

MultiCore Energy Reduction Utilizing Canary FF

Otsuka, Yoshimi
Department of Engineering, Fukuoka University

Sato, Toshinori
System LSI Research Center, Kyushu University | Department of Engineering, Fukuoka University

Yoshiki, Takahito
Department of Engineering, Fukuoka University

Hayashida, Takanori
Department of Engineering, Fukuoka University

<https://hdl.handle.net/2324/18501>

出版情報 : SLRC 論文データベース, pp.922-927, 2010-10. IEEE
バージョン :
権利関係 :

MultiCore Energy Reduction Utilizing Canary FF

Yoshimi Otsuka¹ Toshinori Sato^{1,2} Takahito Yoshiki¹ Takanori Hayashida¹

¹ Fukuoka University, Japan

² Kyushu University, Japan

E-mail: toshinori.sato@computer.org Tel: +81-92-871-6631

Abstract— MultiCore Processor System-on-Chip (MPSoC) is one of the promising technique to satisfy computing demands of the future consumer devices. While MPSoC has an advantage in energy consumption in comparison with high-frequency microprocessor-based system, it is still threatened by increasing energy consumption due to process-voltage-temperature (PVT) variations. It requires large design margins in the supply voltage, resulting in large energy consumption. This paper proposes to utilize a dual-sensing flip-flop (FF), named Canary FF, in order to reduce the overestimated voltage margin. We adopt canary FF to an MPSoC based on Toshiba's MeP and estimate its energy reduction by cycle-based simulations. We find 20.5% energy reduction.

I. INTRODUCTION

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. MultiCore Processor Systems on Chip (MPSoC) is one of the solutions for high-performance and low-energy and it is already adopted in embedded microprocessors.

Unfortunately, MPSoC is still threatened by increasing energy consumption. This is because process-voltage-temperature (PVT) variations require large voltage margins in deep submicron semiconductor technologies. Process variation is predicted to present critical challenges for manufacturability in the future LSIs [1, 5, 13]. The traditional worst-case design may not work since the variation increases design margins it requires. The trend toward lower supply voltage and higher clock frequency makes voltage variations and temperature variations more serious. One of the keys to solve the serious problem is exploiting typical cases. Since worst cases rarely occur, it is better for designers to focus on typical cases. We call it typical-case design methodologies. Recently, several typical-case designs are investigated, such as Razor [2, 3], approximation circuits [7], constructive timing violation (CTV) [8], algorithmic noise tolerance (ANT) [10], and TEAtime [12]. We proposed Canary flip-flop (FF) [9], which is a variation of dual-sensing FF such as Razor FF. Canary FF is utilized to eliminate the overestimated voltage margin. We adopt it to an MPSoC based on Toshiba's MeP [11] and find that it reduces MPSoC energy consumption by 20.5% on average.

This paper is organized as follows. Section II explains the typical-case design methodology. Section III describes related works with an emphasis on Razor. Section IV describes Canary FF. Section V explains our evaluation methodology and Section VI presents experimental results. Finally, Section VII concludes.

II. TYPICAL-CASE DESIGN METHODOLOGIES

Deep submicron semiconductor technologies increase PVT variations, and hence design margins that the traditional worst-case design methodology requires, are increased. The conservative approach may not work. Considering this situation, design methodology should be reconsidered for manufacturability. Typical-case design methodologies are one of the promising ones. It exploits an observation that worst cases are rare. Designers should focus on typical cases rather than worst cases. Since they do not have to consider worst cases, design constraints are relieved, resulting in easy designs.

In the typical-case design methodologies, designers adopt two methods to a circuit design at a time. One is performance-oriented design, where only typical cases are under consideration. Since worst cases are not considered, design constraints are relaxed, resulting in easy designs. The other is function-guaranteed design. While worst cases are considered, designers don't have to consider performance. They only have to guarantee functions, and thus design must be simple, resulting in easy verifications.

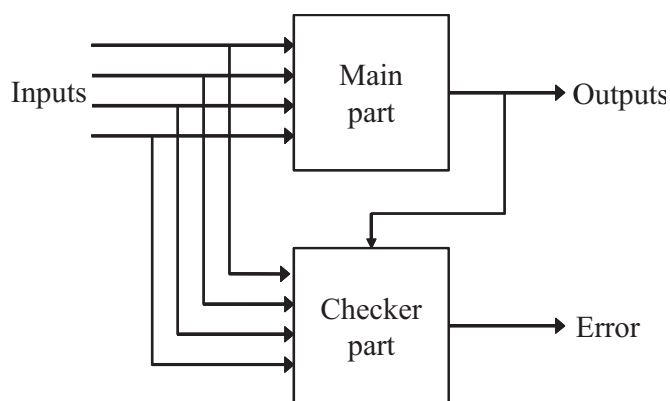


Fig. 1 Typical-Case Design.

We propose one of the typical-case design methodologies. Its concept is as follows. Every critical function in an LSI chip is designed by two methods. The design consists of two components as shown in Fig.1. One is called main part, and the other is called checker part. While two parts share the single function, their roles and implementations are mutually different. On designing the main part, performance is optimized to increase, but correct function is ignored to guarantee. The main part might cause errors. That is, it is implemented by the performance-oriented design. The checker part is provided as a safety net for the unreliable main part. It detects errors that occur in the main part, and thus it has to satisfy all design constraints in the chip. However, on the checker part design, while designers have to guarantee the function, they do not have to optimize neither of performance and power. That is, it is implemented by the function-guaranteed design. If an error is detected by the checker part, the circuit state has to be recovered to a safe point where the error is detected by any means.

III. RELATED WORKS

Examples of the typical-case designs include Razor [2, 3], approximation circuits [7], CTV [8], ANT [10], and TEAtime [12].

In the approximation circuits [7], instead of implementing the complete circuit necessary to realize a desired functionality, a simplified circuit is implemented to approximate it. The approximation circuit works at higher frequency than the complete circuit does, and usually produces correct results. If it fails, the system utilizing the approximation circuit has to recover to a safe point.

CTV [8] exploits input value variations. Considering that the critical path in the system is not always active, clock frequency and supply voltage, which violate critical path delay, are selected in use. In order to guarantee correct operations, the system utilizing CTV has a conservative circuit that realizes a desired functionality to find timing violation.

In ANT [10], information theoretic technique is employed to determine the lower bounds on energy and performance. In order to approach these bounds, circuit- and algorithmic-level techniques are evolved.

TEAtime [12] uses a tracking circuit to mimic the worst-case delay. As long as the tracking circuit works correctly, clock frequency can be increased and supply voltage can be decreased. Usually, a 1-bit-wise critical path is used for the tracking circuit.

A. Razor

Razor [2, 3] 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 voltage margin for power reduction. The voltage control adapts the supply voltage based on timing error rates. Figure 2 shows the Razor's dynamic voltage scaling (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. Note that clock frequency is not changed; that is, it is not a dynamic voltage frequency scaling (DVFS) system. 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.

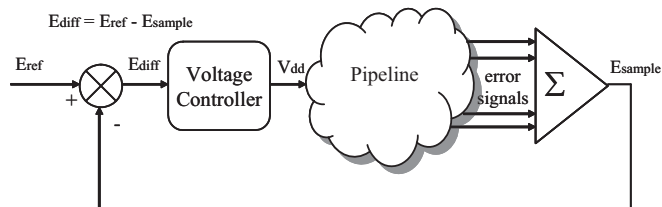


Fig. 2 Razor's DVS System.

In order to detect timing errors, a dual-sensing FF called Razor FF is utilized. Figure 3 shows Razor FF. Each timing-critical FF (main FF) has its shadow FF, where a delayed clock is delivered to meet timing constraints. In other words, the shadow FFs are 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. One of the difficulties on 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 [3].

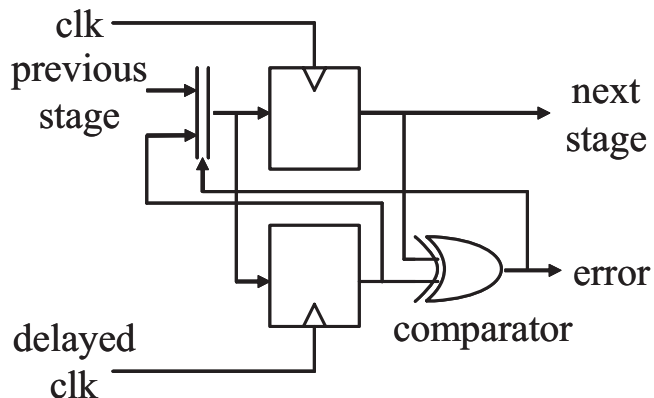


Fig. 3 Razor Flip-Flop.

IV. CANARY

While Razor is a smart technique to eliminate design margins, its circuit implementation could be further improved. We propose a variation of the dual-sensing FFs and coin it Canary FF [9]. Figure 4 shows it.

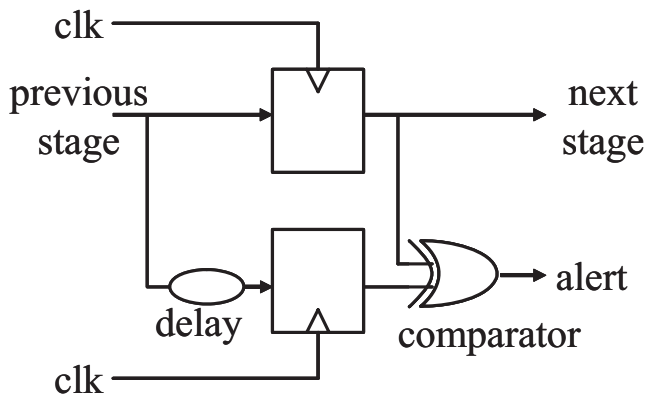


Fig. 4 Canary Flip-Flop.

A. Canary FF

Each FF (main FF) is augmented with a delay buffer and a redundant FF (shadow FF). The shadow 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 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 [3] in Razor FF, and hence its minimum-path length constraint should not be considered.
- Protection offered against timing errors: As explained above, in Canary, the shadow FF protects the main FF against timing errors. This freedom from timing errors eliminates any complex recovery mechanism. Hence, Canary is applicable to the common LSIs as well as modern microprocessors that have the recovery mechanism for branch miss-predictions. If Canary FF predicts a timing error, the supply voltage is increased to satisfy timing constraints.
- Robustness for variations: Canary FF is variation resilient. The delay buffer always has a positive delay, even though parameter variations affect it. Hence, the shadow FF always encounters a timing error before the main FF.

B. Power Reductions with Canary FFs

Figure 5 explains how DVS system utilizes Canary FFs. The horizontal and vertical lines present time and supply voltage, respectively. At regular intervals, the supply voltage is decreased step by step if a timing error is not predicted during the last interval. This exploits input variations. It is well known that input values activating the circuit critical path are limited to a few variations. For example, it is reported that nearly 80% of paths have delays of half the critical time [14]. Timing errors rarely occur even if the timing constraints on the critical path are not satisfied. Input 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 Canary [9] as in Razor [2, 3]. When a timing error is predicted to occur, the supply voltage is increased.

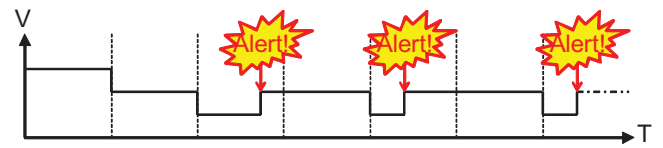


Fig. 5 DVS system utilizing Canary FFs.

C. MultiCore Power Reductions

The target MPSoC is an asymmetric multicore processor (AMP). 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.

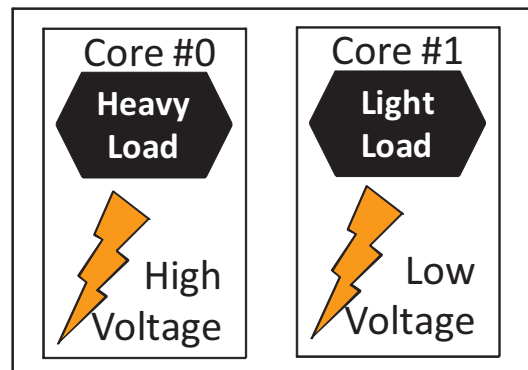


Fig. 6 DVS with Multiple Supplies.

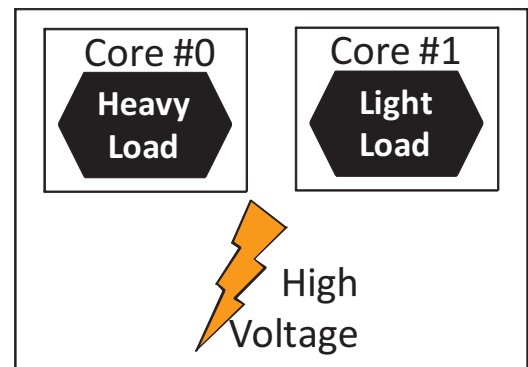


Fig. 7 DVS with Single Supply.

DVS system is utilized for multicore energy reduction. DVS system has been shown to be one of the most energy-efficient design techniques for single-core processors. For multicore processors, two implementations of DVS systems can be considered. One is the MPSoC where different cores operate at different voltage levels based on the operation load as shown in Fig. 6. With multiple supply voltages, only cores with heavy workload run at higher supply voltage to provide

required performance while other cores operate at lower supply voltage or are completely shut down. The other is the one where all cores operate at the same voltage level as shown in Fig. 7. With one scalable supply voltage, all cores run at the voltage that satisfies the demand of the heaviest workload. While the former will achieve larger energy savings than the latter one will do, it requires voltage islands, which increases design complexity and chip area, resulting in larger manufacturing cost. Since we adopt Canary to embedded devices, where cost is one of the most important design constraints, we chose the latter DVS system in this study.

V. EVALUATION METHODOLOGY

MeP simulators provided by Toshiba are used to generate execution traces. They are cycle-based simulators and model a single-core and a dual-core MeP processors [11] in details, respectively. 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.

Each trace is injected into the trace-driven simulator we built. The number of cores can be configured in the in-house simulator. The details of Canary DVS system are implemented. Since the yield of pipeline is mainly determined by the timing error in the execution stage [6], 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 [4], which is shown in Table I. We also consider the rule of thumb; PVT variations require 50-100% design margins [15]. The thresholds are also summarized in Table I. For example, the carry longer than 18 bits at 1.132V signals the error prediction, when we do not consider design margin.

TABLE I. VOLTAGE-THRESHOLD SPECIFICATIONS.

Supply (V)	Freq (GHz)	Threshold (bit)	
		Margin 0%	Margin 50%
1.340	2.1	32	32
1.260	1.8	27	32
1.228	1.6	24	32
1.180	1.4	21	32
1.132	1.2	18	27
1.084	1.0	15	22
1.036	0.8	12	18
0.988	0.6	9	13

We evaluate 10 intervals between supply voltage scaling, which are 100, 200, 500, 1K, 2K, 5K, 10K, 20K, 50K, and

100K clock cycles. It is assumed every supply voltage switching requires 100 clock cycles.

In single-core MeP simulations, each program is executed from beginning to end. On the other hand, in dual-core MeP simulations, each simulation is terminated when one of two programs finishes. We chose this methodology, because we are afraid that longer program might dominate the simulation result. Hence, it should be noted that single-core and dual-core simulation results cannot be directly compared with each other because the parts of programs simulated are different.

VI. RESULTS

First, we evaluate Canary DVS system on the single-core MeP. The interval between voltage switches is varied between 100 and 100K cycles. Figure 8 presents the results. The horizontal line indicates the interval and the vertical line indicates the percentage energy reduction rate. Note that the horizontal line is in log scale. Two line graphs are shown. The lower graph (denoted as *margin 0%*) presents the results when the design margin is not considered. The upper one (denoted as *margin 50%*) presents those when 50% of timing margin is included. The difference between the two graphs is the energy wastes due to the overestimation and is up to 24.1%. Canary DVS system eliminates the wastes and the energy savings is 26.7% on average when the interval is 2000 cycles.

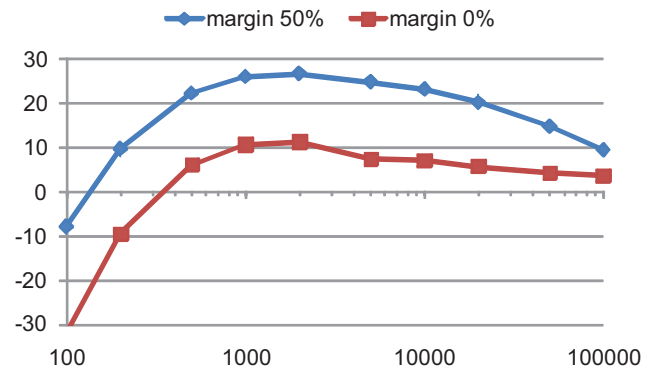


Fig. 8 Energy Reduction Rate.

When the interval is small, energy consumption is rather increased. This is because 100 cycles of overhead cannot be negligible. On the other hand, after the peak, energy savings is gradually decreased as the interval becomes larger. Longer interval will lose the chances where the supply voltage is decreased, as the voltage switch is infrequent.

An example of the distribution of the selected supply voltages is presented in Fig. 9. This shows the case of *bubble* and of 2000-cycle interval. As can be easily seen, Canary DVS system chooses the supply voltage lower than that determined by considering design margin. Over 70% of execution cycles operates at lower than 1.084V.

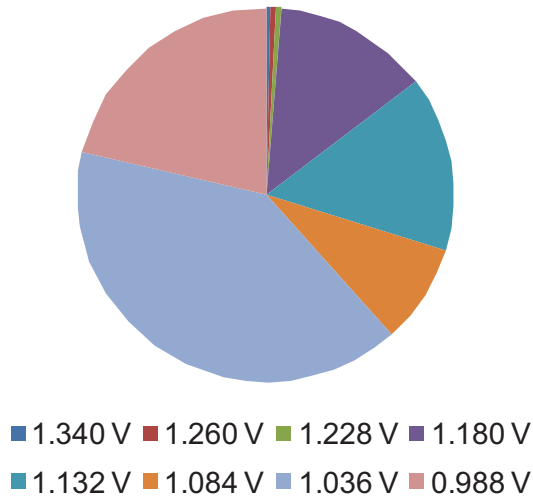


Fig. 9 Supply Voltage Distribution.
(Single core)

TABLE II. (%) ENERGY REDUCTION RATE.
(1000 CYCLE INTERVAL)

	matmul	perm	qsort	queen	sieve
bubble	25.88	15.19	14.86	15.34	39.57
matmul		15.07	14.92	15.04	25.57
perm			14.59	14.97	14.92
qsort				14.46	14.39
queen					15.35

TABLE III. (%) ENERGY REDUCTION RATE.
(2000 CYCLE INTERVAL)

	matmul	perm	qsort	queen	sieve
bubble	26.46	18.31	17.78	18.79	37.22
matmul		18.31	17.72	18.43	26.70
perm			17.57	18.23	18.08
qsort				17.50	17.18

TABLE IV. (%) ENERGY REDUCTION RATE.
(5000 CYCLE INTERVAL)

	matmul	perm	qsort	queen	sieve
bubble	21.50	19.58	18.16	20.69	22.38
matmul		19.44	18.09	20.03	24.96
perm			18.18	19.48	19.04
qsort				18.10	17.10
queen					20.77
Queen					18.82

Next, we show the results for the dual-core MeP. Based on the single-core results, we choose the intervals of 1000, 2000, and 5000 cycles, where energy savings is largest. TABLES II to IV present the results for the cases of 1000, 2000, and 5000

intervals, respectively. Since a pair of six program are selected, we perform ${}^6C_2=15$ simulations for each interval.

When the interval is 1000 cycles, an average of 18.0% energy reduction is achieved. Before simulations, we expected that power saving would be significantly reduced. This is because that the supply voltage can only be decreased when both programs prefer lower supply voltage. Even if only one program prefers higher voltage, the whole MPSoC operates at higher supply voltage. However, fortunately, in the most combinations of programs, energy savings is enough large. As the interval becomes larger, energy consumption is further reduced. When the interval is 5000 cycles, the average energy reduction is 19.8%. Figure 10 presents the distribution of the supply voltage. The executed programs are bubble and matmul, and the interval is 2000 cycles. The voltage that is most frequently selected is 1.180V. 57.9% of execution cycles operate at this voltage. Interestingly, the highest two voltages are rarely selected; at only 2% in total. This means that the supply voltage switch occurs frequently. While it implies that its overhead might increase the execution cycles and thus it might result in energy increase, the simulation results shows the overhead is not too large to make Canary DVS system useless.

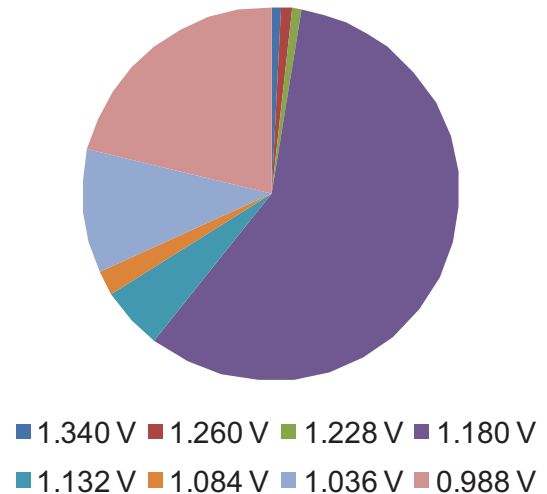


Fig. 10 Supply Voltage Distribution.
(Dual core)

When the interval is 2000 cycles, energy savings is highest and is an average of 20.5%. This confirms that Canary DVS system eliminates the overestimated energy consumption. It chooses lower supply voltages as frequently as possible, and predicts timing errors to prevent MPSoC from incorrect state.

VII. CONCLUSIONS

As the demand of computing power is increased even in the embedded devices, MPSoC becomes more and more attractive. While MPSoC has the advantage of power efficiency in comparison with single-core alternatives, it still consumes wasted energy. This is due to increasing PVT

variations. In order to satisfy timing constraints at the worst case scenarios, the supply voltage should be overestimated. We evaluate it on Toshiba's dual-core MeP processor and found that 20.5% of energy savings is possible even when MPSoC does not have multiple voltage supplies.

ACKNOWLEDGMENT

This work is partially supported by the CREST (Core Research for Evolutional Science and Technology) programs of Japan Science and Technology Agency (JST), and by Grant-in-Aid for Scientific Research (B) #20333319. The authors would like to thank Shunitsu Kohara of Toshiba for helping them use MeP simulators.

REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] Intel Corporation, "Intel Pentium M Processor on 90nm Process with 2-MB L2 Cache," *Datasheet*, 2006.
- [5] T. Karnik, S. Borkar, and V. De, "Sub-90nm Technologies: Challenges and Opportunities for CAD," *International Conference on Computer Aided Design*, 2002.
- [6] 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.
- [7] S.-L. Lu, "Speeding up Processing with Approximation Circuits," *IEEE Computer*, Vol. 37, No. 3, 2004.
- [8] T. Sato and I. Arita, "Constructive Timing Violation for Improving Energy Efficiency," *Compilers and Operating Systems for Low Power*, 2003.
- [9] T. Sato and Y. Kunitake, "A Simple Flip-Flop Circuit for Typical-Case Designs for DFM," *8th International Symposium on Quality Electronic Design*, 2007.
- [10] N. R. Shanbhag, "Reliable and Efficient System-on-chip Design," *IEEE Computer*, Vol. 37, No. 3, 2004.
- [11] Toshiba Corporation, "Media embedded processor," www.semicon.toshiba.co.jp/eng/product/micro/mep/index.html.
- [12] A. K. Uht, "Going beyond Worst-case Specs with TEAtime," *IEEE Computer*, Vol. 37, No. 3, 2004.
- [13] 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.
- [14] K. Usami, M. Igarashi, F. Minami, T. Ishikawa, M. Kanazawa, M. Ichida, and K. Nogami, "Automated Low-Power Technique Exploiting Multiple Supply Voltages Applied to a Media Processor," *IEEE Journal of Solid-State Circuits*, Vol. 33, No. 3, 1998.
- [15] Private communications with LSI designers.