

The Impact of Cache Organization in Optimizing Microprocessor Power Consumption

N. Mohamed, N. Botros, and W. Zhang
Department of Electrical and Computer Engineering
Southern Illinois University, Carbondale, IL 62901-6603

ABSTRACT

In the recent years, power consumption has become increasingly an important design concern as silicon area and performance in modern computer systems design. Several factors have contributed to this trend. Perhaps the most visible have been the remarkable success and growth of Personal Digital Assistants (PDA's), Cellular phones and pagers, etc. This work is an attempt to explore the impact of feature size shrinkage and cache configuration on optimizing power consumption in modern processors. The work studies two commercially known RISC micro-processors. The StrongARM and the Alpha-21064 microprocessors. The latter is the ancestor of the former and is therefore used here as the baseline in our analysis. Our analysis has illustrated quantitatively great power saving when technology is downsized and cache is well organized

Keywords: *Deep Submicron, channel length, device threshold, cache associativity*

1) INTRODUCTION

The StrongARM microprocessor has a fascinating story of drastically decreasing its power consumption from 26 watts- the level of its decedent Alpha21064 [1] -to less than 0.5 watts. Its designers were able to achieve this by aggressively pursuing various techniques across different design hierarchy extending from the device level all the way up to the system level [2]. Since it has been reported that cache consumes well above 40% of total power, we chose to quantitatively explore how influential has been the role of the system cache organization. Moreover, we studied the role

that technology continues to play to achieve this goal as semiconductor industry advances into Deep Submicron (DSM) era. We examined the various trades off between power consumption and performance due to these factors. In the following sections, we present the basic analytical power models used by our power estimating tool: cacti [3] in section 2. We explore the process scaling in section 3 and cache system impact in section 4. The experimental results are stretched in section 5 and we conclude our work in section 6.

2) POWER MODEL

To drive analytical power consumption models, let us consider a simple CMOS inverter shown in Figure 1. Its symmetrical shape, its full logic swing and high noise margins make the circuit a fairly reasonable paradigm for most CMOS circuits. Consider Precharge phase of the circuit when the V_{out} rise from *low-to-high* (in response to *high-to-low* transition of the input voltage V_{in}). Assuming that both transistors are never *on* simultaneously, the inverter reduces to the equivalent circuit shown in Figure 2 and the total energy \mathcal{E} drawn from the power supply is given by

$$\mathcal{E} = \int v_{dd}(t) \cdot V_{dd} \cdot dt = V_{dd} \int C_L (dv_{out}/dt) \cdot dt = C_L V_{dd}^2 \quad (1)$$

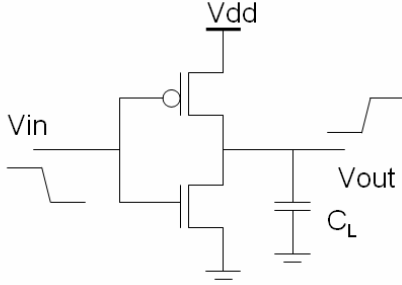


Figure 1: The CMOS inverter

The energy \mathcal{E}_c stored in the C_L is given by

$$\mathcal{E}_c = \int I_{vdd}(t) V_{out} dt = \int C_L (dv/dt) dt = C_L \int V_{out} dV_{out} = \frac{1}{2} C_L V_{dd}^2 \quad (2)$$

Thus, during this phase, half of the drawn energy has been dissipated in the PMOS device while the other half has been stored in the load capacitor C_L .

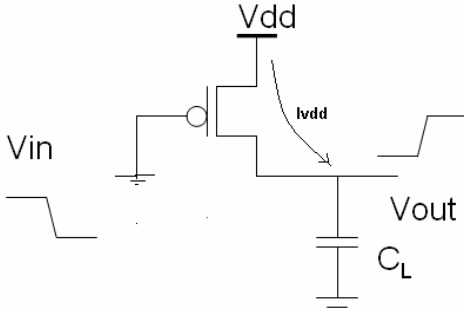


Figure 2: Inverter Equivalent Circuit - precharge phase-

Now, considering the discharge phase when the output voltage V_{out} drops back to *low* in response to the input voltage rise to *high*-the equivalent circuit model is as depicted by Figure 3. The energy \mathcal{E}_n dissipated by the nMOS device is equal to the energy that had been stored in load capacitance. In other words

$$\mathcal{E}_n = \frac{1}{2} C_L V_{dd}^2 \quad (3)$$

It follows that the total energy dissipated in both devices \mathcal{E}_d , is equal to the energy drawn from the power supply

$$\mathcal{E}_d = \int V_{dd}(t) V_{dd} dt = V_{dd} \int C_L (dv_{out}/dt).dt = C_L V_{dd}^2 \quad (4)$$

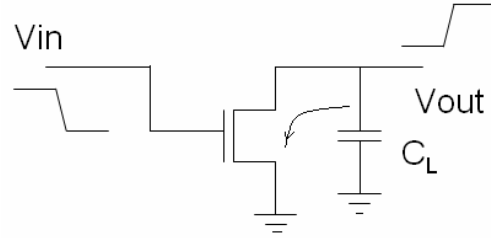


Figure 3: Inverter Equivalent Circuit - discharge phase-

If the inverter switches between those two phases (precharge and discharge) at a rate of f times per second, then the equivalent power dissipation \mathcal{P}_d is given by

$$\mathcal{P}_d = C_L V_{dd}^2 f \quad (5)$$

If we assume further that the inverter is embedded in a larger integrated circuit, as it is normally the case, and the probability that it switches at every cycle is ρ , then more precise power dissipation metric will be

$$\mathcal{P}_d = \rho C_L V_{dd}^2 f \quad (6)$$

This power component is known as the *dynamic power* consumption [3] and is normally contributes to most of power consumption in CMOS circuits. Studies have shown that, for a short period of time $T_{s/c}$, and as the input voltage V_{in} approaches or leaves the benchmark of $V_{dd}/2$, both nMOS and pMOS devices conduct simultaneously causing the instantaneous current to spike high to $I_{s/c}$. This, in turn, consumes portion of the delivered energy and can be computed as

$$E_{s/c} = T_{s/c} V_{dd} \cdot I_{s/c} \quad (7)$$

which leads to an average power known as *short-circuit power* consumption $\mathcal{P}_{s/c}$

$$\mathcal{P}_{s/c} = T_{s/c} V_{dd} I_{s/c} f = V_{dd}^2 C_{s/c} f \quad (8)$$

Finally, it has been observed in many MOS logic families, that while circuits are in steady-state (i.e. when they are experiencing no switching activities), a small current I_{stand} flows between supply rails causing what is called *leakage* power consumption \mathcal{P}_{stand} which is given by

$$P_{stand} = I_{stand} V_{dd} \quad (9)$$

As technology advances into Deep Submicron (DSM), this leakage power component magnifies substantially [5] causing great concerns. CMOS logic circuits are somehow immune to this kind of consumption, Pseudo MOS logic circuits proved to be extremely vulnerable to this standby power loss [6].

In summary, the total power consumption \mathcal{P} can be given by:

$$\mathcal{P} = \text{Dynamic Power} + \text{Short Circuit power} + \text{Leakage power}$$

These components are given by equations (6), (7) and (9) respectively.

3) PROCESS SCALING

This term refers to one of the VLSI method of deducting channel length of the MOSFET device. This finite length as been shown as L figure 4. Over the recent years, device dimensions have decreased exponentially due to the advances in silicon technology [4]. We observe a reduction of rate of approximately 13% per year, halving every 5 years. Figure 5 depict this trend and the near future projection [4].

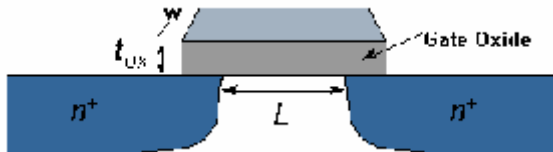


Figure 4: an nMOS Transistor cross-section

Decreasing MOS device channel length translates into consequent decrease in electric resistance and gate capacitance C_{gate} :

$$C_{gate} = \epsilon_{ox}/t_{ox} WL \quad (10)$$

Where ϵ_{ox} is the permeability of the oxide material and t_{ox} is its thickness. W and L are the width and length of the device respectively. This in essence, leads to substantial decrease in power consumption according to equation (6). In reality, this shrinkage of channel length goes hand on hand with scaling down supply voltage which translates into even further reduction in power. Table 1 depicts the voltage reduction associated with feature size scaling across many generations in compliance with the technology roadmap [8]. This trend is showing no sign of abating in the near future as illustrated graphically in Figure 5 below.

Table 1 : Prominent technology generations

Channel length(um)	0.75	0.35	0.18	0.10	0.07
Supply voltage(V)	5.00	3.30	1.80	1.50	0.90

Such reduction in supply voltage comes at the expense of serious degradation in circuit performance [5]. One way to restore performance is to scale down the threshold voltage \mathcal{V}_{th} of the CMOS device as well. However, reducing \mathcal{V}_{th} increases the subthreshold leakage current exponentially since:

$$I_{sat} = I_0 \exp(\mathcal{V}_{gs} - \mathcal{V}_{th})/n\mathcal{V}T \quad (11)$$

Where $I_0 = \mu_o C_{ox} (W/L) \mathcal{V}T^2 \exp(1.8)$ where μ_o is the zero bias mobility, \mathcal{V}_{gs} is the gate to source voltage and, $\mathcal{V}T$ is the thermal voltage which averages about 26mV at T=300K, and n the subthreshold swing coefficient given by $1 + C_d / C_{ox}$ with C_d being the depletion layer capacitance of the source/drain junction.

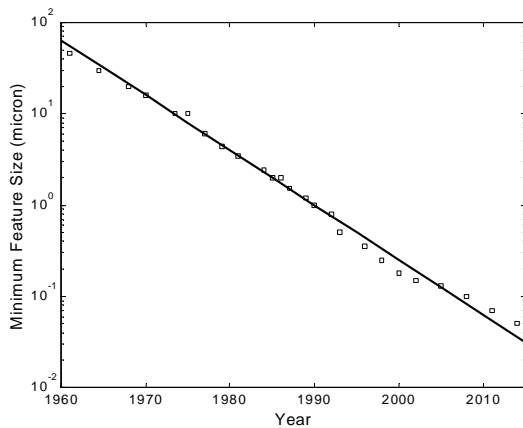


Figure 5: Minimum feature size projection in near future [4]

The designers of StrongARM scaled its fabrication process to $0.35\mu\text{m}$ from the $0.75\mu\text{m}$ which had originally been used to build up its ancestor. This alone has contributed to power reduction of 9.5%. We pursued this methodology and further scaled down the process to $0.18\mu\text{m}$ 1.8V, $0.13\mu\text{m}$ 1.2V, and $0.07\mu\text{m}$ 0.9V. For analysis purposes, we referred to the resulting variants as Strong-2, Strong-3 and Strong-4 respectively (with Strong-1 being the original). We then closely monitored the corresponding power saving and performance loss.

4) CACHE IMPACT

While both processors maintains an on-chip 16KB- instruction and data caches with 32B-cache lines, Alpha runs a direct mapped cache while StrongARM hosts a 32-way set associative. To offset the performance degradation highlighted in previous section, several solutions have been proposed, [5] introduced multi-threshold methodologies where the CMOS device library is extended to include high and low threshold voltages. [6] made use of such a diversified library and approached the problem from higher level of abstraction where behavioral description is optimized. To minimize

leakage power and compensate for performance loss, only components that are located on the critical paths are allowed to use the low voltage thresholds. Here, we tried to tackle the problem at an even higher level of abstraction by studying the impact of various cache organizations while maintaining cache sizes and block size intact. The cache associativity and block size replacement algorithms have been targeted. The simplescalar tool set [7] was used to for measurements and performance evaluation.

To minimize conflict miss rates in both caches, associativity has been increased from 1-way of Alpha to 2-way, 4-way, 8-way, 16-way and 32-way. The obtained simulations suggest that increasing associativity beyond the 8-way seem to have little contribution to performance as we dive into DSM. The hardware complexity and high hit time of Strong-1 32-set associativity does no longer seem to have great appeal.

The three commonly used block replacement policies (namely first in first out FIFO, least recently used LRU and random RANDOM) were tested next using two different benchmarks- *gcc* and *go*. Some sort of inconsistency has been observed. For while both benchmarks disfavor FIFO, the *gcc* with random algorithm recorded slightly better instruction per cycle when compared to LRU. However, it has been the reverse when *go* benchmark was used. Yet for as long as the overall performance is concerned, LRU is clearly proved to be more advantageous.

5) EXPERIMENTAL RESULTS

First, the cache system energy profile of all processors is shown in Figure 6. The advantage of feature size shrinking has obviously been translated into lower power consumption when compared to the original design of Strong-1. To highlight the effects of the cache mapping techniques on circuit power, the consumed energy of the various

processors was then plotted versus associativity in figure 7.

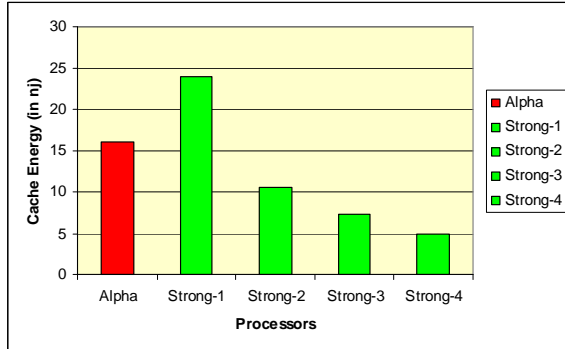


Figure 6: Consumed energy profile for the different system caches

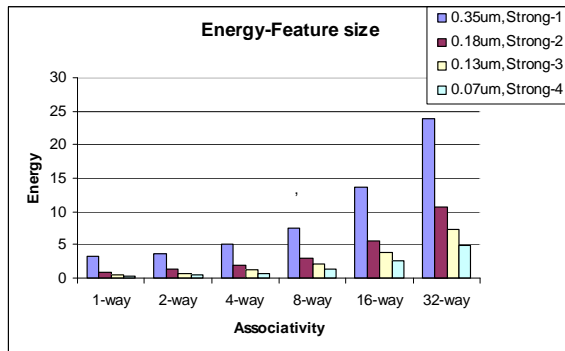


Figure 7: Consumed Energy by system caches of various StrongARM variants versus associativity

The performance measure of the baseline as reflected in terms of the number of instruction per cycles versus cache associativity is plotted in figure 8 using the three block replacement algorithms. This constitutes the original level before decay according to the *gcc* benchmark. The same parameter has been recalculated and plotted using *go* benchmark as given by figure 9.

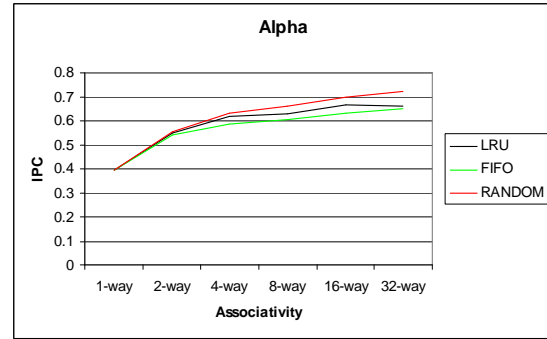


Figure 8: Alpha performance measure versus associativity (original performance) Benchmark: *gcc* (1000000 instructions)

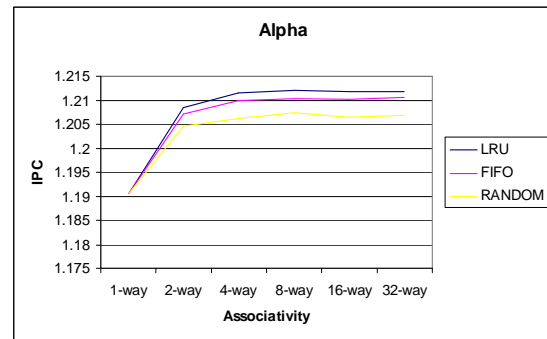


Figure 9: Alpha performance measure versus associativity (original performance) Benchmark: *go* (1000000 instructions)

The same benchmarks were then similarly used to measure the degraded performance of StrongARM. Again, the obtained results are depicted graphically in figures 10 and 11 below.

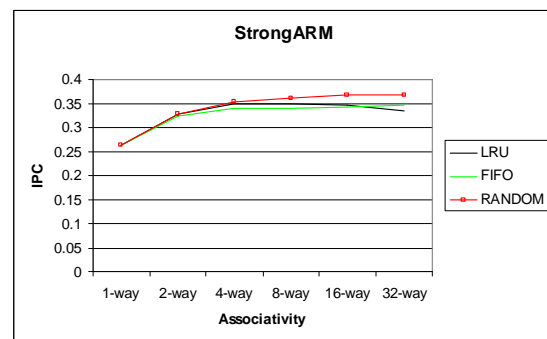


Figure 10: StrongARM performance measure versus associativity Benchmark: *gcc* (1000000 instructions)

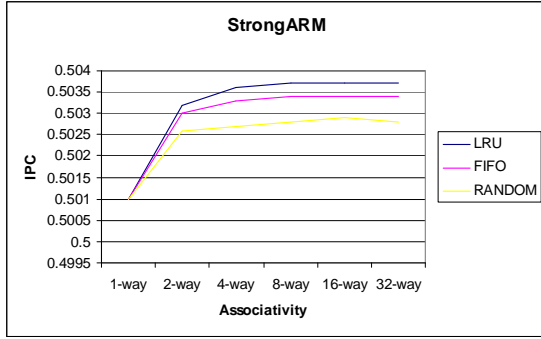


Figure 11: StrongARM performance measure versus associativity
Benchmark: *go* (1000000 instructions)

The trade off between power and performance involving associativity are illustrated below. Figure 12 profiles the baseline processor cache consumption is contrasted with Strong-1; the variant with highest cache power consumption. Figure 13, on the other hand, profiles the baseline processor cache consumption is in contrast to that of Strong-4; the variant with lowest cache power consumption. (Note that the IPC of each processor is multiplied by 10 for mere illustration purposes)

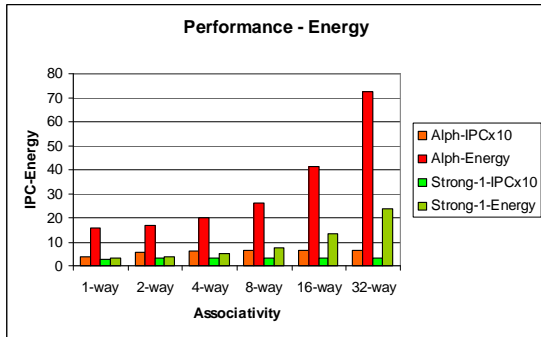


Figure 12: Performance and energy (of Alpha and Strong-1) versus associativity

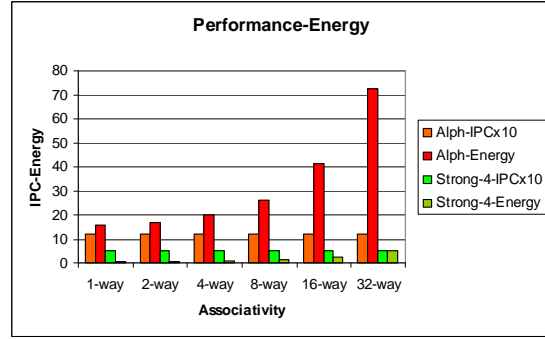


Figure 13: Performance and energy (of Alpha and Strong-4) versus associativity

6) CONCLUSION

In this work we studied the influence of today's rapidly decreased in silicon die area, as one of the elegant ways of optimizing power consumption in modern processor design. Since system caches play critical role in performance and consume great deal of total system energy, we have examined how the cache mapping techniques can trade off power to performance. Two commercially well known processors were used as vehicles in our analysis. We proved that, at no further hardware overhead, an even lower power, higher performer replica of the existing processor can be achieved.

7) REFERENCES

- [1] D. Dobberpuhl et al., "A 200 MHz 64b Dual-Issue CMOS Microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 11 (1992).
- [2] J. Mantanaro et al. "A 160- MHz, 32b , 0.5-W CMOS RISC Microprocessor ", *IEEE JSSC*, pages 1703-1712, Nov. 1996
- [3] G. Reinman and N. Jouppi. CACTI 2.0: An Integrated Cache Timing, Power and Area Model WRL Research Report 2001/2, August 2001
- [4] J. Rabaey, A. Chandrakasan, B. Nikolic, *Digital Integrated Circuits*, Pearson Education, Prentice Hall, 2nd Edition, 2003
- [5] M. Anis et al. "Design and Optimization of Multithreshold CMOS(MTCMOS) Circuits", *IEEE Transactions on Computer*

Aided Design of integrated circuits and systems, vol 22. No. 10, October 2003

[6] K. Khouri and N. Jha “ Leakage Power Analysis and Reduction During Behavioral Synthesis, IEEE Transactions on VLSI systems, Vol.10 No.6. Dec 2002.

[7] D. Bourger and T. Austin, The simplescalar Tool Set Version 2.0, Computer Architecture News, pages 13-24, June 1997

[8]<http://www.itrs.net/Common/2005ITRS/PIDS2005.pd>