Dynamically Reducing Overestimated Design Margin of MultiCores

Toshinori Sato
Fukuoka University
Fukuoka Japan
toshinori.sato@computer.org

Takanori Hayashida
Fukuoka University
Fukuoka Japan
thayashida@fukuoka-u.ac.jp

Ken Yano
Fukuoka University
Fukuoka Japan
yano0828@fukuoka-u.ac.jp

Abstract— MultiCore processor is one of the promising techniques to satisfy computing demands of the future consumer devices. However, MultiCore processor is still threatened by increasing energy consumption due to PVT (Process-Voltage-Temperature) variations. They require large design margins in the supply voltage, resulting in large energy consumption. The combination of DVS (Dynamic voltage scaling) technique and Canary FF (flip-flop), named Canary-DVS, has been proposed to eliminate the overestimated voltage margin but has only been evaluated under the assumption of typical delay. This paper considers C2C (Core-to-Core) variations and evaluates how Canary-DVS eliminates the energy waste under the practical assumption of delay variations. We adopt Canary-DVS to a commercial processor, Toshiba’s quad-core Media embedded Processor (MeP). From Monte Carlo simulations, it is found that energy is reduced by 18.6% on average and there are not any noticeable discrepancies from the typical situations, when 0.064 of \( \frac{\sigma}{\mu} \) value is assumed in gate delay.

Keywords—PVT variations; safety design margin; Canary flip-flop; MultiCore; MPSoC

I. INTRODUCTION

Well known as Moore’s law, advanced semiconductor technologies have increased the number of transistors on a single chip and contributed to improve SoC (Systems on Chip) performance. Unfortunately, however, the aggressive integration technologies recently unveil a serious problem of parameter variations [2, 11, 21]. Variations on a single chip are classified into D2D (Die-to-Die) and WID (Within-Die) variations. Recently, the latter ones have become serious [16]. Random dopant fluctuations and line-edge roughness (Process variations), uneven supply voltage distribution (Voltage variations), and temperature fluctuations (Temperature variations) cause parameter variations. Process variations are essential in semiconductor technologies and they affect each transistor’s threshold voltage, resulting in performance variability. This paper focuses on process variations.

PVT variations influence circuit delay. Even though chips have identical design and environment, some of them may violate timing specifications. Currently, the only practical solution to manage PVT variations is to increase design margin. In other words, the supply voltage is overestimated as shown in Figure 1. Generally, SoC’s maximum clock frequency is determined by considering the worst-case critical path delay and the design margin. In the conservative design, the margins are summed up as shown in the figure. Managing PVT variations is a key to low power. This is the topic of this paper.

![Figure 1. Safety design margin.](image-url)

On the other hand, the current trend towards increasing mobile devices requires high-performance and low-energy microprocessors. Generally, high performance and low energy conflict with each other and it is very difficult to achieve both of them simultaneously. While energy is already the first-class design constraint in embedded systems, it has also become a limiting factor in general-purpose microprocessors, such as those used in data centers. In order to solve the problem, we can exploit parallelism. MPSoC (MultiCore Processor SoC) is one of the solutions for high-performance and low-energy and it is already adopted in embedded microprocessors. In order to further improve energy efficiency, DVS (Dynamic Voltage Scaling) technique is widely adopted. It is one of the most power-efficient design techniques for single-core processors. This paper proposes a technique that combines DVS with a delay-
error prediction method to eliminate the wasteful energy consumption due to the overestimated design margin.

The rest of this paper is organized as follows: Next section summarizes related works. Section 3 explains how the proposed technique eliminates the energy waste. Section 4 introduces our evaluation methodology and Section 5 presents simulation results. And last, Section 6 concludes.

II. RELATED WORKS

Razor [4, 6] permits to violate timing constraints to improve energy efficiency. Razor works at higher clock frequency than that determined by the critical path delay, and removes the supply voltage margin for power reduction. The voltage control adapts the supply voltage based on timing error rates. The control system works to maintain a predefined error rate. At regular intervals the error rate is computed and the rate differential between the predefined and the current error rates calculated. If the differential is positive, it indicates that supply voltage could be decreased. The otherwise indicates that the supply voltage should be increased. In order to detect timing errors, Razor FF (flip-flop) is proposed. Each timing-critical FF (main FF) has its shadow FF, where a delayed clock is delivered to meet timing constrains. In other words, every shadow FF is expected to always hold correct values. If the values latched in the main and shadow FFs do not match, a timing error is detected.

Li et al. [13] improve the robustness of the Razor FF and evaluate it on a superscalar processor design. Das et al. [5] propose the second-generation Razor FF called RazorII FF. Differences from the original Razor FF (or RazorII FF) are the removal of its recovery mechanism and the use of a level-sensitive latch. Instead of implementing the dedicated counterflow datapath, RazorII FF relies on the conventional microarchitectural reply mechanism. This simplifies RazorII FF design and thus realizes its small size. The use of a level-sensitive latch instead of a master-slave FF reduces its power consumption. A critical issue for Razor FF is the susceptibility to datapath metastability. In order to eliminate it, a dynamic transition detector with a time-borrowing datapath latch (TDTB) and a double-sampling static design with a time-borrowing datapath latch (DSTB) are proposed [3].

In order to eliminate the wasteful safety margin, we propose a dual-sensing FF, which we name Canary FF [17]. Difference from the error-detecting FFs above, it predicts timing errors. Hence, Canary FF is applicable to the common SoCs that do not require the recovery mechanism, where the error-detecting FFs are inapplicable. To achieve this, Canary FF has a delay element in datapath instead of in clock network. This feature has another advantage of eliminating the short path problem found in Razor FF [4]. We evaluate its energy efficiency on single-core processors [17] and on MPSoCs [18]. Fuketa et al. [7] combine Canary FF with adaptive speed control to mitigate timing margin variations of a 32-bit ripple carry adder.

To the best of our knowledge, any timing-error-detecting nor timing-error-predicting FFs are not adopted to and are not evaluated on MPSoCs under process variations.

The impact of process variations on MultiCore design has been investigated in [9]. WID variations cause performance asymmetry among cores on the same die. It creates task scheduling problems, which will result in degraded throughput. Teodorescu et al. [20] propose a variation-aware scheduling. While managing task scheduling problems is an interesting research topic, this paper focuses on eliminating the power waste.

III. ENERGY WASTE ELIMINATION

First in this section, it is described how DVS reduces energy consumed by an MPSoC. Next, a technique that eliminating the energy waste is proposed. And last, it is explained how WID variations affect performance and energy consumption of an MPSoC.

A. Per-chip DVS

The target in this study is an AMP (Asymmetric MultiCore Processor). In an AMP, every task runs on its dedicated core. In this study, it is assumed that different cores process different programs, which are independent of each other. In other words, a multiprogramming environment is assumed.

DVS technique is utilized for multicore power management. For multicore processors, two implementations of DVS techniques can be considered. One is per-core DVS where different cores operate at different voltage levels. With multiple supply voltages, only cores that require a higher supply voltage run at the supply voltage while other cores operate at lower supply voltage or are completely shut down. The other is per-chip DVS where all cores operate at the same voltage level. With one scalable supply voltage, all cores run at the voltage. While the per-core DVS will have larger flexibility in power management than the per-chip DVS will do, the per-core DVS requires voltage islands and on-chip regulators, which increase design complexity, chip area, and power consumption, resulting in larger manufacturing cost. Since we focus our attention on embedded devices, where cost is one of the first class design constraints, and since there is little difference in power savings between these implementations [8, 19], we chose the per-chip DVS technique in this study.

B. Canary-DVS

In order to decrease the overestimated supply voltage and thus to eliminate the wasteful power consumption, DVS technique is combined with dual-sensing FFs, such as Razor FF [4, 6] and Canary FF [7, 17]. The technique proposed in this paper combines DVS technique with Canary FF. We name the technique Canary-DVS and have already found the power reduction is promising both for single-core processors and for MPSoCs [17, 18]. However, it has not been evaluated on the MPSoCs that are affected by process variations.
Figure 2 shows Canary FF. Each FF (indicated as main FF) is augmented with a delay buffer and a redundant FF (indicated as shadow FF). The shadow FF is used as a canary in a coal mine to predict timing errors. It runs into the timing error a little bit before the main FF. An error is predicted by comparing values held by the main and shadow FFs. An error prediction signal triggers voltage control.

Figure 3 explains how DVS technique utilizes Canary FF [18]. The horizontal and vertical lines present time and supply voltage, respectively. At regular DVS intervals, the supply voltage is decreased step by step if a timing error is not predicted during the last DVS interval. When a timing error is predicted, the supply voltage is increased. After the first timing error is predicted, timing errors will be repeatedly predicted if the supply voltage is decreased again in the end of the succeeding DVS interval. Since every supply voltage switching makes processor unavailable during the transition, this oscillation has a serious impact on performance and on power efficiency. In order to prevent the oscillation, we use a counter that counts the number of alerts and stop decreasing the supply voltage when the number exceeds a predefined threshold value, as shown in the figure. In this example, the threshold value is two. After a predetermined period passes by, the counter is reset and thus the DVS technique begins to work again.

C. Core-to-Core Variations on MPSoC

WID variations cause different performance and power characteristics across a single die. They consist of systematic and random components. Systematic WID variations exhibit a spatially correlated behaviour and dominate the characteristics of a MultiCore chip [9]. In contrast, random WID variation fluctuates independently from transistor to transistor. The spatially correlated variations behave as C2C (Core-to-Core) variations on a die [9]. This is because each core is small enough that the WID variations affect across cores rather than within the core. Hence, this paper assumes that individual cores have different performance and power characteristics, although they have identical design and environment.

Figure 4 and 5 explain how C2C variations affect performance and power consumption. Figure 4 shows how delays of circuits in a sampled core distribute. The horizontal and vertical axes indicate circuit delay and its corresponding frequency, respectively. The curved line is the frequency distribution of circuit delay in the core. Its right end indicates the critical path delay in the core. The dashed line explains the target delay of the core, which is defined as the specification. Hence, the distance between the target frequency and the critical path delay is the safety design margin of the core. C2C variations move the curved line to the right or to the left. If it is moved to the right, the design margin is decreased. Otherwise, it is increased.
Figure 5 shows the frequency distribution of the critical path delays across all fabricated cores. The horizontal and vertical axes are referred to as delay and its corresponding frequency, respectively. The dashed line indicating the typical critical path delay; that means C2C variations does not affect the critical path delay. If a sampled core’s critical path delay is larger than the dashed line, the core’s safety margin is smaller than the mean. Otherwise, the margin is larger than the mean.

This performance and power asymmetry among cores due to C2C variations creates serious problems on program allocation. Throughput may be diminished, real-time deadline may be missed, and excessive thermal throttling may be triggered [9, 20]. In the case of Canary-DVS, the asymmetry affects both energy reduction and performance. When a sampled core’s safety margin is small, the room for reducing energy consumed by the core is also small. Since we chose per-chip DVS, the room for reducing energy consumed by the chip is also small. In addition, when the margin is small, timing errors are frequently predicted, which results in frequent voltage switch and thus in longer execution cycle. Figure 6 shows an example of how C2C variations result in different energy reduction, where four applications are running on a quad-core processor. The evaluation methodology will be explained later. In this example, only one core has a small safety margin due to slow transistors and the other cores have the typical margin. The horizontal axis indicates the combinations of programs, which will be explained in Table I. The vertical axis indicates how Canary-DVS reduces energy consumption in percentage. The figure shows how the average ratio between maximum and minimum total chip energy consumption changes with different allocations. Surprisingly, more than 5% discrepancy arises among almost all combinations.

In our previous study [18], all cores on a die have the typical safety margin. In this paper, C2C variations are considered and individual cores have different margins.

IV. EVALUATION METHODOLOGY

First in this section, we introduce our simulation environment that evaluates performance and power. Next, we explain how C2C variations are considered in our evaluations.

A. Performance and Power Simulation

We adopt Canary-DVS to Toshiba’s quad-core Media embedded Processor (MeP) [14, 15]. MeP simulator provided by Toshiba Corporation is used to generate execution traces. It is a cycle-based simulator and models quad-core MeP processor in details. We use Stanford Integer Benchmarks; bubble is a program sorting an array using Bubble-sort, matmul is a program multiplying two matrices, perm is a heavily recursive permutation program, qsort is a program sorting an array using Quick-sort, queen is a program solving the eight queens problem, and sieve is a prime sieve of Erasthones program. Since we use quad-core MeP, the combinations of the programs running on the processor is 6C4=15. The explanations of the combinations are summarized in Table I.

<table>
<thead>
<tr>
<th>Table I. COMBINATIONS OF 4 PROGRAMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>bubble matmul perm qsort queen sieve</td>
</tr>
<tr>
<td>BuMaPeQs X X X X</td>
</tr>
<tr>
<td>BuMaPeQu X X X X</td>
</tr>
<tr>
<td>BuMaPeQsSi X X X X</td>
</tr>
<tr>
<td>BuMaQsQu X X X X</td>
</tr>
<tr>
<td>BuMaQsSi X X X X</td>
</tr>
<tr>
<td>BuPeQsQu X X X X</td>
</tr>
<tr>
<td>BuPeQsSi X X X X</td>
</tr>
<tr>
<td>BuPeQsQu X X X X</td>
</tr>
<tr>
<td>MaPeQsQu X X X X</td>
</tr>
<tr>
<td>MaPeQsSi X X X X</td>
</tr>
<tr>
<td>MaQsQsSi X X X X</td>
</tr>
<tr>
<td>PeQsQsSi X X X X</td>
</tr>
</tbody>
</table>

Each trace is injected into the trace-driven simulator we built. The details of Canary-DVS are implemented. It is true that Canary FF has serious area overhead and thus consumes large power. However, our selective replacement method minimizes the number of Canary FFs [12, 23] and hence the area and power overheads are negligible. Based on the assumption that the yield of pipeline is mainly determined by the timing error in the execution stage [13], we observe the length of carry in ALUs. If the carry is longer than the threshold value that determined by the supply voltage, a timing error is predicted. The combination of the threshold and the voltage is estimated by the combination of the clock frequency and the supply voltage of Intel Pentium M [10], which is shown in the left and the middle of Table 2. We also consider the rule of thumb; PVT variations require 50-100% design margins [18]. We assume 50% margin in this evaluation. The thresholds are also summarized in the right of Table II. We assume that the carry propagation delay determines the threshold. Considering the 50% margin, 48-bit (= 32-bit * 1.5) carry propagation is safe at the highest power supply. Since we use 32-bit ALUs, this is an exaggerated specification. By assuming that the safe carry propagation is proportional to the supply voltage, the right
column of Table 2 is calculated. For example, the carry longer than 13 bits at 0.988V signals an error prediction.

We assume the DVS interval between supply voltage switching is 1K clock cycles and also assume every supply voltage switching requires a penalty of 100 clock cycles. Please remember that Canary-DVS does not scale clock frequency and thus the overhead is constant value in absolute time. Figure 7 illustrates the switching interval and the overhead. The horizontal and vertical lines present time and supply voltage, respectively. The threshold value of 2 in the Canary alert sequence is selected for stopping the supply voltage switching. The cycles where supply voltage reduction is prohibited is determined so as to the impact of switching on performance is less than 2%.

<table>
<thead>
<tr>
<th>Supply (V)</th>
<th>Freq (GHz)</th>
<th>Threshold (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.340</td>
<td>2.1</td>
<td>32</td>
</tr>
<tr>
<td>1.260</td>
<td>1.8</td>
<td>32</td>
</tr>
<tr>
<td>1.228</td>
<td>1.6</td>
<td>32</td>
</tr>
<tr>
<td>1.180</td>
<td>1.4</td>
<td>32</td>
</tr>
<tr>
<td>1.132</td>
<td>1.2</td>
<td>27</td>
</tr>
<tr>
<td>1.084</td>
<td>1.0</td>
<td>22</td>
</tr>
<tr>
<td>1.036</td>
<td>0.8</td>
<td>18</td>
</tr>
<tr>
<td>0.988</td>
<td>0.6</td>
<td>13</td>
</tr>
</tbody>
</table>

For every chip sample, $4! = 24$ trace-based simulations, which are explained in the previous section, are necessarily to consider the impact of program allocation. In total, $15 \times 24 \times 10,000 = 3,600,000$ simulations are performed. Finally, the impact of C2C variations on performance and energy is analysed from the simulation results.

V. Results

Figure 8 presents how Canary-DVS reduces energy consumption when the typical design margin is assumed; that is when C2C variations are not considered [18]. The horizontal and vertical axes indicate names of program combinations and the percentage energy reduction rate, respectively. For all combinations, more than 20% energy reduction is attained and the average energy reduction rate is 21.2%. Canary-DVS successfully eliminates the energy waste causes by the overestimated power supply voltage in the case of typical critical path delay. When we compare it with Figure 6, we find that for almost all the combinations the maximum energy reduction shown in Figure 6 is comparable to the energy reduction in the typical case. This means that a MultiCore processor could tolerate C2C variations that negatively affect only one of four cores when the programs are optimally allocated.

B. Variations Consideration

The influence of C2C variations are evaluated by Monte Carlo simulations. For each combination of programs, we generate 10,000 sample chips, whose critical path delays are distributed according to a normal distribution. Its $\sigma/\mu$ value is estimated as follows. The typical $\sigma/\mu$ value in gate delay for 65nm technology, where $\sigma$ is the standard deviation and $\mu$ is the mean delay, is 0.064 [1]. We use it to estimate the $\sigma/\mu$ value in the critical path delay of each core. As explained in the previous section, C2C variations are modelled by the delay variations of ALUs and thus several adders are analysed to estimate the $\sigma/\mu$ value [22]. 10,000 samples are generated for every statistical static timing analysis of adders. It is found that $\sigma/\mu$ value in circuit delay is almost the same for all adders and is 0.018. It is also found that the delay is distributed almost according to the normal distribution. Hence, the normal distribution with 0.018 of $\sigma/\mu$ value in core critical path delay is assumed.

Figure 9 presents the percentage performance loss. The average rate is 1.34% and the loss is negligibly small for all combinations.

Figure 10 shows the percentage energy reduction rate when C2C variations are considered. Its layout is the same to that of Figure 6 and the average, maximum, and minimum are shown. Across all combinations, the average energy reduction rate is 18.6%. This is 2.6 percentage point down from the typical critical path delay case. There is not significant discrepancy between the cases where C2C variations are considered or not. Hence, we’d like to conclude that Canary-DVS efficiently decreases the overestimated power supply voltage and thus eliminates the energy waste. In addition, in the worst cases, approximately 15% of energy reduction is achieved for all combinations. Even if we do not optimize program allocation as in [20], considerable energy reduction is possible. We’d like to
conclude again that Canary-DVS reduces the energy waste under the serious C2C variations.

VI. CONCLUSIONS

Due to the aggressively advanced semiconductor technology, the serious problems of PVT variations are emerging. Even though chips have identical design, some of them may violate timing specifications. This situation requires a large safety margin in the power supply voltage and thus energy consumption is uselessly increased. In order to eliminate the energy waste, we have proposed Canary-DVS. In the previous study, where the typical path delay is only considered, we confirmed that Canary-DVS successfully reduces the waste. In this paper, we consider C2C variations and confirm again that Canary-DVS is a promising technique that eliminates the overestimated safety margin without any severe performance loss. From the extensive Monte Carlo simulations, it is found that energy is reduced by 18.6% on average and performance loss is 1.36% on average, when 0.064 of $\sigma/\mu$ value is assumed in gate delay, which is the typical value in 65nm technology.

One of the future directions regarding Canary-DVS on MPSoCs is investigating optimal program allocation methods. As we have seen in Section 3.3, different program allocations result in different energy reduction and the best allocation almost completely tolerates C2C variations in energy efficiency. This is an interesting research topic.

ACKNOWLEDGMENT

This work is supported in part by JST CREST program, by JSPS Grant-in-Aid for challenging Exploratory Research, and by the fund from Central Research Institute of Fukuoka University. The authors would like to thank Shunitsu Kohara of Toshiba Corporation for helping them use MeP simulator.

REFERENCES


