

# Typical Case Oriented Design Approach by Timing Error Prediction to Tolerate Process Variability\*

Ken YANO<sup>\*\*,\*\*\*</sup> and Toshinori SATO<sup>\*\*</sup>

The demand of low-power and dependable LSI has increased with the progress of semiconductor process technologies and with the spread of portable devices such as smart phones. The conventional design method which considers the worst case scenario makes the design margin very large because the parameter variations in the deep submicron domain become serious. Increasing design margins has serious negative impact on performance. In order to eliminate the excessive design margins, we propose the typical case design method, which utilizes canary flip-flops (FFs). In this paper, we will analyze issues and benefits from incorporating canary FFs in the proposed typical case oriented system design. First, we review the timing error prediction by canary FFs. Next, we introduce two possible implementations of canary FFs, which are soft and hard cells. After that, we will discuss how to selectively replace the original FFs with canary FFs. In the experimental evaluations, we will analyze the power and area overhead incurred by canary FFs on two RISC microprocessors.

**Key Words**: canary FF, timing error, process variability, reliability

## 1 Introduction

Due to the miniaturization of semiconductor devices and the spread of mobile equipment, higher speed and lower power are further requested on LSI designs. In order to reduce power consumption, lower power supply voltage is required. In other words, power budget of embedded systems is diminished. The minimum supply voltage that ensures correct operations is referred as the critical supply voltage. Usually, the critical supply voltage is determined by considering a number of environmental and process conditions, which include unexpected voltage drops in the power supply network, temperature fluctuations, and gate-length and doping concentration variations.

To ensure correct operation under all possible variations, a conservative supply voltage is typically selected by based on corner analysis at design time. Some design margins are added to the critical supply voltage in order to tolerate the uncertainty from the worst-case combination of variabilities. In addition, with process scaling, the environmental and process variabilities are expected to increase, worsening the required voltage margins. However, such a worst-case combination is very rare or even impossible in actual operations. To aggressively reduce power

consumption, Razor FF is proposed [1]. It is based on dynamic detection and correction of timing failures in digital designs and its key idea is to tune the supply voltage by monitoring the error rate during the operation. Since Razor has error correction capability, the operation at sub-critical supply voltage does not constitute a catastrophic failure, but instead represents a trade-off between the power penalties incurred from error correction against additional power savings obtained from operating at a lower supply voltage. Apart from power savings, since Razor requires in-situ correction of timing failure, it has some negative impact on circuit implementation and circuit area.

We propose canary FF, which is another circuit-level technique to tackle the variability issues. Unlike Razor, canary tries to predict timing failures in order to achieve the sub-critical supply voltage operations. In this paper, we analyze the typical-case oriented system design, which utilizes canary FF, in regard to timing analysis, circuit design, performance overhead, and reliability. After describing background of this study in Section 2, we discuss the timing constraints of canary FF in Section 3 and present the circuit design of canary FF in Section 4. Next in Section 5, we describe the selective replacement algorithm for canary FF. Based on the replacement algorithm, we implement two conventional 32-bit RISC processors and analyze the area and power overhead caused by the canary FFs in Section 6.

---

\* Manuscript received May XX, 2015.

\*\* Dept. Electronics Engineering and Computer Science

\*\*\* Currently with ATR.

Then in Section 7, we discuss the reliability of canary FF by analyzing 32-bit Kogge-Stone adder. Finally in Section 8, we conclude the paper with presenting future directions.

## 2 Background

A number of better-than-worst case designs have been proposed to allow circuits operate under the normal conditions rather than the conservative worst-case limits for saving power consumption. One class of such techniques specifies multiple safe combinations of voltage and frequency levels and thus a design can operate at a certain combination and at some time switch between them [2-4]. Another similar circuit technique uses multiple latches which strobe a signal in close succession to locate the critical operating point of a design [5]. The third latch of a triple-latch monitor is always assumed to capture correct value, while the first two latches indicate how close the current operating point is to the critical point.

All techniques mentioned above are based on “always-correct” architecture. In contrast, Razor FF [1] allows voltage scaling beyond the critical point. This is possible since Razor FF incorporates the error detection and correction mechanism to handle the case where a timing failure occurs. It detects timing violations by supplementing critical FFs with a shadow latch that strobes the output of a logic state at a fixed delay, typically half a cycle. Error correction in the Razor-based design involves recovery process using the correct values stored in the shadow latches. To guarantee correct operation, Razor requires two delicate conditions to be met on the circuit behavior, namely short path and long path constraints. Razor II is an improved alternative that performs only error detection, while correction is performed through architectural replay [6].

Unlike Razor FF, the phase synchronized clock pulse is provided to both the main and the shadow FFs of canary FF. Since delay buffer is inserted in front of the shadow FF, the setup time condition becomes severer at the shadow FF than at the main FF. In order to assure that shadow FF will cause a timing failure before the main FF, the delay value should be carefully determined. By combining canary FF with DVFS (Dynamic Voltage Frequency Scaling) mechanism, a large degree of power reduction can be achieved [7]. In order to adopt canary FF

for ASIC design, the selective replacement method is proposed in [8]. By analyzing timing-error-prone paths of functional blocks, FFs at the end of those paths are replaced with canary FFs. This method uses logic-synthesis results from multiple cell libraries to search the candidate FFs for replacement. Hence it can be integrated into EDA tool. In [9], circuit design with the selective canary FFs replacement is presented in detail.

## 3 Timing Error Prediction by Canary FF

Canary FF consists of a pair of FFs, which are the main FF and the shadow FF, a delay buffer, and an XOR gate. Its block diagram is shown in Figure 1. The phase synchronized clock is provided to both the main and shadow FFs. Delay buffer is inserted in front of the shadow FF, hence the setup timing constraint for the shadow FF becomes severer than that for the main FF. The setup timing of the shadow FF largely depends on the delay value. Figure 2 shows the dependence of the setup time on the delay value and on the supply voltage. The horizontal axe indicates the supply voltage and the vertical axe indicates setup time in pico-second. For the eight lines, m-ff and s-ff mean the main and shadow FFs, HL and LH mean the fall and the raise times, and d\_1, d\_2, and d\_3 mean the unit delay, two times of the unit delay, and the three times of the unit delay, respectively. In order to calculate setup time of the shadow FF, we observe Q2 instead of Q1 in Figure 1. A unit delay buffer consists of two inverters. From Figure 2, we can find the followings. First, the setup time of the shadow FF increases proportionally to the delay value.

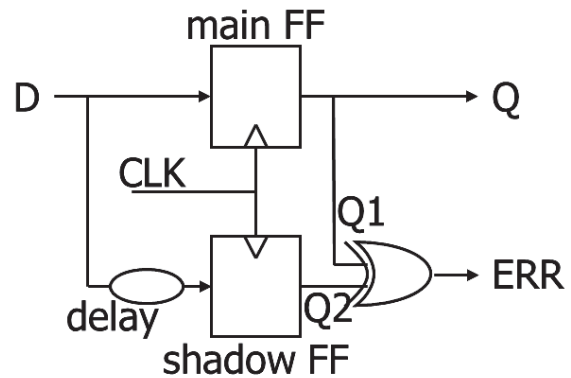


Figure1: Canary FF

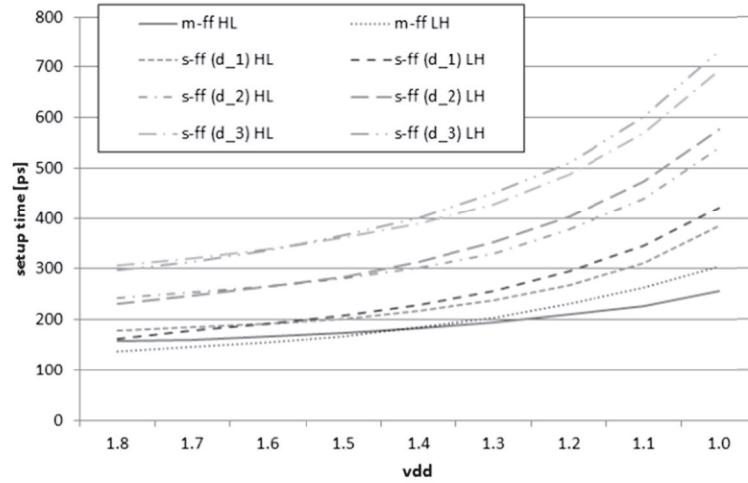


Figure 2: Dependence of setup time on delay value and supply voltage

Second, the setup time increases non-linearly to the supply voltage. Third, the difference of the setup times between the main and shadow FFs increases gradually as the supply voltage is decreased.

As can be seen in Figure 1, a timing error is predicted by comparing the outputs of two FFs. If two values match, the system is operating in the safe zone and the supply voltage can be scaled down. If they do not match, the system enters into the unsafe zone and the timing error signal notifies the absence of safety margin.

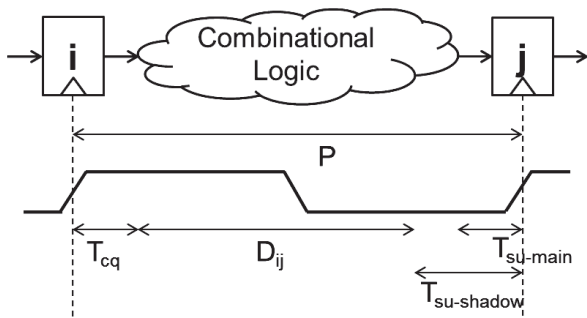


Figure 3: Setup time constraint of canary FF

In this part, we analyze the setup timing constraint of canary FF to support the following discussion in this paper. The hold time constraint does not change even if the shadow FF is introduced because the delay buffer is inserted in front of the shadow FFs. Figure 3 shows the setup time constraint of canary FF. In

this figure, we assume that the sequential element  $i$  is a conventional edge-triggered FF and the element  $j$  is an edge-triggered canary FFs. The setup times of the main and shadow FFs are denoted by  $T_{su-main}$  and  $T_{su-shadow}$ , respectively. Note that  $T_{su-shadow} \geq T_{su-main}$ , as shown in Figure 3. The clock-to-Q delays of the main and shadow FFs are assumed to be same and it is denoted by  $T_{cq}$ .  $P$  is the cycle time. We do not distinguish between propagation and contamination delay (i.e. maximum and minimum delay) of  $T_{cq}$  for simple discussions. The maximum delay of combinational block between sequencing elements  $i$  and  $j$  is denoted by  $D_{ij}$ . We consider three situations:

- 1 No timing error occurred: Data is launched from FF  $i$  at the rising edge of clock and the latest result of computation from the combinational block has to arrive at canary FF  $j$  earlier than the setup time of both main and shadow FFs before the next rising edge of clock, which constitute the setup time constraint:

$$T_{cq} + D_{ij} \leq P - T_{su-shadow} \leq P - T_{su-main}$$

- 2 Timing error predicted: If the latest result of computation from the combinational block arrive at canary FF  $j$  earlier than setup time of the main FF and later than setup time of the shadow FF, error prediction signal is notified, however the main FF still latch the correct data. So the FFs still can transfer data correctly.

- $P - T_{su-shadow} \leq T_{cq} + D_{ij} \leq P - T_{su-main}$
- 3 Timing error occurred: If the latest result of computation from the combinational block arrives at canary FF  $j$  later than setup time of main and shadow FFs, both the main and shadow FFs cannot latch the latest data, hence the error signal may not be notified. This will corrupts the sequence of the correct data transfer.

$$P - T_{su-shadow} \leq P - T_{su-main} \leq T_{cq} - D_{ij}$$

From above discussions, the range of the maximum delay  $D_{ij}$  is divided into three parts in terms of timing error. If the  $D_{ij}$  is less than  $(P - T_{su-shadow} - T_{cq})$ , no timing error is notified and data is transferred correctly. If it is between  $(P - T_{su-shadow} - T_{cq})$  and  $(P - T_{su-main} - T_{cq})$ , the timing error is predicted, however the main FF still can latch the correct data, hence the data is transferred correctly. If it is larger than  $(P - T_{su-main} - T_{cq})$ , both the main and shadow FFs cannot latch the latest data, and thus the timing error might not be notified. Hence, it will cause a catastrophic failure due to wrong sequence of data.

The range of  $D_{ij}$  during which timing error is correctly notified is defined as:

$$(P - T_{su-main} - T_{cq}) - (P - T_{su-shadow} - T_{cq}) = T_{su-shadow} - T_{su-main}$$

Since the difference of setup time of the shadow and main FFs is determined by the delay value of the inserted buffer, it is crucial to determine the appropriate delay value in order to optimize the system performance and also to leave the enough margins for tolerating timing fluctuations caused by process and environmental variations.

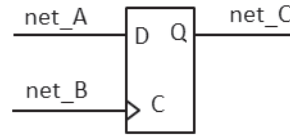
## 4 Design of canary FF

In this section, we consider to implement canary FF. There are two approaches for the implementation. One is to implement it as a soft macro cell and the other is to implement it as a hard macro cell. We describe each approach in the following discussions. For the design and analysis of canary FF, we use the cell library provided from Kyoto University [10] based on Rohm 0.18 um CMOS technology.

### 4.1 Canary FF as soft cell

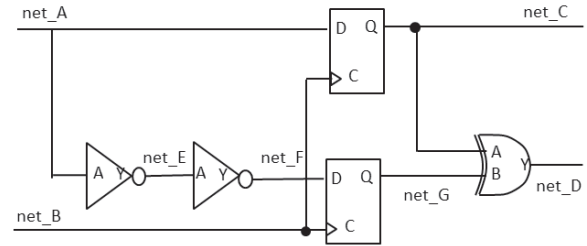
In this approach, canary FF is implemented as a soft cell, which is a combination of existing standard cells, such as D FF, inverter, and XOR gate. We convert a D FF to canary FF. The

original D FF and its description are shown in Figure 4. Figure 5 describes the corresponding canary FF and its description. The original description is automatically converted into that of canary FF. The implementation of the automatic conversion is described in [9]. One problem is that the soft cell may suffer from the clock skew between the two FFs, since they are placed independently by the placement-and-routing (P&R) tool. It is important to decrease the clock skew as much as possible.



```
ROHM18DFP010 reg_1 (.D(net_A), .C(net_B), .Q(net_C));
```

Figure 4: Conventional D FF



```
ROHM18DFP010 reg_1 (.D(net_A), .C(net_B), .Q(net_C));
ROHM18DFP010 UU_1 (.D(net_F), .C(net_B), .Q(net_G));
ROHM18INV010 UU_2 (.A(net_A), .Y(net_E));
ROHM18INV010 UU_3 (.A(net_E), .Y(net_F));
ROHM18XOR2P010 UU_4 (.A(net_C), .B(net_G), .Y(net_D));
```

Figure 5: Soft canary FF cell

### 4.2 Canary FF as hard cell

In this approach, canary FF is implemented as a hard cell. When designing a redundant FF such as canary FF, the area and power overheads are important issues and hence the circuit must be designed optimally. In Figure 6, we propose an optimized circuits of canary FF. In this circuit, the slave latch of the shadow FF is omitted in order to decrease the cell area. Based on the optimized circuit, canary FF is implemented as a double height cell, as shown in Figure 7. The comparisons of the area and the average power between a D FF and canary FF are shown in Table 1. Both of the area and the

power of canary FF are 2.5 times larger than those of a D FF.

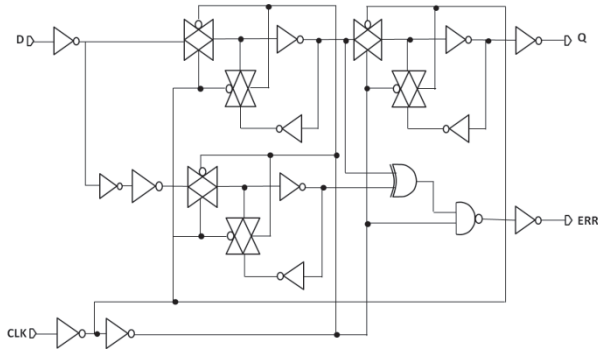


Figure 6: Optimized circuit of canary FF

Table 1: Power and are of D FF and canary FF

	D flip-flops	canary flip-flops
Avg. power[mW]	0.025	0.063
Area[ $\mu\text{m}^2$ ]	51.6	129.0

## 5 Selective replacement of canary FF

In this section, we consider to utilize canary FF in the design of an ASIC such as a microprocessor. The conventional processor typically contains more than tens of thousands sequential elements such as FFs and latches. Hence, replacing all the FFs to canary FFs will have severe negative impact on chip area and power consumption. In order to reduce the number of FFs to be replaced with canary FF, the candidate FFs are carefully selected. We describe an algorithm for this selection in the following discussions.

The flowchart of the selection algorithm is described in Figure 8. The proposed algorithm uses two types of standard cell libraries; “Typ” and “Max”. The “Typ” library is built for considering the typical process and environmental conditions. On the other hand, the “Max” library is built for considering the worst process and environmental conditions.

Given that the functional correctness of the source RTL is already checked, then it is synthesized into a technology mapped netlist using “Typ” cell library. The synthesis is repeated until the minimum clock cycle  $\text{Clock\_typ}$  is determined. The minimum clock cycle  $\text{Clock\_typ}$  is obtained so that there are no timing violations under the typical conditions. The obtained gate-level netlist is saved as the

original netlist. Next, another synthesis is performed by setting the clock cycle to  $\text{Clock\_typ}$  and this time by using “Max” cell library. Since the timing condition of “Max” cell library is severer than that of “Typ” cell library, some paths should be reported as timing errors. By analyzing the result of synthesis, the instances of FFs at the end of vulnerable paths are recorded. Given the source netlist and registered instances of FFs described above, the original netlist is converted into the final netlist by searching the instances of FFs recorded and replacing it to the instance of canary FFs. Using the final netlist, STA (Static Timing Analysis) is performed with required timing constraints to ensure that replaced canary FFs might not cause any timing violations.

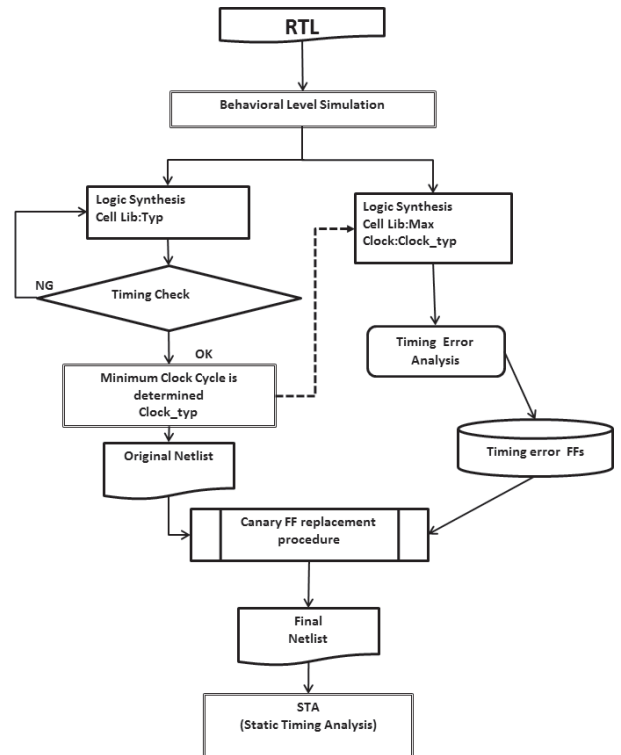


Figure 8: Selective replacement algorithm

By using the selection algorithm, we design two RISC microprocessor cores; MeP [11] and miniMIPS [12]. Logic synthesis is performed by using Synopsys Design Compiler. The two cores are different in instruction set architecture and microarchitecture. Figures 9 and 10 show the layouts of the two cores. In this figure, canary FFs are indicated as white cells. Table 2 shows the statics of the total number of FFs and the percentage of FFs, which are replaced by canary FFs. It turns out that only 1.6 % and 11.6% of

the FFs are selectively replaced in the cases of MeP and miniMIPS, respectively.

Table 2: % of replaced FFs

RTL	Total #	# of flip-flops replaced with canary flip-flops	percentage
MeP	3732	60	1.6%
miniMIPS	1967	228	11.6%

## 6 Power + area overhead of canary FF

In this section, the overheads of area and power in the microprocessor cores, which are caused by introducing canary FF, are evaluated. The P&R tool used in this evaluation is Synopsys IC Compiler. We use four configurations to estimate the area and power overhead caused by canary FF. Each configuration is described as follows.

- 1) Config-T: Logic synthesis and P&R are performed using “Typ” cell library and any D FFs are not replaced by canary FF. This configuration is impractical and is used for the purpose of estimating the optimistic core area and power consumption.
- 2) Config-M: Logic synthesis and P&R are performed using “Max” cell library and any D FFs are not replaced by canary FF. This configuration is a practical but worst case, which considers the worst conditions.
- 3) Config-TC: Logic synthesis and P&R are performed using “Typ” cell library and vulnerable D FFs are replaced with canary FF by using the proposed selective replacement method. This configuration is the proposed case.
- 4) Config-TCA: Logic synthesis and P&R are performed using “Typ” cell library and all D FFs are replaced with canary FFs. This configuration is used for the purpose of estimating how the area and power overhead is reduced by the selective replacement.

Table 3 shows the chip areas of miniMIPS and MeP for each configuration. In the case of miniMIPS core, the area of the impractical case (Config-T) is approximately 26 % smaller than that of the practical case (Config-M). This result clearly indicates that the area overhead is very large when considering worst case conditions. In the proposed case (Config-TC), some D FFs, which are vulnerable to timing errors, are selectively replaced with canary FFs, resulting in the core area reduction of 20% when it is

compared with that of the worst case (Config-M). In addition, the area is comparable to that of the impractical case. In the case of MeP core, the difference among the four configurations is small. This is because a large portion of the core area is occupied by cache memories. Since they are not the target of canary FF, the area overhead is strongly reduced.

Table 3: Processor core area

Core Area	Conf. T	Conf. M	Conf. TC	Conf. TCA
miniMIPS [ $mm^2$ ]	0.436	0.587	0.468	0.591
Norm. Area	0.742	1.00	0.796	1.01
MeP [ $mm^2$ ]	2.66	2.70	2.76	2.96
Norm. Area	0.989	1.00	0.992	1.11

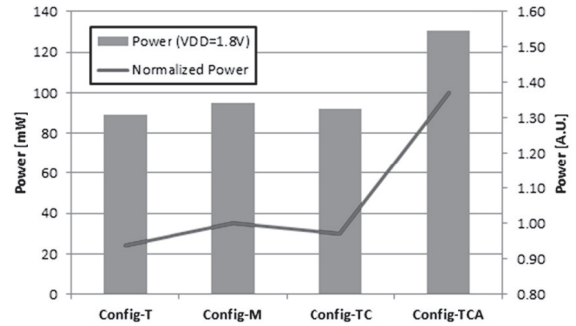


Figure 11: Power consumption of miniMIPS

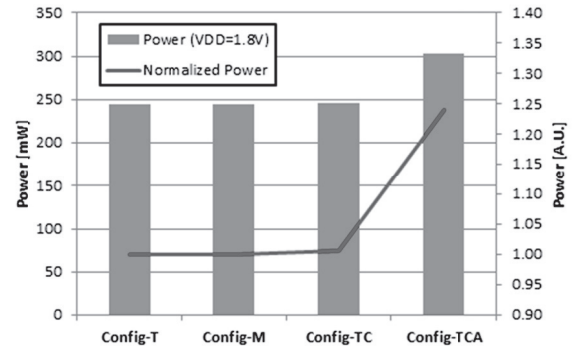


Figure 12: Power consumption of MeP

Figures 11 and 12 show the dynamic power analysis of miniMIPS and MeP processor cores for the different configurations. For the power analysis, toggle rate and signal probability are assumed to be 0.025 and 0.015, respectively, for all registers. In the case of miniMIPS, power is reduced by approximately 5% from the worst case (Config-M) to the proposed case (Config-TC). The difference between the impractical (Config-T) and the proposed (Config-TC) cases is

negligible. In the case of MeP, the difference among Config-T, Config-M, and Config-TC is very small, because the percentage of the replaced FFs is low. In both processor cores, the power overhead is significantly mitigated by the selective replacement algorithm, when we see the cases of Config-TC and Config-TCA.

## 7 Reliability analysis of canary FF

In this section, we discuss the reliability of canary FF by analyzing its behavior. For this analysis, we use a 32-bit Kogge-Stone adder. It is implemented as a SPICE netlist and is evaluated by Monte-Carlo simulations to consider process variations.

The target circuit is shown in Figure 13. The path delay of the circuit becomes largest when the carry propagates from the least significant bit to the most significant bit. For example, when  $A = 0xFFFFFFFF$ ,  $B = 0x00000000$  and  $CIN = 0x1$ , the carry propagation length is largest. From our simulation, the maximum path delay is 2.40 ns for typical case and 3.03 ns for worst case, each of which corresponds to 417 MHz and 330 MHz, respectively. We call these delays  $MaxDelay\_typical$  and  $MaxDelay\_worst$ . The difference between  $MaxDelay\_typical$  and  $MaxDelay\_worst$  is 630 ps.

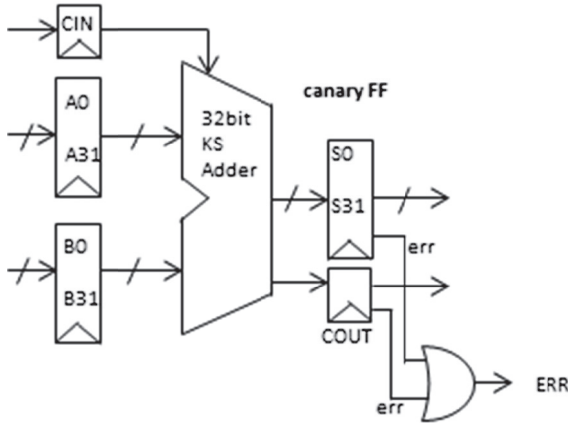


Figure 13: 32-bit Kogee-Stone adder

In the evaluation circuit, the delay buffer of canary FF is three times larger than the unit delay buffer. This is determined from our empirical observations. For the Monte-Carlo simulations, which consider the local process variations, we prepare one hundred set of random data for the input. Figure 14 shows the simulation results and represents the histogram of the delays when the supply voltage is 1.8 V.

We can see the average delay is 1.5 ns.

Figure 15 shows a part of Figure 14 and explains the setup time constraint of canary FFs, which are used in the adder.  $MaxDelay$  in the figure is the maximum path delay between  $MaxDelay\_typical$  and  $MaxDelay\_worst$ . When we determine that the setup time of the shadow FF is three times larger than that of the main FF, the delay margin of the adder is as shown as the hatched area in the figure. When the supply voltage is sufficiently high, the setup timings of both FFs are always satisfied, and hence any timing errors do not occur. As the supply voltage is decreased, the delay histogram shown in Figures 14 and 15 moves to the right and the margin becomes small. Once the setup constraint of the shadow FF is violated, canary FF predicts the timing error, where any margin is not left for the voltage scaling. If we ignore the timing error signal and the supply voltage continues to scale down further, the setup constraint of the main FF is also violated, where canary FF might miss the serious timing error.

In order to verify the above observations, we analyze how timing error notification rate changes when the supply voltage is decreased. Figure 16 shows the results for two different clock frequencies.  $Clock\_lo$  and  $Clock\_high$  are determined when the worst and typical constrains are considered, respectively. As explained above, we assume the local process variations, which are simulated by varying the threshold voltage ( $V_{th}$ ), oxide thickness ( $T_{ox}$ ), and the effective device length ( $L_{eff}$ ) and width ( $W_{eff}$ ) in the SPICE model used in the evaluations. We count the number of error prediction signals.

We call the supply voltage, where the first timing error is predicted when it is decreased from 1.8 V, first error notification (FEN) voltage. Beyond FEN voltage, the error notification rate increases monotonously. It is found that the supply voltage safely scales down up to 1.65 V and 1.40 V for  $Clock\_high$  and  $Clock\_lo$ , respectively. This result confirms the above observations so that the timing margin is larger for the worst case than for the typical case.

Next, let us discuss how the supply voltage is safely decreased. When the supply voltage is decreased beyond FEN voltage, timing error is predicted more frequently. It should be noted that canary FF might miss serious timing errors as mentioned above. Therefore, it is better for the supply voltage not to be decreased beyond

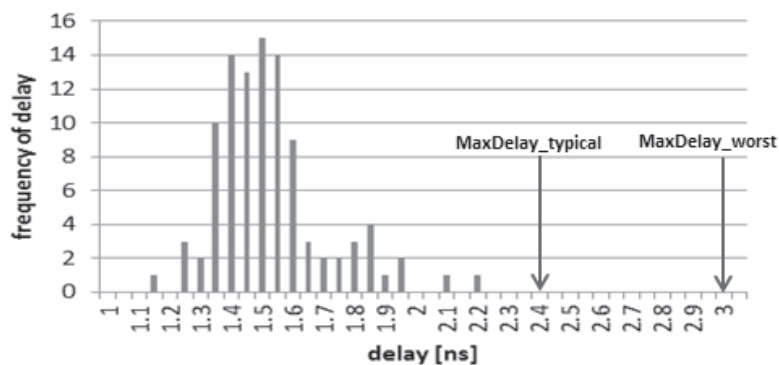


Figure 14: Delay distribution

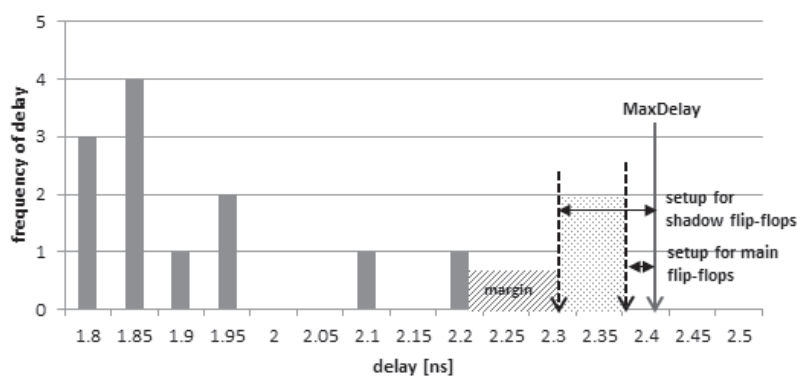


Figure 15: Setup time constraint

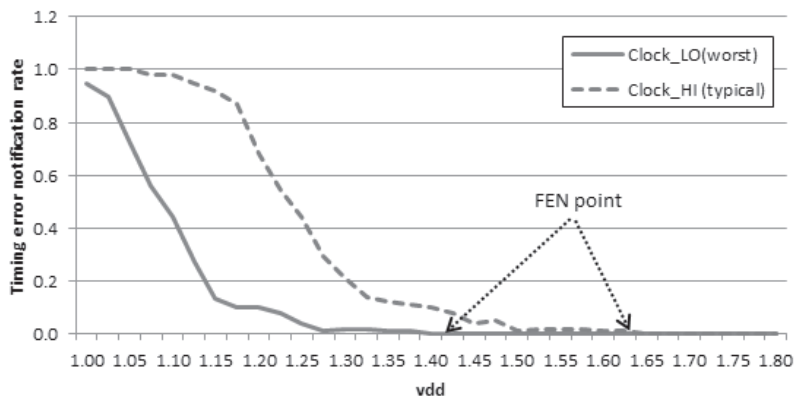


Figure 16: Error notification rate

FEN voltage, especially when the difference of the setup timings of the main and shadow FFs is small. Increasing the safety margin to guarantee correct operations requires a large delay buffer. On the other hand, the larger delay buffer shifts the FEN point to the right, resulting in the loss of voltage scaling.

## 8 Conclusions

The progress of semiconductor technologies makes LSI designs very difficult due to emerging process, voltage, and temperature variations. In order to ensure correct operations for any combinations of variations, conservative design approach requires large guard banding.



Unfortunately, it inevitably diminishes system performance and energy efficiency, even though the worst case scenario of variation conditions rarely happens.

In this paper, we propose a typical case oriented design methodology, which is supported by canary FF and analyze design issues caused by introducing the methodology. We overview the timing issues of canary FF. The amount of delay buffer determines the achievable system performance and the margin for correct operations.

Canary FF can be implemented either as a soft cell for as a hard cell. If it is implemented as a soft cell, it is important to place the main and the shadow FFs as close as possible in order that there is not a large clock skew between them. When canary FF is utilized in semi-custom ASIC such as microprocessors, it is important to limit the number of canary FFs since they have severe negative impact on area and energy. We propose the selective replacement algorithm and integrate it into the commercial EDA tool chain. By using this algorithm, it is shown that area and power overhead is greatly reduced.

On analyzing the reliability of canary FF, we use 32-bit Kogge-Stone adder for simulations. It is verified that timing errors are predicted when the supply voltage scales down. Utilizing canary FF makes it possible that the supply voltage is dynamically controlled to aggressively reduce power consumption.

## Acknowledgements

This study is supported in part by funds (No. 135007) from the Central Research Institute of Fukuoka University and was supported in part by JST CREST DVLSI project. The cell library used on this research was developed by Tamaru and Onodera laboratory, Kyoto University and is released by Kazutoshi Kobayashi of Kyoto Institute of Technology. This work is supported by VDEC [13], the University of Tokyo in collaboration with Synopsys, Inc., Cadence Design Systems, Inc. and ROHM Co.,Ltd.

## Author Contributions

Toshinori Sato contributed to the concept and the design of this study and wrote a part of this manuscript. Ken Yano built the prototype design flow, collected the experimental data, analyzed and interpreted the data, and wrote the most of the manuscript.

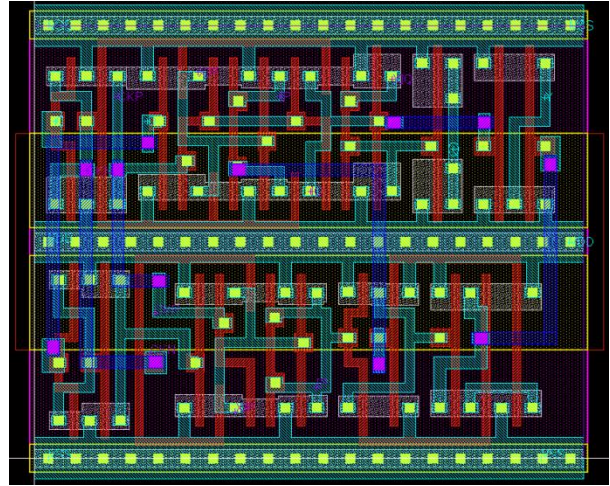


Figure 7: Hard canary FF cell

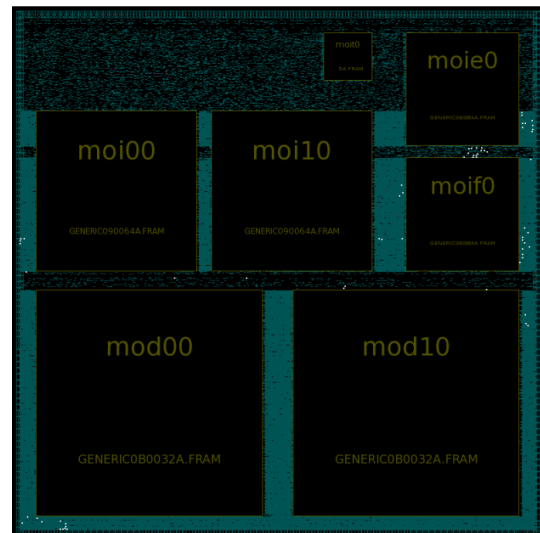


Figure 9: Layout plot of MeP

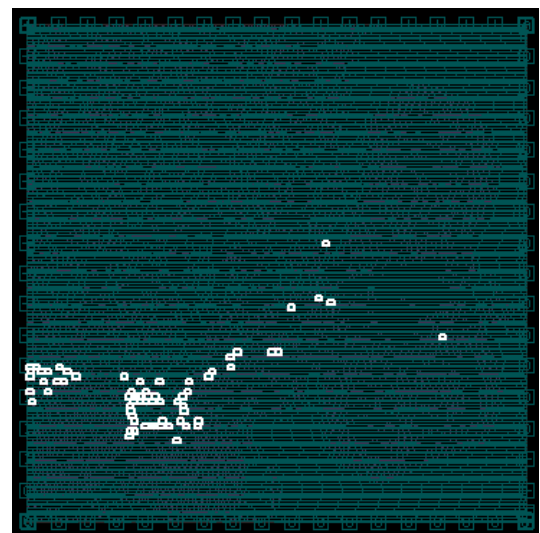


Figure 10: Layout plot of miniMIPS

## References

1. D. Ernst, et al., "Razor: a lowpower pipeline based on circuit-level timing speculation," 36<sup>th</sup> Int. Symp. on Microarchitecture, 2003.
2. A.K. Uht, "Going beyond worst-case specs with teatime," IEEE Computer, 17(3), 2004.
3. N.R. Shanbhag, "Reliable and efficient system-on-chip design," IEEE Computer, 17(3), 2004.
4. K. Hirose, et al., "Delay compensation flip-flop with in-situ error monitoring for low-power and timing-error-tolerant circuit design," Japanese Jour. of Applied Physics, 47(4), 2008.
5. T. Kehl, "Hardware self-tuning and circuit performance monitoring," Int. Conf. on Computer Design, 1993.
6. S. Das, et al., "Razor ii: In situ error detection and correction for pvt and ser tolerance," IEEE Jour. of Solid-State Circuits, 44(1), 2009.
7. T. Sato, et al., "A simple flip-flop circuit for typical-case designs for dfm," 8<sup>th</sup> Int. Symp. on Quality Electronic Design, 2007.
8. Y. Kunitake, et al., "A selective replacement method for timing-error-predicting flip-flops," Midwest Symp. on Circuits and Systems, 2011.
9. K. Yano, et al., "An automated design approach of dependable vlsi using improved canary ff," 7<sup>th</sup> Int. Workshop on Unique Chips and Systems, 2012.
10. H. Onodera, et al., "P2lib: process portable library and its generation system," Custom Integrated Circuits Conf., 1997.
11. A. Mizuno, et al., "Design methodology and system for a configurable media embedded processor extensible to vliw architecture," Int. Conf. on Computer Design, 2002.
12. miniMIPS, <http://opencores.org/> [Accessed: 19 May, 2015].
13. VDEC, <http://www.vdec.u-tokyo/> [Accessed: 19 May, 2015].