TLC: Transmission Line Caches

Bradford M. Beckmann and David A. Wood
Computer Sciences Department
University of Wisconsin—Madison
{beckmann, david}@cs.wisc.edu

Abstract

It is widely accepted that the disproportionate scaling of transistor and conventional on-chip interconnect performance presents a major barrier to future high performance systems. Previous research has focused on wire-centric designs that use parallelism, locality, and on-chip wiring bandwidth to compensate for long wire latency.

An alternative approach to this problem is to exploit newly-emerging on-chip transmission line technology to reduce communication latency. Compared to conventional RC wires, transmission lines can reduce delay by up to a factor of 30 for global wires, while eliminating the need for repeaters. However, this latency reduction comes at the cost of a comparable reduction in bandwidth.

In this paper, we investigate using transmission lines to access large level-2 on-chip caches. We propose a family of Transmission Line Cache (TLC) designs that represent different points in the latency/bandwidth spectrum. Compared to the recently-proposed Dynamic Non-Uniform Cache Architecture (DNUCA) design, the base TLC design reduces the required cache area by 18% and reduces the interconnection network's dynamic power consumption by an average of 61%. The optimized TLC designs attain similar performance using fewer transmission lines but with some additional complexity. Simulation results using full-system simulation show that TLC provides more consistent performance than the DNUCA design across a wide variety of workloads. TLC caches are logically simpler than DNUCA designs, but require greater circuit and manufacturing complexity.

1 Introduction

The disproportionate scaling of VLSI interconnect and transistor performance has been recognized as a key challenge for future high performance systems [17, 35]. This problem manifests itself most strongly in global wires that communicate across a large fraction of a chip. For example, sending a signal across a 2 cm die required only one to two clock cycles at the beginning of this decade [13], but will take over 25 cycles by its end for aggressively clocked processors [14, 18].

The problem of slow global wires has prompted substantial microarchitectural research to reduce their impact on system performance [30, 31]. For example, Kim *et al.* recently proposed a novel design for large on-chip level-2 caches, which are increasingly performance critical due to longer memory latencies, more pressing power constraints, and limited off-chip bandwidth [24]. Their *Dynamic Non-Uniform Cache Architecture (DNUCA)* is a physical organization that exploits the fact that closer cache banks can be accessed more rapidly than more distant banks. DNUCA achieves impressive performance improvements over other alternatives, but introduces significant logical complexity, along with power and area inefficiencies.

An emerging alternative approach to the slow global wire problem is to use on-chip transmission lines [8]. Transmission lines exhibit much lower latencies than conventional wires since their signalling speed is dominated by a relatively short inductive-capacitance (LC) delay rather than a series of a relatively large resistive-capacitance (RC) delays[1]. The speed of the incident wave across a transmission line is analogous to the speed of a ripple moving across water in a bathtub, while the latency across conventional RC wires is analogous to changing the water level of the bathtub.

Despite their substantial speed advantage—up to a factor of 30 by the end of the decade—transmission lines will not replace most conventional on-chip wires because they sacrifice significant bandwidth. Transmission lines require very wide, thick wires and dielectric spacing to operate in the LC range, which are only available in the uppermost layers of a chip's interconnection metal stack. These extremely sparse metal layers are best utilized for

This work was supported by the National Science Foundation (CDA-9623632, EIA-9971256, EIA-0205286, and CCR-0324878), a Wisconsin Romnes Fellowship (Wood), and donations from Intel Corporation and Sun Microsystems, Inc. Dr. Wood has a significant financial interest in Sun Microsystems, Inc.

[1] In other words, the latency of a transmission line to the first order is determined by the speed of light in the dielectric surrounding the interconnect instead of the time to change the charge across the wire's capacitance.

Proceedings of the 36th International Symposium on Microarchitecture (MICRO-36 2003)
0-7695-2043-X/03 $17.00 © 2003 **IEEE**

the few long distance communication links whose latency can have a significant impact on overall system performance.

In this paper, we explore using transmission lines for communication between the storage banks of large on-chip caches and their central controllers. We refer to these caches as *Transmission Line Cache (TLC)* designs. By using long on-chip transmission lines, TLC achieves the following advantages over DNUCA:

- TLC provides consistent high performance for a wide variety of workloads with different sized memory footprints because its entire storage is accessible within 16 cycles using low contention point-to-point links.
- TLC's simple logical design eases logical verification and integration with dynamic instruction schedulers.
- By eliminating repeaters and communicating through on-chip transmission lines that can be routed over the cache banks, TLC consumes 18% less substrate area than DNUCA and allows for more efficient layout.
- TLC reduces the power consumed within the communication network of a large on-chip cache.

However TLC does have the following disadvantages compared to DNUCA:

- TLC's transmission line drivers and receivers require a greater circuit verification effort to ensure proper signalling in the noisy environments of future integrated circuits.
- TLC demands significantly more metal layers resulting in a higher per wafer manufacturing cost than the DNUCA design, which uses conventional interconnect.

The rest of the paper is organized as follows. Section 2 reviews the global wire problem and how the DNUCA design addresses it. Section 3 discusses on-chip transmission lines and the technology assumptions we made for this study. Section 4 describes the family of TLC designs. Section 5 and Section 6 describe the methodology and results of our simulation experiments that compare the performance of TLC and DNUCA.

2 Wire Delay and Caches

As previously mentioned, the delay of conventional interconnect relative to transistors is increasing as integrated circuits move to smaller geometries. Global wires (> 1 mm) are particularly vulnerable because the RC delay of conventional interconnect grows quadratically with distance [42]. To keep wire delay linear with distance, designers insert repeaters to break long wires into multiple shorter segments. However, increasing wire density and operational frequencies dictate an increasing number of repeaters.

Overall, the use of repeaters for global communication leads to three key problems [17]:

- Repeaters require a substantial amount of area for their large transistors.
- Repeaters necessitate disciplined floorplanning to allocate the necessary substrate area at the proper locations.
- Repeaters need many via cuts from the upper metal layers down to the substrate, which congest the interconnection layers below and reduce the overall wire bandwidth.

Furthermore, more localized (< 1 mm) wire delay is also a significant factor in the design of on-chip caches. For instance, current level-2 caches are divided into multiple smaller banks to optimize the individual bank's area/delay tradeoff [25]. While partitioning a cache into banks mitigates the impact of localized wire delay, the global wire delay to access the appropriate banks becomes the dominant factor as chips move to smaller geometries. Kim *et al.* [24] showed that—for a 16 MB L2 cache in the 45 nm generation—the delay to reach individual banks ranged from 3 to 47 cycles. Clearly, a conventional cache with uniformly slow access time would have unacceptable latency and bandwidth.

Kim *et al.* address this problem by defining a family of Non-Uniform Cache Architecture (NUCA) designs. Similar to Figure 1, all practical NUCA designs assume a 2D array of cache banks accessed via a 2D switch interconnect implemented using conventional RC-delay wires. The dedicated communication channels between the cache banks reserve the necessary substrate area for the data link's repeaters. The static, or SNUCA, designs use low-order address bits to determine which cache blocks map to each bank. The dynamic, or DNUCA, designs exploit locality by migrating frequently accessed blocks to the cache banks closest to the controller. Dynamic placement reduces the average access time, but introduces significant additional design complexity, power consumption, and bandwidth demand.

The DNUCA design is a very large (+30-way) set-associative cache, with banks grouped into different bank sets where a given block address may reside. A reference that hits in the closest two banks of a bank set, a *close hit*, takes the minimum time, but a miss may require a search of all the remaining banks in the bank set. DNUCA uses a partial tag structure to avoid this worst case for most accesses. The partial tag structure stores the six least significant bits of all tags and is accessed in parallel with the

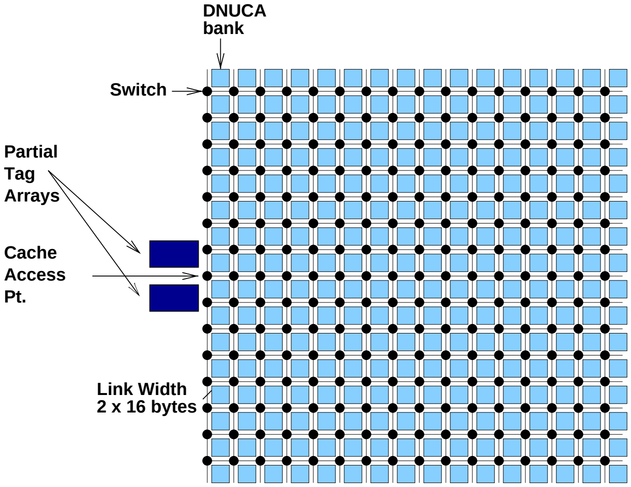

Figure 1. 16 MB DNUCA Block Diagram

closest two banks of the bank set. If a request misses in the closest banks, the partial tag comparison indicates which other banks need to be searched. In some cases, the partial tag check indicates that no other banks need be search, a so-called *fast miss*. Partial tags improve performance directly, by reducing searches, and indirectly, by reducing interconnect link contention.

While partial tags provide many benefits, keeping them consistent with the cache contents introduces significant complexity. In particular, the partial tags must be updated when blocks migrate to closer banks. Due to contention in the mesh network, blocks are not guaranteed to move from one bank to the other in a fixed time. Thus a complex synchronization mechanism is required to ensure that blocks are not missed during a search. While these complications are certainly manageable, they represent a significant additional design and verification effort.

3 On-chip Transmission Lines

Printed-circuit board and other off-chip wire technologies are commonly designed to behave as transmission lines [10]. Conversely, although on-chip transmission lines using non-conventional technology have been explored for over 20 years [38], on-chip wires using CMOS technology are normally designed to operate as lossy RC lines [41]. But with improving fabrication technology, on-chip transmission lines are starting to emerge in CMOS circuits. For example, several current high performance chips use transmission lines for the long global wires (~ 0.75 cm) used for clock distribution [29, 40, 43]. Longer (> 1 cm) transmission lines operating in the 7.5-10 GHz frequency range have been shown to work on CMOS test chips using very wide wires [8] or low operating temperatures [11]. With the introduction of lower-k dielectrics [7] and increasing on-chip frequencies [18], more practical on-chip transmission lines will be available before the end of the decade.

In this paper, we explore on-chip transmission line communication. Specifically, we investigate using single-ended voltage-mode signalling, where standard voltage signals propagate across a single point-to-point link. To reduce reflection noise across these relatively low loss transmission lines, we assumed source-terminated drivers with digitally-tuned resistance [10]. Receivers use a large input impedance termination for full wave reflection of the received signal. Single-ended voltage-mode signalling best fits the low utilization of on-chip interconnection networks.

The physical transmission line is a single long wire that is routed directly from the driver to the receiver without repeaters. Because of the length of transmission lines, thicker and wider metal tracks are required to maintain low wire resistance. Additionally, thicker intermetal dielectrics are necessary to control wire capacitance on these long fat wires so that they can operate as transmission lines. These transmission lines must be laid out in stripline fashion with a reference plane both above and below the transmission line metal layer to provide low resistance return paths for inductive induced currents [32]. While transmission line dimensions are much larger than the dimensions proposed for future conventional interconnect, they are actually very similar to the upper metal layers of previous high performance processors [6] and current silicon microwave chips [33]

At these large wire dimensions, the “skin effect” significantly increases the signals’ susceptibility to noise. The skin effect phenomenon arises because at high frequencies, magnetic repulsion forces current towards the perimeter of the conductor, thereby reducing the wire’s effective cross section. Thus higher frequency signals encounter effective resistances greater than the wire’s DC resistance. This effect is compounded by the fact that a digital pulse is composed of many sinusoidal signals of different frequencies. Because the different components of a digital pulse encounter different effective resistances, the receiver sees a signal that is rounded and stretched out. Noise is a significant issue when receiving these attenuated signals.

To reduce the noise susceptibility, we propose using alternating power and ground shielding [22] lines between each transmission line, in addition to the reference planes above and below the signal layer. Laying out the lines in this manner not only provides several individual low-resistive return paths, but also isolates each line from most capacitive and inductive cross-coupling noise.

Adding metal layers for reference planes will add significant manufacturing cost to the chip compared to conventional CMOS technology. However, the International Technology Roadmap for Semiconductors already projects, for the year 2010, integrating four reference planes into high performance chips to provide inductive shielding and decoupling capacitance [14]. Only time will tell if the benefits of transmission lines will justify their cost, but the history of silicon processing shows us that many complex and expensive enhancements have been adopted, including copper wires [13] and SOI devices [9]. We believe on-chip transmission lines could be the next manufacturing enhancement that drives system performance into the next decade.

4 Transmission Line Cache Designs

One interesting opportunity for on-chip transmission lines is as a low latency interface between the cache's storage and its controller. We targeted our Transmission Line Cache designs for the 45 nm technology generation [14], with an aggressive CPU core operating frequency of 10 GHz [18]. At this design point, the tremendous speed advantage of transmission lines will not only provide improved cache performance, but will also permit trading off some performance for a simpler design consuming less area and power. We analyze a 16MB TLC to allow direct comparison with the 16 MB DNUCA design using the same technology assumptions [24].

TLC exploits the tremendous speed and layout benefits of transmission lines to decouple the cache storage from the cache controller. Because transmission lines can quickly communicate across long distances without using repeaters, the large storage area of the cache can consume the less valuable real estate on the edges of the chip, while the cache controller can be moved to the center of the chip where it can be quickly accessed by the processor core. This design is less feasible using conventional global wires because of their intermediate repeater requirement discussed previously. Conversely, the on-chip transmission lines used by TLC don't require repeaters and can be routed over other logic without congesting intermediate wiring tracks and the substrate area below.

Figure 2 shows the high-level floorplan for the base TLC design. This cache is composed of 32-512 KB banks where half the banks line one edge of the die and the other half line the opposite edge. The space between the banks would be consumed by the processor core and L1 caches. On each edge, the banks are stacked in two columns of eight. Each pair of adjacent banks share two eight-byte wide unidirectional transmission line links to the L2 cache controller, creating a high bandwidth, low latency interface between the controller and the storage banks.

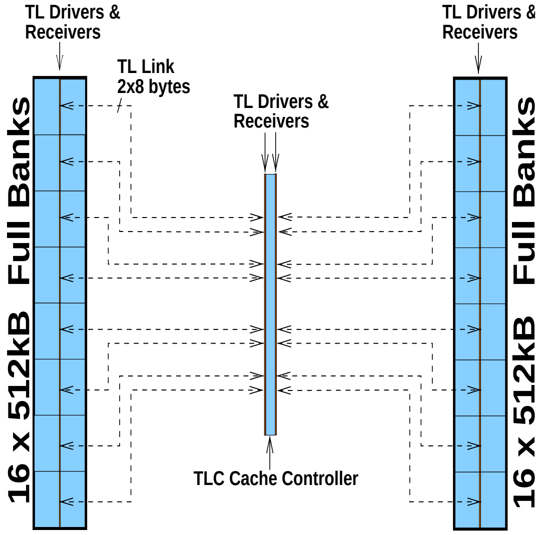

Figure 2. Base TLC Top Level Floorplan

Because the individual transmission lines vary in length, we adjust their width to maintain appropriate resistance and capacitance, as shown in Figure 3 and Table 1. Figure 3 also compares the dimensions of the transmission lines used in TLC with the dimensions of the conventional RC wires used in the DNUCA communication network.

The width of the transmission lines used in TLC determine the size of a cache controller. The cache controller must be tall enough so that all the transmission line links can connect with it. The cache controller must be wide enough so that the each link has a direct connection to the center of the controller where the cache request originates. The TLC cache controller uses conventional wires to communicate between the transmission lines located on its edges and the controller logic located at its center. These conventional wires add up to three additional delay cycles to the TLC access times.

As mentioned in the previous section, transmission lines are increasingly sensitive to noise corruption. To further compensate for skew due to discontinuities across the transmission lines, we enforce extremely conservative setup and hold times of at least 40% of the entire clock cycle for the TLC signals. Remaining faults on the transmission lines could be repaired using end-to-end ECC checks. For instance, the IBM Power 4 already performs ECC checks when accessing the on-chip L2 cache [37]. End-to-end ECC simply means generating and checking the codes in the central controller. We believe that these measures are enough to ensure that single-ended voltage-mode transmission lines will perform correctly in the noisy environments of future chips. If one desires extra reliability, there are other techniques to increase noise immunity such as

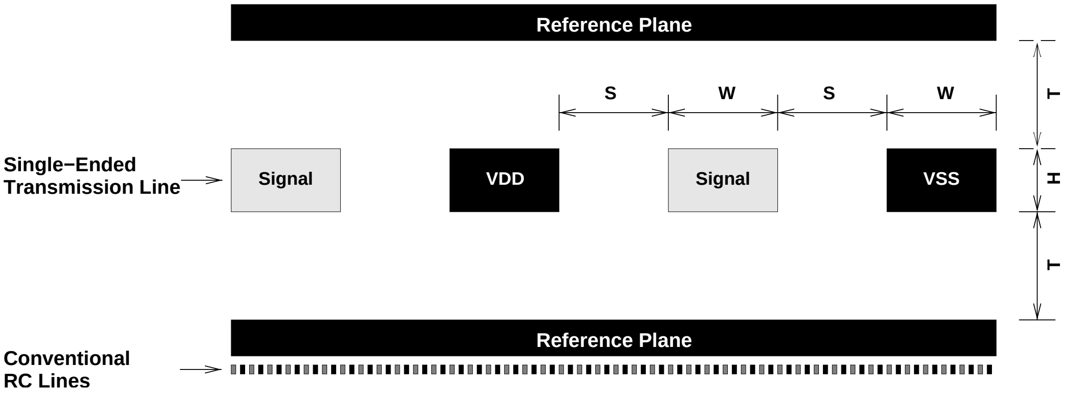

Figure 3. Cross-sectional comparison (45 nm technology)

Table 1. Transmission Line Dimensions

Length	W: Width	S: Spacing	H: Height	T: Thickness
0.9 cm	2.0 μm	2.0 μm	1.75 μm	3.0 μm
1.1	2.5	2.5	1.75	3.0
1.3	3.0	3.0	1.75	3.0

using differential signals with a sinusoidal carrier frequency [8] or current-mode drivers [10].

Optimized Transmission Line Caches. The high bandwidth interface of the base TLC design comes at considerable cost in wire area. We anticipate that extra metal layers will be required to implement the transmission lines needed by the base TLC design. As a cheaper alternative, we consider optimized TLC designs that require fewer transmission lines, perhaps permitting their integration into the existing uppermost metal layers. Figure 4 introduces the top level floorplan of these designs and Table 2 summarizes the parameters of our entire family of TLC designs.

The Optimized TLC designs (TLCopt) are able to reduce the number of required wires through three methods:

- Storing the 64-byte cache block across multiple banks to reduce the amount of data needed to be transferred between the cache controller and an individual bank per cache request.
- Doubling the cache bank size from 512KB to 1MB thus reducing the number of banks the cache controller interfaces with by half.
- Supplying each bank with only enough address information to access the correct set and perform a 6-bit partial tag [21] comparison. The full tag comparison is performed later at the cache controller.

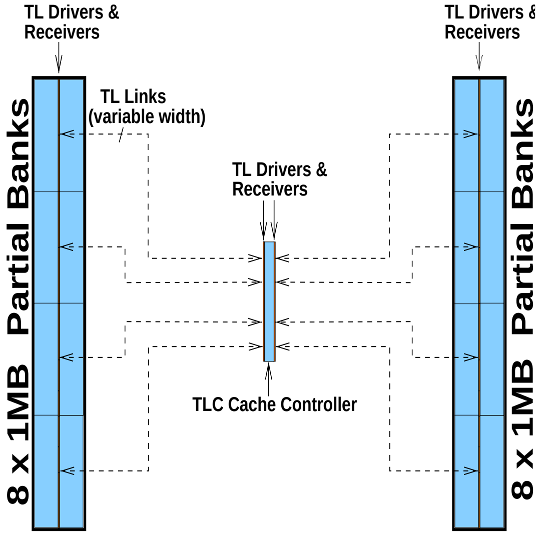

Figure 4. Optimized TLC Top Level Floorplan

In these designs, each bank is responsible for storing only a portion of the most significant bits of the cache tag along with the lower 6-bits of the tag. The bank uses this 6-bit partial tag to do a quick comparison, determining if a request hits. Because all banks holding a block store the same lower 6-bit partial tag, all tag comparisons among them will have the same result. When the banks respond to a load request, they send its higher order tag bits along with the data to the cache controller which performs the full tag comparison. In the infrequent case of multiple partial tag matches, the banks respond with the high order tag bits of all matching entries. The controller determines which set entry, if any, actually matches, and then request the specific block. Because all our TLC designs are exclusive write-back caches, store requests are simply written to the cache without requiring any tag comparisons.

As an additional benefit of using fewer transmission lines, the TLCopt designs require smaller cache controllers. This is because the TLC cache controller's height is

Table 2. Design Parameters

Design	Banks	Banks/ Block	Bank Size	Transmission Lines per Bank Pair	Total Transmission Lines Used	Uncontended Latency	Bank Access Time
TLC							
TLC	32	1	512 KB	128	2048	10 - 16 cycles	8 cycles
TLCopt 1000	16	2	1 MB	126	1008	12 - 13	10
TLCopt 500	16	4	1 MB	64	512	12	10
TLCopt 350	16	8	1 MB	44	352	12	10
NUCA							
SNUCA2	32	1	512 KB	n/a	n/a	9 - 32	8
DNUCA	256	1	64 KB	n/a	n/a	3 - 47	3

determined by the sum of the transmission lines' width and spacing. Reducing the cache controller's size also reduces the communication delay within the controller. This reduction in communication delay between the transmission lines and the central controller logic offsets the increase in the bank access time due to the fewer independent banks used in the TLCopt designs.

5 Methodology

Our evaluation methodology can be broken down into two separate parts. First, we designed and simulated the physical on-chip transmission lines used by TLC. Second, we evaluated the performance and estimated the dynamic power consumption of TLC as it compares to DNUCA using a full-system simulator.

Physical Evaluation. The goal of our physical evaluation was to first investigate the usage of on-chip transmission lines in future technology and then to evaluate their performance. We started by using Linpar [12], a 2-dimensional field-solver program, to extract the inductance, resistance and capacitance characteristics of on-chip transmission lines. Once we had RLC matrices describing the transmission lines, we simulated 10 GHz pulses travelling across the lines using HSPICE. Specifically, we modeled the transmission line's frequency dependent attenuation with HSPICE's W element transmission line model. We simulated four signal wires with shielding wires separating each of them under worst case signalling conditions. We took the output waveforms to determine the latency of the transmission lines, as well as ensured the received signals had an amplitude of at least 75% of V_{dd} and a pulse width of at least 40% of the processor cycle time.

We used the tool ECACTI [1] to determine the access latency and layout of the cache banks. Our models for delay [3], gate capacitance [17], and transistor sizes [34] allowed us to estimate the size and power of future interconnect as well as the switches used in NUCA [23, 39].

Performance Evaluation. We evaluated the system performance of each cache design using a dynamically scheduled SPARC V9 uniprocessor. To simulate our target system, we used the full system simulator Simics [26] extended with a detailed processor [27] and memory system timing model. Our detailed memory timing simulator for DNUCA and TLC included modelling contention within the links, switches and banks in each design. Table 3 summarizes our simulation parameters.

We evaluated all cache designs using 12 different benchmarks: four SPECint 2000 benchmarks (bzip, gcc, mcf, and perl), four SPECfp 2000 benchmarks (equake, lucas, swim, and applu) [36], and four commercial benchmarks described in Table 5 [2]. We warmed up the caches, as shown in Column 3 of Table 4, then evaluated each design over the amount of work indicated in Column 4.

6 Evaluation Results

This section evaluates the impact of TLC on a future high performance microprocessor. Section 6.1 shows that the base TLC design provides comparable overall performance to DNUCA while providing more predictable behavior and consuming less area and power within the interconnect. Section 6.2 evaluates the link utilization of all the TLC designs and shows that the TLCopt designs can attain similar performance to the base TLC design, while using significantly fewer wires.

6.1 TLC vs. DNUCA

Overall Performance. Figure 5 compares the normalized execution time of the DNUCA and base TLC designs using the statically partitioned SNUCA2 cache design as a baseline [24]. SNUCA2 is the static NUCA

Table 3. Simulation Setup

Memory System		Dynamically Scheduled Processor	
split L1 I & D caches	64 KBytes, 2-way, 3cycles	reorder buffer / scheduler	128 / 64 entries
unified L2 cache	16 MBytes, DNUCA or TLC*	pipeline width	4-wide fetch & issue
cache block size	64 Bytes	pipeline stages	30
memory latency	300 cycles	direct branch predictor	3.5 kBytes YAGS
memory size	4 GBytes of DRAM	indirect branch predictor	256 entries (cascaded)
outstanding memory requests	8	return address stack	64 entries

* 4-way set associative, with LRU replacement

Table 4. Evaluation Methodology

Benchmark	Fast Forward	Warm-up	Executed
SPECint 2000	1 - 5 Billion instr.	500 Million instr.	500 Million instr.
SPECfp 2000	500 Mill - 3 Bill instr.	1 Billion instr.	500 Million instr.
Apache	500000 transactions	2000 transactions	2000 transactions
Zeus	500000 transactions	2000 transactions	2000 transactions
SPECjbb	1000000 transactions	15000 transactions	10000 transactions
OLTP	100000 transactions	300 transactions	250 transactions

Table 5. Commercial Workload Description

Static Web Servers (Apache & Zeus): We use Apache 2.0.36 and Zeus 4.2 for SPARC/Solaris 9, configured to use pthread locks and minimal logging as the web server. We use SURGE [5] to generate web requests. We use a repository of 80,000 files (totaling ~2 GB). These files are fetched by 400 clients.
Java Server Workload: SPECjbb (Sjbb). SPECjbb2000 is a server-side java benchmark that models a 3-tier system, focusing on the middleware server business logic. We use Sun's HotSpot 1.4.0 Server JVM. Our experiments use 24 threads and 24 warehouses (a data size of ~500MB per warehouse).
Online Transaction Processing (OLTP): DB2 with a TPC-C-like workload. The TPC-C benchmark models the database activity of a wholesale supplier, with many concurrent users performing transactions. Our OLTP workload is based on the TPC-C v3.0 benchmark using IBM's DB2 v7.2 EEE database management system. We use an 5GB database with 25000 warehouses stored on eight raw disks and an additional dedicated database log disk. We reduced the number of districts per warehouse, items per warehouse, and customers per district to allow for concurrency provided by a larger number of warehouses. There are 16 simulated users.

design using a two-dimensional grid interconnect. Except for some of the SPECFP benchmarks, both TLC and DNUCA significantly improve overall system performance compared to SNUCA2. The lack of performance impact TLC and DNUCA have on the SPECfp benchmarks, lucas, swim, and applu, is due to the extremely high L2 miss rates of these benchmarks, as shown in Columns 3 and 4 of Table 6 [15]. DNUCA is particularly hurt by the low temporal locality and high miss rates of the swim and applu benchmarks. DNUCA inserts all data blocks brought in from memory into the furthest banks from the cache controller and then promotes the blocks to its closer, quickly accessible banks every time the block is accessed. This promotion policy relies on the expectation that most cache requests will be for a small set of frequently accessed blocks. However, in benchmarks where most requests miss in the cache, this policy fails to improve cache access times. This behavior is shown by the low ratio of DNUCA block promotions to block insertions for these two SPECfp benchmarks, Column 6 of Table 6.

The fourth SPECfp benchmark, equake, is particularly interesting. Equake uses a finite element method on sparse matrices to simulate seismic waves propagating in a

Table 6. Benchmark Characteristics

Bench	Total L2 Requests	TLC L2 misses / 1K instr.	DNUCA L2 misses / 1K instr.	DNUCA close hit%	DNUCA promotes / inserts	TLC predictable lookup %	DNUCA predictable lookup %
bzip	4.8×10^6	0.051	0.052	81%	64	92%	56%
gcc	3.8×10^7	0.068	0.070	99	610	99	62
mcf	5.5×10^7	0.019	0.019	48	12000	82	24
perl	2.6×10^6	0.028	0.028	97	9.7	96	90
equake	6.2×10^6	6.8	5.2	16	0.55	90	38
swim	2.4×10^7	40	38	0.7	0.15	98	39
applu	9.0×10^6	16	16	1.0	0.06	98	38
lucas	7.8×10^6	13	12	7.2	0.15	99	49
apache	1.5×10^7	4.8	3.8	67	3.7	98	61
zeus	1.4×10^7	6.4	4.8	60	2.5	97	57
Sjbb	$7.1 \text{x} 10^6$	2.3	2.3	58	1.9	93	59
oltp	3.3×10^6	0.93	0.79	89	13	98	77

large basin [4]. Like the other three SPECfp benchmarks, equake streams through a lot of data, but Equake also has a large data set that it frequently accesses. DNUCA's frequency replacement policy separates the two groups of data within its highly associative sets, so that the streaming data does not evict the frequently accessed data. On the other hand, TLC's LRU replacement policy is unable to disambiguate between the two data sets leading to a higher miss rate and lower performance for this benchmark.

Figure 5 also shows DNUCA and TLC perform very well for the SPECint and commercial workloads that have much lower miss rates. While DNUCA significantly improves performance for these workloads which have a high percentage of hits to the closest banks of cache, TLC significantly improves the performance of workloads like mcf which has a large memory footprint [15]. Overall, TLC moves the cache storage away from the processor core, while providing comparable performance improvement to DNUCA over the set of benchmarks.

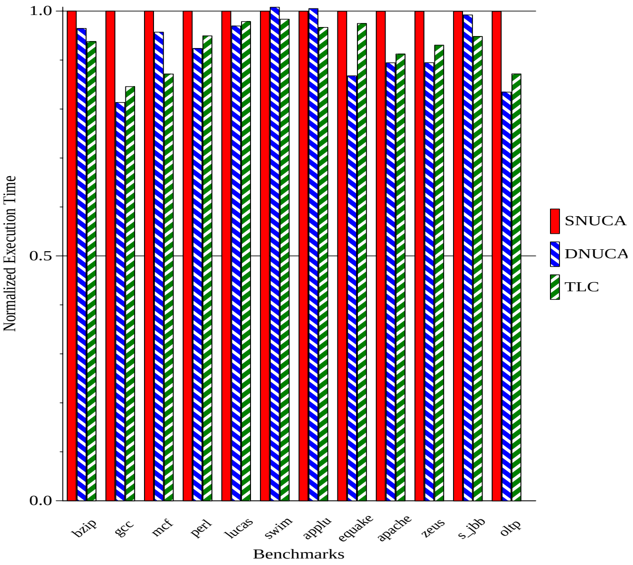

Figure 5. Normalized Execution Time

Performance Predictability. TLC exhibits more predictable performance than DNUCA because it provides a more consistent response latency for L2 cache accesses. Therefore an instruction scheduler can rely on TLC's predictable latency for scheduling dynamic operations, thus simplifying its circuits. Additionally, schedulers performing speculative memory scheduling on L2 accesses will encounter significantly fewer replays using TLC.

TLC's statically partitioned banks and high bandwidth interface enable TLC to provide more consistent lookup latency than DNUCA. Figure 6 plots the mean cache lookup latency for the two cache designs over all twelve benchmarks. As expected, TLC encounters more bank contention due to its fewer banks and longer bank access latencies, while DNUCA encounters more contention in the routing network to and from the banks. The key observation is that TLC offers a more consistent mean lookup latency of around 13 cycles for all the benchmarks, while the mean lookup latency of DNUCA varies tremendously among benchmarks.

Columns 7 and 8 of Table 6 compare the predictability of lookup latency for the TLC and DNUCA designs.

Column 7 shows that 10% or less of TLC lookup latencies are mispredicted for all but mcf. Column 8 shows that at least 40% of DNUCA lookups are mispredicted, for two-thirds of the benchmarks. Because TLC has a predictable lookup latency and a high fraction of non-delayed requests, TLC can be easily integrated into a dynamic instruction scheduler, while the wide variation of access times for DNUCA significantly complicates dynamic instruction scheduling.

Furthermore, as pipelines become deeper with a greater distance between when the instruction issue and execution stages, we believe aggressive dynamic schedulers will perform speculative memory scheduling on L2 accesses. Speculative memory scheduling is a technique performed by current high performance microprocessors to improve the load-to-use latency between instructions. Rather than waiting for a cache “hit” signal, some processors with predictable cache lookup latencies [16, 20] will either predict a load hits in the cache and speculatively issue the load’s dependent instructions or predict that the load misses and issue other independent instructions instead. Speculative memory scheduling reduces load-to-use latency by allowing dependent instructions to meet their source data at the execution units as soon as possible, while not wasting valuable issue bandwidth in the scheduler. However, when the scheduler mispredicts that a load will hit in the cache, the speculatively issued dependent instructions must be replayed.

Speculative memory scheduling on L2 accesses is significantly more difficult due to their difficult to predict access latencies. One solution is to access the L2 cache tags early to provide a hint of when the data will arrive. For example, Itanium 2 [28] uses a centralized tag structure to provide early hit or miss indication for its 256KB L2 cache.

However, due to increasing wire delays, a centralized cache tag structure for future large on-chip caches may be impractical. For instance the tag array for a 16 MB cache is nearly 1 MB, accessing such a large tag array will add several cycles of unnecessary latency to many cache lookups. Instead, we believe future large caches will use a more distributed design and be partitioned into banks of tag and data arrays. These more distributed caches, like the NUCA caches, will have much lower mean access times than a centralized cache of similar size. However, the wide variance in their access times will only add to the non-determinism of their accesses. On the other hand, TLC has very predictable lookup latency and therefore could be easily integrated into future aggressive schedulers.

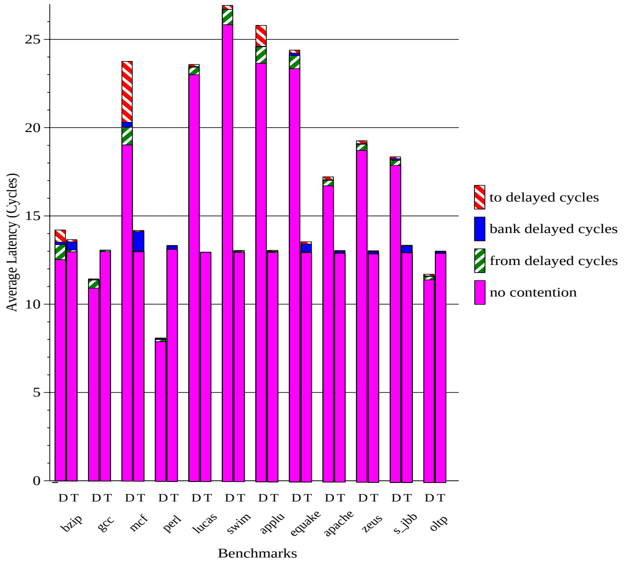

Figure 6. Mean Cache Lookup Latency (Cycles)

Table 7. Consumed Substrate Area

Cache Design	Storage Area	Channel Area	Controller Area	Total Area
DNUCA	92 mm^2	17 mm^2	1.1 mm^2	110 mm^2
TLC	77	3.1	10	91

Table 8. Cache Communication Network Characteristics

Cache Design	Total Transistors	Total Transistor Gate Width
DNUCA	1.2 x 10^7	440 million λ
TLC	1.9 x 10^5	20 million λ

Area. Although TLC requires additional metal layers, it significantly reduces substrate area and hence overall die size. Table 7 breaks down the substrate area requirements of DNUCA and TLC. Table 7 shows that the latency benefit of DNUCA’s smaller banks comes at an increased cost in bank area (Column 2) and an even greater increased cost in routing channel area (Column 3). Column 4 shows that the DNUCA partial tag structure adds a relatively small amount to the total cache area. On the other hand, TLC’s large dense banks and lack of repeaters in the communication network saves storage and channel area, though its cache controller area is much larger due to its interface with the wide transmission line wires. Overall, TLC reduces substrate area by 18% compared to DNUCA.

Power. Current low-power, low-voltage drivers [19] for off-chip transmission lines consume too much static

Table 9. Dynamic Components

Benchmark	DNUCA Banks Accessed/Request	TLC Banks Accessed/Request	DNUCA Network Dynamic Power	TLC Network Dynamic Power
bzip	2.3 banks	1 bank	150 mW	56 mW
gcc	2.0	1	150	100
mcf	2.6	1	350	150
perl	2.0	1	63	36
equake	2.5	1	87	23
swim	2.5	1	190	56
applu	2.5	1	110	34
lucas	2.5	1	57	17
apache	2.4	1	200	67
zeus	2.4	1	170	53
Sjbb	2.4	1	130	43
oltp	2.1	1	220	90

power to be implemented for low utilized on-chip signals. Instead, TLC uses a more traditional single-ended voltage-mode driver with active high signalling. These drivers not only save power as compared to their contemporary low-voltage counterparts, but actually allow TLC to consume less power than a cache using conventional RC interconnect. The rest of this section breaks down both the TLC and DNUCA interconnection networks' static and dynamic power consumption to show how TLC can save power compared to DNUCA.

While determining the exact static power consumption is difficult early in the design process, it is well understood that static power is dominated by transistor leakage current which is directly dependent on transistor width [34]. By removing intermediate switches, latches, and repeaters, as well as not requiring a partial tag array, TLC significantly reduces the transistor demand of the cache communication network. As shown in Table 8, we estimate an over 50 fold reduction of transistors for TLC in comparison to DNUCA. Table 8 also indicates the total transistor gate width would be reduced by over an order of magnitude. Therefore the TLC communication network will save leakage power versus the DNUCA network.

Dynamic power dissipation is dependent on the signalling strategy of the interconnect. Signalling across conventional RC interconnect using repeaters relies on charging and discharging the capacitance of each wire segment from one voltage value to another. Therefore for conventional signalling, dynamic power equals the power required to change the voltage, V, across the wire's total capacitance, C, for a given frequency, f, and data activity factor, α [34]:

$$\begin{matrix}\mathit{Conventional\ Signalling}\\ \mathit{Dynamic\ Power}\end{matrix} = \alpha \times C \times V^2 \times f$$

In voltage-mode transmission line signalling, the dynamic power consumed is the power required to create the incident wave. At the driver, the transmission line looks like a resistor equal to the characteristic impedance of the line. Therefore the power supplied by the driver is determined by voltage across its internal resistance, R_D, in series with the transmission line's characteristic impedance, Z_0, for the duration of the signal pulse, t_b, [10]:

$$\begin{matrix}\mathit{Transmission\ Line}\\ \mathit{Dynamic\ Power}\end{matrix} = \alpha \times t_b \times \frac{V^2}{(R_D + Z_0)} \times f$$

Comparing the dynamic power dissipation of matched voltage-mode transmission lines ($R_D = Z_0$) to that of conventional wires, one sees that when $t_b/(2 \times Z_0) < C$, transmission lines will consume less dynamic power than conventional interconnect. As cycle times continue to decrease, this relationship will hold for long global links beyond ~1 cm in length.

Table 9 compares the dynamic power components of the two cache designs. While the total amount of dynamic power is relatively small for both designs, TLC does reduce dynamic power dissipation within the communication network by utilizing on-chip transmission lines. TLC also significantly reduces the number of banks accessed per cache request leading to a greater reduction in dynamic power consumption as compared to DNUCA.

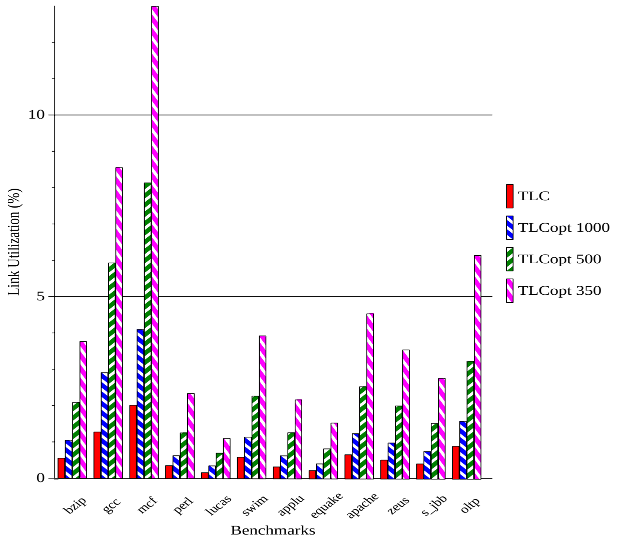

Figure 7. TLC Average Link Utilizaiton

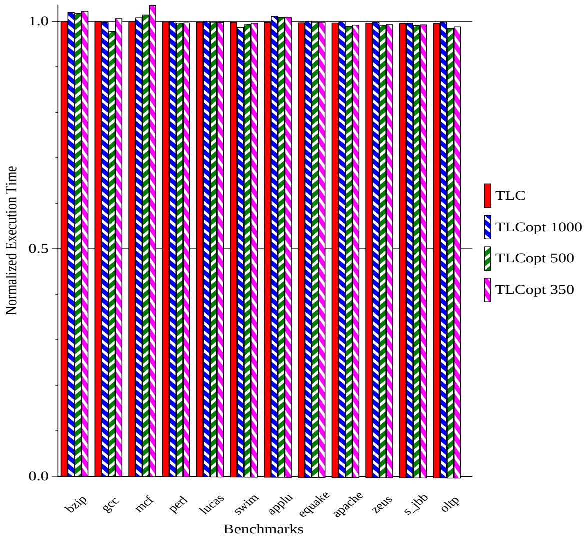

Figure 8. TLC Normalized Execution Time

6.2 The Family of TLC designs

This section evaluates link utilization for the family of TLC designs and shows that similar performance to the base TLC design can be achieved using significantly fewer wires. Link utilization is the percentage of cycles where the transmission lines actually communicate data. Figure 7 plots the average link utilization for each TLC design across the spectrum of benchmarks. One should first notice that the base TLC link utilization never exceeds 2% for any benchmark and for most benchmarks it hovers below 1%. This extremely low utilization shows that the base TLC design has more bandwidth than necessary. As expected, the TLCopt designs have an increasing degree of link utilization consistent with their reduction in transmission line wires. However, even the utilization of the TLCopt 350 design remains relatively low, never surpassing 13%.

Figure 8 shows that this increase in link and bank contention for the TLCopt designs does not translate into a significant performance degradation compared to the base TLC design. For some benchmarks, the TLCopt designs achieve slight improvements in execution time due to their slightly lower cache access latencies (Table 2). Overall multiple partial tag matches in the TLCopt designs occurred in approximately 1% of the cache lookups, thus the increased messages sent between the cache controller and the banks has little effect on performance.

7 Conclusions

We have proposed an alternative family of cache designs using emerging on-chip transmission line technology. On-chip transmission lines offer a significant latency advantage to conventional global interconnect for communicating distances greater than a few millimeters. However, due to their power and bandwidth characteristics, on-chip data transmission lines will be practically limited to long (> 1cm) performance critical signals.

TLC is one such application of on-chip transmission lines. Our TLC designs perform comparably to the previous DNUCA strategy, while saving area and power. Furthermore, they provide a spectrum of reduced logical complexity solutions, but require significant circuit and manufacturing cost. To combat the increased wire demand of the base TLC design, we introduced three optimized TLC designs that consume less wires and perform comparably for most benchmarks.

Acknowledgments

We thank Peter Hsu for inspiring this work and giving us helpful feedback. We thank Doug Burger, Steve Keckler, Changkyu Kim and the Texas CART group for help with the DNUCA comparison. We thank Virtutech AB, the Wisconsin Condor group, and the Wisconsin Computer Systems Lab for their help and support. We thank Alaa Alameldeen, Brian Fields, Mark Hill, Mike Marty, Carl Mauer, Kevin Moore, Min Xu, the Wisconsin Computer Architecture Affiliates, and the anonymous reviewers for their comments on this work.

References

[1] V. Agarwal, S. W. Keckler, and D. Burger. The Effect of Technology Scaling on Microarchitectural Structures. *Technical Report TR-00-02, Department of Computer Sciences, University of Texas at Austin*, May 2001.

[2] A. R. Alameldeen, M. M. K. Martin, C. J. Mauer, K. E. Moore,

M. Xu, D. J. Sorin, M. D. Hill, and D. A. Wood. Simulating a $2M Commercial Server on a $2K PC. *IEEE Computer*, 36(2):50–57, Feb. 2003.

[3] B. S. Amrutur and M. A. Horowitz. Speed and Power Scaling of SRAMs. *IEEE Transactions on Solid-State Circuits*, 35(2):175–185, Feb. 2000.

[4] H. Bao, J. Bielak, O. Ghattas, L. F. Kallivokas, D. R. O'Hallaron, J. R. Shewchuk, and J. Xu. Large-scale simulation of elastic wave propagation in heterogeneous media on parallel computers. *Computer Methods in Applied Mechanics and Engineering*, pages 85–102, 1998.

[5] P. Barford and M. Crovella. Generating Representative Web Workloads for Network and Server Performance Evaluation. In *Proceedings of the 1998 ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, pages 151–160, June 1998.

[6] B. J. Benschneider and et. al. A 300-MHz 64-b Quad-Issue CMOS RISC Microprocessor. *IEEE Journal of Solid-State Circuits*, 30(11):1203–1214, Nov 1995.

[7] A. S. Brown. Fast Films. *IEEE Spectrum*, 20(2):36–40, Feb. 2003.

[8] R. T. Chang, N. Talwalkar, C. P. Yue, and S. S. Wong. Near Speed-of-Light Signaling Over On-Chip Electrical Interconnects. *IEEE Journal of Solid-State Circuits*, 38(5):834–838, May 2003.

[9] C. T. Chaung. Design Considerations of SOI Digital CMOS. In *Proceedings of the IEEE 1998 International SOI Conference*, pages 5–8, 1998.

[10] W. J. Dally and J. W. Poulton. *Digital Systems Engineering*. Cambridge University Press, 1998.

[11] A. Deutsch. Electrical Characteristics of Interconnections for High-Performance Systems. *Proceedings of the IEEE*, 86(2):315–355, Feb. 1998.

[12] A. R. Djordjevic, M. B. Bazdar, T. K. Sarkar, and R. F. Harrington. *Matrix Parameters for Multiconductor Transmission Lines: Software and User's Manual*. Artech House, 1989.

[13] I. T. R. for Semiconductors. ITRS 1999 Edition. Semiconductor Industry Association, 1999.

[14] I. T. R. for Semiconductors. ITRS 2002 Update. Semiconductor Industry Association, 2002. http://public.itrs.net/Files/2002Update/2002Update.pdf.

[15] J. L. Henning. SPEC CPU2000: Measuring CPU Performance in the New Millennium. *IEEE Computer*, 33(7):28–35, July 2000.

[16] G. Hinton, D. Sager, M. Upton, D. Boggs, D. Carmean, A. Kyker, and P. Roussel. The microarchitecture of the Pentium 4 processor. *Intel Technology Journal*, Feb. 2001.

[17] R. Ho, K. W. Mai, and M. A. Horowitz. The Future of Wires. *Proceedings of the IEEE*, 89(4):490–504, Apr. 2001.

[18] M. S. Hrishikesh, N. P. Jouppi, K. I. Farkas, D. Burger, S. W. Keckler, and P. Shivakumar. The Optimal Logic Depth Per Pipeline Stage is 6 to 8 Inverter Delays. In *Proceedings of the 29th Annual International Symposium on Computer Architecture*, May 2002.

[19] S. Kempainen. LVDS Provides Higher Bit Rates, Lower Power, and Improved Noise Performance. http://www.measurement.tm.agilent.com/insight/2000_v5_i2/insight_v5i2_articl% e01.shtml, 2000.

[20] R. E. Kessler. The Alpha 21264 Microprocessor. *IEEE Micro*, 19(2):24–36, March/April 1999.

[21] R. E. Kessler, R. Jooss, A. Lebeck, and M. D. Hill. Inexpensive Implementations of Set-Associativity. In *Proceedings of the 16th Annual International Symposium on Computer Architecture*, May 1989.

[22] S. P. Khatri and et. al. A Novel VLSI Layout Fabric for Deep Sub-Micron Applications. In *Design Automation Conference*, pages 491–496, June 1999.

[23] C. Kim. Personal Communication, May 2003.

[24] C. Kim, D. Burger, and S. W. Keckler. An Adaptive, Non-Uniform Cache Structure for Wire-Dominated On-Chip Caches. In *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Oct. 2002.

[25] G. K. Konstadinidis and et. al. Implementation of a Third-Generation 1.1-GHz 64-bit Microprocessor. *IEEE Journal of Solid-State Circuits*, 37(11):1461–1469, Nov 2002.

[26] P. S. Magnusson et al. Simics: A Full System Simulation Platform. *IEEE Computer*, 35(2):50–58, Feb. 2002.

[27] C. J. Mauer, M. D. Hill, and D. A. Wood. Full System Timing-First Simulation. In *Proceedings of the 2002 ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, pages 108–116, June 2002.

[28] C. McNairy and D. Soltis. Itanium 2 Processor Microarchitecture. *IEEE Micro*, 23(2):44–55, March/April 2003.

[29] M. Minzuno, K. Anjo, Y. Sumi, M. Fukaishi, H. Wakabayashi, T. Mogami, T. Horiuchi, and M. Yamashina. Clock Distribution Networks with On-Chip Transmission Lines. In *Proceedings of the IEEE 2000 International Interconnect Technology Conference*, pages 3–5, 2000.

[30] R. Nagarajan, K. Sankaralingam, D. Burger, and S. Keckler. A Design Space Evaluation of Grid Processor Architectures. In *Proceedings of the 34th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 40–51, Dec. 2001.

[31] S. Palacharla and J. E. Smith. Complexity-Effective Superscalar Processors. In *Proceedings of the 24th Annual International Symposium on Computer Architecture*, pages 206–218, June 1997.

[32] D. A. Priore. Inductance on Silicon for Sub-micron CMOS VLSI. In *Proceedings of the 1993 Symposium on VLSI Circuits*, pages 17–18, 1993.

[33] M. Racanelli and et. al. Ultra High Speed SiGe NPN for Advanced BiCMOS Technology. *Electron Devices Meeting, IEDM Technical Digest. International*, pages 15.3.1–15.3.4, 2001.

[34] D. Sylvester, W. Jiang, and K. Keutzer. BACPAC - Berkeley Advanced Chip Performance Calculator website. http://www-device.eecs.berkeley.edu/ dennis/bacpac/.

[35] D. Sylvester and K. Keutzer. Getting to the Bottom of Deep Submicron II: a Global Wiring Paradigm. In *Proceedings of the 1999 International Symposium on Physical Design*, pages 193–200, 1999.

[36] Systems Performance Evaluation Cooperation. SPEC Benchmarks. http://www.spec.org.

[37] J. M. Tendler, S. Dodson, S. Fields, H. Le, and B. Sinharoy. POWER4 System Microarchitecture. IBM Server Group Whitepaper, Oct. 2001.

[38] F. F. Tsui. JSP - A Research Signal Processor in Josephson Technology. *IBM Journal of Research and Development*, 24(2):243–252, Mar. 1980.

[39] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik. Orion: A Power-Performance Simulator for Interconnection Networks. In *Proceedings of the 35th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 294–305, Nov. 2002.

[40] J. D. Warnock and et. al. The Circuit and Physical Design of the POWER4 Microprocessor. *IBM Journal of Research and Development*, 46(1):27–51, Jan. 2002.

[41] N. Weste and K. Eshragian. *Principles of CMOS VLSI Design: A Systems Perspective*. Addison-Wesley Publishing Co., 1982.

[42] C.-Y. Wu and M.-C. Shiau. Delay Models and Speed Improvement Techniques for RC Tree Interconnections Among Small-Geometry CMOS Inverters. *IEEE Journal of Solid-State Circuits*, 25(5):1247–1256, Oct 1990.

[43] T. Xanthopoulos, D. W. Bailey, M. K. G. Atul K. Gangwar, A. K. Jain, and B. K. Prewitt. The Design and Analysis of the Clock Distribution Network for a 1.2 GHz Alpha Microprocessor. In *Proceedings of the IEEE 2001 International Solid-State Circuits Conference*, pages 402–403, 2001.