Non-Uniform Set-Associative Caches for Power-Aware Embedded Processors

Seiichiro Fujii[1] and Toshinori Sato[2]

[1] Kyushu Institute of Technology,
680-4, Kawazu, Iizuka, Fukuoka, 820-8502 Japan
seiichi@mickey.ai.kyutech.ac.jp
[2] PRESTO, JST, Japan
toshinori.sato@computer.org

Abstract. Power consumption is becoming one of the most important constraints for microprocessor design in nanometer-scale technologies. Especially, as the transistor supply voltage and threshold voltage are scaled down, leakage energy consumption is increased even when the transistor is not switching. This paper proposes a simple technique to reduce the static energy due to subthreshold leakage current. The key idea of our approach is to allow the ways within a cache to be accessed at different speeds and to place infrequently accessed data into the slow ways. We use dual-V_t technique to realize the non-uniform set-associative cache, and propose a simple replacement policy to reduce average access latency. Experimental results on 32-way set-associative caches demonstrate that any severe increase in clock cycles to execute application programs is not observed and significant static energy reduction can be achieved, resulting in the improvement of energy-delay^2 product.

1 Introduction

Power consumption is becoming one of the most important concerns for microprocessor designers in nanometer-scale technologies, especially in the design of embedded processors for intelligent mobile devices because they require high performance as well as long battery life. Until recently, the primary source of energy consumption in digital CMOS circuits has been the dynamic power that is caused by dynamic switching of load capacitors. The trend of the reduction in transistor size reduces capacitance, resulting in less dynamic power consumption. Microprocessor designers have relied on scaling down the supply voltage, resulting in further dynamic power reduction [9, 13]. In addition, many architectural technologies to reduce power have been proposed by reducing the number of switching activities [15]. To maintain performance scaling, however, threshold voltage must also be scaled down with supply voltage. Unfortunately, this increases subthreshold leakage current exponentially. The International Technology Roadmap for Semiconductors (ITRS) predicts an increase in leakage current by a factor of two per generation [20]. Borkar estimates a factor of 5 increases in leakage energy in every generation [3].

Many of techniques [5, 10, 11, 18] proposed to address this problem have focused on cache memory that is a major energy consumer of the entire system because leakage energy is a function of the number of transistors. For example, the Alpha 21264 and the StrongARM processors use 30% and 60% of the die area for cache memories [15]. Current efforts at static energy reduction have focused on dynamically resizing active area of caches [5, 10, 11, 18]. In contrast, this paper proposes a technique that statically partitions each set into power-hungry and low-power parts. This static approach is suitable for embedded processors due to its low area and power overhead and design complexity.

The organization of the rest of this paper is as follows: Section 2 discusses the motivation of our work. Section 3 presents our concept to reduce leakage energy in caches, and explains a non-uniform set-associative cache as an implementation of the concept. Section 4 presents experimental results and discussion on the effectiveness of our approach. Finally, Section 5 concludes the paper.

2 Motivations

2.1 Leakage Energy

Power consumption in a CMOS digital circuit is governed by the equation:

$$P = P_{active} + P_{off} \tag{1}$$

where P_{active} is the active power and P_{off} is the leakage power. The active power P_{active} and gate delay t_{pd} are given by

$$P \propto f \bullet C \bullet V_{dd}^{\ 2} \tag{2}$$

$$t_{pd} \propto \frac{V_{dd}}{(V_{dd} - V_t)^{\alpha}} \tag{3}$$

where f is the clock frequency, C_{load} is the load capacitance, V_{dd} is the supply voltage, and V_t is the threshold voltage of the device. a is a factor dependent upon the carrier velocity saturation and is approximately 1.3 - 1.5 in advanced MOSFETs. Based on Eq.(2), it can easily be found that a power-supply reduction is the most effective way to lower power consumption. However, Eq.(3) tells us that reductions in the supply voltage increase gate delay, resulting in a slower clock frequency, and thus diminishing the computing performance of the microprocessor. In order to maintain high transistor switching speeds, it is required that the threshold voltage is proportionally scaled down with the supply voltage.

On the other hand, the leakage power can be given by

$$P_{off} = I_{off} \bullet V_{off} \tag{4}$$

where I_{off} is subthreshold leakage current, which is the major part of leakage current. The subthreshold leakage current I_{off} is dominated by threshold voltage V_t in the following equation:

$$I_{off} \propto 10^{-\frac{V_t}{S}} \quad (5)$$

where S is the subthreshold swing parameter and is around 85mV/decade [20].

Thus, lower threshold voltage leads to increased subthreshold leakage current and increased static power. Maintaining high transistor switching speeds via low threshold voltage gives rise to a significant amount of leakage power consumption.

2.2 Dual-V_t CMOS

The dual-threshold (dual-V_t) technique [19, 22] addresses the tradeoff decision between high performance and low leakage power. Transistors located on critical paths are assigned low threshold voltage, whereas transistors that are not critical to timing can tolerate high threshold voltage and slow switching speeds. The selection of threshold voltages are conducted at design time, and no additional circuits to dynamically control threshold voltages are required. Table 1 shows leakage current for high and low threshold voltage transistors in a 70nm process technology [7]. We can see that the leakage energy of transistors with high threshold voltage is a factor of 75 smaller than that of transistors with low threshold voltage. Hence, replacing a low-V_t transistor with a high-V_t transistor results in substantial energy reduction. In addition, different from gated-V_{dd} [18] technique, which cannot maintain data in cache memories when the power supply is cut off and thus is not applicable to cache memories, any additional cache misses does not occur in the dual-V_t technique. In summary, dual-V_t is a simple and efficient technique for static energy reduction of embedded processors, for which die size is of large concern as well as energy consumption.

Table 1. Impact of V_t on Leakage Current [7]

Tr type	V_{dd}	V_t	I_{off}
High-V_t	0.75	0.4	26
Low-V_t	0.75	0.2	1941

2.3 Related Work

Current efforts at static energy reduction have focused on dynamically resizing active area of caches [5, 10, 11, 18]. These architectural techniques employ circuit techniques. VT+ADR cache [5] and SA cache [10] use VT-CMOS [14] and ABC-MOS [16], respectively, both of which control the substrate bias to reduce leakage current in sleep mode by rising up threshold voltage. Decay cache [11] and DRI cache [18] use gated-V_{dd} [18], which shuts off the supply voltage to SRAM cells to reduce leakage current. These circuit techniques have some disadvantages. Gated-V_{dd} loses the state within the memory cell in the sleep mode. Thus, additional cache misses might

occur, resulting in additional dynamic power consumption. In contrast, VT-CMOS and ABC-MOS can retain stored data in the sleep mode. However, VT-CMOS require a triple-well structure and a charge-pump circuit, and ABC-MOS requires an additional power line that must be distributed throughout the memory array.

Abella et al. [1] and Balasubramonian et al. [2] proposed to utilize criticality information to reduce dynamic and leakage energy in data cache. Accesses in critical paths are served by fast and power-hungry cache bank, and those not in critical paths are served by slow and low-power bank. Their techniques are based on dual-V_t technology and thus are simple in leakage reduction. However, the additional critical path predictor is required for identifying every instruction's criticality.

3 Non-Uniform Set-Associative Cache

3.1 Our Approach

In this paper, we propose a simple technique for leakage energy reduction. It uses the dual-V_t technology [19, 22]. As explained above, the selection of threshold voltages are conducted at design time, and thus no additional circuitry is required for dual-V_t. While loss of adaptability might consume more leakage power than the dynamic techniques, removal of the power overhead as well as the complex circuitry is beneficial especially for embedded processors. From these considerations, we adopt dual-V_t to our proposed technique for static energy reduction. The key idea of our approach is to allow the ways within a cache to be accessed at different speeds and to place infrequently accessed data into the slow ways. This exploits dynamic information regarding data criticality in order to reduce power, as circuit techniques, such as transistor size optimizations [8] and clustered voltage scaling technique [21], exploit static information regarding timing criticality. Only critical data will be placed into the fast ways based on a principle of the locality. The difference from the techniques based on criticality [1, 2] is that cache partitioning is way-based instead of bank-based. While this technique is originally proposed for high-performance processors [4], we adopt it for low-power embedded processors and we call our proposed cache *non-uniform set-associative* (NUSA) cache.

3.2 Implementation Details

While we use a 2-way set-associative cache for explanation, our proposal is applicable to any set- and full-associative caches. The non-uniform 2-way set-associative cache consists of a pair of a fast and a slow ways. Transistors with low threshold voltage are used for implementing the fast way. Similarly, transistors with high threshold voltage are used for implementing the slow way. In this NUSA, once a way is allocated to a referred datum, the datum is placed in the same way until replacement. Thus, frequently used data might be placed in the slow way. This is not desir-

able, since processor performance is diminished. In order to place frequently used data into the fast way, we extend the NUSA by exploiting the locality in way reference. The locality means that a way recently accessed will be referred again in the near future. Thus, it is desirable to place data used last into the fast way. If the way accessed last is the slow way, or if the cache does not hold referred data and the slow way is allocated, we exchange the lines between the fast and the slow ways as shown in Fig.1.

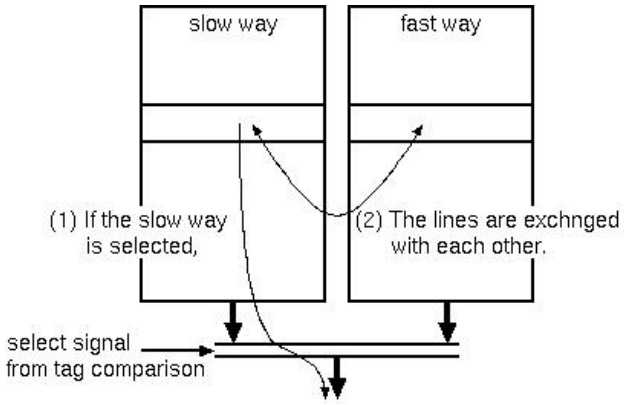

Fig.1. Way-Exchange

As mentioned above, the NUSA is applicable to any set- and full-associative caches. That means the NUSA can have multiple slow ways and multiple fast ways. Replacement in each group of the ways is based on LRU (Least Recently Used) policy. The way-exchange is performed as follows. The most recently used line in the slow ways is exchanged with the least recently used line in the fast ways.

It should be noted that a structural hazard can occur during every way-exchange. For example, consider two load instructions are executed. When the first datum is found in one of the slow way, the first load takes 2 cycles. However, the datum needs to be written into the fast way in the following cycle. As a result, data cache cannot be used for other access in the cycle. This structural hazard increases the slow-way access latency. In addition, the way-exchange consumes dynamic power. However, if the reduction in the leakage energy is larger than the increase in the dynamic energy, the NUSA with the way-exchange is a good solution since it improves execution cycles.

Modern high performance processors utilize schedules dynamic instruction order speculatively [12], and thus ambiguous execution latency due to the different access speed of the NUSA is not desirable. However, we focus on embedded processors, most of which are in-order and single issue processors, where pipeline stall after cache access is easier than in high-end processors.

4 Experimental Results

4.1 Simulation Methodology

We implemented our simulator using the HotLeakage [17, 23]. We modified the HotLeakage to execute ARM instruction set architecture and to utilize the NUSA. The processor model evaluated is based on Intel XScale processors [9]. A single-ported 32KB, 32B block, 32-way set-associative L1 instruction and data caches are used. The baseline model has caches, which have a load latency of 1 cycle. We evaluate non-uniform 32-way set associative L1 instruction and data caches, each of which has one primary fast way and remaining 31 slow ways (NUSA-1), and the other NUSA instruction and data caches, each of which consists of 2 fast ways and remaining 30 slow ways (NUSA-2). Both of them have the load latency of 1 cycle when the requested datum is placed in the fast ways and otherwise has a latency of 2 cycles. The structural hazard in the port during every way-exchange is considered in detail. During the event of an exchange, a new instruction or datum cannot be fetched. Hence, while instruction fetch suffers a large penalty every way-exchange, data fetch may suffer a smaller penalty if data cache is not accessed immediately after a slow-way access. This slow cache is pipelined and has the same throughput of 1 with the baseline model. The replacement policy is based on LRU. No L2 cache is used.

Table 2. Benchmark Programs

cjpeg	JPEG encode
gjpeg	JPEG decode
lame	MP3 encode
mad	MPEG audio decode
susan	Image recognition
tiff2bw	TIFF convert
tiff2rgba	TIFF convert
tiffdither	TIFF convert
tiffmedian	TIFF convert

The MiBench [6] is used for this study. It is developed for use in the context of embedded, multimedia, and communications applications. It contains image processing, communications, and DSP applications. We use original input files provided by University of Michigan. Table 2 lists the benchmarks we used. All programs are com-

compiled by the GNU GCC with the optimization options specified by University of Michigan. Each program is executed to completion.

4.2 Energy Parameters

In order to implement the NUSA, we use technology data for dual-V_t included in HotLeakage [23] and we assume a 70nm process technology. Because the drowsy mode [17, 23], the ABC-MOS [16], and the gated-V_{dd} [18] are implemented in HotLeakage, we compare the NUSA based on dual-V_t with these techniques. While each leakage control technique has extra hardware and dynamic and leakage power penalty, we consider only subthreshold leakage power consumed in caches because modeling the extra hardware in detail is under construction.

4.3 Results

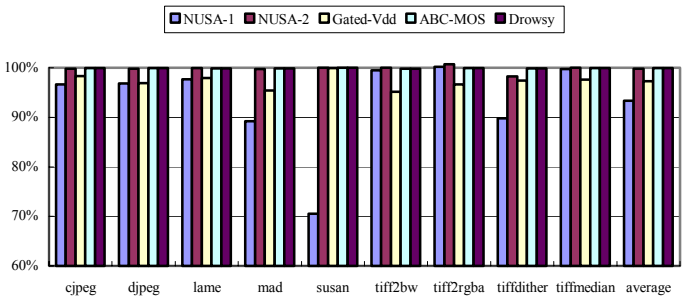

Fig.2. %Relative IPC (I$)

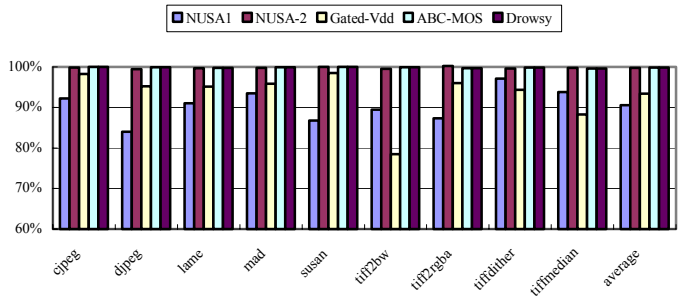

Fig.3. %Relative IPC (D$)

Fig.s 2 and 3 show processor performance relative to that having the baseline caches. For each group of 5 bars, the first two bars (see from left to right) indicate performance of the NUSA-1 and -2, respectively. The next three bars indicate performance of the gated-V_{dd}, the ABC-MOS, and the drowsy mode, respectively. Fig.2 presents the

case where instruction cache utilizes the leakage control techniques, and Fig.3 is for data cache. We can find performance degradation is largest in the NUSA-1, which has only 1 fast way. However, if the fast way is increased to 2, that is the case of the NUSA-2, the performance loss is eliminated. In summary, all leakage control techniques except the gated-V_{dd} maintain processor performance. This is because only gated-V_{dd} loses the state within the memory cell in the sleep mode.

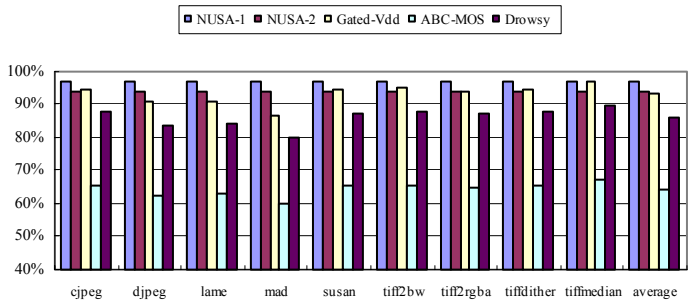

Fig.4. %Leakage Energy Reduction (I$)

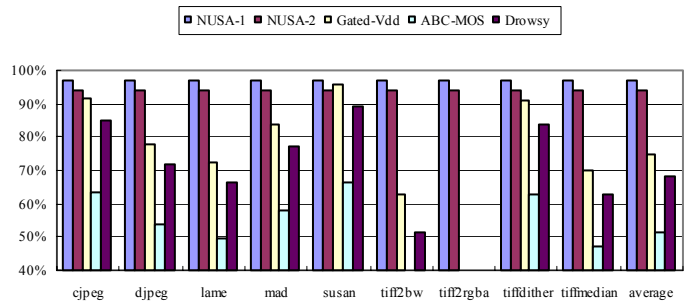

Fig.5. %Leakage Energy Reduction (D$)

Fig.s 4 and 5 show energy consumption relative to that having the baseline cache. The layout is the same with Fig.s 2 and 3. As you can see, the NUSA-1 is the best in leakage energy reduction. In contrast, the reduction using ABC-MOS is significantly small in comparison with other techniques. Because the difference between the NUSA-1 and -2 is small and because the performance loss is considerably smaller in the NUSA-2 than the NUSA-1, the NUSA-2 might be the best solution so far.

Fig.s 6 and 7 shows energy-delay-square product (ED^2P) relative to that having the baseline cache. The layout is the same with Fig.s 2 and 3. As you can see, the NUSA-1 is better in improving energy efficiency than the NUSA-2, especially in the instruction cache case, while the latter is better in performance than the former.

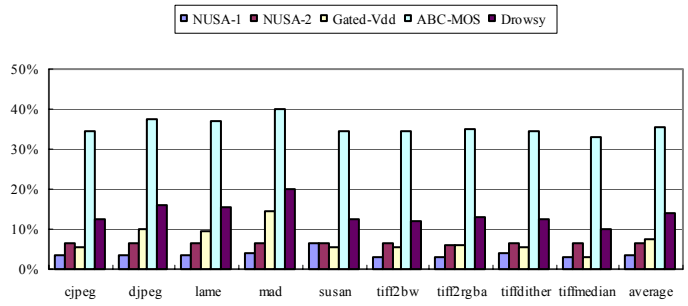

Fig.6. %Relative ED^2P (I$)

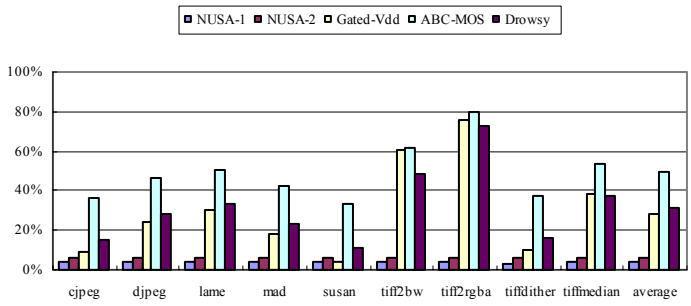

Fig.7. %Relative ED^2P (D$)

5 Conclusions

In this paper, we have proposed a simple technique to reduce the static energy consumed in caches. The key idea of our approach is to allow the ways within a cache to be accessed at different speeds and to place infrequently accessed data into the slow ways. Simulation results showed that any severe increase in clock cycles to execute the application program was not observed and significant static energy reduction could be achieved, resulting in the improvement of ED^2P.

References

1. Abella J.,Gonzalez A.: Power efficient data cache designs, International Conference on Computer Design (2003)

2. Balasubramonian R., Srinivasan V., Dwarkadas S., Buyuktosunoglu A.: Hot-and-cold: using criticality in the design of energy-efficient caches, Workshop on Power-Aware Computer Systems (2003)
3. Borker S.: Design challenges of technology scaling, IEEE Micro, Vol.19, No.4 (1999)
4. Burger D.: Technology scaling challenges for microprocessors and systems, Invited Lecture, COOL Chips V (2002)
5. Fujioka R., Katayama K., Kobayashi R., Ando H., Shimada T.: A preactivating mechanism for a VT-CMOS cache using address prediction, International Symposium on Low Power Electronics and Design (2002)
6. Guthaus M.R., Ringenberg J.S., Ernst D., Austin T.M., Mudge T., Brown R.B.: MiBench: A free, commercially representative embedded benchmark suite, Workshop on Workload Characterization (2001)
7. Hanson H., Hrishikesh M.S., Agarwal V., Keckler S.W., Burger B.: Static energy reduction techniques for microprocessor caches, International Conference on Computer Design (2001)
8. Hashimoto M., Onodera H.: Post-Layout transistor sizing for power reduction in cell-based design, IEICE Transactions on Fundamentals, Vol.E84-A, No.11 (2001)
9. Intel Co.: Intel XScale technology, http:// developer.intel.com/design/intelxscale/ (2002)
10. Ishihara T., Asada K.: An architectural level energy reduction technique for deep-submicron cache memories, Asia and South Pacific Design Automation Conference (2002)
11. Kaxiras S., Hu Z., Narlikar G., McLellan R.: Cache-line decay: a mechanism to reduce cache leakage power, Workshop on Power Aware Computer Systems (2000)
12. Kim I., Lipasti M.: Understanding scheduling replay schemes, International Symposium on High Performance Computer Architecture (2004)
13. Klaiber A.: The technology behind Crusoe processors, Transmeta Co., White Paper (2000)
14. Kuroda T., Fujita T., Mita S., Nagamatsu T., Yoshioka S., Sano F., Norishima M., Murota M., Kato M., Kinugasa M., Kakumu M., Sakurai T.: A 0.9V, 150MHz, 10mW, 4mm^2, 2-D discrete cosine transform core processor with variable-threshold-voltage scheme, International Solid State Circuit Conference (1996)
15. Manne S., Klauser A., Grunwald D.: Pipeline gating: speculation control for energy reduction, International Symposium on Computer Architecture (1998)
16. Nii K., Makino H., Tujihashi Y., Morishima C., Hayakawa Y.: A low power SRAM using auto-backgate-controlled MT-CMOS, International Symposium on Low Power Electronics and Design (1998)
17. Parikh D., Zhang Y., Sanlaranarayanan K., Skadron K., Stan M.: Comparison of state-preserving vs. non-state preserving leakage control in caches, Workshop on Duplicating, Deconstructing and Debunking (2003)
18. Powell M., Yang S.H., Falsafi B., Roy K., Vijaykumar T.N: Gated-Vdd: a circuit technique to reduce leakage in deep-submicron cache memories, International Symposium on Low Power Electronics and Design (2000)
19. Sirichotiyakul S., Edwards T., Oh C., Zuo J., Dharchoudhury A., Panda R., Blaauw D.: Stand-by power minimization through simultaneous threshold voltage selection and circuits sizing, International Design Automation Conference (1999)
20. Sylvester D., Kaul H.: Power-driven challenges in nanometer design, IEEE Design & Test of Computers, Vol.18, No.6 (2001)
21. Usami K., Horowiz M.: Clustered voltage scaling technique for low-power design, International Symposium on Low Power Design (1995)
22. Wei L., et al.: Design and optimization of low voltage and high performance dual threshold CMOS circuits, International Design Automation Conference (1998)
23. Y. Zhang Y., Parikh D., Sanlaranarayanan K., Skadron K., Stan M.: Hotleakage: a temperature-aware model of subthreshold and gate leakage for architecture, Techinical Report CS-2003-05, University of Virginia (2003)