Vol.2012-ARC-198 No.1
2012/1/19

データ保持性を利用したキャッシュのパワーゲーティング手法

金　均東[†1]　武田　清大[†1]
三輪　忍[†1]　中村　宏[†1]

Caches consume large amount of leakage power because of their large area and massive transistors. To handle leakage power of caches, several works using power-gating(PG) was proposed . Even though PG is capable of high leakage saving, energy overhead by dismissing data is a big shortcoming of PG. In this paper, we focus on the data retentiveness of PG. This nature was not focused on previous works. Voltage of SRAM cell does not decrease to zero immediately after PG and this phenomenon is valuable to relive energy overhead for data recovery. We also propose a circuit to utilize data retentiveness. With the oracle knowledge control, we examined leakage saving potential of our proposal for L1 instruction and data cache. Results show that utilizing retentiveness of PG have big potential of leakage saving.

1. Introduction

As technology scales down, leakage power becomes one of the most important issue in VLSI design. Scaling down of transistor size brings on increasing of DIBL(Drain Induced Barrier Lowering) effect and allows more transistors to be packed onto processors which result in consuming significant amount of leakage power. The on-chip caches are one of the main candidates for leakage reduction. Large capacity of cache is implemented for high performance purpose in modern high-end processors. Since cache consists of many transistors and occupies large area in die, it is responsible for significant fraction of overall leakage power.

Power-gating(PG) is a promising technique to handle leakage power because of its high leakage saving capability. A high threshold transistor, which is called sleep transistor, is inserted between the circuit and power supply (or ground). Sleep transistor serves as a local power switch to make a high-impedance path to power supply. By turning off the sleep transistor, circuitry goes into sleep mode, and then the voltage of circuit decreases to near zero. In this case, leakage power is dramatically reduced. Since circuitry is unusable in sleep mode, turning off should be done in idle time. Fortunately, caches have lower per area activity than logic component like execution units, applying PG to caches provides much of chance in leakage saving.

Therefore, several leakage reduction techniques for caches have been proposed using PG such as DRI cache[11] and cache decay[6]. DRI cache is to resize instruction cache to fit just the working set of the code, and turning off the rest of cache. Cache decay is a finer grain approach than DRI cache. Cache decay shuts down each individual cache line when the line is not accessed for a predetermined amount of time.

However, utilizing PG on caches has a risk of large amount of energy overhead for recovering data. When a cache line goes into sleep state, the state of SRAM cell is disappeared. If the data diminished by sleep is referenced, extra energy is induced by accessing lower hierarchy cache which would not be spent in non-PG case. This overhead decreases net leakage saving during a sleep interval. Furthermore, when energy overhead overwhelms leakage saving, it consume more energy than non-PG case. Transition to sleep mode is not desirable in this time. Therefore, data recovery energy decreases not only net leakage saving for one sleep interval but also leakage saving chance.

There are some state-preserving approaches such as drowsy cache[7] and DRG cache[5] which do not suffer from data recovery energy. They maintain proper voltage on SRAM cell to keep state even in low-leakage mode. However, this is also fundamental limita-

†1 東京大学
Tokyo University

tion which decrease leakage reduction capability in low-leakage mode compare to PG. PG achieves better leakage saving than state-preserving approaches when PG does not suffer from data recovery energy such as when data is not used again, or data recovery energy is ignorable because of large leakage saving with long time of sleep

In previous studies, PG was considered as non-state-preserving technique. However the voltage of cache line does not decrease to near zero immediately after turning off the sleep transistor. This means that time interval for waking up cache line without losing data exists and that PG has potential for further leakage saving. In this paper, we propose a novel PG scheme which relives data recovering energy for per line PG on caches. The main idea of this work is to utilize data retentiveness on PG cache when data in slept cell is still available. By utilizing data retentiveness, accessing lower hierarchy memory for data recovery is not required in short sleep. This results in increasing overall leakage saving. We also address a method to check the availability of data with ignorable overhead. We examined leakage saving potential of our proposal compare to that of conventional PG scheme for L1 instruction and data cache with oracle control.

The contributions of our work are summarized as below.

- We propose a new PG scheme to utilize data retentiveness on caches. It provides additional chances for leakage saving
- We propose a method to check the availability of data in cache.
- We clarify the energy model of utilizing data retentiveness compared to the model of conventional PG.
- We observe that leakage power of conventional PG is highly improved by using data retentiveness.

Rest of the paper is organized as follows. Section 2 presents related work. In section 3, we clarify how our energy model is different from that of conventional PG. Section 4 presents cache design considering data retentiveness. In section 5, we address experimental methodology and simulation set up. Section 6 shows experimental results and their analysis. Finally, we summarize this paper in section 7.

2. Related Works

Drowsy cache[7] saves leakage power by putting cache lines into low-power "drowsy" mode which make cache lines to be supplied by lower voltage. By keeping the SRAM cell's in lower voltage, drowsy has capability of data preservation because SRAM cell keeps enough effective voltage. Therefore, drowsy does not suffer from the energy overhead for data recovering. However, drowsy should pay a cost of lower leakage saving than decay in low-power mode because of the data retention. The requirement of two power supply is another shortcoming of drowsy cache in terms of worsening area overhead and design complexity.

DRG-cache[5] is another approach to achieve both of leakage saving and data retention. DRG-cache is designed that the voltage of VGND is saturated to the voltage where data is retainable after long time of sleep. They also utilize a single threshold transistor to both of the SRAM cell and the leakage path cutting-off switch. This is another merit of DRG-cache as process cost perspective. Same as drowsy cache, less leakage saving capability for data retention is one shortcoming of this work. VGND voltage should keep in enough voltage for data retention which means less tighter leakage cut off. Additionally, considering the variation problem, the saturation VGND voltage should be designed even lower. With the temperature and process variation, saturation VGND voltage significantly varies. With the severe increase of VGND voltage has risk of data lost. Moreover, a lot of SRAM cells in cache also increase the risk of data lost. Therefore, the VGND voltage should designed conservatively enough considering these issues.

Several works focused on the merit and demerit of cache decay and drowsy cache. Li et al[9] examined the effectiveness of decay and drowsy for L1 data caches with different parameters. Their show that L2 latency deterministic parameter on the comparison. For fast L2 latency, cache decay show better result in both of performance and leakage saving. They also evaluate considering gate leakage power with different thickness of gate oxide value. Meng et al[10] have shown that under oracle knowledge of the access stream, the best approach is a hybrid scheme using both of cache decay and drowsy cache. Even thought, their work is based on the theoretical model, the purpose of their work is discovering the limits of leakage saving on caches.

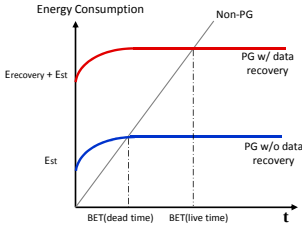

図 1 Energy Model

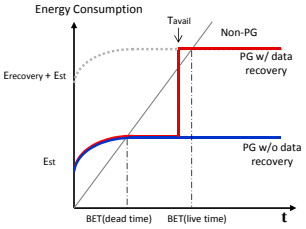

図 2 Energy Model Considering Data Retentiveness

3. Energy Model

This chapter represents leakage saving improvement by data retentiveness. Firstly, we clarify how to distinguish the occurrence of energy overhead in the perspective of cache behavior. Then, we explain how energy overhead effect to leakage saving chance and effectiveness. We also show that the improvement of leakage saving by data retentiveness varies depending on the cache behavior and sleep interval.

Cache decay uses terminology "live time" and "dead time" to classify the existence of data recovering energy by PG. After warming up cache, cache access result in hit of miss. Hit determines the "live time" which is the time interval between this hit time and previous access time. In this case, cache does not need to access to lower level cache. In other words, this means that dismissing of current data requires recovery energy. Therefore, when cache line goes into sleep mode during live time, data recovery consumes energy overhead. On the other hand, " dead time " determined by replacing current data to another data which is required currently. The interval between this replacement time and previous access is "dead time". In this case, dismissing of current data does not effect to energy for accessing lower hierarchy cache. In brief, sleeping during live time requires data recovery which result in energy overhead. On the contrary, sleeping during dead time does not consume energy overhead.

Because of energy overhead, PG does not always guarantee leakage saving. To determine PG chance and saving ratio correctly, break-even-time(BET) should be considered. BET is the minimum sleep time to pay back the energy overhead. Only the sleep time longer than BET guarantee energy saving. Therefore, the leakage saving benefit for live time is smaller than dead time. Note that dead time also should considers BET because sleep mode transient requires energy to turning on/off sleep transistor. However, energy for turning on/off sleep transistor is much smaller than data recovery energy. The energy of non-PG, PG for dead time and live time are represented as

$$E_{non-PG}(\tau) = I_{non-PG} V_{dd} \tau \quad (1)$$

$$E_{PG\_dead}(\tau) = \int_0^\tau I_{PG}(t) V_{dd} dt + E_{ST} \quad (2)$$

$$E_{PG\_live}(\tau) = \int_0^\tau I_{PG}(t) V_{dd} dt + E_{ST} + E_{recovery} \quad (3)$$

Inon-PG, and IPG represent the leakage current in non-PG state and sleep state respectively. Note that IPG decrease while time is passing. EST and Erecovery, refer energy overhead for turning on/off sleep transistor and data recovery respectively. $E_{recovery}$ is zero when there was not data recovery.

Figure1 shows the concept of energy consumption of non-PG, PG during dead and live time. When PG is not applied to SRAM cell, energy consumption increase linearly as time passed because leakage flows continuously. When PG is applied to SRAM cell, it consumes overhead energy as sleep start. As time passed, I_{PG} decreases and then reaches to small constant value. Depend on the existence of $E_{recovery}$, BET is heavily different. Our evaluation with hspice and CACTI[1)] show that BET without $E_{recovery}$ is relatively small and is under 50ns even in longest case in 45nm technology. However, BET with $E_{recovery}$ is much bigger and around 10μs. It means that under 10μs of sleep interval, sleep is not proper choice and even consumes more energy than non-PG case.

Figure2 shows energy consumption of utilizing data retentiveness. Here, T_{avail} refers the interval time until data is retained. When data is still available in sleep state. The data can be reused by coming back to active mode. When the sleep interval is shorter than T_{avail}, cache line go into sleep mode in live time without suffering data recovering energy. In this manner, utilizing data retentiveness provides further change to save leakage energy.

By utilizing data recoverability, equation 3 is modified as

$$E_{PG\_retentiveness}(\tau) = \int_0^{\tau} I_{PG}(t)V_{dd}dt + E_{ST} + E_{recovery}u(t - T_{avail}) \quad (4)$$

4. PG Scheme Considering Data Retentiveness

4.1 Data Retentiveness of SRAM cell

VGND voltage make important role in PG cache in terms of data retentiveness in sleep mode. Figure4 shows the variation of V_x and V_{VGND} after sleep. Here V_x refers the voltage difference between VDD and VGND When a cache line go into sleep mode, the leakage path between VGND and real GND is shut off. As the charge from VDD is accumulated to VGND, VGND voltage gradually increase until reaching the saturation voltage. When the current characteristic of cell and sleep transistor become same, VGND is saturated.

In non-sleep mode, the data in SRAM cell is preserved by firmly strapped to VDD and GND. However, when sleep transistor turned off, the voltage of node Q begins to increase as VGND increase. Therefore possibility of lost data increases.

The data retainable DRG-cache was designed that VGND voltage is saturated to 0.45V at VDD 2.5V of 0.25μs technology to preserve data. Note that their VGND saturation voltage is conservatively decided considering temperature and process variation. Because of this variation, their VGND saturation voltage increases to over 2V which result in data lost. This limitation decrease the leakage save compare to full power-gating because higher V_x means less tighter leakage pass cut. This is common phenomenon for data retention capable leakage saving techniques.

However, even thought the VGND voltage is saturated to VDD, there are intervals whose data is still retained because VGND voltage does not immediately increase to VDD in sleep mode. 8) reported that the lowest data retention voltage for SRAM cell was 190mV in 90nm technology. In this paper, we conservatively assume that SRAM cell's data is preserved until V_x is over 30% of VDD voltage. This value match well with pervious work. Drowsy cache set 30% of VDD voltage for data retentive low leakage mode. The data lost voltage V_x of DRG-cache was about 20% of VDD which is smaller than our assumption. Our experimental result shows that this interval range from 6.19μs to 9.72μs depend on temperature and sleep transistor width. Details are addressed in chapter 6.

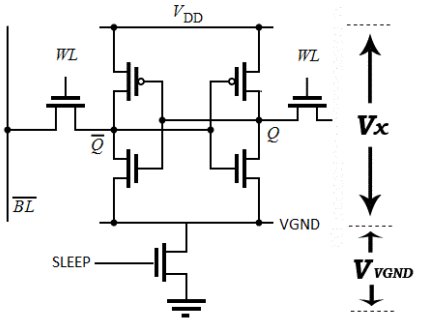

図 3 Voltage Division of SRAM and Sleep Transistor

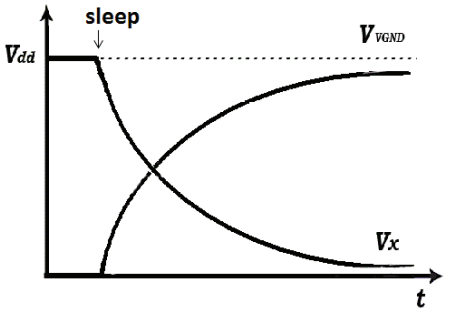

図 4 VGND Change

4.2 Availability Check Scheme

We shows our new cache design in Figure5. There are two modification in our proposed cache design compare to that of conventional cache. Firstly, we adapt voltage detector to guarantee the data recoverability. We use on-chip voltage detector based on clocked sense amplifier same as presented in[12)] with a little modification. The voltage detector compares VGND and Vref voltage. When VGND voltage increase to higher than Vref, voltage detector outputs 1 which means that data is lost. Vref is set to 0.77V by simple voltage divider using register. Secondly, the output of voltage detector is connected to valid bit of each cache line. When voltage detector output is 1 resets the valid bit. Even thought a cache is in sleep mode, when valid bit is not reset, the cache data is available and can be recovered by just wake it up. Enable signal is activated only when cache is accessed, therefore the power consumption of voltage detector is small enough.

Voltage detector consumes under 50fJ for one activation. PMOS holder and high threshold NMOS helps to reduce leakage power in standby mode. In standby mode, under 10nW of leakage power is consumed. Area overhead is less than 0.7% when cache size and block size is 32KB and 64B respectively. Both of them are very small and ignorable. Note that voltage comparator controls each cache block independently. The

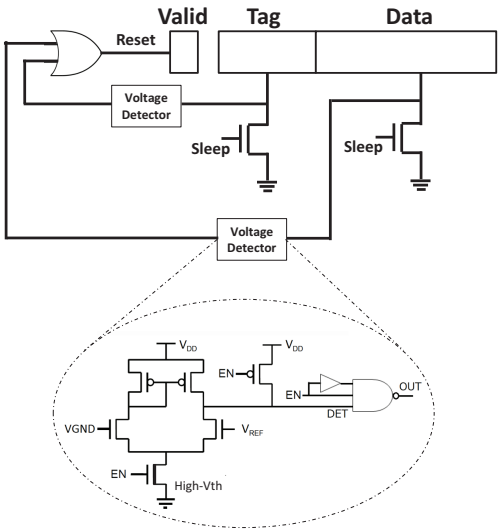

図 5 Availability Check Scheme

valid bit is not the target of leakage reduction.

5. Evaluation Methodology and Simulation Set Up

To evaluate the leakage saving potential of our proposal, we make a comparison between cache decay and our proposal for L1 instruction and data caches. We adapt oracle decision policy. When energy overhead overwhelm leakage saving, cache does not go into sleep mode. When idle time is dead time, cache sleeps all the idle time longer than the very short BET by turning on/off sleep transistor. In this case, proposed PG scheme has same leakage saving as cache decay. In live time, when idle time is longer than the BET considering $E_{recovery}$, both of the cache decay and proposal go into sleep mode. Moreover, proposal has additional leakage saving chance when idle time is shorter than T_{avail} because even though idle time is shorter that BET, our PG scheme does not suffer from $E_{recovery}$.

For the evaluation of T_{avail} and $E_{recovery}$, we use CACTI and hspice with Nangate 45nm[2]. The width of sleep transistor is designed as 5% and 10% of total width of SRAM cell. To obtain the idle time of L1 instruction and data cache, we use a cycle accurate processor simulator Onikiri2[3]. We assume 2GHz out-of-order processor whose configuration resembles as much as possible that of Alpha processor. We show processor configuration in Table 1. In our simulation, we use 29 applications from the SPEC CPU2006[4]. We simulated 1 billion cycles with ref data sets.

表 1 Processor Configuration

Issue Width	4
Functional Units	2 Integer and 1 FP ALUs 1 Integer multiplier/divider 1 FP multiplier/divider
L1 I-cache	32KB, 64B line, 4-way 3 cycle latency
L1 D-cache	32KB, 64B line, 4-way 3 cycle latency
L2 unified cache	1MB, 64B line, 8-way 15 cycle latency
Memory latency	200 cycles

6. Evaluation Results

6.1 Data Retentiveness Interval

Table 2 shows the T_{avail} and saturation voltage of VGND(Vsat) under different temperature and sleep transistor size. The results show that as temperature increase, T_{avail} decreases. The higher temperature CMOS has, the more leakage current flows. Therefore the charging speed of VGND increase which result in shorter T_{avail}

Temperature also effect to Vsat In table 2, as temperature increase Vsat is decrease. This is because of the IV-curve characteristic of CMOS. When the current of SRAM cell and sleep transistor is same, VGND voltage is saturated.

Lower V_{sat} let SRAM cell to have longer T_{avail}, because the increasing slope of VGND decrease as VGND voltage is close to Vsat. However in our experiment, charging speed increasing with temperature was more deterministic of T_{avail}.

Sleep transistor size is also another factor to vary T_{avail}. As we said above, VGND voltage saturate to the voltage where the current of SRAM cell and sleep transistor. Larger sleep transistor size mean lower equivalent resistance of sleep transistor which result in saturation of lower voltage between GND and VGND.

表 2 Data Retention Interval

Sleep transistor size	Temperature (°C)	V_{sat} (V)	T_{avail} (μs)
5%	25	1.06	8.87
5%	40	1.00	7.60
5%	60	0.98	6.19
5%	80	0.97	4.87
10%	25	0.91	9.72
10%	40	0.87	8.70
10%	60	0.86	7.44
10%	80	0.79	6.39

6.2 Leakage Saving on Caches

6.2.1 L1 Instruction Cache

Figure 6 shows normalized leakage energy of L1 instruction cache. The width of sleep transistor is 5% of total width of SRAM cell. For instruction cache, average leakage energy of cache decay varies from 17% to 19% depend on the temperature. On the other hand, average leakage energy of proposal varies from 7% to 8%. The biggest leakage energy difference between cache decay and proposal is gobmk.

Libquantum shows the lowest leakage energy difference. Libquantum also shows the maximum leakage energy of decay and proposal under 1% of leakage energy. Almost of the idle time distribution of libquantum is dead time. Therefore, libquantum does not suffers from $E_{recovery}$. This also means that there is no room to relieve $E_{recovery}$.

On the contrary, our work achieves much of leakage saving for sjeng and gobmk. Short(under 1K cycles) live time occupies more than 30% of their total idle time distribution. The difference of idle time distribution and results between libquantum and sjeng shows how idle time distribution effects leakage saving improvement of data retentiveness.

6.2.2 L1 Data Cache

Figure 7 shows normalized leakage energy of L1 data cache. For data cache, average leakage energy of cache decay varies from 15% to 18% depend on the temperature. Average leakage energy of proposal varies from 8% to 9%. The leakage saving effectiveness of cache decay and proposal are almost same as that of instruction cache.

Libquantum, bzip2, milc and lbm shows very low leakage energy with both of the cache decay and proposal. Since they have dead time dominant idle time distribution, there are very small difference in leakage saving. Gromacs, cactusADM, gcc, and namd

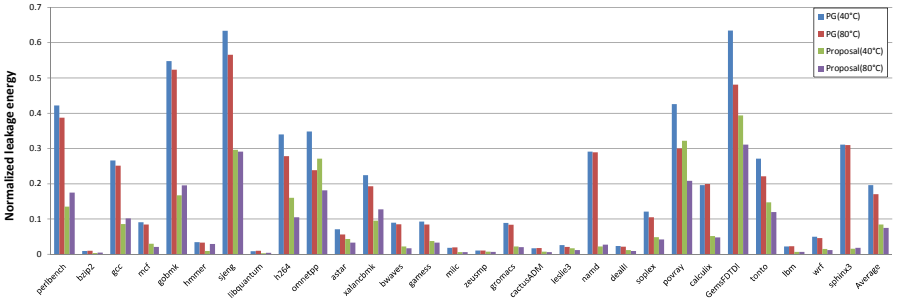

図 6 Leakage Energy of L1 Instruction Cache

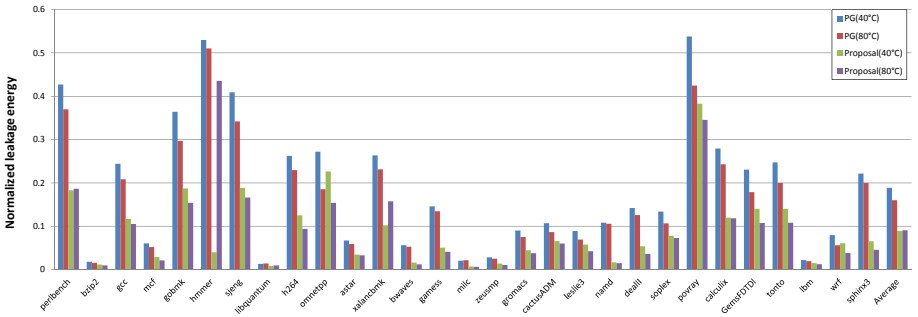

図 7 Leakage Energy of L1 Data Cache

also have dead time dominated idle time distribution. However, these benchmarks has slice more live time. When the distribution of this live time is for short interval length, the leakage saving of proposal increase.

On the other hand, the idle time interval of calculix, aphinx3 and gobmk are dominated by live time. Therefore their cache decay leakage energy are relatively big among data cache result. For perlbench, the live times whose interval length below 5μs occupies 30% of total idle time. Retentiveness utilized scheme effectively use this sleep chance and achieves further 20% of leakage saving.

Hmmer shows interesting result. There are big difference in leakage saving for proposal depends on the temperature. More than 90% of live time exist between T_{avail} of 40 °C and 80 °C condition. This is the reason why the leakage saving improvement heavily varies.

7. Conclusion

In this paper, we propose a new PG scheme utilizing data availability for caches. Result shows that proposal achieves additional leakage saving chance before the SRAM cell lost data. To check data availability, we use voltage detector whose overhead is ignorable. Our experimental result show that the available time ranges from 6.19μs to 9.72μs after SRAM cell go into sleep mode. To verify the effectiveness of utilizing data retentiveness, we adapt proposal to L1 instruction and data caches with oracle policy.

For both of L1 instruction and data caches, the usage of retentiveness achieves high leakage energy saving. The leakage energy normalized to non-PG scheme vary from 15% to 19% in average for L1 instruction and data cache. Leakage energy of proposal varies from 8% to 9% which is more than 2 times smaller than that of L1 caches.

With oracle policy control, we verified the potential of utilizing retentiveness.

謝辞 本研究の一部は、NEDO「ノーマリーオフコンピューティング基盤技術開発」事業による。

参 考 文 献

1) CACTI. http://www.hpl.hp.com/research/cacti/.
2) Nangate open cell library. http://www.nangate.com/.
3) Processor Simulator Onikiri2. http://www.mtl.t.u-tokyo.ac.jp/onikiri2/.
4) SPEC CPU2006 suite, The Standard Performance Evaluation Corporation. http://www.spec.org/cpu2006/.
5) Amit Agarwal, Hai Li, and Kaushik Roy. Drg cache: a data retention gated-ground cache for low power. In *Proceedings of the 39th annual Design Automation Conference*, DAC '02, pages 473–478, New York, NY, USA, 2002. ACM.
6) Stefanos Kaxiras, Zhigang Hu, and Margaret Martonosi. Cache decay: exploiting generational behavior to reduce cache leakage power. In *Proceedings of the 28th annual international symposium on Computer architecture*, ISCA '01, pages 240–251, New York, NY, USA, 2001. ACM.
7) NamSung Kim, Krisztián Flautner, David Blaauw, and Trevor Mudge. Drowsy instruction caches: leakage power reduction using dynamic voltage scaling and cache sub-bank prediction. In *Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture*, MICRO 35, pages 219–230, Los Alamitos, CA, USA, 2002. IEEE Computer Society Press.
8) A.Kumar, Huifang Qin, P.Ishwar, J.Rabaey, and K.Ramchandran. Fundamental data retention limits in sram standby experimental results. In *Quality Electronic Design, 2008. ISQED 2008. 9th International Symposium on*, pages 92 –97, march 2008.
9) Yingmin Li, Dharmesh Parikh, Yan Zhang, Karthik Sankaranarayanan, Mircea Stan, and Kevin Skadron. State-preserving vs. non-state-preserving leakage control in caches. In *Proceedings of the conference on Design, automation and test in Europe - Volume 1*, DATE '04, pages 10022–, Washington, DC, USA, 2004. IEEE Computer Society.
10) Yan Meng, Timothy Sherwood, and Ryan Kastner. Exploring the limits of leakage power reduction in caches. *ACM Trans. Archit. Code Optim.*, 2:221–246, September 2005.
11) Michael Powell, Se-Hyun Yang, Babak Falsafi, Kaushik Roy, and T.N. Vijaykumar. Gated-vdd: a circuit technique to reduce leakage in deep-submicron cache memories. In *Proceedings of the 2000 international symposium on Low power electronics and design*, ISLPED '00, pages 90–95, New York, NY, USA, 2000. ACM.
12) K. Usami, Y. Goto, K. Matsunaga, S. Koyama, D. Ikebuchi, H. Amano, and H.Nakamura. On-chip detection methodology for break-even time of power gated function units. In *Low Power Electronics and Design (ISLPED) 2011 International Symposium on*, pages 241 –246, aug. 2011.