An MTTF-aware Design and Post-Silicon Validation Methodology for Adaptive Voltage Scaling

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2019

Yutaka MASUDA

Publications

Journal Article (Refereed)

[J1] Y. Masuda, T. Onoye, and M. Hashimoto, "Activation-aware slack assignment for time-to-failure extension and power saving," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 11, pp. 2217-2229, Nov. 2018.

[J2] Y. Masuda, T. Onoye, and M. Hashimoto, "Performance evaluation of software-based error detection mechanisms for supply noise induced timing errors," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E100-A, no. 7, pp. 1452-1463, July 2017.

International Conference Papers (Refereed)

[I1] Y. Masuda, J. Nagayama, H. Takeno, Y. Ogawa, Y. Momiyama, and M. Hashimoto, "Comparing voltage adaptation performance between replica and in-situ timing monitors," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, Nov. 2018.

[I2] Y. Masuda and M. Hashimoto, "MTTF-aware design methodology of error prediction based adaptively voltage-scaled circuits," in *Proceedings of Asia and South Pacific Design Automation Conference*, pp. 159-165, Jan. 2018.

[I3] Y. Masuda and M. Hashimoto, "Design and test of adaptively voltage scaled circuits," *SIGDA Student Research Forum at Asia and South Pacific Design Automation Conference*, Jan. 2018.

[I4] Y. Masuda, M. Hashimoto, and T. Onoye, "Critical path isolation for time-to-failure extension and lower voltage operation," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, Nov. 2016.

[I5] Y. Masuda, M. Hashimoto, and T. Onoye, "Hardware-simulation correlation of timing error detection performance of software-based error detection mechanisms,"

in *Proceedings of IEEE International Symposium on On-Line Testing and Robust System Design*, pp. 84-89, July 2016.

[I6] Y. Masuda, M. Hashimoto, and T. Onoye, "Measurement of timing error detection performance of software-based error detection mechanisms and its correlation with simulation," *ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, Mar. 2016.

[I7] Y. Masuda, M. Hashimoto, and T. Onoye, "Performance evaluation of software-based error detection mechanisms for localizing electrical timing failures under dynamic supply noise," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 315-322, Nov. 2015.

[I8] S. Iizuka, Y. Masuda, M. Hashimoto, and T. Onoye, "Stochastic timing error rate estimation under process and temporal variations," in *Proceedings of IEEE International Test Conference*, Sep. 2015.

International Conference Papers (Unrefereed)

[C1] M. Hashimoto and Y. Masuda, "MTTF-aware design methodology for adaptive voltage scaling," in *Proceedings of China Semiconductor Technology International Conference*, Mar. 2018 (invited paper).

Domestic Conference Papers (Unrefereed)

[D1] Y. Masuda, J. Nagayama, H. Takeno, Y. Ogawa, Y. Momiyama, and M. Hashimoto, "Comparing voltage adaptation performance between replica and in-situ timing monitors," in *Proceedings of IPSJ DA Symposium*, Aug. 2018 (in Japanese).

[D2] M. Sato, Y. Masuda, M. Hashimoto, "Experimental study on power reduction by approximate computing with voltage over-scaling," in *IEICE Technical Report*, VLD2017-123, pp. 205–210, Feb. 2018 (in Japanese).

[D3] Y. Masuda and M. Hashimoto, "MTTF-aware design methodology of error prediction based adaptively voltage-scaled circuits, " in *Proceedings of IPSJ DA Symposium*, pp. 139–144, Aug. 2017 (in Japanese).

[D4] Y. Masuda, M. Hashimoto, and T. Onoye, "Critical path isolation for time-to-failure extension and lower voltage operation," in *Proceedings of IPSJ DA Symposium*, pp. 32–37, Sep. 2016 (in Japanese).

[D5] M. Sato, Y. Masuda, S. Iizuka, T. Onoye, and M. Hashimoto, "Validating stochastic error rate estimation and improving its numerical stability," in *Proceedings of IPSJ DA Symposium*, pp. 38–43, Sep. 2016 (in Japanese).

[D6] Y. Masuda, M. Hashimoto, and T. Onoye, "Performance evaluation of software-based error detection mechanisms for localizing electrical timing failures under dynamic supply noise," in *Proceedings of IPSJ DA Symposium*, pp. 193–198, Aug. 2015 (in Japanese).

[D7] Y. Masuda, M. Hashimoto, and T. Onoye, "Performance evaluation of software-based quick error detection technique for localizing electrical failures due to dynamic power supply noise," in *Proceedings of IPSJ DA Symposium*, pp. 203–208, Aug. 2014 (in Japanese).

Summary

Aggressive device miniaturization due to VLSI technology scaling has been improving the average device performance. Circuits, meanwhile, have become sensitive to static manufacturing variability and dynamic environmental fluctuation. These static and dynamic variations directly lead to circuit reliability degradation. To overcome variability mentioned above, a traditional worst-case (WC) design gives design and operational margins to ensure correct circuit operation in design time and in field, respectively. However, as the performance variation becomes significant, such margins tend to be too painful for designers. Therefore, the conventional WC design with guard-banding is becoming less efficient.

The most effective tuning knob for post-silicon performance compensation is supply voltage control, and then adaptive voltage scaling (AVS) is intensively studied. AVS is expected to minimize process, voltage, temperature, and aging (PVTA) margin of each chip and allocate only a small margin taking into account the entire lifetime. The conventional PVTA margins, which are determined by the worst chip across all the variation sources, are excessive in most of the chips, and they can be exploited as the source of power reduction.

To put the AVS circuit into practical use, a designer needs (1) design methodology, (2) performance evaluation in design time, and (3) post-silicon validation methodology. In the AVS circuit, the sensors, which estimate the timing slack of the main logic and detect/predict the timing errors, are embedded, and the supply voltage is adjusted referring to the sensor output. Note that timing errors can occur even with this AVS due to, for example, insufficient sensor insertion. Therefore, to design the reliable AVS, the design parameters such as sensor type and insertion place should be determined carefully, and then the circuit performance needs to be validated in terms of power and lifetime. After the design parameters are fixed and the circuit performance is estimated, the design needs to be validated. Note that even in the well-designed AVS circuit, some fast-transient delay fluctuation such as supply noise may induce timing errors. Hence, in the post-silicon validation, each chip needs to be verified in terms of whether the AVS can appropriately work in field under various operation conditions. In the post-silicon validation, once an unexpected system behavior is observed, the circuit operation is analyzed. In this analysis, the most challenging tasks is error localization since the time interval between the error occurrence and the detection of such an abnormal behavior

is quite long. Due to such a long error detection latency, it is difficult to know when and where the timing error occurred. Consequently, to facilitate the error localization, the post-silicon validation methodology which can quickly detect the timing error is essential. Here, as a performance evaluation framework, a stochastic error rate estimation method, which quickly estimates a mean time to failure (MTTF) and average power dissipation, was proposed. However, neither the design methodology which enhances the AVS performance under an MTTF constraint nor the post-silicon validation method which facilitates the timing error localization in AVS operation are not fully studied yet.

This thesis studies the design and post-silicon validation of AVS and proposes a design and post-silicon validation methodology for the AVS circuit. The proposed design methodology consists of three steps: (1) select the type of sensor, (2) optimize the main logic under AVS, and (3) insert sensors into the optimized main logic. In the first step of the sensor selection, this thesis discusses supply voltage reductions achieved by AVS circuits with different sensors, i.e., timing error predictive FF and critical path replica. In this work, we give the MTTF as a design constraint and compare the trade-offs of clock period and average supply voltage between AVS circuits with these two sensors.

In the second step of the main logic design, this thesis introduces the MTTF as a design constraint and optimizes the design with an activation-aware slack assignment (ASA). The MTTF constraint helps explore a set of necessary operating conditions, such as clock period and supply voltage, and reduces the operation margin from the WC design while keeping the target MTTF. This margin reduction directly leads to the supply voltage reduction. ASA, meanwhile, gives timing slacks to non-intrinsic active critical paths by ECO, where non-intrinsic critical paths are timing paths whose slacks were originally large but are reduced by downsizing and replacement to high-V_{th} cells for power savings. Thus, ASA reduces the number of active critical paths whose delays are very close to those of the intrinsic critical paths, i.e., timing paths whose slacks cannot be reduced by re-synthesis, replacement to low-V_{th} cells, and sizing. In this case, we can expect that circuits with ASA have fewer paths where timing errors are likely to occur, which can reduce the number of monitoring paths with AVS and thus can contribute to facilitating the following sensor insertion. Also, the reduction of active critical paths helps to extend the MTTF and thus reduce the supply voltage. Therefore, ASA can further improve performance from the simple MTTF-aware operation of conventionally designed main logic.

In the third step of the sensor insertion, this thesis aims at maximizing the MTTF to reduce the power dissipation, which is based on a hypothesis that a circuit with the longer MTTF has a larger room for power saving. To maximize the MTTF, the supply voltage should be adjusted frequently and thus timing critical paths should be monitored frequently. Based on this consideration, this thesis proposes a novel insertion method that maximumly decreases the sum of gate-wise timing failure probabilities. Note that the timing failure probability of a flip-flop (FF) is the joint probability of activation and timing violation probabilities of the FF, and the gate-wise failure probability is calcu-

lated from timing failure probabilities of endpoint FFs. By exploiting the information on the paths with the higher timing failure probability, the proposed sensor insertion makes AVS efficiently monitor the timing-critical and highly-active FFs. Moreover, by maximizing the sum of gate-wise failure probabilities, the proposed sensor insertion can cover the larger set of instances that can contribute to causing timing errors.

With the proposed design methodology consisting of these three steps, 38.0% power reduction is achieved while satisfying the target MTTF. This work also experimentally confirmed that simultaneous optimization of sensors and main logic synergistically enhances the performance and reliability of AVS. For example, proposed sensor selection and insertion methodology achieved the target MTTF whereas straightforward slack-oriented sensor insertion did not satisfy the target MTTF at all. Moreover, the proposed main logic optimization is highly compatible with the sensor optimization and further improves the AVS performance, e.g, the proposed ASA further saves power by 10.6% from the AVS circuit without main logic optimization.

As for the post-silicon validation methodology, this thesis devises the error detection mechanisms for short latency (EDM-L) and evaluates the performance of EDM-L for timing error localization with a noise-aware logic simulator and 65-nm test chips assuming the following two EDM-L usage scenarios: (1) localizing a timing error occurred in the original program and (2) localizing as many potential timing errors as possible. Simulation results show that the EDM-L cannot locate supply noise induced timing errors in the original program in the first scenario, but it detected 86% of non-masked errors in the second scenario, which mean the EDM-L performance of detecting supply noise induced timing errors affecting execution results is high. Hardware measurement results show that the EDM-L detects 25% of original timing errors and 56% of non-masked errors. These measurement results were not consistent with the simulation results. We found that this inconsistency came from (1) the design of the power distribution network, and (2) the definition of FMAX used for evaluation. By updating the simulation setup, the EDM-L performance evaluated by the simulation was consistent with that by the chip measurement. The devised EDM-L helps to localize the supply noise induced timing errors in post-silicon validation and thus contributes to improving reliability of the AVS circuit.

The design and post-silicon validation methodology established in this thesis helps to construct the reliable AVS, which overcomes the PVTA variation and thus dramatically improves the performance of VLSI circuit. The proposed design methodology enables designers to provide the MTTF aware design flow which is essential for the reliable VLSI design. The post-silicon validation methodology provided by this thesis helps to localize the timing error with short latency. This quick error localization substantially reduces the debugging costs in the post-silicon validation and therefore mitigates the design time-to-market, which is the one of the most serious constraints in the VLSI design.

Acknowledgments

First of all, I would like to express my deepest gratitude to Professor Masanori Hashimoto in Osaka University for providing me a precious opportunity and an excellent environment to study as a doctoral student in his laboratory. All of my productive researches are credited to none other than him. His advanced perspective and thoughtful advice led me to successful achievements.

I would like to appreciate Professor Takao Onoye and Professor Tatsuhiro Tsuchiya in Osaka University for detailed reviews and insightful suggestions.

I would like to express my sincere appreciation to Professor Takao Onoye in Osaka University for providing me a precious opportunity to study VLSI design and testing in his laboratory when I was pursuing the Bachelor's degree and the Master's degree.

I would like to express my sincere appreciation to Assistant Professor Jaehoon Yu in Osaka University and Professor Yoshinori Takeuchi in Kindai University for precious discussions and support.

My appreciation also goes to Mr. Yoichi Momiyama, Mr. Jun Nagayama in Socionext Inc. for technical discussions and suggestions.

I would like to demonstrate my gratitude to Mr. Shoichi Iizuka in Panasonic Corporation and Mr. Masahiro Sato in NTT docomo Inc. for daily discussions and their support when I was pursuing the Master's degree in Osaka University.

I would like to thank other colleagues who belong or belonged to the Integrated System Design Laboratory in Osaka University for daily discussions and their support: Dr. Tomoki Sugiura, Mr. Koichi Mitsunari, Mr. Wang Liao, Mr. Ryutaro Doi, Mr. Jun Chen, Mr. Tai-Yu Cheng, Mr. Ryo Shirai, and Mr. Kosuke Nishi. I express my heartfelt thanks to all members of Integrated System Design Laboratory in Osaka University for having an interesting and comfortable time in the laboratory. I would like to thank Mrs. Asako Murakami for her various support throughout my student life. I also would like to thank all of my friends for having good times.

Finally, I give my thank to my family for supporting my livelihood.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter describes the background and objectives of this thesis. This thesis focuses on adaptive voltage scaling (AVS), which adaptively controls the supply voltage to save the power dissipation while preventing timing error occurrence. This thesis studies the design and post-silicon validation of AVS. Following sections firstly explain the background, secondly discuss the concept of AVS, requirements, and challenges for AVS, and lastly, describe the objectives of this thesis.

1.1 Background

In 1965, Gordon Moore provided a principle that the cost of integrated circuits was minimized by doubling the number of transistors on an integrated circuit every year. To achieve the cost reduction along with improved transistor performance, controlled power dissipation, and improved functionality, Dennard *et al.* provided the laws for the classical complementary metal oxide semiconductor (CMOS) scaling in 1974. These fundamental principles have been adopted in recent decades and have provided a simple and straightforward path for technology scaling.

Continued technology scaling has improved the device performance and increased transistor density [1, 2]. Thanks to this significant performance improvement, the semiconductor industry has been grown up with expanding the market. For example, since the late of the 1990s, the semiconductor industry expanded rapidly due to the continued steady growth of mobile, digital consumer and entertainment markets [3]. Also, with the dramatic advancement of information technology, several innovative technologies are being implemented [4], where typical application fields are biotechnology, robotics, health, artificial intelligence, autonomous vehicles, and virtual reality. These technologies combined with cloud computing, internet-of-things (IoT), and big data enable us to access various types of information and utilize them. In this way, aggressive device miniaturization due to technology scaling has been improving the average device performance, generating innovative technology, and expanding the application field.

Table 1.1: Prediction of power supply voltage and threshold voltage reduction [9].

Year	2015	2017	2019	2021
V_{dd} [V]	0.80	0.75	0.70	0.65
V_{th} [mV]	351	336	333	326

As the density and size of the chips and systems continue to increase, power savings become one of the most critical missions for the semiconductor industry [5, 6]. For example, the chip consumes a large amount of dynamic power when the enormous number of transistors switch at very high rates [7]. In addition, as transistors have become so small, off-state leakage current flows and additional static power is dissipated [6]. Such enormous power dissipation makes the packaging and cooling much difficult under limited volume and cost [5]. In another case, the high power dissipation may prevent the low-power applications such as mobile battery-operated systems from achieving the desired performance. As a result, most circuit designers are required to confront the problem of delivering high performance under severe power constraints [8].

One conventional strategy toward the low power design is to reduce the supply voltage since the dynamic power, which occupies a significant portion of power consumption, is proportional to the square of the supply voltage [7]. Tab. 1.1 shows the prediction of the power supply voltage (V_{dd}) and the threshold voltage (V_{th}) of logic devices, which was reported in International Technology Roadmap for Semiconductors (ITRS) roadmap [9]. From Tab. 1.1, we can see that it is expected to keep the continuous supply voltage and threshold voltage reductions along with technology scaling [9].

Here, to progress further technology scaling and supply voltage reduction, we need to overcome a significant reliability challenge, namely, the performance variation. As technology node advances, integrated circuits have become more sensitive to static manufacturing variability [10–13] and dynamic environmental fluctuation, e.g., supply voltage noise [14–16] and device aging [17–25]. Manufacturing variability is caused by the inability to precisely control the fabrication process at small-feature technologies [11, 12]. Due to this difficulty, device parameters such as the device length, doping concentrations, and oxide thickness vary and thus the delay characteristic of the device significantly changes. As a result, the circuit delay variation originating from the process variation becomes critical.

Supply noise occurs due to switching current I and mainly comes from resistive IR drop and inductive $L\frac{dI}{dt}$ noise. R and L represent the resistance and the inductance of the power mesh network, power pads, and device package [15]. Along with technology scaling, the supply voltage becomes lower as mentioned above. Here, in general, the cell delay T and supply voltage V_{dd} follows the α-power law MOS model in [26].

$$T = \beta\{\frac{V_{dd}}{(V_{dd} - V_{th})^{\alpha}}\}. \tag{1.1}$$

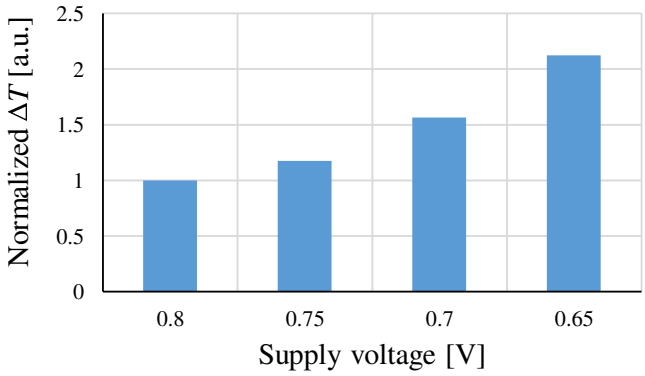

Figure 1.1: The impact of supply noise on delay fluctuation becomes larger in lower supply voltage. Y-axis is normalized with ΔT at the supply voltage of 0.8V.

Note that α is a coefficient to express the carrier-velocity saturation effect [27] and β is the proportional factor. Eq. (1.1) tells us that the impact of supply noise on the circuit delay increases at a lower supply voltage even when the same amount of supply noise happens. As an example, let us consider the situation where the supply noise of ΔV_{dd} happens. In this case, the cell delay changes as the following equation.

$$T' = \beta\{\frac{V_{\mathrm{dd}} - \Delta V_{\mathrm{dd}}}{(V_{\mathrm{dd}} - \Delta V_{\mathrm{dd}} - V_{\mathrm{th}})^{\alpha}}\}, \tag{1.2}$$

where T' is the cell delay when supply noise happens. Note that V_{dd} is replaced by $V_{\mathrm{dd}} - \Delta V_{\mathrm{dd}}$ to take into account the supply noise impact on the cell delay. Here, from Eqs. (1.1) and (1.2), the cell delay increase due to the supply noise can be approximated as follows.

$$\Delta T = T' - T = \beta\{\frac{V_{\mathrm{dd}} - \Delta V_{\mathrm{dd}}}{(V_{\mathrm{dd}} - \Delta V_{\mathrm{dd}} - V_{\mathrm{th}})^{\alpha}} - \frac{V_{\mathrm{dd}}}{(V_{\mathrm{dd}} - V_{\mathrm{th}})^{\alpha}}\}, \tag{1.3}$$

where ΔT is the delay increase due to supply noise. Then, let us substitute each pair of V_{dd} and V_{th} in Tab. 1.1 to Eq. (1.3) as an example. Also, let us set α to 1.3, which is the value reported in [6], and ΔV_{dd} to 100 mV. Fig. 1.1 shows the cell delay increase due to supply noise which is derived from Eq. (1.3) and the above settings. Fig. 1.1 shows that the impact of supply noise on cell delay increases dramatically as the supply voltage decreases along with technology node improvement, e.g., ΔT at the supply voltage of 0.65V is 2.12 times of the ΔT at 0.8V.

In addition to the supply noise, device aging which is another temporal variation degrades the performance gradually in the field. Aging is represented by negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), and hot carrier injection (HCI). Similarly to supply noise, the impact of aging on cell delay becomes significant with according to the supply voltage reduction. For example, in Eq. (1.1), if the threshold voltage degrades by ΔV_{th} due to aging, the ΔT can be represented as

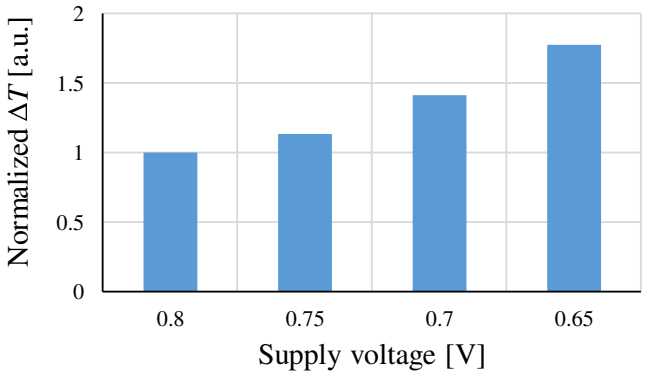

Figure 1.2: The impact of aging on delay increase becomes larger in lower supply voltage. Y-axis is normalized with ΔT at the supply voltage of 0.8V.

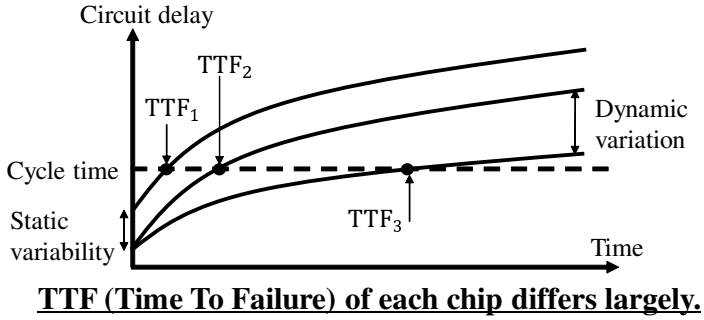

Figure 1.3: TTF variation due to static and dynamic delay variation.

follows.

$$\Delta T = \beta\{\frac{V_{dd}}{(V_{dd} - \Delta V_{th} - V_{th})^{\alpha}} - \frac{V_{dd}}{(V_{dd} - V_{th})^{\alpha}}\}. \tag{1.4}$$

Fig. 1.2 shows a similar example to Fig. 1.1, where pairs of V_{dd} and V_{th} in Tab. 1.1 and ΔV_{th} of 30mV are substituted to Eq. (1.4). Fig. 1.2 indicates that the impact of threshold voltage degradation on cell delay increases as the supply voltage decreases.

These static and temporal variations directly lead to circuit reliability degradation. Fig. 1.3 illustrates the time to failure (TTF) variation originating from the stochastic properties of manufacturing variability and aging process. Fig. 1.3 includes three curves, where each curve represents the circuit delay transition of each chip. Statistical characterization of manufacturing variability is studied comprehensively in the last decades [11,13,28], and its statistical modeling is now a common practice. Also, threshold voltage variation due to aging such as NBTI fluctuates statistically. Due to these statistical properties, the time when the circuit delay exceeds the clock cycle time, which corresponds to TTF, varies as illustrated in Fig. 1.3. To overcome variability mentioned above, a traditional worst-case (WC) design gives design and operational margins to ensure correct circuit operation in design time and in a field, respectively. However, as

the performance variation becomes significant, such margins tend to be too painful for designers. Therefore, the conventional WC design with guard-banding is becoming less efficient, and an adaptive post-silicon performance compensation is eagerly demanded as a promising countermeasure.

1.2 Adaptive Voltage Scaling

This section explains the concept of adaptive voltage scaling (AVS), which is a promising countermeasure for the performance variation, discusses requirements and challenges for AVS, and describes problems of conventional works.

1.2.1 Concept

The most effective tuning knob for post-silicon compensation is supply voltage control. AVS, which is the design concept that each chip dynamically adjusts the supply voltage taking into account its performance, is intensively studied [28–53]. Note that AVS is different with dynamic voltage and frequency scaling (DVFS) [54] and boosting [55, 56], which are another V_{dd} and frequency control techniques and currently used in some commercial processors. In a DVFS system, the supply voltage and operating frequency are dynamically adjusted according to workload to reduce the overall energy consumption [29]. In boosting, the clock frequency will be increased from the nominal one under the power, temperature, and current specification limit [55]. While DVFS and boosting select the pair of the supply voltage and clock frequency referring to the workload and monitored performance/power/temperature, the reference pair of the voltage and frequency, which are mostly determined in design time, include the timing guardband for keeping the safe operation. On the other hand, AVS is expected to minimize process, voltage, temperature, and aging (PVTA) of each chip and allocate only a small margin for keeping correct operation throughout the entire life as shown in Fig. 1.4. The conventional PVTA margins, which are determined by the worst chip across all the variation sources, are excessive in most of the chips, and they can be exploited as the source of power reduction.

The most popular strategy of the AVS implementation studied so far is to use sensors as shown in Fig. 1.5. In this AVS, sensors are embedded, and they detect or predict the timing error occurrence. Then, V_{dd} control logic adjusts the supply voltage referring to the sensor output. Once the timing slack of the monitoring paths or flip-flops (FFs) reaches the required lower bound, the supply voltage is increased, and the instructions are re-executed if necessary. Otherwise, the supply voltage is decreased for power savings. Here, if the inserted sensors perfectly work, the timing slack of the critical paths can be frequently monitored and hence the supply voltage is controlled to the minimum one while keeping the correct operation. Motivated by this expectation, various types of sensors were studied in a last decade [28–53]. Note that there are two AVS strategies

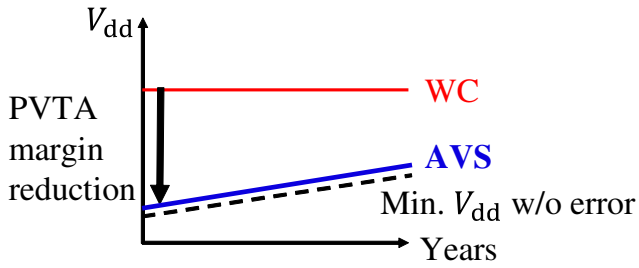

Figure 1.4: Supply voltages of AVS and the conventional WC design in device lifetime. Ideal AVS minimizes PVTA margin of each chip.

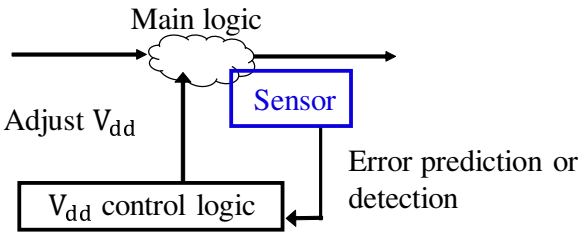

Figure 1.5: AVS using sensors. The embedded sensor detects or predicts the timing error occurrence. V_{dd} control logic adjusts the supply voltage referring the sensor output.

in literatures: error detection and recovery based control with, for example, Razor [29], and error prediction and prevention based control with in-situ timing sensors [1] or critical path replica [30]. In both the strategies, sensors are embedded to detect/predict timing errors, and the supply voltage is controlled according to the sensor outputs.

Razor [29] was proposed at the earlier stage of the AVS. Razor FF consists of the main FF, a shadow latch, a comparator which compares the output of the main FF and the latch, and an error recovery mechanism. Since the shadow latch is transparent in the positive phase of the clock, the timing error in the main FF is expected to be detected in comparator. When the comparator finds the timing error in the main FF, the error recovery mechanism restores the logic in the main FF. Thus, Razor enables us to scale the supply voltage to the point of the first failure in a die for a given frequency [29]. Therefore, all margins due to global and local PVT variations are eliminated, resulting in significant energy savings. However, Razor has a disadvantage that it introduces significant hold-time constraints which cause large overhead and difficult timing closure. To tackle the overhead problem of Razor, different Razor-flavor sensors such as Bubble Razor [38], Razor-lite [40], and iRazor [28] were proposed. These techniques require a re-execution mechanism to correct timing errors. The re-execution is performed through architectural replay, which is often integrated into high-performance processors to sup-

[1] There are several names for the same structure: canary FF [57], slack monitor [42], and error predictive FF [35]

port branch prediction. However, it is impracticable for general sequential circuits and simple processors [35].

At around the same time when Razor arose, another sensor called critical path replica [30] was proposed. Critical path replica consists of a delay-chain which mimics the critical path of the actual design. The propagation delay through this replica path is monitored, and voltage and frequency are scaled as long as the replica path meets timing [32]. Replica can be implemented separately with the main logic unlike in-situ Razor, and hence the replica elements are non-intrusive and are well suited for monitoring global variations on the chip. However, replica suffers from capturing the variations that are local to actual circuits such as random manufacturing variations and circuit aging [58].

Another promising sensor for the AVS is double sampling based error predictive FF [57]. Unlike the Razor technique which detects and corrects the timing error, the error predictive FF predicts the timing error by checking whether the intentionally delayed data transition causes the error [59]. Besides, this sensor can accurately capture the actual path delay [58], which is a distinct difference with replica and other generic sensors such as ring oscillators. Therefore, the error predictive FF has a possibility of exploiting timing margins for both global and local variation to power reduction. Moreover, the FF does not need the hold buffers unlike Razor as far as the main logic satisfies the hold timing constraint.

As explained above, various types of sensors for AVS were proposed aiming at PVTA margin elimination. However, the AVS still has a risk of causing timing errors, which is exemplified in Fig. 1.6. For example, when the inserted sensors such as error predictive FFs are not activated frequently enough, AVS cannot check the timing slack of critical paths and thus cannot adjust supply voltage appropriately. In this case, the delay fluctuation due to, for example, supply noise may induce timing errors before AVS controls the supply voltage as shown in Fig. 1.6. In another case, when we use replica or some generic monitors, the sensors involve delay mismatch between the sensed delay and actual delay due to random delay variation, which may cause the timing error due to excessively reduced supply voltage. From this point of view, for practical AVS which precisely estimates the timing slack and controls the supply voltage, the designer needs to carefully determine the type of sensors, their designs, and their locations, which will be discussed in Section 1.2.2.

1.2.2 Requirements

Fig. 1.7 shows the key processes for realizing the AVS circuit: (1) design, (2) performance evaluation in design time, and (3) post-silicon validation. The first process is to design the AVS circuit. In design time, designers need to determine the design parameters of AVS, e.g., which types of sensors are used, how many sensors are inserted, and where to embed sensors. The next process is the performance evaluation. If the AVS

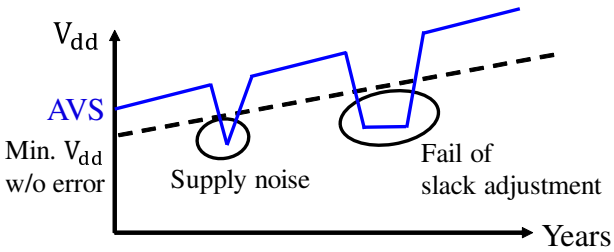

Figure 1.6: If the sensors do not work well, timing error may happen in the AVS circuit.

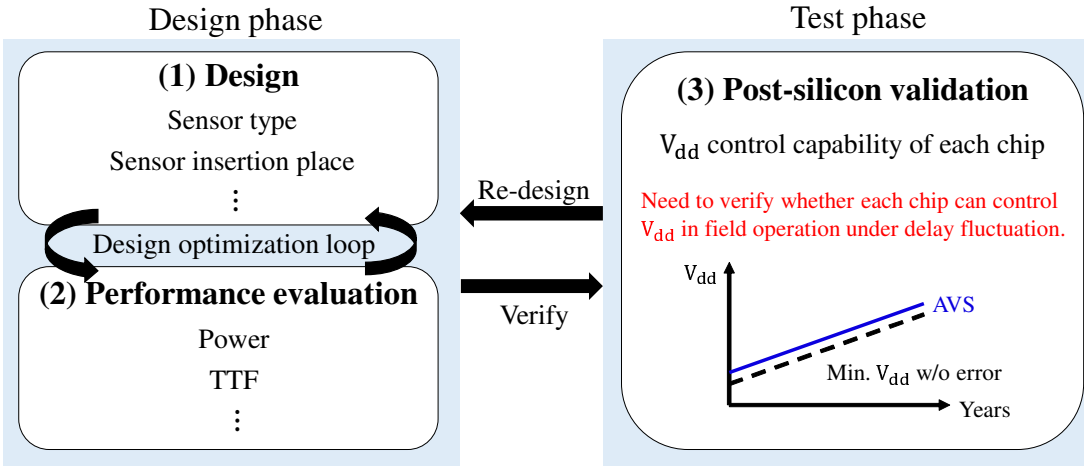

Figure 1.7: Key processes for realizing the AVS circuit: (1) design, (2) performance evaluation, and (3) post-silicon validation.

circuit does not satisfy the design constraint such as the lifetime and power, designers may go back to the design process and change the design parameter. If the design meets the given constraints, the design will be verified in the third process, i.e., post-silicon validation. In the third process, each chip is verified in terms of whether the AVS can appropriately work in field under various operation conditions. If the verification tells us that AVS fails to keep correct operation in some conditions, we may need to increase the operation margin or re-design the AVS circuit. The remaining of this subsection will discuss the requirements for the design, performance evaluation, and post-silicon validation.

In the design process, designers may aim to minimize the power dissipation. For this purpose, the inserted sensors need to predict or detect timing errors appropriately to adjust the supply voltage. In other words, the sensors are required to estimate the timing slack of critical paths accurately. Otherwise, the voltage control logic may fail to adjust the supply voltage and thus delay fluctuation by aging and supply noise may cause timing errors, which was previously explained with Fig. 1.6. Note that TTF, which is the length of time until a chip starts to cause timing errors, can be a metric to evaluate

the error detection or prediction performance of AVS quantitatively. In another case, inadequate sensor insertion cannot reduce design and operation margins. To avoid these unsuccessful AVS designs and eliminate unnecessary margins, the sensor selection and insertion need to be validated in terms of TTF and margin reduction.

Then, the performance evaluation of the AVS design is discussed. In this evaluation, the power dissipation and the maximum frequency may be the good metric. In addition, the design constraints such as the area and TTF can be examined. If the design quality is not good enough, we need to re-design the circuit and re-evaluate the performance repeatedly with compromising a given design time to market. Here, a naive approach to calculate the TTF is to execute gate-level simulation in the huge variation parameter space repeatedly. However, the probability that actual timing errors occur is quite low[2], and hence the simulation time required to reproduce these errors is prohibitively long. For example, when we evaluate the rate of timing errors that occur once per one month, the simulation time exceeds 10^8 years [60] even for a simple embedded processor.

Recently, a stochastic framework that estimates a mean TTF (MTTF) is proposed in [60, 61]. Iizuka *et al.* model circuit operation under dynamic delay variations as a continuous-time Markov process [60]. The continuous-time Markov process modeling enables us to estimate the MTTF in a reasonable time. In a test case, the MTTF is estimated 10^{12} times faster than a logic simulator. Reference [61] extended the framework proposed in [60] to consider manufacturing variability, temporal environmental fluctuation, and aging in the MTTF estimation. Also, [61] takes into account workload dependent path activation probabilities. With this framework, it becomes possible to know, for example, the trade-off between the MTTF and supply voltage. Therefore, this thesis utilizes this stochastic framework for the performance evaluation.

After the design parameters are fixed and the circuit performance is estimated, the design needs to be validated. Here, validation tests can be classified with pre-silicon validation and post-silicon validation. Pre-silicon validation inspects the design thoroughly by time-consuming functional verification and code review to exclude functional bugs [62] before the tape-out. In the post-silicon validation, the coverage is severely limited by slow simulation speeds [63]. As a result, eliminating all bugs in the design before tape-out is very difficult [64] and thus a dependence on post-silicon validation to verify remaining bugs is increasing.

The post-silicon validation gives a wide variety of test patterns to the fabricated chip at various operating conditions. One main goal of the post-silicon validation is to know error occurrence conditions in fabricated chips to debug errors and thus further improve the design quality and reliability. Note that for some types of chips which are mass-produced, designers may just throw away defective chips at the cost of yield. In this validation, designers need to debug not only logic errors but also electrical timing errors, which cause a system failure in a logically correct design due to a very specific electrical property of the chip, e.g., the supply voltage variation, temperature gradient, crosstalk

[2]Otherwise, such circuits with frequent error occurrence are useless.

noise, and so on [65]. This validation step is essential for the reliable AVS circuit since some fast-transient delay fluctuation such as supply noise may induce timing errors even in the well-designed AVS circuit. Also, the localization of the electrical timing error is very significant since the reproduction of this error is much difficult compared with functional errors. Therefore, it is crucial for designers to verify the AVS circuit in various operating conditions and find out occurrence conditions of electrical timing errors.

In the post-silicon validation, once an unexpected system behavior is observed, we start on analyzing the circuit operation. In this analysis, we need to (1) notice error occurrence, (2) localize the error in place, e.g., ALU and cache controller, and time, and (3) manifest the occurrence condition [65]. The most efforts for this analysis are made in (1) and (2) [66], and hence reducing these efforts are highly demanded. Error occurrence is often detected by observing abnormal behaviors, such as system crash, segmentation fault, and invalid opcode. End-result-check [67], which compares the execution result with the expected result, can also be used to find error occurrence. The next step is error localization, and it is challenging since the time interval between the error occurrence and the detection of such an abnormal behavior is quite long. It sometimes reaches billions of clock cycles [68]. Due to such a long error detection latency, it is difficult to know when and where it occurred, since the trace buffer such as Instruction Footprint Recording and Analysis (IFRA) [69], which is often used to record signals on a chip for post-silicon debug, has limited record depth of, for example, thousands of clock cycles. Therefore, reducing the error detection latency is essential to facilitate the error localization.

1.2.3 Challenges for AVS and Problems of Conventional Works

This subsection discusses the challenges for design and test requirements for AVS and problems of conventional works.

First, let me explain the design challenges. One difficult challenge is to determine where to insert sensors for minimizing power while satisfying area and TTF constraints. As previously mentioned in Sec. 1.2.2, AVS circuit may fail to adjust the supply voltage when the margin checking in the chip is not perfect or when fast-transient supply noise happens. In other words, even if the AVS is implemented, the possibility of timing error occurrence may not be completely reduced to zero. From this perspective, designers need to determine the set of design parameters taking into account the extremely low probability of error occurrence to guide design optimization in design time. Here, conventional works [28–53] do not consider the TTF of the AVS circuit and thus their design may not ensure the target TTF. Some of conventional works claim that by tuning the operational margin through some design parameter optimization, the possibility of timing error occurrence can be reduced to almost zero and the time to failure can be extended to over years [61]. However, any systematic tuning strategies are not provided

to designers. Naive ad-hoc optimization can degrade the performance of the AVS circuit significantly. Therefore, to enhance the performance under AVS, designers need to determine the design parameters taking into account the TTF of every chip in field operation.

Another design challenge is how to optimize the main logic under AVS. Note that in the conventional works, the identical main logic is used regardless whether or not AVS is implemented. The conventional circuit optimization forces the circuit to increase the number of critical paths to decrease the area and power. For example, low-V_{th} cells included in non-critical paths may be replaced with high-V_{th} cells for the leakage power reduction. As a result, the optimized design achieves the lower power/area but has a lot of critical paths. Monitoring such a large number of critical paths needs a large number of sensors, and thus it requires enormous area overheads. From this point of view, to implement AVS systems that fully exploit run-time adaptation and eliminate the redundant margin, we should pay attention to the main logic circuit under AVS in addition to the sensing circuit. However, the conventional works do not customize the main logic for AVS. In this case, the following sensor insertion becomes much complex and difficult tasks since the AVS circuit needs to achieve the target TTF while only a small portion of critical paths can be monitored.

Then, the challenges for post-silicon validation is discussed. As mentioned in Section 1.2.2, quick detection of the errors occurrence is crucially important to facilitate the validation. Recently, software-based error detection techniques, which add error detection capability to test patterns, are used for error localization in the post-silicon validation. While conventional hardware-based error detection approaches, e.g., assertion-based detection [70, 71], require additional hardware resources and need to consider where and how to implement such structures [72], software-based approaches do not need error detection hardware. In other words, the software-based approach can eliminate the risk of the design change by embedding error detection elements which are needed by hardware-based approaches.

There are several software-based error detection techniques, which include quick error detection (QED) transformation [67,68,73] and error detection mechanism (EDM) transformation [74,75]. QED decomposes the input program into blocks and duplicates each block within the program at the assembly level. Also for every pair of the original and duplicated blocks, QED inserts a register-level consistency check that compares calculation results. With this fine-grained checking, QED succeeded in dramatically reducing error detection latency. Reference [68] reported that for specific logic errors, QED improved error detection latency by six orders of magnitude, i.e., from billions of cycles to a few thousand cycles. This shorter error detection latency helps improve the efficiency of post-silicon validation.

EDM adds data and code redundancy to an input program written in a high-level source language (e.g., C and C++), and generates a special program. Rebaudengo *et al.* [74] proposed EDM for detecting soft errors based on the consideration that data and

code redundancy can be used not only for verifying software but also for detecting error occurrences in hardware, e.g., logic errors and timing errors in register and memory. Reference [74] reported that for random bit flips injected to data memory the error detection coverage was over 90%. In [75], the coverage of over 80% was achieved for a single bit flip that occurred in registers. Here, various programs, e.g., random instruction tests, architecture-specific focused tests, and end-user applications such as operating systems and games can be given as an input program. The main advantage of EDM transformation lies in the fact that it can be applied to a high-level source code independent of the underlying hardware.

Here, for the post-silicon validation of the AVS circuit, not only logic error localization but also electrical timing error localization is essential. Especially, the localization of the supply noise induced timing errors may be one of the most important missions for the AVS circuit. While fast transient supply noise can cause sudden large delay fluctuation in the circuit, e.g., within a few nanoseconds, the sensors in AVS may fail to monitor timing slacks of critical paths in this clock cycle. Even if the sensor can fortunately monitor the timing slack, it may be difficult for the voltage control logic to adjust the supply voltage according to the delay fluctuation speed of the fast supply noise. Therefore, such supply noise has a potential to cause timing errors even when the AVS is well-designed, and hence the validation of this timing error is significant to ensure the design quality. However, in conventional works, the performance of software-based error detection techniques for the electrical timing error has not been studied explicitly, and their effectiveness against supply noise induced timing errors is not clear.

1.3 Objective of This Thesis

The main objective of the thesis includes constructing the design and post-silicon validation methodology for establishing reliable AVS. For such a reliable AVS, this thesis takes into account the MTTF of the AVS circuit and studies the MTTF-aware design methodology of the AVS circuit. To achieve the target MTTF while minimizing the timing margin, the sensor type selection and the sensor insertion need to be carefully performed. Based on this consideration, this thesis divides the designing AVS into three components: sensor type selection, the main logic optimization for facilitating the sensor insertion, and the sensor implementation to the main logic. This thesis attacks each component, combines them, and constructs a design methodology. In addition, to keep the correct operation in various operating conditions after the fabrication, this thesis studies a post-silicon validation methodology. This thesis focuses on the supply noise induced timing error, which is one of the most critical errors in AVS operation, and thus studies the error localization methodology for ensuring the reliability of the AVS against the supply noise.

Fig. 1.8 illustrates the structure of the thesis including the MTTF-aware design methodology and the test methodology. The proposed design methodology consists

of three steps: (1) select the type of sensor, (2) optimize the main logic under AVS, and (3) insert sensors to the optimized main logic.

In the first step of the sensor selection, Chapter 2 discusses supply voltage reductions achieved by AVS circuits with different sensors. This chapter focuses on error prediction based AVS and thus selects two representative sensors as candidate: in-situ sensors [35, 42, 57] and critical path replica [30, 33, 36]. Remind that in the error prediction based AVS, any error recovery mechanisms are not necessary as long as the prediction is appropriate, whereas the error detection and recovery strategy requires a re-execution mechanism for error correction which is difficult to implement in general sequential circuits. This thesis quantitatively evaluates the average supply voltage taking into account manufacturing variability at time zero, subsequent voltage elevation due to aging and dynamic supply noise. Depending on the requirement of TTF, the achievable trade-off between clock period and average supply voltage becomes different. This work gives the MTTF as a design constraint and compares the trade-offs of AVS circuits with different sensors. For such MTTF-aware trade-off analysis, this thesis utilizes a stochastic framework proposed by Iizuka *et al.* [61].

In the second step of the main logic design, Chapter 3 explicitly introduces the MTTF as a design constraint and optimizes the design with an activation-aware slack assignment (ASA). The MTTF constraint helps explore a set of necessary operating conditions, such as clock period and supply voltage, and reduces the operation margin from the WC design while keeping the target MTTF. This margin reduction directly

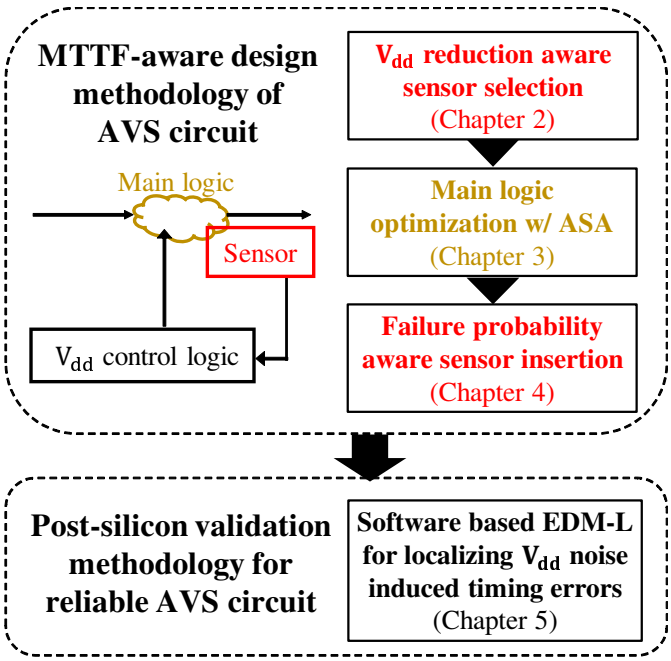

Figure 1.8: Overall structure of this thesis.

leads to the supply voltage reduction. ASA, meanwhile, gives timing slacks to non-intrinsic active critical paths by engineering change order (ECO), where non-intrinsic critical paths are timing paths whose slacks were originally large but are reduced by downsizing and replacement to high-V_{th} cells for power savings. Consequently, ASA reduces the number of active critical paths whose delays are very close to those of the intrinsic critical paths, i.e., timing paths whose slacks cannot be reduced by re-synthesis, replacement to low-V_{th} cells, and sizing. In this case, it can be expected that circuits with ASA have fewer paths where timing errors are likely to occur, which can reduce the number of monitoring paths with AVS and thus can contribute to facilitating the following sensor insertion. Also, the reduction of active critical paths helps extend the TTF since the timing failure probability of such paths can be dramatically decreased. This TTF extension can be converted to the supply voltage reduction since the circuit can achieve the target MTTF at the lower supply voltage. Therefore, ASA can further improve performance from the MTTF-aware operation.

In the third step of the sensor insertion, Chapter 4 aims at maximizing the MTTF for the supply voltage reduction and power saving. To maximize the MTTF of the AVS circuit, the timing slack of the critical paths should be monitored frequently to adjust the supply voltage appropriately, which was mentioned in Section 1.2.2. Therefore, for efficiently monitoring such paths, this thesis proposes a novel insertion method that maximumly decreases the sum of gate-wise timing failure probabilities. Note that the timing failure probability of a FF is a joint probability of activation and timing violation probabilities of the FF, and the gate-wise failure probability is calculated from the timing failure probability of endpoint FFs. By exploiting the information on the paths with the higher timing failure probability, the proposed sensor insertion makes AVS efficiently monitor the timing-critical and highly-active FFs. Moreover, by maximizing the sum of gate-wise failure probabilities, the proposed sensor insertion can cover the larger set of instances that can contribute to causing timing errors. In this work, we find out a set of FFs that maximally reduces the sum of gate-wise failure probabilities by solving instance covering problem as an integer linear programming (ILP) problem.

As for the post-silicon validation methodology, Chapter 5 devises the error detection mechanisms for short latency (EDM-L) and evaluates the timing error localization capability of EDM-L for supply noise induced errors. This work considers two scenarios of EDM-L usage in post-silicon validation: (1) localizing a timing error that occurred in the original program and (2) localizing as many potential timing errors as possible. This work investigates how often EDM-L transformation can work in above two scenarios using a supply noise aware simulation framework and 65 nm test chip. This thesis discusses the inconsistency between simulation results and hardware measurement and find out two possible reasons: (1) the design of power distribution network, and (2) the definition of FMAX used for evaluation. By updating the simulation setup, this work confirmed that the EDM performance evaluated by the simulation was consistent with that by the chip measurement.

The rest of this thesis is organized as follows. Chapter 2 discusses supply voltage reductions achieved by AVS circuits with different sensors and demonstrates TEP-FF achieves the target MTTF with lower average V_{dd}. Chapter 3 proposes a MTTF-aware ASA for minimizing power dissipation of voltage scaled circuit under AVS. Chapter 4 proposes a MTTF-aware design methodology for the AVS circuit. The proposed design methodology optimizes both the voltage scaled circuit under the AVS and the sensor. In this chapter, we propose a novel TEP-FF insertion method that minimizes the sum of gate-wise timing failure probabilities aiming at MTTF maximization. Chapter 5 devises the EDM-L and evaluates the performance of EDM-L for timing error localization with a noise-aware logic simulator and 65 nm test chips. Lastly, concluding remarks are given in Chapter 6.

Chapter 2

Supply Voltage Reduction Aware Sensor Selection

This chapter discusses supply voltage reductions achieved by AVS circuits with different sensors [76]: in-situ sensors and replica. This work gives the MTTF as a design constraint and compares the trade-offs of AVS circuits with different sensors.

2.1 Introduction

This section presents a strategy for comparing in-situ sensors and replica. First, let us discuss the expected reduction of supply voltage thanks to AVS with in-situ TEP-FF and replica and highlights the points for discussion in this chapter.

Fig. 2.1 exemplifies the expected V_{dd} reduction effects obtained by TEP-FF based AVS and replica based AVS under an MTTF constraint. Note that X-axis is the clock period and Y-axis is the average supply voltage. Smaller clock period or average V_{dd} means that the circuit achieves the performance improvement or the V_{dd} reduction. The top black curve represents the conventional WC design that accumulates timing margins assuming the worst PVTA condition. The second blue curve shows the trade-off curve of replica based AVS, and the elimination of timing margins for global variation is expected to reduce supply voltage. The third red curve corresponds to AVS with TEP-FF, and this AVS is supposed to lower supply voltage further by exploiting design margins for intra-die random variation.

A crucially important issue in investigating Fig. 2.1 is that the trade-off analysis must be conducted under the same MTTF constraint. If we accept the shorter MTTF, we can aggressively reduce the supply voltage and consequently the trade-off curve shifts. References [44, 47] compare critical path replica and in-situ slack monitor and experimentally show that replica fails to capture within-die variations such as random manufacturing variations. For example, [44] reports that in-situ slack monitor needs only 0.9% timing margins whereas replica requires 4.2% margins to ensure correct

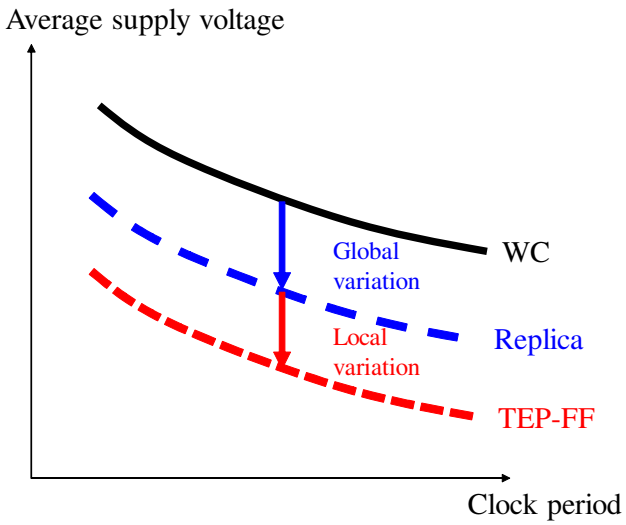

Figure 2.1: Expected voltage reduction thanks to the AVS with in-situ TEP-FF or replica under the MTTF constraint.

operation at nominal PVTA condition. However, conventional works [44, 47] do not explicitly take into account the MTTF constraint and the impact of dynamic delay variations such as supply noise and aging. As mentioned earlier, the performance of the AVS circuits having different MTTFs cannot be directly compared. Also, appropriate margining for dynamic variations are indispensable in actual designs. To derive reliable implications from the comparison, we need to prepare a setup that can fairly compare the performance in practical situations. From this standpoint of view, it is necessary for designers to take into account not only static variation but the MTTF constraint and dynamic variations.

Based on the above consideration, this chapter addresses the following question: how much voltage reduction can be achieved by TEP-FF based AVS and replica based AVS from the conventional WC design under static and dynamic variations and the MTTF constraint. To answer this question, this work utilizes a stochastic error rate estimation method [61] and evaluates the MTTF and average supply voltage taking into account static manufacturing variability and dynamic variations such as supply noise and aging. Moreover, this chapter formulates their design problems as similar design optimization problems and compares the solutions aiming at a fair comparison. More precisely, this chapter defines the same objective function and similar design constraints using identical metrics. Sections 2.2 and 2.3 describe a TEP-FF based AVS design and a replica based AVS design, respectively. The V_{dd} reduction achieved by each AVS will be experimentally demonstrated for an industrial design, a cipher circuit, and an embedded processor in Section 2.4.

2.2 Designing TEP-FF Based AVS

Fig. 2.2 shows the AVS circuit which is composed of a voltage scaled circuit, voltage control logic, and TEP-FF. TEP-FF consists of a normal FF, delay buffers and a comparator, e.g., XOR gate. When the timing margin is gradually decreasing, a timing error occurs at TEP-FF before the main FF captures a wrong value due to delay buffers, which enables us to know that the timing margin of the main FF is not large enough. A warning signal is generated to predict the timing errors. Note that TEP-FF is expected to convert timing margins for intra-die random variations to V_{dd} reduction since it shares main logic and its variation.

This work defines the design optimization problem for TEP-FF based AVS as follows.

- Objective
  - Minimize : V_{dd}
- Variables
  - $B_{\mathrm{TEP}_i}(1 \leq i \leq N_{\mathrm{FF}})$
- Constraints
  - $\mathrm{MTTF} \geq \mathrm{MTTF}_{\mathrm{min}}$
  - $N_{\mathrm{TEP}}(= \sum_{i=1}^{N_{\mathrm{FF}}} B_{\mathrm{TEP}_i}) \leq N_{\mathrm{TEP}}^{\mathrm{max}}$

The objective of this problem is to minimize V_{dd} aiming at power minimization. The variable for optimization is B_{TEP_i}. B_{TEP_i} is a binary variable, and it becomes 1 when i-th FF is replaced by TEP-FF. The primary constraint is the MTTF, and the lower bound of the MTTF ($\mathrm{MTTF}_{\mathrm{min}}$) is given as a constraint. The second constraint gives the upper bound of the number of TEP-FF ($N_{\mathrm{TEP}}^{\mathrm{max}}$), and this limits the area increase due to TEP-FF insertion. To make AVS work well, TEP-FF should monitor timing margins of paths that have a higher probability of timing error occurrence and output warning signals to prevent the error occurrence. For this purpose, this work inserts TEP-FFs to voltage-scaled circuits using the timing failure probability, which is the joint probability of the timing violation probability and the activation probability, as a metric. In other words, the inserted sensors check timing margins of critical paths more frequently, and thus it enables temporally fine voltage control and helps to avoid timing error occurrence.

2.3 Designing Replica Based AVS

Fig. 2.3 shows the AVS circuit which is composed of a voltage scaled circuit, voltage control logic, and critical path replica. Critical path replica includes replicated logic, delay buffers, and an edge detector. The edge detector checks the edge timing for every

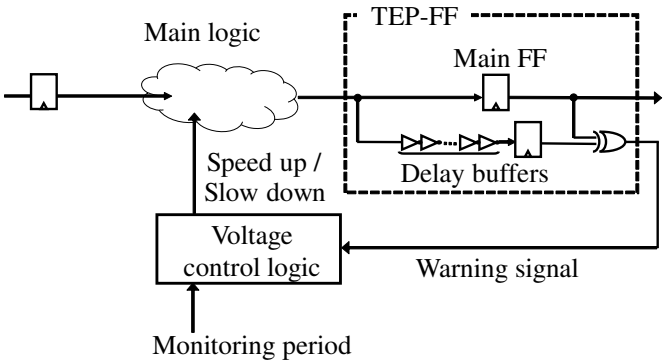

Figure 2.2: Assumed AVS with TEP-FF.

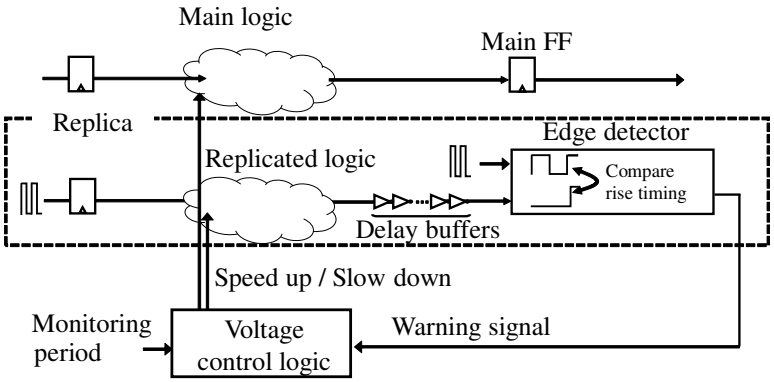

Figure 2.3: Assumed AVS with replica.

clock cycle and generates a warning signal when the edge is too late. Therefore, the timing margin can be measured much more frequently compared with TEP-FF.

To attain the same sensitivity of the replica to variations with the voltage scaled circuit, the replica should include many paths in the voltage scaled circuit. However, it requires a large area cost, and hence Kim *et al.* designed a compact replica with a comprehensive sensitivity analysis in design time [53]. On the other hand, to pursuit a discussion that is independent of replica implementation methods, this chapter assumes that the inserted replica can perfectly reproduce the delay characteristics of the paths that are selected for monitoring. With this setup, the accuracy of critical path delay measurement degrades only due to within-die variation, which is considered in our analysis. Note that Tschanz *et al.* proposed to integrate a tunable replica and tune it after fabrication [31]. Similarly, TEP-FF can be tuned if it is designed with tunable buffer. However, post-fabrication tuning during chip test is expensive for most of the products, and hence it is not considered in this dissertation.

Similarly to Section 2.2, this work formulates the design optimization of AVS with

replica as follows.

- Objective
  - Minimize : V_{dd}
- Variables
  - $B_{\mathrm{replica}_j}(1 \leq j \leq N_{\mathrm{path}})$
- Constraints
  - $\mathrm{MTTF} \geq \mathrm{MTTF}_{\mathrm{min}}$
  - $N_{\mathrm{replica}}(= \sum_{j=1}^{N_{\mathrm{path}}} B_{\mathrm{replica}_j}) \leq N_{\mathrm{replica}}^{\mathrm{max}}$

The objective of this problem is identical with that of Section 2.2. The variable for optimization is B_{replica_j}. B_{replica_j} is a binary variable, and it becomes 1 when the j-th path is replicated. As the primary constraint, the lower bound of the MTTF ($\mathrm{MTTF}_{\mathrm{min}}$) is given, which is the same as the first constraint in Section 2.2. The second gives the upper bound of the number of replicated paths ($N_{\mathrm{replica}}^{\mathrm{max}}$), and this limits the area increase due to replica implementation. Note that the constraint of area overhead by AVS with TEP-FF and AVS with replica will be set identically to keep fairness. Similarly to Section 2.2, this chapter focuses on the timing failure probability and insert replica for sensing paths whose timing failure probabilities are high.

2.4 Evaluation

This section experimentally evaluates supply voltage reduction of AVS with in-situ TEP-FF and replica from the conventional WC design. First, Section 2.4.1 explains the evaluation setup, and Section 2.4.2 demonstrates the average supply voltage of each AVS. Then, Section 2.4.3 discusses the performance difference between AVS circuits with TEP-FF and replica.

2.4.1 Evaluation Setup

This work used an industrial image signal processor (ISP), an advanced encryption standard (AES) circuit, and an OR1200 OpenRISC processor, which is a 32-bit RISC microprocessor with five pipeline stages, as target circuits. ISP was designed by a commercial place and route (P&R) tool with a 28 nm industrial standard cell library and AES and OpenRISC were laid out with a 45 nm Nangate standard cell library [77]. Also, standard cell memories [78–80] were used as SRAMs in OpenRISC processor. The post-layout circuits include 3,133,640 combinational logic cells, 16,870 latches, and 374,880 FFs in ISP, 1,276,989 combinational logic cells, 589,890 latches, and 2,504 FFs in OpenRISC, and 17,948 combinational logic cells and 530 FFs in AES, respectively.

This work prepared eight supply voltages from 0.90 V to 0.76 V with a 20 mV interval in ISP and six supply voltages from 1.20 V to 0.95 V with a 50 mV interval in AES and OpenRISC. The MTTF of 1.00×10^{17} cycles, i.e., 10.5 years in ISP, 1.6 years in AES, and 13.7 years in OpenRISC, is set as $\mathrm{MTTF}_{\mathrm{min}}$. Note that the above $\mathrm{MTTF}_{\mathrm{min}}$ is just an example, and other constraints of $\mathrm{MTTF}_{\mathrm{min}}$ can be given similarly. With this setup, several TEP-FF or replica circuits were inserted to the voltage-scaled circuits. The constraints of area overhead by TEP-FF or replica circuits are set to 0.1% for ISP and OpenRISC and 1.0% for AES, respectively. In other words, the upper bound of the number of TEP-FF and replica paths, i.e., $N_{\mathrm{TEP}}^{\mathrm{max}}$ and $N_{\mathrm{replica}}^{\mathrm{max}}$ are 483 and 69 in ISP, 30 and 9 in AES, 50 and 11 in OpenRISC, respectively. This work inserted the delay buffers whose delay were comparable to the delay variation caused by 20 mV supply noise in ISP and 50 mV one in AES and OpenRISC, where these numbers of 20 mV and 50 mV correspond to one level decrement of the supply voltage. Note that, in our evaluation, TEP-FF and replica circuits are ideally inserted to voltage-scaled circuits for simplicity. In other words, this work calculated the MTTF from delay characteristics of laid out voltage scaled circuits and the nominal delay and variation of logic cells in TEP-FF and replica. Therefore, the area overhead by replica is denoted as the sum of the cell area of the target monitoring path and delay buffers. Similarly, the area overhead by TEP-FF is denoted as the sum of the cell area of duplicated FF, comparator, and delay buffers. One of the future works includes to take into account the control circuit and its wirings to TEP-FF and replica.

The MTTF and average supply voltage under PVTA variation were evaluated by a stochastic MTTF estimation framework proposed in [61]. To estimate the MTTF with [61], this work evaluated the timing failure probability, which is the joint probability of the timing violation probability and the activation probability. This work derived the activation probability of each path by associating logic simulation [81] and static timing analysis (STA) [82]. As for workload, we selected one for ISP aiming to maximize power consumption. In OpenRISC, three benchmark programs (crc, sha, and dijkstra) were chosen from MIBenchmark [83]. For each program, 30 sets of input data were prepared for MTTF estimation. Totally, this evaluation used 90 ($= 3 \times 30$) workloads. In AES, 1,000 random test patterns were used.

In addition, the timing violation probability was derived as followings. First, probability density functions of gate delay variability were generated according to the assumed variations. Then, sensitivity-based SSTA (such as [84] and [85]) were executed with common path pessimism removal (e.g., [86] and [87]) to obtain the canonical-form expression of the timing violation probability. Lastly, the timing violation probability was calculated by integrating the canonical-form expression with MATLAB 2016b [88]. Here, to calculate the meaningful MTTF, practical delay variations should be considered. This evaluation took into account the following variations.

- Dynamic supply noise, which is assumed to temporally fluctuate between -90 mV and 70 mV in ISP and between -50 mV and 50 mV in AES and OpenRISC.

- Manufacturing variability, which is assumed to consist of intra-die random variation and inter-die variation. In ISP, the inter-die variation is extracted from the difference of delay characteristics between TT, i.e., typical-typical, library and SS, i.e., slow-slow, global library and the intra-die variation is calculated with on-chip variation coefficient defined in the 28 nm standard cell library. In AES and OpenRISC, both the intra-die random variation and inter-die variation include NMOS and PMOS threshold voltage variation of σ = 30 mV and gate length variation of σ = 1 nm, respectively.

- NBTI aging, whose model was obtained by fitting a trapping/de-trapping model [89] to the measured data in [90]. Note that, in ISP, this NBTI model is not used since the on-chip variation coefficient in the 28 nm standard cell library already includes aging-induced delay variation. In AES and OpenRISC, six degradation states of 0 mV, 0.5 mV, 1 mV, 5 mV, 10 mV and 15 mV are prepared. Note that [90] measures the NBTI degradation with a stress probability of 100%, and thus the NBTI model used in the experiment does not consider recovery situation. Our future work includes to investigate the adequacy of the degradation status assignment and consider the relationship between the degradation and the activation probability.

- Temperature gradation, which is assumed to temporally fluctuate between -10°C and 110°C in ISP. Note that this temperature gradation is not taken into account in AES and OpenRISC.

In MTTF evaluation, the clock period was swept from 2,500 ps to 3,300 ps in ISP, 450 ps to 550 ps in AES, and from 4,000 ps to 5,500 ps in OpenRISC. For each clock period, AVS dynamically adjusts the supply voltage. In the experiment, the monitor period for AVS was varied from 10^6 cycles to 10^{15} cycles. Here, the monitor period of 10^6 cycles means, if no error prediction signals are outputted for 10^6 cycles, the supply voltage is decreased. The minimum monitor period, i.e., 10^6 cycles, is about 3.3 ms in ISP, 0.5 ms in AES, and 4.3 ms in OpenRISC, respectively, and it is longer than the response time of the fast transient voltage regulator, e.g., 1.6 μs in [91].

2.4.2 V_{dd} Reduction by TEP-FF and Replica

Fig. 2.4 shows trade-offs between the minimum average supply voltage and the clock period under the MTTF constraint of 10^{17} cycles, where (a) in ISP, (b) in AES, and (c) in OpenRISC, respectively. The black cross plots represent the conventional WC design with guard-banding for PVTA variation. The blue circular plots and red square plots correspond to AVS circuits with replica and with TEP-FF, respectively. This section examines the evaluation results from the V_{dd} reduction effect thanks to AVS with TEP-FF and replica.

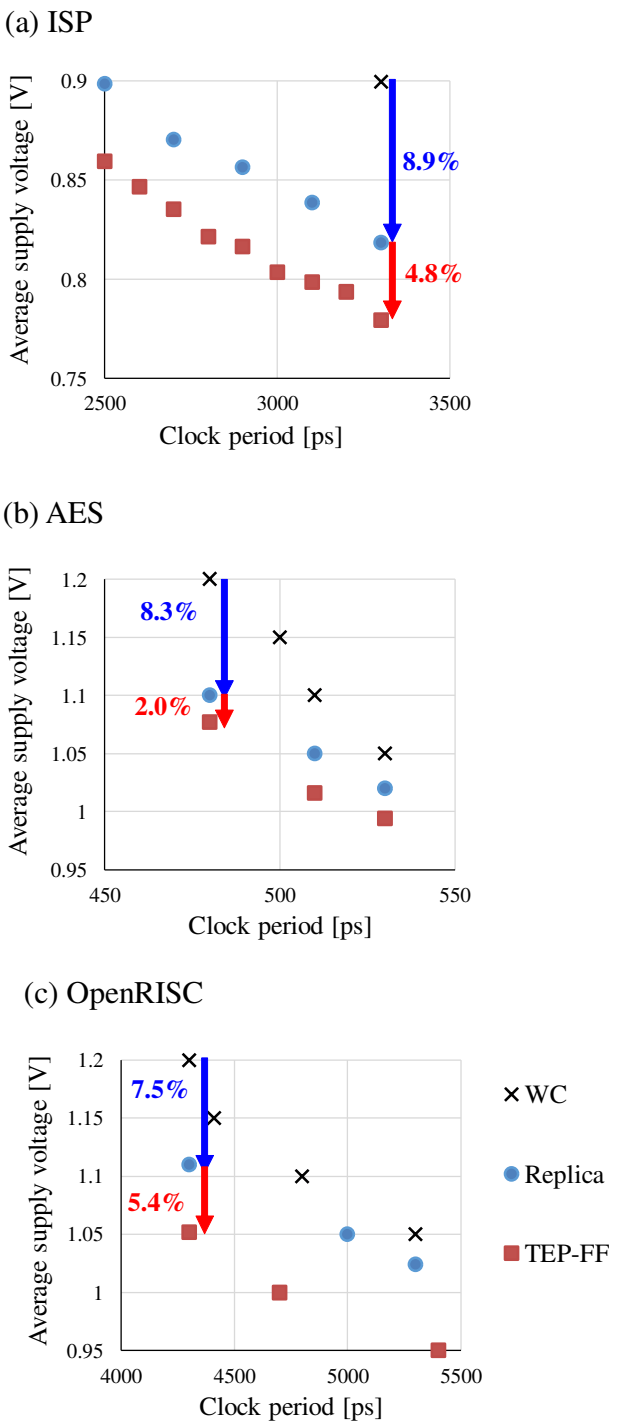

Figure 2.4: Trade-off comparison between the conventional WC design, AVS with TEP-FF, and AVS with replica. (a) ISP, (b) AES, (c) OpenRISC.

First, the black and blue/red plots are compared to clarify the performance improvement thanks to AVS with replica/TEP-FF. Fig. 2.4 shows that both replica based AVS and TEP-FF based AVS reduce average supply voltage from the conventional WC design while keeping the target MTTF. For example, in Fig. 2.4(a), at a clock period of 3,300 ps, AVS with replica achieved the target MTTF at an average supply voltage of 0.82 V, whereas the conventional WC design required 0.90 V operation. In other words, replica based AVS achieved 8.9% V_{dd} reduction from 0.90 V to 0.82 V. Similarly, in Fig. 2.4(b) and Fig. 2.4(c), AVS with replica achieved 8.3% V_{dd} reduction from 1.20 V to 1.10 V at clock period of 480 ps and 7.5% V_{dd} reduction from 1.20 V to 1.11 V at clock period of 4,300 ps, respectively. As for AVS with TEP-FF, it achieved 13.3% V_{dd} reduction from 0.90 V to 0.78 V in ISP (Fig. 2.4(a)), 10.0% V_{dd} reduction from 1.20 V to 1.08 V in AES (Fig. 2.4(b)), and 12.5% V_{dd} reduction from 1.20 V to 1.05 V in OpenRISC (Fig. 2.4(c)), respectively. This work experimentally confirmed that AVS with replica and TEP-FF made the significant voltage margin reduction both in ISP, AES, and OpenRISC at the cost of 0.1% area increase in ISP and OpenRISC and 1.0% in AES.

Next, AVS with replica and AVS with TEP-FF are compared. Fig. 2.4 shows that AVS with TEP-FF further reduces the average supply voltage from AVS with replica. For example, AVS with TEP-FF achieved 4.8% V_{dd} reduction from 0.82 V to 0.78 V at the clock period of 3,300 ps in ISP, 2.0% V_{dd} reduction from 1.10 V to 1.08 V at 480 ps in AES, and 5.4% V_{dd} reduction from 1.11 V to 1.05 V at 4,300 ps in OpenRISC. This voltage reduction reveals that TEP-FF helps to exploit more timing margin than replica. TEP-FF converts the timing margin of intra-die random variation to V_{dd} reduction whereas replica needs to keep this margin, which will be discussed in Section 2.4.3. From the above, this section experimentally confirmed that TEP-FF achieved the larger supply voltage reductions in field operation and satisfied the MTTF constraint.

2.4.3 Discussion

This section discusses the difference between AVS with replica and AVS with TEP-FF investigating the impact of intra-die random variation on the MTTF. This section evaluates the MTTF in a case that an identical set of paths are monitored by TEP-FF and replica. The difference of the MTTF in this experiment is supposed to originate from how much the intra-die random variation can be considered by each sensor. As mentioned earlier, TEP-FF shares the intra-die variation with the main logic and hence it exploits the timing margin for the intra-die variation.

Fig. 2.5 shows the MTTF comparison between TEP-FF based AVS and replica based AVS in ISP. Note that in the MTTF calculation, there were cases where no timing errors occurred, i.e., the MTTF is ∞. In the figure, the infinity MTTF is plotted as 10^{20} cycles to include it in the figure. The number of inserted TEP-FFs is set to 483, and the inserted TEP-FFs sense 15,285 activated paths. Hence, the number of replicated paths is set to

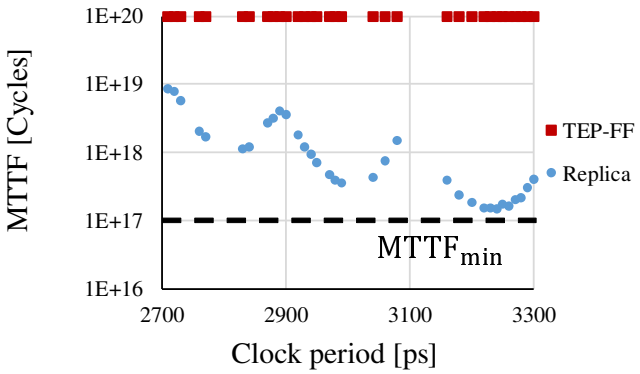

Figure 2.5: MTTF comparison between TEP-FF and replica (ISP).

15,285. Fig. 2.5 shows that the MTTF of TEP-FF based AVS is longer than that of replica based AVS even though the identical set of paths are monitored. Thus, this section experimentally confirmed that TEP-FF more exploited the timing margin for the intra-die random variation to MTTF extension resulting in larger voltage reduction observed in the previous section.

2.5 Conclusion

This chapter focused on timing sensors necessary for AVS implementation and compared in-situ TEP-FF and critical path replica in terms of the supply voltage reduction. This work gave the MTTF as a design constraint and compared the trade-offs of AVS circuits with these two sensors. Experimental results showed that TEP-FF based AVS and replica based AVS achieved up to 13.3% and 8.9% supply voltage reduction, respectively, while satisfying the target MTTF.

Chapter 3

Main Logic Optimization for Time-to-Failure Extension and Power Saving

This chapter proposes the MTTF-aware ASA for minimizing power dissipation of main logic under AVS [92]. ASA gives timing slacks to critical paths and reduces the number of active critical paths, which reduces the supply voltage and thus power dissipation. The proposed optimization includes both a pre-ASA circuit design and ASA implementation. The former pre-ASA design prepares several design candidates laid out with different timing constraints and selects the most promising candidate regarding power. For this selection, every candidate is analyzed to estimate the minimum supply voltage after ASA that can achieve the target MTTF. Then, the proposed methodology selects a set of FFs for ASA using integer linear programming such that it reduces the sum of gate-wise failure probabilities maximumly, and performs P&R ECO.

3.1 Introduction

In the conventional VLSI design flow, the circuit delay is adjusted to given target speed, e.g., clock period, with saving the power and area. For example, if there are several non-critical paths whose delay is much smaller than those of critical paths, these paths will be downscaled or replaced with high-threshold-voltage (high-V_{th} gates and thus critical paths, which are called non-intrinsic critical paths hereafter. Consequently, the conventional design optimization dramatically increases the number of non-intrinsic critical paths and therefore it becomes difficult to monitor such a large number of critical paths by sensors of AVS.

This chapter proposes a design methodology for MTTF-aware ASA design. The proposed methodology explicitly introduces the MTTF as a design constraint and optimizes the design with ASA. ASA gives timing slacks to non-intrinsic active critical

paths by ECO. Thus, ASA reduces the number of active critical paths whose delays are very close to those of the intrinsic critical paths. Therefore, it can be expected that circuits with ASA have fewer paths where timing errors are likely to occur, which can reduce the number of monitoring paths with AVS and thus can contribute to facilitating the following sensor insertion. Also, the reduction of active critical paths helps to extend the TTF and thus reduce the supply voltage since timing failure probabilities of such paths can be dramatically decreased. This chapter assumes that the supply voltage can be set for each chip individually for power minimization. This situation is called chip-wise voltage assignment hereafter.

Here, the design methodology needs to prepare a pre-ASA circuit, choose paths to which ASA is applied, and determine timing slacks for each path. To save power maximumly by ASA, this chapter firstly proposes a selection method of the pre-ASA circuit from several design candidates. Note that, with chip-wise voltage assignment, the optimal design is not obvious since voltage scaling varies power and speed and the impact of ASA depends on the pre-ASA circuit. For each candidate, the proposed method estimates minimum supply voltage after ASA (V_{min}) at which the circuit can achieve the target MTTF, and evaluates the power dissipation of circuit at V_{min}. Thus, the circuit whose estimated power is minimum can be chosen. Secondly, this chapter proposes FF based ASA that assigns timing slack to each FF. We develop a FF selection method using integer linear programming (ILP) that maximizes the sum of gate-wise failure probabilities aiming to improve the MTTF maximumly. Thirdly, for each target FF, this chapter extracts necessary timing slacks to sustain the target MTTF at V_{min} and give these timing slacks as constraints to P&R ECO.

Fig. 3.1 illustrates the expected power savings. The top black curve represents the conventional WC design that adds timing margins assuming the worst PVTA condition. The middle yellow curve is also WC design, but it optimizes operating conditions such as supply voltage and clock period so that the design satisfies the target MTTF. The bottom blue curve corresponds to the proposed ASA with MTTF-aware operation. The proposed ASA is expected to attain a better trade-off between power dissipation and clock period. This chapter will experimentally demonstrate these power saving effects in an embedded processor and a cipher circuit.

The rest of this chapter is organized as follows. Section 3.2 introduces the concept of ASA and formulates the problem of ASA circuit design including the pre-ASA circuit design. Section 3.3 describes the overview of the proposed design methodology which is composed of pre-ASA circuit design and ASA implementation. Section 3.4 introduces the selection method of pre-ASA candidates and identifies the most promising one in terms of power after ASA. Section 3.5 applies ASA to the selected pre-ASA circuit. Section 3.6 experimentally evaluates the performance improvement thanks to MTTF-aware design regarding power. Lastly, concluding remarks are given in Section 3.7.

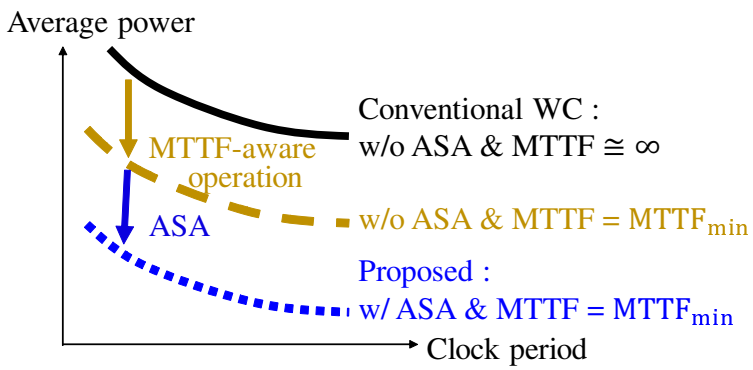

Figure 3.1: The proposed MTTF-aware design reduces power dissipation thanks to MTTF-aware operation and design optimization with ASA.

3.2 Activation-aware Slack Assignment and Problem Formulation

This section, first, explains the concept of ASA. Next, this section formulates the ASA that aims to save power while keeping the MTTF as an optimization problem.

3.2.1 ASA

Before introducing ASA, let us first explain the conventional design. The left side of Fig. 3.2(a) illustrates the path delay distribution of a conventionally designed circuit, and the right side shows the pair of the activation probability and the timing violation probability of non-intrinsic critical paths. In the conventional circuit design flow, cell instances included in non-critical paths are replaced with smaller cells and high-V_{th} cells to reduce power dissipation and area. Therefore, the number of paths whose delays are close to the critical path delay increases. This replacement decreases timing margin of the paths that go through the replaced instances and may increase the timing error occurrence probability under variations. In other words, more instances are prone to cause path delay variations.

ASA increases timing slacks of highly-activated non-intrinsic critical paths. The left side of Fig. 3.2(b) exemplifies the path delay distribution of the ASA circuit. As ASA enforces larger slacks on highly activated paths, highly-activated paths sustain timing margin even when gate delay varies. Accordingly, as shown in the right side of Fig. 3.2(b), timing violation probabilities in these paths are dramatically reduced compared to the conventional circuit, which is the main advantage of the ASA. These reductions extend the MTTF and consequently save power as mentioned in Section 3.1. Here, it should be noted that ASA partially loses the power and area reduction acquired by the conventional design optimization. From this sense, we need to find a better trade-

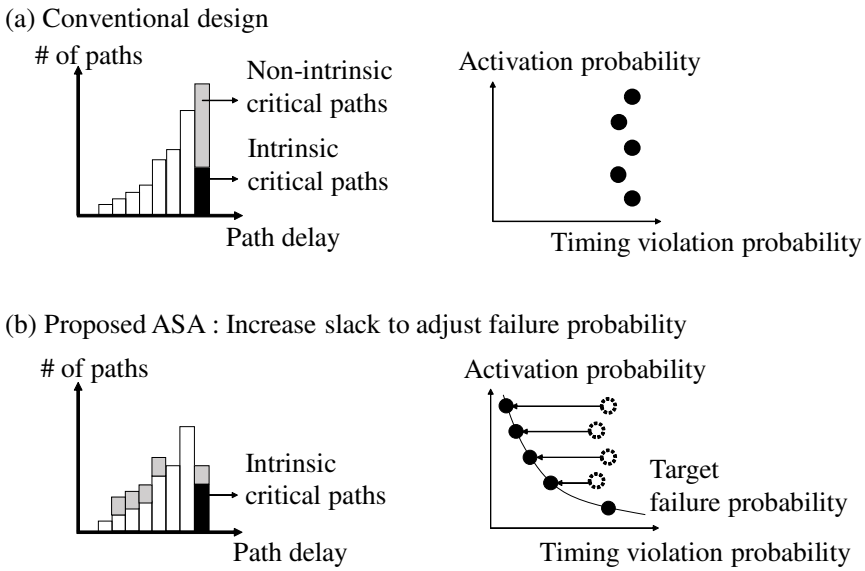

Figure 3.2: Path delay distributions (left side), and the activation probability and the timing violation probability of non-intrinsic critical paths (right side) of circuits. (a) the conventional design without ASA, (b) proposed ASA.

off relation between the timing error occurrence probability and power. To pursuit the better trade-off, the proposed ASA adjusts the failure probability of the path to the target failure probability as shown in the right side of Fig. 3.2(b). In other words, the amount of slack increase is assigned to reduce power and area overheads while satisfying the target MTTF. Thanks to this assignment, the proposed ASA can save the overhead while extending the MTTF and saving power. Remind that the failure probability is defined as the product of the activation probability and the timing violation probability of a path, and the target failure probability can be calculated from the target MTTF, where the detail is given in Section 3.4.2. The proposed design methodology of ASA will be explained in Section 3.5.

3.2.2 Problem Formulation

The concept of ASA was explained above using the path delay distribution depicted in Fig. 3.2. However, the path-based design optimization for ASA circuits is not efficient since the number of paths in a circuit is huge. Instead, this chapter chooses FF-based design optimization for ASA circuits. Fig. 3.3 exemplifies two-step FF-based ASA: (1) increase setup time of the target i-th FF by Δsetup_i artificially and re-layout the design as an ECO process, and (2) restore the original setup time for the successive analysis process. It should be noted that modifying the setup time is just one implementation and there are other ways to perform FF-based ASA. For example, timing derate factors

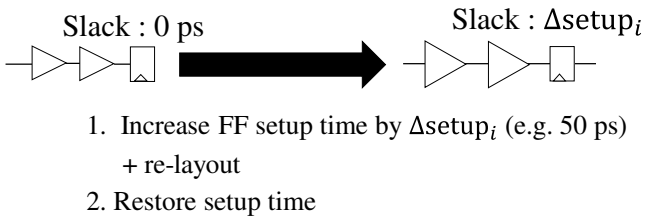

Figure 3.3: An example of FF-based ASA.

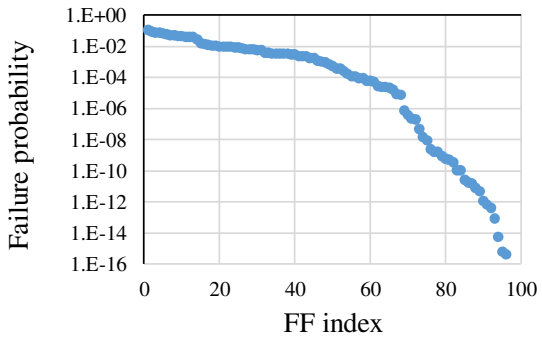

Figure 3.4: Failure probabilities of FFs are largely different.

for each FF may be manipulated expecting the same result of the FF-based slack assignment. With this FF-based ASA, the paths ending at the target FF are enforced to have the slack that is larger than Δsetup_i.

Note that if there are intrinsic critical paths whose path delays cannot be shortened, such paths cannot have the slack of Δsetup_i. After the ASA, the circuit area increases since conventional designs exploit such slacks for area reduction. ASA circuits have more timing margin but involve the larger area.

An important observation in this work is that all the FFs do not have the same contribution to the MTTF. Fig. 3.4 shows timing failure probabilities of FFs in an OR1200 OpenRISC processor. Fig. 3.4 shows that several FFs have high failure probabilities, which dominantly determine the MTTF. This result motivates us to smartly select a small number of target FFs that impact the MTTF. In this case, the area overhead of ASA can be mitigated.

Based on the discussion above, the problem of ASA circuit design is formulated as follows.

- Input
  - N_{CKT} pre-ASA candidates

- Output
  - one ASA circuit
- Objective
  - Minimize : Power = $\min(\text{Power}_1, \cdots, \text{Power}_{N_{\text{CKT}}})$
- Constraints
  - $\text{MTTF}_j \geq \text{MTTF}_{\text{min}} (1 \leq j \leq N_{\text{CKT}})$
  - $\text{Area}_j \leq \text{Area}_{\text{max}} (1 \leq j \leq N_{\text{CKT}})$
  - $N_{\text{LVth}_j} \leq N_{\text{LVth}}^{\text{max}} (1 \leq j \leq N_{\text{CKT}})$
- Variables
  - $\Delta\text{setup}_{i,j} \quad (1 \leq i \leq N_{\text{FF}}, 1 \leq j \leq N_{\text{CKT}})$

The inputs of this problem are N_{CKT} pre-ASA candidates, and the output is one ASA circuit. The objective of this problem is to minimize the power of the ASA circuit. The ASA circuit is constrained by the MTTF (MTTF_{min}), circuit area (Area_{max}), and the number of low-V_{th} cells ($N_{\text{Lvth}}^{\text{max}}$). Chip-wise voltage assignment adjusts the supply voltage to minimize power dissipation while satisfying the target MTTF_{min}. The variables $\Delta\text{setup}_{i,j}$ are the slacks given to FFs in j-th pre-ASA circuit, where $\Delta\text{setup}_{i,j}$ is given to the layout ECO as an intentional increase in setup time of i-th FF_i in j-th pre-ASA circuit. N_{FF} is the number of FFs in the circuit, and it is identical in all the pre-ASA circuits. When $\Delta\text{setup}_{i,j} = 0$, i-th FF_i is not included in the set of target FFs of j-th pre-ASA circuit. Thus, the number of target FFs, i.e., N_{ASA}, is expressed as the number of FFs whose $\Delta\text{setup}_{i,j}$ is larger than 0. Here, MTTF_j depends on $\Delta\text{setup}_{i,j}$ and supply voltage, and these relations are evaluated by the stochastic error rate estimation method [61]. Area_j and N_{LVth_j} depend on $\Delta\text{setup}_{i,j}$, and it is given by the layout tool after P&R ECO.

3.3 Overview of Proposed ASA

A difficulty to solve the formulated problem is the non-linear relations among MTTF_j, Area_j, N_{LVth_j}, and $\Delta\text{setup}_{i,j}$. Also, the evaluations of MTTF_j, Area_j, and N_{LVth_j} need relatively long CPU time, and hence an explicit optimization is not efficient concerning CPU time. Thus, to determine the set of $\Delta\text{setup}_{i,j}$ efficiently, this work proposes a two-step procedure shown in Fig. 3.5.

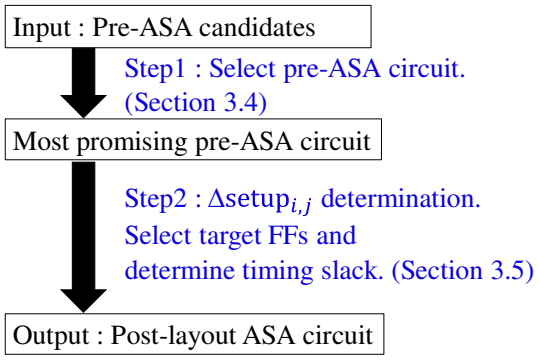

Figure 3.5: The proposed design methodology with two-step procedure: (1) select most promising pre-ASA circuit in terms of power, (2) perform ASA to selected one.

Fig. 3.5 shows the overview of the proposed design methodology which includes both pre-ASA circuit design and ASA implementation. The first procedure screens the pre-ASA candidates using the trade-off analysis between the MTTF and power and identifies the most promising candidate that is expected to achieve the lowest power operation after ASA, and this candidate is given to the second step. After this candidate selection, the circuit parameter of j is fixed, and the following second step of ASA implementation will determine $\Delta\text{setup}_{i,j}$. With this screening process, the determination of the j-th circuit and i-th FF can be decoupled. The detail of the screening procedure will be explained in Section 3.4.

After pre-ASA circuit selection, the proposed methodology implements ASA to the selected circuit and determines $\Delta\text{setup}_{i,j}$. For various N_{ASA}, i.e., the number of FFs that ASA is applied to, the proposed methodology decides the set of target FFs and their $\Delta\text{setup}_{i,j}$ aiming at MTTF maximization. Here, we are expecting that a circuit with the longer MTTF has a larger room for power saving and N_{ASA} is related to the increase in area and the number of low-V_{th} cells. Then, for each set of $\Delta\text{setup}_{i,j}$, P&R ECO is performed to obtain Area_j and N_{LVth_j} and evaluate the trade-off relation between the supply voltage and MTTF_j using the stochastic error rate estimation method. The evaluation results identify the set of $\Delta\text{setup}_{i,j}$ that minimizes power while satisfying the constraints of MTTF_j, Area_j, and N_{LVth_j}.

Taking this approach, for each N_{ASA}, we need to select N_{ASA} FFs and determine $\Delta\text{setup}_{i,j}$ of the selected FFs. Section 3.5.2 explains how to select N_{ASA} target FFs, and Section 3.5.3 presents how to determine $\Delta\text{setup}_{i,j}$.

3.4 Design of Pre-ASA Circuit

The important consideration in this work is how to design the pre-ASA circuit to obtain the better ASA circuit. One possible approach is to prepare a pre-ASA circuit that

is designed at the maximum operating frequency (FMAX) and performs ASA. This pre-ASA circuit tends to include low-V_{th} cells and large-area cells and consequently increases dynamic and static power. Here, the circuit designed at looser frequency may be flexible for an additional design change in ECO compared to the FMAX design, and hence ASA may provide better optimization results. Please remind that chip-wise voltage assignment compensates the frequency difference in design time with voltage scaling after fabrication. Therefore, it is not apparent which pre-ASA circuits achieve the minimum power operation after ASA and post-fabrication voltage assignment while satisfying the target MTTF at a given clock period.

This section proposes a method to select the pre-ASA circuit that is expected to be the most power-efficient from candidates. This supposes that pre-ASA candidates are synthesized and laid out with various clock periods since the clock period has the largest impact on the trade-off between speed and area/power. The other constraints, such as maximum transition time, area, and power are kept unchanged for simplicity. The proposed selection method firstly estimates the minimum supply voltage after ASA (V_{min}) at which the circuit can achieve the given target MTTF for each candidate. Then, the proposed method evaluates the circuit power with each V_{min} and selects the circuit whose power is minimum among the candidates. The power evaluation and comparison can be performed using EDA tools with relatively short CPU times. The explicit computation to find V_{min} cannot be conducted regarding CPU time since the solution space of ASA is huge. When the pre-ASA circuit has N_{FF} FFs, the total combinational number of FF selection for ASA, N_{comb}, is $\sum_{N_{\mathrm{ASA}}=1}^{N_{\mathrm{FF}}} C(N_{\mathrm{FF}}, N_{\mathrm{ASA}})$. In case of $N_{\mathrm{FF}} = 1,000$ and $N_{\mathrm{ASA}} = 100$, for example, N_{comb} reaches $7.18{\times}10^{139}$.

To tackle this issue, this work focuses on the MTTF-dominant FF, which is expected to cause the timing error at the highest supply voltage in the circuit. If the MTTF-dominant FF can be found efficiently, the CPU time of V_{min} estimation can be dramatically reduced. In this case, the iteration times of ECO for finding the MTTF-dominant FF is limited to the number of FFs (N_{FF}), and this is much smaller than N_{comb}. From the above, to estimate V_{min} of each candidate circuit efficiently, the proposed method executes the following two steps for each candidate: (1) finding the most MTTF-dominant FF after ASA, and (2) calculating V_{min}.

3.4.1 Finding the MTTF-Dominant FF

To find the most MTTF-dominant FF after ASA, the proposed method increases the timing slack for each FF as much as possible and calculates the failure probability of the FF of interest, where the failure probability is the joint probability of the timing violation probability and the activation probability. This work calculates the timing violation probability by performing SSTA and derives the activation probability of each path by associating the signal transition time in logic simulation and the path delay in STA as shown in Fig. 3.6.

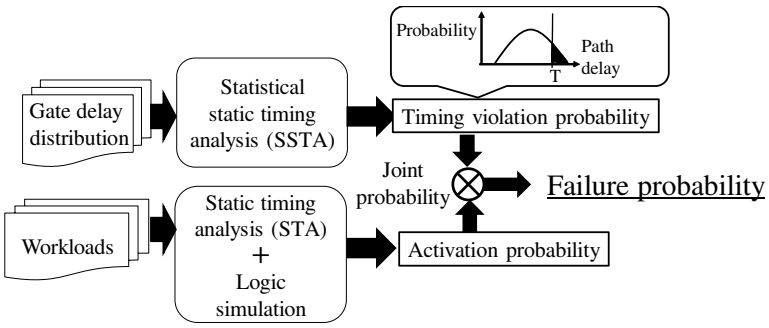

Figure 3.6: Failure probability calculation.

This work regards the FF with the highest failure probability as the MTTF-dominant FF. Here, the most MTTF-dominant FF varies depending on the supply voltage. Therefore, failure probabilities at various supply voltages are evaluated and utilized for V_{min} estimation in the next subsection.

3.4.2 Calculating V_{min}

Next, the proposed method estimates V_{min} by comparing the failure probability with the target one at each supply voltage. This work calculates the target failure probability from the target MTTF with the worst-case assumption in which all the FF have the identical highest failure probability, and timing error occurrences at all FFs are uncorrelated.

$$P^{\mathrm{max}}_{\mathrm{fail\_FF}} = \frac{1}{N_{\mathrm{FF}} \times \mathrm{MTTF}_{\mathrm{min}}}. \tag{3.1}$$

In Eq. (3.1), $P^{\mathrm{max}}_{\mathrm{fail\_FF}}$ is the upper bound of the failure probability of FFs, which is the target failure probability. N_{FF} is the number of FFs, and $\mathrm{MTTF}_{\mathrm{min}}$ is the lower bound of the MTTF, i.e., the target MTTF. In other words, Eq. (3.1) calculates the target failure probability to meet $\mathrm{MTTF}_{\mathrm{min}}$ even when all the N_{FF} FFs have the target failure probability.

Eq. (3.1) is derived from the equation below with Maclaurin expansion.

$$\mathrm{MTTF}_{\mathrm{min}} = \frac{1}{1 - (1 - P^{\mathrm{max}}_{\mathrm{fail\_FF}})^{N_{\mathrm{FF}}}}. \tag{3.2}$$

Here, $1 - P^{\mathrm{max}}_{\mathrm{fail\_FF}}$ represents the lower bound of the probability that no errors occur at a FF during a unit time, where the unit time is defined as a clock cycle in this work. Thus, $(1 - P^{\mathrm{max}}_{\mathrm{fail\_FF}})^{N_{\mathrm{FF}}}$ is the lower bound of the probability that no errors occur in all the FFs during a clock cycle. Therefore, $1 - (1 - P^{\mathrm{max}}_{\mathrm{fail\_FF}})^{N_{\mathrm{FF}}}$ is the upper bound of the probability that timing error occurs during a clock cycle. Note that Eq. (3.2) simply multiplies failure probabilities of FFs, which means that timing error occurrences at different FFs are treated as uncorrelated events.

Next, Eq. (3.2) is derived. Here, the MTTF-dominant FF has the highest failure probability among N_{FF} FFs. Then, if the failure probability of the MTTF-dominant FF is smaller than $P^{\max}_{\mathrm{fail\_FF}}$, the probability that no timing error occurs in the circuit during a clock cycle, i.e. $\prod_{i=1}^{N_{\mathrm{FF}}}(1 - P_{\mathrm{fail\_FF}_i})$ is larger than $(1 - P^{\max}_{\mathrm{fail\_FF}})^{N_{\mathrm{FF}}}$ since $P^{\max}_{\mathrm{fail\_FF}}$ is larger than $P_{\mathrm{fail\_FF}_i}$ for each i ($1 \leq i \leq N_{\mathrm{FF}}$). Note that $P_{\mathrm{fail\_FF}_i}$ denotes the failure probability of the i-th FF. In this case,

$$\mathrm{MTTF}_{\min} = \frac{1}{1 - \prod_{i=1}^{N_{\mathrm{FF}}}(1 - P_{\mathrm{fail\_FF}_i})} > \frac{1}{1 - (1 - P^{\max}_{\mathrm{fail\_FF}})^{N_{\mathrm{FF}}}}. \tag{3.3}$$

From the above, when the failure probability of the MTTF-dominant FF is smaller than $P^{\max}_{\mathrm{fail\_FF}}$, the circuit satisfies the target MTTF of $\mathrm{MTTF}_{\min}$. Therefore, the proposed method finds $V_{\min}$ where the failure probability of the MTTF-dominant FF is smaller than $P^{\max}_{\mathrm{fail\_FF}}$ obtained from Eq. (3.1).

3.5 ASA Implementation

This section proposes a design method that applies ASA to the pre-ASA circuit designed in Section 3.4.

3.5.1 Design Flow

Fig. 3.7 shows the proposed design flow of the ASA circuit. First, the proposed design method selects target FFs for ASA and determines $\Delta\mathrm{setup}_{i,j}$ for these FFs. Section 3.5.2 explains how to select N_{ASA} target FFs, and Section 3.5.3 presents how to determine $\Delta\mathrm{setup}_{i,j}$. Then, the proposed method modifies the circuit to actualize $\Delta\mathrm{setup}_{i,j}$ through ECO processes.

Firstly, let us explain why this work focuses on the layout ECO-based ASA, not logic synthesis-based ASA. The synthesis-based ASA has more flexibility in changing the circuit structure compared with ECO-based ASA. However, there are substantial differences in path delays before and after layout designs since the interconnects have a large impact on timing. When ASA is applied in logic synthesis, the intention of ASA may disappear in P&R process. Consequently, this work manipulates timing slacks in ECO phase to make sure that the intention of ASA is reflected in the final layout.

Then, let us explain why the proposed ASA gives $\Delta\mathrm{setup}_{i,j}$ to three ECO processes from the placement stage to the routing stage. When the ASA enforces the target FFs to increase timing slack, some other FFs and combinational cells need re-placement and re-routing. Also, clock tree re-synthesis might be necessary. Fig. 3.8 shows such a tendency. For each FF, the largest data arrival time at path endpoints are extracted and compared between the following two ASA conditions: (1) perform ASA only with routing ECO and its optimization, and (2) perform ASA with placement, clock tree

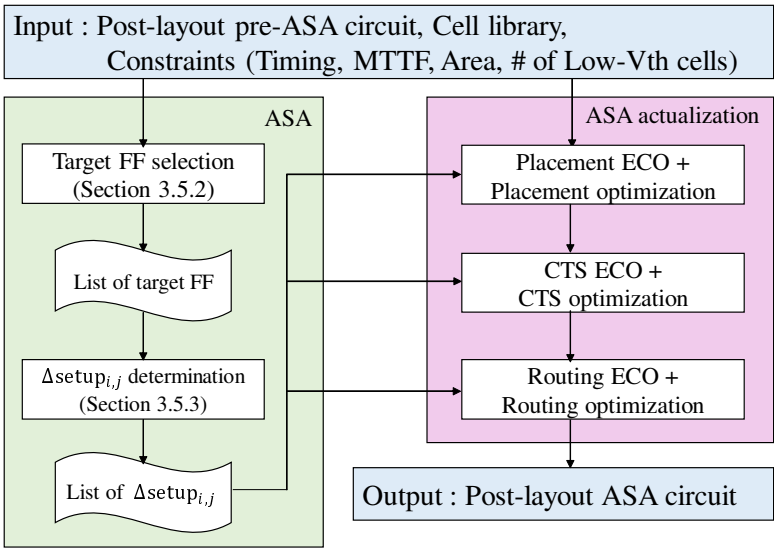

Figure 3.7: The proposed ASA design flow. The proposed ASA first selects target FFs (Section 3.5.2) and determines $\Delta setup_{i,j}$ for target FFs (Section 3.5.3). Then, proposed ASA performs placement, CTS and routing ECO.

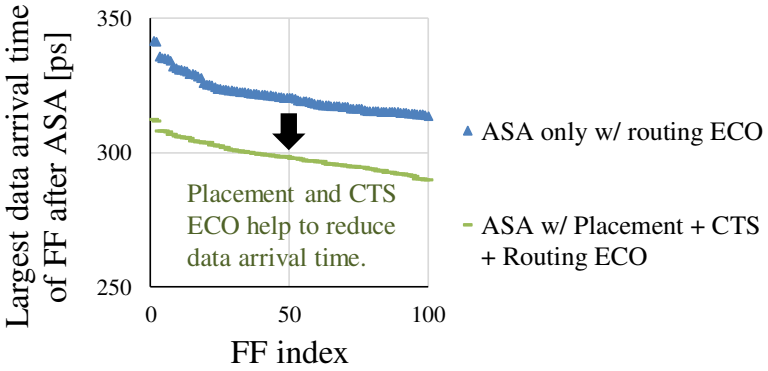

Figure 3.8: Largest data arrival time of each FF after ASA at typical PVTA condition in AES circuit.

synthesis (CTS) and routing ECO and their optimizations. Here, the AES circuit is used, where the detail will be explained in Section 3.6. The worst data arrival time of each FF are obtained by STA at the typical PVTA condition. Fig. 3.8 indicates that three ECO processes at the placement, CTS and, routing achieve the smaller path delay than a single ECO process at routing. Therefore, the proposed ASA performs three ECO processes to more precisely achieve $\Delta setup_{i,j}$ in the layout modification.

3.5.2 Target FF Selection

This section presents a selection method of target FFs aiming at MTTF maximization since MTTF extension can be converted to power saving. An approach of FF selection is to choose target FFs with the descending order of the failure probability. This approach is based on the idea that the FFs with high failure probabilities are more likely to cause timing error. Thus, increasing slack of such FFs improves the MTTF efficiently. However, this selection strategy does not take into account the common paths between FFs, e.g., how many instances are shared between paths and how much paths are shared between FFs. If a set of FFs shares the most of paths, increasing timing slacks of the small number of FFs may be enough for the set, which contributes to reducing the number of target FF selection. In the VLSI circuit, there are many sets of FFs that share the clock path and data path. If we ignore the common path and choose FFs according to their failure probabilities only, the selected FFs may share the large part of the clock and data paths. In this case, the most of the selected FFs could be redundant and thus wasteful in terms of the circuit area and power. To overcome this issue, the proposed method introduces gate-wise failure probabilities. The gate-wise failure probability denotes how much the instance contributes to the timing error. The proposed method first distributes failure probabilities from endpoint FF to instances at the upper stream of the FF as the gate-wise failure probability. Then, the proposed method selects target FFs by solving the covering problem of instances weighted with the gate-wise failure probability to maximize the sum of gate-wise failure probabilities aiming at MTTF maximization.

Fig. 3.9 shows a simple example, where the circuit is composed of ten combinational logic cells and four FFs. Fig. 3.9(a) exemplifies the proposed FF selection, and Fig. 3.9(b) chooses FFs with descending order of the failure probability. The numbers attached to each gate are the gate-wise failure probabilities, where their computation is explained later. Let us suppose $N_{\mathrm{ASA}} = 2$ in the following.

When the slack times of FF2 and FF4 are increased, the slack times of L1, L3, L4, L5, L6, L7, L9, and L10 are also increased. In this case, even if a delay variation occurs at one of L1, L3, L4, L5, L6, L7, L9 and L10, the variation might be concealed by the increased slack. The expected probability of error reduction corresponds to the sum of gate-wise failure probabilities and it is 0.21 (= 0.02 + 0.02 + 0.02 + 0.03 + 0.03 + 0.03 + 0.03 + 0.03). If FF3 and FF4 are chosen like Fig. 3.9(b), the slack times of L5, L6, L7, L8, L9 and L10 are increased. In this case, the reduced failure probability is 0.18 (= 0.03 × 6) and this amount of reduction is smaller than the previous one. In this case, TTF tends to be shorter.

This work proposes an ILP based FF selection method. FF selection problem (or instance covering problem) is formulated as follows.

- Objective
  - Maximize : $\sum_{k=1}^{N_{\mathrm{inst}}} (P_{\mathrm{fail\_inst_k}} \times B_{\mathrm{inst}_k})$

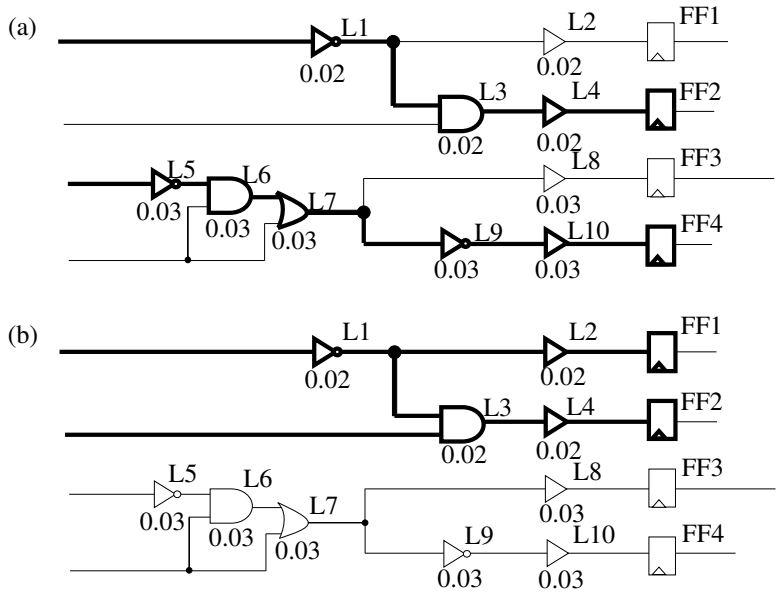

Figure 3.9: Example to select target FFs. (a) The proposed ILP selects FF2 and FF4, and the expected probability of error reduction is 0.21. (b) If FF1 and FF2 are selected, the expected probability of error reduction is 0.08.

- Constraints
  - $0 \leq B_{\mathrm{inst}_k} \leq 1 \quad (1 \leq k \leq N_{\mathrm{inst}})$
  - $0 \leq B_{\mathrm{FF}_i} \leq 1 \quad (1 \leq i \leq N_{\mathrm{FF}})$
  - $\sum_{i=1}^{N_{\mathrm{FF}}} B_{\mathrm{FF}_i} \leq N_{\mathrm{ASA}}$
  - $B_{\mathrm{inst}_k} \leq \sum_{i=1}^{N_{\mathrm{FF}}} (B_{\mathrm{FF}_i} \times B_{\mathrm{FF}_i\_\mathrm{inst}_k})$
- Variables
  - $B_{\mathrm{FF}_i} \quad (1 \leq i \leq N_{\mathrm{FF}})$

The number of instances in the circuit is N_{inst}. The objective of this ILP problem is to maximize the sum of $(P_{\mathrm{fail\_inst_k}} \times B_{\mathrm{inst}_k})$. $P_{\mathrm{fail\_inst_k}}$ is the gate-wise failure probability of k-th instance, representing how much k-th instance contributes to timing error. B_{inst_k} is a binary variable and it becomes 1 when k-th instance is located in the upper stream of any target FF. Therefore, the sum of $P_{\mathrm{fail\_inst_k}} \times B_{\mathrm{inst}_k}$ represents the gate-wise failure probability reduction. In this problem, binary variables B_{FF_i} are assigned, where B_{FF_i} becomes 1 when i-th FF is selected as target FFs.

The first and second constraints are given to restrict B_{inst_k} and B_{FF_i} to binary numbers. The third constraint means that the number of target FFs for ASA should be equal to or less than N_{ASA}. The fourth constraint is a key constraint that defines the relation

between B_{inst_k} and B_{FF_i}. $B_{\text{FF}_i\_\text{inst}_k}$ is a binary constant which is determined by the circuit topology, and it becomes 1 when k-th instance is included in the paths ending at i-th FF. The product term of $B_{\text{FF}_i} \times B_{\text{FF}_i\_\text{inst}_k}$ becomes 1 when both B_{FF_i} and $B_{\text{FF}_i\_\text{inst}_k}$ are 1. B_{inst_k} becomes 0 only when the product of B_{FF_i} and $B_{\text{FF}_i\_\text{inst}_k}$ is 0 for all the FFs. If k-th instance is included in the paths ending at target FFs, at least one of the products of B_{FF_i} and $B_{\text{FF}_i\_\text{inst}_k}$ become 1. In this case, B_{inst_k} can be 1. In this ILP formulation, the sum of $(P_{\text{fail\_inst}_k} \times B_{\text{inst}_k})$ is maximized, and hence B_{inst_k} is necessarily assigned to be 1.

The remaining issue is $P_{\text{fail\_inst}_k}$ calculation. The failure probabilities at individual FFs, $P_{\text{fail\_FF}_i}$, can be computed referring to Fig. 3.6. Now, $P_{\text{fail\_inst}_k}$ can be calculated using $P_{\text{fail\_FF}_i}$ as follows.

$$P_{\text{fail\_inst}_k} = \max_{1 \le i \le N_{\text{FF}}} \left\{ \frac{P_{\text{fail\_FF}_i}}{\sum_{k=1}^{k_{\max}} (B_{\text{FF}_i\_\text{inst}_k})} \right\}. \tag{3.4}$$

The above equation assumes that each instance included in the fan-in cone of FF_i has the same contribution to a timing error for simplicity, and hence the $P_{\text{fail\_FF}_i}$ is divided by the number of instances in the fan-in cone of FF_i. When we need to consider the different contributions of each instance due to, for example, different intrinsic variation sensitivities of the instances themselves, we may distribute $P_{\text{fail\_FF}_i}$ to each gate wise failure probability taking into account the different sensitivities. Note that an instance can be included in the fan-in cones of multiple FFs and hence, a max operation is performed in Eq. (3.4).

This work uses ILP to derive an exact solution. For this meaning, other techniques, such as SAT, could be used for the FF selection. Besides, ILP has proven to be NP-hard [93] in general and thus the ILP may not be suitable for large-scale optimization problems due to computational cost. To mitigate the computational cost, this work only considers timing-critical and activating paths in ASA, which will be explained in Section 3.6.1, and reduces the size of the design problem. Although the weakness of the ILP is not completely solved, the FF selection problem of the two benchmark circuits, which will be explained in Section 3.6.1, can be solved in a few seconds thanks to the problem size reduction. When the circuit size becomes larger and the CPU time is unacceptable, we need to, for example, find an approximate solution or partition the circuit into sub-circuits for problem size reduction.

3.5.3 $\Delta\text{setup}_{i,j}$ Determination

Next, this section determines $\Delta\text{setup}_{i,j}$ for the set of target FFs selected in the previous section. Fig. 3.10 shows our ASA strategy. The proposed method gives timing slacks for each target FF so that the failure probability of each FF is equal to or smaller than the target failure probability, i.e., $P^{\max}_{\text{fail\_FF}}$. Note that ASA cannot change the activation probability but can adjust the timing violation probability. Therefore, this work adjusts the timing violation probability of each FF and thus sets the failure probability to the

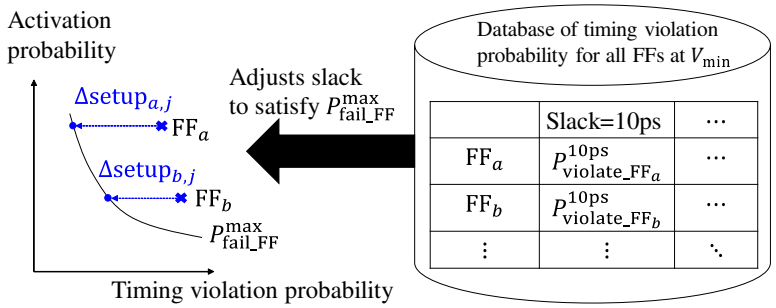

Figure 3.10: $\Delta setup_{i,j}$ determination. Proposed ASA adjusts the setup slack so that each FF satisfies $P^{max}_{fail\_FF}$.

$P^{max}_{fail\_FF}$. In this work, the relation of the timing violation probability and the timing slack is obtained from SSTA.

In summary, the proposed design needs tasks of logic simulation and STA for deriving activation probabilities of paths/FF, SSTA for deriving timing violation probabilities, pre-ASA circuit selection, ILP for selecting target FFs for ASA, ASA with layout ECO, and MTTF calculation, whereas the conventional WC design requires only STA or SSTA. In case of the MTTF-aware design w/o ASA, logic simulation, STA, SSTA, and MTTF calculation are necessary.

3.6 Experimental Evaluation

This section experimentally evaluates the performance improvement thanks to the proposed MTTF-aware design in comparison to the conventional WC design. Section 3.6.1 explains the evaluation setup and Section 3.6.2 demonstrates the power saving effects by MTTF-aware operation and ASA individually. Lastly, Section 3.6.3 examines the power saving effects thanks to the proposed MTTF-aware design and discusses the effectiveness of the proposed ASA.

3.6.1 Evaluation Setup

This work used the AES circuit and OR1200 OpenRISC processor as target circuits. These two circuits were laid out by a commercial P&R tool with a 45 nm Nangate standard cell library [77]. The minimum clock period of post-layout circuits at 1.20 V in the typical PVTA conditions and the worst-case were 3,150 ps and 3,800 ps in OpenRISC and 370 ps and 480 ps in AES, respectively. Hereafter, the target clock period was set to 3,800 ps in OpenRISC and 480 ps in AES, and then ASA optimized the timing slack of the FF/path for these target clock periods.

The post-layout circuits included 23,247 combinational logic cells, 2,504 FFs, 2

macro cells of standard cell memory in OpenRISC, and 17,948 combinational logic cells and 530 FFs in AES, respectively. Thus, sets of N_{inst} and N_{FF} were 23,249 and 2,504 in OpenRISC, 17,948 and 530 in AES, respectively. This work used Gurobi Optimizer 7.0 [94] to solve the ILP problem defined in Sections 3.5.2. The solver was executed on a 2.4 GHz Xeon CPU machine under the Red Hat Enterprise Linux 6 operating system with 1 TB memory. The required CPU times for solving the proposed ILP problem with Gurobi optimizer were at most 2.56 seconds in AES and 0.53 seconds in OpenRISC. The evaluation took into account the following variations, which are similar to Section 2.4.1.

- Manufacturing variability, which is assumed to consist of intra-die random variation and inter-die variation. Both intra-die and inter-die variation includes NMOS and PMOS threshold voltage variation of σ = 10 mV.
- Dynamic supply noise, which is assumed to fluctuate between -50 mV and 50 mV by 10 mV with eleven steps.
- NBTI aging model with six degradation states of 0 mV, 0.5 mV, 1 mV, 5 mV, 10 mV, and 15 mV are prepared. The necessary information is obtained similarly with Section 2.4.1.

As for workload, this work selected three benchmark programs (crc, sha, and dijkstra) from MIBenchmark [83] for OpenRISC and 1,000 random test patterns for AES, which are similar to Section 2.4.1. The number of activated paths, i.e., $N_{\text{act\_path}}$, was 167,626 in OpenRISC and 81,829 in AES, respectively. Fig. 3.11 shows the distributions of FF activation probabilities in AES and OpenRISC. We can see that FFs in OpenRISC are less activated and their activation probabilities are widely spread, which suggests the ASA is more effective to OpenRISC.

In the evaluation, the MTTF of 10 years in OpenRISC and 1.6 years in AES were set as MTTF_{min}. From Eq. (3.1), the upper bound of the failure probability of a FF, i.e., $P^{\text{max}}_{\text{fail\_FF}}$, is set to 3.99×10^{-21} [1/cycle] in OpenRISC and 1.88×10^{-20} [1/cycle] in AES. These $P^{\text{max}}_{\text{fail\_FF}}$ were used in $\Delta\text{setup}_{i,j}$ determination as explained in Section 3.5.3.

In MTTF evaluation, this work placed emphasis on not to miss paths which affect the MTTF, i.e., a conservative approach should be taken. Here, considering all the activated paths may lead to prohibitively long CPU time. Also, as previously shown in Fig. 3.4, several FFs may have high failure probabilities and dominantly determine the MTTF, i.e., taking into account these paths or FFs may be enough to evaluate the MTTF. Motivated by these considerations, this work selected the paths for MTTF evaluation with the following two steps. Firstly, this work calculated the maximum standard deviation of path delay variation for all the activate paths and extracted the largest standard deviation of the path delay variation. Secondly, this work calculated the timing violation probability assuming that all the paths had the above largest standard deviation and selected the paths whose failure probabilities were higher than the target failure

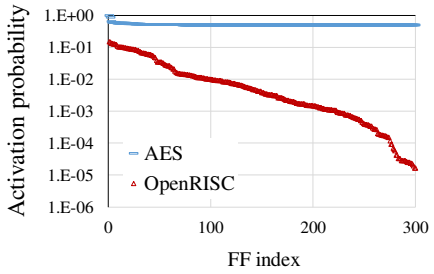

Figure 3.11: Activation probabilities of FFs.

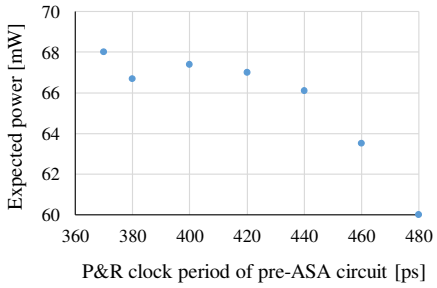

Figure 3.12: Expected minimum power after ASA in AES.

probability. Note that the target failure probability of paths $P^{\max}_{\text{fail\_path}}$ is calculated with the following equation, which is similar to Eq. (3.1).

$$P^{\max}_{\text{fail\_path}} = \frac{1}{N_{\text{act\_path}} \times \text{MTTF}_{\text{min}}}. \tag{3.5}$$

The paths whose failure probability is lower than $P^{\max}_{\text{fail\_path}}$ do not contribute to the violation of the target MTTF even with the worst delay variation, and hence these paths were excluded from $N_{\text{act\_path}}$ paths in MTTF evaluation. Thus, this work selected 1,227 paths from 167,626 paths in OpenRISC and 21,067 paths from 81,829 paths in AES, and evaluated the MTTF.

Area_{max} and $N^{\max}_{\text{LVth}}$ were set to 2.02 mm^2 and 4,494 in OpenRISC, and 0.05 mm^2 and 17,231 in AES, respectively. These values come from the area and the number of low-V_{th} cells of the pre-ASA circuits designed at 3,150 ps in OpenRISC and 370 ps in AES, respectively.

With this setup, ASA was performed to both AES and OpenRISC. The number of pre-ASA candidate circuits was seven in AES, where P&R clock periods of these pre-ASA circuits were 370, 380, 400, 420, 440, 460 and 480 ps. As for OpenRISC, two candidates with 3,150 ps and 3,800 ps were given. Then, this work evaluated the MTTF, average supply voltage, and average power under PVTA variation by the stochastic MTTF estimation framework [61]. This work prepared seven supply voltages from 1.20 to 0.90 V with 50 mV interval and swept clock period with 10 ps interval from 300 to 1,000 ps in AES and from 3,000 to 8,000 ps in OpenRISC. Power dissipation was calculated for each pair of supply voltage and clock period with a commercial power estimation tool [82], which reports dynamic and leakage power separately. Note that dynamic power is much higher than leakage in both OpenRISC and AES with our evaluation setup.

3.6.2 Evaluation Results

Selection of Pre-ASA Circuit and ASA Implementation

Fig. 3.12 shows the estimation results of the expected minimum power after ASA for each pre-ASA candidate in AES. Fig. 3.12 shows that the pre-ASA candidate designed at 480 ps is the most promising one regarding power. Then the pre-ASA circuit that is laid out at 480 ps were selected. Note that the expected minimum supply voltage that satisfies the target MTTF after ASA, V_{min}, is 0.90 V in all the pre-ASA candidates. Similar to AES, we evaluate the expected minimum power of OpenRISC and select the circuit that is laid out at 3,800 ps.

Next, ASA is performed to the chosen candidate. Figs. 3.13 and 3.14 show the area and the number of low-V_{th} cells in the ASA circuits. In both the figures, the area and the number of low-V_{th} cells are normalized by $\mathrm{Area}_{\mathrm{max}}$ and $N_{\mathrm{LVth}}^{\mathrm{max}}$, respectively. Taking into account the constraints of $\mathrm{Area}_{\mathrm{max}}$ and $N_{\mathrm{LVth}}^{\mathrm{max}}$, we set N_{ASA}, which is the number of FFs to which ASA is applied, to 300 in both AES and OpenRISC. An interesting observation is that the proposed ASA circuits of AES and OpenRISC have a smaller area and a smaller number of low-V_{th} cells compared to the pre-ASA circuit. For example, when N_{ASA} equals 300, circuit area is reduced from $\mathrm{Area}_{\mathrm{max}}$ to $0.986 \times \mathrm{Area}_{\mathrm{max}}$ by 1.4% in OpenRISC and by 6.4% in AES. This observation will be discussed in Section 3.6.3.

Power Saving Effects

Fig. 3.15 shows trade-offs between the power dissipation and the clock period under the MTTF constraint of 10 years in OpenRISC and 1.6 years in AES. The black quadrilateral plots represent the conventional WC design with guard-banding for PVTA variation. The yellow circular plots are also the WC design, but the MTTF-aware chip-wise voltage assignment is performed. The blue triangular plots correspond to the proposed ASA with MTTF-aware operation. Note that one trade-off corresponds to one circuit, e.g., the blue trade-off shows the pair of the power dissipation and clock period of the identical ASA circuit. To evaluate trade-off relation for the pre-ASA circuit and the ASA circuit, at each clock period, the minimum supply voltage for satisfying the target MTTF was explored and then the power dissipation was calculated. This section examines individual contributions of MTTF-aware operation and the proposed ASA in addition to the overall power saving.

First, this section compares the black quadrilateral and blue triangular plots. Fig. 3.15 shows that the proposed MTTF-aware design reduces average power while satisfying the MTTF constraint. In Fig. 3.15(a), at a clock period of 3,800 ps, the proposed design achieves the target MTTF with an average power of 13.2 mW, whereas the conventional WC design consumes 23.0 mW. The power saving due to the proposed design is 42.3%. Similarly, in Fig. 3.15(b), at 480 ps, the proposed design achieves 49.6% power saving from 132.0 mW to 66.5 mW. It is experimentally confirmed that the proposed

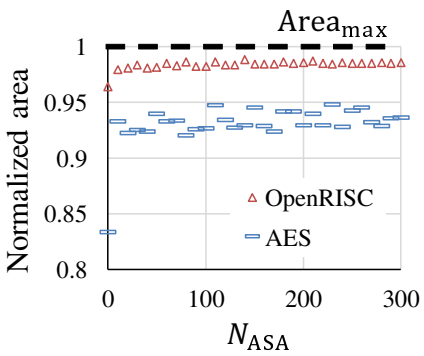

Figure 3.13: Normalized area of ASA circuits. Y-axis is normalized by Area_{max}.

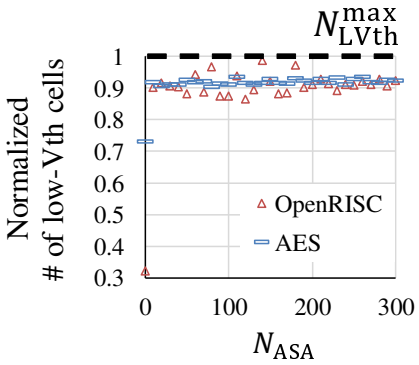

Figure 3.14: Normalized number of low-V_{th} cells in ASA circuits. Y-axis is normalized by $N^{\text{max}}_{\text{LVth}}$.

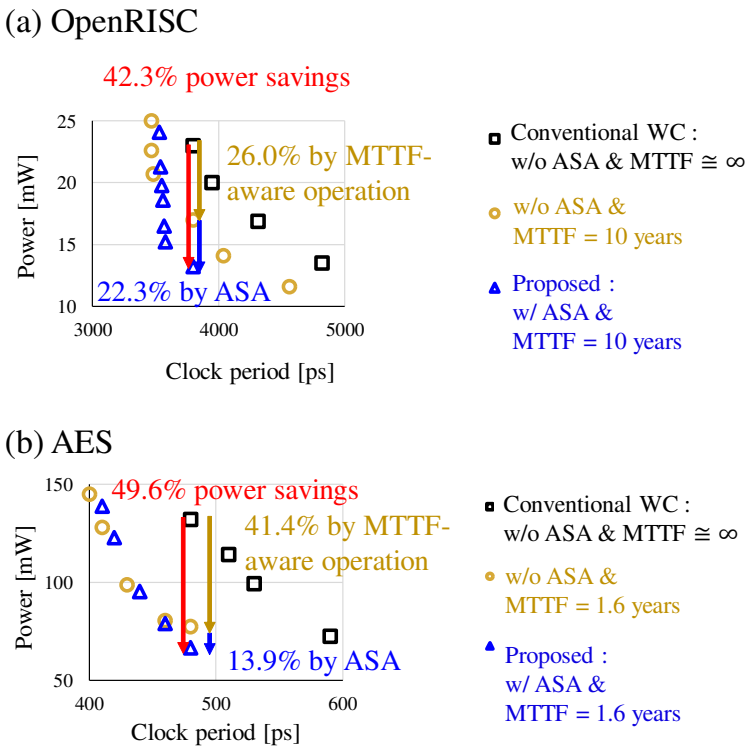

Figure 3.15: Trade-offs between clock period and average power. (a) OpenRISC, (b) AES.

MTTF-aware design made the significant power saving both in AES and OpenRISC while reducing circuit area by 1.4% in OpenRISC and 6.4% in AES.

Second, this section compares the conventional WC with and without MTTF-aware

operation, i.e., black quadrilateral and yellow circular plots. Fig. 3.15 shows that the MTTF-aware operation improves performance from the conventional WC. For example, at 3,800 ps, MTTF-aware operation achieves 26.0% power saving from 23.0 mW to 17.0 mW in OpenRISC and 41.4% power saving from 132.0 mW to 77.3 mW in AES. This power saving effects reveal that MTTF-aware operation can significantly reduce the excessive operation margin while satisfying the target MTTF without any circuit modification.

Third, this section compares the yellow circular and blue triangular plots to clarify the performance improvement by the proposed ASA. Fig. 3.15 shows that the proposed ASA further improves performance from the conventional WC with MTTF-aware operation. For example, at 3,800 ps, MTTF-aware operation achieves 22.3% power saving from 17.0 mW to 13.2 mW in OpenRISC and 13.9% power saving from 77.3 mW to 66.5 mW in AES. Here, it should be noted that AES has many FFs with the higher activation probability as shown in Fig. 3.11, which means that paths having the slack of 0 or close to 0 tend to have the higher failure probability. Thus, the effectiveness of the activation-aware slack assignment is smaller in AES than in OpenRISC. It is also observed that the performance improvement thanks to ASA is the largest around the target clock periods of 3,800 ps in OpenRISC and 480 ps in AES and it becomes smaller as the period goes away from the target one since ASA optimized the circuit at the target clock period under the MTTF constraint. There could be room for improvement at different clock periods.

3.6.3 Discussion

This subsection first examines the power saving in terms of V_{dd}, area and the number of low-V_{th} cells, and second discusses the effectiveness of the proposed ASA regarding FF selection and slack determination.

Reduction of V_{dd}, Area, and the Number of Low-V_{th} Cells

The performance evaluation results in Section 3.6.2 showed that the proposed design saved power significantly. Let us investigate its reason.

Firstly, this section examines the supply voltage reduction effects by the proposed MTTF-aware design. Fig. 3.16 shows trade-offs between the average supply voltage and the clock period under the MTTF constraints of 10 years in OpenRISC and 1.6 years in AES. We can see that the proposed design, which corresponds to the blue triangular plots, achieves the target MTTF at a lower supply voltage compared with conventional WC design, i.e., black quadrilateral plots. For example, in Fig. 3.16(a), at a clock period of 3,800 ps, the proposed design achieves the target MTTF at an average supply voltage of 0.90 V, whereas the conventional WC design requires 1.20 V operation, which means the proposed design achieves 25.0% V_{dd} reduction from 1.20 V to 0.90 V. Thanks to this V_{dd} reduction, the circuit power dissipation is dramatically reduced as shown in

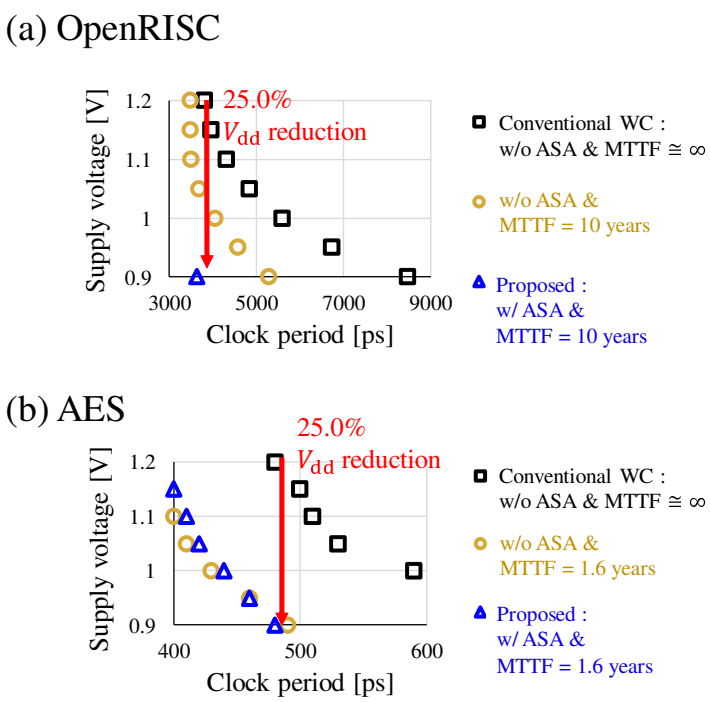

Figure 3.16: V_{dd} reduction by the proposed MTTF-aware design. (a) OpenRISC, (b) AES.

Fig. 3.15. Also in AES, the proposed MTTF-aware design reduces the supply voltage from 1.20 V to 0.90 V and achieves 25.0% V_{dd} reduction as shown in Fig. 3.16(b).

Secondly, this section investigates the area and the number of low-V_{th} cells of ASA circuits, which are partially shown in Figs. 3.13 and 3.14. Figs. 3.17 and 3.18 show the area and the number of low-V_{th} cells of the conventional and proposed ASA circuits. In both figures, the area and the number of low-V_{th} cells are normalized by Area_{max} and N_{LVth}^{max} similar to Figs. 3.13 and 3.14. In this context, conventional ASA is supposed to use pre-ASA circuits that are laid out at FMAX, e.g., 3,150 ps in OpenRISC and 370 ps in AES. On the other hand, the proposed methodology performs ASA to circuits laid out for longer periods of 3,800 ps in OpenRISC and 480 ps in AES. Note that pre-ASA AES circuit designed at 370 ps has only 81 FFs whose failure probabilities are larger than 0, and hence, the maximum N_{ASA} is set to 81.

Figs. 3.17 and 3.18 show that the proposed ASA reduces the area and the number of low-V_{th} cells from conventional ASA circuits and even from their pre-ASA circuits. For example, at $N_{ASA} = 300$, in OpenRISC, the proposed ASA reduces the area by 1.4% from Area_{max} to $0.986 \times \text{Area}_{max}$ and the number of low-V_{th} cells by 7.7% while the conventional ASA increases the area by 0.1% and increases the number of low-V_{th} cells by 13.1%. Similarly, at $N_{ASA} = 300$ in AES, the proposed ASA reduces the area by

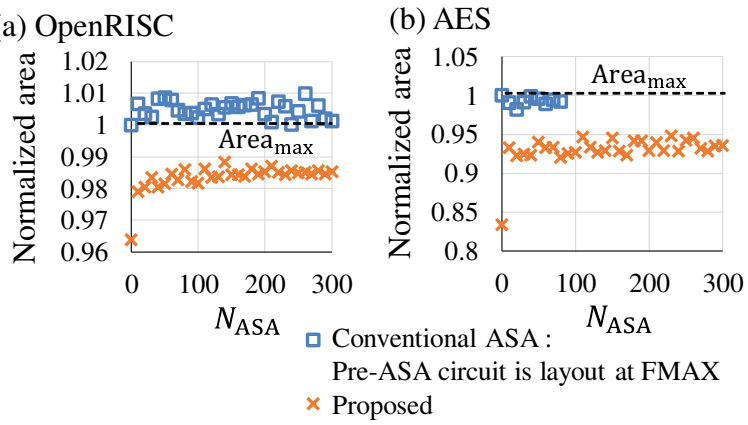

Figure 3.17: Area of conventional and proposed ASA circuits. Y-axis is normalized by Area_{max}. Proposed ASA reduces area from pre-ASA circuits. (a) OpenRISC, (b) AES.

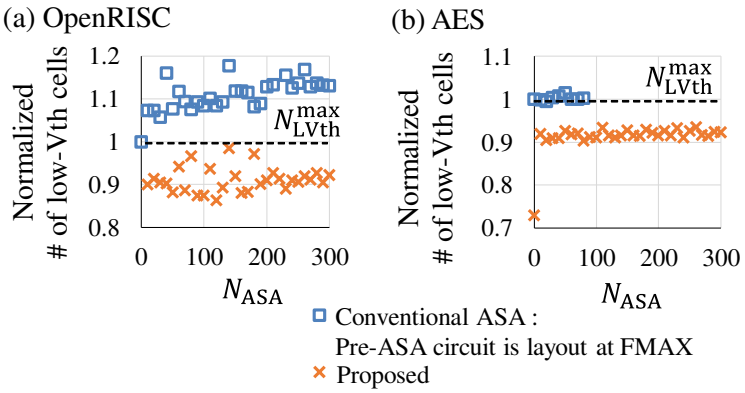

Figure 3.18: The number of low-V_{th} cells of the conventional and proposed ASA circuits. Y-axis is normalized by N_{LVth}^{max}. Proposed ASA reduces the number of low-V_{th} cells from pre-ASA circuits. (a) OpenRISC, (b) AES.

6.4% and decreases the number of low-V_{th} cells by 7.7%. These reductions directly decrease the dynamic and static power dissipation. Thus, this section confirmed that the proposed pre-ASA circuit selection further contributes to improving power in addition to V_{dd} reduction by ASA.

Effectiveness of FF Selection for ASA

Next, this section compares the proposed methodology with the following two approaches focusing on the effectiveness of the target FF selection for ASA.

C1: Choose FFs in an ascending order of slack time.

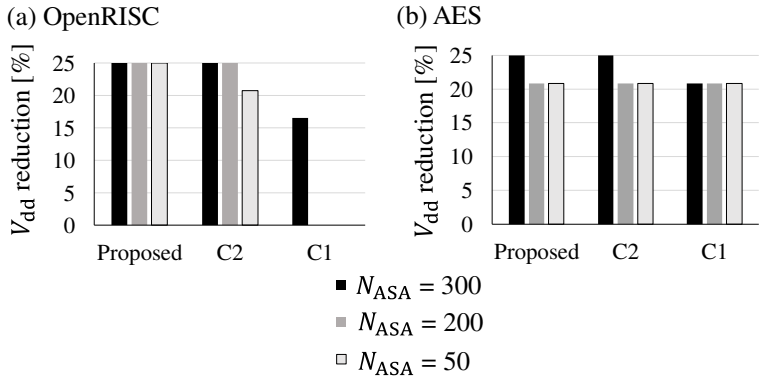

Figure 3.19: Achieved V_{dd} reduction. (a) OpenRISC, (b) AES.

C2: Choose FFs in a descending order of the failure probability.

The first approach of C1 supposes that timing-critical FFs are most likely to cause timing error. This method needs only STA or SSTA timing reports, and hence this approach is more tractable. The second approach of C2 places importance on the failure probability. Remind that the failure probability is defined as the joint probability of the timing violation probability and the activation probability. To calculate the activation probability, we need to perform a logic simulation with prospective workloads or to calculate signal transition rates analytically by, for example, [95].

Fig. 3.19 summarizes the achieved supply voltage reduction with the proposed methodology, C1 and C2. The proposed methodology achieves the largest supply voltage reduction, which is 25.0% from 1.20 V to 0.90 V, in the cases of N_{ASA} = 50, 200 and 300 in OpenRISC and N_{ASA} = 300 in AES. Besides, C2 also achieves the 25.0% supply voltage reduction in the cases of N_{ASA} = 200 and N_{ASA} = 300 in OpenRISC and N_{ASA} = 300 in AES. Here, the difference between C1 and C2 in AES is much smaller than OpenRISC. For example, at N_{ASA} = 50, both C1 and C2 achieves 20.8% V_{dd} reduction in AES whereas C1 could not reduce supply voltage and C2 attains 20.8% V_{dd} reduction in OpenRISC. A possible reason is that, in AES, activation probabilities of FFs are much higher and more similar than OpenRISC as shown in Fig. 3.11, and hence timing critical FFs are likely to have higher failure probabilities.

Fig. 3.20 shows the MTTF comparison, where N_{ASA} = 50 and V_{dd} = 0.90 V. From this figure, we can see that the proposed methodology attains the best trade-off relation between the MTTF and clock period. For example, from Fig. 3.20(a), at a clock period of 3,800 ps, the proposed ASA improves the MTTF, compared to C1, from 3.32×10^{-14} years to 9.56×10^{1} years by 15 orders of magnitude and, compared to C2, from 9.07×10^{-9} years to 9.56×10^{1} years by 10 orders of magnitude. Similarly, from Fig. 3.20(b), at a clock period of 480 ps, the proposed ASA improves the MTTF by 6 orders of magnitude from C1 and by 5 orders of magnitude from C2. Thus, the MTTF

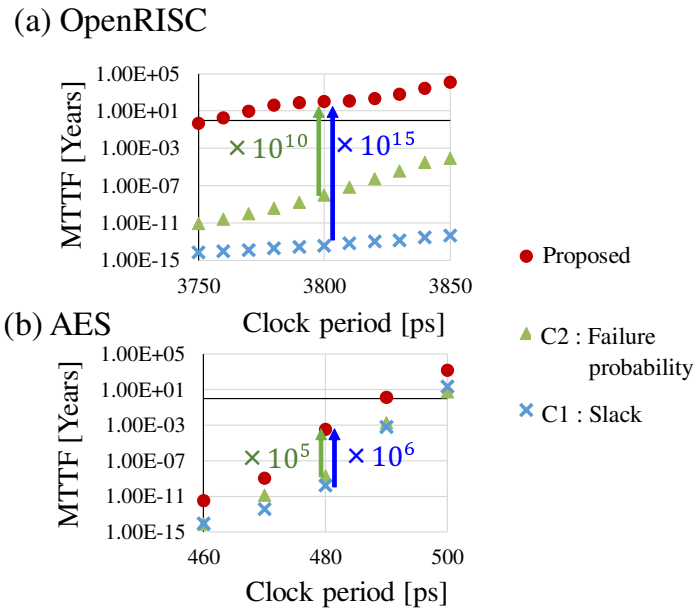

Figure 3.20: MTTF comparison. Proposed FF selection improves the MTTF significantly compared with C1, and C2. (a) OpenRISC, (b) AES.

improvement of the proposed ASA is remarkable even while the proposed ASA also reduces the area and the number of low-V_{th} cells. The longer MTTF means fewer timing errors in field, which is also desirable for resilient circuit designs, such as Razor [29] and Tunable Replica Circuit (TRC) [33], and error prediction technique, e.g., TEP-FF [35]. With the ASA, the power dissipation of such resilient circuits could be reduced further and the reliability would improve, Chapter 4 will demonstrate that the simultaneous optimization of the main logic with AVS and the TEP-FF insertion further extends the MTTF and thus saves the power dissipation.

Effectiveness of Slack Determination

Lastly, this section investigates the importance of slack determination, i.e., $\Delta\text{setup}_{i,j}$ determination, by ASA. Fig. 3.21 shows the comparison results of the area and the number of low-V_{th} cells between the proposed ASA and the naive approach that increases slack as much as possible. In this comparison, the identical pre-ASA OpenRISC circuit laid out at 3,800 ps is given to clarify the effectiveness of the $\Delta\text{setup}_{i,j}$ determination. From Fig. 3.21, we can see that the proposed ASA saves both the area and the number of low-V_{th} cells. For example, at $N_{\text{ASA}} = 300$, the proposed ASA saves the area by 1.6% and reduces the number of low-V_{th} cells by 8.1%. These reductions contribute to dynamic and static power reduction. These results show that the determination of $\Delta\text{setup}_{i,j}$ in the proposed ASA more contributes to power saving compared with the conventional ASA

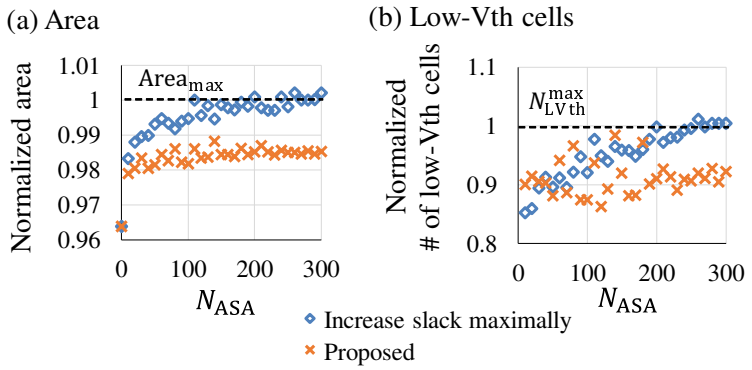

Figure 3.21: Comparison of (a) area and (b)the number of low-V_{th} cells in OpenRISC. Y-axis is normalized by Area_{max} in (a) and by $N_{\text{LVth}}^{\text{max}}$ in (b).

that increases slack maximumly. Note that, in AES, the proposed ASA also increases slack maximumly, and thus the area and the number of low-V_{th} cells are identical to those of the conventional ASA.

3.7 Conclusion

This chapter proposed the MTTF-aware design methodology. The key ideas of the proposed design methodology are MTTF-aware operation and design optimization with ASA, where the ASA gives timing slacks to non-intrinsic active critical paths and reduces the number of activated paths whose delays are very close to those of the inherent critical paths. The proposed optimization includes pre-ASA circuit design and ASA implementation. In pre-ASA circuit design, the proposed methodology selects the most promising one from candidates regarding power dissipation. For each candidate, this selection estimates the minimum supply voltage after ASA (V_{min}) at which the circuit can achieve the target MTTF and evaluates the power at V_{min}. Thus, this chapter chooses the circuit whose estimated power is minimum. Then, the proposed methodology selects a set of FFs for FF-based ASA using integer linear programming (ILP) so that it reduces the sum of gate-wise failure probabilities maximumly. This chapter evaluated the MTTF of circuits with and without ASA and examined how much power saving could be obtained while satisfying the target MTTFs of 10 years in OpenRISC and 1.6 years in AES. The evaluation results showed that the circuits with ASA achieved 49.6% power saving in AES circuit and 42.3% power saving in OpenRISC processor. Also, thanks to pre-ASA design in the proposed methodology, 6.4% of the area and 7.7% of low-V_{th} cells are reduced in the AES circuit, and 1.4% of the area and 7.7% of low-V_{th} cells are reduced in the OpenRISC processor.

Chapter 4

MTTF-aware Design Methodology of Adaptively Voltage Scaled Circuit

This chapter proposes a MTTF-aware design methodology of AVS circuit [96]. The proposed design methodology optimizes both the voltage scaled circuit under the AVS and the sensor. As for the voltage scaled circuit design, this design methodology utilizes ASA which is previously explained in Chapter 3. As for the sensor, this methodology focuses on TEP-FF which is selected in Chapter 2, and this work proposes a novel TEP-FF insertion method that minimizes the sum of gate-wise timing failure probabilities aiming at MTTF maximization.

4.1 Introduction

As discussed in Chapter 1, to implement AVS systems that fully exploit run-time adaptation and eliminate the redundant margin, we should pay attention to the main logic circuit under AVS in addition to the sensing circuit. As for sensor selection, Chapter 2 experimentally confirmed that TEP-FF contributes to reducing redundant timing margin for both the global and random performance variation. Also, as for voltage scaled circuit design under AVS, Chapter 3 suggests that ASA which allocates larger slack to highly active paths, could improve the efficacy of the AVS and enable further power savings with extremely low error rate.

This chapter proposes a design methodology for AVS circuits. The proposed methodology optimizes both the main logic under AVS and sensing circuit. In the main logic design, this design methodology performs a MTTF-aware ASA which is explained in Chapter 3 and estimates the MTTF of AVS circuits with a stochastic framework [61]. As for the sensing circuit design, the methodology focuses on TEP-FF, which is selected in Chapter 2, and this work proposes a novel sensor insertion method that maximumly decreases the sum of gate-wise timing failure probabilities, where the timing failure probability is the joint probability of activation and timing violation probabilities. By

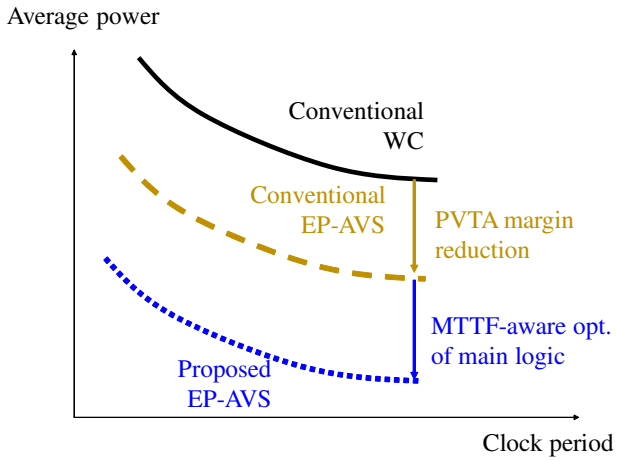

Figure 4.1: Expected performance improvement thanks to the proposed AVS design methodology.

exploiting the information on the paths with the higher timing failure probability, the proposed sensor insertion makes AVS efficiently monitor the timing-critical and highly-active FFs. Experimental results show that MTTF-aware main logic design is highly compatible with AVS, and they mutually enhance and provide further power savings and performance improvement with margin elimination.

Fig. 4.1 illustrates the expected power saving effects. The top black curve represents the conventional WC design that adds timing margins assuming the worst PVTA condition. The middle yellow and bottom blue curves correspond to the conventional AVS without main logic optimization and the proposed AVS with the ASA. The proposed AVS is expected to attain a better trade-off relation between the clock period and power thanks to the main logic optimization. This power saving effects in an embedded processor and a cipher circuit will be experimentally demonstrated.

The rest of this chapter is organized as follows. Section 4.2 explains the overview of the proposed design which consists of the main logic optimization and the sensing circuit optimization. Section 4.3 explains the proposed sensor insertion methodology, which is applied to the ASA circuit designed referring to Chapter 3. Section 4.4 evaluates the trade-off between average power and the clock period of the conventional WC design, conventional AVS, and the proposed AVS and demonstrates the power saving effects thanks to the proposed AVS. Lastly, concluding remarks are given in Section 4.5.

4.2 Overview of Proposed Design Methodology for AVS

The proposed design methodology for AVS consists of the ASA for the main logic under AVS and the insertion of error prediction sensors. Referring to Chapter 2, this

chapter assumes the AVS circuit which is previously illustrated in Fig. 2.2. Section 4.2.1 formulates the design optimization problem of AVS. Then, Section 4.2.2 explains the overview of the proposed design methodology.

4.2.1 Problem Definition of AVS Design

Based on the discussion in the Section 4.2.1, this work formulates the design optimization of AVS including ASA and TEP-FF insertion.

- Input
  - N_{CKT} pre-ASA candidates
- Output
  - one AVS circuit
- Objective
  - Minimize : Power = $\min(\mathrm{Power}_1, \cdots, \mathrm{Power}_{N_{\mathrm{CKT}}})$
- Constraints
  - $\mathrm{MTTF}_j \geq \mathrm{MTTF}_{\mathrm{min}} (1 \leq j \leq N_{\mathrm{CKT}})$
  - $\mathrm{Area}_{\mathrm{ASA}_j} \leq \mathrm{Area}_{\mathrm{ASA}}^{\mathrm{max}} (1 \leq j \leq N_{\mathrm{CKT}})$
  - $\mathrm{Area}_{\mathrm{TEP}_j} \leq \mathrm{Area}_{\mathrm{TEP}}^{\mathrm{max}} (1 \leq j \leq N_{\mathrm{CKT}})$
  - $\mathrm{NLvth}_{\mathrm{ASA}_j} \leq \mathrm{NLvth}_{\mathrm{ASA}}^{\mathrm{max}} (1 \leq j \leq N_{\mathrm{CKT}})$
- Variables
  - $\Delta\mathrm{setup}_{i,j} \quad (1 \leq i \leq N_{\mathrm{FF}}, 1 \leq j \leq N_{\mathrm{CKT}})$
  - $B_{\mathrm{TEP}_{i,j}} \quad (1 \leq i \leq N_{\mathrm{FF}}, 1 \leq j \leq N_{\mathrm{CKT}})$

The inputs of this problem are N_{CKT} pre-ASA candidates, and the output is one AVS circuit in which ASA is applied and TEP-FFs are inserted. The objective of this problem is to minimize the power of the AVS circuit. The AVS circuit is constrained by the MTTF ($\mathrm{MTTF}_{\mathrm{min}}$), circuit area ($\mathrm{Area}_{\mathrm{ASA}}^{\mathrm{max}}$ and $\mathrm{Area}_{\mathrm{TEP}}^{\mathrm{max}}$), and the number of low-V_{th} cells ($\mathrm{NLvth}_{\mathrm{ASA}}^{\mathrm{max}}$). The variables $\Delta\mathrm{setup}_{i,j}$ are the slacks given to FFs in j-th pre-ASA circuit by ASA, where $\Delta\mathrm{setup}_{i,j}$ is given to the layout ECO as an intentional increase in setup time of i-th FF_i in j-th pre-ASA circuit. N_{FF} is the number of FFs in the circuit, and it is identical in all the pre-ASA circuits. When $\Delta\mathrm{setup}_{i,j} = 0$, i-th FF_i is not included

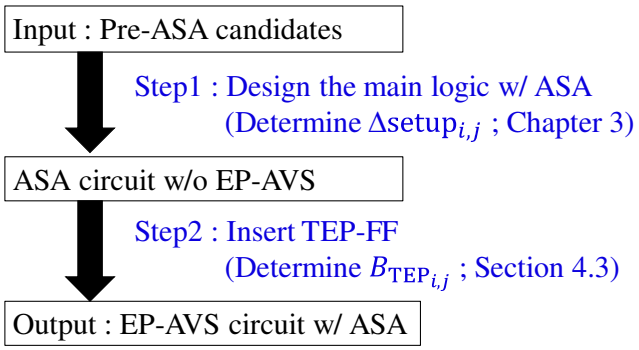

Figure 4.2: Overview of the proposed design. The proposed design methodology with a two-stage procedure: (1) Design the main logic under AVS with ASA, (2) Insert TEP-FF.

in the set of target FFs for ASA of j-th pre-ASA circuit. Thus, the number of target FFs in j-th pre-ASA circuit is expressed as the number of FFs whose $\Delta\mathrm{setup}_{i,j}$ is larger than 0. $B_{\mathrm{TEP}_{i,j}}$ is a binary variable, and it becomes 1 when i-th FF in j-th circuit is replaced to TEP-FF. Therefore, the number of TEP-FF in j-th pre-ASA circuit is expressed as the number of FFs whose $B_{\mathrm{TEP}_{i,j}}$ equals to 1. Here, MTTF_j depends on $\Delta\mathrm{setup}_{i,j}$ and $B_{\mathrm{TEP}_{i,j}}$ and these relations are evaluated by the stochastic error rate estimation method [61]. $\mathrm{Area}_{\mathrm{ASA}_j}$ and $\mathrm{NLvth}_{\mathrm{ASA}_j}$ vary depending on $\Delta\mathrm{setup}_{i,j}$, and $\mathrm{Area}_{\mathrm{TEP}_j}$ is determined by $B_{\mathrm{TEP}_{i,j}}$.

4.2.2 Overview of Proposed AVS

A difficulty to solve the formulated problem is the non-linear relations among MTTF_j, $\mathrm{Area}_{\mathrm{ASA}_j}$, $\mathrm{Area}_{\mathrm{TEP}_j}$, $\mathrm{NLvth}_{\mathrm{ASA}_j}$, $B_{\mathrm{TEP}_{i,j}}$, and $\Delta\mathrm{setup}_{i,j}$. Also, the evaluations of MTTF_j, $\mathrm{Area}_{\mathrm{ASA}_j}$, $\mathrm{Area}_{\mathrm{TEP}_j}$, and $\mathrm{NLvth}_{\mathrm{ASA}_j}$ need relatively long CPU time, and hence an explicit optimization is difficult concerning CPU time. Thus, to determine the set of $\Delta\mathrm{setup}_{i,j}$ and $B_{\mathrm{TEP}_{i,j}}$ efficiently, this work proposes a two-step procedure.

Fig. 4.2 shows the overview of the proposed design which includes both the main logic design and sensor insertion. The proposed design methodology solves this problem with the two-stage procedure. The first stage designs the main logic under AVS using ASA referring to Chapter 3, i.e., determines $\Delta\mathrm{setup}_{i,j}$, and the second stage performs TEP-FF insertion, i.e., determines $B_{\mathrm{TEP}_{i,j}}$. The following section explains the TEP-FF insertion.

4.3 Sensing Circuit Insertion

To make AVS work well, TEP-FFs need to output the error prediction signals frequently to adjust the supply voltage, and hence it is desirable to insert TEP-FFs to highly activated FFs. Also, FFs with small slacks need fewer delay buffers in TEP-FFs. Accordingly, FFs having the higher timing failure probabilities satisfy both the desirable properties above. This work, therefore, proposes a novel TEP-FF insertion method that minimizes the sum of gate-wise timing failure probabilities aiming at MTTF maximization, which has a similarity with the ASA previously exemplified in Chapter 3. The proposed insertion method consists of the following two steps: (1) calculating timing failure probabilities, and (2) finding out a set of FFs that maximally reduces the sum of gate-wise failure probabilities by solving instance covering problem as an ILP problem.

In the first step, the proposed method calculates timing failure probabilities of FFs as explained in Fig. 3.6. Then, this work computes gate-wise failure probabilities, $P_{\text{fail\_inst}_k}$, from $P_{\text{fail\_FF}_i}$ with Eq. (3.4), which is similar to Section 3.5.2.

In the second step, this work proposes the FF selection methodology that maximizes the sum of gate-wise timing failure probabilities. When we insert TEP-FFs to such FFs, inserted TEP-FFs can monitor a set of gates which maximumly contribute to causing timing errors. Note that TEP-FFs help to prevent timing errors from occurring due to the delay variation of monitoring gates. Consequently, the proposed methodology maximumly reduces the timing failure probability and thus maximumly extends the MTTF. This work formulates this FF selection problem as an ILP problem to derive the exact solution. The ILP formulation is as follows:

- Input
  - one ASA circuit
- Output
  - one AVS circuit
- Objective
  - Maximize : $\sum_{k=1}^{N_{\text{inst}}} (P_{\text{fail\_inst}_k} \times B_{\text{inst}_k})$
- Constraints
  - $0 \leq B_{\text{inst}_k} \leq 1 \quad (1 \leq k \leq N_{\text{inst}})$
  - $0 \leq B_{\text{TEP}_i} \leq 1 \quad (1 \leq i \leq N_{\text{FF}})$
  - $\sum_{i=1}^{N_{\text{FF}}} B_{\text{TEP}_i} \leq N_{\text{TEP}}^{\max}$
  - $B_{\text{inst}_k} \leq \sum_{i=1}^{N_{\text{FF}}} (B_{\text{TEP}_i} \times B_{\text{FF}_i\_\text{inst}_k})$

- Variables
  - $B_{\mathrm{TEP}_i} \quad (1 \leq i \leq N_{\mathrm{FF}})$

The input of this problem is the ASA circuit designed and selected referring to Chapter 3 and the output is one AVS circuit. The number of instances in the circuit is N_{inst}. The objective of this ILP problem is to maximize the sum of $(P_{\mathrm{fail\_inst}_k} \times B_{\mathrm{inst}_k})$. Remind that $P_{\mathrm{fail\_inst}_k}$ is the gate-wise failure probability of k-th instance and B_{inst_k} is a binary variable which becomes 1 when k-th instance is included in paths ending at target FFs for TEP-FF insertion. As similar to Section 3.5.2, the sum of $P_{\mathrm{fail\_inst}_k} \times B_{\mathrm{inst}_k}$ represents the gate-wise failure probability reduction. In this problem, binary variables B_{TEP_i} are assigned, where B_{TEP_i} becomes 1 when i-th FF is selected to target FFs for TEP-FF insertion.

The first and second constraints are given to restrict B_{inst_k} and B_{TEP_i} to binary numbers. The third constraint means that the number of target FFs for TEP-FF insertion should be equal to or less than $N_{\mathrm{TEP}}^{\mathrm{max}}$ and this constrains the area overhead due to TEP-FF insertion. The fourth constraint defines the relation between B_{inst_k} and B_{TEP_i} which has a similarity with the fourth constraint in Section 3.5.2.

4.4 Experimental Evaluation

This section experimentally evaluates the performance improvement from the conventional WC design to the proposed AVS. Section 4.4.1 explains the evaluation setup. Section 4.4.2 shows the performance improvement results regarding power saving effects and demonstrates that the proposed AVS extends the MTTF from the conventional TEP-FF implementation which inserts TEP-FF with ascending order of FF setup slack.

4.4.1 Experimental Setup

This work used the AES circuit and OR1200 OpenRISC processor as target circuits. These two circuits were designed by a commercial P&R tool [97] with a 45 nm Nangate standard cell library [77]. The minimum clock period of post-layout circuits at 1.20 V in the typical PVTA conditions and the worst-case are 3,150 ps and 4,260 ps in OpenRISC and 370 ps and 480 ps in AES, respectively. Hereafter, the target clock period is set to 4,260 ps in OpenRISC and 480 ps in AES, and then ASA optimizes the timing slack of FF/path for these target clock periods.

This work used Gurobi Optimizer 7.0 [94] to solve the ILP problem defined in Section 4.3. The solver was executed on a 2.4 GHz Xeon CPU machine under the Red Hat Enterprise Linux 6 operating system with 1 TB memory. The required CPU times were at most 7.13 seconds in AES and 1.46 seconds in OpenRISC. The evaluation took into account the following variations similarly with Chapters 2 and 3.

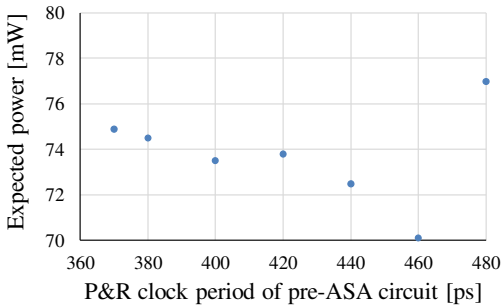

Figure 4.3: Expected minimum power after ASA in AES.

- Dynamic supply noise, which is assumed to fluctuate between -50 mV and 50 mV by 10 mV with eleven steps.
- Manufacturing variability, which is assumed to consist of the intra-die random variation and inter-die variation. Both the intra-die random variation and inter-die variation include NMOS and PMOS threshold voltage variation of σ = 30 mV and gate length variation of σ = 2 nm.
- NBTI aging model with six degradation states of 0 mV, 0.5 mV, 1 mV, 5 mV, 10 mV and 15 mV are prepared. The necessary information is obtained similarly with Section 2.4.1.

As for workload, this work selected three benchmark programs (crc, sha, and dijkstra) from MIBenchmark [83] for OpenRISC and 1,000 random test patterns for AES, which are similar to Sections 2.4.1 and 3.6.1.

This work set the MTTF of 1.00×10^{17} cycles, i.e., 10 years in OpenRISC and 1.6 years in AES, as $\mathrm{MTTF}_{\mathrm{min}}$. This work prepared seven supply voltages, i.e., from 1.20 V to 0.90 V with 50 mV interval, and swept clock period from 450 ps to 500 ps in AES and from 4,000 ps to 6,000 ps in OpenRISC. Note that, at each clock period, AVS dynamically adjusted the supply voltage within the range from 1.20 V to 0.90 V.

With this setup, this work performed ASA to both AES and OpenRISC. The number of pre-ASA candidate circuits was seven in AES, where P&R clock periods of these pre-ASA circuits were 370, 380, 400, 420, 440, 460, and 480 ps. As for OpenRISC, seven candidates with 3,150, 3,200, 3,400, 3,600, 3,800, 4,000, and 4,200 ps were given. Fig. 4.3 shows the estimation results of the expected minimum power after ASA for each pre-ASA candidate in AES. From Fig. 4.3, we can see that the pre-ASA candidate designed at 460 ps is the most promising one regarding power. Then the pre-ASA circuit that was laid out at 460 ps were selected. Similar to AES, we evaluated the expected minimum power of OpenRISC candidate circuits and selected the circuit laid out at 4,000 ps. Next, ASA was performed to the chosen candidates. The constraints for the overhead of the area and the number of low-V_{th} cells by ASA were set to 0.7% and 0.0% for OpenRISC and the maximum number of target FFs for ASA, i.e., $N_{\mathrm{ASA}}^{\mathrm{max}}$ was set to

255. Similarly, $N_{\mathrm{ASA}}^{\mathrm{max}}$ is set to 255 in AES. Note that in AES, the ASA circuit achieved the 6.2% area reduction compared with the pre-ASA circuit design at 370 ps.

Next, several TEP-FFs were inserted to the voltage-scaled circuits. The constraint of area overhead for TEP-FF was set to 0.8% for both AES and OpenRISC, and the number of maximum TEP-FF ($N_{\mathrm{TEP}}^{\mathrm{max}}$) was set to 5 in AES and 20 in OpenRISC, respectively. When inserting TEP-FF, we need to determine the number of delay buffers for each TEP-FF. This work inserted the delay buffers whose delay were comparable to the delay variation caused by 50 mV supply noise, where this number of 50 mV corresponds to one level decrement of the supply voltage.

The MTTF and average supply voltage under PVTA variation were evaluated by the stochastic MTTF estimation framework [61]. In the experiment, the monitor period for AVS was swept from 10^6 to 10^{13} clock cycles.

4.4.2 Evaluation Results

This section first shows power savings thanks to the proposed AVS, and then examines the effectiveness of the TEP-FF insertion methodology.

Power Saving Effects

Fig. 4.4 shows trade-offs between the minimum average power and the clock cycle under the MTTF constraint of 10^{17} cycles, where (a) in OpenRISC and (b) in AES, respectively. The black square plots represent the conventional WC design with guard-banding for PVTA variation. The yellow triangular and blue cross plots correspond to the conventional AVS which optimizes only the sensing circuit, and the proposed AVS which optimizes both the main logic under AVS and sensing circuit, respectively. Here, the TEP-FFs in the conventional AVS were inserted by the method in Section 4.3. This section examines the evaluation results from the following two aspects: (1) overall power saving effect thanks to the proposed AVS, and (2) difference of the power dissipation between the proposed and conventional AVS.

First, the black square and blue cross plots are compared to clarify the overall performance improvement thanks to the proposed AVS. Fig. 4.4 shows that the proposed AVS saves average power while keeping the target MTTF. For example, in Fig. 4.4(a), at a clock period of 4,260 ps, the proposed AVS achieved the target MTTF with an average power of 13.4 mW, whereas the conventional WC design required 21.6 mW. In other words, AVS achieved 38.0% power savings from 21.6 mW to 13.4 mW. Similarly, in Fig. 4.4(b), at a clock period of 480 ps, the proposed AVS achieved 22.6% power savings from 183.0 mW to 141.5 mW. It is experimentally confirmed that the proposed AVS made the significant power savings both in AES and OpenRISC. The proposed AVS increases the circuit area by 1.5% in OpenRISC and decreases the area by 5.4% in AES.

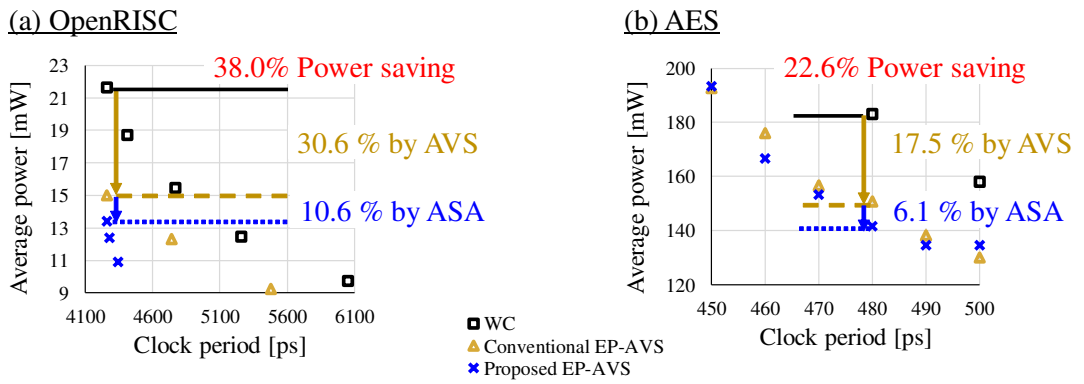

Figure 4.4: Trade-offs relation between clock period and average power. (a) OpenRISC, (b) AES.

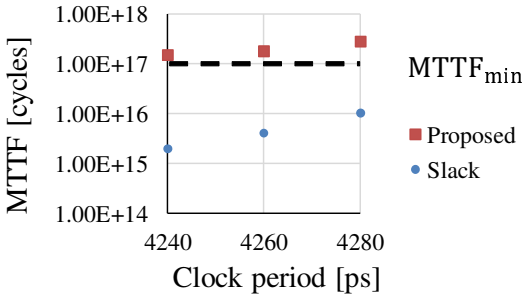

Figure 4.5: MTTF comparison in OpenRISC.

Next, the conventional AVS and proposed AVS, i.e. yellow triangular and blue cross plots, are compared. Fig. 4.4 shows that the proposed AVS further improves power dissipation from the conventional AVS. For example, the proposed AVS achieved 10.6% power savings from 15.0 mW to 13.4 mW at a clock period of 4,260 ps in OpenRISC and 6.1% power savings from 150.8 mW to 141.5 mW at a clock period of 480 ps in AES. These power savings reveal that the ASA for the main logic works well and the simultaneous optimization of the main logic under AVS and the sensing circuit enhance the efficacy of AVS. It is also observed that the performance improvement thanks to ASA is the largest around the target clock periods of 4,260 ps in OpenRISC and 480 ps in AES and it becomes smaller as the period goes away from the target one since ASA optimized the circuit at the target clock period under the MTTF constraint, which was also reported in Chapter 3. There could be room for improvement at different clock periods.

Effectiveness of the Proposed TEP-FF Insertion

Next, let us evaluate the effectiveness of the proposed TEP-FF insertion methodology that takes into account the failure probabilities of individual FFs. For comparison, we also evaluate the performance of the conventional TEP-FF insertion method, e.g., [35, 42, 57], that selects the insertion locations according to the order of FF setup slacks. Fig. 4.5 shows the comparison results in OpenRISC. We can see that the proposed method achieved much longer MTTF than the conventional slack-based TEP-FF insertion method. To make things worse, the conventional method cannot satisfy the given MTTF constraint at all.

4.5 Conclusion

This chapter focused on AVS and proposed a design methodology for AVS circuits. The proposed design methodology optimizes both the main logic under AVS and sensing circuits taking into account the timing failure probabilities of FFs. The quantitative MTTF and power evaluation results showed that the proposed AVS design methodology achieved 38.0% power saving while satisfying the target MTTF thanks to the ASA and failure probability based TEP-FF insertion.

Chapter 5

Performance Evaluation of Error Detection Mechanisms for Supply Noise induced Timing Errors

In previous chapters, the design methodology including the main logic optimization under AVS and sensor insertion is discussed. This chapter examines the post-silicon validation methodology for the fabricated AVS circuit [98]. In this chapter, we devise the software-based EDM-L for facilitating the post-silicon validation of the AVS circuit and evaluate the performance of EDM-L for timing error localization with a noise-aware logic simulator and 65 nm test chips assuming the following two EDM-L usage scenarios: (1) localizing a timing error that occurred in the original program and (2) localizing as many potential timing errors as possible.

5.1 Introduction

For reliable AVS circuits, not only the design methodology but the post-silicon validation methodology are essential. In the post-silicon validation of the AVS circuit, designers need to verify whether the AVS circuit can sustain correct operation by keeping appropriate timing margins at various operating conditions. As mentioned in Chapter 1, in AVS operation, the fast supply noise within a clock cycle may induce timing errors since the error prediction sensors may not have a capability of monitoring such a fast delay fluctuation. As for the delay fluctuation like the slow temperature gradation the sensors can observe the timing slack of the circuit in time and thus the circuit may prevent error occurrence. From this point of view, supply noise induced timing errors may become one of the most serious electrical timing errors for the post-silicon validation of AVS circuit. Based on this consideration, this chapter focuses on the supply noise induced timing error.

To localize supply noise induced timing error, this chapter devises the software-

based EDM-L and evaluates the performance of EDM-L for the error localization. Firstly, this work presents a case study that considers program-dependent supply noise with a supply noise-aware simulation framework. Next, this work evaluates and reports the EDM-L performance for supply noise induced timing error localization using 65 nm test chips. In addition, the reason of the inconsistency between the measurement and simulation results is investigated and then two possible reasons are pointed out: (1) the design of power distribution network (PDN), i.e., the magnitude of dynamic power supply noise, and (2) the definition of timing error occurrence frequency. By updating the simulation setup, this work confirms that the measurement and simulation results are well correlated.

This work considers two scenarios of EDM-L usage in post-silicon validation: (1) localizing the exact supply noise induced timing error that occurred in the original program, and (2) localizing as many potential errors as possible which could lead to abnormal behaviors. For the first scenario, two necessary conditions must be satisfied: error reproducibility, and diversity between the executions of the duplicated blocks. For the second scenario, only the diversity must be satisfied. This chapter discusses the utility of EDM-L for supply noise induced timing error localization in these two scenarios on the basis of experimental results.

The rest of this chapter is organized as follows. Section 5.2 first explains conventional EDM and our EDM-L transformation. Then Section 5.2 examines the necessary conditions in which EDM-L localizes a supply noise induced timing error. Section 5.3 presents a case study that investigates whether EDM-L transformation is helpful to localize the supply noise induced timing errors with a supply noise-aware simulation framework. Section 5.4 presents the performance evaluation of EDM with fabricated test chips, and Section 5.5 examines the experimental results and discusses the consistency between the measurement and simulation results. Lastly, concluding remarks are given in Section 5.6.

5.2 Localizing Supply Noise Induced Timing Error with EDM

This section explains EDM transformations, discusses two scenarios of EDM usage in post-silicon validation, and describes the necessary conditions for error localization in each scenario.

5.2.1 EDM Transformation

To detect an error quickly after its occurrence, EDM converts an input program to a special program using several transformation techniques. The EDM transformation and error detection described in [74] are exemplified in Fig. 5.1, where the original EDM

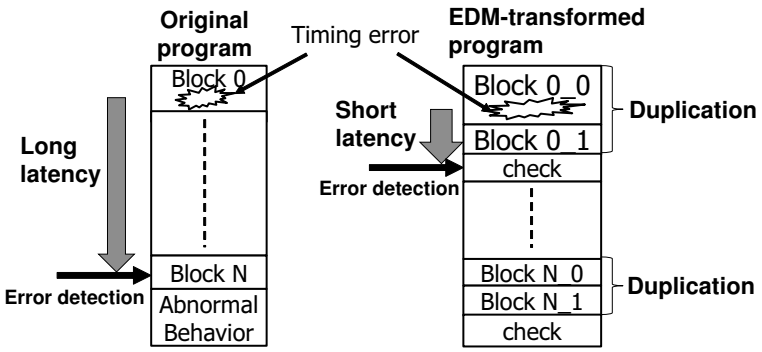

Figure 5.1: Error detection by EDM transformation.

transformation in [74] is hereafter called EDM-O. The transformation is performed at C/C++ level. Fig. 5.2 gives an example of EDM-O-transformed code. First, EDM-O divides an input program into blocks, where each block consists of a set of operations in series. In the EDM-O block generation, a new block starts when a branch operation or variable read operation is found [74]. Here, variable read operations include those that read values stored in a register or stored in a memory. In the example code in Fig. 5.2, operations "a = b;" and "c = b;" are variable read operations since the value of variable b stored in a register or a memory is read. Operation "b =1;" is not treated as a variable read operation since the constant value of 1 often comes from the immediate field of processor instructions. Second, EDM-O duplicates each block. The paired original and duplicated blocks are aligned in sequence. In the example code shown in Fig. 5.2, operation "a = b;" is duplicated to "a0 = b0;" and "a1 = b1;", and operations "b = 1;" and "c = b;" are duplicated similarly. Third, for all the pairs of the original and duplicated blocks, EDM-O inserts check operations to compare the read values, i.e., in Fig. 5.2, the values stored in b0 and b1 are compared after "a1 = b1;" and "c1 = b1;". Consequently, the EDM-O-transformed program executes the original block, the duplicated block and the check operation in sequence for all the pairs of the original and duplicated blocks, where in some cases a branch operation is sandwiched between the check operation and the next original block.

EDM-O is originally developed for the purpose of improving soft-error detection coverage [74], and hence the check is constantly inserted after each variable read to know whether bit flips occurred in the memory, registers, or FFs. EDM-O is useful for soft error detection, but it can be improved for shortening error detection latency of supply noise induced timing errors, i.e., the elapsed time between the occurrence of supply noise induced timing error and its detection. The supply noise induced timing error arises as a write fail to memory, registers or FFs, and once a correct value is stored the value will not be corrupted by supply noise induced timing errors. This means we can know whether the supply noise induced timing error occurred or not immediately after the write operation. This is the main difference from the soft error. Therefore, to

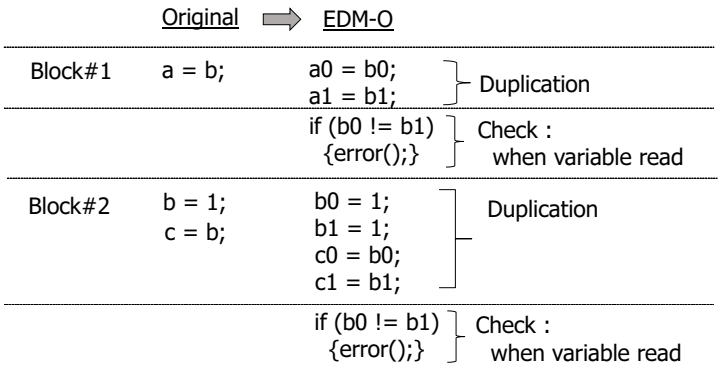

Figure 5.2: An example of EDM-O code.

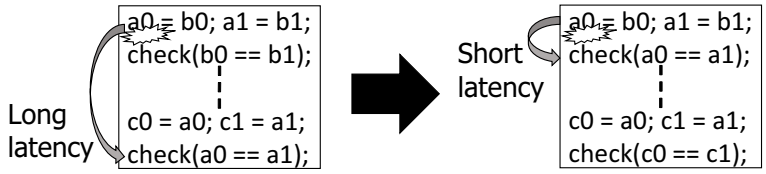

Figure 5.3: Difference of error detection latency between EDM-O (left) and EDM-L (right).

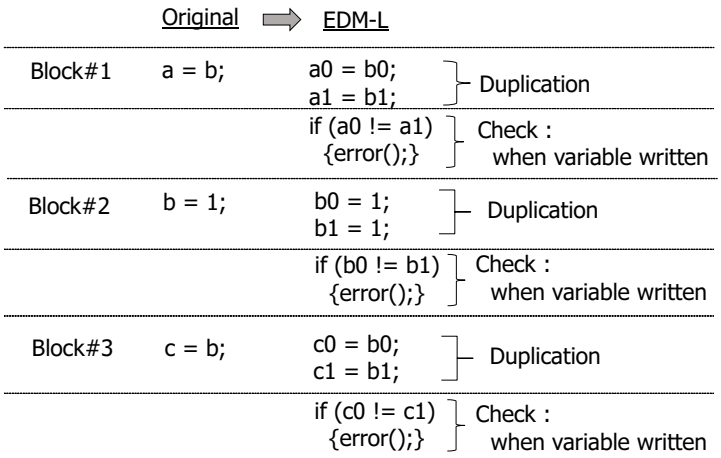

Figure 5.4: An example of EDM-L code.

use EDM for quickly detecting supply noise induced timing errors, EDM should check the values in the memory, register or FFs after they are written. For this purpose, check operations should be performed immediately after variables are written, not read. The left figure of Fig. 5.3 illustrates such an example. Suppose the memory write of variable $a0$ at the first line failed. In this case, the memory of $a0$ is not accessed for a long

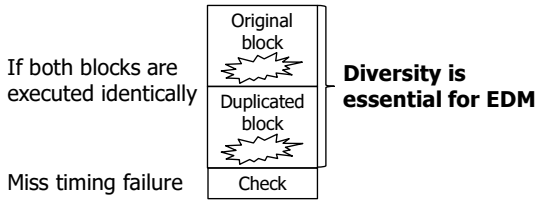

Figure 5.5: Diversity is necessary to satisfy detectability.

time and the inserted check is performed after a long time elapses. To shorten the error latency, the check is performed immediately after the memory/register/FF write access (right figure of Fig. 5.3). Motivated by this, this chapter devised EDM for short Latency (EDM-L). Fig. 5.4 shows an example of EDM-L-transformed code. EDM-L inserts check operations for every variable write. Note that EDM-L performs check operation after "$b1$ = 1;" whereas EDM-O does not check after this operation, which means EDM-L can quickly check whether timing error occurred in $b0$ or $b1$ compared to EDM-O. Consequently, when an error occurs in the original block, we can expect that the next check operation detects the error occurrence.

Furthermore, to satisfy detectability, the diversity between the original block and the duplicated block is crucially important. If the original block and the duplicated block are identical, the same error would occur in both the blocks and the check operation fails to detect the error as illustrated in Fig. 5.5. In the EDM transformation, the original blocks and the duplicated blocks often split the memory space to gain the diversity, where the different memory addresses are expected to have different access times. EDM can include various transformations to maximize the diversity.

5.2.2 EDM Usage Scenarios and Necessary Conditions for Error Detection

This section lists two EDM usage scenarios in post-silicon validation and discusses the necessary conditions that EDM needs to satisfy in each scenario.

This work considers the following two scenarios.

Scenario1:
: When an original program was running, the supply noise induced timing error occurred. We want to localize this error using EDM transformation.

Scenario2:
: We want to localize as many potential bugs as possible.

This section first examines the necessary conditions for the first scenario. In Scenario1, EDM should satisfy the two conditions below simultaneously (Fig. 5.6).

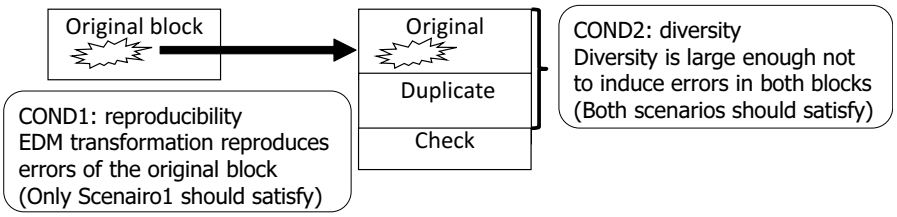

Figure 5.6: Two conditions for EDM to localize supply noise induced timing error in Scenario1.

COND1:

EDM-transformed program reproduces the error which occurred in the original input program.

COND2:

EDM gives enough diversity so that the paired original and duplicated blocks output different computational results.

The first COND1 condition is necessary to investigate the root cause of the error observed in the original program. To reproduce the error occurrence, the EDM-transformed program should maintain the similar behavior of the original program. If EDM does not reproduce the same error, the error localization of the original program is impossible.

The second COND2 is the fundamental condition for EDM to work. If the original and duplicated blocks output the same wrong values, the inserted check operation misses the error. Focusing on the second COND2 condition, dynamically fluctuating factors, such as supply noise, might help increase the diversity. The diversity originates from the timing characteristics of the fabricated chip under test and dynamically fluctuating factors. For example, power supply noise varies depending on the running program, which may improve the diversity.

On the other hand, COND1 is thought to become more difficult to satisfy as dynamically fluctuating factors become more significant. The supply noise, for example, of the chip on which the original program is running can be different from the noise of the EDM-transformed program. In this case, the error observed in the original program may disappear in the EDM-transformed program, and a new error may arise at another program location.

As stated above, Scenario1 requires that both COND1 and COND2 are satisfied. However, previous studies did not focus on COND1. It is not clear whether or how often COND1 can be satisfied in the EDM-transformed programs. In addition, it is not clear whether EDM satisfies COND2 for realistic supply noise induced timing error. The next section experimentally investigates whether these two conditions are satisfied under dynamic power supply noise with a noise-aware logic simulation framework.

In Scenario2, the error observed in the original program does not need to be reproduced. Moreover, inducing a new error could be preferable since potential bugs could be localized. Therefore, COND1 is not necessary. Only COND2 needs to be satisfied. The necessary condition for Scenario2 is the subset of the condition for Scenario1 and hence the experiments for Scenario1 are valid for Scenario2 as well.

5.3 Simulation-based Evaluation of EDM Transformation

This section experimentally investigates whether EDM transformation works well in Scenario1 and Scenario2. The experiment supposes that dynamic power supply noise is the primary source of electrical timing bugs, and it accurately reproduces the impact of EDM on the dynamic supply noise and the consequent timing variations.

5.3.1 Experimental Setup

Our experimental evaluation was performed for an industrial embedded processor (Toshiba MeP processor). This processor was synthesized and laid out with an industrial 65nm library. In this experiment, the post-layout design was used for the simulations which will be explained later. This experiment took three C-language benchmark programs, dijkstra, crc, and sha from MiBenchmark [83] as original input programs. Although these three programs were selected as typical workloads for the MeP processor, EDM transformation can be applied to other various types of programs similarly as previously explained in Section 1.2.3. This work implemented an EDM translator and used this translator to get EDM-transformed programs.

In EDM transformation, two types of check operations are inserted [74]: (1) data check and (2) code flow check. For the data checking, each variable v is duplicated as $v0$ and $v1$. Then, the consistency check is performed every time $v0$ ($v1$) is read in EDM-O. In EDM-L, the data check is performed every time $v0$ ($v1$) is written. The code flow check aims to detect an illegal change of the code execution flow, such as an illegal jump operation. The code flow check is inserted as follows.

First, EDM identifies all the basic blocks, i.e., branch-free sequences, in the program and checks whether all the statements in every basic block are executed in sequence by numbering the basic blocks. Second, checks for every test statement (e.g., if, else if, while) are inserted. EDM inserts the opposite test to both the true and false clauses to detect an illegal execution flow. The last target is call and return operation. Every procedure, i.e., function in the program, is associated with its unique number, and the number is checked for every call of the procedure. In this section, we duplicated all variables and inserted all data checks and code flow checks. We call this transformation as full-EDM.

Table 5.1: Impact of EDM-L transformation on cycle time and cache miss.

	execution cycles		inst. cache misses		data cache misses	
	orig.	EDM	orig.	EDM	orig.	EDM
dijkstra	24512	69838	45	161	11	20
	(1.00)	(2.85)	(1.00)	(3.58)	(1.00)	(1.82)
sha	30757	97831	44	167	25	42
	(1.00)	(3.18)	(1.00)	(3.80)	(1.00)	(1.68)
crc	19975	57252	9	29	35	71
	(1.00)	(2.87)	(1.00)	(3.22)	(1.00)	(2.03)

A value in parentheses is the ratio of full-EDM-L divided by original.

Table 5.2: Impact of EDM-O transformation on cycle time and cache miss.

	execution cycles		inst. cache misses		data cache misses	
	orig.	EDM	orig.	EDM	orig.	EDM
dijkstra	24512	65693	45	150	11	20
	(1.00)	(2.68)	(1.00)	(3.33)	(1.00)	(1.82)
sha	30757	120487	44	213	25	52
	(1.00)	(3.92)	(1.00)	(4.84)	(1.00)	(2.08)
crc	19975	60000	9	23	35	65
	(1.00)	(3.00)	(1.00)	(2.56)	(1.00)	(1.86)

A value in parentheses is the ratio of full-EDM-O divided by original.

For the original and duplicated blocks, the same input data was stored at two different addresses of data memory, and each block accessed its own data in the data memory. Tables 5.1 and 5.2 list increases in the number of execution cycles and the number of cache misses by full-EDM-L and full-EDM-O, respectively. The number of execution cycles increased three to four times, and the increase in the number of instruction cache misses was similar. The increase in the data cache miss was roughly double, which is consistent with a fact that the data size is doubled in the full-EDM-transformed program.

This work evaluated and compared the error occurrences in the original and EDM-transformed programs by logic simulation. Our logic simulation framework concurrently simulates two MeP designs: one is at register transfer (RT) level and the other is at gate level. The RT-level logic simulation is performed with the zero-delay model, and hence the output is always correct disregarding the clock frequency and the given supply voltage. On the other hand, the gate-level logic simulation includes timing information and then may output wrong values. In this work, a noise-aware logic simulation

method [99] is adopted to take explicitly into consideration program-dependent dynamic supply noise. This simulation method will be explained in the next subsection. Once an inconsistency is detected at a FF between RT-level and gate-level simulations, we can immediately notice a timing error occurrence. Thanks to this, we can know the exact location of timing error in time and space.

The comparisons of error occurrence between the original and full-EDM-L programs were performed for the following 300 situations. Due to manufacturing variability, each chip has different delay characteristics. To reproduce this, 10 MeP chips were hypothetically fabricated by Monte-Carlo method assuming that each instance delay randomly fluctuated with the standard deviation of 25% of the typical instance delay. In addition, depending on the final products, the LSI package may change. This work assumed 10 package conditions, i.e., 10 sets of equivalent circuit parameters of the power distribution network. The equivalent circuit model will be shown in the next subsection. In summary, $100 = 10 \times 10$ samples were evaluated for each program, i.e., 300 samples in total. Similarly, the full-EDM-O program was evaluated in 300 situations.

This work focused on the first error that occurred in the program execution, and its location was considered to check whether COND1 was satisfied. The minimum clock cycle that caused timing errors was searched with 2 ps interval. When we decomposed the program into blocks, we numbered the blocks from the beginning. This work regarded the difference between the block numbers as the proximity of error occurrence locations. When the difference is zero, the timing error is reproduced at the same block in the EDM-transformed program and COND1 is satisfied. COND2 was evaluated by checking whether the program was terminated by the check operation. Even if a timing error occurred in the EDM-transformed programs, the check operation sometimes misses the error. This case can be categorized into two groups: silent error and masked error. In the silent error case, the execution result is different from the correct result. In the case of the masked error, the execution result is correct.

5.3.2 Noise-aware Logic Simulation

This work used a noise-aware logic simulation method that could consider dynamic power supply noise in gate-level logic simulation [99]. The dependence of gate delay on supply voltage was first evaluated with HSPICE [100] and it is expressed using a delay element whose delay is controlled by digital signals representing the supply voltage. Here, this delay element is described at RTL. By attaching this delay element to every gate and dynamically changing the digital signal that represents supply voltage, we can reproduce voltage-dependent gate delay in logic simulation.

When performing the above noise-aware logic simulation, we need to give a waveform of dynamic power supply noise. This work prepared noise waveform information with the following two steps. First, this work simulated the post-layout MeP design with the original and EDM-transformed programs by a transistor-level circuit simula-

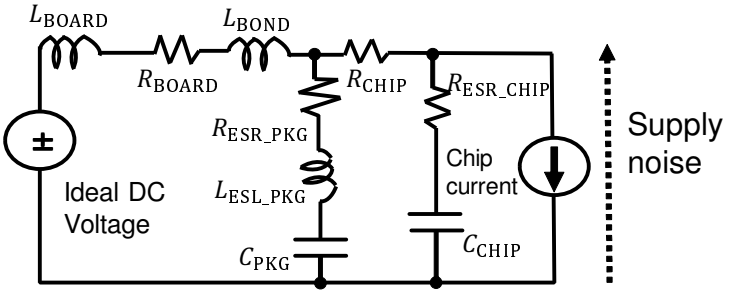

Figure 5.7: An equivalent circuit of power distribution network.

tor [101], and obtained waveforms of the current consumed by MeP for each program. Here, it should be noted that the transistor-level simulation to obtain the current waveform is very time consuming and it took three days for sha-full-EDM-O program. To make the CPU time needed for the entire evaluation in this work acceptable, the two-step procedure was adopted. Next, this work gave this current waveform to the equivalent circuit of Fig. 5.7 and obtained the waveform of dynamic power supply noise. The nominal supply voltage was 1.0 V. To reproduce various package assemblies and obtain corresponding noise waveforms, this work used 10 sets of power distribution network (PDN) parameters in Fig. 5.7. The parameter setting is explained in the following.

This work varied three parameters of C_{PKG}, R_{ESR_PKG} and L_{ESL_PKG} representing the package capacitor, and one parameter of C_{CHIP} representing the on-chip capacitor. The other five parameters were fixed as follows: L_{BOARD}=0.1 nH, R_{BOARD}=5 mΩ, L_{BOND}=0.3 nH, R_{CHIP}=0.1 Ω and R_{ESR_CHIP}=0.3 Ω. This work prepared five configurations of the package capacitor: (1) no package capacitor, (2) one NPO capacitor, (3) one X7R capacitor, (4) ten NPO capacitors in parallel and (5) ten X7R capacitors in parallel, where NPO and X7R are commercially available popular ceramic capacitors [102]. C_{PKG}, R_{ESR_PKG} and L_{ESL_PKG} of NPO and X7R are (100 pF, 0.3 Ω, 0.6 nH) and (1 nF, 0.6 Ω, 0.6 nH), respectively [102, 103]. As for the on-chip capacitance C_{CHIP}, two values of 3.5 nF and 0.3 nF were prepared. Consequently, 10 (=5×2) sets of PDN parameters were prepared and used to obtain the noise waveforms. Examples of the noise waveforms are shown in Figs. 5.8 and 5.9. These noise waveforms were given to the noise-aware logic simulation.

5.3.3 Evaluation Results

Fig. 5.10 shows how many samples satisfied COND1 of reproducibility and COND2 of detectability. For every timing error in the original program, this work checked if COND1 and COND2 are satisfied in the EDM-transformed program. Among 600 timing error samples, we could not find a sample that satisfied COND1 and COND2 simultaneously, which suggests EDM is less helpful in Scenario1. In addition, over 75% of errors satisfied neither COND1 nor COND2. Comparing full-EDM-L with full-EDM-

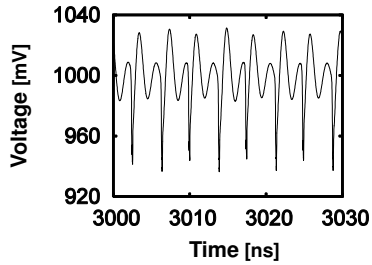

Figure 5.8: A waveform example of inductive fluctuation.

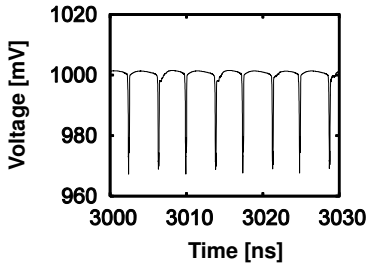

Figure 5.9: A waveform example of resistive drop.

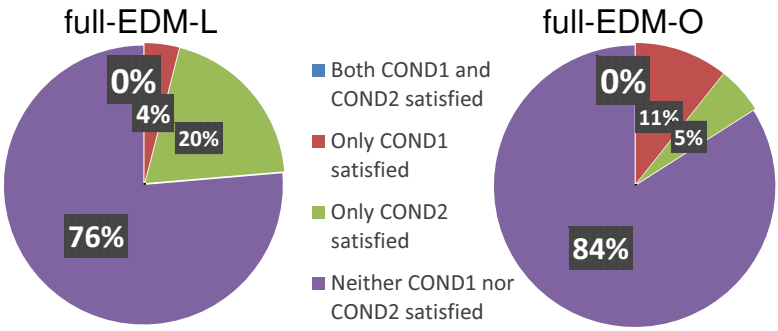

Figure 5.10: Evaluation results of full-EDM.

O, we can see the difference in the proportion that only COND1/COND2 is satisfied. Following sections examine the results for COND1 and COND2 separately in detail.

COND1

Figs. 5.11 and 5.12 show the proximity of the error occurrence in the original and full-EDM dijkstra, crc, and sha programs. Remind that the proximity is defined as the difference of the block numbers of the error occurrence. The block number difference of zero means that the same error is observed in the original and EDM-transformed programs. In EDM-L, 10% of errors in crc and 2% of errors in dijkstra were reproduced. In sha, no errors were reproduced. In EDM-O, over 30% of errors were reproduced in crc, but no errors were reproduced in dijkstra and sha. As a whole, EDM-L and EDM-O reproduced only 4% and 11% errors, respectively. Such low reproduction ratios are mainly due to the following two reasons.

The first reason is that EDM changes supply voltage noise since the block duplication and check insertion change the instruction sequence and the usage of circuit blocks, such as memory and general purpose registers. In other words, even when the same instructions are performed, the supply noise could change, because the used registers and

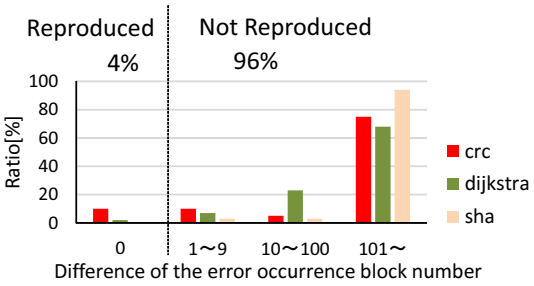

Figure 5.11: COND1: difference of block numbers of first error occurrence between original and full-EDM-L programs. For each program, the number of samples is 100.

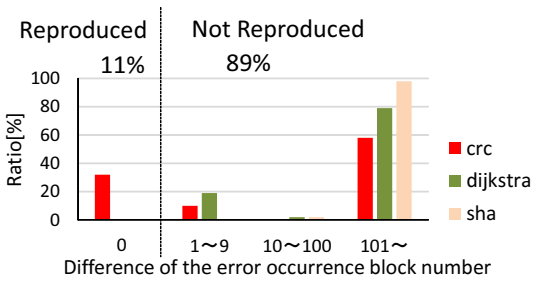

Figure 5.12: COND1: difference of block numbers of first error occurrence between original and full-EDM-O programs. For each program, the number of samples is 100.

memory addresses are different and the inductive noises excited in the previous clock cycles are superposed. Fig. 5.13 shows a comparison of noise waveforms between the original and full-EDM dijkstra programs, where the same instruction was performed in this clock cycle. We can see that the voltage waveforms are not identical. For further investigation, this work evaluated the minimum supply voltage within a clock cycle every time mov instruction was performed. Fig. 5.14 shows a histogram of the minimum voltage in the original crc program. We can see that the minimum voltage value ranges from 941 mV to 947 mV even though the same instruction of mov is performed. This waveform difference prevents the error reproduction.

The second reason is that EDM lengthens the program execution as previously shown in Tables 5.1 and 5.2. As the program becomes longer, a new timing error, which is different from the error observed in the original program, is more likely to occur. In addition, the duplication and frequent check insertion change the instruction component of the program. Fig. 5.15 shows the ratios of the instructions executed in sha-original and sha-full-EDM-L programs, respectively. We can see that instruction ratio of the EDM and original programs are considerably different. For example, in EDM, the number of load word (lw) instruction increases because the used memory space is doubled, and a number of branch if equal (beq) instructions are introduced due to check insertion. These instruction variations not only affect the processor behavior but also enlarge the noise difference, which makes the error reproduction difficult.

COND2

Next, COND2 is examined. Figs. 5.16 and 5.17 show the proportions of silent errors, masked errors and detected errors. For detected errors, the histogram of the error detection latency is presented. In EDM-L, we can see that 77% of errors are masked and 2% are silent errors, whereas 87% are masked errors and 7% are silent errors in EDM-O.

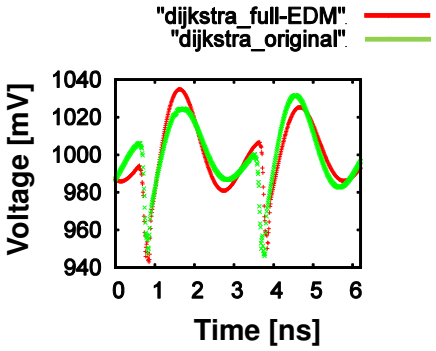

Figure 5.13: Voltage inconsistency between original and full-EDM programs.

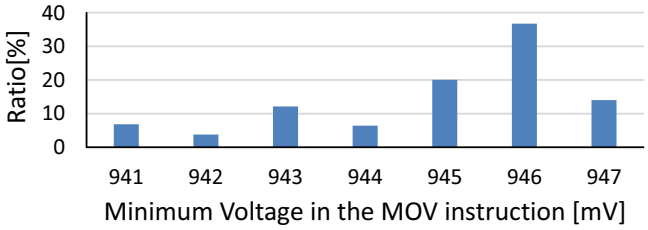

Figure 5.14: The histogram of the supply voltage when MOV instructions were executed (crc-original).

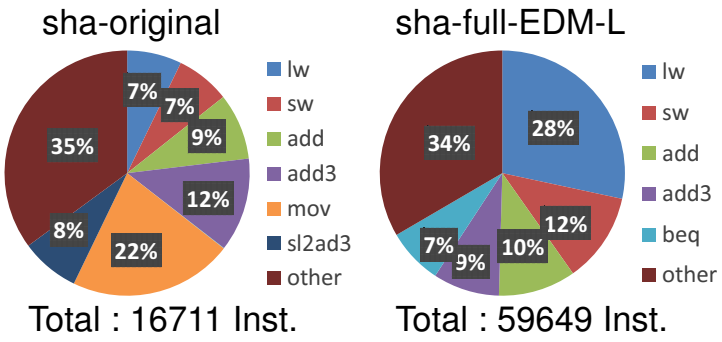

Figure 5.15: Proportion of executed instructions in sha-original, sha-full-EDM-L.

In other words, most of the supply noise induced timing errors did not propagate to the memory and general purpose registers.

Among the non-masked errors, 86% errors were detected within 1000 cycles in EDM-L, while 38% in EDM-O. This result indicates that the EDM-L performance of detecting supply noise induced timing errors that affect execution results is high. Therefore, this work experimentally confirmed that EDM-L is helpful to detect noise induced errors with shorter error detection latency. In other words, we can use EDM-L in Sce-

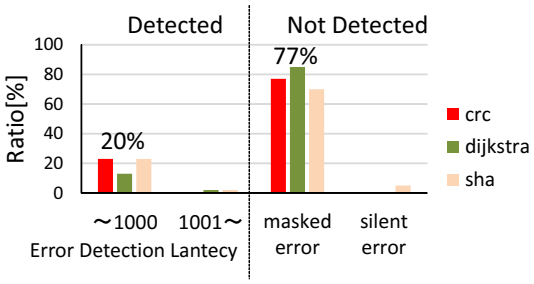

Figure 5.16: COND2 : Error classification in full-EDM-L. For each program, the number of samples is 100.

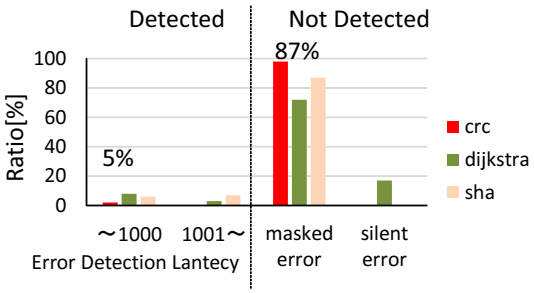

Figure 5.17: COND2 : Error classification in full-EDM-O. For each program, the number of samples is 100.

nario2. For the errors having long error detection latency, we found a tendency that the first error was not detected and the second or later error was detected by the check operation. As for silent errors, one possible reason is that identical timing errors occur both in the original block and the duplicated block, and the check operation fails to detect them, which was previously explained with Fig. 5.5. On the other hand, the EDM-O performance was not good. The proportion of silent errors was larger, and the detection latency was longer. Clearly, for the purpose of quick error detection in post-silicon validation, EDM-L is much better than EDM-O.

5.4 Hardware Measurement

This section experimentally investigates whether EDM transformation works well in Scenario1 and Scenario2 with 65 nm test chips, and compares theses results with previous simulation.

5.4.1 Measurement Setup

First, this section explains the experimental setup. This work used a 32-bit embedded processor (Toshiba MeP processor) implemented and fabricated in 65 nm CMOS technology. A chip photo is shown in Fig. 5.18. The chip size is 4.2 mm $\times$ 2.1mm.

This work took three C-language benchmark programs, dijkstra, crc, and sha from MiBenchmark [83] as similar to Section 5.3. Fig. 5.19 shows the measurement setup consisting of a test chip, a device under test (DUT) board, a DC voltage source and a PC. The packaged test chip is mounted on a DUT board. The DUT board, which also includes a Stratix Ⅲ FPGA and SDRAM, is used as a logic analyzer and a pattern generator. For example, the data that should be stored in the instruction memory and the data memory of MeP processor is first transferred from PC to the DUT board through USB cable, and then the data is loaded to the on-chip SRAMs. Also, after the program

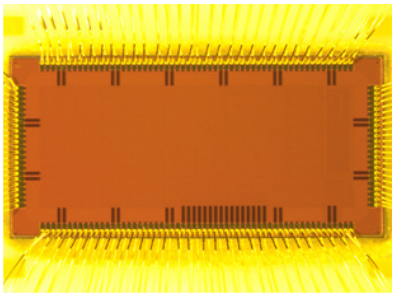

Figure 5.18: A photo of 65 nm test chip of MeP processor. Die size is 4.2 mm × 2.1mm.

execution of MeP processor, the data in the on-chip data memory is downloaded to PC through the DUT board. This work also used an external DC source (Agilent6611C) to supply the voltage to the test chip.

With this setup, we can obtain the Shmoo plot taking the following procedure. In each measurement, this work set the clock frequency and supply voltage given to the test chip. Then, the data uploading, the program execution and the data downloading are executed as explained before. When the downloaded data is identical to the expected data, the program execution is thought to be correct. When there is an inconsistency, it is thought that the program execution failed. This measurement is repeated sweeping clock frequency and supply voltage. This work obtained the Shmoo plots of the original and EDM-L programs (dijkstra, crc, and sha) for five test chips. The frequency interval was 5 MHz and the supply voltage was swept between 1.0 and 1.4 V with 0.1 V interval. Figs. 5.20 and 5.21 are the Shmoo plots of the fastest and slowest test chips among the five chips, where the sha-full-EDM-L program was executed. Even while the five chips were taken from the same wafer, the chip speed is different. Here, we define a term of FMAX. For each program execution, each chip and each supply voltage, we can find the FMAX at which the execution result starts to be incorrect. This frequency is defined as the FMAX. For example, in the Shmoo plot of Fig. 5.20, the FMAX at 1.0 V is 225 MHz, which is 15 MHz lower than that in Fig. 5.21.

5.4.2 Performance Evaluation Method

This work focuses on the first error that affects the execution result and checks whether EDM satisfies COND1 and COND2.

For this purpose, we need to know when the timing error occurs. However, in the hardware, it is difficult to know in which clock cycle the timing error occurs, unlike the simulation. Therefore, this work takes the following evaluation procedure. We change the clock frequency during the program execution as shown in Fig. 5.22. The program execution starts at 10 MHz, and the processor initialization completes at this frequency. Note that 10 MHz is low enough for the correct processor operation. When the user program execution starts, the clock frequency is changed to the FMAX. After N_{fast} clock

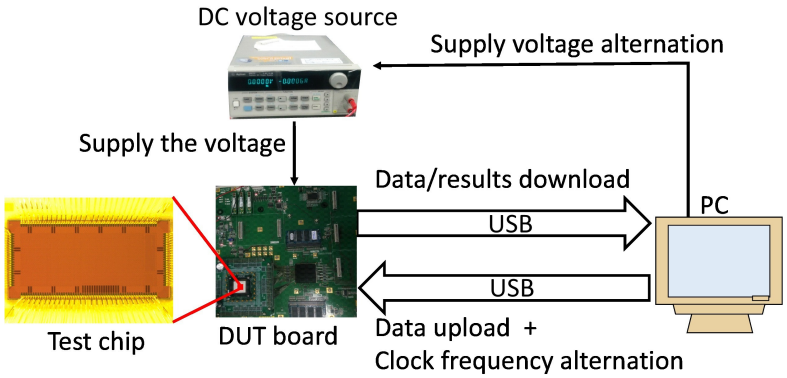

Figure 5.19: Measurement setup.

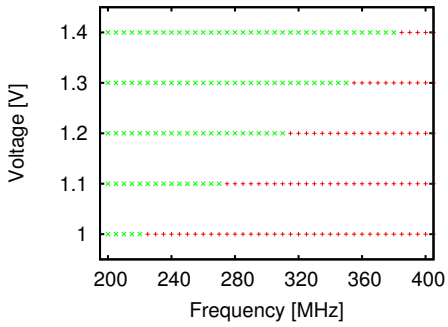

Figure 5.20: Shmoo plot of the slowest chip (chip #1).

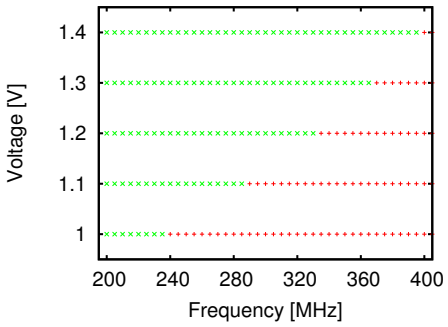

Figure 5.21: Shmoo plot of the fastest chip (chip #5).

cycles have passed, the clock frequency is again changed to 10 MHz. Under this configuration, if the execution result is incorrect, we can know the first error occurred within the first N_{fast} cycles. If the execution result is correct, no error occurred. This work repeats this measurement by changing N_{fast} in binary search manner and finally identifies the clock cycle when the first error occurred.

Remind that, when we decomposed the program into blocks, we numbered the blocks from the beginning. Accordingly, we can know in which block the first error occurred from N_{fast}. This work regards the difference of the error occurrence block numbers as the proximity of error occurrence locations in a similar way to the previous simulation. When the difference is zero, the timing error is reproduced at the same block in the EDM-transformed program and COND1 is satisfied. Evaluation for COND2 was also similar to Section 5.3. If the checker works, the error detection latency can be obtained from the difference between terminated clock cycle of the program and the error occurrence clock cycle.

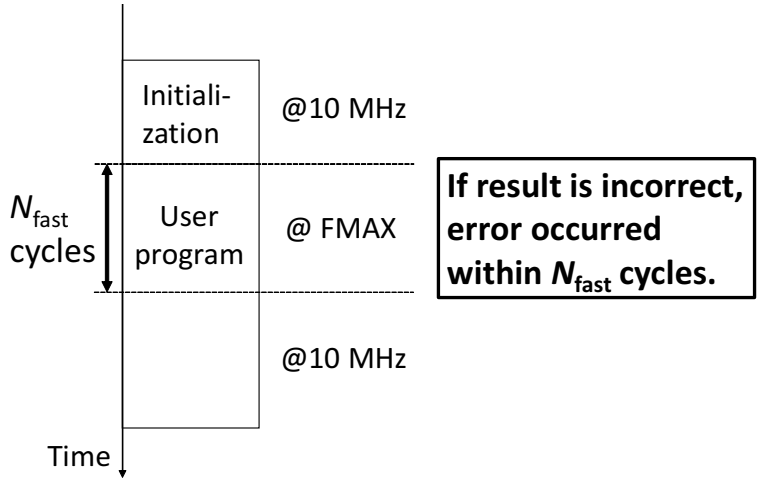

Figure 5.22: Procedure of error cycle identification.

5.4.3 Evaluation Results

Fig. 5.23(a) shows the ratio of the samples that satisfied COND1 of reproducibility and COND2 of detectability. The chip measurement result shows that 25% of the errors in the original program can be reproduced and quickly detected. Remind that, in the simulation result which is shown in the left figure of Fig. 5.10, EDM could not satisfy COND1 and COND2 simultaneously. In addition, the proportion that only COND1/COND2 is satisfied differs between the chip measurement and simulation. These differences will be discussed in the next section.

COND1

Fig. 5.23(b) shows the proximity of the errors occurrence in the original and the full-EDM dijkstra, crc and sha programs in the chip measurement. In the chip measurement, 66% of errors in crc and 20% of errors in dijkstra were reproduced. Note that in sha, no errors were reproduced. As a whole, EDM-L reproduced 29% errors in the chip measurement whereas 4% of errors were reproduced in the simulation as shown in Fig. 5.11. These differences of the reproducibility are supposed to be due to the following two reasons.

The first reason is the difference in the PDN between the simulation model and the hardware. In the previous simulations, ten different PDNs are used for the simulation to evaluate the performance against various supply noises, and they are not prepared to aim at modeling the test chip.

The second reason is the definition difference of the FMAX of the error occurrence between simulation and hardware measurement, whereas MeP processor was operated at the FMAXs in both the simulation and hardware measurement. In the simulation,

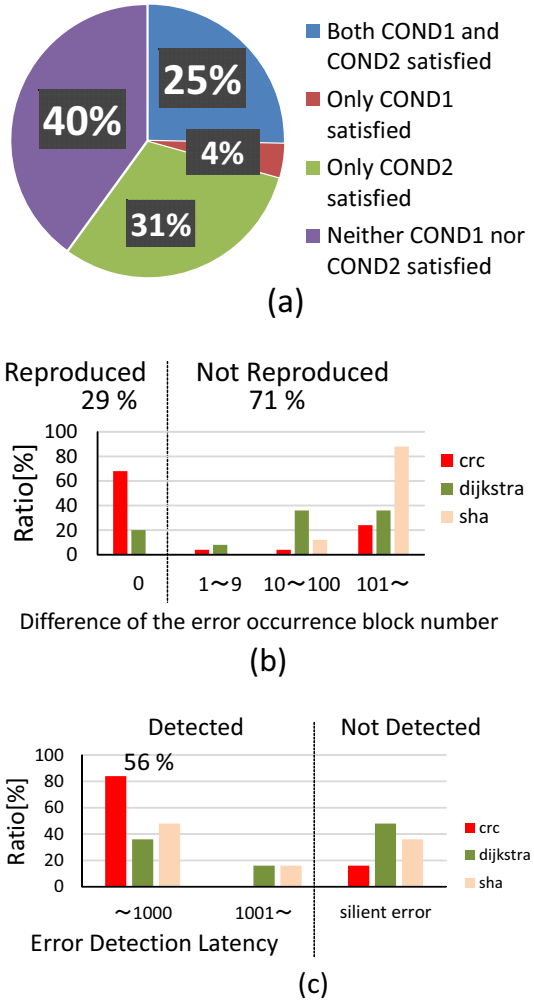

Figure 5.23: Results of chip measurement. (a) COND1+COND2(Scenario1), (b) COND1, and (c) COND2(Scenario2).

the FMAX was defined as the frequency at which a timing error started to occur at a flip-flop no matter whether the timing error affected the execution result or not (i.e., no matter whether it is masked or not). Hereafter, this FMAX is called the FMAX of timing error. In the chip measurement, the FMAX was defined as the frequency at which timing errors started to affect the execution result because the FMAX of timing error cannot be obtained in the hardware measurement. This FMAX is called the FMAX of incorrect execution. Here, the FMAX of incorrect execution is equal to or higher than that the FMAX of timing error. In other words, this work executed the original and the EDM programs at the higher frequency in the measurement compared to the simulation setup. Fig. 5.24 exemplifies the cycle time difference between the FMAXs of timing error and incorrect execution. This result was obtained by the simulation with full-EDM-L

programs. We can see that 70% of the samples have ≥ 0.2 ns difference.

COND2

Next, COND2 is examined. this section first categorized the measured samples into detected samples and not detected samples. In the chip measurement, this work focused on the errors affecting the execution result, and hence not detected samples correspond to silent errors, whereas not detected samples include silent errors and masked errors in the simulation. Fig. 5.23(c) shows the proportions of detected errors and silent errors in the chip measurement. For detected errors, the histogram of the error detection latency is presented. From Fig. 5.23(c), we can see that 56% of the errors are quickly detected and 33% are silent errors. Compared with Figs. 5.16 and 5.23(c), the EDM-L performance of detecting supply noise induced timing errors affecting execution result in simulation evaluation is higher than in the chip measurement (86% versus 56%).

5.5 Correlation between the Measurement and Simulation

The experimental results in the previous section show that in the chip measurement, COND1 is more satisfied and COND2 is less satisfied compared to the simulation. The possible reasons for these differences are (1) the difference of the power distribution network, and (2) the difference of the FMAX definition, as described in the previous section. This section improves the correlation between the measurement and simulation by updating the simulation setup taking into account these two possible reasons.

5.5.1 Power Distribution Network

The chip measurement results in the previous section lead to a hypothesis that the supply noise in the test chip is smaller than that in the simulation and hence the errors are more likely to be reproduced in the chip measurement.

To verify the above hypothesis, let us suppose the test chip has ideal PDN as an extreme case. In other words, the simulation based evaluation is executed in a similar way to Section 5.3 except that the supply voltage is fixed and the supply noise is zero. In this simulation, 3 programs (dijkstra, crc, and sha) and 10 chips are prepared, and hence totally 30 samples are evaluated.

Fig. 5.25(a) shows the evaluation results of Scenario1. We can see that EDM could not satisfy COND1 and COND2 simultaneously, which is not consistent with chip measurement results. On the other hand, the proportion of COND1 satisfaction increased from 4% to 13%, approaching to the measurement result of 29% (Fig. 5.23(a)).

Next, Fig. 5.25(b) shows the results of Scenario2, and we can see that 76% of the errors are masked error and there are no silent errors. Focusing on the non-masked

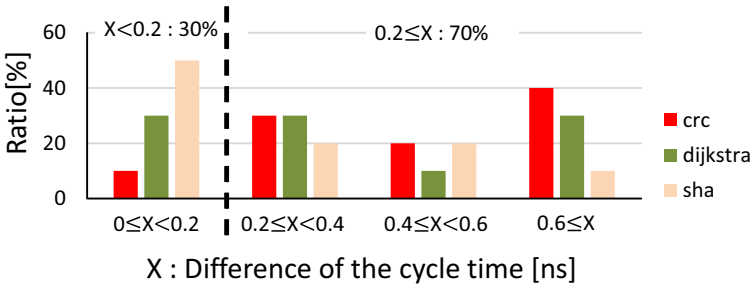

Figure 5.24: Histogram of cycle time difference between FMAXs of timing error and incorrect execution.

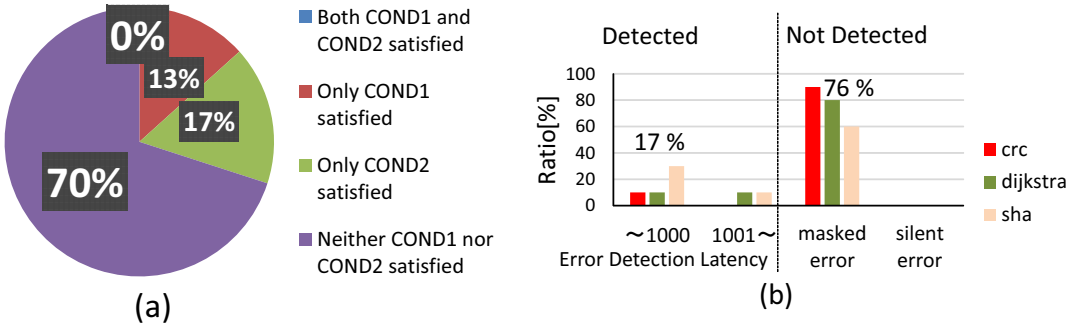

Figure 5.25: Simulation results with ideal PDN. FMAX of timing error is used for evaluation. (a) Scenario1, and (b) Scenario2.

error, 70% were quickly detected. The ratio of quick detection degraded from 86% but it approaches to 56% of the chip measurement result (Fig. 5.23(c)).

5.5.2 FMAX

In the chip measurement, the FMAX of incorrect execution was used while the FMAX of timing error was used in the simulation in Section 5.3. To clarify the difference originating from this difference of FMAX, the FMAX of incorrect execution was applied to the simulation. In the simulation here, the FMAX of incorrect execution was searched with 200 ps interval, which is also similar to the measurement setup. 3 programs and 10 chips are used in a similar way to the previous evaluation. In addition, ten PDNs, which are used in Section 5.3, are prepared. Consequently, we evaluated whether EDM satisfied COND1/COND2 for 300 samples.

Fig. 5.26(a) shows the result for Scenario1. Comparing Figs. 5.26(a) and 5.23(a), we can find a large difference in the proportion that both the reproducibility and detectability are satisfied, which is 0% in the simulation and 25% in the chip measurement. Fig. 5.26(b) shows the result for Scenario2. The detectability for the non-masked error

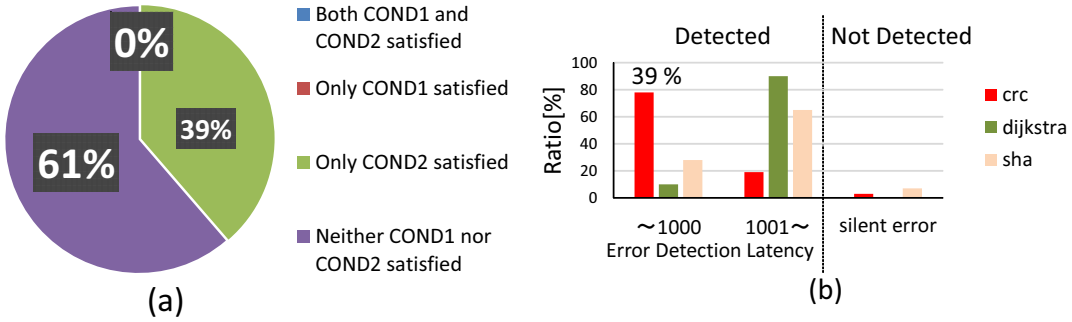

Figure 5.26: Simulation results with not ideal PDN. FMAX of incorrect execution is used for evaluation. (a) Scenario1, and (b) Scenario2.

has become close to between the simulation and chip measurement, where it is 39% in the simulation and 56% in the chip measurement (Fig. 5.23(c)).

5.5.3 Power Distribution Network and FMAX

Finally, this section applied the ideal PDN with no noise and the FMAX of incorrect execution to the simulation.

Fig. 5.27(a) shows the result for Scenario1. Figs. 5.27(a) and 5.23(a) indicate that the proportions that both reproducibility and detectability are satisfied are almost the same and they are 23% in the simulation and 25% in the chip measurement. We can also see that the portions of only reproducibility/detectability is satisfied and neither satisfied are consistent between the simulation and chip measurement.

Fig. 5.27(b) shows the Scenario2 results. 43% of the errors are quickly detected and 13% are silent errors. Comparing Fig. 5.27(b) with Fig. 5.23(c), we can see that detectability for the non-masked error is similar, that is 43% in the simulation and 56% in the chip measurement.

Based on the discussion above, we can conclude that the simulation with ideal PDN and FMAX of incorrect execution reproduced the chip measurement results. This means that the supply noise in the test chip is smaller than that in the simulation, which is quite natural since the test chip was not designed for the purpose of EDM performance evaluation and hence the PDN was robustly designed.

5.6 Conclusion

This chapter tackled the error localization of the supply noise induced timing errors, which is one of the most challenging tasks in the post-silicon validation. In this chapter, we devised the software-based error detection mechanisms for short latency (EDM-L)

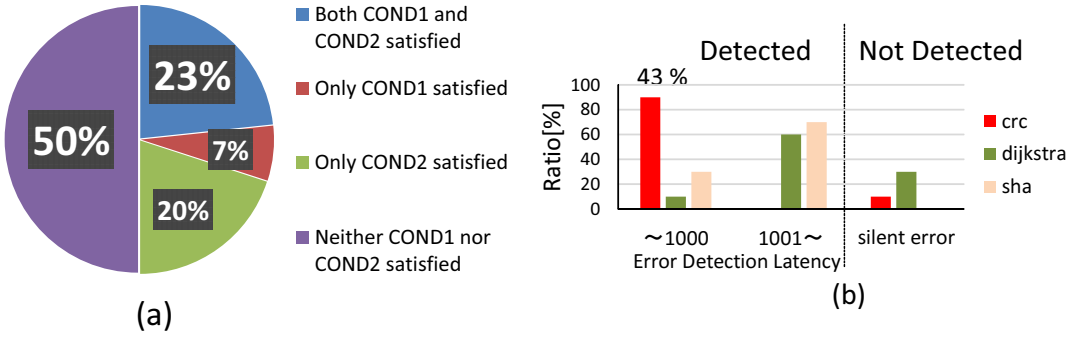

Figure 5.27: Simulation results with ideal PDN. FMAX of incorrect execution is used for evaluation. (a) Scenario1, and (b) Scenario2.

and experimentally evaluated the error detection performance of the EDM-L transformation for supply noise induced timing errors. To discuss the effectiveness of EDM-L for the timing error localization, two EDM usage scenarios are supposed: localizing the supply noise induced timing error that occurred in the original program (Scenario1), and localizing as many potential errors as possible (Scenario2). This chapter experimentally evaluated the error detection performance in these two scenarios with a noise-aware logic simulator and 65 nm test chips. Simulation results showed that the EDM-L cannot locate supply noise induced timing errors in the original program in the first scenario, but it detects 86% of non-masked errors in the second scenario, which mean the EDM-L performance of detecting supply noise induced timing errors affecting execution results is high. Hardware measurement results showed that the EDM-L detects 25% of original timing errors and 56% of non-masked errors. These measurement results were not consistent with the simulation results. We found that this inconsistency came from (1) the design of power distribution network, and (2) the definition of FMAX used for evaluation. By updating the simulation setup, the EDM-L performance evaluated by the simulation was consistent with that by the chip measurement. In other words, this work confirmed that the devised EDM-L helps to localize the supply noise induced timing errors with both the simulation and hardware measurement. This result reveals that the EDM-L facilitates the post-silicon validation of the designed AVS circuit and thus improves the reliability of the AVS circuit against the supply noise.

Chapter 6

Conclusion

This thesis studied the AVS and proposed the design and post-silicon validation methodology for the AVS circuit. The proposed design methodology consists of three steps; (1) select the type of sensor, (2) optimize the main logic under AVS, and (3) insert sensors to the optimized main logic.

In the first step of the sensor selection, this thesis discussed the supply voltage reductions achieved by AVS circuits with different sensors in Chapter 2. This work focused on error prediction based AVS and thus selected two representative sensors as candidates; in-situ sensors and critical path replica. This work gave the MTTF as a design constraint and compared the trade-offs of the clock period and the average supply voltage between AVS circuits with these two sensors. For such MTTF-aware trade-off analysis, the stochastic error rate estimation framework was utilized. Experimental results showed that TEP-FF based AVS and replica based AVS achieved up to 13.3% and 8.9% supply voltage reduction, respectively, while satisfying the target MTTF.

In the second step of the main logic design, this thesis explicitly introduced the MTTF as a design constraint and optimized the design with ASA in Chapter 3. ASA gives timing slacks to non-intrinsic active critical paths by ECO and hence reduces the number of critical paths. The reduction of active critical paths helps extend the TTF and thus reduces the supply voltage since timing failure probabilities of such paths can be dramatically decreased. The proposed optimization includes the pre-ASA circuit design and ASA implementation. In the pre-ASA circuit design, the proposed methodology selected the most promising one from candidates regarding power dissipation. The evaluation results showed that the ASA circuit achieved 13.9% to 22.3% power saving from an MTTF-aware design without main logic optimization. Also, thanks to the pre-ASA design in the proposed methodology, 6.4% of the area and 7.7% of low-V_{th} cells were reduced in the AES circuit, and 1.4% of the area and 7.7% of low-V_{th} cells were reduced in the OpenRISC processor.

In the third step of the sensor insertion, this thesis aimed at maximizing the MTTF for maximumly reducing the power dissipation. To maximize the MTTF, this thesis proposed a novel sensor insertion method that maximumly decreases the sum of gate-

wise timing failure probabilities. By exploiting the information on the paths with the higher timing failure probability, the proposed sensor insertion makes AVS efficiently monitor the timing-critical and highly-active FFs. Moreover, by maximizing the sum of gate-wise failure probabilities, the proposed sensor insertion can cover the larger set of instances that can contribute to causing timing errors. With the proposed design methodology consisting of above three steps, i.e. sensor type selection, main logic optimization, and sensor insertion, 38.0% power reduction was achieved while satisfying the target MTTF. We also experimentally confirmed that the proposed main logic deign is highly compatible with AVS and the proposed sensor insertion methodology achieved the much longer MTTF compared with the conventional slack-oriented insertion methodology.

As for the post-silicon validation methodology, this thesis devised the EDM-L and evaluated the performance of EDM-L for timing error localization with a noise-aware logic simulator and 65-nm test chips assuming the following two EDM-L usage scenarios; (1) localizing a timing error that occurred in the original program and (2) localizing as many potential timing errors as possible. Simulation results showed that the EDM-L could not locate supply noise induced timing errors in the original program in the first scenario, but it detected 86% of non-masked errors in the second scenario, which mean that the EDM-L performance of detecting supply noise induced timing errors that affect execution results is high. Hardware measurement results showed that the EDM-L detected 25% of original timing errors and 56% of non-masked errors. These measurement results were not consistent with the simulation results. We found that this inconsistency came from (1) the design of power distribution network, and (2) the definition of FMAX used for evaluation. By updating the simulation setup, the EDM-L performance evaluated by the simulation was consistent with that by the chip measurement, which indicates that the devised EDM-L contributes to localizing the supply noise induced errors and thus improving the reliability of the AVS circuit.

This thesis contributes to constructing reliable AVS circuits which is supported by the MTTF-aware design methodology, quantitative performance analyses taking into account the power dissipation and the practical long MTTF, and the post-silicon validation methodology for localizing the supply noise induced timing errors. However, there are still challenges for practical use. Even though sensor selection and insertion are discussed in Chapters 2 and 4, the amount of delay buffers in each sensor is not optimized in detail while it directly affects the timing margins reserved in AVS operation. This thesis determined it according to the amount of delay increase that corresponds to one level decrement of the supply voltage. Although this setting relatively works well, which can be seen in the MTTF evaluation results in Chapters 2 and 4, there remains a design space to explore. The buffer delay itself fluctuates due to static and dynamic variations, and hence design parameters of buffers, such as gate width and threshold voltage, and their combination should be also investigated since they have different characteristics of delay sensitivity to supply noise, aging, the amount of random delay variation, area, and

power.

Another future work is to design and fabricate the reliable AVS chips. To realize such chips, it needs to consider the impact of sensors and the voltage control logic on the design layout and validate the AVS circuit and its design methodology via the chip measurement. Here, an important mission is to mitigate the layout change due to sensors and the voltage control logic insertion. Note that the inserted sensors and the voltage control logic affect the cell placement, clock tree synthesis, and routing in design layout. In other words, they may degrade the delay characteristics of the AVS circuit due to the increase in cell delay and wiring delay. Although this thesis ideally inserted sensors to voltage-scaled circuits for simplicity, it is essential to verify whether sensors and the voltage control logic can be inserted to the desired places. Another crucial point is to verify the AVS circuit via the chip measurement. In this verification, the circuit power and performance can be evaluation metrics. In addition, chip-dependent performance variation and delay mismatch between pre-fabrication and post-fabrication may be examined. Measuring above metrics for fabricated chips further improves the reliability of the designed AVS.

Bibliography

[1] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol. 19, no. 4, pp. 23–29, July 1999.

[2] W. Huang, K. Rajamani, M. R. Stan, and K. Skadron, "Scaling with design constraints: Predicting the future of big chips," *IEEE Micro*, vol. 31, no. 4, pp. 16–29, July 2011.

[3] C. Hwang, "New paradigms in the silicon industry," in *Proceedings of IEEE International Electron Devices Meeting*, pp. 1–8, Dec. 2006.

[4] S. Lee, "Technology scaling challenges and opportunities of memory devices," in *Proceedings of IEEE International Electron Devices Meeting*, pp. 1.1.1–1.1.8, Dec. 2016.

[5] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, Apr. 1992.

[6] N. S. Kim, T. Austin, D. Baauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, and V. Narayanan, "Leakage current: Moore's law meets static power," *IEEE Transactions on Computers*, vol. 36, no. 12, pp. 68–75, Dec. 2003.

[7] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. USA: Addison-Wesley Publishing Company, 2010.

[8] L. Benini, A. Bogliolo, and G. D. Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 3, pp. 299–316, June 2000.

[9] International technology roadmap for semiconductors 2.0 2015 edition executive report. [Online]. Available: https://www.dropbox.com/sh/3jfh5fq634b5yqu/AADYT8V2Nj5bX6C5q764kUg4a?dl=0/.

[10] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45 nm early design exploration," *IEEE Transactions on Electron Devices*, vol. 53, no. 11, pp. 2816–2823, Nov. 2006.

[11] A. Agarwal, D. Blaauw, and V. Zolotov, "Statistical timing analysis for intra-die process variations with spatial correlations," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 900–907, Nov. 2003.

[12] S. R. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "Varius: A model of process variation and resulting timing errors for microarchitects," *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, no. 1, pp. 3–13, Feb. 2008.

[13] K. J. Kuhn, M. D. Giles, D. Becher, P. Kolar, A. Kornfeld, R. Kotlyar, S. T. Ma, A. Maheshwari, and S. Mudanai, "Process technology variation," *IEEE Transactions on Electron Devices*, vol. 58, no. 8, pp. 2197–2208, Aug. 2011.

[14] M. Tehranipoor and K. M. Butler, "Power supply noise: A survey on effects and research," *IEEE Design and Test of Computers*, vol. 27, no. 2, pp. 51–67, Mar. 2010.

[15] K. Arabi, R. Saleh, and X. Meng, "Power supply noise in SoCs: Metrics, management, and measurement," *IEEE Design and Test of Computers*, vol. 24, no. 3, pp. 236–244, May 2007.

[16] H. H. Chen and D. D. Ling, "Power supply noise analysis methodology for deep-submicron VLSI chip design," in *Proceedings of IEEE/ACM Design Automation Conference*, pp. 638–643, June 1997.

[17] B. Zhang and M. Orshansky, "Modeling of NBTI-induced PMOS degradation under arbitrary dynamic temperature variation," in *Proceedings of IEEE International Symposium on Quality Electronic Design*, pp. 774–779, Mar. 2008.

[18] W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and Y. Cao, "The impact of NBTI on the performance of combinational and sequential circuits," in *Proceedings of IEEE/ACM Design Automation Conference*, pp. 364–369, June 2007.

[19] J. F. Zhang and W. Eccleston, "Positive bias temperature instability in MOSFETs," *IEEE Transactions on Electron Devices*, vol. 45, no. 1, pp. 116–124, Jan. 1998.

[20] P. Heremans, R. Bellens, G. Groeseneken, and H. E. Maes, "Consistent model for the hot-carrier degradation in n-channel and p-channel mosfets," *IEEE Transactions on Electron Devices*, vol. 35, no. 12, pp. 2194–2209, Dec. 1988.

[21] S. Zafar, Y. Kim, V. Narayanan, C. Cabral, V. Paruchuri, B. Doris, J. Stathis, A. Callegari, and M. Chudzik, "A comparative study of NBTI and PBTI (charge trapping) in SiO2/HfO2 stacks with FUSI, TiN, Re gates," in *Proceedings of*

IEEE Symposium on VLSI Technology and Circuits. Digest of Technical Papers, pp. 23–25, June 2006.

[22] T. Grasser, B. Kaczer, W. Goes, H. Reisinger, T. Aichinger, P. Hehenberger, P. Wagner, F. Schanovsky, J. Franco, M. T. Luque, and M. Nelhiebel, "The paradigm shift in understanding the bias temperature instability: From reaction–diffusion to switching oxide traps," *IEEE Transactions on Electron Devices*, vol. 58, no. 11, pp. 3652–3666, Nov. 2011.

[23] S. Mahapatra, N. Goel, S. Desai, S. Gupta, B. Jose, S. Mukhopadhyay, K. Joshi, A. Jain, A. E. Islam, and M. A. Alam, "A comparative study of different physics-based NBTI models," *IEEE Transactions on Electron Devices*, vol. 60, no. 3, pp. 901–916, Mar. 2013.

[24] S. Mukhopadhyay, N. Goel, and S. Mahapatra, "A comparative study of NBTI and PBTI using different experimental techniques," *IEEE Transactions on Electron Devices*, vol. 63, no. 10, pp. 4038–4045, Oct. 2016.

[25] V. M. van Santen, J. Martin-Martinez, H. Amrouch, M. M. Nafria, and J. Henkel, "Reliability in super- and near-threshold computing: A unified model of RTN, BTI, and PV," *IEEE Transactions on Circuits and Systems—Part I: Regular Papers*, vol. 65, no. 1, pp. 293–306, Jan. 2018.

[26] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.

[27] B. J. Sheu, D. L. Scharfetter, P. . Ko, and M. . Jeng, "Bsim: Berkeley short-channel IGFET model for MOS transistors," *IEEE Journal of Solid-State Circuits*, vol. 22, no. 4, pp. 558–566, Aug. 1987.

[28] Y. Zhang, M. Khayatzadeh, K. Yang, M. Saligane, N. Pinckney, M. Alioto, D. Blaauw, and D. Sylvester, "iRazor: Current-based error detection and correction scheme for PVT variation in 40-nm ARM Cortex-R4 processor," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 2, pp. 619–631, Feb. 2018.

[29] S. Das, D. Roberts, S. Lee, S. Pant, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "A self-tuning DVS processor using delay-error detection and correction," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 792–804, Apr. 2006.

[30] A. Drake, R. Senger, H. Deogun, G. Carpenter, S. Ghiasi, T. Nguyen, N. James, M. Floyd, and V. Pokala, "A distributed critical-path timing monitor for a 65nm high-performance microprocessor," in *Proceedings of IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, pp. 398–399, Feb. 2007.

[31] J. Tschanz, K. Bowman, S. Walstra, M. Agostinelli, T. Karnik, and V. De, "Tunable replica circuits and adaptive voltage-frequency techniques for dynamic voltage, temperature, and aging variation tolerance," in *Proceedings of IEEE Symposium on VLSI Technology and Circuits. Digest of Technical Papers*, pp. 112–113, June 2009.

[32] S. Das, C. Tokunaga, S. Pant, W. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw, "RazorII: In situ error detection and correction for PVT and SER tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, Jan. 2009.

[33] K. A. Bowman, J. W. Tschanz, S. L. Lu, P. A. Aseron, M. M. Khellah, A. Raychowdhury, B. M. Geuskens, C. Tokunaga, C. B. Wilkerson, T. Karnik, and V. K. De, "A 45 nm resilient microprocessor core for dynamic variation tolerance," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 194–208, Jan. 2011.

[34] D. Bull, S. Das, K. Shivashankar, G. S. Dasika, K. Flautner, and D. Blaauw, "A power-efficient 32 bit ARM processor using timing-error detection and correction for transient-error tolerance and adaptation to PVT variation," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 18–31, Jan. 2011.

[35] H. Fuketa, M. Hashimoto, Y. Mitsuyama, and T. Onoye, "Adaptive performance compensation with in-situ timing error predictive sensors for subthreshold circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 2, pp. 333–343, Feb. 2012.

[36] K. Chae and S. Mukhopadhyay, "All-digital adaptive clocking to tolerate transient supply noise in a low-voltage operation," *IEEE Transactions on Circuits and Systems—Part II: Express Briefs*, vol. 59, no. 12, pp. 893–897, Dec. 2012.

[37] Y. Ikenaga, M. Nomura, S. Suenaga, H. Sonohara, Y. Horikoshi, T. Saito, Y. Ohdaira, Y. Nishio, T. Iwashita, M. Satou, K. Nishida, K. Nose, K. Noguchi, Y. Hayashi, and M. Mizuno, "A 27% active-power-reduced 40-nm CMOS multimedia SoC with adaptive voltage scaling using distributed universal delay lines," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 4, pp. 832–840, Apr. 2012.

[38] M. Fojtik, D. Fick, Y. Kim, N. Pinckney, D. M. Harris, D. Blaauw, and D. Sylvester, "Bubble Razor: Eliminating timing margins in an ARM Cortex-M3 processor in 45 nm CMOS using architecturally independent error detection and correction," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 66–81, Jan. 2013.

[39] C. R. Lefurgy, A. J. Drake, M. S. Floyd, M. S. Allen-Ware, B. Brock, J. A. Tierno, J. B. Carter, and R. W. Berry, "Active guardband management in Power7+ to save energy and maintain reliability," *IEEE Micro*, vol. 33, no. 4, pp. 35–45, July 2013.

[40] I. Kwon, S. Kim, D. Fick, M. Kim, Y. Chen, and D. Sylvester, "Razor-lite: A light-weight register for error detection by observing virtual supply rails," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 9, pp. 2054–2066, Sept. 2014.

[41] S. Kim and M. Seok, "Variation-tolerant, ultra-low-voltage microprocessor with a low-overhead, within-a-cycle in-situ timing-error detection and correction technique," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 6, pp. 1478–1490, June 2015.

[42] A. Benhassain, F. Cacho, V. Huard, M. Saliva, L. Anghel, C. Parthasarathy, A. Jain, and F. Giner, "Timing in-situ monitors: Implementation strategy and applications results," in *Proceedings of IEEE Custom Integrated Circuits Conference*, pp. 1–4, Sept. 2015.

[43] J. L. Nunez-Yanez, "Adaptive voltage scaling with in-situ detectors in commercial FPGAs," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 45–53, Jan. 2015.

[44] V. Huard, F. Cacho, A. Benhassain, and C. Parthasarathy, "Aging-aware adaptive voltage scaling of product blocks in 28nm nodes," in *Proceedings of IEEE International Reliability Physics Symposium*, pp. 7C–2–1–7C–2–7, Apr. 2016.

[45] J. L. Nunez-Yanez, M. Hosseinabady, and A. Beldachi, "Energy optimization in commercial FPGAs with voltage, frequency and logic scaling," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1484–1493, May 2016.

[46] J. P. Kulkarni, C. Tokunaga, P. A. Aseron, T. Nguyen, C. Augustine, J. W. Tschanz, and V. De, "A 409 GOPS/W adaptive and resilient domino register file in 22 nm tri-gate CMOS featuring in-situ timing margin and error detection for tolerance to within-die variation, voltage droop, temperature and aging," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 1, pp. 117–129, Jan. 2016.

[47] S. Mhira, V. Huard, A. Benhassain, F. Cacho, S. Naudet, A. Jain, C. Parthasarathy, and A. Bravaix, "Dynamic adaptive voltage scaling in automotive environment," in *Proceedings of IEEE International Reliability Physics Symposium*, pp. 3A–4.1–3A–4.7, Apr. 2017.

[48] J. Wang and S. Wei, "Process/voltage/temperature-variation-aware design and comparative study of transition-detector-based error-detecting latches for timing-error-resilient pipelined systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2893–2906, Oct. 2017.

[49] W. Jin, S. Kim, W. He, Z. Mao, and M. Seok, "In situ error detection techniques in ultralow voltage pipelines: Analysis and optimizations," *IEEE Transactions*

on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 3, pp. 1032–1043, Mar. 2017.

[50] B. Keller, M. Cochet, B. Zimmer, J. Kwak, A. Puggelli, Y. Lee, M. Blagojević, S. Bailey, P. Chiu, P. Dabbelt, C. Schmidt, E. Alon, K. Asanović, and B. Nikolić, "A RISC-V processor SoC with integrated power management at submicrosecond timescales in 28 nm FD-SOI," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 7, pp. 1863–1875, July 2017.

[51] M. Cho, S. T. Kim, C. Tokunaga, C. Augustine, J. P. Kulkarni, K. Ravichandran, J. W. Tschanz, M. M. Khellah, and V. De, "Postsilicon voltage guard-band reduction in a 22 nm graphics execution core using adaptive voltage scaling and dynamic power gating," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 50–63, Jan. 2017.

[52] C. Gonzalez, M. Floyd, E. Fluhr, P. Restle, D. Dreps, M. Sperling, R. Rao, D. Hogenmiller, C. Vezyrtis, P. Chuang, D. Lewis, R. Escobar, V. Ramadurai, R. Kruse, J. Pille, R. Nett, P. Owczarczyk, J. Friedrich, J. Paredes, T. Diemoz, S. Islam, D. Plass, and P. Muench, "The 24-core POWER9 processor with adaptive clocking, 25-Gb/s accelerator links, and 16-Gb/s PCIe Gen4," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 1, pp. 91–101, Jan. 2018.

[53] J. Kim, K. Choi, Y. Kim, W. Kim, K. Do, and J. Choi, "Delay monitoring system with multiple generic monitors for wide voltage range operation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 1, pp. 37–49, Jan. 2018.

[54] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, "A dynamic voltage scaled microprocessor system," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, Nov. 2000.

[55] T. Mudge and U. Holzle, "Challenges and opportunities for extremely energy-efficient processors," *IEEE Micro*, vol. 30, no. 4, pp. 20–24, July 2010.

[56] B. Acun, P. Miller, and L. V. Kale, "Variation among processors under turbo boost in HPC systems," in *Proceedings of ACM International Conference on Supercomputing*, pp. 6:1–6:12. ACM, 2016.

[57] T. Sato and Y. Kunitake, "A simple flip-flop circuit for typical-case designs for DFM," in *Proceedings of IEEE International Symposium on Quality Electronic Design*, pp. 539–544, Mar. 2007.

[58] L. Lai, V. Chandra, R. C. Aitken, and P. Gupta, "SlackProbe: A flexible and efficient in situ timing slack monitoring methodology," *IEEE Transactions on*

Computer-Aided Design of Integrated Circuits and Systems, vol. 33, no. 8, pp. 1168–1179, Aug. 2014.

[59] J. Zhou, X. Liu, Y. Lam, C. Wang, K. Chang, J. Lan, and M. Je, "Hepp: A new in-situ timing-error prediction and prevention technique for variation-tolerant ultra-low-voltage designs," in *Proceedings of IEEE Asian Solid-State Circuits Conference*, pp. 129–132, Nov. 2013.

[60] S. Iizuka, M. Mizuno, D. Kuroda, M. Hashimoto, and T. Onoye, "Stochastic error rate estimation for adaptive speed control with field delay testing," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 107–114, Nov. 2013.

[61] S. Iizuka, Y. Masuda, M. Hashimoto, and T. Onoye, "Stochastic timing error rate estimation under process and temporal variations," in *Proceedings of IEEE International Test Conference*, Oct. 2015.

[62] Q. Guo, T. Chen, Y. Chen, R. Wang, H. Chen, W. Hu, and G. Chen, "Pre-silicon bug forecast," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 3, pp. 451–463, Mar. 2014.

[63] A. DeOrio, J. Li, and V. Bertacco, "Bridging pre- and post-silicon debugging with BiPeD," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 95–100, Nov. 2012.

[64] A. Adir, A. Nahir, G. Shurek, A. Ziv, C. Meissner, and J. Schumann, "Leveraging pre-silicon verification resources for the post-silicon validation of the IBM POWER7 processor," in *Proceedings of IEEE/ACM Design Automation Conference*, pp. 569–574, June 2011.

[65] P. Patra, "On the cusp of a validation wall," *IEEE Design and Test of Computers*, vol. 24, no. 2, pp. 193–196, Mar. 2007.

[66] D. Josephson, "The good, the bad, and the ugly of silicon debug," in *Proceedings of IEEE/ACM Design Automation Conference*, pp. 3–6, July 2006.

[67] T. Hong, Y. Li, S. Park, D. Mui, D. Lin, Z. A. Kaleq, N. Hakim, H. Naeimi, D. S. Gardner, and S. Mitra, "QED: Quick error detection tests for effective post-silicon validation," in *Proceedings of IEEE International Test Conference*, pp. 1–10, Nov. 2010.

[68] D. Lin, T. Hong, Y. Li, E. S, S. Kumar, F. Fallah, N. Hakim, D. S. Gardner, and S. Mitra, "Effective post-silicon validation of system-on-chips using quick error detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 10, pp. 1573–1590, Oct. 2014.

[69] S. Park, T. Hong, and S. Mitra, "Post-silicon bug localization in processors using instruction footprint recording and analysis (IFRA)," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1545–1558, Oct. 2009.

[70] A. A. Bayazit and S. Malik, "Complementary use of runtime validation and model checking," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 1052–1059, Nov. 2005.

[71] M. Boule and Z. Zilic, "Incorporating efficient assertion checkers into hardware emulation," in *Proceedings of IEEE International Conference on Computer Design*, pp. 221–228, Oct. 2005.

[72] S. Mitra, S. A. Seshia, and N. Nicolici, "Post-silicon validation opportunities, challenges and recent advances," in *Proceedings of IEEE/ACM Design Automation Conference*, pp. 12–17, June 2010.

[73] D. Lin, E. Singh, C. Barrett, and S. Mitra, "A structured approach to post-silicon validation and debug using symbolic quick error detection," in *Proceedings of IEEE International Test Conference*, pp. 1–10, Oct. 2015.

[74] M. Rebaudengo, M. S. Reorda, M. Torchiano, and M. Violante, "Soft-error detection through software fault-tolerance techniques," in *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 210–218, Nov. 1999.

[75] B. Nicolescu and R. Velazco, "Detecting soft errors by a purely software approach: method, tools and experimental results," in *Proceedings of IEEE Design, Automation and Test in Europe Conference and Exhibition*, pp. 57–62 suppl., Mar. 2003.

[76] Y. Masuda, J. Nagayama, H. Takeno, Y. Ogawa, Y. Momiyama, and M. Hashimoto, "Comparing voltage adaptation performance between replica and in-situ timing monitors," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–8, Nov. 2018.

[77] Nangate, NanGate FreePDK 45nm Cell Library. [Online]. Available: http://www.nangate.com/?page_id=2325/.

[78] A. Teman, D. Rossi, P. Meinerzhagen, L. Benini, and A. Burg, "Controlled placement of standard cell memory arrays for high density and low power in 28nm FD-SOI," in *Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference*, pp. 81–86, Jan. 2015.

[79] P. Meinerzhagen, C. Roth, and A. Burg, "Towards generic low-power area-efficient standard cell based memory architectures," in *Proceedings of IEEE International Midwest Symposium on Circuits and Systems*, pp. 129–132, Aug. 2010.

[80] J. Shiomi, T. Ishihara, and H. Onodera, "Fully digital on-chip memory using minimum height standard cells for near-threshold voltage computing," in *Proceedings of IEEE International Workshop on Power and Timing Modeling, Optimization and Simulation*, pp. 44–49, Sept. 2016.

[81] Synopsys, "VCS User Guide," 2013.

[82] Synopsys, "Primetime User Guide," 2015.

[83] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "MiBench: A free, commercially representative embedded benchmark suite," in *Proceedings of IEEE International Workshop on Workload Characterization*, pp. 3–14, Dec. 2001.

[84] H. Chang and S. S. Sapatnekar, "Statistical timing analysis under spatial correlations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 9, pp. 1467–1482, Sept. 2005.

[85] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, S. Narayan, D. K. Beece, J. Piaget, N. Venkateswaran, and J. G. Hemmett, "First-order incremental block-based statistical timing analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 10, pp. 2170–2180, Oct. 2006.

[86] V. Garg, "Common path pessimism removal: An industry perspective: Special session: Common path pessimism removal," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 592–595, Nov. 2014.

[87] J. Zejda and P. Frain, "General framework for removal of clock network pessimism," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 632–639, Nov. 2002.

[88] Mathworks, MATLAB. [Online]. Available: https://www.mathworks.com/products/matlab/.

[89] J. B. Velamala, K. B. Sutaria, H. Shimizu, H. Awano, T. Sato, G. Wirth, and Y. Cao, "Compact modeling of statistical BTI under trapping/detrapping," *IEEE Transactions on Electron Devices*, vol. 60, no. 11, pp. 3645–3654, Nov. 2013.

[90] H. Awano, M. Hiromoto, and T. Sato, "Variability in device degradations: Statistical observation of NBTI for 3996 transistors," in *Proceedings of IEEE European Solid State Device Research Conference*, pp. 218–221, Sept. 2014.

[91] Y. Li, X. Zhang, Z. Zhang, and Y. Lian, "A 0.45-to-1.2-V fully digital low-dropout voltage regulator with fast-transient controller for near/subthreshold circuits," *IEEE Transactions on Power Electronics*, vol. 31, no. 9, pp. 6341–6350, Sept. 2016.

[92] Y. Masuda, T. Onoye, and M. Hashimoto, "Activation-aware slack assignment for time-to-failure extension and power saving," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 11, pp. 2217–2229, Nov. 2018.

[93] P. Cappello and K. Steiglitz, "Some complexity issues in digital signal processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 5, pp. 1037–1041, Oct. 1984.

[94] October Sky, Gurobi Optimizer. [Online]. Available: https://www.octobersky.jp/products/gurobi.html/.

[95] F. N. Najm, "Transition density: a new measure of activity in digital circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 2, pp. 310–323, Feb. 1993.

[96] Y. Masuda and M. Hashimoto, "MTTF-aware design methodology of error prediction based adaptively voltage-scaled circuits," in *Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference*, pp. 159–165, Jan. 2018.

[97] Cadence, "Innovus User Guide," 2017.

[98] Y. Masuda, T. Onoye, and M. Hashimoto, "Performance evaluation of software-based error detection mechanisms for supply noise induced timing errors," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E100.A, no. 7, pp. 1452–1463, 2017.

[99] M. Ueno, M. Hashimoto, and T. Onoye, "Real-time on-chip supply voltage sensor and its application to trace-based timing error localization," in *Proceedings of IEEE International Symposium on On-Line Testing and Robust System Design*, pp. 188–193, July 2015.

[100] Synopsys, "HSPICE User Guide," 2012.

[101] Synopsys, "NanoSim User Guide," 2011.

[102] L. D. Smith, R. E. Anderson, D. W. Forehand, T. J. Pelc, and T. Roy, "Power distribution system design methodology and capacitor selection for modern CMOS technology," *IEEE Transactions on Advanced Packaging*, vol. 22, no. 3, pp. 284–291, Aug. 1999.

[103] T. Roy, L. Smith, and J. Prymak, "ESR and ESL of ceramic capacitor applied to decoupling applications," in *Proceedings of IEEE Topical Meeting on Electrical Performance of Electronic Packaging*, pp. 213–216, Oct. 1998.