

## Cache Controller Design With Run-time Power Gating

LEI ZHAO,<sup>†</sup> XU HUI,<sup>†</sup> NAOMI SEKI,<sup>†</sup> SAITO YOSHIKI,<sup>†</sup>  
YOHEI HASEGAWA,<sup>†</sup> KIMIYOSHI USAMI<sup>††</sup> and HIDEHARU AMANO<sup>†</sup>

A leakage-efficient cache controller design is presented in this paper. The key insight is that a large circuits subset is accessed only when cache misses happen. By utilizing the run-time power gating technique, such a subset can be dynamically powered-off as a power gated domain. Two simple but effective sleeping control policies are proposed to assure the final leakage reduction effect; and to eliminate the impact of wake-up process, a latency cancellation mechanism is also proposed. Evaluation results show, in 90nm CMOS technology, 69% and 64% of leakage power can be reduced for instruction cache controller and data cache controller without performance degradation.

### 1. Introduction

Power consumption has become a major concern in modern processor design. In the previous generations of CMOS technology, dynamic power dominated the total power consumption, and the leakage power is only considered in standby mode. However, leakage power exhibits an exponential-increasing characteristic as the feature size scaled down. If current scaling trend holds, leakage power will soon become the dominant source of power consumption<sup>1)</sup>. Suppressing leakage current is hence critical.

Due to the growing gap between the speed of CPU and memory, modern microprocessor designs devote a large fraction of semiconductor area to the cache system. For instance, 60% of StrongARM chip area has been occupied by cache system. Traditionally, cache system had low power density because only a small portion of the on-chip memory is accessed each clock cycle. But this is no longer true when leakage becomes dominant. The leakage power will become the leading source of the cache power consumption and thus the total power of a chip. In 70nm process<sup>2)</sup>, more than 60% of power can be consumed in L1 caches.

Cache memory is the main target to attack leakage because of the large number of transistors it holds. A number of leakage control mechanisms had been proposed for cache memory. The central idea behind these designs is to exploit the temporal locality of cache memory. By putting infrequently or unused cache lines into a low power mode, much of the leakage can be reduced. The DRI cache<sup>3)</sup> varies the size of the cache memory by tuning off the unused section by power gating, and 62% of leakage power can be reduced. Cache decay<sup>4)</sup> employs power gating on cache

memory at line granularity by monitoring the accessing pattern of each cache line, if there is no access to a cache line during a given period, the cache line will be turned off, and it results in roughly 70% reduction in L1 data cache leakage energy. An alternative technique to save leakage energy is reducing power supply for the selected cache line, by periodically turning all cache lines into low power supply mode, drowsy data cache<sup>2)</sup> can reduce the total leakage energy by 75%. By combining an on-demand wake-up policy with drowsy cache<sup>5)</sup>, the instruction cache leakage can be reduced up to 92.7%.

Efficient cache memory leakage control mechanisms stand out cache controllers as the next target for leakage reduction. Moreover, as the cache control scheme increasingly becomes complex, it is no longer a negligible part in leakage-efficient cache system design, especially for embedded processors. In our low power MIPS R3000 based CPU – Geysler-0<sup>6)</sup>, the cache controllers take up 23% and 40% of the leakage consumption of 8K 2-way set-associative instruction cache and data cache, respectively (Artisan's 90nm SRAM cells A1080SF0WG873M and F10801S0K7X74K are used to store tags and data).

In this paper, we propose an approach to apply the Run-Time Power Gating (RTPG) technique to the cache controller design. Leakage reduction effect can be achieved by dynamically putting the unused components into a sleep mode based on the analysis of accessing pattern and break-even point. The evaluation result shows the leakage power dissipation can be reduced up to 69% for instruction cache controller and 64% for data cache controller at 100°C, and zero performance penalties can be achieved by latency concealing mechanism.

The remainder of this paper is organized as follows. After an overview of the fine-grained power gating technique in Section 2, the power domain partition

<sup>†</sup> Graduate School of Science and Technology, Keio University  
<sup>††</sup> Shibaura Institute of Technolog

method and break even time analysis will be discussed in Section 3; in Section 4, dynamic sleeping control policies will be proposed for both instruction cache control and data cache controller; the evaluation results will be shown in Section 5, and Section 6 is for conclusion.

## 2. Fine-grain Run-time Power Gating

Power gating is a well-known technique to reduce leakage current. Between logic components and the real ground, a Virtual Ground (VGND) line is formed by inserting a high  $V_{th}$  power switch. While the power switch is turned off, the VGND line is charged up to the voltage nearby Vdd, the sub-threshold leakage is cut off, and the entire power domain which is connected with the power switch is turned into a sleep mode; when the power switch is turn on, parasitic capacitance on the VGND line is discharged and the power domain is turned back into the active mode. We define the wake-up time as the restoring time from the sleep mode to the active mode. Although this technique has a benefit that the target domain can be designed with the common design method, the wake-up time is the order of micro seconds. Such a long wake-up time prohibits it being used for on-demand leakage reduction applications, like cache controller.

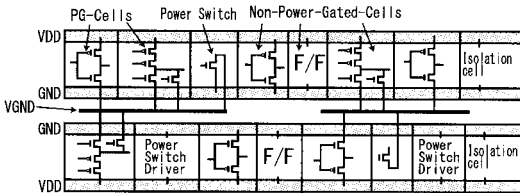


Fig. 1 A VGND architecture for a fine-grained RTPG

Fine-grained run-time power gating<sup>7)</sup>, which is a technique to minimize leakage power during the operation by powering on and off circuit components in a finer temporal/spatial granularity, is used in this paper. Unlike the conventional method, the entire power domain is partitioned into smaller local domains. VGND line and the power switch are shared only within the local domain. We can optimize the power switch size in a fine granularity with a set of cells each of which has its own VGND lines (PG-cell). By properly partitioning local domains and fine-tuning the power switch size, the wake-up time can be reduced to a few nano-seconds.

As shown in Fig.1, since existing ground rail in the PG-cells is used as the real ground, non-power-gated cells such as flip-flops, clock buffers, repeaters, isolation cells and power-switch drivers can coexist with

the power gated cells in the row. This allows us to utilize the conventional timing-driven P&R and clock tree synthesis when power gated cells and non-power-gated cells coexist in a target domain.

The design methodology for fine-grained RTPG had been established<sup>8)</sup>. At first, we follow conventional standard cell placement flow and then swap the cell with the same-sized PG-cell. PG-cell is connected to a VGND pin instead of the real ground rail. After inserting properly sized power switches and power switch drivers, signal wires and local VGND lines are routed.

Although fine-grain RTPG can reduce leakage current effectively, it is not a non-overhead technique. The state transfer between the sleep mode and the active mode consumes extra dynamic power for power switches, isolation cells and buffers in sleep signal wires. Even in the sleeping mode, leakage current decreases with time but still continues to flow until the capacitance on the VGND and output nodes of logic gates are charged up. The Break-Even Time (BET), which means the time point when the aggregated leakage energy savings equal to the energy overheads due to the power domain state transfer, is a crucial parameter to assure the overall power saving effect. We will illustrate the BET analysis flow in the next section.

## 3. Power Domain Partition and BET Analysis

To apply the fine-grained RTPG technique into cache controller design, we must divide the common design into different power domains. The Normal domain (NM domain) is implemented by common design methodology with standard cells, and it is always in an active mode. In contrast, the Power Gated domain (PG domain), which can be switched to the sleep mode when in idle state, follows the design methodology mentioned in Section 2.

Power domain partition has a determined effect on the BET and the final leakage reduction result. Since the RTPG is not a non-overhead technique, the partition method need to be well investigated. Two basic rules should be obeyed to assure the final leakage saving effect: 1) because the leakage power is a function of the number of transistors, leakage saving efficiency can be achieved by choosing a certain size PG-domain; 2) the chosen PG domain should have a high probability of long sleeping time, so as to compensate for the extra energy caused by power switches, isolation cells and sleep signal buffers.

### 3.1 Power domain partition

Cache controllers are typical finite state machine (FSM) circuits, and only a set of circuits are used each clock cycle. To maintain the right FSM state transi-

tion, the state flip-flops must be in the active mode, regardless of their working states. This means that our fine-grained RTPG can only be employed by combinational circuits. Paper<sup>7)</sup> proposed a RTPG FSM design flow by taking advantage of the clock gating enable signals. But in that paper, BET analysis is based on an analytical model, and it is hard to be implemented in on-demand applications. Here, we do the power domain partition based on the analysis of the accessing pattern of cache controllers.

The working states of cache controller can be divided into three types: the Processor Access Requirement Operations (PARO); the Next-level Memory Hierarchy Access Operations (NMHAO); and the Operating System Related Operations (OSRO). The PARO answers for all operations related with processor-cache interface, such as cache memory access, hit/miss detection and cache state update. This part shares the highest access frequency, but just consumes a small fraction of the semiconductor area. The NMHAO controls all next-level memory hierarchy accessing operations. It often takes up a much larger size than PARO, but only be accessed when cache misses happen. The OSRO only happens when operating system gives out explicit signals.

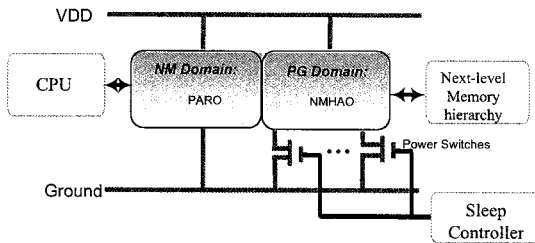


Fig. 2 Power Domain Partition

Fig.2 shows our power domain partition result. All logic circuits related to PARO are treated as an NM domain due to its high accessing frequency and small area; on the other hand, the PG domain includes all logic components realizing functions of NMHAO. Because this part is used only when cache misses happen, it stands a good chance to stay in sleep mode for a long period. Usually the function of OSRA can be implemented by reusing the corresponding parts of PARO and NMHAO, so here we do not treat it as individual domain. Even by fine-grained RTPG, the impact of wake-up time still can not be omitted (about 5ns here). When working at 200Hz, the waking up process of PG domain will lead one clock cycle stall. To conceal the latency penalty of waking up when misses happen, the first state after the cache-miss is overlapped by PARO and NMHAO, hence zero

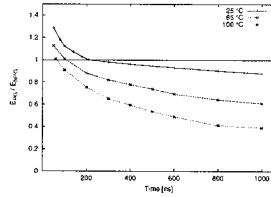


Fig. 3 BET of ICC

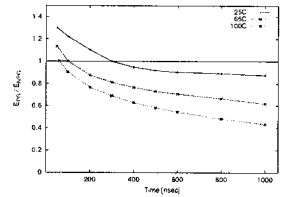


Fig. 4 BET of DCC

performance penalties can be achieved.

### 3.2 BET Analysis

To evaluate the BET of PG domain, a real design of fine-grained RTPG is needed, since the number of power switches is dependent on the target domain. The entire flow for fine-grained RTPG design is illustrated as follows.

First, the PG domain is designed with Verilog-HDL and synthesized by Synopsys's Design Compiler as an individual module. Then, the standard cell is replaced with same size PG cells, and power switches are inserted between VGND of cells and real ground by hand. Isolation cells are also inserted at all output ports in order to prevent path-through current when power is shut off, and a port for sleep signal wire must be prepared and connected to every power switch. At present, we couldn't prepare PG-cells for all standard cells in the library due to the limited designing time. This degrades the variety of cell-selection and will enlarge the area. Next, we process placement and routing by Synopsys's Astro. With Sequence Design's tool called CoolPower, the number of sleep power switches is optimized so as to make the time for wake up as short as possible. Then the energy dissipation for PG domain ( $E_{PG}$ ) can be analyzed by the extracted RC from post-layout data. By changing the sleep cycles, we can get  $E_{PG}$  at different sleeping period. We also evaluate the energy dissipation of non-powered counterparts ( $E_{NPG}$ ). The ratio  $E_{PG}/E_{NPG}$  is plotted against sleeping time in Fig.3 and Fig.4 for the PG domain of Instruction Cache Controller (ICC) and Data Cache Controller (DCC), respectively. The time point at which the ratio equals 1 is the BET. We do such a evaluation at three temperatures: 25°C, 65°C and 100°C.

Two facts need to be further explained. First, the BET reduces with increasing the temperature. This is because sub-threshold leakage increases with the temperature, and the fine-grained RTPG is more efficient in high temperature. Another fact is that the BET has no direct relationship with the area of PG domain, although in our case the BET of ICC is less than that of DCC which occupies a larger area. The BET is de-

terminated by the number and size of power switches. The extra energy dissipation introduced by isolation cells and enable signal buffers also affects the result. Thus the BET analysis for each domain must be based on the circuit level simulation, and the simple mathematical model is difficult to apply. The clock cycles of break-even time of each module is summarized in Table.1

**Table 1** Break-Even Time for PG domains

Temperature °C	Break-Even Time [cycles]	
	ICC PG domain	DCC PGdomain
25	42	60
65	18	20
100	12	12

#### 4. Dynamic Sleeping Control Policy

Ideally, the PG domain should be switched into the sleep mode when known its idle state will last longer than BET. But this kind of oracle prediction mechanism is not exist in a real design. The sleeping control policies, which decide when to shut down the PG domain, has a decisive effect on leakage saving.

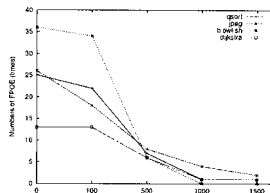
If a sleep event lasts less than its BET, the energy will be increased by power gating instead of being saved. We refer such a case as Fault Power-Off Event (FPOE). On the contrary, we call the sleep event whose sleeping time is longer than BET as Safety Power-Off Event (SPOE). The energy reduction result of a PG domain ( $E_{PG,reduc}$ ) can be expressed by following expression:

$$E_{PG,reduc} = E_{L,NPG} - E_{L,SPOE} - E_{L,FPOE} - E_{OvC}$$

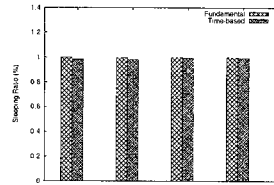
Where the  $E_{L,NPG}$  is the leakage energy dissipated when PG domain is always kept in the active mode;  $E_{L,SPOE}$  and  $E_{L,FPOE}$  are the leakage energy caused by SPOEs and FPOEs; and  $E_{OvC}$  is the extra energy introduced by the sleeping control logic, which includes both dynamic energy and leakage energy. Here, we conclude a simple guideline for the sleeping control policy design: 1) the SPOEs should be recognized as much as possible, the longer a sleep event stays in the sleep mode, the better leakage reduction effects can be achieved; 2) the  $E_{L,FPOE}$  poses a negative effect on energy saving, sleeping control policies should be capable of filtering out such cases; 3) the circuits engaged by sleeping control policies introduce extra energy dissipation, so simple policies share a high priority.

Based on the design guideline, three different policies are proposed in this section:

**Fundamental police:** The most straight-forward sleeping control policy is to turn PG domain into the sleep mode automatically as soon as the cache con-



**Fig. 5** Number of FPOEs for ICC



**Fig. 6** Sleeping Time Ratio of ICC

troller exits the cache miss handle process. It will be kept in such a mode until the next cache miss is detected. This fundamental policy shares the merit of no needing to add extra logic circuits, but taking a risk of degrading the leakage saving by FPOEs.

**Time-based policy:** An alternative to the fundamental policy is to put PG domain into the sleep mode only if a pre-set number of cycles (threshold) has elapsed since its last accesses. Such a time-based policy can effectively reduce the number of FPOEs at the cost of missing further opportunities to turn PG domain into the sleep mode during the idle times that are shorter than threshold. The fundamental policy can also be treated as special case of time-based policy when choosing 0 as the threshold.

**Double-checked policy:** The double-checked policy comes from the observation that the behavior of previous cache misses can be used to predict the timing slot between current miss and the next. A two-bit shifter is employed to record the behavior of last 2 sleep events, if the sleep event is a SPOE, the corresponding bit in shifter will be set to 1; otherwise it will be set to 0. The PG domain can only be powered-off when both the two bits is 1. This policy can relieve the sleeping time deterioration caused by time-based policy to some extent, but the number of FPOEs may increase.

Here, the efficiency of proposed policies are checked by 4 benchmark programs, Fig.5 shows the number of FPOEs at different threshold by the time-based policy, the results are obtained by clock cycle level simulation based on a 8K 2-way set-associative instruction cache. As the threshold increases, the number of FPOEs decreases dramatically. When choosing 1000ns as the threshold, this policy can avoid most of the FPOEs. Fig.6 compares the ratio of the sleeping time to the running time between the fundamental policy and a 1000ns-threshold time-based policy, from which we can see the sleeping ratio degradation is almost negligible. This is because in our test, the cache misses mainly come from compulsory miss and conflict miss<sup>9)</sup>, and the time slot between sequential groups of

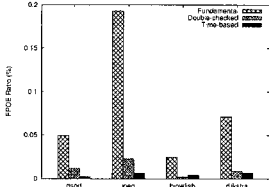


Fig. 7 FPOE Ratio of ICC

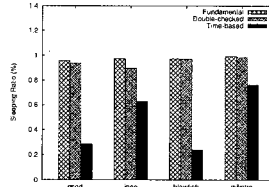


Fig. 8 Sleeping Time Ratio of DCC

cache misses is often a long period.

Unlike instruction cache, the distribution of data cache misses is more like a uniform distribution and it is difficult to choose a suitable threshold for all applications. Fig.7 shows the ratio of the number of FPOEs to the number of cache misses by three different sleeping control policies for a 8K 2-way set-associative data cache, and Fig.8 compares their sleeping ratio. The 4000ns threshold time-based policy can achieve the best FPOE-reduction effect, but degrades the sleeping ratio significantly. On the other hand, the double-checked policy has an insignificant impact on sleeping ratio, and can keep the number of FPOEs in an acceptable range.

The leakage power is sensitive to the temperature, as mentioned in section 3, the BET tends to become short when the chip becomes hot. Because both the time-based policy and double-checked policy have a counter to calculate the pre-set threshold, if thermal sensors can give an explicit signal to cache controllers, these pre-set threshold can be adjusted to achieve a better leakage reduction effect.

## 5. Evaluation Results

### 5.1 Power Analysis Method

Based on the design flow presented in section 2, an 8K 2-way set-associative instruction cache controller with the time-based policy (1000ns threshold) and an 8K 2-way set-associative data cache controller with the double-checked policy are designed. We also created their non-power-gated counterparts to do the comparison. In this section, we illustrate the power analysis method for our fine-grained RTPG design. To simplify the evaluation process, here, we assume that no extra power will be introduced by sleeping control policies.

The power consumption in the active mode is evaluated with a standard method. That is, the switching probability is measured with post-layout simulation, and the SAIF file is generated. Synopsys’s Power-Compiler is used to compute the power consumption with post-layout net-list.

Unlike the active mode, the power consumption in the sleep mode is complicated since the leakage power dissipated in a sleep event changes with the length of the sleeping time. We analyze the leakage current using transistor level simulations while changing the sleeping time. The sleeping cycles at every sleep event are captured by running an RTL simulation. We re-code both the sleeping cycles and the leakage energy caused by this event into a look-up-table, for example, if a N clock cycles sleep event happens, the sleeping cycles N and the corresponding power consumption  $Lws_N$  will be recorded as pairs. After getting the leakage power of the non-power-gated counterpart ( $Lwa$ ) with the stand method, the average leakage power  $Lw$  in a PG domain can be computed with the following expression:

$$Lw = \sum_i (Lws_i * \frac{M_i * i}{T}) + Lwa * (1 - \frac{\sum_i (M_i * i)}{T})$$

Where, it is assumed that  $i$  cycles sleeps occur  $M(i)$  times in the program whose total number of execution clock cycles is  $T$ .

By using the RTL model of Geysers<sup>6</sup>), a MIPS R3000 CPU; a clock-level simulation can be executed. Here, we use 4 benchmark programs at the power analysis, three of them are from MiBench<sup>10</sup>): Quick Sort from mathematics package, Dijkstra from the network package and Blowfish from the security package. As an example of media processing JPEG encoder program is also used.

### 5.2 Leakage Reduction Efficiency

Fig.9 shows the leakage consumption of both ICC and DCC @25°C. For comparison, the leakage power of their non-power-gated counterparts are also shown. At 25°C, 24% and 20% leakage power can be reduced for ICC and DCC respectively.

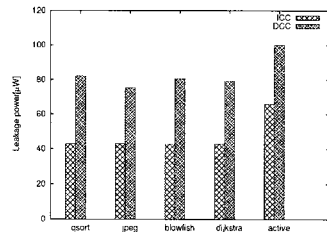


Fig. 9 Leakage comparison @ 25°C

Fig.10 and Fig.11 show the normalized power dissipation for ICC and DCC at 25°C, 65°C, and 100°C. As the temperature increases, the leakage reduction effect can be improved greatly. The overall leakage reduction of ICC is better than DCC due to their different access pattern. Although the area of DCC

is much larger than ICC, the short-time-lasting sleep events degrade the final leakage saving.

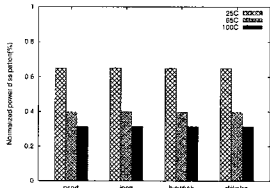


Fig. 10 Normalized power dissipation of ICC

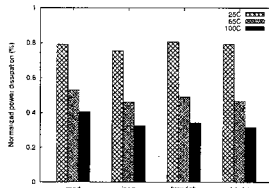


Fig. 11 Normalized power dissipation of DCC

### 5.3 Area Overhead

Table.2 shows the area overhead caused by the fine-grained RTPG technique.

Table 2 Area Overhead of PG Domain

PG domain	Non-PG [ $\mu\text{m}^2$ ]	PG [ $\mu\text{m}^2$ ]	Overhead [%]
ICache	6773.70	7346.55	8.5
DCache	11768.55	12914.25	9.7

The area overhead of PG domain compared with their non-power-gated counterpart is mainly caused by sleep transistors and isolation cells. As mentioned in Section 3, the incomplete set of PG-cells will result an extra area overhead. In the table, the non-power-gated counterparts are designed with the same limited cell library as PG domain, thus the difference comes from substantial overhead of the fine-grained RTPG, that is, the sleep transistors, and isolation cells. The numbers of inserted sleep transistors are optimized by Cool Power. When the target domain area is larger, more sleep transistors are required for optimization, and vice versa, thus the area overhead ratio for sleep transistors tends to be almost the same. The overhead difference comes from isolation cells. DCC has 14 more bits in terms of output port, so it shares a larger area overhead than ICC.

## 6. Conclusions

In this paper, we proposed an approach to apply the run-time power gating technique into cache controller design. After partitioning a cache controller into different power domains, leakage saving can be achieved by putting the power gated domain into a low power mode on the fly. We also propose two simple but effective sleeping control policies for both instruction cache controller and data cache controller, based on their accessing pattern. Evaluation results show at 100°C, more than 60% of leakage power can be reduced for both instruction cache controller and data

cache controller at a reasonable area penalty.

### Acknowledgments

The authors would like to thank VLSI Design and Education Center (VDEC), Synopsys, Cadence, Sequence Design, STARC, and Japan Science and Technology Agency (JST) CREST for their support.

### References

- 1) ITRS: Int'l Technology Roadmap for Semiconductor, <http://public.itrs.net> (2001).
- 2) Kim, N., Flautner, K., Blaauw, D. and Mudge, T.: Circuit and microarchitectural techniques for reducing cache leakage power, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 2, pp. 167–184 (2004).
- 3) Yang, S., Powell, M., Falsafi, B. and Vijaykumar, T.: Exploiting choice in resizable cache design to optimize deep-submicron processor energy-delay, *High-Performance Computer Architecture, 2002. Proceedings. Eighth International Symposium on*, pp. 151–161 (2002).
- 4) Kaxiras, S., Hu, Z. and Martonosi, M.: Cache decay: exploiting generational behavior to reduce cache leakage power, *Proceedings of the 28th annual international symposium on Computer architecture*, pp. 240–251 (2001).
- 5) Chung, S. and Skadron, K.: On-Demand Solution to Minimize I-Cache Leakage Energy with Maintaining Performance, *Transactions on Computers*, Vol. 57, No. 1, pp. 7–24 (2008).
- 6) N.Seki, e.: A Fine Grain Dynamic Sleep Control Scheme in MIPS R3000 (in japanese), *Proc. on SAC-SIS 2008* (2008).
- 7) Usami, K. and Ohkubo, N.: A Design Approach for Fine-grained Run-Time Power Gating using Locally Extracted Sleep Signals, *ICCD 2006.*, pp. 155–161 (2006).
- 8) T.Kashima, K.Usami, e.: Architecture design for low-power multiplier applying run time power gating (in japanese), *VLD*, Vol. 73, pp. 7–12 (2006).
- 9) Hennessy, J. and Patterson, D.: *Computer Architecture: A Quantitative Approach, 4th edition*, Morgan Kaufmann (2003).
- 10) Guthaus M.R., Ringenberg J.S., e.: MiBench: A free, commercially representative embedded benchmark suite, *Workload Characterization, 2001. WWC-4. 2001 IEEE International Workshop on*, pp. 3–14 (2001).