The Effect of Technology Scaling on Microarchitectural Structures

Vikas Agarwal* Stephen W. Keckler Doug Burger

Computer Architecture and Technology Laboratory
Department of Computer Sciences
*Department of Electrical and Computer Engineering
Tech Report TR2000-02
The University of Texas at Austin
cart@cs.utexas.edu — www.cs.utexas.edu/users/cart

ABSTRACT

In this report, we describe technology-driven models for wire capacitance, wire delay, and microarchitectural component delay. We used a 3D-field solver (Space3D) to generate our capacitance model based on technology parameters derived from the International Technology Roadmap for Semiconductors (ITRS). This reports shows how we used the ITRS parameters and our capacitance model produce a reasonable set of scaling rules that we then used to modify an existing cache model (ECACTI). Finally the report then shows how this modified model can be used to generate access times for various microarchitectural structures like caches, register files, TLBs and other array like structures. Based on the wire model, it is predicted that it will take 12–32 cycles to traverse the chip in top-level metal under optimistic assumptions about adjoining wires. Based on our component model it is predicted that access a 64KB cache in a 35nm technology with a clock rate of 13.5GHz (as projected by ITRS) will take as many as 7 cycles. To obtain lower latency accesses, designers will have to either reduce the clock rate or use smaller storage structures in such a senario.

1 Introduction

In this technical report, we explore the scalability of microarchitectural structures as technology shrinks from the current 250 nm feature sizes to the projected 35 nm in 2014. With detailed wire and component models, we show that today's designs scale poorly with technology. We show that designers must select among longer latency, smaller structures, or slower clocks. The motivation behind this work is to be able to predict more accurately the performance of future microprocessors by incorporating information about the access time of various on chip structures into microarchitectural simulation.

In Section 2, we describe trends in transistor switching and wire transmission delay based on our analytical wire delay model which is derived from a capacitance model created using a 3D field solver and technology parameters from the 1999 International Technology Roadmap for Semiconductors (ITRS) [1]. Since the latency of a microarchitectural structure also depends on the clock rate, this section also dicusses projected trends in the clock rate of the various technology generations. The effect of the various trends on access latencies is also discussed in this section. Section 3 describes how we use the wire delay model and projected material and thin-film properties to estimate microarchitectural component access times in future technologies. This model is based on a modified version of the ECACTI cache delay analysis tool [2].

In Section 4, we use these models to calculate access time as a function of structure parameters and technology generation. We model most of the major storage oriented components of a microprocessor core, such as caches, register files, and queues. We show that the inherent trade-off between access time and capacity will force designers to limit or even decrease the size of the structures to meet clock rate expectations. For example, our models show that in a 35 nm implementation with a 10 GHz clock, accessing even a 4 KB level-one cache will require 3 clock cycles.

Section 5 describes related work, and Section 6, details the conclusions we draw from our results and describe the implications for future microprocessor designs.

2 Technology Trends

Microprocessor performance improvements have been driven by developments in silicon fabrication technology that have caused transistor sizes to decrease. Reduced feature sizes have provided two benefits. First, since transistors are smaller, more can be placed on a single die, providing area for more complex microarchitectures. Second, technology scaling reduces transistor gate length and hence transistor switching time. If microprocessor cycle times are dominated by gate delay, greater quantities of faster transistors contribute directly to higher performance.

However, faster clock rates and slower wires will limit the number of transistors reachable in a single cycle to be a small fraction of those available on a chip. Reducing the feature sizes has caused wire width and height to decrease, resulting in larger wire resistance due to smaller wire cross-sectional area. Unfortunately, wire capacitance has not decreased proportionally. Even though wire surface area is smaller, the spacing between wires on the same layer is also being reduced. Consequently, the decreased parallel-plate capacitance is offset by increased coupling capacitance to neighboring wires. In this section we describe how we developed an analytical wire delay model based on emprical capacitance results from a 3D field solver. We use the simple first-order models to demonstrate the effect of technology scaling on chip-wide communication delays and clock rate improvements. Since the latency is clock cycles is more relevant than the absolute delay along a segment of wire, we also dicuss trends in the clock rate and the impact clock rate can have on wire latency. We use these models to reason about how designers can expect future microarchitectures

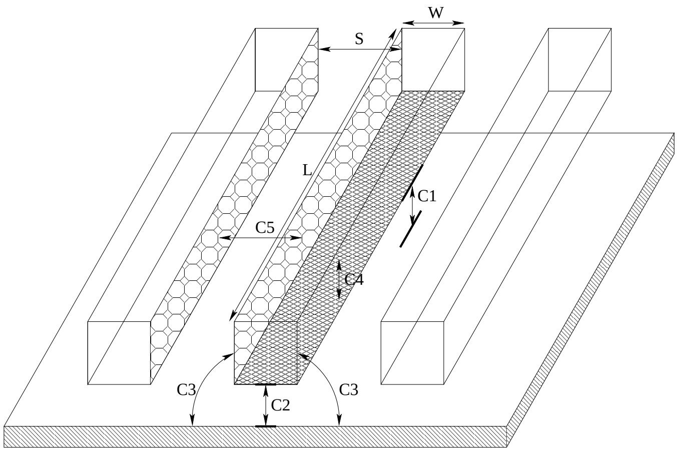

Figure 1: The components of the capacitance in our analytical capacitance model.

to be scaled.

2.1 Analytical Wire Model

Since the delay of a wire is directly proportional to the product of its resistance and capacitance, we developed models for these parameters across all of the technology generations of interest. To compute wire resistance per unit length, $(\Omega/\mu m)$ we use the simple equation, $R_{wire} = \frac{\rho}{W \times H}$, where ρ is wire resistance, W is wire width, and H is wire height. Computing capacitance per unit length $(fF/\mu m)$ is more complicated due to the interactions among multiple conductors. To model the capacitance, we use empirical results obtained from Space3D, a three-dimensional field solver [11]. Wire capacitance includes components for conductors in lower and higher metal layers as well as coupling capacitance to neighboring wires in the same layer. For each fabrication technology, we provided Space3D with the geometry for a given wire with other wires running parallel to it on the same layer and perpendicular on the layers above and below. The layers above and below are assumed to have a dense distribution of wires on them. We vary wire height, width, and spacing in the Space3D input geometries, and use least-mean-squared curve fitting to derive the coefficients for the model. By assuming that all conductors other than the modeled wire are grounded, and thus not accounting for Miller-effect coupling capacitance, our capacitance model is optimistic compared to thei worst-case environment of a wire in a real system.

Our analytical capacitance model used the simple Equation 1 based on Figure 1 to generate the parasitic capacitance for a wire of length L, width W and having spacing S from the adjacent wires.

$$\begin{aligned} C_{wire} &= C_1 + C_2 + C_3 + C_4 + C_5 \\ &= L \times C_L + W \times C_W + S \times C_S + L \times W \times C_{LW} + \frac{L}{S} \times C_{LS} \end{aligned} \tag{1}$$

The first and second terms ($C_1 = L \times C_L$ and $C_2 = W \times C_W$) in this equations refers to the edge capacitance from the edges of the wire to the substrate. The third term ($C_3 = S \times C_S$) represents the increased fringing capacitance to ground as the wires are placed further and further apart. The fourth term ($C_4 = L \times W \times C_{LW}$) is the term representing the area capacitance to ground which goes up as the area of the wire increases. The last term ($C_5 = \frac{L}{S} \times C_{LS}$) represents the coupling capacitance to the adjacent wire, which increases as the length (L) of the wire increases and decreases as the wires a placed further apart (S). Table 1 shows the values of all five terms

Gate (nm)	Level	$C_1(fF)$	$C_2(fF)$	$C_3(fF)$	$C_4(fF)$	$C_5(fF)$
250	mid	0.066	0.030	0.023	0.033	0.051
	top	0.066	0.042	0.032	0.046	0.036
180	mid	0.066	0.022	0.069	0.035	0.140
	top	0.065	0.034	0.063	0.033	0.154
130	mid	0.064	0.019	0.063	0.032	0.158
	top	0.061	0.024	0.062	0.029	0.182
100	mid	0.063	0.009	0.057	0.029	0.174
	top	0.061	0.021	0.054	0.026	0.198
70	mid	0.062	0.003	0.056	0.028	0.183
	top	0.060	0.013	0.058	0.026	0.204
50	mid	0.059	0.000	0.053	0.026	0.204
	top	0.058	0.004	0.058	0.025	0.213
35	mid	0.059	0.000	0.053	0.025	0.211
	top	0.056	0.001	0.052	0.023	0.233

Table 1: Terms in our capacitance model and their sum for a $1\mu m$ wire.

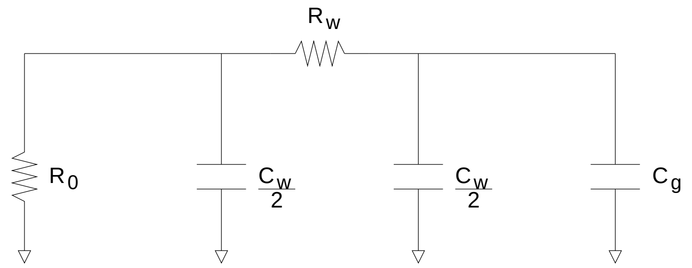

Figure 2: The RC network used to calculate wire delay in Equation 2 for a buffered wire

of our analytical model and the total capacitance for a $1\mu m$ wire in mid and top level metal from technologies from 250nm down to 35nm. As can be seen from the table capacitance per unit length of wire stays roughly constant across the various technology generations. The reason for this is that, as the capacitance to the substrate decreases as the wire dimensions get smaller, the coupling capacitance increases, as adjacent wires are placed closer together.

The derived values for R_{wire} and C_{wire} form the core of our wire delay model. Given the fabrication technology, and the wire length, width, and spacing, our model computes the end-to-end wire transmission delay. For the load on the remote end of the wire, we assume a minimum-size inverter, which has a small gate capacitance relative to the wire capacitance. We assume optimal repeater placement in our model to reduce the delay's dependence on wire length from quadratic to linear. Each repeater is an inverter with PFET and NFET sizes chosen to minimize overall wire delay. We use a π circuit (as shown in figure 2) to model each wire segment in a repeated wire, as described in [12], and calculate the overall delay as a function of wire length L using Equation 2.

$$D_{wire} = \frac{L}{l_0}(R_0(C_g + C_{wire}) + p + R_{wire}(\frac{C_{wire}}{2} + C_g)) \tag{2}$$

R_0 is the on-resistance of the repeater, C_g is the gate capacitance of the repeater, l_0 is the length of a wire segment between repeaters, p is the intrinsic delay of a repeater, and R_{wire} and C_{wire} are the resistance and capacitance of the wire segment between two repeaters. Using this equation, the transmission delay for a 5mm top-level wire more than doubles from 170ps to 390ps over the range of 250nm to 35nm technologies. When possible, increasing the wire width is an attractive strategy for reducing wire delay. Increasing the wire width and spacing by a factor of four for top level

Gate Length (nm)	Dielectric Constant κ	Metal ρ ($\mu\Omega$-cm)	Mid-Level Metal Width (nm)	Aspect Ratio	R_{wire} ($m\Omega/\mu m$)	C_{wire} ($fF/\mu m$)	Top-Level Metal Width (nm)	Aspect Ratio	R_{wire} ($m\Omega/\mu m$)	C_{wire} ($fF/\mu m$)
250	3.9	3.3	500	1.4	107	0.202	700	2.0	34	0.222
180	2.7	2.2	320	2.0	107	0.333	530	2.2	36	0.350
130	2.7	2.2	230	2.2	188	0.336	380	2.5	61	0.359
100	1.6	2.2	170	2.4	316	0.332	280	2.7	103	0.361
70	1.5	1.8	120	2.5	500	0.331	200	2.8	164	0.360
50	1.5	1.8	80	2.7	1020	0.341	140	2.9	321	0.358
35	1.5	1.8	60	2.9	1760	0.348	90	3.0	714	0.366

Table 2: Projected fabrication technology parameters.

metal reduces the delay for a 5mm wire from 390ps to 210ps in a 35nm process, at a cost of four times the wire tracks for each signal. In this study, we assume the wire widths shown in Table 2.

2.2 Wire Scaling

Our source for future technology parameters pertinent to wire delay is the 1999 ITRS [1]. Although the roadmap outlines the targets for future technologies, the parameters described within are not assured. Nonetheless, we assume that the roadmap's aggressive technology scaling predictions (particularly those for conductor resistivity ρ and dielectric permittivity κ) can be met. We also use the roadmap's convention of subdividing the wiring layers into three categories: (1) *local* for connections within a cell, (2) *intermediate*, or mid-level, for connections across a module, and (3) *global*, or top-level, for chip-wide communication. To reduce communication delay, wires are both wider and taller in the mid-level and top-level metal layers. In our study of wire delay, we focus on mid-level and top-level wires, and use the the wire width, height, and spacing projected in the roadmap.

Table 2 displays the wire parameters from 250nm to 35nm technologies, including the derived wire resistance per unit length (R_{wire}) and capacitance per unit length (C_{wire}) for both mid-level and top-level metal layers. The values are based on our analytical model as described in greater detail in Section 3. R_{wire} increases enormously across the technology parameters, with notable discontinuities at the transition to 180nm, due to copper wires, and 70nm, due to an anticipated drop in resistivity from materials improvements projected in the ITRS [1]. However, to limit the effect of shrinking wire width, wire aspect ratio (ratio of wire height to wire width) is predicted to increase up to a maximum of three. Larger aspect ratios increase the coupling capacitance component of C_{wire}, which is somewhat mitigated by reductions in the dielectric constant of the insulator between the wires. Even with the advantages of improved materials, the intrinsic delay of a wire, $R_{wire} \times C_{wire}$, is increasing with every new technology generation. These results are similar to those found in other studies by Horowitz [3] and Sylvester [4].

2.3 Clock Scaling

While wires have slowed down, transistors have been getting dramatically faster. To first order, transistor switching time, and therefore gate delay, is directly proportional to the gate length. In this paper we use the *fanout-of-four* (FO4) delay metric to estimate circuit speeds independent of process technology technologies [3]. The FO4 delay is the time for an inverter to drive four copies of itself. Thus, a given circuit limited by transistor switching speed has the same delay measured in number of FO4 delays, regardless of technology. Reasonable models show that under typical conditions, the FO4 delay, measured in picoseconds (ps) is equal to $360 \times L_{drawn}$, where L_{drawn} is

Gate (nm)	Chip Area (mm^2)	16FO4 Clk f_{16} (GHz)	14FO4 Clk f_{14} (GHz)	12FO4 Clk f_{12} (GHz)	10FO4 Clk f_{10} (GHz)	8FO4 Clk f_8 (GHz)	6FO4 Clk f_6 (GHz)	SIA Clk f_{SIA} (GHz)
250	400	0.69	0.79	0.93	1.11	1.39	1.85	0.75
180	450	0.97	1.10	1.29	1.54	1.93	2.57	1.25
130	567	1.34	1.53	1.78	2.14	2.67	3.56	2.10
100	622	1.74	1.98	2.31	2.78	3.47	4.63	3.50
70	713	2.48	2.83	3.31	3.97	4.96	6.61	6.00
50	817	3.47	3.97	4.63	5.56	6.94	9.26	10.00
35	937	4.96	5.67	6.61	7.94	9.92	13.20	13.50

Table 3: Projected chip area and clock rate.

the minimum gate length for a technology, measured in microns. Using this approximation, the FO4 delay decreases from 90ps in a 250nm technology to 12.6ps in 35nm technology, resulting in circuit speeds improving by a factor of seven, just due to technology scaling.

The FO4 delay metric is important as it provides a fair means to measure processor clock speeds across technologies. The number of FO4 delays per clock period is an indicator of the number of levels of logic between on-chip latches. Microprocessors that have a small number of FO4 delays per clock period are more deeply pipelined than those with more FO4 delays per clock period. As shown by Kunkel and Smith [5], pipelining to arbitrary depth in hopes of increasing the clock rate does not result in higher performance. Overhead for the latches between pipeline stages becomes more significant as the number of levels of logic within a stage decreases too much. Pipelining in a microprocessor is also limited by dependencies between instructions in different pipeline stages. To execute two dependent instructions in consecutive clock cycles, the first instruction must compute its result in a single cycle. With current microarchitectures this requirement can be viewed as a lower bound on the amount of work that can be performed in a useful pipeline stage, and could be represented as the computation of an addition instruction. Under this assumption, a strict lower bound on the clock cycle time is 5.5 FO4 delays, which is the computation delay of a highly optimized 64-bit adder, as described by Naffziger [6]. When accounting for latch overhead and the time to bypass the output of the adder back to the input for the next instruction, reducing the clock period to 8 FO4 delays will present significant design challenges.

In Figure 3, we plot microprocessor clock periods (measured in FO4 delays) from 1992 to 2014. The horizontal lines represent the 8 FO4 and 16 FO4 clock periods. The clock periods projected by the ITRS shrink dramatically over the years and reach 5.6 FO4 delays at 50nm, before increasing slightly to 5.9 FO4 delays at 35nm. The Intel data represent five generations of x86 processors and show the reduction in the number of FO4 delays per pipeline stage from 53 in 1992 (i486DX2) to 15 in 2000 (Pentium III) to 10 in 2001 (Pentium 4), indicating substantially deeper pipelines. The isolated circles represent data from a wider variety of processors published in the proceedings of the International Solid State Circuits Conference (ISSCC) from 1994 to 2000. Both the Intel and ISSCC data demonstrate that clock rate improvements have come from a combination of technology scaling and deeper pipelining, with each improving approximately 15-20% per year. While the trend toward deeper pipelining will continue, reaching eight FO4 delays will be difficult, and attaining the SIA projected clock rate is highly unlikely. In Table 3, we show the resulting clock rates across the spectrum of technologies, assuming varying level of pipelining from six FO4 gate delay per pipeline stage to sixteen FO4 per pipeline stage.

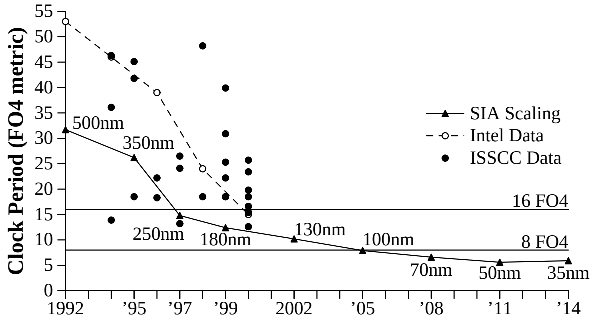

Figure 3: Clock scaling measured in FO4 inverter delays. The aggressive (8 FO4) and conservative (16 FO4) clocks are constant across technologies, but the SIA roadmap projects less than 6 FO4 delays at 50nm and below.

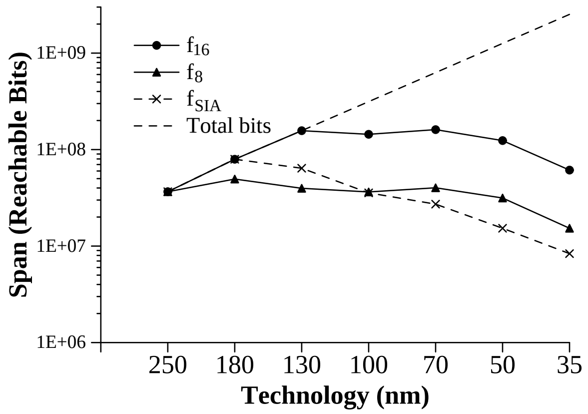

Figure 4: Reachable chip area in top-level metal, where area is measured in six-transistor SRAM cells.

2.4 Wire Delay Impact on Microarchitecture

The widening gap between the relative speeds of gates and wires will have a substantial impact on microarchitectures. With increasing clock rates, the distance that a signal can travel in a single clock cycle decreases. When combined with the modest growth in chip area anticipated for high-performance microprocessors, the time (measured in clock periods) to send a signal across one dimension of the chip will increase dramatically. Our analysis below uses the clock scaling described above and the projected chip areas from the SIA Roadmap, as shown in Table 3.

Based on the wire delay model, we compute the chip area that is reachable in a single clock cycle. Our unit of chip area is the size of a six-transistor SRAM cell, which shrinks as feature size is reduced. To normalize for different feature sizes across the technologies, we measure SRAM cell size in λ, which is equal to one-half the gate length in each technology. We estimate the SRAM cell area to be $700\lambda^2$, which is the median cell area from several recently published SRAM papers [7, 8, 9]. Our area metric does not include overheads found in real SRAM arrays, such as the area required for decoders, power distribution, and sense-amplifiers. Additionally, it does not reflect the size of a single-cycle access memory array; the area metric includes all bits reachable within a one-cycle, one-way transmission delay from a fixed location on the chip, ignoring parasitic capacitance from

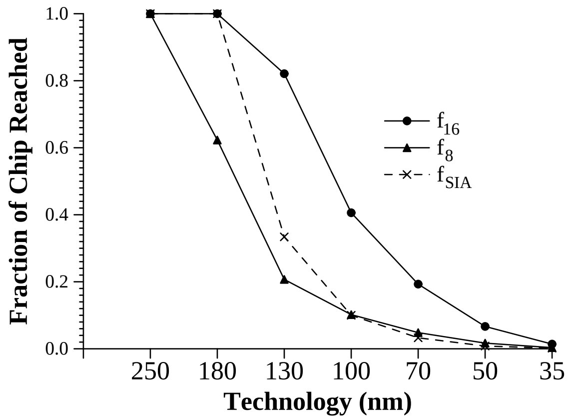

Figure 5: Fraction of total chip area reachable in one cycle.

the SRAM cells.

Figure 4 shows the absolute number of bits that can be reached in a single clock cycle, which we term *span*, using top-level wires for f_{16}, f_8, and f_{SIA} clock scaling. The wire width and spacing is set to the minimum specified in the SIA Roadmap for top-level metal at each technology. Using f_{16} clock scaling, the span first increases as the number of bits on a chip increases and the entire chip can still be reached in a single cycle. As the chip becomes communication bound at 130nm, multiple cycles are required to transmit a signal across its diameter. In this region, decreases in SRAM cell size are offset equally by lower wire transmission velocity, resulting in a constant span. Finally, the span begins to decrease at 50nm when the wire aspect ratio stops increasing and resistance becomes more significant. The results for f_8 are similar except that the plateau occurs at 180nm and the span is a factor of four lower than that of f_{16}. However, in f_{SIA} the span drops steadily after 180nm, because the clock rate is scaled superlinearly with decreasing gate length. These results demonstrate that clock scaling has a significant impact on architectures as it demands a trade-off between the size and partitioning of structures. Using high clock rates to meet performance targets limits the size of pipeline stages and microarchitectural structures, while tightly constraining their placement. If lower clock rates can be tolerated, then microarchitects can give less consideration to the communication delay to reach large and remote structures.

Figure 5 shows the fraction of the total chip area that can be reached in a single clock cycle. Using f_8 in a 35nm technology, less than 0.4% of the chip area can be reached in one cycle. Even with f_{16}, only 1.4% of the chip can be reached in one cycle. Similar results have been observed in prior work [10]. If microarchitectures do not change over time, this phenomenon would be unimportant, since the area required to implement them would decrease with feature size. However, microarchitectures have become more complex because architects acquired more transistors with each new fabrication technology, and used them to improve overall performance. In future technologies, substantial delay penalties must be paid to reach the state or logic in a remote region of the chip, so microarchitectures that rely on large structures and global communication will face more serious challenges in the future than they do today.

2.5 Summary

While transistor speeds are scaling approximately linearly with feature size, wires are getting slower with each new technology. Even assuming low-resistivity conductors, low-permittivity dielectrics, and higher aspect ratios, the *absolute* delay for a fixed-length wire in top-level metal with optimally placed repeaters is increasing with each generation. Only when the wire width and spacing is increased substantially can the wire delay be kept constant. Due to increasing clock frequencies, wire delays are increasing at an even higher rate. As a result, chip performance will no longer be determined solely by the number of transistors that can be fabricated on a single integrated circuit (capacity bound), but instead will depend upon the amount of state and logic that can be reached in a sufficiently small number of clock cycles (communication bound).

The argument made by Sylvester and Keutzer [4] that wire delays will not affect future chip performance holds only if wire lengths are reduced along with gate lengths in future technologies. Traditional microprocessor microarchitectures have grown in complexity with each technology generation, using all of the silicon area for a single monolithic core. Current trends in microarchitectures have increased the sizes of all of the structures, and added more execution units. With future wire delays, structure size will be limited and the time to bypass results between pipeline stages will grow. If clock rates increase at their projected rates, both of these effects will have substantial impact on instruction throughput.

3 ECACTI Analytical Models

In order to study the effect of the various technology trends on the access time of microarchitectural structures, we developed our own analytical access time models. To achieve this, we used our analytical wire delay model to modify an existing cache model (ECACTI). This section explains how our wire model was used to modify ECACTI.

To model the various storage-oriented components of a modern microprocessor, we started with an extended version of the original CACTI cache modeling tool [2], called ECACTI [13]. Given the capacity, block size, associativity, number ports, and number of data and address bits, ECACTI considers a number of alternative cache organizations and computes the minimum access time. ECACTI automatically splits the cache into banks and chooses the number and layout of banks that incurs the lowest delay. When modeling large memory arrays, ECACTI presumes multiple decoders, with each decoder serving a small number of banks. For example with a 4MB array, ECACTI produces 16 banks and four decoders in a 35nm technology. Note that this model is optimistic, because it does not account for driving the address from a central point to each of the distributed decoders.

We extended ECACTI to include technology scaling, using the projected parameters from the ITRS. SRAM cell sizes and transistor parasitics, such as source and drain capacitances, are scaled according to their anticipated reduction in feature size for future technologies. We assume that the word-lines are run from a decoder across its neighboring banks in mid-level metal, and that this in mid-level metal does not affect the size of the SRAM cell. Parameters that are not explicitly specified in the ITRS are assumed to obey simple scaling rules e.g. load capacitance decreases linearly with feature size. Unlike Amrutur and Horowitz [12] we further make the optimistic assumption that the sense-amplifier threshold voltage will decrease linearly with technology, which gives us better sense times than if the threshold voltage does not scale.

The input parameters to ECACTI are the following (in order):

- Number of address bits supplied to the cache
- Number of bits that are output by the cache
- Size of cache in bytes

- Block size in bytes
- Associativity (FA for fully associative)
- Number of read-write ports (optional, default 1)
- Number of extra read ports (optional)
- Number of extra write ports (optional)

An example of a typical ECACTI run is as follows:

```
cacti.25 32 64 65536 32 2 1 0 0
```

The above example is a simulation of a 64KB, 1-ported, 2-way set associative cache in a 250 nm technology. The block size is 32 bytes, while the number of address bits is 32 and the number of output bits is 64. Such a run will produce an output like:

```
Cache Parameters:
  Size in bytes: 65536
  Number of sets: 1024
  Associativity: 2
  Block Size (bytes): 32
  Read/Write Ports: 1
  Read Ports: 0
  Write Ports: 0

Access Time: 1.32017e-09
Cycle Time:  1.8542e-09
Senseext_scale: 0.10

Best Ndwl (L1): 1
Best Ndbl (L1): 4
Best Nspd (L1): 1
Best Ntwl (L1): 1
Best Ntbl (L1): 4
Best Ntspd (L1): 1
Nor inputs (data): 3
Nor inputs (tag): 3
```

```
Time Components:
 data side (with Output driver) (ns): 1.26469
 tag side (ns): 1.1866
 decode_data (ns): 0.377527
 wordline_data (ns): 0.431051
 bitline_data (ns): 0.080696
 sense_amp_data (ns): 0.1798
 senseext_driver (ns): 0.0620447
 decode_tag (ns): 0.377527
 wordline_tag (ns): 0.121566
 bitline_tag (ns): 0.0790588
 sense_amp_tag (ns): 0.0806
 compare (ns): 0.255581
 mux driver (ns): 0.229503
 sel inverter (ns): 0.0427631
 data output driver (ns): 0.133568
 total data path (without output driver) (ns): 1.13112
 total tag path is set assoc (ns): 1.1866
 precharge time (ns): 0.534035
```

Note that we have created a family of CACTI simulators with one for each technology generation from 250nm down to 35nm. The example show above corresponds to using the simulator for the 250nm technology. It we wanted to run the same example in a 180nm technology, we would use cacti.18 instead.

Apart from modeling direct-mapped and set associative caches, we used our extended version of ECACTI to explore other microarchitectural structures. For example, a register file is essentially a direct mapped cache with more ports, but fewer address and data bits than a typical L1 data cache. Content adressable memories (CAMs) are modeled as fully associative structures with the appropriate number of tag and data bits per entry. We use a similar methodology to examine issue windows, reorder buffers, branch prediction tables, and TLBs.

Gate (nm)	16FO4 Clk	14FO4 Clk	12FO4 Clk	10FO4 Clk	8FO4 Clk	6FO4 Clk	SIA Clk
250	2	2	3	3	4	5	2
180	2	2	3	3	4	5	3
130	2	2	3	3	4	5	3
100	2	2	3	3	4	5	4
70	2	2	3	3	4	5	5
50	2	3	3	4	4	6	6
35	3	3	3	4	5	6	6

Table 4: Latency in cycles for a 64KB, 2-way set associative, 2 ported, 32B block cache in various technologies.

4 Microarchitectural Structures

In addition to reducing the chip area reachable in a clock cycle, both the widening gap between wire and gate delays and superlinear clock scaling has a direct impact on the scaling of microarchitectural structures in future microprocessors. Clock scaling is more significant than wire delay for small structures, while both wire delay and clock scaling are significant in larger structures. The large memory-oriented elements, such as the caches, register files, instruction windows, and reorder buffers, will be unable to continue increasing in size while remaining accessible within one clock cycle. In this section, we use the analytical models to examine the access time of different structures from 250nm to 35nm technologies based on the structure organization and capacity. We demonstrate the trade-offs between access time and capacity that are necessary for the various structures across the technology generations.

4.1 Caches

Using our extended ECACTI, we measured the memory structure access time, while varying cache capacity, block size, associativity, number of ports, and process technology. While cache organization characteristics do affect access time, the most critical characteristic is capacity. In Figure 6, we plot the access time versus capacity for a dual-ported, two-way set associative cache. The maximum cache capacities that can be reached in 3 cycles for the f_{16}, f_8 and f_{SIA} clocks are also plotted as "isobars". Note that the capacity for a three cycle access cache decreases moderately for f_{16} and f_8, but falls off the graph for f_{SIA}.

For each technology, the access time increases as the cache capacity increases. Even with substantial banking, the access time goes up dramatically at capacities greater than 256KB. For a given cache capacity, the transition to smaller feature sizes decreases the cache access time, but not as fast as projected increases in clock rates. In a 35nm technology, a 32KB cache takes one to six cycles to access depending on the clock frequency. One alternative to slower clocks or smaller caches is to pipeline cache accesses and allow each access to complete in multiple cycles. Due to the non-linear scaling of capacity with access time, adding a small number of cycles to the cache access time substantially increases the available cache capacity. For example, increasing the access latency from four to seven cycles increases the reachable cache capacity by about a factor of 16 in a 35nm technology. The results shown in Figure 6 apply to all of the cache-like microarchitectural structures that we examine in this study, including L1 instruction and data caches, L2 caches, register files, branch target buffers, and branch prediction tables.

Table 4 shows the latency in clock cycles to access a 64KB, 2-way set associative, 2 ported, 32B block size cache for the various clock rates specified in Table 3. A comprehensive set of access times

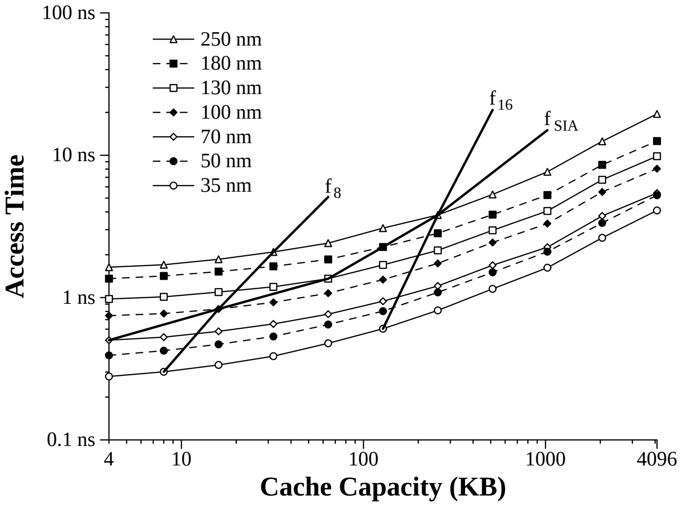

Figure 6: Access time for various L1 data cache capacities.

Gate (nm)	16FO4 Clk	14FO4 Clk	12FO4 Clk	10FO4 Clk	8FO4 Clk	6FO4 Clk	SIA Clk
250	1	2	2	2	2	3	1
180	1	2	2	2	2	3	2
130	1	2	2	2	2	3	2
100	1	2	2	2	2	3	2
70	1	2	2	2	2	3	3
50	2	2	2	2	3	3	3
35	2	2	2	2	3	3	3

Table 5: Latency in cycles for a 128 entry, 10 ported, 80bits per entry register file in various technologies.

for various cache configurations and technology generations is shown in Tables 8 to 21 in Appendix A.

4.2 Register Files

While our cache model replicates current design methodologies, our register file model is more aggressive in design. Although register files have traditionally been built using single-ended full swing bit lines [14], larger capacity register files will need faster access provided by differential bit-lines and low-voltage swing sense-amplifiers similar to those in our model. For our register file modeling, the cache block size is set to the register width and the associativity is set to 1. The main difference between a register file and direct mapped cache is that the register file has a significantly higher number of ports. For a large ported register file, the size of each cell in the register file increases linearly in both dimensions with the number of ports. Also, when modeling a register file, we need to set the number of output bits to match the size of each entry in the register file.

Our capacity results for the register file are similar to those seen in caches. Our results show that register files with many ports will incur larger access times. For example, in a 35nm technology, going from ten ports to 32 ports increases the access time of a 64-entry register file from 172ps to 274ps. Increased physical size and access time makes attaching more execution units to a single global register file impractical. Table 5 shows the latency in clock cycles to access a 128 entry, 10 ported, 80bits per entry register file for the various clock rates specified in Table 3. A comprehensive set of access times for various simple register file configuration from a 250nm technology down to a 35nm technology can be found in Tables 22 to 28 in Appendix B.

4.3 Content Addressable Memories

The final set of components that we model are those that require global address matching within the structure, such as the instruction window and the TLB. These components are typically implemented as content addressable memories (CAMs) and can be modeled as a fully associative cache. Our initial model of the instruction window includes a combination of an eight-bit wide CAM and a 40-bit wide direct mapped data array for the contents of each entry. The issue window has eight ports, which are used to insert four instructions, write back four results, and extract four instructions simultaneously. Since we assume that the eight-ported CAM cell and corresponding eight-ported data array cell are port-limited, we compute the area of these cells based on the width and number of bit-lines and word-lines used to access them. Note that we model only the structure access time and do not consider the latency of the instruction selection logic.

Figure 7 shows the access time for this configuration as a function of the number of instructions in the window. As with all of the memory structures, the access time increases with capacity. The increase in access time is not as significant as in the case of the caches, because the capacities considered are small and all must pay an almost identical penalty for the fully associative match on the tag bits. Thus, in this structure, once the initial tag match delay has been computed, the delay for the rest of the array does not increase significantly with capacity. A 128-entry, eight-port, eight-bit tag instruction window has an access time of 227ps in a 35nm process, while a 12-bit tag raises the access time of a same size window to 229ps. A 128-entry, 32-port, eight-bit tag instruction window (as might be required by a 16-issue processor) has an access time of 259ps in a 35nm technology. Note that all of these results ignore the increase in complexity of the selection logic as we increase the issue window size and the port count in the issue window. We anticipate that the capacity and port count of the register file and the complexity of the selection logic will ultimately place a limit on the issue width of superscalar microarchitectures [15].

A comprehensive set of access times for various simple CAM configuration from a 250nm technology down to a 35nm technology can be found in Tables 29 to 35 in Appendix C.

4.4 Validation

In order to better validate our analytical structure model based on the CACTI code, we compared the access times of various structures as predicted by our code to the latency output by a netlist representing a cross-section of the circuit simulated in CACTI. Using the simple CMOS models obtained by linear scaling, the access times produced by CACTI and the netlist diverged greatly below 100nm feature size. We discovered that with simple linear technology scaling, below 100nm the delay of an inverter simulated in HSPICE did not match expected delay from analytical models. To account for this, the CMOS transistor models were adjusted slightly to achieve a better match between the HSPICE and analtyical models. The parameters in the HSPICE model that were adjusted include the substrate doping, the threshold voltage and the mobility. The access times generated by CACTI and the HSPICE deck for a 2-ported, 2-way associative cache are listed in Table 6. This level of accuracy is comparable to the accuracy of the original CACTI model when compared to equivalent HSPICE results.

We compared our analytical model to other models and related implementations. In a 250nm technology, we compute the access time for a 64KB L1 data cache to be 2.4ns. This access time is

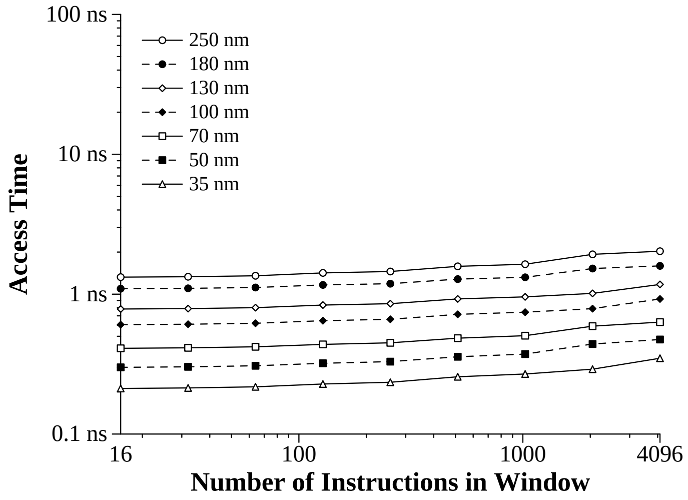

Figure 7: Access time vs. issue window size across technologies.

comparable to that of the 700MHz Alpha 21264 L1 data cache. Furthermore, for a 4MB cache in a 70nm technology, our model predicts an access time of 33 FO4 delays which matches the 33 FO4 access time generated by Amrutur and Horowitz for a similar cache [12].

4.5 Summary

Because of increasing wire delays and faster transistors, memory-oriented microarchitectural structures are not scaling with technology. To access caches, register files, branch prediction tables, and instruction windows in a single cycle will require the capacity of these structures to decrease as clock rates increase. In Table 7, we show the number of cycles needed to access the structures from the Compaq Alpha 21264, scaled to a 35nm process for each of the three methods of clock scaling. With constant structure capacities, the L1 cache will take up to seven cycles to access, depending on how aggressively the clock is scaled.

5 Related Work

Our work on the access time model is based on the original CACTI code that was written by Wilton and Jouppi [2], and then enhanced by Reinman and Jouppi [13]. Our code is based on this enhanced version of the code. The results that we have obtained in term of access time for caches match well with the results that are obtained but Horowitz, et al. [12] in their work on scaling of caches across a range of technologies. Our results support the work done by Palacharla, et al. [15] on the increasing complexity of the issue logic for wider and wider issue machines. Wider issue machines require more ports for the register file and issue window. Increasing the number of ports increases the access time and makes it difficult to design a high clock rate wide issue machine.

6 Conclusion

In this study, we examined the effects of technology scaling on wire delays and clock speeds in CMOS technologies down to 35nm. We found that communication delays will become significant

Technology	Capacity	2-way set associative							
		4	8	16	32	64	128	256	512
250nm	CACTI	1.37	1.43	1.54	1.67	1.85	2.17	2.55	3.30
	HSPICE	1.06	1.10	1.35	1.43	1.69	2.28	2.70	3.52
180nm	CACTI	1.17	1.21	1.29	1.40	1.54	1.72	2.01	2.58
	HSPICE	0.94	0.97	1.14	1.20	1.70	1.81	2.07	2.56
130nm	CACTI	0.83	0.86	0.92	1.00	1.11	1.25	1.49	1.95
	HSPICE	0.67	0.70	0.85	0.90	1.09	1.49	1.73	2.12
100nm	CACTI	0.63	0.66	0.70	0.77	0.85	0.99	1.19	1.60
	HSPICE	0.45	0.49	0.56	0.71	0.77	1.17	1.44	1.86
70nm	CACTI	0.41	0.43	0.47	0.51	0.58	0.67	0.81	1.10
	HSPICE	0.31	0.34	0.34	0.47	0.55	1.00	1.25	1.67
50nm	CACTI	0.31	0.32	0.35	0.49	0.46	0.54	0.68	0.94
	HSPICE	0.23	0.34	0.46	0.62	0.67	1.01	1.22	1.62
35nm	CACTI	0.21	0.23	0.25	0.28	0.33	0.39	0.50	0.70
	HSPICE	0.28	0.29	0.25	0.43	0.47	1.12	1.23	1.57

Table 6: Comparison of access times (ns) generated from CACTI versus access time generated by the SPICE deck.

Structure Name	f_{SIA}	f_8	f_{16}
L1 cache 64K (2 ports)	7	5	3
Integer register file 64 entry (10 ports)	3	2	1
Integer issue window 20 entry (8 ports)	3	2	1
Reorder buffer 64 entry (8 ports)	3	2	1

Table 7: Projected access time (cycles) at 35nm.

for global signals. Even under the best conditions, the latency across the chip in a top-level metal wire will be 12–32 cycles, depending on clock rate. In advanced technologies, the delay (in cycles) of memory oriented structures increases substantially due to increased wire latencies and aggressive clock rates.

While our results predict that existing microarchitectures do not scale with technology, we have in fact been quite generous to potential microprocessor scaling. Our wire performance models conservatively assume very low-permittivity dielectrics, resistivity of pure copper, high aspect ratio wires, and optimally placed repeaters for the smaller technologies. Our models for structure access time further assume a hierarchical decomposition of the array into sub-banks, word-line routing in mid-level metal wires, and cell areas that do not depend on word-line wire width.

Our models show that dense storage structures will become considerably slower relative to projected clock rates, and will adversely affect instruction throughput. While structure access time remains effectively constant with the clock rate up to 70nm technologies, at 50nm and below, wire delays become significant. If clocks are scaled superlinearly relative to decreases in gate length, access times for these structures increase correspondingly. For example, when designing a level-one data cache in a 35nm technology, an engineer will be faced with several unattractive choices. First, the engineer may choose an aggressive target clock rate, and attempt to design a low access penalty cache. At the aggressive SIA projection of 13.5 GHz (which is likely unrealistic), even a single-ported *512 byte* cache will require three cycles to access. Second, the designer may opt for a larger cache with a longer access time. Given our conservative assumptions about cache designs, a 64KB L1 data cache would require at least seven cycles to access at the aggressive clock rate. Finally, the designer may choose a slower clock but a less constrained cache. At 5 GHz (16 FO4 delays), a 32KB cache can be accessed in two cycles. A more complete analysis of the effect these trends have on pipeline depths and configurations can be found in [16].

Acknowledgments

We would like to thank other members of the CART group at UT for their help in the experiments. This research is supported by IBM University Partnership Program awards for Doug Burger and Steve Keckler and the IBM Cooperative Fellowship awarded to Vikas Agarwal.

References

[1] "The international technology roadmap for semiconductors." Semiconductor Industry Association, 1999.

[2] S. J. Wilton and N. P. Jouppi, "An enhanced access and cycle time model for on-chip caches," Tech. Rep. 95/3, Digital Equipment Corporation, Western Research Laboratory, 1995.

[3] 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.

[4] D. Sylvester and K. Keutzer, "Rethinking deep-submicron circuit design," *IEEE Computer*, vol. 32, pp. 25–33, November 1999.

[5] S. R. Kunkel and J. E. Smith, "Optimal pipelining in supercomputers," in *Proceedings of the 13th Annual International Symposium on Computer Architecture*, pp. 404–411, June 1986.

[6] S. Naffziger, "A subnanosecond 0.5μm 64b adder design," in *Digest of Technical Papers, International Solid-State Circuits Conference*, pp. 362–363, February 1996.

[7] G. Braceras, A. Roberts, J. Connor, R. Wistort, T. Frederick, M. Robillard, S. Hall, S. Burns, and M. Graf, "A 940MHz data rate 8Mb CMOS SRAM," in *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 198–199, February 1999.

[8] H. Shimizu, K. Ijitsu, H. Akiyoshi, K. Aoyama, H. Takatsuka, K. Watanabe, R. Nanjo, and Y. Takao, "A 1.4ns access 700MHz 288Kb SRAM macro with expandable architecture," in *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 190–191, 459, February 1999.

[9] C. Zhao, U. Bhattacharya, M. Denham, J. Kolousek, Y. Lu, Y.-G. Ng, N. Nintunze, K. Sarkez, and H. Varadarajan, "An 18Mb, 12.3GB/s cmos pipeline-burst cache SRAM with 1.54Gb/s/pin," in *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 200–201,461, February 1999.

[10] D. Matzke, "Will physical scalability sabotage performance gains?," *IEEE Computer*, vol. 30, pp. 37–39, September 1997.

[11] A. J. van Genderen and N. P. van der Meijs, "Xspace user's manual," Tech. Rep. ET-CAS 96-02, Delft University of Technology, Department of Electrical Engineering, August 1996.

[12] B. Amrutur and M. Horowitz, "Speed and power scaling of SRAMs," *IEEE Journal of Solid State Circuits*, vol. 35, pp. 175–185, February 2000.

[13] G. Reinman and N. Jouppi, "Extensions to cacti," 1999. Unpublished document.

[14] S. Rixner, W. J. Dally, B. Khailany, P. Mattson, U. J. Kapasi, and J. D. Owens, "Register organization for media processing," in *Proceedings of the Sixth International Symposium on High-Performance Computer Architecture*, January 2000.

[15] S. Palacharla, N. P. Jouppi, and J. Smith, "Complexity-effective superscalar processors," in *Proceedings of the 24th Annual International Symposium on Computer Archtecture*, pp. 206–218, June 1997.

[16] M. S. Hrishikesh, S. W. Keckler, and D. Burger, "Impact of technology scaling on instruction execution throughput," Tech. Rep. TR2000-06, Department of Computer Sciences, The University of Texas at Austin, November 2000.

A Cache Access Times

Cache access times for various cache configurations in ns are listed in this appendix. The capacity, associativity, block size and number of ports is varied along with the technology.

Size		Direct mapped Block size (bytes)				2-way set associative Block size (bytes)			
(KB)	Ports	32	64	128	256	32	64	128	256
4	1	1.12	1.17	1.31	1.58	1.37	1.51	1.81	2.38
	2	1.19	1.26	1.45	1.85	1.45	1.63	2.03	2.88
8	1	1.23	1.27	1.36	1.65	1.43	1.55	1.84	2.50
	2	1.33	1.39	1.55	1.98	1.55	1.70	2.15	3.09
16	1	1.36	1.41	1.52	1.75	1.53	1.63	1.89	2.61
	2	1.50	1.58	1.76	2.19	1.71	1.85	2.25	3.34
32	1	1.53	1.59	1.69	1.94	1.67	1.76	2.04	2.80
	2	1.75	1.86	2.03	2.53	1.92	2.09	2.54	3.74
64	1	1.74	1.77	1.90	2.19	1.85	1.96	2.20	3.07
	2	2.08	2.14	2.38	2.96	2.18	2.39	2.97	4.50
128	1	2.07	2.13	2.26	2.57	2.14	2.28	2.58	3.43
	2	2.68	2.76	3.06	3.63	2.77	3.08	3.64	5.47
256	1	2.47	2.54	2.70	3.10	2.55	2.71	3.11	4.08
	2	3.36	3.46	3.79	4.62	3.47	3.80	4.63	6.77
512	1	3.22	3.29	3.45	4.09	3.30	3.47	4.10	5.08
	2	4.94	5.04	5.28	6.66	5.07	5.30	6.68	8.71
1024	1	4.51	4.60	4.80	5.53	4.62	4.81	5.55	6.93
	2	7.18	7.33	7.63	9.25	7.35	7.65	9.27	12.29
2048	1	6.55	6.64	6.87	7.35	6.66	6.89	7.37	9.86
	2	11.95	12.11	12.49	13.28	12.15	12.53	13.32	18.37
4096	1	10.11	10.24	10.53	11.15	10.26	10.56	11.17	14.49
	2	18.71	18.96	19.47	20.52	19.00	19.51	20.56	27.68
8192	1	15.24	15.49	16.09	17.31	15.53	16.13	17.35	19.51
	2	31.42	31.88	32.90	34.99	31.95	32.97	35.06	40.05

Table 8: Cache access time in ns for various cache configurations in a 250nm technology.

Size (KB)	Ports	4-way set associative Block size (bytes) 32	64	128	256	8-way set associative Block size (bytes) 32	64	128	256
4	1	1.53	1.80	2.31	NA	1.87	2.34	NA	NA
	2	1.66	2.03	2.82	NA	2.12	2.87	NA	NA
8	1	1.57	1.82	2.42	3.74	1.90	2.46	3.74	NA
	2	1.73	2.14	3.03	5.79	2.24	3.08	5.79	NA
16	1	1.65	1.88	2.54	4.01	1.95	2.57	4.01	8.63
	2	1.89	2.25	3.31	6.30	2.34	3.33	6.30	15.59
32	1	1.78	2.02	2.72	4.34	2.10	2.76	4.34	9.26
	2	2.09	2.54	3.74	6.97	2.59	3.74	6.97	16.83
64	1	2.00	2.20	3.07	4.82	2.27	3.07	4.82	10.10
	2	2.42	2.97	4.50	7.99	2.97	4.50	7.99	18.56
128	1	2.29	2.58	3.43	5.62	2.63	3.43	5.62	11.32
	2	3.08	3.64	5.47	9.64	3.64	5.47	9.64	21.14
256	1	2.71	3.11	4.08	6.76	3.11	4.08	6.76	13.21
	2	3.80	4.63	6.77	12.13	4.63	6.77	12.13	25.10
512	1	3.47	4.10	5.08	8.51	4.10	5.08	8.51	15.92
	2	5.30	6.68	8.71	16.10	6.68	8.71	16.10	31.06
1024	1	4.81	5.55	6.93	10.80	5.55	6.93	10.80	20.06
	2	7.65	9.27	12.29	20.82	9.27	12.29	20.82	40.42
2048	1	6.89	7.37	9.86	13.84	7.37	9.86	13.84	26.60
	2	12.53	13.32	18.37	27.18	13.32	18.37	27.18	55.28
4096	1	10.56	11.17	14.49	19.54	11.17	14.49	19.54	33.90
	2	19.51	20.56	27.68	39.19	20.56	27.68	39.19	71.61
8192	1	16.13	17.35	19.51	28.61	17.35	19.51	28.61	44.22
	2	32.97	35.06	40.05	59.29	35.06	40.05	59.29	94.31

Table 9: Cache access time in ns for various cache configurations in a 250nm technology.

Size		Direct mapped Block size (bytes)				2-way set associative Block size (bytes)			
(KB)	Ports	32	64	128	256	32	64	128	256
4	1	0.91	0.94	1.04	1.25	1.17	1.28	1.51	1.93
	2	0.96	1.00	1.13	1.42	1.22	1.36	1.64	2.22
8	1	1.01	1.03	1.09	1.29	1.22	1.32	1.54	2.00
	2	1.08	1.11	1.21	1.50	1.30	1.42	1.70	2.34
16	1	1.11	1.14	1.21	1.36	1.30	1.38	1.59	2.06
	2	1.21	1.25	1.38	1.62	1.41	1.52	1.79	2.47
32	1	1.24	1.27	1.36	1.52	1.40	1.47	1.70	2.17
	2	1.37	1.43	1.57	1.87	1.56	1.65	1.97	2.68
64	1	1.40	1.43	1.50	1.72	1.54	1.61	1.80	2.36
	2	1.63	1.67	1.80	2.22	1.75	1.85	2.23	3.05
128	1	1.66	1.70	1.80	1.99	1.72	1.81	2.02	2.63
	2	2.02	2.07	2.26	2.64	2.08	2.27	2.66	3.67
256	1	1.95	1.99	2.13	2.34	2.00	2.14	2.35	3.06
	2	2.52	2.57	2.82	3.24	2.58	2.83	3.25	4.76
512	1	2.53	2.57	2.66	3.04	2.58	2.67	3.06	3.76
	2	3.63	3.68	3.81	4.52	3.70	3.83	4.54	5.97
1024	1	3.39	3.44	3.55	4.06	3.45	3.56	4.07	4.91
	2	5.00	5.07	5.23	6.25	5.09	5.25	6.27	8.01
2048	1	4.94	4.98	5.11	5.37	5.00	5.12	5.39	6.67
	2	8.25	8.34	8.54	8.95	8.37	8.56	8.98	11.51
4096	1	7.18	7.24	7.39	7.72	7.25	7.41	7.74	9.52
	2	12.16	12.29	12.55	13.10	12.32	12.58	13.12	16.86
8192	1	10.66	10.78	11.09	11.75	10.80	11.12	11.78	13.42
	2	20.12	20.34	20.86	21.95	20.38	20.90	21.99	25.31

Table 10: Cache access time in ns for various cache configurations in a 180nm technology.

Size		4-way set associative				8-way set associative			
		Block size (bytes)				Block size (bytes)			
(KB)	Ports	32	64	128	256	32	64	128	256
4	1	1.31	1.50	1.86	NA	1.59	1.92	NA	NA
	2	1.39	1.64	2.16	NA	1.74	2.23	NA	NA
8	1	1.34	1.53	1.93	2.87	1.62	1.99	2.87	NA
	2	1.45	1.69	2.27	4.22	1.80	2.34	4.22	NA
16	1	1.41	1.58	1.99	3.02	1.67	2.04	3.02	6.49
	2	1.56	1.80	2.41	4.50	1.90	2.47	4.50	10.27
32	1	1.50	1.69	2.09	3.20	1.78	2.15	3.20	6.83
	2	1.69	1.97	2.62	4.85	2.08	2.68	4.85	11.67
64	1	1.65	1.80	2.30	3.46	1.89	2.35	3.46	7.26
	2	1.90	2.23	3.04	5.38	2.31	3.06	5.38	12.55
128	1	1.81	2.02	2.57	3.91	2.07	2.64	3.91	7.90
	2	2.27	2.66	3.67	6.26	2.66	3.67	6.26	13.87
256	1	2.14	2.35	3.06	4.54	2.43	3.06	4.54	8.90
	2	2.83	3.25	4.76	7.57	3.25	4.76	7.57	15.91
512	1	2.67	3.06	3.76	5.54	3.06	3.76	5.54	10.33
	2	3.83	4.54	5.97	9.71	4.54	5.97	9.71	18.96
1024	1	3.56	4.07	4.91	7.45	4.07	4.91	7.45	12.51
	2	5.25	6.27	8.01	13.69	6.27	8.01	13.69	23.73
2048	1	5.12	5.39	6.67	9.61	5.39	6.67	9.61	16.16
	2	8.56	8.98	11.51	17.48	8.98	11.51	17.48	31.78
4096	1	7.41	7.74	9.52	12.94	7.74	9.52	12.94	22.50
	2	12.58	13.12	16.86	24.01	13.12	16.86	24.01	46.03
8192	1	11.12	11.78	13.42	18.02	11.78	13.42	18.02	29.14
	2	20.90	21.99	25.31	34.87	21.99	25.31	34.87	58.34

Table 11: Cache access time in ns for various cache configurations in a 180nm technology.

Size		Direct mapped Block size (bytes)				2-way set associative Block size (bytes)			
(KB)	Ports	32	64	128	256	32	64	128	256
4	1	0.66	0.68	0.77	0.94	0.83	0.92	1.09	1.43
	2	0.69	0.73	0.84	1.09	0.87	0.98	1.20	1.68
8	1	0.73	0.75	0.80	0.97	0.87	0.94	1.12	1.49
	2	0.78	0.82	0.90	1.15	0.92	1.01	1.25	1.78
16	1	0.80	0.83	0.89	1.03	0.92	0.99	1.15	1.55
	2	0.88	0.91	1.02	1.25	1.00	1.09	1.33	1.91
32	1	0.89	0.92	1.00	1.15	1.00	1.05	1.23	1.64
	2	1.00	1.05	1.17	1.45	1.11	1.19	1.50	2.12
64	1	1.02	1.05	1.11	1.30	1.11	1.16	1.31	1.81
	2	1.20	1.23	1.34	1.69	1.26	1.35	1.70	2.49
128	1	1.22	1.25	1.33	1.51	1.25	1.34	1.52	2.05
	2	1.49	1.53	1.69	2.06	1.54	1.70	2.06	3.02
256	1	1.44	1.48	1.59	1.79	1.49	1.60	1.80	2.46
	2	1.89	1.94	2.14	2.52	1.95	2.15	2.53	3.85
512	1	1.91	1.94	2.03	2.40	1.95	2.04	2.41	3.00
	2	2.78	2.83	2.96	3.63	2.84	2.97	3.64	4.86
1024	1	2.55	2.60	2.70	3.19	2.61	2.71	3.20	3.91
	2	3.82	3.89	4.05	5.01	3.90	4.06	5.02	6.48
2048	1	3.80	3.85	3.97	4.22	3.86	3.98	4.23	5.49
	2	6.44	6.51	6.71	7.12	6.53	6.72	7.14	9.60
4096	1	5.53	5.59	5.75	6.07	5.61	5.76	6.08	7.82
	2	9.45	9.57	9.83	10.38	9.58	9.84	10.39	13.97
8192	1	8.38	8.50	8.81	9.46	8.52	8.83	9.48	10.89
	2	15.84	16.07	16.60	17.69	16.09	16.62	17.71	20.57

Table 12: Cache access time in ns for various cache configurations in a 130nm technology.

Size (KB)	Ports	4-way set associative Block size (bytes) 32	64	128	256	8-way set associative Block size (bytes) 32	64	128	256
4	1	0.95	1.10	1.39	NA	1.16	1.44	NA	NA
	2	1.01	1.21	1.66	NA	1.29	1.70	NA	NA
8	1	0.96	1.12	1.45	2.31	1.19	1.49	2.31	NA
	2	1.04	1.26	1.78	3.48	1.34	1.80	3.48	NA
16	1	1.01	1.15	1.51	2.45	1.22	1.55	2.45	5.43
	2	1.12	1.34	1.91	3.73	1.41	1.91	3.73	9.14
32	1	1.07	1.22	1.59	2.62	1.30	1.64	2.62	5.76
	2	1.22	1.50	2.12	4.07	1.53	2.12	4.07	9.97
64	1	1.19	1.31	1.77	2.86	1.38	1.81	2.86	6.19
	2	1.38	1.70	2.49	4.57	1.70	2.49	4.57	10.84
128	1	1.34	1.52	2.01	3.27	1.55	2.06	3.27	6.81
	2	1.70	2.06	3.02	5.37	2.06	3.02	5.37	12.12
256	1	1.60	1.80	2.46	3.84	1.81	2.46	3.84	7.77
	2	2.15	2.53	3.85	6.54	2.53	3.85	6.54	14.07
512	1	2.04	2.41	3.00	4.73	2.41	3.00	4.73	9.13
	2	2.97	3.64	4.86	8.43	3.64	4.86	8.43	16.93
1024	1	2.71	3.20	3.91	6.37	3.20	3.91	6.37	11.18
	2	4.06	5.02	6.48	11.59	5.02	6.48	11.59	21.34
2048	1	3.98	4.23	5.49	8.03	4.23	5.49	8.03	14.52
	2	6.72	7.14	9.60	14.74	7.14	9.60	14.74	28.54
4096	1	5.76	6.08	7.82	10.78	6.08	7.82	10.78	20.04
	2	9.84	10.39	13.97	20.10	10.39	13.97	20.10	39.46
8192	1	8.83	9.48	10.89	15.47	9.48	10.89	15.47	25.26
	2	16.62	17.71	20.57	29.94	17.71	20.57	29.94	50.43

Table 13: Cache access time in ns for various cache configurations in a 130nm technology.

Size (KB)	Ports	Direct mapped Block size (bytes) 32	64	128	256	2-way set associative Block size (bytes) 32	64	128	256
4	1	0.52	0.54	0.61	0.76	0.63	0.70	0.84	1.11
	2	0.55	0.57	0.67	0.88	0.66	0.75	0.92	1.38
8	1	0.57	0.59	0.64	0.78	0.66	0.72	0.86	1.16
	2	0.61	0.65	0.71	0.93	0.70	0.77	0.97	1.48
16	1	0.64	0.65	0.70	0.83	0.70	0.76	0.89	1.21
	2	0.69	0.72	0.81	1.01	0.75	0.83	1.03	1.59
32	1	0.70	0.72	0.79	0.92	0.77	0.80	0.96	1.28
	2	0.78	0.82	0.92	1.16	0.84	0.92	1.17	1.77
64	1	0.81	0.83	0.88	1.04	0.85	0.89	1.05	1.43
	2	0.96	0.98	1.07	1.37	0.99	1.07	1.37	2.06
128	1	0.96	0.99	1.05	1.23	0.99	1.06	1.24	1.65
	2	1.19	1.21	1.33	1.67	1.21	1.34	1.68	2.48
256	1	1.16	1.19	1.29	1.45	1.20	1.29	1.45	2.02
	2	1.52	1.56	1.73	2.04	1.57	1.74	2.05	3.19
512	1	1.56	1.59	1.66	1.97	1.60	1.67	1.98	2.50
	2	2.26	2.31	2.43	2.96	2.32	2.44	2.96	4.05
1024	1	2.08	2.12	2.21	2.62	2.13	2.21	2.63	3.24
	2	3.10	3.17	3.30	4.10	3.17	3.31	4.11	5.34
2048	1	3.17	3.20	3.30	3.52	3.21	3.31	3.53	4.61
	2	5.28	5.34	5.51	5.88	5.35	5.52	5.89	7.99
4096	1	4.59	4.64	4.78	5.06	4.65	4.78	5.06	6.56
	2	7.71	7.81	8.04	8.53	7.81	8.05	8.54	11.55
8192	1	7.04	7.15	7.42	7.98	7.16	7.42	7.98	9.27
	2	12.91	13.11	13.58	14.56	13.12	13.59	14.57	17.05

Table 14: Cache access time in ns for various cache configurations in a 100nm technology.

Size (KB)	Ports	4-way set associative Block size (bytes) 32	64	128	256	8-way set associative Block size (bytes) 32	64	128	256
4	1	0.73	0.85	1.08	NA	0.91	1.13	NA	NA
	2	0.77	0.93	1.38	NA	1.00	1.38	NA	NA
8	1	0.74	0.86	1.13	1.95	0.93	1.18	1.95	NA
	2	0.79	0.97	1.48	2.98	1.04	1.48	2.98	NA
16	1	0.78	0.89	1.18	2.07	0.95	1.22	2.07	4.79
	2	0.86	1.04	1.59	3.20	1.11	1.59	3.20	7.83
32	1	0.82	0.96	1.27	2.22	1.01	1.30	2.22	5.08
	2	0.93	1.17	1.77	3.49	1.22	1.77	3.49	8.84
64	1	0.91	1.05	1.43	2.43	1.08	1.44	2.43	5.46
	2	1.07	1.37	2.06	3.92	1.37	2.06	3.92	9.61
128	1	1.06	1.24	1.65	2.77	1.24	1.65	2.77	6.02
	2	1.34	1.68	2.48	4.58	1.68	2.48	4.58	10.72
256	1	1.29	1.45	2.02	3.25	1.45	2.02	3.25	6.86
	2	1.74	2.05	3.19	5.55	2.05	3.19	5.55	12.40
512	1	1.67	1.98	2.50	4.00	1.98	2.50	4.00	8.04
	2	2.44	2.96	4.05	7.07	2.96	4.05	7.07	14.83
1024	1	2.21	2.63	3.24	5.35	2.63	3.24	5.35	9.81
	2	3.31	4.11	5.34	9.73	4.11	5.34	9.73	18.48
2048	1	3.31	3.53	4.61	6.95	3.53	4.61	6.95	12.63
	2	5.52	5.89	7.99	12.60	5.89	7.99	12.60	24.31
4096	1	4.78	5.06	6.56	9.23	5.06	6.56	9.23	17.33
	2	8.05	8.54	11.55	16.87	8.54	11.55	16.87	34.10
8192	1	7.42	7.98	9.27	13.30	7.98	9.27	13.30	22.48
	2	13.59	14.57	17.05	25.15	14.57	17.05	25.15	43.76

Table 15: Cache access time in ns for various cache configurations in a 100nm technology.

Size (KB)	Ports	Direct mapped Block size (bytes) 32	64	128	256	2-way set associative Block size (bytes) 32	64	128	256
4	1	0.37	0.39	0.45	0.56	0.41	0.46	0.57	0.79
	2	0.39	0.41	0.49	0.66	0.43	0.50	0.66	1.04
8	1	0.40	0.42	0.46	0.58	0.43	0.47	0.59	0.84
	2	0.43	0.46	0.52	0.70	0.46	0.53	0.70	1.13
16	1	0.44	0.46	0.51	0.62	0.47	0.51	0.62	0.89
	2	0.48	0.51	0.58	0.76	0.51	0.58	0.77	1.23
32	1	0.49	0.51	0.56	0.68	0.51	0.56	0.68	0.97
	2	0.54	0.57	0.65	0.88	0.58	0.65	0.88	1.38
64	1	0.56	0.58	0.63	0.75	0.58	0.63	0.75	1.11
	2	0.66	0.68	0.76	0.99	0.68	0.76	0.99	1.63
128	1	0.65	0.68	0.74	0.87	0.68	0.74	0.88	1.28
	2	0.80	0.83	0.94	1.20	0.83	0.94	1.20	1.99
256	1	0.78	0.81	0.89	1.04	0.81	0.89	1.04	1.47
	2	1.03	1.07	1.20	1.49	1.07	1.21	1.49	2.34
512	1	1.05	1.08	1.15	1.41	1.08	1.15	1.41	1.86
	2	1.50	1.55	1.68	2.12	1.56	1.69	2.12	3.06
1024	1	1.38	1.42	1.50	1.83	1.42	1.50	1.84	2.38
	2	2.06	2.12	2.26	2.92	2.13	2.26	2.93	4.00
2048	1	2.10	2.14	2.24	2.45	2.14	2.24	2.45	3.50
	2	3.48	3.57	3.74	4.14	3.57	3.75	4.14	6.05
4096	1	2.99	3.04	3.17	3.45	3.05	3.18	3.45	4.89
	2	5.08	5.19	5.42	5.90	5.19	5.42	5.90	8.78
8192	1	4.57	4.68	4.95	5.49	4.69	4.95	5.49	6.58
	2	8.49	8.70	9.16	10.12	8.70	9.17	10.12	12.31

Table 16: Cache access time in ns for various cache configurations in a 70nm technology.

Size (KB)	Ports	4-way set associative Block size (bytes) 32	64	128	256	8-way set associative Block size (bytes) 32	64	128	256
4	1	0.48	0.58	0.79	NA	0.62	0.80	NA	NA
	2	0.51	0.66	1.04	NA	0.71	1.04	NA	NA
8	1	0.49	0.60	0.84	1.48	0.63	0.84	1.48	NA
	2	0.53	0.70	1.13	2.27	0.74	1.13	2.27	NA
16	1	0.51	0.62	0.89	1.59	0.64	0.89	1.59	3.64
	2	0.58	0.77	1.23	2.47	0.80	1.23	2.47	6.33
32	1	0.56	0.68	0.97	1.72	0.68	0.97	1.72	3.91
	2	0.65	0.88	1.38	2.74	0.88	1.38	2.74	6.85
64	1	0.63	0.75	1.11	1.92	0.75	1.11	1.92	4.28
	2	0.76	0.99	1.63	3.13	0.99	1.63	3.13	7.59
128	1	0.74	0.88	1.28	2.22	0.88	1.28	2.22	4.80
	2	0.94	1.20	1.99	3.73	1.20	1.99	3.73	8.65
256	1	0.89	1.04	1.47	2.65	1.04	1.47	2.65	5.58
	2	1.21	1.49	2.34	4.60	1.49	2.34	4.60	10.21
512	1	1.15	1.41	1.86	3.30	1.41	1.86	3.30	6.68
	2	1.69	2.12	3.06	5.93	2.12	3.06	5.93	12.47
1024	1	1.50	1.84	2.38	4.09	1.84	2.38	4.09	8.30
	2	2.26	2.93	4.00	7.42	2.93	4.00	7.42	15.81
2048	1	2.24	2.45	3.50	5.30	2.45	3.50	5.30	10.80
	2	3.75	4.14	6.05	9.78	4.14	6.05	9.78	20.98
4096	1	3.18	3.45	4.89	6.98	3.45	4.89	6.98	13.47
	2	5.42	5.90	8.78	13.01	5.90	8.78	13.01	26.25
8192	1	4.95	5.49	6.58	10.50	5.49	6.58	10.50	17.65
	2	9.17	10.12	12.31	20.11	10.12	12.31	20.11	34.70

Table 17: Cache access time in ns for various cache configurations in a 70nm technology.

Size (KB)	Ports	Direct mapped Block size (bytes) 32	64	128	256	2-way set associative Block size (bytes) 32	64	128	256
4	1	0.27	0.29	0.34	0.44	0.31	0.35	0.45	0.68
	2	0.29	0.32	0.39	0.55	0.32	0.39	0.56	0.96
8	1	0.30	0.32	0.36	0.47	0.32	0.36	0.47	0.72
	2	0.32	0.35	0.42	0.60	0.35	0.42	0.60	1.05
16	1	0.33	0.35	0.39	0.50	0.35	0.39	0.50	0.78
	2	0.37	0.40	0.47	0.67	0.40	0.47	0.67	1.16
32	1	0.37	0.39	0.43	0.57	0.39	0.44	0.57	0.86
	2	0.42	0.45	0.53	0.78	0.46	0.53	0.78	1.32
64	1	0.44	0.45	0.50	0.63	0.46	0.50	0.63	1.00
	2	0.53	0.56	0.64	0.89	0.56	0.65	0.89	1.58
128	1	0.52	0.54	0.60	0.74	0.54	0.60	0.74	1.19
	2	0.66	0.68	0.80	1.08	0.68	0.80	1.09	1.97
256	1	0.65	0.67	0.75	0.91	0.68	0.75	0.91	1.39
	2	0.89	0.93	1.09	1.39	0.94	1.09	1.40	2.36
512	1	0.90	0.94	1.02	1.24	0.94	1.02	1.24	1.77
	2	1.29	1.34	1.50	1.95	1.34	1.51	1.95	3.06
1024	1	1.21	1.25	1.33	1.70	1.25	1.33	1.70	2.28
	2	1.90	1.96	2.10	2.86	1.96	2.10	2.86	4.04
2048	1	1.93	1.98	2.09	2.35	1.98	2.09	2.35	3.31
	2	3.07	3.16	3.34	4.18	3.16	3.35	4.18	5.91
4096	1	2.78	2.84	2.97	3.25	2.84	2.98	3.25	4.81
	2	4.87	4.99	5.23	5.74	4.99	5.24	5.74	8.90
8192	1	4.54	4.66	4.92	5.48	4.66	4.93	5.48	6.69
	2	8.68	8.91	9.41	10.17	8.91	9.41	10.18	12.92

Table 18: Cache access time in ns for various cache configurations in a 50nm technology.

Size		4-way set associative Block size (bytes)				8-way set associative Block size (bytes)			
(KB)	Ports	32	64	128	256	32	64	128	256
4	1	0.36	0.45	0.68	NA	0.49	0.68	NA	NA
	2	0.41	0.56	0.96	NA	0.57	0.96	NA	NA
8	1	0.37	0.47	0.72	1.39	0.51	0.72	1.39	NA
	2	0.43	0.60	1.05	2.30	0.61	1.05	2.30	NA
16	1	0.39	0.50	0.78	1.50	0.53	0.78	1.50	3.71
	2	0.47	0.67	1.16	2.52	0.67	1.16	2.52	6.83
32	1	0.44	0.57	0.86	1.65	0.57	0.86	1.65	4.00
	2	0.53	0.78	1.32	2.81	0.78	1.32	2.81	7.41
64	1	0.50	0.63	1.00	1.86	0.63	1.00	1.86	4.40
	2	0.65	0.89	1.58	3.24	0.89	1.58	3.24	8.22
128	1	0.60	0.74	1.19	2.19	0.74	1.19	2.19	4.98
	2	0.80	1.09	1.97	3.90	1.09	1.97	3.90	9.39
256	1	0.75	0.91	1.39	2.65	0.91	1.39	2.65	5.83
	2	1.09	1.40	2.36	4.85	1.40	2.36	4.85	11.11
512	1	1.02	1.24	1.77	3.34	1.24	1.77	3.34	7.04
	2	1.51	1.95	3.06	6.29	1.95	3.06	6.29	13.60
1024	1	1.33	1.70	2.28	4.19	1.70	2.28	4.19	8.81
	2	2.10	2.86	4.04	7.94	2.86	4.04	7.94	17.29
2048	1	2.09	2.35	3.31	5.43	2.35	3.31	5.43	11.50
	2	3.35	4.18	5.91	10.31	4.18	5.91	10.31	22.94
4096	1	2.98	3.25	4.81	7.13	3.25	4.81	7.13	14.48
	2	5.24	5.74	8.90	13.71	5.74	8.90	13.71	28.98
8192	1	4.93	5.48	6.69	10.84	5.48	6.69	10.84	19.01
	2	9.41	10.18	12.92	20.69	10.18	12.92	20.69	37.89

Table 19: Cache access time in ns for various cache configurations in a 50nm technology.

Size (KB)	Ports	Direct mapped Block size (bytes) 32	64	128	256	2-way set associative Block size (bytes) 32	64	128	256
4	1	0.19	0.20	0.24	0.31	0.21	0.24	0.31	0.48
	2	0.20	0.22	0.28	0.40	0.23	0.28	0.40	0.70
8	1	0.21	0.22	0.25	0.33	0.23	0.25	0.33	0.52
	2	0.23	0.25	0.30	0.44	0.25	0.30	0.44	0.77
16	1	0.23	0.25	0.28	0.36	0.25	0.28	0.36	0.57
	2	0.26	0.29	0.34	0.49	0.29	0.34	0.49	0.87
32	1	0.26	0.28	0.31	0.41	0.28	0.31	0.41	0.64
	2	0.30	0.33	0.39	0.58	0.33	0.39	0.58	1.01
64	1	0.32	0.32	0.36	0.46	0.33	0.36	0.46	0.75
	2	0.39	0.41	0.48	0.66	0.41	0.48	0.66	1.23
128	1	0.38	0.39	0.44	0.54	0.39	0.44	0.54	0.89
	2	0.49	0.51	0.60	0.82	0.51	0.60	0.82	1.52
256	1	0.48	0.49	0.55	0.68	0.50	0.55	0.68	1.03
	2	0.67	0.70	0.81	1.07	0.70	0.81	1.07	1.79
512	1	0.67	0.70	0.76	0.94	0.70	0.76	0.94	1.34
	2	0.98	1.03	1.15	1.52	1.03	1.15	1.53	2.36
1024	1	0.91	0.94	1.00	1.27	0.94	1.00	1.28	1.75
	2	1.46	1.51	1.62	2.18	1.51	1.62	2.18	3.17
2048	1	1.47	1.51	1.60	1.79	1.51	1.60	1.79	2.58
	2	2.41	2.48	2.63	3.23	2.48	2.63	3.24	4.72
4096	1	2.12	2.17	2.28	2.51	2.17	2.28	2.51	3.73
	2	3.80	3.89	4.09	4.51	3.89	4.10	4.51	6.98
8192	1	3.50	3.59	3.81	4.26	3.59	3.81	4.26	5.15
	2	6.81	6.99	7.40	8.14	7.00	7.40	8.14	10.07

Table 20: Cache access time in ns for various cache configurations in a 35nm technology.

Size		4-way set associative				8-way set associative			
		Block size (bytes)				Block size (bytes)			
(KB)	Ports	32	64	128	256	32	64	128	256
4	1	0.25	0.32	0.48	NA	0.35	0.48	NA	NA
	2	0.29	0.40	0.70	NA	0.41	0.70	NA	NA
8	1	0.26	0.33	0.52	1.01	0.36	0.52	1.01	NA
	2	0.30	0.44	0.77	1.71	0.44	0.77	1.71	NA
16	1	0.28	0.36	0.57	1.10	0.37	0.57	1.10	2.72
	2	0.34	0.49	0.87	1.89	0.49	0.87	1.89	5.07
32	1	0.31	0.41	0.64	1.23	0.41	0.64	1.23	2.97
	2	0.39	0.58	1.01	2.14	0.58	1.01	2.14	5.56
64	1	0.36	0.46	0.75	1.41	0.46	0.75	1.41	3.31
	2	0.48	0.66	1.23	2.51	0.66	1.23	2.51	6.25
128	1	0.44	0.54	0.89	1.68	0.54	0.89	1.68	3.80
	2	0.60	0.82	1.52	3.06	0.82	1.52	3.06	7.24
256	1	0.55	0.68	1.03	2.07	0.68	1.03	2.07	4.52
	2	0.81	1.07	1.79	3.86	1.07	1.79	3.86	8.70
512	1	0.76	0.94	1.34	2.65	0.94	1.34	2.65	5.54
	2	1.15	1.53	2.36	5.04	1.53	2.36	5.04	10.80
1024	1	1.00	1.28	1.75	3.17	1.28	1.75	3.17	7.03
	2	1.62	2.18	3.17	6.06	2.18	3.17	6.06	13.92
2048	1	1.60	1.79	2.58	4.18	1.79	2.58	4.18	9.21
	2	2.63	3.24	4.72	8.03	3.24	4.72	8.03	18.48
4096	1	2.28	2.51	3.73	5.58	2.51	3.73	5.58	11.00
	2	4.10	4.51	6.98	10.86	4.51	6.98	10.86	22.15
8192	1	3.81	4.26	5.15	8.60	4.26	5.15	8.60	14.70
	2	7.40	8.14	10.07	16.64	8.14	10.07	16.64	29.56

Table 21: Cache access time in ns for various cache configurations in a 35nm technology.

B Register File Access Time

Register File access times for various register files are listed in this appendix. The capacity, entry size and number of ports is varied along with the technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	0.80	0.86	0.94	1.02	0.87	0.93	1.03	1.13	0.89	0.96	1.06	1.15
64	0.83	0.91	1.01	1.13	0.92	1.00	1.11	1.22	0.94	1.04	1.15	1.27
128	0.91	1.00	1.14	1.28	1.00	1.11	1.27	1.44	1.03	1.15	1.31	1.48
256	1.00	1.12	1.30	1.49	1.09	1.22	1.43	1.66	1.12	1.27	1.49	1.73

Table 22: Register file access time in ns for various configurations in a 250nm technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	0.68	0.71	0.77	0.82	0.71	0.77	0.82	0.87	0.74	0.78	0.83	0.90
64	0.70	0.75	0.81	0.87	0.75	0.80	0.87	0.95	0.76	0.82	0.91	0.98
128	0.75	0.81	0.90	0.99	0.81	0.87	0.97	1.07	0.83	0.90	1.01	1.13
256	0.82	0.90	1.02	1.13	0.88	0.98	1.11	1.25	0.90	1.00	1.15	1.30

Table 23: Register file access time in ns for various configurations in a 180nm technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	0.48	0.50	0.53	0.57	0.51	0.54	0.57	0.61	0.52	0.55	0.59	0.64
64	0.49	0.52	0.56	0.60	0.53	0.56	0.61	0.67	0.54	0.58	0.64	0.70
128	0.53	0.57	0.62	0.68	0.57	0.62	0.69	0.76	0.59	0.65	0.72	0.80
256	0.58	0.63	0.71	0.78	0.63	0.69	0.78	0.88	0.65	0.72	0.82	0.93

Table 24: Register file access time in ns for various configurations in a 130nm technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	0.36	0.37	0.39	0.41	0.39	0.41	0.43	0.46	0.40	0.42	0.45	0.48
64	0.37	0.39	0.42	0.45	0.41	0.43	0.46	0.50	0.42	0.45	0.49	0.53
128	0.41	0.43	0.47	0.50	0.45	0.48	0.52	0.57	0.46	0.50	0.55	0.60
256	0.45	0.48	0.53	0.58	0.49	0.54	0.60	0.66	0.51	0.56	0.63	0.70

Table 25: Register file access time in ns for various configurations in a 100nm technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	0.24	0.25	0.26	0.27	0.27	0.28	0.30	0.32	0.28	0.29	0.31	0.33
64	0.25	0.27	0.28	0.30	0.28	0.30	0.32	0.35	0.29	0.31	0.34	0.37
128	0.28	0.30	0.32	0.34	0.31	0.33	0.37	0.40	0.32	0.35	0.38	0.42
256	0.31	0.33	0.36	0.39	0.35	0.37	0.41	0.46	0.36	0.39	0.44	0.49

Table 26: Register file access time in ns for various configurations in a 70nm technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	0.17	0.18	0.20	0.21	0.20	0.21	0.23	0.24	0.21	0.22	0.24	0.25
64	0.18	0.20	0.21	0.22	0.21	0.22	0.24	0.26	0.21	0.23	0.25	0.28
128	0.21	0.22	0.24	0.25	0.23	0.25	0.28	0.31	0.24	0.26	0.29	0.33
256	0.23	0.25	0.27	0.30	0.26	0.28	0.32	0.36	0.27	0.30	0.34	0.39

Table 27: Register file access time in ns for various configurations in a 50nm technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	0.12	0.13	0.14	0.15	0.14	0.15	0.16	0.17	0.14	0.15	0.17	0.18
64	0.13	0.14	0.15	0.16	0.14	0.16	0.17	0.19	0.15	0.16	0.18	0.20
128	0.14	0.15	0.17	0.18	0.16	0.18	0.20	0.22	0.17	0.19	0.21	0.24
256	0.16	0.17	0.19	0.21	0.18	0.20	0.23	0.26	0.19	0.21	0.25	0.29

Table 28: Register file access time in ns for various configurations in a 35nm technology.

C Content Addressable Memory Access Time

Content Addressable Memory (CAM) access times for various CAMs are listed in this appendix. The capacity, entry size and number of ports is varied along with the technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	1.39	1.41	1.44	1.46	1.46	1.49	1.53	1.58	1.48	1.52	1.57	1.62
64	1.43	1.42	1.45	1.48	1.47	1.50	1.55	1.59	1.50	1.53	1.58	1.64
128	1.44	1.48	1.52	1.51	1.52	1.56	1.62	1.62	1.54	1.59	1.66	1.67
256	1.47	1.50	1.55	1.59	1.55	1.59	1.65	1.72	1.57	1.62	1.69	1.77
512	1.55	1.61	1.69	1.65	1.63	1.70	1.80	1.77	1.65	1.73	1.85	1.83
1024	1.60	1.66	1.74	1.83	1.68	1.75	1.86	1.97	1.71	1.79	1.90	2.03
2048	1.77	1.90	1.84	1.93	1.85	2.00	1.96	2.08	1.88	2.04	2.01	2.14
4096	1.86	1.99	2.18	2.37	1.95	2.10	2.31	2.54	1.98	2.14	2.37	2.62
8192	2.29	2.63	3.10	3.59	2.39	2.75	3.26	3.80	2.42	2.80	3.33	3.89

Table 29: CAM time in ns for various configurations in a 250nm technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	1.13	1.15	1.17	1.19	1.19	1.21	1.24	1.27	1.22	1.24	1.27	1.31
64	1.14	1.16	1.18	1.20	1.21	1.23	1.25	1.28	1.23	1.25	1.29	1.32
128	1.18	1.20	1.23	1.22	1.24	1.27	1.31	1.31	1.26	1.29	1.34	1.34
256	1.20	1.22	1.25	1.28	1.26	1.29	1.33	1.37	1.28	1.32	1.36	1.41
512	1.26	1.30	1.35	1.31	1.33	1.37	1.44	1.41	1.35	1.40	1.47	1.45
1024	1.29	1.33	1.39	1.44	1.36	1.41	1.47	1.54	1.39	1.44	1.51	1.58
2048	1.43	1.51	1.45	1.50	1.50	1.59	1.54	1.61	1.52	1.62	1.58	1.66
4096	1.49	1.57	1.68	1.80	1.56	1.66	1.78	1.92	1.59	1.69	1.83	1.97
8192	1.81	2.02	2.31	2.60	1.89	2.11	2.42	2.74	1.92	2.15	2.47	2.80

Table 30: CAM time in ns for various configurations in a 180nm technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	0.81	0.82	0.84	0.85	0.86	0.87	0.89	0.91	0.87	0.89	0.91	0.94
64	0.82	0.83	0.84	0.86	0.87	0.88	0.90	0.92	0.88	0.90	0.92	0.95
128	0.85	0.86	0.88	0.88	0.89	0.91	0.94	0.94	0.91	0.93	0.96	0.97
256	0.86	0.88	0.90	0.92	0.91	0.93	0.96	0.99	0.93	0.95	0.98	1.02
512	0.91	0.94	0.98	0.95	0.96	0.99	1.04	1.02	0.97	1.01	1.06	1.05
1024	0.94	0.97	1.00	1.04	0.99	1.02	1.07	1.12	1.00	1.04	1.10	1.15
2048	1.04	1.10	1.06	1.10	1.09	1.16	1.13	1.18	1.11	1.18	1.16	1.22
4096	1.09	1.15	1.24	1.33	1.14	1.22	1.32	1.43	1.16	1.24	1.35	1.47
8192	1.34	1.51	1.75	1.99	1.40	1.58	1.83	2.09	1.42	1.61	1.87	2.14

Table 31: CAM time in ns for various configurations in a 130nm technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	0.63	0.63	0.64	0.65	0.66	0.67	0.69	0.70	0.68	0.69	0.70	0.72
64	0.64	0.64	0.65	0.66	0.67	0.68	0.69	0.71	0.69	0.70	0.71	0.73
128	0.65	0.66	0.68	0.67	0.69	0.71	0.72	0.72	0.70	0.72	0.74	0.74
256	0.67	0.68	0.69	0.71	0.70	0.72	0.74	0.76	0.72	0.73	0.76	0.78
512	0.70	0.73	0.76	0.73	0.74	0.77	0.80	0.79	0.76	0.78	0.82	0.81
1024	0.73	0.75	0.78	0.81	0.77	0.79	0.83	0.87	0.78	0.81	0.85	0.89
2048	0.81	0.86	0.82	0.85	0.85	0.91	0.88	0.92	0.87	0.93	0.90	0.95
4096	0.85	0.90	0.98	1.05	0.89	0.96	1.04	1.12	0.91	0.97	1.06	1.15
8192	1.07	1.22	1.43	1.63	1.12	1.28	1.49	1.71	1.13	1.30	1.52	1.75

Table 32: CAM time in ns for various configurations in a 100nm technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	0.43	0.43	0.44	0.44	0.45	0.46	0.47	0.48	0.46	0.47	0.48	0.49
64	0.44	0.44	0.44	0.45	0.46	0.47	0.47	0.48	0.47	0.48	0.49	0.50
128	0.44	0.45	0.46	0.47	0.47	0.48	0.49	0.51	0.48	0.49	0.51	0.52
256	0.47	0.46	0.47	0.48	0.49	0.49	0.50	0.52	0.50	0.50	0.52	0.54
512	0.47	0.49	0.51	0.50	0.50	0.52	0.55	0.54	0.51	0.53	0.56	0.56
1024	0.49	0.51	0.53	0.55	0.52	0.54	0.57	0.59	0.53	0.55	0.58	0.61
2048	0.54	0.58	0.57	0.59	0.57	0.61	0.61	0.64	0.59	0.63	0.62	0.66
4096	0.58	0.62	0.67	0.72	0.61	0.65	0.71	0.78	0.62	0.67	0.73	0.80
8192	0.72	0.82	0.96	1.10	0.75	0.86	1.01	1.16	0.76	0.87	1.03	1.18

Table 33: CAM time in ns for various configurations in a 70nm technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	0.31	0.31	0.32	0.32	0.33	0.34	0.34	0.35	0.34	0.35	0.36	0.37
64	0.32	0.32	0.32	0.33	0.34	0.34	0.35	0.36	0.35	0.35	0.36	0.37
128	0.32	0.33	0.34	0.34	0.34	0.35	0.36	0.38	0.35	0.36	0.38	0.39
256	0.33	0.34	0.34	0.35	0.35	0.36	0.37	0.39	0.36	0.37	0.38	0.40
512	0.35	0.36	0.38	0.37	0.37	0.38	0.40	0.40	0.38	0.39	0.42	0.42
1024	0.36	0.37	0.39	0.41	0.38	0.40	0.42	0.45	0.39	0.41	0.44	0.46
2048	0.40	0.43	0.42	0.44	0.43	0.46	0.46	0.49	0.43	0.47	0.47	0.50
4096	0.43	0.46	0.50	0.55	0.45	0.49	0.54	0.60	0.46	0.50	0.56	0.62
8192	0.54	0.62	0.73	0.84	0.57	0.65	0.77	0.90	0.58	0.67	0.79	0.92

Table 34: CAM time in ns for various configurations in a 50nm technology.

Number of Entries	32 bit entries No. of Ports				64 bit entries No. of Ports				80 bit entries No of. Ports			
	3	6	10	14	3	6	10	14	3	6	10	14
32	0.22	0.22	0.23	0.23	0.23	0.24	0.24	0.25	0.24	0.24	0.25	0.26
64	0.22	0.22	0.23	0.23	0.24	0.24	0.25	0.25	0.24	0.25	0.25	0.26
128	0.23	0.23	0.24	0.24	0.24	0.25	0.26	0.26	0.25	0.26	0.27	0.27
256	0.23	0.24	0.25	0.25	0.25	0.26	0.27	0.28	0.25	0.26	0.27	0.29
512	0.25	0.26	0.26	0.26	0.26	0.27	0.28	0.29	0.27	0.28	0.30	0.30
1024	0.26	0.27	0.28	0.30	0.27	0.29	0.31	0.33	0.28	0.29	0.32	0.34
2048	0.29	0.31	0.30	0.32	0.31	0.33	0.33	0.36	0.31	0.34	0.34	0.37
4096	0.31	0.33	0.37	0.41	0.33	0.36	0.40	0.45	0.33	0.37	0.41	0.46
8192	0.40	0.46	0.56	0.65	0.42	0.49	0.59	0.69	0.42	0.50	0.61	0.71

Table 35: CAM time in ns for various configurations in a 35nm technology.