Local wiring levels are relatively unaffected by traditional scaling. RC delay, however, is dominated by global interconnect and the benefit of material changes (wire and dielectric) alone is insufficient to meet overall performance requirements.

2.2.1 On-chip Wiring Architectural Trend

Before we look at how technologies will scale, we will first look more closely at the current technology in Figure 6. to set some of the geometry assumptions that are used

when exploring scaled technologies. The assumed 0.18μm baseline technology has eight layers of aluminum interconnect, with upper layers wider and taller than lower ones. The lowest metal layers, M1 and M2, have the finest pitch and hence the highest resistivity, and it predominantly connects nets within gates or between relatively close gates. The middle layers, M3 and M4, have a wider pitch than M1 and M2 and connect routes within functional units. The next layers, M5 and M6, have the widest pitch and the lowest resistivity for carrying global routes. The final top layers, M7 and M8 are reserved for routing power and the system clock.

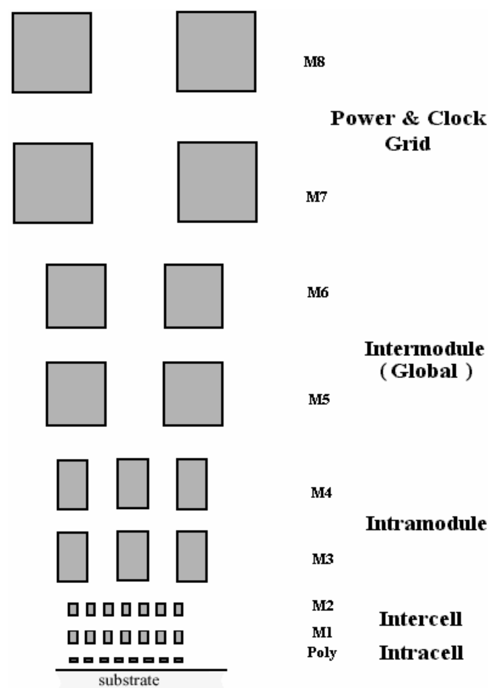

Figure 6. On-chip Wiring Architecture

Although not obvious from Figure 6, each metal layer is orthogonal (parallel and perpendicular in 3-D space) to its adjacent neighbor. Such a wiring architecture not only

provides uniform to routing but also reduces inter-layer noise (these are the capacitors tied to ground in Figure 2).

	Short-Term							Long-Term		
Year of Production	2001	2002	2003	2004	2005	2006	2007	2010	2013	2016
MPU/ASIC 0.5 Pitch (nm)	150	130	107	90	80	70	65	45	32	22
Number of metal levels	8	8	8	9	10	10	10	10	11	11
Number of optional levels--ground planes/capacitors	2	2	4	4	4	4	4	4	4	4
Local wiring pitch (nm)	350	295	245	210	185	170	150	**105**	**75**	**50**
Local Wiring A/R (for Cu wire)	1.6	1.6	1.6	1.7	1.7	1.7	1.7	**1.8**	**1.9**	**2**
Intermediate wiring pitch (nm)	450	380	320	265	240	215	195	135	**95**	**65**
Intermediate wiring dual Damascene A/R (Cu wire)	1.6	1.6	1.7	1.7	1.7	1.7	1.8	1.8	**1.9**	**2**
Minimum global wiring pitch (nm)	670	565	475	460	360	320	290	205	140	**100**
Global wiring dual Damascene A/R (Cu wire)	2	2	2.1	2.1	2.2	2.2	2.2	**2.3**	**2.4**	**2.5**
Inter-level metal insulator (minimum expected)-bulk dielectric constant (κ)	<2.7	<2.7	<2.7	<2.4	<2.4	<2.4	**<2.1**	**<1.9**	**<1.7**	**<1.6**
Manufacturable Solutions Known				Manufacturable Solution **Not** Known						

Table 1. International Technology Roadmap for Semiconductor: 2001 Interconnect Report

2.2.2 Wire/Dielectric Projection

A wire has three parameters that are directly scaled by the roadmap; height, width and the spacing. These are considered the fundamental parameters that composite measurements such as the pitch and aspect ratio. Wiring pitch is the lateral unit of a wire's width and the spacing allocated between adjacent wires. When designing critical paths, implementers can resort to modifying the size or ratio (width/spacing) of the wiring pitch to reduce crosstalk in order to meet timing. The aspect ratio (A/R) is a unitless measurement that represents the height of a wire with respect to the width (the height of a wire tends to be static and uniform within a layer). Obviously, the aspect ratio and wiring pitch are related and can be tweaked when necessary.

Interconnect architecture not only consist of wires but also dielectric material that acts as an insulator in Figure 7. Ideally, a dielectric functions as a barrier that fully confines the electric fields caused by current that is drawn within a wire. Unlike the wire that is measured in resistance, dielectric material are assigned a permitivity constant that measures the quantity of electrical noise that the material will allow.

2.2.3 Scaling the Wiring Layers

According to Table 1, current wiring architectures implement smaller pitches for local wires and gradually increase for global interconnects. It is well known that global interconnect delays are noticeable in current chip architectures and when scaled are expected to cultivate enormous parasitics that overshadow the subordinate routing layers[ITRS2001]. For this reason, global interconnects should be scaled less aggressively than local and intermediate wiring tiers. Surprisingly each wiring tier exhibits similar short and long term dimensional scaling of 86% and 70% respectively in Tables 2 and 3. Evidence of consistent near-term scaling warrants a design to violate the minimum pitch (local and intermediate tier) to facilitate complex chip architectures.

Near Term Scaling							
Year	2001-2	2002-3	2003-4	2004-5	2005-6	2006-7	
Scaling Dimension	150->130	130->107	107->90	90->80	80->70	70->65	**Average**
Local Tier	0.843	0.831	0.857	0.881	0.919	0.882	**0.866**
Intermediate Tier	0.844	0.842	0.828	0.906	0.896	0.907	**0.863**
Global Tier	0.843	0.841	0.968	0.783	0.889	0.906	**0.865**
Transistor	0.867	0.823	0.841	0.889	0.875	0.929	**0.871**

Table 2. Near Term Scaling [**ITRS2002**]

Long Term Scaling				
Year	2010	2013	2016	
Scaling Dimension	65->45	45->32	32->22	**Average**
Local Tier	0.700	0.714	0.667	**0.694**
Intermediate Tier	0.692	0.704	0.684	**0.693**
Global Tier	0.707	0.683	0.714	**0.701**
Transistor	0.692	0.711	0.688	**0.697**

Table 3. Long Term Scaling [**ITRS2002**]

Current wiring architectures have adopted the 8-layer system shown in Figure 6 and expect to grow to 15 layers with the optional ground planes/capacitors in Figure 7. The inclusion of extra layers is a great challenge for fabrication engineers and will necessitate the more aggressive long term scaling of 70%.

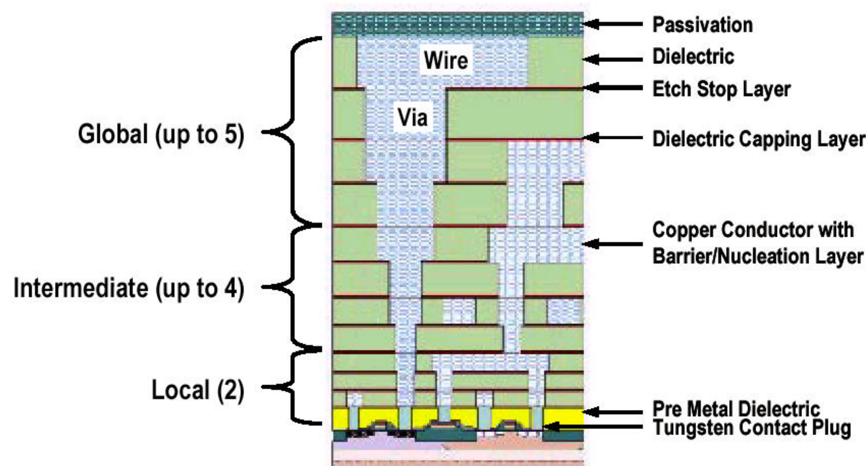

Figure 7. Cross Section of Hierarchical Scaling

2.2.4 Related Interconnect Research

Many research avenues were traversed in order to improve interconnect performance. Enhancing interconnect communication can occur at the device, circuit or architecture level. Although solutions at each level improve upon the communication delay, it’s apparent that collaboration is needed to sustain the metal/dielectric era to its maximum longevity.

Various interconnect innovations are currently under research that attempts to improve global communication. A new approach utilizes electromagnetic transmission through free space that opens the doorway to communication at the speed of light. First, optical interconnects has many advantages, but also has several clear areas requiring significant research. This research mainly explores which signals to include in optical communications and those that remain in conventional metal dielectric. It will be necessary to fabricate highly responsive emitters and detectors as well as the ability to overcome electromagnetic coupling.

Boosters are circuits that accelerate global signals faster than repeaters. A booster is a circuit such that it detects a transition early on a wire and then speeds up the transition. When compared to optimally placed repeaters, boosters have improved performance, reduced circuit complexity and are better suited for driving global wires. Boosters are also able to drive bi-directional signals, rather than interrupt the flow of the signal the booster operates in parallel with the long wire. This short-term application may allow engineers to obey Moore's Law for the next few generations.

Optical computing is often spoken of as the next leap in computing technology. The main principle involves the recognition that the cross-section of a light beam represents a two-dimensional array of data. In ordinary optics, such arrays usually correspond to images but, in considering their computational significance, this need not necessarily be so. The

main components of optical computers are those familiar to us from ordinary optical systems but used in a novel fashion:

- **Lasers** are used both because of the coherence of their light and its relative concentration. They are effectively the power supplies of the system.
- **Lenses** distribute or concentrate the light beams as required
- **Beam splitters** permit the effective duplication of the light beam.
- **Shutters** control activity in the system.
- **CCD arrays** convert images representing output data into electronic signals used for either control feedback or interfacing.
- **Spatial light modulators** encode information into spatially distributed optical patterns.
- **Holographic plates** by means of which data can be stored and retrieved in the form of holographic patterns.

"It has been shown by several researchers that the global interconnect performance needed for future generations of ICs cannot be achieved even with the most optimistic values of metal resistivity and dielectric constant … the most promising near-term solution (available in the next few years) anticipated for the global interconnect problem is design modifications to reduce the length of metal lines, combined with advances in Cu-low κ metallization." [**ITRS 12/2001**]

Chapter 3

Floorplanning On-chip Memory Elements

Floorplanning is the arrangement and connection of circuit modules on the physical chip. Some examples of circuit modules are memory controllers, floating point execution units and cache banks. Floorplanning is the interface between architects and circuit designers, it is considered to be low-level by architects and high-level to circuit designers. Architect use floorplanning early in a chip's development when determining the processor's specifications and is used throughout the development of a chip by circuit designers. Floorplanning is an important, iterative and time-consuming process.

Many objectives are achievable through a good floorplan. Some objectives include:

- Reduce the wire length
- Reduce the chip area
- Minimize the power consumption
- Optimize I/O requirements

Floorplanning is becoming more important as wire delay dominates microprocessor performance. When scaling is applied to a design the wire delay and the chip area increases. The growing area allows architects to pack more hardware onto a chip. This leads to an increase in the wires needed to allow the multitude of hardware to communicate. Floorplanning is crucial to a competitive design that consists of small geometric modules with high communication costs. A processor design is considered

competitive when it is fast, low power, and consumes very little area or any combination of the three.

Floorplanning is an iterative process that is necessary for the slightest change to a chip's architecture. Some changes that can warrant a new floorplan are:

- Adjustments to the wiring pitch
- Modifications to the cache size
- New technology
- Clock rate adjustments
- Power rails adjustments

Changing the wiring pitch can either accommodate or hinder communication by altering the number of wires on a given metal layer. Such an adjustment can force a new floorplan to dedicate more or less space for wire routing between modules.

Caches are known for consuming a great portion of the chip's area and this is expected to continue. For example, the HP PA-8700 and the Intel Itanium 2 contain 2.25MB and 3MB of on-chip cache. The memory bottleneck is an interconnect problem and is expected to be the key issue for many years to come [**HPCA2002**]. Floorplans are becoming more interconnect-centric and dependent on the spacious on-chip data structures. (cont)

Chapter 4.

Dynamic Non-Uniform Cache Architecture Components

Prior architecture research introduced multi-ported, banked and pipelined caches to overcome the penalty of long cache accesses, but each approach has its own drawback. Although multi-ported cells can satisfy more requests simultaneously, the extra logic increases the chip area and timing delay per bit and the benefits quickly diminishes when more than three or more ports is supported. Banked caches allowed cache accesses to overlap but this organization is susceptible to bank conflicts when enough addresses reference the same bank. Despite the ability to pipeline cache requests, cache latencies greater than 2 or 3 cycles have proven to negatively affect performance [18].

More recent work shows performance improvement in cache latency as cache designs progress from uniform caches to non-uniform caches in Figure 1 [1].

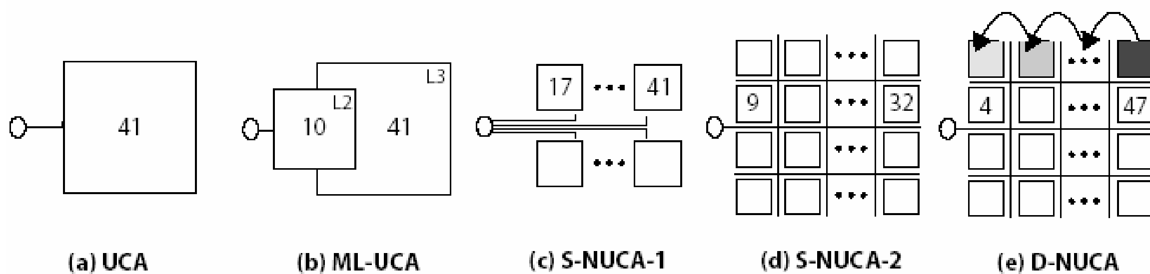

Figure 8. Cache Organizations [1]

The UCA cache (uniform cache architecture) is the traditional cache architecture that required the same clock cycles for all cache accesses. The ML-UCA (Multi-Level

Uniform Cache Architecture) is the notion of having multiple levels of cache, where the smaller cache is a subset of the larger cache structures. The S-NUCA-1 (Static Non-Uniform Cache Architecture) is the first non-uniform architecture that is evaluated. This cache organization requires direct wiring to each of the banks where each bank is assigned a specific latency for every bank access. The second non-uniform cache architecture (S-NUCA-2) introduces network characteristics in cache architectures. The S-NUCA-2 represents a grid of networked cache banks that use shared busses to transmit data. Finally the D-NUCA (Dynamic Non-Uniform Cache Architecture) is an upgrade to the S-NUCA-2 where the most recently used data are stored in the banks closest to the processing core.

There are a variety of cache organizations worthy of exploration but this research focuses on the performance comparison of a S-NUCA2 and D-NUCA cache on an Alpha 21364. The D-NUCA cache will also explore a torus, mesh and hypercube interconnect topology and compare their impact on microprocessor performance.

Unlike the S-NUCA2, the D-NUCA cache organization introduces the ability to store data in cache banks physically closest to the processing core. This chapter describes the main intelligence components of the D-NUCA [1] and the minor architectural changes made to adapt to the Alpha 21364. The D-NUCA cache consists of four major components: Mapping, Search, Promotion and Insertion.

4.1 Data Mapping

The D-NUCA cache explored various cache bank-mapping schemes that relied on tag organizations and the number of associative ways. There are three methods of allocating banks to bank sets and ways: *simple mapping, fair mapping* and *shared mapping* [1]. The simple mapping requires that each column of banks in the cache becomes a bank set, and all banks within that column comprise a set-associative way. Therefore, the cache is searched for a line by first selecting the bank column, and perform a tag match on banks within that column of the cache. The two drawbacks of this scheme are that the number of rows may not correspond to the number of desired ways in each bank set, and that latencies to access all bank sets are not the same; some bank sets will be faster than others, since some rows are closer to the cache controller than others.

The *fair mapping* scheme attempts to equalize the average access latency by distributing the banks fairly on the chip. But unlike the simple mapping technique, the complex wire routing leaves little to no room for supporting interconnection topologies used in this research, such as the torus, mesh and hypercube.

The *shared mapping* policy, shown in Figure 9c, attempts to provide the fastest bank access to all bank sets by sharing the closest banks among multiple sets. This policy requires that if, bank sets share a single bank, then all banks in the cache must support *n*-ways. Furthermore, Figure 9c illustrates that each of the farthest bank sets shares half of the closest bank for one of the closest bank sets. This policy results in some bank sets having a slightly higher bank associativity than the others, which can offset the slightly

increased average access latency to that bank set. That strategy is illustrated in Figure 9c, in which the bottom bank of column 3 caches lines from columns 1 and 3, the bottom bank of column 4 caches lines from columns 2 and 4, and so on. In this example the farthest four (1, 2, 7, and 8) of the eight bank sets share the closest banks of the closest four (3, 4, 5, and 6).

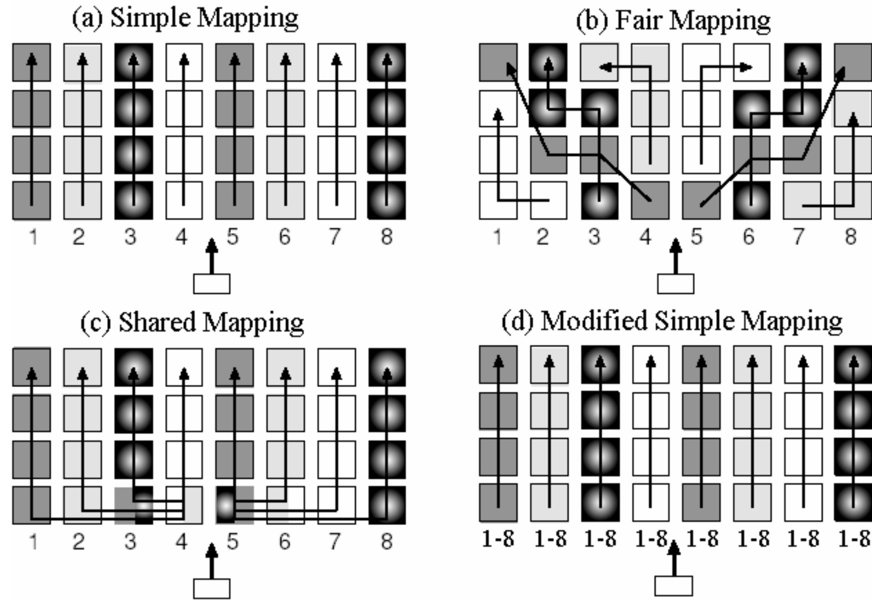

Figure 9. Mapping Schemes (a,b,c [1])

For this research work, the fair and shared mapping policy were rejected due to the irregular wiring and bank organization, respectively. This research work uses a modified simple mapping scheme to allocate banks to a search path (or bank grouping). The modified simple mapping scheme organizes the banks into a continuous range of set numbers rather than by associative ways. Unlike the simple mapping policy, each bank contains all n-ways of the set. Therefore a modified simple approach can assign set 0 through set 8 to a bank versus the same bank holding only set 0, 4, 8 in the original simple mapping scheme. Therefore a full set is accessible within a bank. Because the

simple mapping scheme requires little wiring overhead, there are a number of vacant wiring levels to support a torus, mesh or hypercube interconnect topology.

4.2 Bank Search

There are a couple of search policies for determining the location of a cache block. The two policies [1] are the incremental and broadcast search. In Figure 10a, the incremental search checks a bank for the data block and must return false before the next bank is checked within a search path. Therefore, bank 0 then bank 1 and so forth are accessed exclusively. Once the last bank (in this case, bank 3) returns false then a cache miss occur. This *incremental search* policy can be time consuming, especially for cache misses. Therefore the *broadcast search* policy broadcasts the data to the entire search, essentially searching every cache bank simultaneously. This approach is network intensive and can potentially block read hits from being read in a timely manner. But despite the network intensity, the broadcast policy serves as the best policy because the network congestions are expected to arrive in burst. Although each bank in the search path will be busy simultaneously, the cache banks will be available after time x, where x is the number cycles required to decode a cache block within a bank.

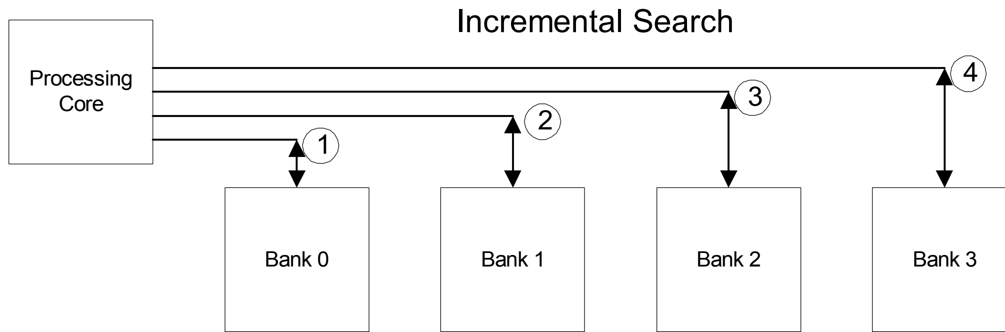

Figure 10a. Incremental Bank Search

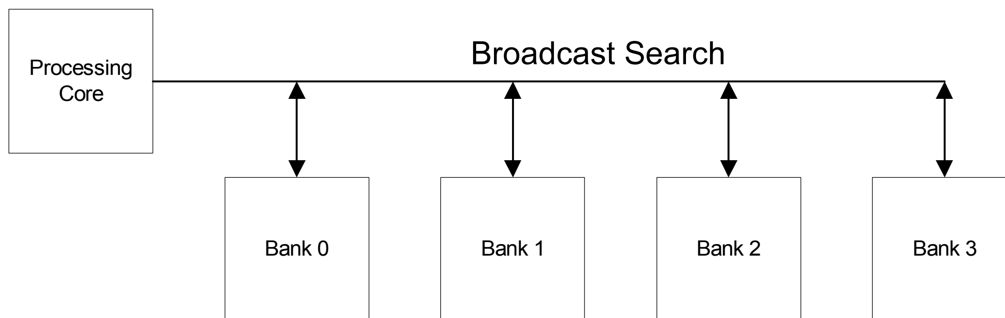

Figure 10b. Broadcast Bank Search

4.3 Data Promotion Scheme

The original D-NUCA cache promotes data incrementally as shown in Figure 11a. As shown, a data request to a very far bank triggers a promotion. In this case, once the data is read out of the cache bank, the data is then sent to the processor core and the next closest bank in the search path. If the next closest bank is full then a cache set must be demoted to the next farthest bank. Their goal is to minimize the global traffic when data is swapped between two banks. This current research work assumes an LRU policy that promotes data to the closest bank as the data travel to the processing core in Figure 5b. This is synonymous to the snooping protocols used in multiprocessor systems [8]. In order to offset the network congestion that can occur during the demotion process, the

research assumes dual-ported cells, a single port for reading and for writing data. This will allow reads and writes to occur simultaneously.

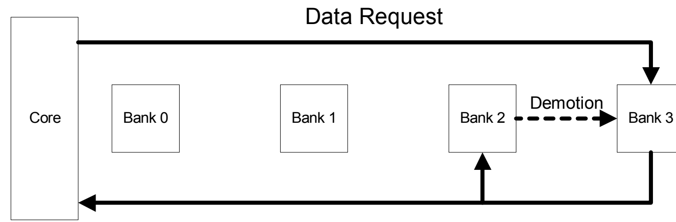

Figure 11a. Incremental Promotion

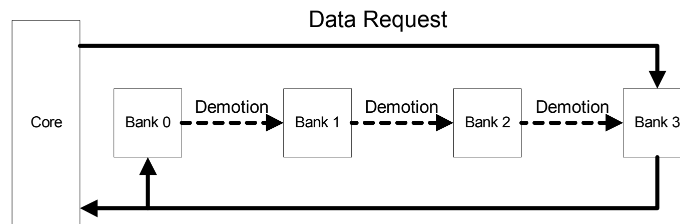

Figure 11b. Absolute Promotion

Chapter 5

Research Environment

The purpose of this research is to model dynamic non-uniform cache architectures with wire delay modeling and simulate the effect that various wiring schemes have on throughput. The research environment consists of four major components: **Floorplan Generation**, **Bank Timing**, **Generate Wire Latency** and **Architectural Simulations**. **Floorplan Generation** is the phase of the research that scales and populates a floorplan with cache banks. The **Bank Timing** phase uses Cacti[16] to evaluate the access time of a cache bank. The **Generate Wire Latency** converts wire's length into a wire's latency. Finally, the **Architectural Simulation** uses Simplescalar and SPEC2000 to simulate the new floorplan and generate performance statistics.

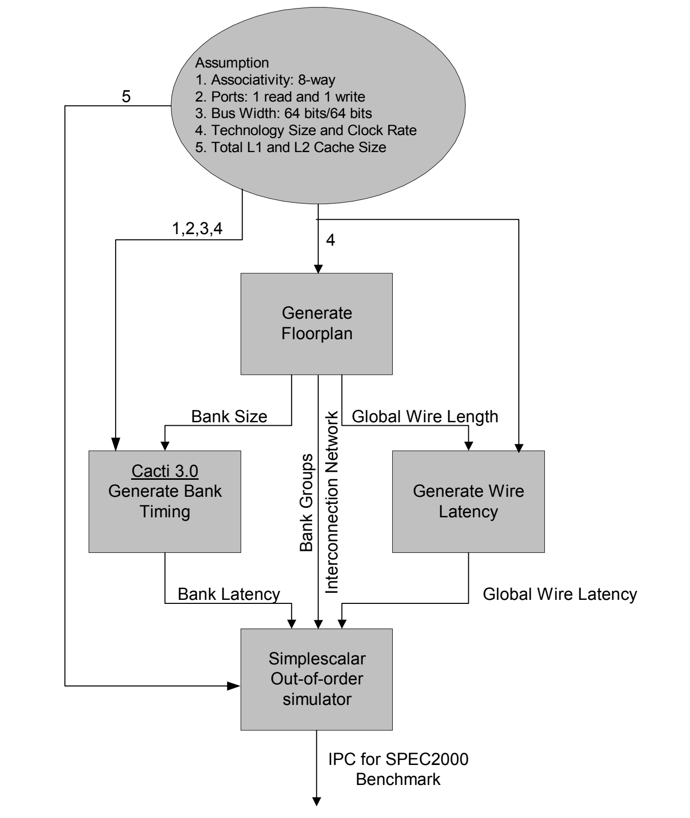

Figure 12. Simulation Flowchart

5.1 Generating the Floorplan

The purpose of Figure 13 is to establish the experimental flow for determining the effect of a wire-dominated cache. The research begins with a floorplan of the Alpha 21364, which is a 21264 core that is supplemented with 1.75MB of L2 cache and multiprocessor support. The Alpha 21364 consumes 4cm^2 of chip area for 0.18um technology. This floorplan serves as the basis for applying scaling techniques for projecting floorplans in smaller technologies. The following parameters were incorporated in each floorplan:

Parameter	Notes
Technology Size	Scales according to the ITRS
Cache Size	Maximum that can fit on-chip
# of Read and Write Ports	One of each port for all technologies
Associativity	8-way
Bus Width	64 bits

Chip Area	10% area increase per technology

Table 4. Cache and Chip Layout Assumptions

5.2 Generating the Bank Timing

In order to generate the cache bank timing, the research uses the Cacti 3.0 tool [16]. Cacti 3.0 is an integrated cache access time, cycle time, aspect ratio and power model. Given the above assumptions made in Table 4, Cacti 3.0 is fed the cache size, number of ports, cache associativity and the bus width. Cacti 3.0 uses these parameters to generate the time (ns) required to access a cache block within a cache bank. **(a paragraph about Cacti evaluation technique)**

Cacti Parameters	Value
Number of Subbanks	1
Total Cache Size	0.25MB
Size of Subbank	0.25MB
Number of Sets	512
Associativity	8-way
Block Size	64-bytes
Read Ports	1
Write Ports	1
Power (Vdd)	1.3V
Access Time (ns)	**2.51115**
Access Time (cycles)	**5.27**

Table 5. Cacti Parameters for 130nm floorplan

5.3 Global Wire Delay Table

The global wire delay table carries the task of converting the wire lengths extracted from a floorplan into communication delay (in cycles). The table was generated using SPICE and the ITRS 2002 (International Technology Roadmap for Semiconductor) projections [2]. The ITRS projection includes the wire width, height, spacing and the dielectric permittivity. For each wire length entry, optimally placed repeaters were inserted, using the Bakoglu's method to decrease the communication delay [19]. This is common practice in the industry. The table below graphically represents only pure wire delay and does not consider the latch delay. The latch delay is considered when converting wire delay to wire latency. Each latency conversion assumes 10% of a clock cycle to be added to the wire delay. Therefore a wire length of 8mm in 90nm technology would translate into 1.98 cycles, but with the inclusion of the latch delay, the total delay in cycle would rise above 2.00 cycles. Applying the ceiling function would finally bring the latency to 3 cycles, because microprocessors are not partial-cycle driven.

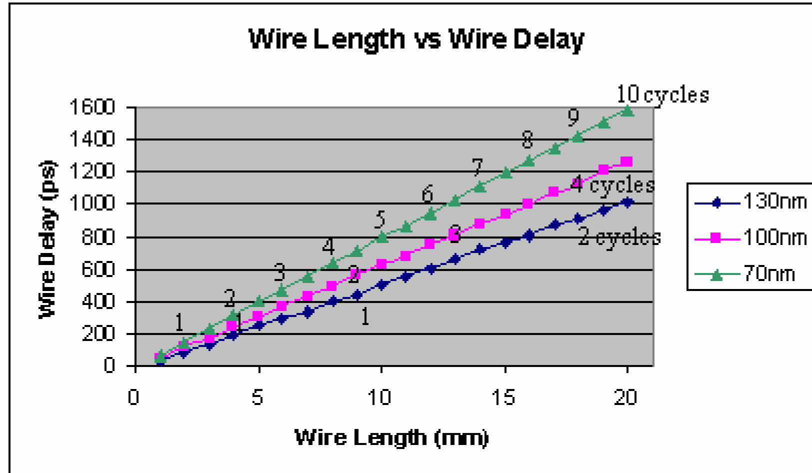

Figure 13. Global Wire Delay

This time delay plus the communication delay are entered into an extended version of Simplescalar that supports non-uniform caches and the global routing delay to and from the ports of a cache bank. Simplescalar then executes a set of benchmarks using the new parameters to generate performance statistics later shown in Table 2 and 3.

5.4 Simplescalar Extended

Simplescalar is an architecture simulator that will model the Alpha 21364. A few modifications were made to Simplescalar to also model a non-uniform cache system with wire delay support. In the extended version of Simplescalar, the user is capable of specifying the quantity, size and the optimal transmission delay for each cache banks. The examples below are the necessary parameters to simulate the cache system previously shown in Figure 3.

-cache:latency_array0:255:6:511:7:767:8:1023:9:1279:6:1535:7:1791:8:2047:9:

2303:6:2559:7:2815:8:3071:9:3327:6:3583:7:3839:8:4095:9

In the above example, the option specifies 4096 sets that are partitioned into 16 sub-banks. The first bank can hold up to 256 sets with a starting address of 0x000 and requires an access latency of 1 cycle, the next bank can also hold 256 sets but has a starting set address 256 (or 0x100) with an access latency of 2 cycles.

-cache:search_path {Bank15, Bank14, Bank13, Bank12: Bank11, Bank10, Bank9, Bank8: Bank7, Bank6, Bank5, Bank4: Bank3, Bank2, Bank1, Bank0}

-cache:alt_search_path {Bank3, Bank7, Bank11, Bank15: Bank2, Bank6, Bank10, Bank14: Bank1, Bank5, Bank9, Bank13: Bank0, Bank4, Bank8, Bank12}

The above two parameters define the search path and alternate search when rerouting around a busy node is necessary. For the configuration above, there are 4 defined search paths. When a cache read is required, a preliminary tag comparison determines which path to search. In the event that a cache hit occurs and the data collides with a busy node then an alternate route is chosen to for the data to travel. The only constraint is that an alternate route can only exist between neighboring cache banks.

Chapter 6

D-NUCA2 Performance Analysis

The section compares the performance of a S-NUCA to a D-NUCA2 system. The research examines scaling study for 130nm, 90nm, and 65nm. The second section conducts a topology study of a D-NUCA2 system in 90nm that support a torus, mesh and a hypercube finally a study of the promotion scheme.

6.1 Technology Study

The technology trend uses a 21364 floorplan as the basis for comparing a S-NUCA to a D-NUCA2. The Alpha 21364 is a model of the Alpha 21264 with large on-chip L2 caches and multiprocessor support. The data in Table 15 correlates two separate cache systems. The first cache system is a traditional static non-uniform cache architecture where data always exist in a specific physical area on the chip. The second cache system is a dynamic non-uniform cache architecture where popular data is allowed to migrate to physically closer cache banks. The results of Table 15 demonstrate that the IPC generated for each benchmark was unaffected much by communication delay and that pipelined cache accesses could easily hide the wire delay overhead. These results were somewhat expected since global delay is around a cycle for most on-chip point-to-point transmissions. The average IPC improvement was a miniscule 0.25% despite noticeable miss rates for both cache organizations. This implies that very little migration occurs from far banks to closer banks.

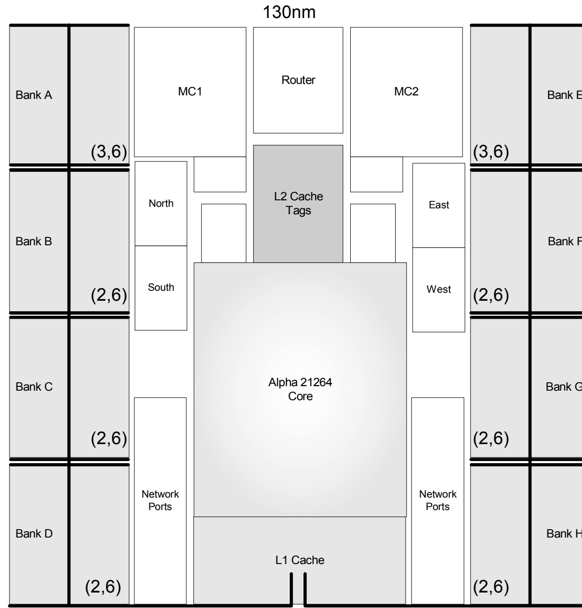

Figure 14. Alpha 21364 (130nm): L2 Cache 2MB

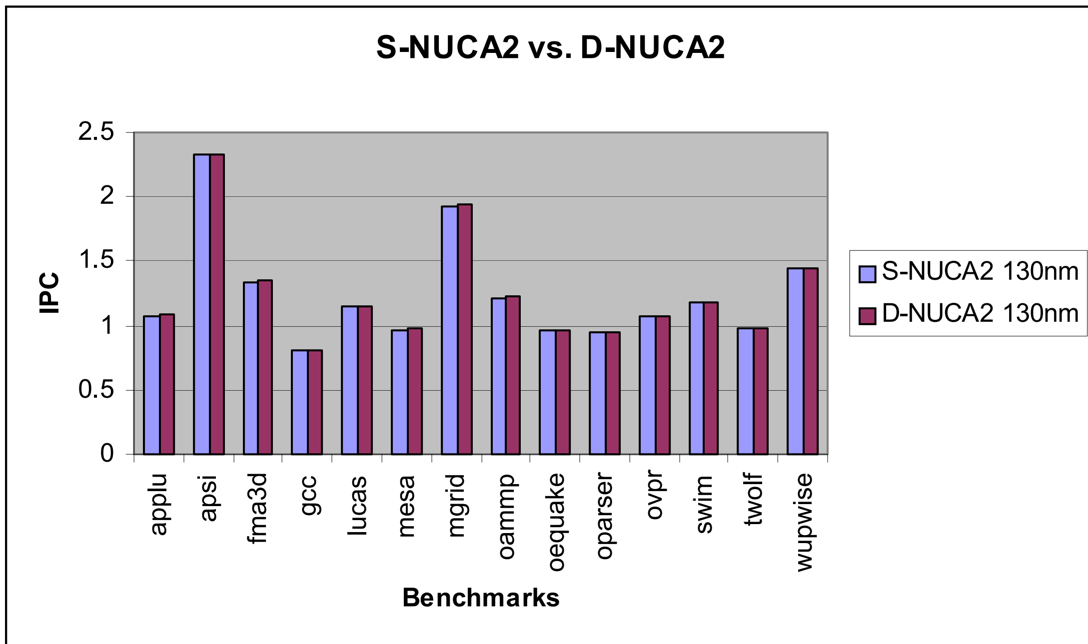

Figure 15. D-NUCA2 Speedup over S-NUCA2 for 130nm: L2 Cache 2MB

The Alpha floorplan (90nm) in Figure 16 is a 21364 with 8MB of L2 cache. Figure 16 shows a considerable smaller processor core that is under 50% of the original core. The

90nm processor core also consumes a smaller percentage of the chip area because of the growing chip area per process generation [2]. The unused area of the chip is filled with 0.125MB cache banks. The cache bank size was reduced to make more room for supporting hardware when scaling from 130nm to a 90nm process.

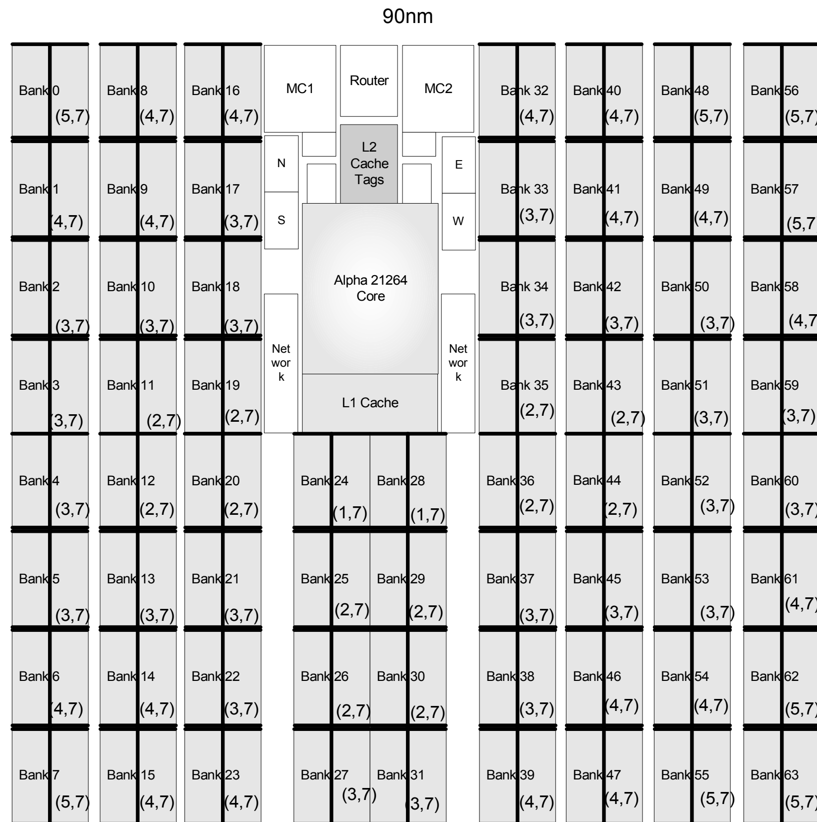

Figure 16. Alpha 21364 Floorplan 90nm: L2 Cache 8MB

The 90nm version of the Alpha 21364 has 8MB of 64 banks each contain 0.125MB memory modules. The banks are organized into groups of 8 banks creating cubic nodes across the chip. The groups are further broken down into two subgroups of four banks. The two subgroups are restricted from exchanging data but are interconnected for routing purposes in Figure 17.

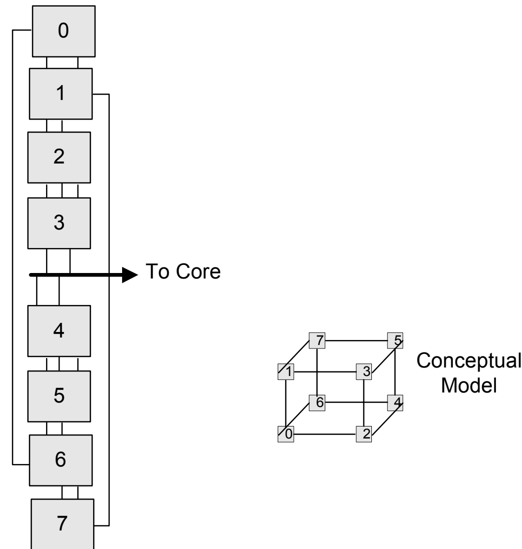

Figure 17. Hypercube Interconnection Scheme

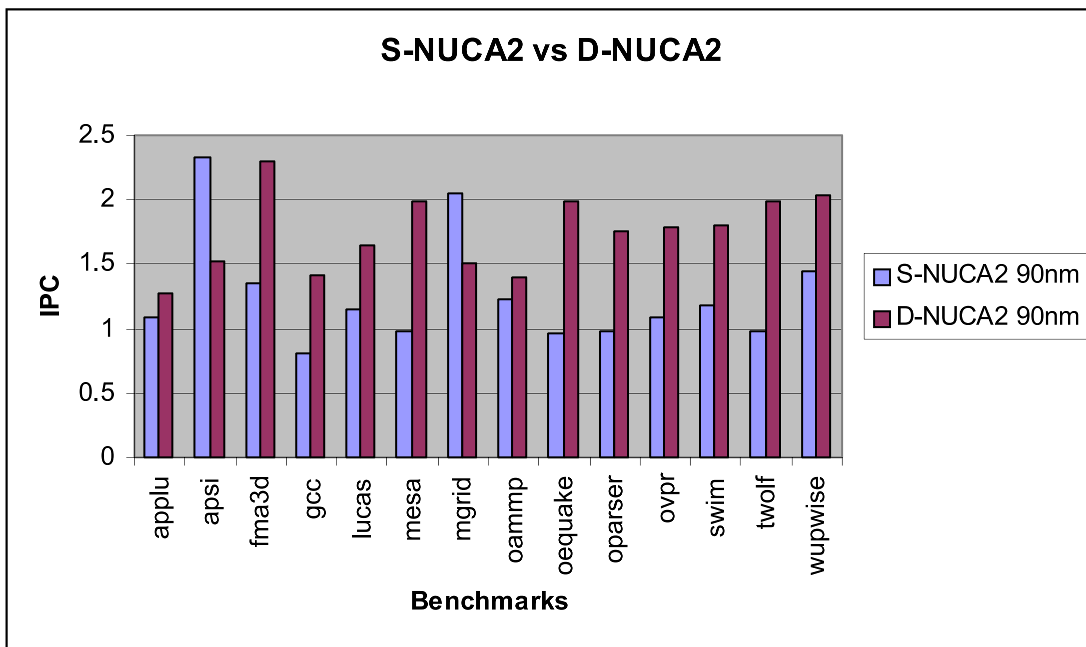

Figure 18. D-NUCA2 Speedup over S-NUCA2 for 90nm: L2 Cache 2MB

The average IPC improvement showed a 43% improvement across the benchmarks with the exception of two benchmarks. This is characteristic of excessive collision between far read accesses and data demotions from closer to farther banks. This implies that the data set is

small enough to fit inside the L2 cache banks. The low D-NUCA2 average miss rate from the 130nm to the 90nm floorplan also confirms this. The technology study in Figure 10 correlates the throughput of an S-NUCA2 and D-NUCA2 for some benchmarks in SPEC2000. The study explores the IPC for 2MB, 8MB, and 16MB with a corresponding floorplan in 130nm, 90nm and 65nm. Each of the benchmarks shows a significant improvement for D-NUCA2 systems. Surprisingly for most benchmarks, the D-NUCA2 for 90nm outperforms the S-NUCA for a 65nm.

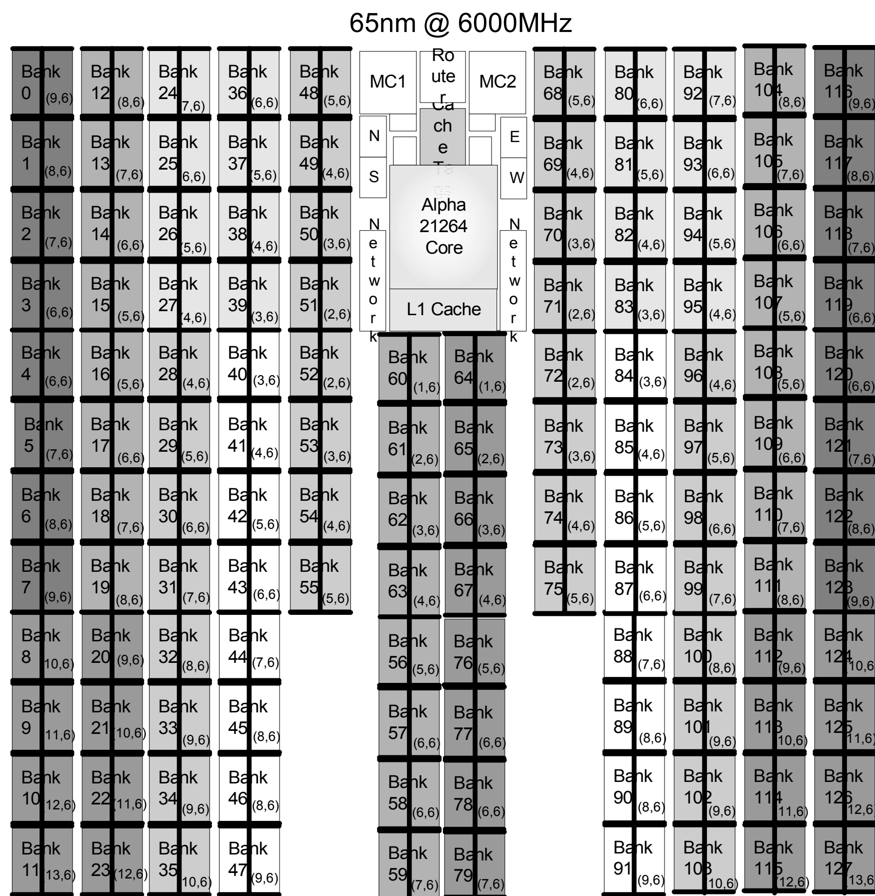

Figure 19. Alpha 21364 Floorplan 65nm: L2 Cache 16MB

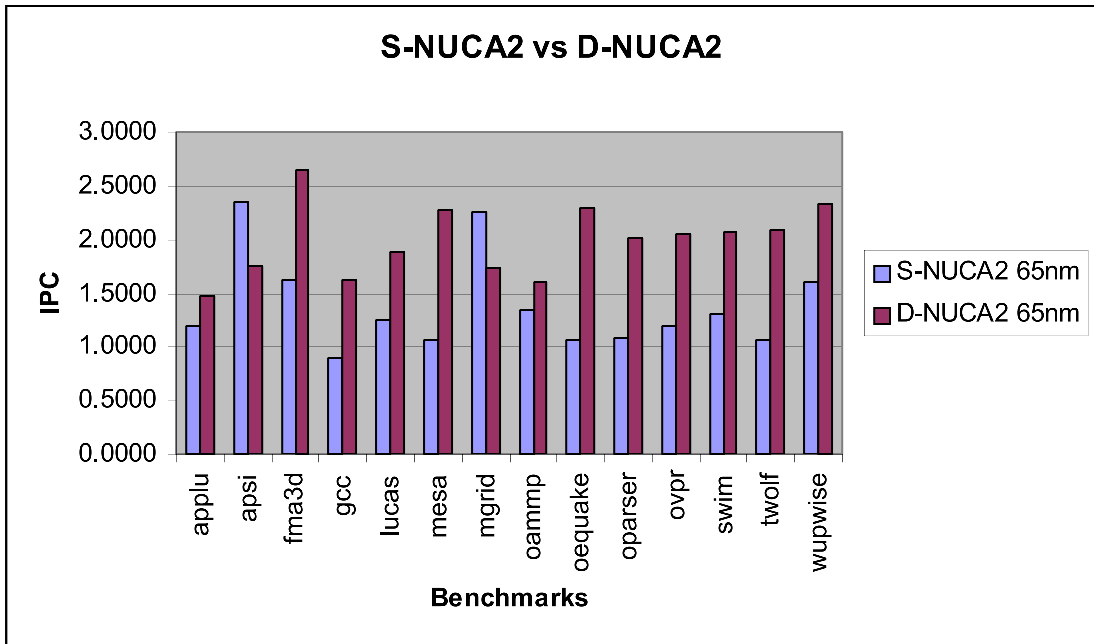

Figure 20. D-NUCA2 Speedup over S-NUCA2 for 65nm: L2 Cache 16MB

(Insert paragraph discussing the possible performance limiter)

In general, D-NUCA2 shows significant improvement over a S-NUCA for large cache systems. But the D-NUCA2 shows some limitations on performance when technology migrates to 65nm. At 65nm, where the simulated cache size is 16MB, the cache system is broken down into 128 cache banks. At this stage, bank contention becomes an issue and prevents data from taking the shortest path to the processor. Research results show that this occurs frequently in 65nm technology and on occasion for *apsi* and *mgrid* in 90nm where bank contention negatively affects performance.

(Talk about the algorithmic nature of the above mentioned benchmarks)

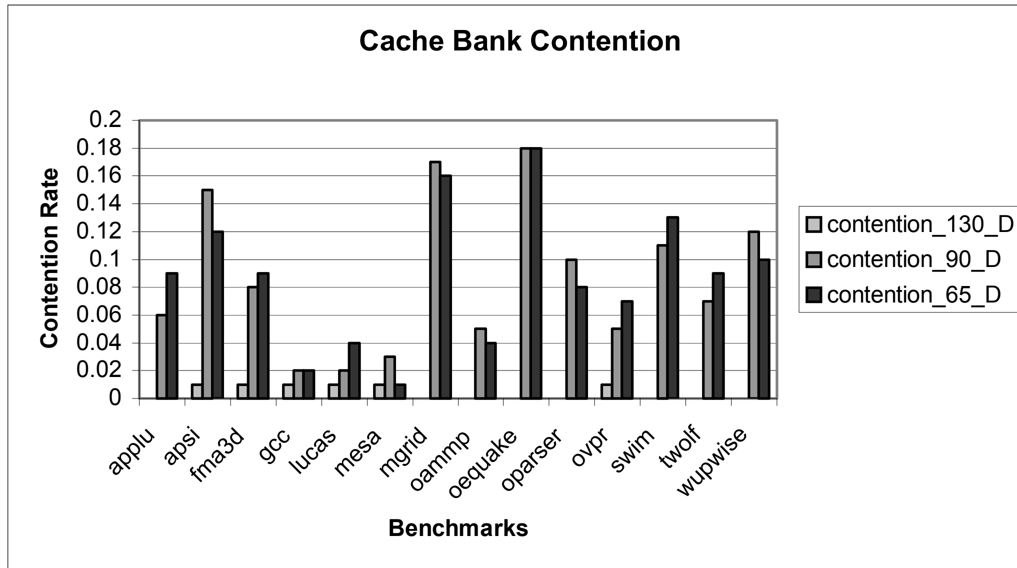

Figure 21. Cache Bank Contention for D-NUCA2

6.2 Bank Interconnection Topology Study

The topology study shows the performance trend for a D-NUCA system connected in a torus, mesh and hypercube network. The number of available wiring layers and the possibility of reducing the average latency motivated the idea of this topology study. As expected each of the benchmark showed an improvement as the interconnection network complexity increased. Because of the increased wiring complexity, data was less likely to stall because of a flexible network that is very capable of rerouting the data to the processor. For this reason the torus performs poorly. Given a single node, data can only travel to another single node. In a mesh and hypercube configuration, a piece of data has the option of one or two other nodes for rerouting, respectively. The hypercube outperforms the mesh network by providing reroutes with fewer hops. This translates into a smaller average latency in Figure 17. In Figure 18, the three interconnection topologies are

examined from a single node. As shown, node 0 is the reference node and it is immediately shown that as the complexity of the interconnect topology increase the number of available nodes for transferring data also increase. From node 0, the torus topology has a single node (Node 1) that can route data to the processing core while the mesh topology can transmit requested data to either node 1 or node 4. Therefore in the event that a node 1 is busy servicing a bank search, then the data can be routed to node 4.

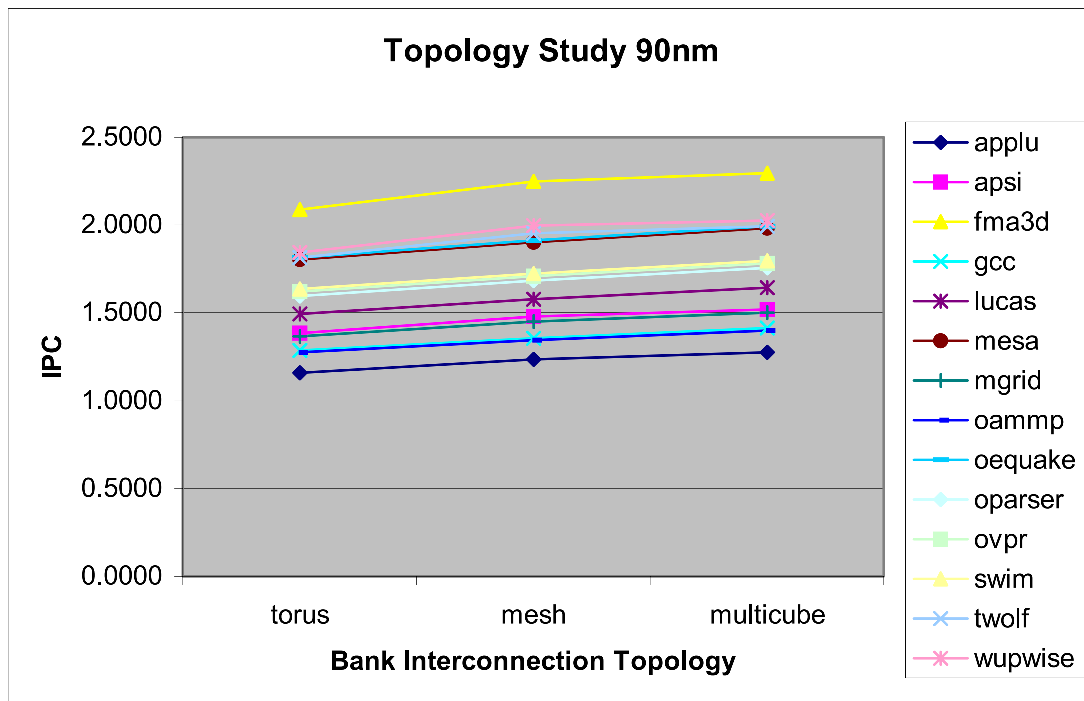

Figure 22. Bank Interconnection Topology Study

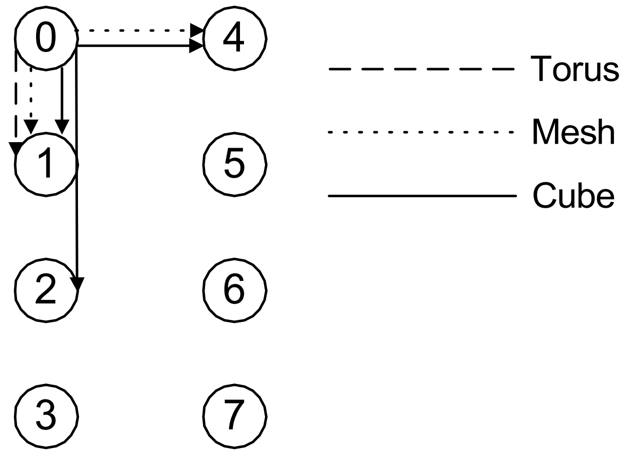

Figure 23. Interconnect Topology Routing Comparisons

6.3 Data Migration Study
(Insert Performance Diagrams)

6.4 Chapter Summary

7. Conclusions and Future Work

7. References (Must update numbers and add the remaining references from previous proposal)

[1] C. Kim, D. Burger, S. Keckler, "An Adaptive Non-Uniform Cache Structure for Wire-Delay Dominated On-Chip Caches", ASPLOS, October 2002. San Jose, California.

[2] International Technology Roadmap for Semiconductors 2002 Update. December 2002. http://public.itrs.net/Files/2002Update/2002Update.pdf

[3] S. Hamilton, "Taking Moore's Law into the next Century", Computer, January 1999, pp. 43-8.

[4] D. Sylvester, K. Keutzer, "Getting to the Bottom of Deep Submicron", IEEE Computer, 1999.

[5] T. Pinkston, Y. Patt, B. Dally, B. Horst, A. Agarwal, Jose Duato, T. Basil "What will have the greatest impact in 2010: The processor, the memory or the interconnect?", IEEE MICRO 2002. http://www.usc.edu/dept/ceng/pinkston/presentations/statistics.html

[6] D. Sima, T. Fountain, P. Kacsuk, "Advanced Computer Architectures: A Design Space Approach", Addison Wesley, 1997

[7] J.L. Hennessy and D.A Patterson, Computer Archtiecture: A Quantitative Approach: 3rd Edition, Morgan Kaufmann, San Francisco, VA, 2002.

[8] D. Sima, T. Fountain, P. Kacsuk, "Advanced Computer Architectures: A Design Space Approach", Reading, MA: Addison-Wesley, 1990.

[9] A. Kleinosowski, J. Flynn, N. Meares, D. Lilja, "Adapting the SPEC benchmark suite for simulation based computer architecture research" WWC-3 pp. 73-82, 2000.

[10] S. Mukherjee, P. Bannon, S. Lang, A. Spink, D. Webb, "The Alpha 21364 Network Architecture"

[11] T. Austin, "A User's and Hacker's Guide to the SimpleScalar Architectural Research Tool Set," 1997.

[12] R. Kessler, "The Alpha Microprocessor: Out-of-Order Execution at 600MHz," Compaq Computer Corporation, August 1998.

[13] D. Matzke, "Will Physical Scalability Sabotage Performance Gains?", Computer, Sept. 1997, pp. 37-40.

[14] V. Agarwal, M. S. Hrishikesh, S.W. Keckler, and D. Burger, "Clock rate vs. IPC: The end of the road for conventional microprocessors" In Proceedings of the 27th Annual International Symposium on Computer Architecture, pages 248–259, June 2000.

[15] M. Horowitz, R. Ho, and K. Mai. The future of wires. In Seminconductor Research Corporation Workshop on Interconnects for Systems on a Chip, May 1999.

[16] P. Shivakumar and N.P. Jouppi. Cacti 3.0: An integrated cache timing, power and area model. Technical report, Compaq Computer Corporation, August 2001

[17] R. Kessler, "Analysis of Multi-Megabyte Secondary CPU Cache Memories". PhD thesis, University of Wisconsin Madison, December 1989.

[18] K. Wilson and K. Olukotun, "Designing High Bandwidth On-Chip Caches," In Proceedings of the 27th Annual International Symposium of Computer Architecture, June 1997.
[19] H. Bakoglu, "Circuits, Interconnections and Packaging for VLSI", Reading, MA: Addison-Wesley, 1990.