Energy/Power Breakdown of Pipelined Nanometer Caches (90nm/65nm/45nm/32nm)

Samuel Rodriguez and Bruce Jacob
Electrical and Computer Engineering Department
University of Maryland, College Park
{samvr,blj}@eng.umd.edu

ABSTRACT

As transistors continue to scale down into the nanometer regime, device leakage currents are becoming the dominant cause of power dissipation in nanometer caches, making it essential to model these leakage effects properly. Moreover, typical microprocessor caches are pipelined to keep up with the speed of the processor, and the effects of pipelining overhead need to be properly accounted for.

In this paper, we present a detailed study of pipelined nanometer caches with detailed energy/power dissipation breakdowns showing where and how the power is dissipated within a nanometer cache. We explore a three-dimensional pipelined cache design space that includes cache size (16kB to 512kB), cache associativity (direct-mapped to 16-way) and process technology (90nm, 65nm, 45nm and 32nm).

Among our findings, we show that cache bitline leakage is increasingly becoming the dominant cause of power dissipation in nanometer technology nodes. We show that subthreshold leakage is the main cause of static power dissipation, and that gate leakage is, surprisingly, not a significant contributor to total cache power, even for 32nm caches. We also show that accounting for cache pipelining overhead is necessary, as power dissipated by the pipeline elements is a significant part of cache power.

Categories and Subject Descriptors

B.3.2 [Memory Structures] : Cache memories.

General Terms

Design, Performance.

Keywords

Cache design, nanometer design, pipelined caches.

1. INTRODUCTION

Power dissipation has become a top priority for today's microprocessors. What was previously a concern mainly for mobile devices has also become of paramount importance for general-purpose and even high-performance microprocessors, especially with the recent industry emphasis on processor "Performance-per-Watt."

As transistors become steadily smaller with improving fabrication process technologies, device leakage currents are expected to significantly contribute to processor power dissipation [7]. Cache power dissipation has typically been significant, but increasing static power will have a greater impact on the cache since most of its transistors are inactive (dissipating no dynamic power, only static) during any given access. It is therefore essential to properly account for these leakage effects during the cache design process.

ISLPED'06, October 4–6, 2006, Tegernsee, Germany.

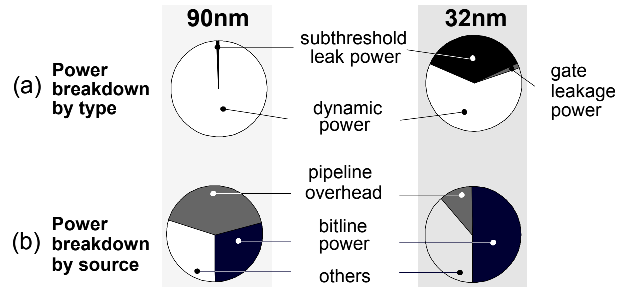

Figure 1: Power breakdowns of a 64kB-4way cache. (a) Dynamic and static power dissipation components for a 64kB-4W cache in 90nm and 32nm, (b) Major components of power dissipation for a 64kB-4way 90nm and 32nm pipelined cache

Moreover, microprocessor caches are obviously not designed to exist in a vacuum – they exist to complement the processor by hiding the relatively long latencies of the lower level memory hierarchy. As such, typical caches (specifically level-1 caches and often level-2) are pipelined and clocked at the same frequency as the core. Explicit pipelining of the cache will involve a non-trivial increase in access-time (because of added flop delays) and power dissipation (from the latch elements and the resulting additional clock power), and these effects must be accounted for properly, something which is not currently done by existing publicly available cache-design tools.

In this paper, we use analytical modeling of cache operation combined with nanometer BSIM3v3/BSIM4 SPICE models [5,13] to analyze the behavior of various cache configurations. We break down cache energy/power dissipation to show how much energy/power each individual part of a cache consumes, and what fraction can be attributed to dynamic (switching) and static (subthreshold leakage and gate tunneling) currents. We explore a three-dimensional cache design space by studying caches with different sizes (16kB to 512kB), associativities (direct-mapped to 16-way) and process technologies (90nm, 65nm, 45nm and 32nm).

Among our findings, we show that cache bitline leakage is increasingly becoming the dominant cause of power dissipation in nanometer technology nodes. We show that subthreshold leakage is the main cause of static power dissipation, but surprisingly, gate leakage tunneling currents do not become a significant contributor to total cache power even for the deep nanometer nodes. We also show that accounting for cache pipelining overhead is necessary, as power dissipated by the pipeline elements are a significant part of cache power.

2. BACKGROUND

2.1 Leakage

Dynamic power is dissipated by a circuit whenever transistors switch to change the voltage in a particular node. In the process, energy is consumed by charging (or discharging) the node's parasitic capacitive loads and, to a lesser degree, by possible short-circuit currents that flow during the small but finite time window when the transistor pullup and pulldown networks are fully or partially turned "on," providing a low-impedance path from supply to ground.

On the other hand, static power is dissipated by leakage currents that flow even when the device is inactive. Different leakage mecha-

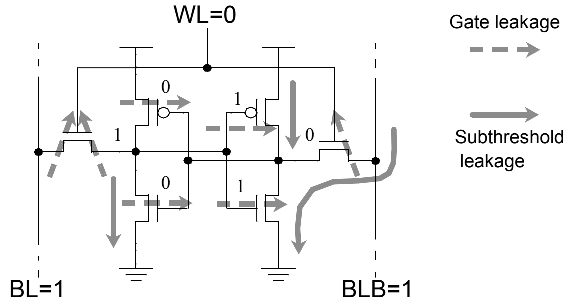

Figure 2: Memory cell leakage currents. An inactive six-transistor memory cell (6TMC) showing the subthreshold leakage and gate leakage currents flowing across the devices

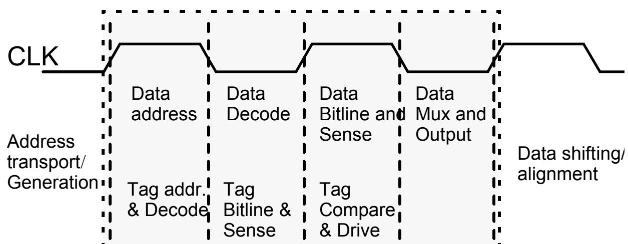

Figure 3: Cache pipeline diagram. The shaded region shows the part of the pipeline that we model

nisms exist for MOS transistors [2], but the two most important ones are lumped into the subthreshold leakage current and the gate leakage current. The subthreshold leakage current has been extensively studied [2,3,6,18,22]. It is mainly caused by the generational reduction in the transistor threshold voltage to compensate for device speed loss when scaling down the supply voltage, with the consequence of exponentially increasing subthreshold leakage current. The gate leakage current has been less extensively studied because of its relatively smaller value compared to subthreshold leakage for older technologies, but is increasingly receiving more attention [10,17,18,23] as it is expected to be comparable to the subthreshold leakage current in the deep nanometer nodes. Gate leakage currents flow whenever voltages are applied to transistor terminals, producing an electric field across gate oxides that are getting thinner (20 to less than 10 angstroms) resulting in significant leakage currents due to quantum tunneling effects.

Figure 2 shows a typical 6-transistor memory cell (6TMC) and the typical leakage currents involved for the memory cell idle state (i.e. wordline is off, one storage node is "0" and the other is "1"). Although a sizeable number of transistors in a cache are active for any given access, the vast majority of memory cells are in this inactive state, dissipating static power. Cache designers currently account for subthreshold leakage current (most problematic of which is the bitline leakage since this not only affects power, but also circuit timing and hence functionality), but not much attention is given to gate leakage in the publicly available cache design tools like CACTI [25,19,20] and eCACTI [14]. For better accuracy, this paper also considers the gate leakage currents as shown in the 6TMC diagram.

2.2. Pipelined Caches

To keep up with the speed of a fast microprocessor core while providing sufficiently large storage capacities, caches are pipelined to subdivide the various delays in the cache into different stages, allowing each individual stage to fit into the core's small clock period. Figure 3 shows a typical pipeline diagram for a cache. The given timing diagram shows operations being performed in both phases of the clock. Figure 4 shows a possible implementation of a pipeline latch that easily facilitates this phase-based operation.

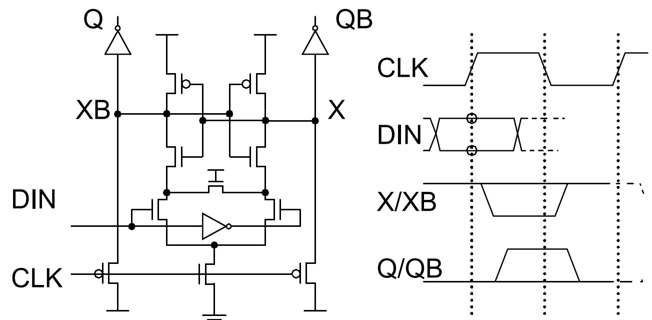

Figure 4: Pipeline latch. An example of a pipeline latch that can be used to implement the phase-based operations in the cache [8][15]. Also shown is the latch's timing diagram

The major publicly-available cache analysis tools CACTI and eCACTI use implicit pipelining through wave pipelining, relying on regularity of the delay of the different cache stages to separate signals continuously being shoved through the cache instead of using explicit pipeline state elements. Unfortunately, this is not representative of modern designs, since cache wave-pipelining is not being used by contemporary microprocessors [21,24]. Although wave pipelining has been shown to work in silicon prototypes [4], it is not ideally suited for high-speed microprocessor caches targeted for volume production which have to operate with significant process-voltage-temperature (PVT) variations. PVT variations in a wave-pipelined cache cause delay imbalances which, in the worst case, lead to signal races that are not possible to fix by lowering the clock frequency. Hence, the risk for non-functional silicon is increased, resulting in unattractive yields. In addition, wave-pipelining does not inherently support latch-based design-for-test (DFT) techniques that are critical in the debug and test of a microprocessor, reducing yields even further. On the contrary, it is easy to integrate DFT "scan" elements inside pipeline latches that allow their state to be either observed or controlled (preferably both). This ability facilitates debugging of microprocessor circuitry resulting in reduced test times that directly translate to significant cost savings. Although not immediately obvious at first, these reasons make it virtually necessary to implement explicit pipelining for high-volume microprocessors caches.

3. EXPERIMENTAL METHODOLOGY

Although initially based on CACTI and eCACTI tools, the analysis tool we have developed bears little resemblance to its predecesors. In its current form, it is, to the best of our knowledge, the most detailed and most realistic (i.e. similar to realistically implementable high-performance caches) cache design space explorer publicly available.

The main improvements of our tool compared to CACTI/eCACTI are the following:

- More optimal decode topologies [1] and circuits [16] along with more realistic device sizing ensures that cache inputs present a reasonable load to the preceeding pipeline stage (at most four times the load of a 1X strength inverter).[1]
- Accurate modeling of explicit cache pipelining to account for delay and energy overhead in pipelined caches.
- Use of BSIM3v3/BSIM4 SPICE models and equations to perform calculation of transistor drive strengths, transistor RC parasitics, subthreshold leakage and gate leakage. Simulation characteristics for each tech node are now more accurate.[2]
- More accurate RC interconnect parasitics by using local, intermediate and global interconnects (as opposed to using the

1. Both CACTI and eCACTI produce designs with impractically large first-stage inverters at the cache inputs. This shifts some of the burden of driving the cache decode hierarchy to the circuits preceding the cache, resulting in overly optimistic delay and power numbers.

same wire characteristics for all interconnects, as is being done by CACTI and eCACTI), and accurate analytical modeling of these structures. In addition, a realistic BEOL-stack[3] was used for each technology node.

An important note to make when discussing dynamic and static power and/or energy is that it is only possible to combine the two into a single measurement by assuming a specific frequency. Dynamic power inherently describes how much energy is consumed in a single switching event, and an assumption of how often that event happens is necessary to convert the energy value into power dissipation (hence the activity factor and frequency components in standard power equations). On the other hand, static leakage inherently describes the amount of current flow, and hence the instantaneous power, at any given time. Converting this into energy requires an assumption of the amount of time the current is flowing. Table 1 shows the values for frequency and supply voltages that were used for this study, where the values are chosen from historical [26] and projected [11,12] data.

2. For delay and dynamic power computations, CACTI and eCACTI use hardcoded numbers based on 0.80um technology and use linear scaling to translate power and delay numbers to the desired technology.
3. The Back-End-Of-Line stack refers to the fabrication steps performed after the creation of the active components. Use of a realistic BEOL-stack results in more accurate modeling of the interconnect. In CACTI and eCACTI, a single interconnect characteristic was assumed.

Table I. VDD and frequency used for each tech node (T=50 C)

	250nm	180nm	130nm	90nm	65nm	45nm	32nm
VDD	2.0 V	1.8 V	1.6 V	1.4 V	1.3 V	1.2 V	1.1 V
Freq. (GHz)	0.8 GHz	1.5GHz	1.8GHz	2.4GHz	2.6GHz	2.8GHz	3.0GHz

4. RESULTS AND DISCUSSIONS

4.1. Dynamic and static power

Figure 5 shows the power dissipation of the different cache configurations as a function of process technology. Each column of plots represents a specific process technology, while each row represents a specific cache size. Each plot shows the dynamic, subthreshold leakage, gate leakage, and total power as a function of associativity for the given cache size and technology node.

The most basic observation here is that total power is dominated by the dynamic power in the larger technology nodes but is dominated by static power in the deep nanometer nodes (with the exception of very highly-associative small to medium caches).

This can be seen from the plots for the 90nm and 65nm nodes, where the dynamic power comprises the majority of the total power. In the 45nm node, the subthreshold leakage power is significant

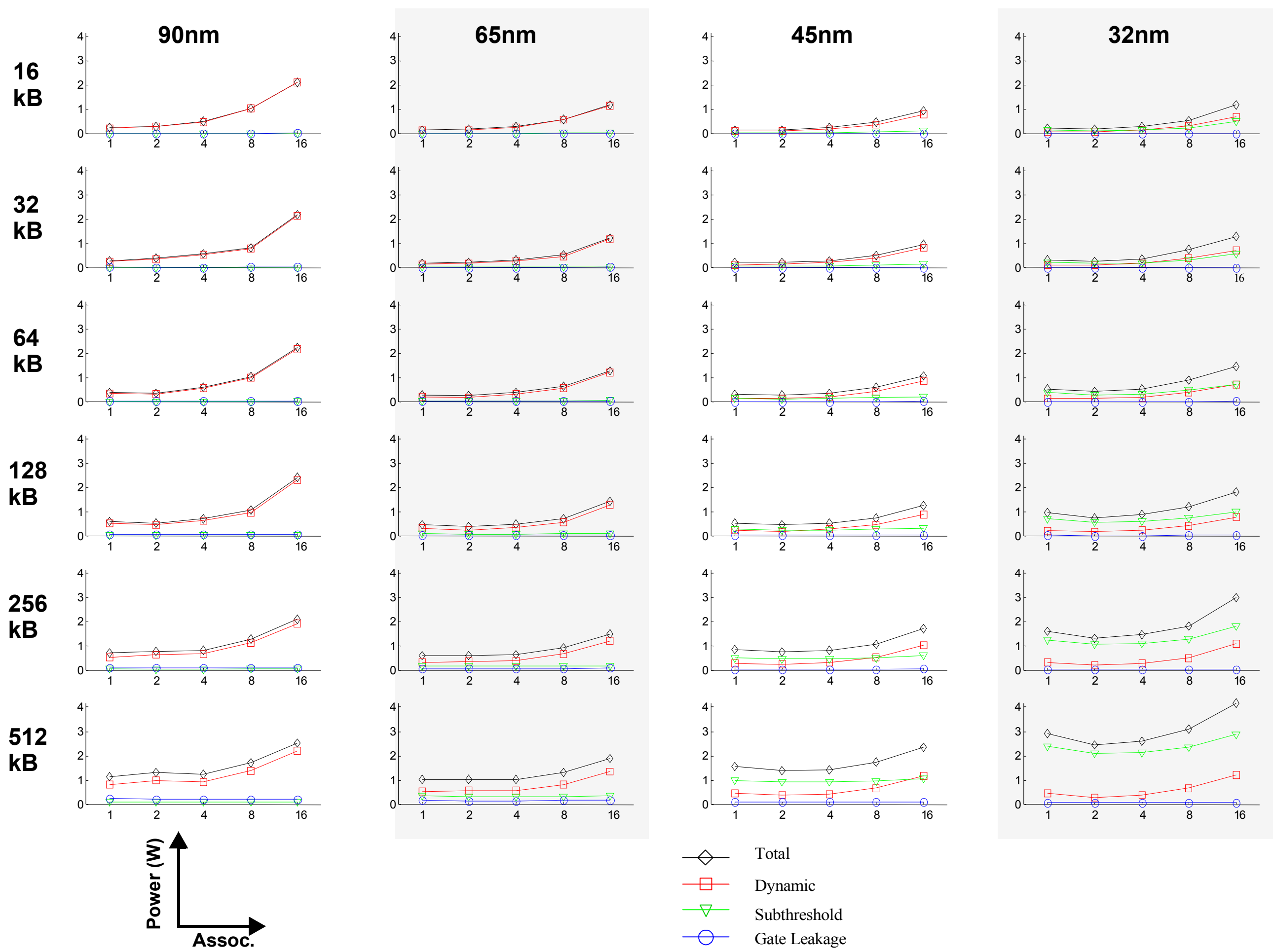

Figure 5: Power consumption vs. technology node of different cache configurations. The plots show how total power consumption is broken down into dynamic power and static power (due to subthreshold leakage and gate leakage). Each column of plots represents a single tech node, while a single row represents a specific cache size. Within each plot, the power dissipation is shown in the y-axis, with increasing associativitesrepresented in the x-axis.

enough that it becomes the dominant component for some configurations. For small caches of any associativity, dynamic power typically dominates because there are fewer leaking devices that contribute to subthreshold leakage power. But as cache sizes increase, the number of idle transistors that dissipate subthreshold leakage power also increases, making the subthreshold leakage power the dominant component of total power except for the configurations with high associativity (which requires more operations to be done in parallel, resulting in dissipation of more dynamic power). In the 32nm node, the subthreshold leakage power is already comparable to the dynamic power even for small caches (of any associativity), and starts to become really dominant as cache sizes increase (even at high associativities where we expect the cache to burn more dynamic power).

A surprising result that can be seen across all the plots is the relatively small contribution of gate leakage power to the total power. This is surprising given all the attention that gate leakage current is receiving and how it is expected to be one of the dominant leakage mechanisms. It turns out that this is a result of many factors, including the less aggressive gate oxide thickness scaling in the more recent ITRS roadmaps [11,12], and that Vdd scaling and device size scaling tend to decrease the value of the gate tunneling current. The net effect of a slightly thinner gate oxide (increasing tunneling), slightly lower supply voltage (decreasing tunneling), and smaller devices (decreasing tunneling) from one generation to the next might actually result in an effective decrease in gate leakage.

Some of the plots in Figure 5 (e.g. the 256k and 512k caches for the 45nm and 32nm node) exhibit a sort of "saddle" shape, which shows that increasing cache associativity from direct-mapped to 2-way or 4-way does not automatically cause an increase in power dissipation, as the internal organization may allow a more power optimal implementation of set-associative caches compared to direct-mapped caches, especially for medium to large-sized caches.

A final observation for Figure 5 is that technology scaling is capable both of increasing or decreasing the total power of a cache. Which direction the power goes depends on which power component is dominant for a particular cache organization. Caches with dominant dynamic power (e.g. small caches/highly associative caches) will enjoy a decrease in cache power as dynamic power decreases because of the net decrease in the CV^2f function, while caches with dominant static power (e.g. large caches/medium-sized low-associativy caches) will suffer from an increase in cache power as the static power increases.

4.2. Detailed power breakdown

Figure 6 shows a detailed breakdown of cache power for three different cache configurations (16kB-4W, 32kB-4W, 64kB-4W). Each plot represents a single configuration implemented in the four nanometer nodes. For each of these nodes, total cache power is shown, along with its detailed breakdown. Power dissipation for each component, as shown by each vertical bar in the plot, is also broken down into its dynamic, subthreshold leakage, and gate leakage components.

The first notable point here is that for these cache configurations, going from 90nm to 65nm results in power decrease; going from 65nm to 45nm results in a smaller power decrease; finally, going from 45nm to 32nm causes a significant power increase. Again, these observations are caused by the intertwined decrease and increase of dynamic and subthreshold leakage power, respectively, as we go from one technology node to the next.

A second point is that for most of the plots, it can be seen that the two main contributors to cache power are the bitlines and the pipeline overhead (with the data_dataout component, which accounts for the power in driving the output data buses, also significant for some configurations). Pipeline overhead, shown in the plots as data_pipe_ovhd and tag_pipe_ovhd, accounts for the power dissipated in the pipeline latches along with the associated clock tree circuitry.

This is an important observation as it shows that the overhead associated with pipelining the cache is often a very significant component of total power, and that most of this power is typically dynamic. This can be easily seen in Figure 7, which shows the fraction of total power dissipated by pipeline overhead for all the configurations studied. This is especially noteworthy since we model aggressive clock gating where only latches that need to be activated actually see a clock signal. Consequently, we expect the pipeline overhead to be even more significant for circuits that use less aggressive clock-gating mechanisms. Lastly, it should be noted that since most of the power in the pipeline overhead is dynamic, it should decrease for smaller technology nodes, as seen in Figure 5.

Another point of importance is that although dynamic power in general tends to decrease because of the net effect of increased frequency but decreased supply voltage and device capacitances, this is the case only for circuits with device-dominated loading (i.e. gate and diffusion capacitances). Circuits with wire-dominated loading may actually see dynamic power go up as we go from one generation to the next, since a wire capacitance decrease due to shorter lengths and slightly smaller coupling area will typically be offset by the shorter distances between wires that cause significantly increased capacitances. This can be seen in Figure 6, where the dynamic power for the data_dataout component increases for all three configurations, mostly since the load for data_dataout is wire-dominated, as it involves only a few high-impedance drivers driving a long wire across the entire cache.

The power breakdown of representative cache configurations as shown in Figure 6 shows that most of the power in a pipelined, nanometer cache is dissipated in the bitlines, the pipeline overhead, and possibly the data output drivers. To get a better view of the design space, we can lump together similar components in order to show the entire space. In Figure 8, we subdivide total power into five categories -- bitline power, pipeline overhead power, decoder power, data processing power (data/tag sensing, tag comparison, data output muxing and driving), and lump all remaining blocks into a single component labeled "others." Results for the 65nm and 32nm nodes for all the configurations are then plotted together.

It is interesting to see that while the power dissipation due to the bitlines monotonically increases as cache size increases for both techology nodes, the power due to the other components does not necessarily do so. For instance, the pipeline power of the 65nm 128k-16 way and 256kB-16 way caches, where the pipeline power is significantly reduced from one configuration to the next. The reason here is that the bitline power mainly depends on the cache size (assuming static power is the dominant cause of bitline power) and not the cache implementation. The other power components, on the other hand, greatly depend on the particular implementation, so their values may noticeably fluctuate from one implementation to the next, especially if there exist implementations that cause an increase in the power of one component but results in a significant power improvement in

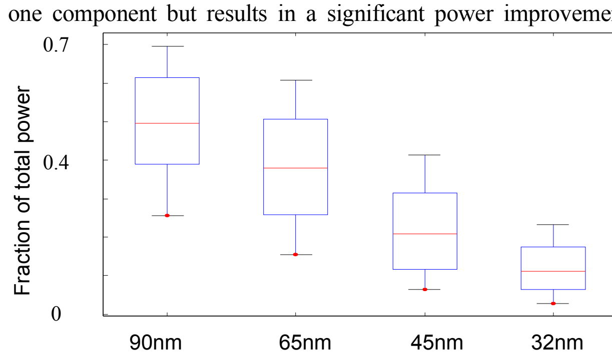

Figure 7: Pipeline overhead contribution to total cache power. Distribution of data showing the fraction of total power attributed to pipeline overhead for all cache configurations, where each plot shows min, max, median, and quartile information

some other part of the cache. It is important to realize that the optimization scheme that we used optimized total power, and not component power. As long as a specific implementation has a better power characteristic compared to another, relative values of specific cache components were not considered important.

It is easily seen from the different cache configurations in Figure 8 that that pipeline power (as represented by pipe_ovhd) will typically be a non-trivial contributor to the total cache power. Any analysis that fails to account for this pipelining overhead and instead assumes that caches can be trivially interfaced to a microprocessor core will be inaccurate. Although the figure also shows that the contribution of the pipeline overhead to total power is reduced from 65nm to 32nm compared to the contribution of the "dataproc" component (mainly because of the different effect of technology to the dynamic power of the two components), it should be noted that our study modeled conservative scaling of the dielectric constants of the interlevel dielectrics (i.e. we did not model aggressively scaled low-K interconnects). If wire delay becomes especially problematic in future generations and more aggressive measures are taken to improve the interconnect performance, we can expect the relative contribution of the pipeline overhead to the total power to increase in significance.

A final observation that we want to point out from Figure 8 is that increasing cache associativity may not necessarily result in a significant increase in total power, especially for the larger caches at 65nm and below. The power for these configurations are typically dominated by the bitline leakage, reducing the effect of any increase in power due to higher associativity. In some cases, the total power dissipation can actually decrease with an increase in associativity, as a

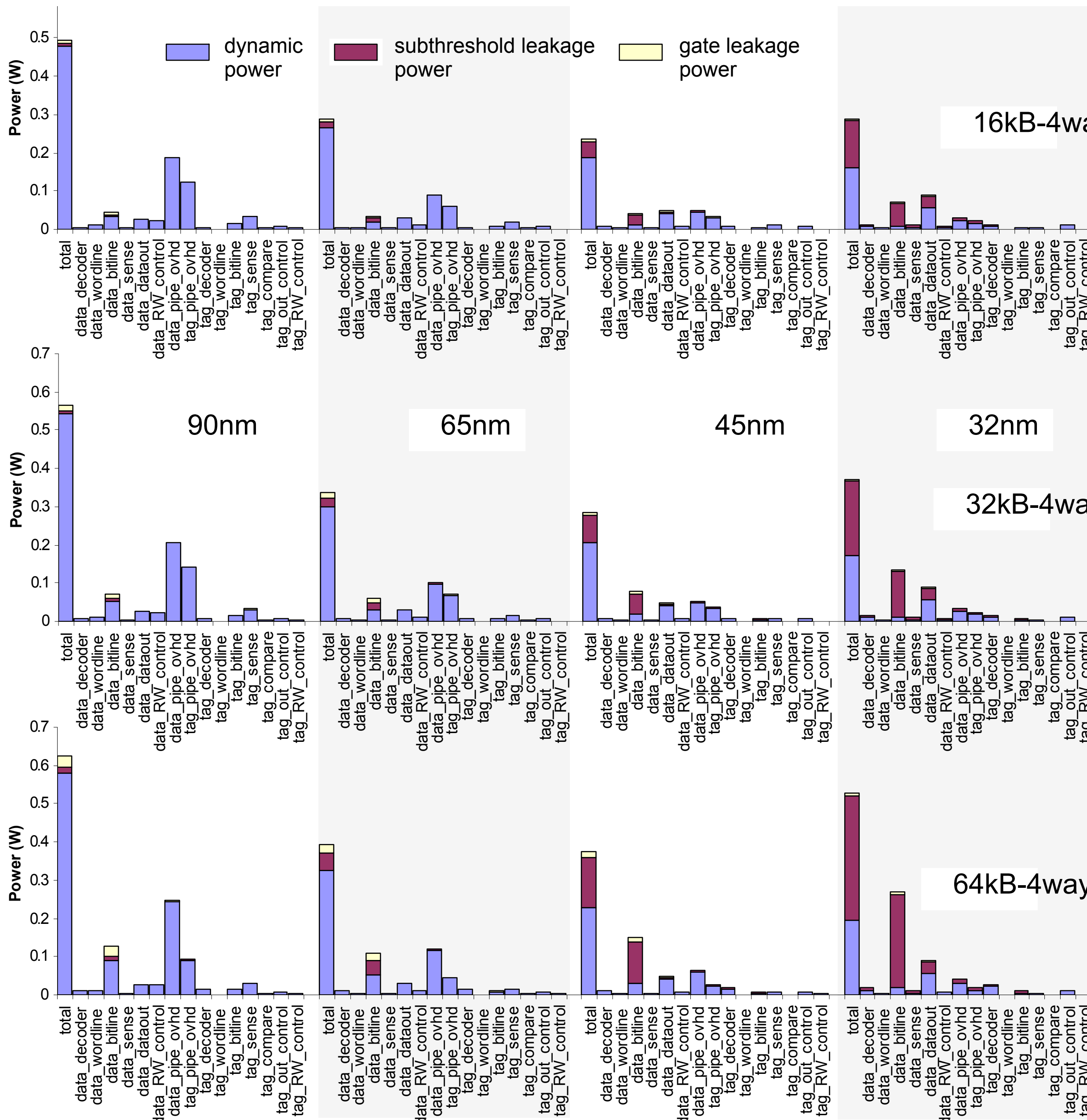

Figure 6: Detailed cache power breakdown. Detailed power breakdown for three different cache configurations showing the amount of dynamic, subthreshold leakage and gate leakage power being consumed by different parts of the cache. Each plot shows four sets of valuescorresponding to different technology nodes. (Note that all three plots use the same scale)

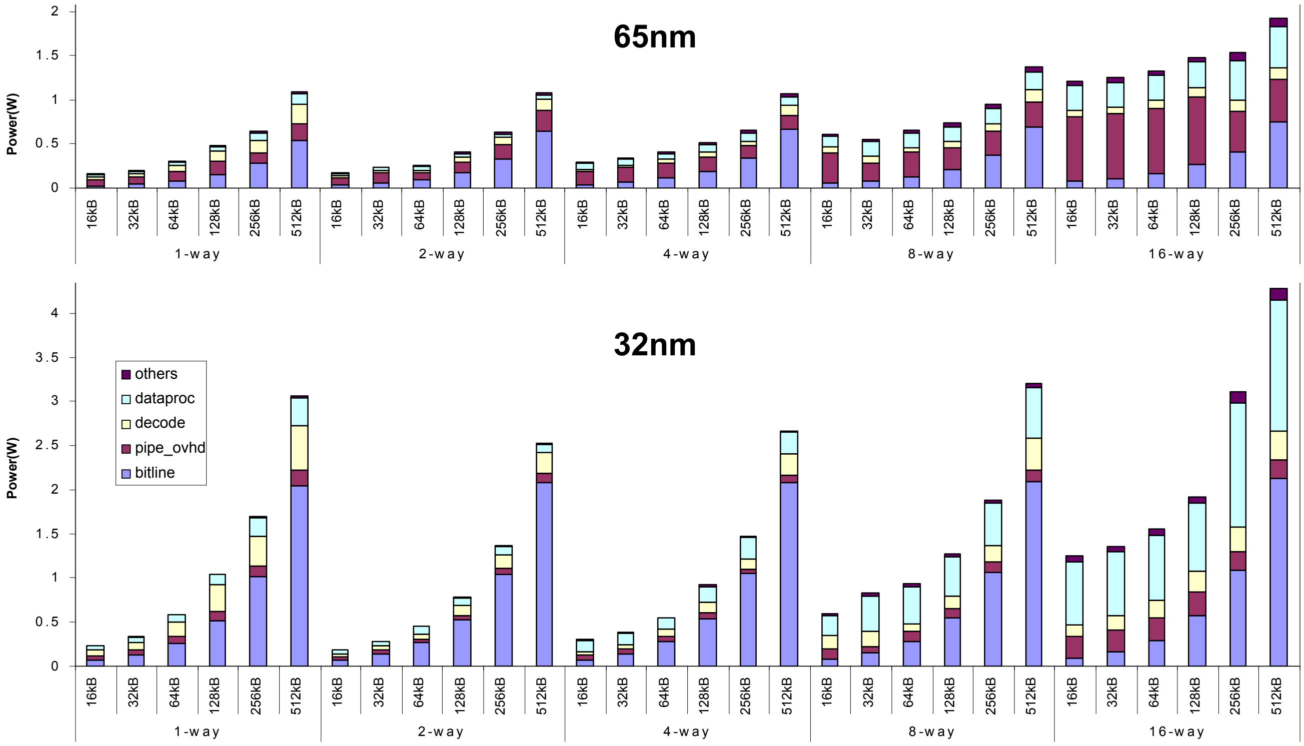

Figure 8: Power contributions of different cache parts. Power breakdown showing major cache power contributors for the 65nm and 32nm technology nodes for different cache sizes and associativities. (Note that the y-axis for both plots use the same scale)

particular configuration is able to balance the overall static and dynamic power for all the cache components. This knowledge will be useful in enabling large, high-associativity caches suitable for use in L2 caches (and higher).

5. CONCLUSION

We have presented a detailed power breakdown of the different nanometer pipelined cache configurations in a three-dimensional design space consisting of different cache sizes, associativities and process technologies.

We have shown that the power of nanometer caches will continue to increase, and that this increase will be primarily driven by static power due to subthreshold leakage currents. In addition, we saw that static power due to gate leakage tunneling currents does not contribute significantly to the cache power, even in the deep nanometer technology nodes.

Lastly, we have argued that practical microprocessor caches virtually require the use of pipeline latches to facilitate designing PVT variation-tolerant caches and debug and test-friendly designs. But in using explicit pipelining, accurate cache analysis requires accounting for the overhead introduced by these pipeline latches, as these represent a significant portion of the cache's power dissipated (in some corner cases being the dominant cause of power).

6. REFERENCES

[1] B.Amrutur and M.Horowitz, "Fast low-power decoders for RAMs," IEEE JSSC, vol.36, no.10, Oct 2001.

[2] Mohab Anis, "Subthreshold leakage current: challenges and solutions," ICM 2003.

[3] S.Borkar, "Circuit techniques for subthreshold leakage avoidance, control and tolerance," IEDM 2004.

[4] W.P.Burleson et al., "Wave-pipelining: A tutorial and research survey," IEEE Trans. VLSI Systems, vol.6, no.5, Sep 1998.

[5] Y.Cao et al., "New paradigm of predictive MOSFET and interconnect modeling for early circuit design," Proc. of CICC, pp.201-204, 2000.

[6] Z.Chen, M.Johnson, L.Wei and K.Roy, "Estimation of standby leakage power in CMOS circuits considering accurate modeling of transistor stacks," in Proc. Int.Symp.Low Power Electronics and Design, 1998.

[7] B.Doyle, et al., "Transistor elements for 30nm physical gate lengths and beyond," Intel Technology Journal, Vol.6, Page(s): 42-54, May 2002.

[8] P.Gronowski et al., "A 433-MHz 64-b quad-ussue RISC microprocessor, "JSSC, vol.31, no.11, Nov.1996, pp.1687-1696.

[9] J.P.Halter and F.Najm, ,"A gate-level leakage power reduction method for ultra-low-power CMOS circuits," in Proc. IEEE Custom Integrated Circuits Conf., 1997.

[10] F.Hamzaoglu and M.R.Stan, "Circuit-level techniques to control gate-leakage for sub-100nm CMOS," ISLPED August 2002.

[11] "International Technology Roadmap for Semiconductors 2005 Edition," Semiconductor Industry Association, http://public.itrs.net.

[12] "International Technology Roadmap for Semiconductors 2003 Edition," Semiconductor Industry Association, http://public.itrs.net.

[13] Predictive technology model. http://www.eas.asu.edu/~ptm

[14] M.Mamidipaka and N.Dutt, "eCACTI: An enchanced power estimation model for on-chip caches," Center for Embedded Computer Systems, Technical Report TR 04-28, Oct. 2004.

[15] J.Montanaro et al., "A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor, JSSC, vol.31 no.11, Nov. 1996, pp.1703-1714.

[16] H.Nambu, et.al., "A 1.8-ns access, 550MHz, 4.5-Mb CMOS SRAM," IEEE J. Solid-State Circuits, vol.33, no.11, pp.1650-1658, Nov. 1998.

[17] R.Rao, J.L. Burns and R.B.Brown, "Circuit techniques for gate and subthreshold leakage minimization in future CMOS technologies," , 2003. ESSCIRC '03..

[18] R.M.Rao, J.L.Burns and R.B.Brown, "Circuit techniques for gate and subthreshold leakage minimization in future CMOS technologies," ESSCIRC '03. Proceedings of the 29th European Solid-State Circuits Conference, 2003.

[19] G.Reinman and N.Jouppi, "CACTI 2.0: An integrated cache timing and power model," WRL Research Report 2000/7, Feb.2000.

[20] P.Shivakumar and N.Jouppi , "CACTI 3.0: An integrated cache timing, power and area model," WRL Research Report 2001/2, Aug.2001.

[21] Riedlinger, R., Grutkowski, T., "The high-bandwidth 256 kB 2nd level cache on an Itanium microprocessor," ISSCC 2002.

[22] Y.Ye, S.Borkar and V.De, "A new technique for standby leakage reduction in high-performance circuits," in Symp.VLSI Circuits Dig.Tech.Papers, 1998.

[23] Y.C.Yeo et al., "Direct tunneling gate leakage current in transistorswith ultrathin nsilicon nitride gate dielectric," IEEE Electron Device Letters, Nov. 2000.

[24] Weiss, D., Wuu, J.J., Chin, V., "An on-chip 3MB subarray-based 3rd levl cache on an Itanium microprocessor," ISSCC 2002.

[25] S.J.Wilton and N.P.Jouppi, "CACTI: An enhanced cache access and cycle time model," IEEE JSSC, vol.31, no.5, May 1996.

[26] The Tech Report. http://techreport.com/cpu/.