Critical Issues Regarding A Variation Resilient Flip-Flop

Sato, Toshinori
System LSI Research Center, Kyushu University

Kunitake, Yuji
Graduate School of Computer Science and System Engineering, Kyushu Institute of Technology

http://hdl.handle.net/2324/8313

出版情報：The 14th Workshop on Synthesis And System Integration of Mixed Information technologies, pp.280-286, 2007-10-15
バージョン：
権利関係：

Critical Issues Regarding A Variation Resilient Flip-Flop

Toshinori Sato
System LSI Research Center
Kyushu University
toshinori.sato@computer.org

Yuji Kunitake
Department of Artificial Intelligence
Kyushu Institute of Technology
y-kunitake@klab.ai.kyutech.ac.jp

Abstract - Deep submicron technologies increase parameter variations, which will make microprocessor designs very difficult since every variation requires a large safety margin for achieving specified timing yield. This means higher supply voltage, which results in large energy consumption. Razor flip-flop (FF) is a clever technique to eliminate the supply voltage margin by exploiting circuit-level timing speculation. It combines dynamic voltage scaling technique with the error detection and recovery mechanism. This paper presents an improvement of Razor FF in removing delayed clock, which complicates timing design. It is named canary FF. This paper discusses critical issues regarding the canary FF. When the issues were solved, the canary FF would achieve 10% of power reduction by exploiting input value variations, while further power reduction by eliminating design margins is expected.

Key words: variations, low-power, DVS, Razor, microprocessors

I Introduction

Due to the aggressive technology scaling, parameter variations are increasing and thus have become a serious problem in processor designs [4, 12, 22]. Generally, processor's maximum clock frequency is determined by considering the worst-case critical-path delay and a safety margin. The margin is required since delays are not constant due to parameter variations. Parameter variations include process, voltage, and temperature (PVT) variations, each of which requires the margin. In the worst-case design, the critical path delay and the margins are summed up and thus PVT variations have a serious impact on supply voltage to satisfy required operating frequency and to improve timing yield of microprocessors. In other words, managing parameter variations is a key to power reduction.

Razor [7, 9] is an adaptable dynamic voltage scaling (DVS) technique, which uses a timing-error tolerant flip-flop (FF) to scale the supply voltage to the point where every margin described above is eliminated. This allows supply voltage reduction, resulting in significant energy savings. While Razor is a smart technique for energy reduction, it has a complexity in circuit implementations; delayed clock. This paper introduces canary logic, an improvement of the Razor. In our previous study [19], adopting the canary logic on carry select adder showed the potential in power reduction of 30%. This paper investigates critical issues on the canary logic and presents preliminary simulation results for the entire processor.

This paper is organized as follows. Section II describes Razor with some related works. Section III introduces the canary logic. Section IV discusses critical issues on the canary FF. Section V presents experimental results. Finally, Section VI concludes.

II. Razor

Razor [7, 9] 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. Figure 1 shows the Razor's DVS system. If the error rate is low, it indicates that the supply voltage could be decreased. On the other hand, if the rate is high, it indicates that the supply voltage should be increased. The control system works to maintain a predefined error rate, E_{ref}. At regular intervals the error rate, E_{sample}, is computed and the rate differential, $E_{diff} = E_{ref} - E_{sample}$, is calculated. If the differential is positive, it indicates that supply voltage could be decreased. The otherwise indicates that the supply voltage should be increased.

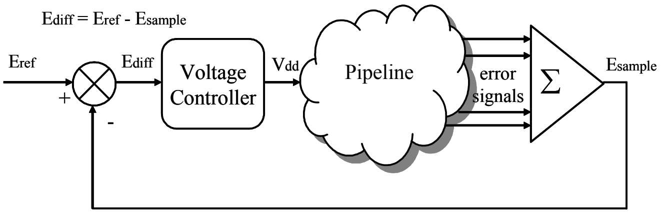

Fig. 1. Razor's DVS System

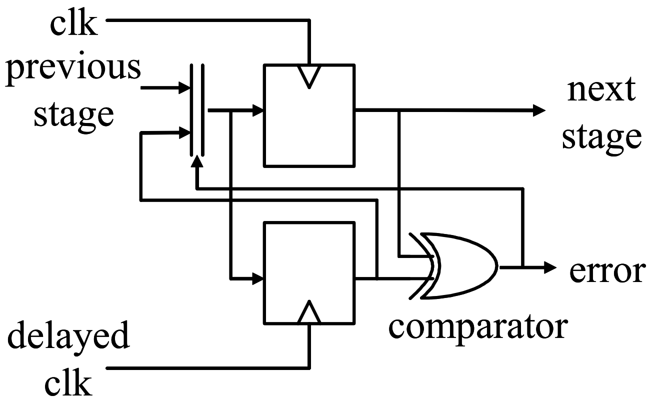

Fig. 2. Razor Flip-Flop

In order to detect timing errors, Razor FF shown in Fig.2 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. When the timing error is detected in microprocessor pipelines, the processor state is recovered to a safe point where the error occurs. One of the difficulties in Razor is how it is guaranteed that the shadow FF could always latch correct values. The delayed clock has to be carefully designed considering so-called short path problem [9]. Delay buffer is inserted to solve the problem [7]. This makes set-up-time constraint of the shadow FF severer, resulting in smaller timing margin.

A. Related Work

Li et al. [15] improve the robustness of the Razor FF and evaluate it by a superscalar processor design.

iRoC Technologies [8, 18] proposes to utilize the shadow FF to detect soft errors. Two implementations are considered. One is very similar to the Razor FF and requires delayed clock. The other does not require delayed clock, but the input to the shadow FF is delayed. When two values stored in the main and shadow FFs do not match, a soft error is detected. By adjusting the delay, maximum transient-pulse duration can be changed.

NEC [17] proposes to utilize the shadow FF to predict wearout failures. Every combinational logic block is duplicated and a failure part of the main circuit is switched into its redundant copy. In order to predict the failure, defect-prediction FF is proposed. It utilizes the shadow FF. There is a delay line between the previous logic stage and the shadow FF and the shadow FF might violate timing constraints even when the main FF does not. Hence, by comparing values stored in the main and shadow FFs, the increase in the path delay due to the wearout can be predicted.

Agarwal et al. [1] propose a similar technique with NEC's defect-prediction FF. Intel [23] also proposes the similar technique, which is an extension of the soft-error resilient FF [16] in order to support process variation diagnosis.

Another proposal from Intel [2] utilizes the shadow latch to detect timing error caused by parameter variations[†]. It does not share the value delivered to the main latch, but holds the value that passes over the main latch. Hence, timing constraint is severer in the shadow latch than in the main latch. When these latches are closed and two values do not match, a timing error is detected.

To the best of our knowledge, the work by Kehl [13] is the first that looks at using the basic delaying mechanism, though for a different purpose. Incoming data is sampled several times at different sampling rates. All samples are compared with the incoming data and the comparison results are used to adjust clock frequency.

Calhoun et al. [6] propose canary FF, which is easy to fail at higher supply voltage than other FFs on the critical paths. It is used for reducing the supply voltage during standby mode. The supply voltage is reduced until the canary FF is fail, resulting in power saving due to smaller leakage current.

III. Canary Logic

While Razor is a smart technique to eliminate design margins, its circuit implementation could be further improved. We replace the Razor FF with our proposed one, which we call canary FF [19]. We accidentally named this FF "canary FF", since we did not notice the existence of the canary FF proposed in [6] when we started the present study. We hope the readers are not confused.

A. Canary Flip-Flop

The canary FF is augmented with a delay element and the shadow FF, as shown in Fig.3. The canary FF is used as a canary in a coal mine to help detect whether a timing error is about to occur. Timing errors are predicted by comparing the main FF value with that of the shadow FF, which runs into the timing error a little bit before the main FF. Alert signal triggers voltage or frequency control. Utilizing the canary FFs has the following three advantages.

- **Elimination of the delayed clock:** Using single phase clock significantly simplifies clock tree design. It also eliminates the short path problem [9] in the Razor FF, and hence its minimum-path length constraint should not be considered.
- **Protection offered against timing errors:** As explained above, the shadow FF protects the main FF against timing errors. This freedom from timing errors eliminates any complex recovery mechanism. The selector placed in front of the main FF is removed, leading that some timing pressure is relaxed. Instead, the signal generated by the comparator triggers voltage or frequency control. If the timing error is alerted, the supply voltage stops falling or the clock frequency is felt down.
- **Robustness for variations:** The canary FF is variation resilient. The delay element always has a positive delay, even though parameter variations affect it. Hence, the shadow FF always encounters a timing error before the main FF.

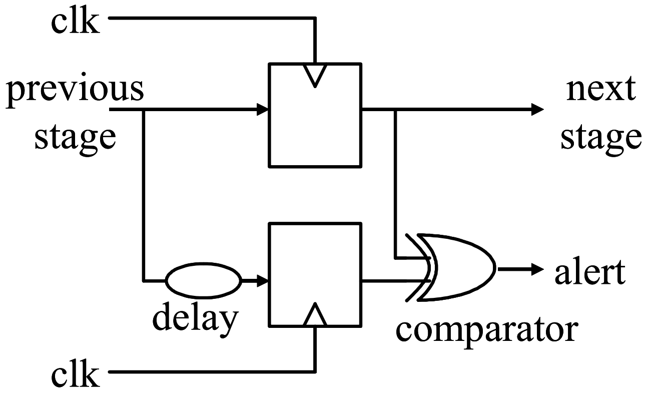

Fig. 3. Canary Flip-Flop

B. Canary FF Implementation via Scan Reuse

Scan resources, which is implemented for production testing, can be reused to realize the canary FF. Figure 4 shows a scan FF design [16] that consists of a system FF (the lower part in the figure) and a scan portion (the upper part in the figure). The SI input is connected to the SO output of the next scan FF to be a shift register. In the test mode, clocks SCA and SCB are applied to shift a test pattern into latches LA and LB.

[†] The variation resilient latch is not described in the paper [2]. But, it is introduced in the presentation slide.

Next, the UPDATE clock is applied to write the test pattern in LB into the system latch, PH1. Then, the CLK clock is applied to capture the system response to the test pattern. After that, the CAPTURE signal is applied to move the contents of PH1 to LA. And last, clocks SCA and SCB are applied to shift the system response out.

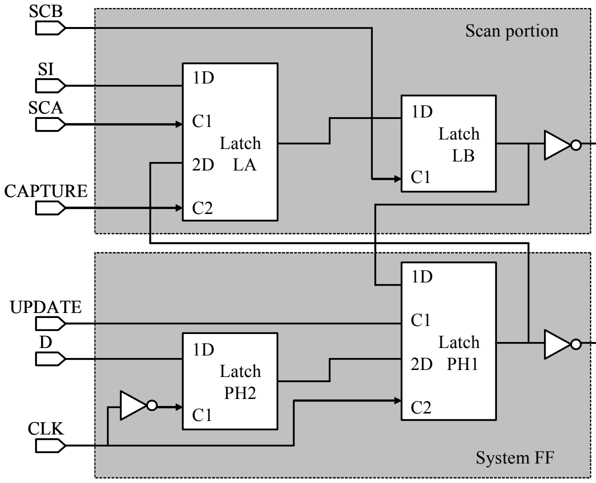

Fig. 4. Scan Cell [16]

In system operation mode, latches LA and LB are not utilized. The canary FF can be implemented with a little hardware cost by reusing the latches in the scan portion. Figure 5 shows how reusing scan FF design realizes the canary FF. The FF design's test mode operation is identical to the design in Fig.4. In system operation mode, latches LA and LB hold the replicas of PH2 and PH1, respectively. If any timing error does not occur, the ALERT signal is low and thus the delayed signal of D is written into LA. Once a timing error is detected, the value complementary to D is stored in LA, resulting in keeping the failure state. The reuse is possible since the canary logic does not require delayed clock. This is not possible in the case of Razor.

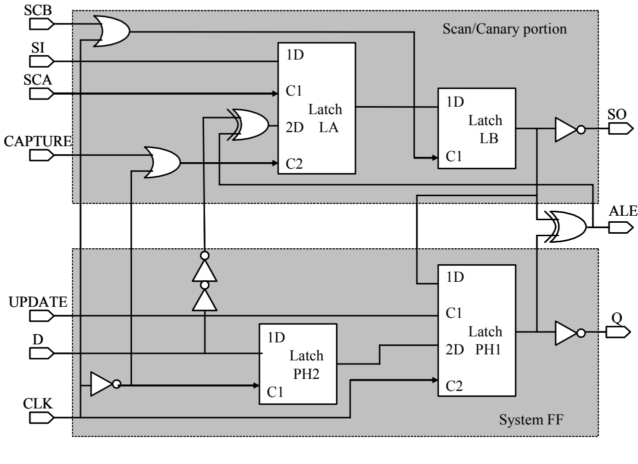

Fig. 5. Canary's Scan Cell

C. *Power Reduction with Canary FFs*

Figure 6 explains how DVS techniques utilize the canary FFs. The horizontal and vertical lines present time and supply voltage, respectively. At regular intervals, the supply voltage is decreased if a timing error is not predicted during the last interval. This is possible since input values activating the critical path are limited to a few variations. Timing errors rarely occur even if the timing constraints on the critical path are not satisfied. The input value variations can be exploited to decrease the supply voltage. Because the supply voltage is lower than that determined by the critical path delay, significant power reduction is achieved in the canary logic as in Razor [7, 9]. When a timing error is predicted to occur, the supply voltage is increased.

There are two strategies to increase power supply voltage. One is called STEP strategy, which increase supply voltage to the next higher one as shown in Fig.6. The other is called RESET strategy, which increase supply voltage to the highest one. In the following section, we will evaluate which one is more energy efficient.

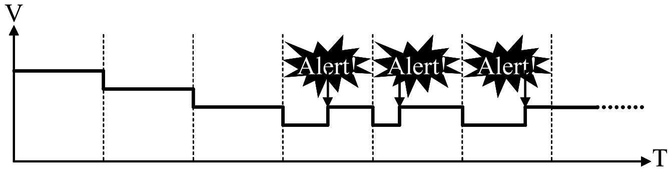

Fig. 6. Canary's DVS (STEP strategy)

IV. Critical Issues on Canary Logic

In order to make the canary logic implementable, there are several open issues, which should be solved.

A. *Metastability of Shadow FFs*

Since timing constraints at the shadow FF are not always respected, its state might be metastable. Metastability might cause logical errors as well as increases circuit delay. This is the most critical issue shared by the redundant FFs listed in Section II-*A*. In Razor, metastability need not be resolved correctly. It is only required to detect a metastable state and to treat it as a timing error. In the canary logic, the situations are same. The metastability detector in Razor [9] can be used. Good news is that the detection of metastability can be delayed since it is expected that the main FF always holds correct values. In contrast, Razor requires fast detection since error signals are used to select input values to Razor FFs.

B. *Timing Error in Main FFs*

In the canary logic, it is expected that the main FF always latches correct values. However, it is not always true. Once critical path delay is speculatively violated, there is a possibility that a timing error occurs in both the main and shadow FFs. We do not assume any sudden change in critical path delay. We think this is a practical assumption if the condition of frequency and supply voltage is gradually changed and if the delay element is carefully designed. Nonetheless, a safety net is necessary and it is a topic of our future study. On the other hand, if the canary FF is used to predict circuit failures due to aging as in [1, 17], this issue becomes less serious.

C. *Delay Element Design*

As mentioned above, the delay element design is very important since its delay value determines the robustness of the canary FF. A critical issue is that the delay value will change according to the supply voltage, if the delay element is realized as an active circuit. Good news is that the delay will increase as the supply voltage is reduced. In other words, the more aggressive the timing speculation is, the more conservative the timing-error prediction is. The voltage dependency of the delay value is safe.

D. *Power Consumed by Additional Circuits*

The additional circuits to implement the canary FFs consume power, even though the scan reuse does not require severe hardware cost. Power consumed by the shadow FFs and the delay elements might be significant. In Razor, power consumed by the shadow FFs and delay buffers occupies 3% of total power consumption [7]. Hence, we expect the power overhead will be small.

E. *Collection of Error Predictions*

The canary's DVS system has to collect error predictions from all canary FFs. This is because just one error prediction requires the change in supply voltage. The on-chip network for the collection might have serious impact on chip area. It also might have a large latency, which has severe impact on cycle time. For the former issue, we are currently investigating to reuse scan network. For the latter issue, we expect that multiple cycle latency will be tolerable as long as the main FFs hold correct values as explained in Section IV-*A*.

We are also considering Multiple Voltage Domain (MVD) microarchitecture, which is a variant of Multiple Clock Domain (MCD) microarchitecture [21]. MVD is different from MCD in that clock frequency in MVD is unique in all domains, and it is not scaled even if supply voltage is changed. When a canary FF predicts an error, just one domain including the FF increases its supply voltage. MVD eliminates the requirement of the large on-chip network.

V. Evaluation

In this section, we show how canary FF can exploit input value variations to reduce energy consumption when the critical issues listed above are solved. First, we show how timing error rate is determined. Second, we describe architectural-level simulation environment. And last, we present preliminary simulation results.

A. *Timing Error Rate*

We estimate timing error rates of the entire microprocessor using a 32b carry select adder (CSLA), since the yield of pipeline is mainly determined by the timing error in execution stage [15]. SYNOPSIS DesignCompiler logic-synthesizes the CSLA with Hitachi 0.18um standard cell libraries. The combinations of the clock frequency and the supply voltage of Intel Pentium M [11], which is shown in TABLE I, are used. We project the highest clock frequency, which is determined by CSLA's critical path delay reported by DesignCompiler, onto Pentium's highest clock frequency. In order to estimate how timing error occurs, we simulate the CSLA using Cadence Verilog-XL simulator. Gate-level simulation results are shown in Fig.7. It is observed that supply voltage reduction down to 1.18V suffers little timing errors. We use the timing error rates in architectural-level simulations explained in the next section.

TABLE I
Frequency – Voltage Specifications

F(GHz) V_{dd}(V)	2.1 1.340	1.8 1.276	1.6 1.228	1.4 1.180
F(GHz) V_{dd}(V)	1.2 1.132	1.0 1.084	0.8 1.036	0.6 0.988

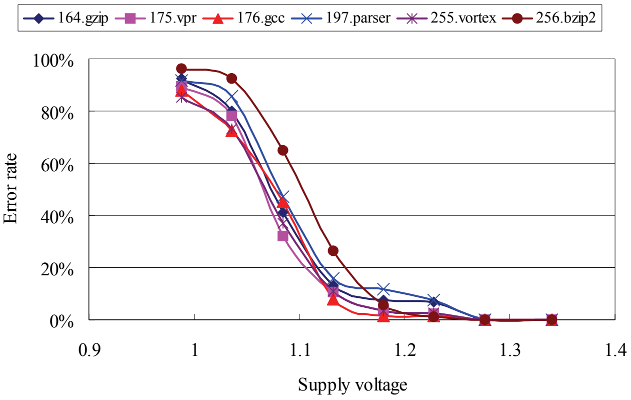

Fig. 7. Error Rate - V_{dd}

B. *Architectural-level Simulation Environment*

SimpleScalar/PISA tool set [3, 5] is used for architectural-level simulation. TABLE II summarizes processor configurations. Six integer programs from SPEC2000 CINT benchmark are used. For each program, 1 billion instructions are skipped before actual simulation begins. After that each program is executed for 2 billion instructions.

We evaluate three intervals between supply voltage scaling, which are 100K, 1M, and 10M clock cycles. It is assumed every supply voltage switching requires 10µs [10].

We do not know how large safety margin every variation requires, since we do not perform a real processor design. Therefore, in the evaluations here, we consider how we can exploit input value variations for energy reduction. Further energy reduction must be expected when we eliminate design margins required by other parameter variations.

As explained above, we model the timing error rate of the entire processor based on that of the CSLA. We assume that timing errors occur at random at the error rate shown in Fig.7 in every supply voltage. Please note that actual input variations are not considered. In our previous study on the other circuit-level timing speculation technique, it was found that performance results with and without considerations of actual input variations did not show any significant difference [14]. Hence, we expect that the evaluation methodology in the present paper is enough accurate for the preliminary evaluation.

TABLE II
Processor Configurations

Clock frequency	2 GHz
Fetch width	8 instructions
L1 instruction cache	16K, 2 way, 1 cycle
Branch predictor	gshare + bimodal
gshare predictor	4K entries, 12 histories
Bimodal predictor	4K entries
Branch target buffer	1K sets, 4 way
Dispatch width	4 instructions
Instruction window size	128 entries
Issue width	4 instructions
Integer ALUs	4 units
Integer multiplires	2 units
Floating ALUs	1 unit
Floating multiplires	1 unit
L1 data cache ports	2 ports
L1 data cache	16K, 4 way, 2 cycles
Unified L2 cache	8M, 8 way, 10 cycles
Memory	Infinite, 100 cycles
Commit width	8 instructions

C. Results

Figure 8 shows which supply voltage is selected during program execution. It is observed that only three of eight voltage modes are used.

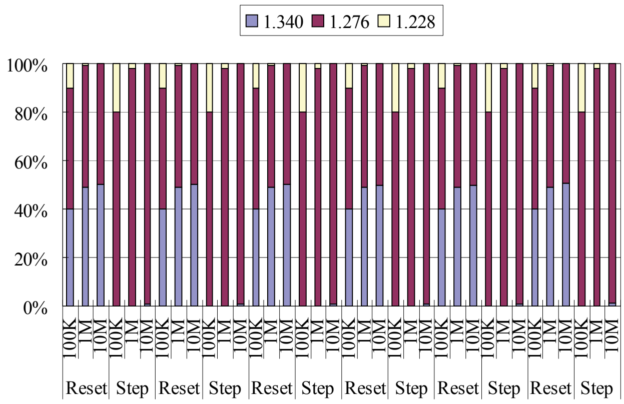

Fig. 8. Breakdown of Supply Voltage

First, we compare RESET and STEP strategies. Regardless of programs and intervals, STEP strategy selects lower supply voltage more frequently. This matches what we expected, since RESET strategy selects the highest supply voltage when a timing error is predicted. STEP strategy will achieve larger energy reduction than RESET strategy.

Next, we consider how the length of the intervals affects voltage selection. It is observed that shorter interval tends to use lower voltage more frequently. This also matches what we expected, since large interval increases the period before the next lower supply voltage is selected. On the other hand, shorter interval will increase overhead on performance since it switches supply voltage more frequently.

Figure 9 shows the percentage increase in execution cycles.

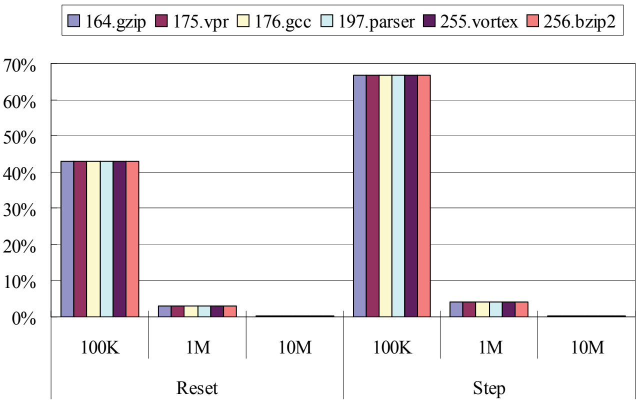

Fig. 9. Percentage Increase in Execution Cycles

First, the influence of program characteristics is considered. As can be easily seen, different programs have a similar impact on performance. This matches what we expected, since Fig.7 does not show considerable difference in timing error rate among programs.

Second, we compare STEP and RESET strategies. Regardless of programs and intervals, STEP strategy suffers large performance penalty than RESET strategy does. Referring back to Fig.8, it is observed that STEP strategy uses lower supply voltage more frequently than RESET strategy does. This implies that STEP strategy often encounters timing error, and thus often changes supply voltage, resulting in large switching overhead. While, in the previous section, we expected that STEP strategy might achieve larger energy reduction than RESET strategy, its large performance penalty will mitigate the energy savings.

Next, we consider how the length of intervals affects performance. Regardless of programs and strategies on voltage increase, longer intervals have less impact on performance. Shorter intervals increase the number of voltage switching, resulting in larger performance overhead.

Figure 10 presents the percentage reduction in energy consumption. It does not include power consumed by the working scan circuits and the delay elements.

First, there is not any significant difference among programs, while absolute values are different.

Second, the influence of intervals is considered. As we expected in the previous section, 100K cycle of interval suffers the most serious penalty due to voltage switching. For all programs, energy consumption is increased as much as 50%. 10M cycle of interval suffers the least penalty and achieves approximately 10% of energy reduction.

Next, we compare RESET and STEP strategies. STEP strategy always achieves larger energy reduction than RESET strategy does, regardless of intervals and programs. While STEP strategy suffers larger performance penalty than RESET strategy does, the former consumes less power than the latter does since the former uses lower supply voltage more frequently than the latter does.

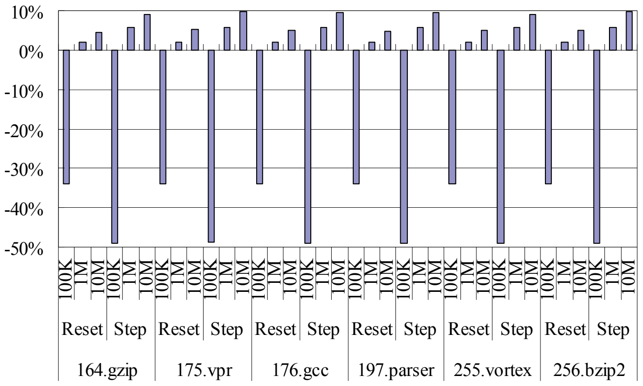

Fig. 10. Percentage Reduction in Energy Consumption

Figure 11 shows the percentage reduction in energy, energy-delay product (EDP), and energy-delay-square product (ED2P). For every metric, STEP strategy with 100M of interval marks the best score regardless of programs. It is observed that overhead due to voltage switching is significant.

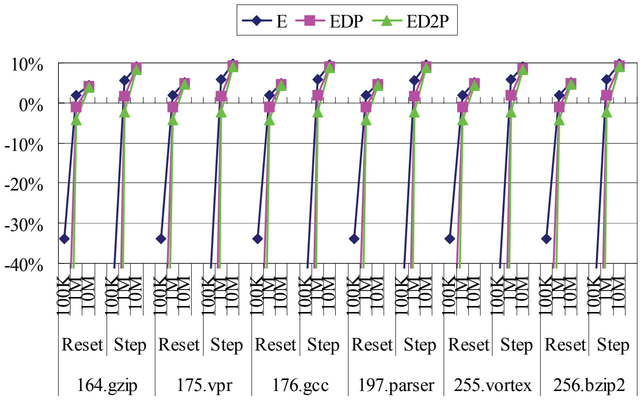

Fig. 11. Percentage Reduction in Energy (E), E-Delay (D) Product (P), and ED^{2}P

The voltage scaling strategy in the present paper shows an oscillation in supply voltage. After the first timing error is predicted, timing errors is repeatedly alerted, as shown in Fig.12. Since every supply voltage switching makes processor unavailable during the transition, this oscillation has a serious impact on performance and on power efficiency. A solution to prevent the oscillation can be found in [20].

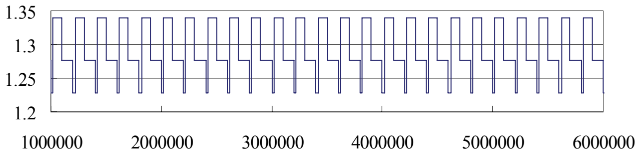

Fig. 12. Oscillation in Supply Voltage (164.gzip)

VI. Conclusions

As the complexity of the semiconductor manufacturing process increases, it is likely that parameter variations will be more difficult to control. Under the situations, deep submicron semiconductor technologies will make the worst-case design impossible, since they can not provide design margins that it requires. In order to attack the problem, we proposed the canary logic as an alternative of Razor, which is a smart technique to eliminate design margins. The canary logic eliminates the delayed clock required by the Razor logic, resulting in easy design. The canary FF relies on the delay element, which always has a positive delay, and hence they are variation resilient.

In this paper, we utilize the canary logic for energy reduction by exploiting input value variations. Since timing errors are expected to rarely occur even if the timing constraints on the critical path are not satisfied, input value variations can be exploited to decrease the supply voltage. Because the supply voltage is lower than that determined by the critical path delay, significant power reduction is achieved. From the detailed simulation results, we found that up to 10% of the potential energy reduction. Since PVT variations usually require 50-100% design margins [24], further energy reduction can be expected.

Acknowledgment

We gratefully acknowledge comments and advices provided by the members of the SoC Laboratory of Kyushu University. Hitachi 0.18um standard cell libraries are provided by VDEC (VLSI Design and Education Center) in the University of Tokyo. This work is partially supported by the CREST (Core Research for Evolutional Science and Technology) program of Japan Science and Technology Agency (JST), and by Grant-in-Aid for Scientific Research (KAKENHI) (A) #19200004 from Japan Society for the Promotion of Science (JSPS).

References

[1] M. Agarwal, B. C. Paul, M. Zhang, and S. Mitra, “Circuit Failure Prediction and Its Application to Transistor Aging,” 25th VLSI Test Symposium, 2007.

[2] M. Annavaram, E. Grochowski, and P. Reed, “Implications of Device Timing Variability on Full Chip Timing,” 13th International Symposium on High-Performance Computer Architecture, 2007.

[3] T. Austin, E. Larson, and D. Ernst, “SimpleScalar: an Infrastructure for Computer System Modeling,” IEEE Computer, Vol. 35, No. 2, 2002.

[4] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, “Parameter Variations and Impact on Circuits and Microarchitecture,” 40th Design Automation Conference, 2003.

[5] D. Burger and T. M. Austin, “The SimpleScalar Tool Set, Version 2.0,” ACM SIGARCH Computer Architecture News, Vol. 25, No. 3, 1997.

[6] B. H. Calhoun and A. P. Chandrakasan, “Standby Power Reduction Using Dynamic Voltage Scaling and Canary Flip-Flop Structures,” IEEE Journal of Solid-State

Circuits, Vol. 39, No. 9, 2004.

[7] S. Das, P. Sanjay, D. Roberts, L. S. Lee, D. Blaauw, T. Austin, T. Mudge, and K. Flautner, "A Self-Tuning DVS Processor Using Delay-Error Detection and Correction," Symposium on VLSI Circuits, 2005.

[8] E. Dupont, M. Nicolaidas, and P. Rohr, "Embedded Robustness IPs for Transient-Error-Free ICs," IEEE Design & Test of Computers, Vol. 19, No. 3, 2002.

[9] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," 36th International Symposium on Microarchitecture, 2003.

[10] S. Gochman, R. Ronen, I. Anati, A. Berkovits, T. Kurts, A. Naveh, A. Saeed, Z. Sperber, and R. C. Valentine, "The Intel Pentium M Processor: Microarchitecture and Performance," Intel Technology Journal, Vol. 7, No. 2, 2003.

[11] Intel Corporation, "Intel Pentium M Processor on 90nm Process with 2-MB L2 Cache," Datasheet, 2006.

[12] T. Karnik, S. Borkar, and V. De, "Sub-90nm Technologies: Challenges and Opportunities for CAD," International Conference on Computer Aided Design, 2002.

[13] T. Kehl, "Hardware Self-Tuning and Circuit Performance Monitoring," International Conference on Computer Design, 1993.

[14] Y. Kunitake, A. Chiyonobu, K. Tanaka, and T. Sato, "Challenges in Evaluations for a Typical-Case Design Methodology," 8th International Symposium on Quality Electronic Design, 2007.

[15] H. Li, Y. Chen, K. Roy, and C.- K. Koh, "SAVS: A Self-Adaptive Variable Supply-Voltage Technique for Process-Tolerant and Power-Efficient Multi-Issue Superscalar Processor Design," 11th Asia and South Pacific Design Automation Conference, 2006.

[16] S. Mitra, N. Seifert, M. Zhang, Q. Shi, and K. S. Kim, "Robust System Design with Built-In Soft-Error Resilience," IEEE Computer, Vol. 38, No. 2, 2005.

[17] T. Nakura, K. Nose, and M. Mizuno, "Fine-Grain Redundant Logic Using Defect-Prediction Flip-Flops," International Solid-State Circuits Conference, 2007.

[18] M. Nicolaidis, "Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies," 17th VLSI Test Symposium, 1999.

[19] T. Sato and Y. Kunitake, "A Simple Flip-Flop Circuit for Typical-Case Designs for DFM," 8th International Symposium on Quality Electronic Design, 2007.

[20] T. Sato and Y. Kunitake, "Exploiting Input Variations for Energy Reduction," 17th International Workshop on Power and Timing Modeling, Optimization and Simulation, 2007.

[21] G. Semeraro, D.H. Albonesi, S.G. Dropsho, G. Magklis, S. Dwarkadas, and M.L. Scott, "Dynamic Frequency and Voltage Control for a Multiple Clock Domain Microarchitecture," 35th International Symposium on Microarchitecture, 2002.

[22] O. S. Unsal, J. W. Tschanz, K. Bowman, V. De, X. Vera, A. Gonzalez, and O. Ergin, "Impact of Parameter Variations on Circuits and Microarchitecture," IEEE Micro, Vol. 26, No. 6, 2006.

[23] M. Zhang, TM Mak, J. Tschanz, K. S. Kim, N. Seifert, and D. Lu, "Design for Resilience to Soft Errors and Variations," 13th International On-Line Testing Symposium, 2007.

[24] Private communications with LSI designers.