Improved Modeling and Data Migration for Dynamic Non-Uniform Cache Accesses

Christopher Cowell, Csaba Andras Moritz and Wayne Burleson
[ccowell, andras, burleson]@ecs.umass.edu
University of Massachusetts Amherst

Abstract
Growing wire delay and clock rates limit the amount of cache accessible within a single cycle. Non-uniform cache access (NUCA) has been proposed as a solution to this problem in Kim et al, 2002 [1], and performance has been analyzed for various cache organizations and technology assumptions. Innovations included cache organizations which dynamically migrated data between blocks within the cache (D-NUCA) resulting in 11% improvement in SPEC2000 benchmarks over a static (S-NUCA) approach. Our work duplicates, verifies and extends the work of [1] in the following ways: 1) a commercial microprocessor, the Compaq Alpha 21364 is used for a realistic floorplan (an admitted limitation by the authors of [1]), cache sizes and wire delay estimates, 2) process technology nodes 130nm, 90nm and 65nm are used to explore the scaling of the proposed approach, and 3) new topologies and policies are developed for migrating data within the cache. Our results generally corroborate those of [1] and show that the realistic floorplan results in a 16% increased performance. Furthermore, our improved topology and policies for movement of data within the cache result in still improved performance of 43%. It should be noted that there is wide variation in the improvement of the different SPEC2000 benchmarks, thus pointing to future compiler-level approaches to D-NUCA exploitation.

Our work has been supported in part by SRC Task 766 and an SRC/IBM Masters Scholarship.

1. Introduction

Interconnect is a huge 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 cache sizes are increasing and as on-chip cache access require multiple cycles due to wire delay.

1.1 Previous Work

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 [18]. 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. More recent work shows performance improvement in the access latency as cache designs progress from uniform caches to non-uniform caches in Figure 1 [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 to be explored but this research uses an S-NUCA2 configuration as a basis for evaluating the modified D-NUCA cache on an Alpha 21364 for a technology study (130nm, 90nm and 65nm) and a topology study for a torus, mesh and hypercube on-chip interconnection network. The rest of the paper will compare/contrast the architectural components of the D-NUCA systems in Section 2, followed by a description of the simulation environment and methodology in Section 3. The remaining two sections analyze the simulated results and present possible extensions to this work in Sections 4 and 5 respectively. Section 6 is the supplementary Appendix containing graphs and tables.

2. D-NUCA Components Comparison

The section describes the key architecture components that separate a D-NUCA from an S-NUCA2 system followed by the key difference between D-NUCA1 [1] and D-NUCA2 (this paper).

2.1 Data Mapping

Data mapping is the organization of data among the cache banks. The D-NUCA1 system in Figure 2a shows an 8-way set associative cache consisting of 32 banks. Each arrow represents a single way of the entire cache for each mapping scheme. The simple mapping scheme organizes each cache way to a numbered column. This mapping strategy is considered simple because the banks themselves are wired into vertical columns. The shared mapping scheme upgrades the simple mapping scheme by mapping data in such a way to equalize the average access delay for all sets. The four closest banks to the processing core (first rows of column 3,4,5 and 6) are composed of data that maps to each of the 8 cache ways.

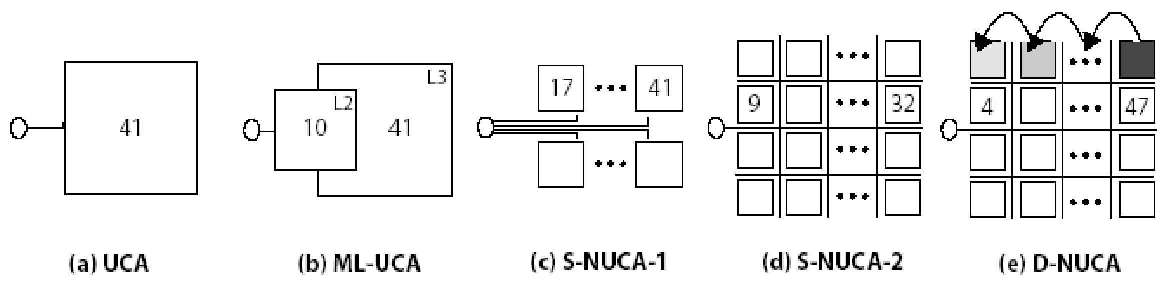

Figure 1. Cache Organizations [1]

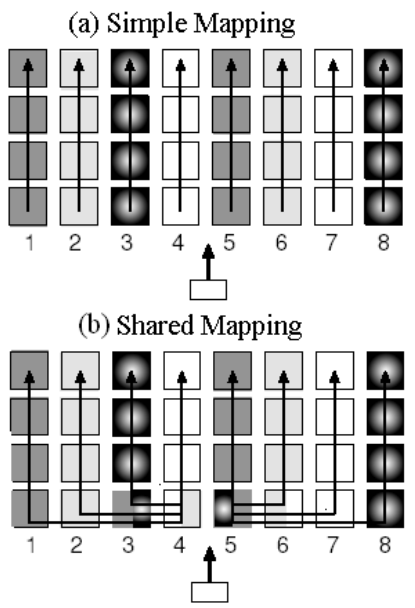

Figure 2a. Mapping Schemes [1]

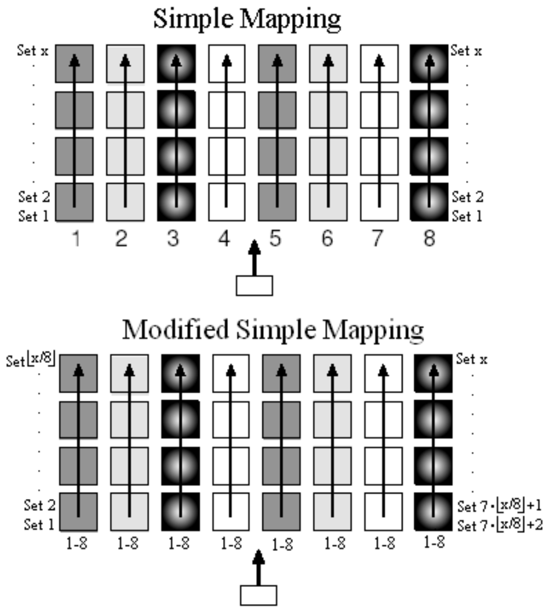

Figure 2b. Modified Simple Scheme

A simple mapping approach is preferred, because the shared mapping requires irregular wiring and data mapping to spliced banks. Because the simple mapping scheme requires little wiring overhead, there are a number of vacant wiring levels to overlay a torus, mesh or hypercube interconnection network. For this research work, data is mapped slightly different by assigning sets to individual banks in Figure 2b. This arrangement allocates all blocks within a set to a bank. Therefore a numbered set always points to a single bank. The D-NUCA2 uses a modified simple mapping scheme to allocate banks to a bank set. Therefore a modified simple approach can assign all the blocks of set 0 through set 8 to a bank versus distributing the blocks of a set amongst the banks in the original simple mapping scheme.

2.2 Bank Search

The D-NUCA1 explored two bank search policies for determining the location of a cache block. The two policies are the incremental and the multicast search. In Figure 3, the incremental search policy checks the closest bank by doing a partial tag search. If the closest bank does not generate a tag match, then a partial tag search is executed on the next closest bank and so on, until either there is a match or a cache miss. The multicast policy performs a partial tag search on all banks within a bank set in parallel. Although the multicast provides faster average access times, checking the banks simultaneously can be sensitive to contention. The multicast search policy showed the best performance and was chosen as the search policy for the D-NUCA2.

2.3 Promotion, Insertion and Eviction Policy

The original D-NUCA cache promotes data incrementally as shown in Figure 5a. As shown, a data request to a far bank triggers a promotion. When accessing a far bank, the promotion occurs when the data transmits to the processing core. If the next closest bank is full then a cache set must be demoted to the next farthest bank, essentially swapping two blocks. Their goal is to minimize the global traffic when data is swapped between two banks, by restricting data movement to neighboring 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 and invalidates the set in the farther bank location. In the event of a promotion and the closest bank is full, a set in the closest bank is demoted to the next farthest bank.

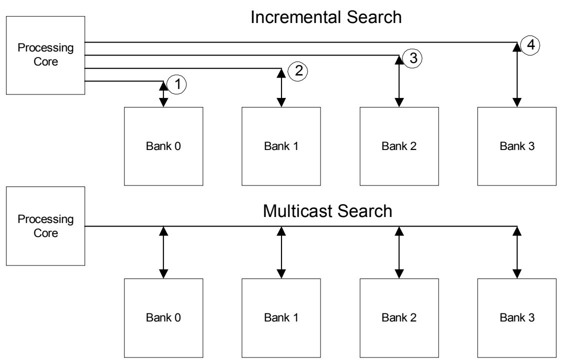

Figure 3. Bank Search Policies

Demotion continues until bank has a vacancy In order to offset the network contention that can occur during the demotion process, the research assumes dual-ported cells, a single port for reading and another for writing data. This will allow reads and writes to occur simultaneously.

The D-NUCA1 explored a number of insertion policies when retrieving new data from the lower level of cache. The best performance occurred when incoming data was inserted into cache banks that is located a moderate distance from the processing core and eviction policy that always evicts from the farthest banks. The D-NUCA2 uses an insertion policy that places new data at the head and an eviction policy that removes data from the farthest cache banks.

To summarize in Table 1, this research attempts to improve upon the D-NUCA model created at UT-Austin by introducing wire latency modeling, a more aggressive promotion policy, and the ability to interconnect banks into a variety of topologies.

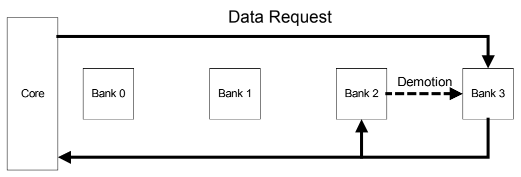

Figure 4a. Incremental Promotion

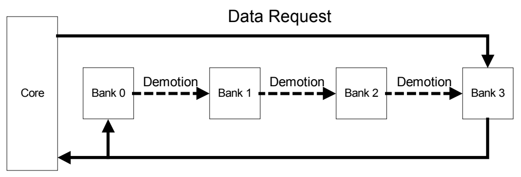

Figure 4b. Absolute Promotion

	D-NUCA1	D-NUCA2
Data Mapping	Shared Mapping	Modified Simple Mapping
Interconnection Scheme	Mesh	User Defined
Search Policy	Incremental or Multicast	Multicast
Insertion Policy	Head, Middle or Tail	Head
Promotion Policy	2-bank/1-hit	All banks/1-hit
Eviction Policy	Tail	Tail

Table 1. Simulator Model Comparison

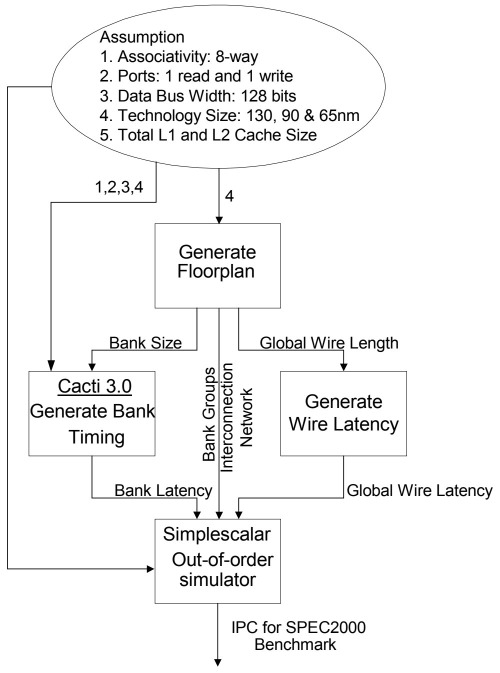
Assumption
1. Associativity: 8-way
2. Ports: 1 read and 1 write
3. Data Bus Width: 128 bits
4. Technology Size: 130, 90 & 65nm
5. Total L1 and L2 Cache Size
1,2,3,4
4
Generate Floorplan
Bank Size
Global Wire Length
Bank Groups Interconnection Network
Cacti 3.0 Generate Bank Timing
Generate Wire Latency
Bank Latency
Global Wire Latency
Simplescalar Out-of-order simulator
IPC for SPEC2000 Benchmark

Figure 5. Simulation Flowchart

3.3 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) pro-jections [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 later 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 0.8cm 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.

This access 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.

3.4 Simplescalar Extended

Simplescalar is an architecture simulator that will model the Alpha 21364. A few modifications were made to Simplescalar to also model a dynamic non-uniform cache system with wire delay support. In the extended version of Simplescalar, the use 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.

```
cache:latency_array{
        0:255:1:511:2:767:3:1023:3:1279:1:
        1535:2:1791:3:2047:4:2303:1:2559:2:2815:3:3
        071:4:3327:1:3583:2:3839:3:4095:4}
```

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 starting with set 0x000 which requires a communication latency of 1 cycles, the next bank can also hold 256 sets but starts with set 256 (or 0x100) with a communication latency of 2 cycles. Since all banks are restricted to the same size, all bank access delays are constant.

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

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

In the event that a node is busy servicing a request, depending on the inter-connection scheme, it is possible to reroute the data to the processing core. The above two parameters define the bank sets and the alternate path for rerouting around a busy node is necessary. For the configuration above, there are 4 defined search paths. When cache is read, a preliminary tag comparison determines which bank set to search and initiates a multicast. In the event that a cache hit occurs and the data collides with a busy node then an alternate route is chosen for the data to travel. The only constraint is that data can only be rerouted between neighboring (point-to-point) cache banks. Therefore in the above *alt_path* parameter, a reroute can be performed between **Bank3** & **Bank7** but not between **Bank3** & **Bank11**.

4. Simulation Results

The section compares the performance of a S-NUCA to a D-NUCA2 system. The research generates statistics for 130nm, 90nm and 65nm, and a topology study of a D-NUCA2 system that supports a torus, mesh and a hypercube.

4.1 Technology Trend

The technology trend uses a 21364 floorplan as the basis for comparing a S-NUCA to a D-NUCA. The Alpha 21364 is a model of the Alpha 21264 with large on-chip L2 caches and multiprocessor support. The results of Table 2 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 point-to-point transmissions. The average IPC improvement was a miniscule 0.25% despite a noticeable miss rate. This implies that pipelined cache access is capable of hiding small multi-cycle delay (less than 3 cycles) within a sizeable cache structures [18].

The Alpha floorplan (90nm) in Figure 8 is a 21364 with 8MB of L2 cache. Figure 8 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 to a 90nm process.

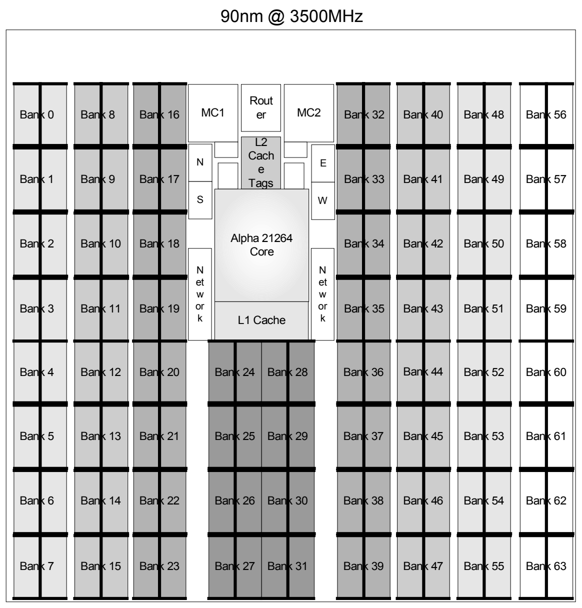

Figure 8. Alpha 21364 Floorplan 90nm

The 90nm version of the Alpha 21364 has 8MB of 64 banks each contain 0.125MB memory modules. The banks are organized in 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 described in Section 3.

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-NUCA 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.

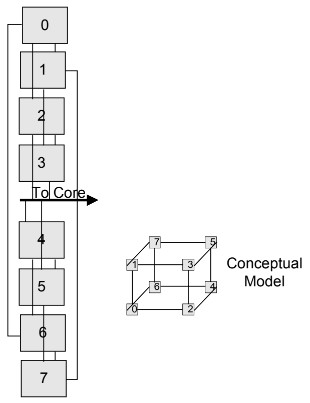

Figure 9. Cubic Interconnection Scheme

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.

4.2 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 11a and b.

5. Conclusion

Non-uniform cache access (NUCA) has been proposed as a solution to this problem of wire dominated cache access in Kim et al, 2002 [1]. Our works attempts to reproduce their research environment on an already existing chip floorplan as well as extend and defend their concept with architectural enhancements and a wire topology study. In general, our results corroborate those of UT-Austin and in using their best-reported configuration were able to boost performance by 43% when using a multi-cube bank interconnection scheme.

The strength of our simulation environment is the extraction of wire delay from an existing floorplan and simulated wire latency using Hspice and ITRS 2002 assumptions. The future work includes improved evaluation techniques that involve much longer simulations, studies that vary the cache bank size for very large on-chip caches. For these results, SPEC2000 was used and a simulation method that simulated the execution of 200 million instructions after fast-forwarding the 100 million instructions. Finally, because contention was an issue for very large caches, varying the cache bank size can extend superb performance improvements down to 30nm where enormous amounts of cache can fit on-chip.

6. Appendix

Benchmark	S-NUCA2	D-NUCA2	Difference %	Miss Rate %
applu	1.0833	1.2742	17.6	3.80
apsi	2.3266	1.5198	-34.7	17.2
fma3d	1.3463	2.2941	70.4	0.13
gcc	0.8118	1.4136	74.1	1.54
lucas	1.1432	1.6427	43.7	0.41
mesa	0.9707	1.9829	104.3	0.14
mgrid	2.0492	1.5003	-26.8	8.91
oammp	1.2196	1.4002	14.8	4.40
oequake	0.963	1.9928	106.9	0.17
oparser	0.985	1.7556	78.2	0.62
ovpr	1.0795	1.7822	65.1	0.36
swim	1.1829	1.7965	51.9	0.50
twolf	0.9711	1.9937	105.3	0.01
wupwise	1.4514	2.0269	39.7	0.90
		Average	**43.0**	**2.79**

Table 3. Alpha 21364 90nm

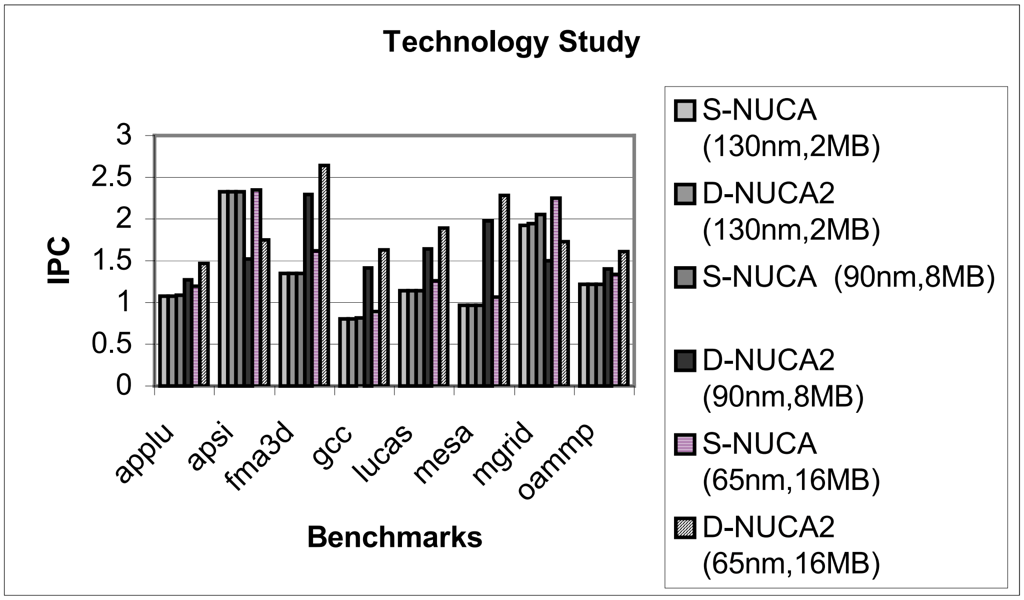

Figure 10a. Technology Trend

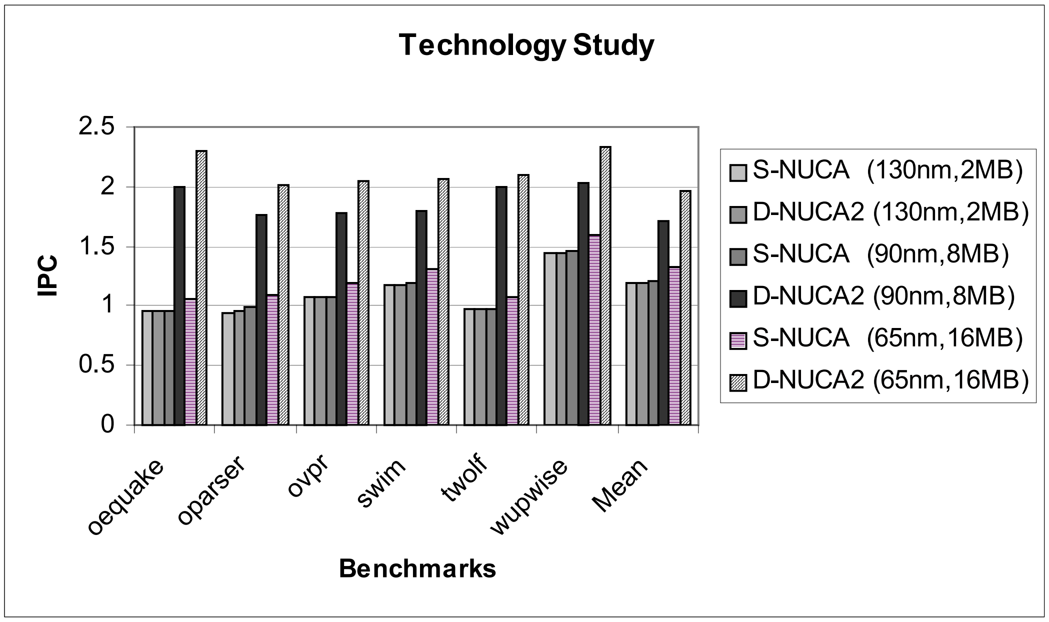

Figure 10b. Technology Trend (continued)

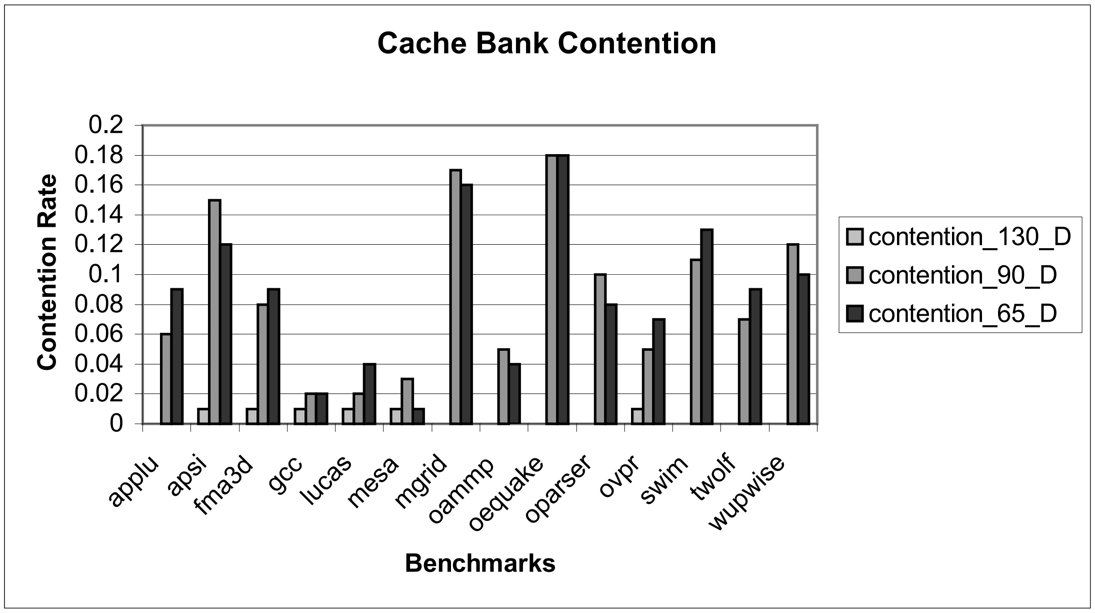

Figure 11. Cache Bank Contention for D-NUCA2

6. References

[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.