





Master's Thesis

# A Novel Methodology for Error-Resilient Circuits in Near-Threshold Computing

Jaemin Lee

Department of Electrical Engineering Graduate School of UNIST

2017



# A Novel Methodology for Error-Resilient Circuits in Near-Threshold Computing

Jaemin Lee

Department of Electrical Engineering

Graduate School of UNIST



# A Novel Methodology for Error-Resilient Circuits in Near-Threshold Computing

A thesis/dissertation submitted to the Graduate School of UNIST in partial fulfillment of the requirements for the degree of Master of Science

Jaemin Lee

6.12.2017

Approved by

Advisor Seokhyeong Kang



# A Novel Methodology for Error-Resilient Circuits in Near-Threshold Computing

Jaemin Lee

This certifies that the thesis/dissertation of Jaemin Lee is approved.

6.12.2017

signature

Advisor: Seokhyeong Kang

signature

Jaehyouk Choi

signature Seong-Jin Kim



## Abstract

The main goal of designing VLSI system is high performance with low energy consumption. Actually, to realize the human-related techniques, such as internet of things (IoTs) and wearable devices, efficient power management techniques are required. Near threshold computing (NTC) is one of the most well-known techniques which is proposed for the trade-off between energy consumption and performance improvement. With this technique, the solution would be selected by the lowest energy with highest performance.

However, NTC suffers a significant performance degradation, which is prone to timing errors. However, main goal of Integrated Circuit (IC) design is making the circuit to always operate correctly though worst-case condition. But, in order to make the circuit always work correctly, considerable area and power overheads may occur. As an alternative, better-than-worst-case (BTWC) design paradigm has been proposed. One of the main design of BTWC design includes error-resilient circuits which detect and correct timing errors, though they cause area and power overheads.

In this thesis, we propose various design methodologies which provide an optimal implementation of error-resilient circuits. Slack-based, sensitivity-based methodology and modified Quine-McCluskey (Q-M) algorithm have been exploited to earn the minimum set of error-resilient circuits without any loss of detection ability.

From sensitivity-based methodology, benchmark results show that the optimal designs reduces up to 46% monitoring area without compromising error detection ability of the initial error-resilient design.

From the Quine-McCluskey (Q-M) algorithm, benchmark results show that optimal design reduces up to 72% of flip-flops which are required to be changed to error-resilient circuits without compromising an error detection ability. In addition, more power and area reduction can be possible when reasonable underestimation of error detection ability is accepted. Monte-Carlo analysis validates that our proposed method is tolerant to process variation.



# Contents

I.	Introduction
II.	Related Works5
	2.1 Near-Threshold Computing (NTC) Technology
	2.2 Error-Resilient Technique
	2.2.1 Error Detection
	2.2.2 Error Prediction7
	2.2.3 Error Masking
	2.2.4 Resilient Design Optimization
	2.3 Algorithms for Logic Minimization10
III.	Sensitivity-Based Sorting Methodology12
	3.1 Experimental Setup
	3.2 Experimental Results
	3.2.1 Eight by Eight Multiplier14
	3.2.2 Benchmark Circuits15
IV.	Modified Quine-McCluskey Methodology20
	4.1 Experimental Setup
	4.2 Experimental Results
	4.2.1 Simulation Results with Voltage Scaling25
	4.2.2 Simulation Results with Underestimation of Error Detection Rate27
	4.2.3 Simulation Results with Considering Process Variability
V.	Conclusion



# **List of Figures**

Fig. 1-1 Energy and delay in different supply voltage operating regions [1]

Fig. 1-2 The architecture and timing diagram of Razor flip-flop [2]

Fig. 2-1 The architecture and timing diagram of DSTB [3]

Fig. 2-2 The architecture of Canary flip-flop [8]

Fig. 2-3 The architecture and timing diagram of TIMBER flip-flop [4]

Fig. 2-4 Steps of slack distribution of endpoint [19]

Fig. 3-1 Used error detection system

Fig. 3-2 Simulation flow of the experiment

**Fig. 3-3** AES Cipher: Error rate for various number of flip-flops swapping with the sorted flip-flops by slack-based methodology

Fig. 3-4 AES Cipher: Error rate for various number of flip-flops swapping with the sorted flip-flops by sensitivity-based methodology

**Fig. 3-5** JPEG Encoder: Error rate for various number of flip-flops swapping with the sorted flip-flops by slack-based methodology

**Fig. 3-6** JPEG Encoder: Error rate for various number of flip-flops swapping with the sorted flip-flops by sensitivity-based methodology

Fig. 4-1 Steps of the proposed design flow with modified Q-M method

Fig. 4-2 Simulation flow of the experiment



Fig. 4-3 AES Cipher: Simulation result with voltage scaling

Fig. 4-4 JPEG Encoder: Simulation result with voltage scaling

Fig. 4-5 MPEG2: Simulation result with voltage scaling

Fig. 4-6 AES Cipher: Underestimation by various Q-M reduction ratio

Fig. 4-7 JPEG Encoder: Underestimation by various Q-M reduction ratio

Fig. 4-8 MPEG2: Underestimation by various Q-M reduction ratio

Fig. 4-9 AES Cipher: Monte-Carlo simulation results

Fig. 4-10 MPEG2: Monte-Carlo simulation results



# **List of Tables**

 Table 2-1 Comparison of various techniques for timing error resilience [4]

Table 3-1 Experimental result with eight by eight multiplier

 Table 3-2 AES Cipher: Underestimation with two different sorting methodology at 0.80V

Table 3-3 JPEG Encoder: Underestimation with two different sorting methodology at 0.82V

Table 4-1 Benchmark information

 Table 4-2 AES Cipher: DSTB reduction with proposed algorithm under process variation

Table 4-3 MPEG2: DSTB reduction with proposed algorithm under process variation



# Nomenclature

ІоТ	Internet of Thing
NTC	Near Threshold Computing
IC	Integrated Circuit
BTWC	Better-Than-Worst-Case
Q-M	Quine-McCluskey
SoC	System-on-Chip
PVT	Process, Voltage and Temperature
CTS	Clock Tree Synthesis
DSTB	Double Sampling with a Time-Borrowing latch
TDTB	Transition Detector with a Time-Borrowing latch
TACD	Transistor-Aware Completion Detection
PEDFF	Phase-adjustable Error Detection Flip-Flop
DCFF	Delay-Compensation Flip-Flop
P&R	Place and Route
К-Мар	Karnaugh Map
SOP	Sum-Of-Product
POS	Product-Of-Sum
AES	Advanced Encryption Standard
JPEG	Joint Photographic Experts Group
SDF	Standard Delay Format
SDC	Synopsys Design Constraints
SAIF	Switching Activity Interchange Format



- SPEF Standard Parasitic Exchange Format
- MPEG2 Moving Picture Expert Group
- WC Worst Corner
- TC Typical Corner
- BC Best Corner



# **Chapter I**

## Introduction

A primary goal of designing VLSI System is achieving low energy consumption with high performance. Recently, human-related techniques, such as internet of things (IoTs) and wearable devices, have been proposed in low power system-on-chip (SoC) design. However, to foster these industries, efficient power management techniques are required.

Near-threshold computing (NTC) is one of the famous techniques for a well-balanced trade-off between delay and energy as shown in Fig. 1-1. However, in near-threshold region, process, voltage and temperature (PVT) variations cause significant performance degradation and timing errors, and thus they inhibit the use of NTC [1].

One of the main goal of IC design is make the circuit always operate correctly, even under the worst-case environment. However, when designing for worst-case condition, it may incur significant area and power overheads.



Fig. 1-1 Energy and delay in different supply voltage operating regions [1]



Alternatively, Better-than-worst-case (BTWC) design has been proposed. The goal of BTWC design is to remove some (or every) timing violations and make the circuit operate at a lower operation voltage or higher clock frequency for energy efficiency. One of the BTWC design is error detection and correction. In order to detect and correct these timing errors and make it feasible for lowering the operation voltage near NTC level (from 400 to 600 mV), error-resilient circuits are required for dependable operations. Error-resilient circuits detect timing violations by replacing the conventional flip-flops to error-resilient flip-flops (i.e. Razor flip-flops) and generate an error signal if timing errors are detected.



Fig. 1-2 The architecture and timing diagram of Razor flip-flop [2]



Razor flip-flop [2], the most well-known technique to detect and correct timing errors, is shown in Fig. 1-2. A Razor flip-flop is composed to a conventional flip-flop which is used as a main cell to transfer the data and a latch which is used as an additional cell for error detection. Then, the captured data of the flip-flop and the latch will be compared using a comparator. Also, additional delayed clock is required for correct data catch using a shadow latch. Between normal clock and delayed clock, timing difference exist and it is set to safety margin.

As shown in Fig. 1-2, main flip-flop catches the data at the positive edge of the normal clock (clock in Fig. 1-2). Then the shadow latch catches the data during the delayed clock (clock\_d in Fig. 1-2) is high. Then, the error signal becomes high when the captured data of the main flip-flop differ from those of the shadow latch. Then, the data will not be transferred to the next pipeline stage and the wrong data will be corrected to the correct data in the shadow latch by using instruction replay system. After the correction, the flip-flop will transfer the data to the next pipeline stage again.

However, as a Razor flip-flop requires a delayed clock, an additional clock port is required. Thus, it may require additional area for delayed clock generation. Also, it is difficult to maintain the safety margin during clock tree synthesis (CTS) stage. Razor flip-flop also contains a meta-stability issue since it uses a flip-flop as a main cell for data transfer to the next pipeline stage. To solve these issues, many different error-resilient circuits, such as double sampling with a time-borrowing latch (DSTB) and transition detector with a time-borrowing latch (TDTB), have been introduced [3].

By replacing the conventional flip-flop to the error-resilient flip-flop, it is possible to detect and correct timing errors of the circuits. However, error-resilient flip-flops generates power and area increase of the design as they require additional cells, such as a shadow latch and a comparator in Razor flip-flop. Therefore, it is required to reduce the number of flip-flops to be changed to error-resilient flip-flops for low overhead.

However, if suitable flip-flops are not changed to error-resilient flip-flops, some timing errors may not be detected. Then, the system would transfer the wrong data to the next pipeline stage and the output data will become different. Therefore, in order to select and change the necessary flip-flops to be changed to error-resilient flip-flops, a methodology to select proper flip-flops for the replacement is necessary.

In this thesis, we propose some new design methodologies to find the optimal number of required errorresilient circuits which are possible to detect timing errors of every operating cycle. Then, each proposed methodology has been applied in designing of several benchmark circuits to validate the efficiency of the idea. The contributions of our work are explained as follows.



- We propose new methodologies to find an optimal set of required error-resilient circuits by using various methods, such as sensitivity calculation and Quine-McCluskey method.
- We compare an underestimation of the optimized design from the initial design by lowering supply voltage.
- We compare the benefits of the optimized design in area and power from the initial design.

The remained parts of the thesis is composed as follows. Chapter II contains the related works about NTC technology, error-resilient systems and algorithms for logic minimization. Chapter III describes the simulation setup and results for sensitivity-based sorting methodology. Chapter IV shows the simulation setup and results of modified Quine-McCluskey methodology. Chapter V summarizes and concludes the thesis.



# **Chapter II**

## **Related Work**

In order to use an error-resilient system, NTC technology, error-resilient systems are required. Also, for low-overhead system, algorithms for logic minimization can be used to reduce the number of flip-flops. Related techniques and previous works are explained in this section.

#### 2.1 Near-Threshold Computing (NTC) Technology

NTC is a technology which lowers the operation voltage to the threshold voltage level of the device for optimal energy efficiency. However, significant problem of NTC technology is significant delay increase and timing errors by PVT variations. As shown in Fig. 1-1, lowering the voltage to the near-threshold region yields an energy reduction about 10X with compromising approximately 10X of performance degradation. Therefore, additional margin is required to solve the timing issues. [1]

However, as the technology scales down, the margin becomes increased to avoid performance degradation and thus, supply voltage becomes limited. So, error-resilient techniques are required for performance improvement.

#### 2.2 Error Resilient Technique

M. Choudhury et al. [4] compares various techniques for error resilience in three different categories which are error detection, prediction and masking. Table 2-1 simplifies the comparison among them.

Feature	Error Detection	Error Prediction	Error Masking		
Error Recovery	Rollback /	No error	No error		
Mechanism	Instruction Replay				
Sequential	Large	Large	Large		
Overhead	Large	Large	Large		
Combinational	Small	None	Small		
Overhead	Sinun	Tone	Silluit		
Tachniquas	Razor, Razor II, TACD,	Canary flip-flop, TRC,	PEDFF, DCFF,		
reciniques	TDTB, DSTB	Sensors	TIMBER		

Table 2-1 Comparison of various techniques for timing error resilience [4]



#### 2.2.1 Error Detection

As shown in the Table 2-1, one of the famous error detection techniques is Razor flip-flop. From the idea of Razor flip-flop, Razor II [5] and Transistor-Aware Completion Detection (TACD) circuit [6] have been proposed to achieve better throughput, area and energy. Razor II is the improved design of Razor flip-flop for power, energy and area overhead reduction. Also, TACD has been introduced to detect the activity of the outputs of the logic. This technique performs better than Razor flip-flop in throughput, area and energy. However, as these circuits transfer the data to the next pipeline stage using conventional flip-flop, data meta-stability issue occurs.



Fig. 2-1 The architecture and timing diagram of DSTB [3]



DSTB and TDTB [3] have been proposed to solve the data meta-stability issue. These circuits transfer the data to the next pipeline stage by using a latch as a main cell. Especially, DSTB uses one latch and one flip-flop as the same as Razor flip-flop as shown in Fig. 2-1. Also, as DSTB does not use a delayed clock, additional area and power reduction can be achieved.

Cross Edge Technique [7] has been proposed to change the checking window for different paths. Postedge checking window detects the timing violations at the critical path, but it may cause a race condition at short path. And pre-edge checking window causes the performance penalty at the critical path. Therefore, changing the checking window flexibly by analyzing the path is used.

In case of error detection, errors are not removed while detecting timing errors. Therefore, error recovery system (i.e. rollback or instruction replay) is required to correct the data.

#### 2.2.2 Error Prediction

Canary flip-flop [8] has been named from a canary in a coal mine as it helps to predict whether a timing error would occur or not. Fig. 2-2 shows the architecture of Canary flip-flop. Timing errors are predicted by comparing the result of main flip-flop with that of the Canary flip-flop, which detects a bit earlier than main flip-flop. When an error signal occurs (or is predicted), it triggers the voltage or frequency controller to prevent the timing error.

Stability checker design [9] predicts timing errors caused by a steady increase in delay by wearout and aging. Error prediction can also be possible by duplicating critical paths or timing errors on the duplicated paths [10]. However, this approach has some limitations since (i) the duplicated paths and critical paths in the design may work on different workloads or variations and (ii) the critical paths may change continuously.



Fig. 2-2 The architecture of Canary flip-flop [8]



By using error prediction circuit, it is possible to anticipate the timing error in advance. Then, the error signal controls the voltage or frequency before the circuit transfers the data to the next pipeline stage. Therefore, this system does not require additional error correction system for error recovery.

#### 2.2.3 Error Masking

Phase-adjustable error detection flip-flop (PEDFF) [11] masks the error temporally based on delaying the clock for one cycle after the timing error detection to correct the system. However, this technique is difficult to achieve high performance because of short cycle time, but long latency. In delay-compensation flip-flop (DCFF) [12], an edge detector detects timing errors which are occurred near the clock edge and the delayed clock is used to resample the data and correct the value of the data path by borrowing the time of the next pipeline stage.



Fig. 2-3 The architecture and timing diagram of TIMBER flip-flop [4]



TIMBER [4] masks timing errors by using the borrowed time from the next successful pipeline stage. Fig. 2-3 shows the architecture and the timing diagram of TIMBER flip-flop.

With these circuits, the system is not allowed to generate any timing errors. Thus, these circuits have an advantage that they do not require additional hardware supports, such as instruction replay or roll-back as timing errors do not exist.

#### 2.2.4 Resilient Design Optimization

With these error-resilient registers, some design optimization techniques have been suggested. Choudhury et al. [13] proposes a low overhead solution for masking timing errors on timing-critical paths in logic circuits. This methodology reduces the power and area overhead of the resilient designs. However, they have adjusted resilient techniques to speed-paths without considering the tradeoff of the benefits of error resilience and the costs of additionally inserted margin in data paths.



Fig. 2-4 Steps of slack distribution of endpoint [19]



Wan et al. [14] proposes a circuit optimization technique named DynaTune for timing speculation based on the dynamic behavior of a circuit. Liu et al. [15] proposes cost-effective circuit level re-synthesis solution to reduce area and energy consumption overhead by minimizing the number of flip-flops with timing errors. Also, Yuan et al. [16] proposes an in-situ timing error masking technique called InTimeFix. DynaTune has an advantage in throughput improvement and the others in area and power reduction, respectively. However, as these optimization techniques occur at the synthesis stage, these cannot cover the timing violation after place and route (P&R) stage. Then, this issue can degrade the solution quality.

Kahng et al. [17,18,19] also have proposed some design optimization techniques, recovery-driven design, power-aware slack redistribution, selective-endpoint optimization and clock skew optimization. Recovery-driven design technique [17] is a design approach which optimizes a processor module for a target timing error rate instead of exact operation. This methodology gives a significant power benefits. Both selective-endpoint optimization and clock skew optimization techniques are integrated in an iterative optimization flow which understands information of toggle rate and trade-off between cost of resilience and margin of the paths.

Fig. 2-4 illustrates the basic idea of this optimization approach. In the initial design, a lot of endpoints have timing errors at the target frequency. After selective-endpoint optimization, it is possible to optimize selected endpoints to reduce the resilience overheads. After clock skew optimization, it is possible to increase timing slacks of endpoints which have timing violations by controlling the arrival time of each endpoint. This flow achieves significant energy reduction compared to conventional design. By iteratively performing these techniques, it becomes possible to minimize the resilience cost.

#### 2.3 Algorithms for Logic Minimization

Many techniques for logic minimization have been proposed to simplify and optimize the gate-level implementation of a logic function. One of the most famous methods is Karnaugh Map (K-Map) method [20]. Karnaugh proposes a technique to simplify the Boolean expressions using a modified truth table to allow minimal sum-of-products (SOP) and product-of-sums (POS) expressions to be obtained. However, this technique is out of work when the number of variables are more than six.

Quine [21] proposes a way to simplify truth functions. McCluskey [22] simplifies and extends the method presented by Quine. This methodology also minimizes the Boolean functions. It is functionally identical to K-Map, but the tabular form makes it more applicable for computer algorithm. Also, it gives a deterministic way to find the minimum set of Boolean function.



The order of the Quine-McCluskey is separated in three parts. First, it is required to find all the implicants of the function. Then, we add them in a table. Finally, by using these prime implicants table, it is possible to find the essential prime implicants of the function that cover the function. This method is widely used as it is a viable methodology to handle a lot of variables.



# **Chapter III**

## Sensitivity-Based Sorting Methodology [24]

By changing the conventional flip-flop to the error-resilient flip-flop, it is possible to detect and correct timing errors of the circuits. Until recent works, every flip-flop has been replaced to error-resilient flip-flop to detect any error occurrence in many cycles. However, error-resilient flip-flops increase area and power of the design, and all the flip-flops do not required to be swapped into error-resilient flip-flops. In this chapter, we propose several design flows to insert the monitoring circuits.

To reduce the number of flip-flops, it is required to find which flip-flop should be optimized. Thus, sorting methodology should be chosen. In this chapter, we propose two different sorting methodologies, (i) sorting with timing slack of flip-flops in an ascending order, and (ii) sorting with the sensitivity of flip-flop in an ascending order. The sensitivity value is determined as

#### Sensitivity = Slack / Switching Activity,

where the flip-flop is more sensitive to timing errors when the switching activity of the flip-flop is small while the timing slack of the flip-flop is large. Thus, the flip-flop with low sensitivity contains higher probability to generate timing errors and it becomes better candidate to be swapped to error-resilient flipflop.



Fig. 3-1 Used error detection system



#### 3.1 Experimental Setup

Used error detection system for verification of the idea is shown in Fig. 3-1. Some or all flip-flops in the circuits are changed to DSTBs for error detection and correction. Error signals (i.e.  $Error[0] \sim Error[n-1]$ ) generated in each DSTB are compared using OR tree in the system and the system generates a signal named *'Error\_signal'*. Then, the signal controls the voltage or frequency regulator to raise the voltage or lower the frequency of the system.

Fig. 3-2 shows the implementation flow of our experiment. In order to verify the error detection system, designs have been generated in Verilog format and synthesized by TSMC 65-nm technology using *Synopsys Design Compiler* [26]. After synthesis has been finished, place & route (P&R) stage has been conducted to remove hold violations using hold buffer insertion with *Cadence SoC Encounter* [27]. Then, the generated netlist after P&R is used as a reference design of our further simulations. Then, gate-level simulation has been conducted using *Cadence NC-Sim* [28].



Fig. 3-2 Simulation flow of the experiment



To compare the performance of the system, Advanced Encryption Standard (AES) Cipher which is an encryption modules and Joint Photographic Experts Group (JPEG) Encoder which is an encoding module for image files, have been used as benchmark circuits. 530 flip-flops exist in AES Cipher and 4332 flip-flops exist in JPEG Encoder.

Gate delays of each design are obtained in the standard delay format (SDF) file generated using *Synopsys PrimeTime* [29]. SDF files are generated at different voltage range from 0.5V to 1.2V in 0.01-V increments. Thess delay files are used for further gate-level simulations. While conducting the simulations, outputs of the latch and the flip-flop are compared with XOR gates. Simulations have been run for 10,000 cycles with randomized input patterns. With various clock period constraints in the Synopsys design constraints (SDC) file, differently synthesized Verilog files are generated using *Synopsys Design Compiler* [26]. Optimal constraints can also be found by comparing various information such as area, power and operation voltage.

Switching activity values of each flip-flop can be found in the switching activity interchange format (SAIF) generated by *Synopsys PrimeTime* [29]. We first check whether the error bit of each DSTB (i.e. the output of the XOR gate in DSTB shown in Fig. 3-1) has been toggled or not. Then, if the DSTB toggles the error bit, we check how many times the error signal (i.e. Error[0]) has been toggled.

#### 3.2 Experimental Result

### 3.2.1 Eight by Eight Multiplier

For the functionality check of the proposed methodology, simple experiments are conducted with a simple eight by eight multiplier coupled with input to output flip-flops. Many different factors are evaluated with various clock constraints.

Clock Constraint (ns)		1.8			2.2			2.8		
Error Rate (%)	5	10	15	5	10	15	5	10	15	
Area ( <i>um</i> <sup>2</sup> )	1521			1518			1478			
Power ( <i>uW</i> )	89.4	83.4	80.9	91.3	89.2	83.1	93.9	89.5	86.0	
Voltage (V)	0.69	0.67	0.66	0.70	0.69	0.67	0.71	0.69	0.68	

Table 3-1 Experimental result with eight by eight multiplier



The implementation results using a simple multiplier are explained in Table 3-1. Tightly synthesized design will be generated with a small clock constraint (1.8 ns) and loosely synthesized when the clock constraint is large (2.8 ns). As target error rate increases, operation voltage and power decrease within the same constraints.

#### 3.2.2 Benchmark Circuits

AES Cipher and JPEG Encoder are used as benchmark circuits to prove the benefits of our method. Gatelevel simulations are conducted after changing flip-flops to DSTBs from 100% to 40% with sorting methodologies. DSTBs are inserted based on the timing slack only or sensitivity, which considers both timing slack and switching activity of each flip-flop in order to find an optimal operating point.

Area and power become smaller as the number of DSTBs is reduced in various number of flip-flops swapping cases. But, a further analysis is conducted to check the error detection ability of the designs having smaller number of DSTBs, since the errors can be ignored when DSTBs are not replaced from the conventional flip-flops.

For finding the tendency of the number of errors and checking the error detection ability of the designs with various number of DSTBs, simulations are conducted as shown in Fig. 3-2. Then, the number of errors are found in several cases of different number of DSTBs swapped with various sorting methodologies, slack-based methodology and sensitivity-based methodology.

Fig. 3-3 shows the experimental results of the benchmark, AES Cipher when the flip-flops are sorted in an ascending order of the slack. If the slack of the flip-flop is small, the flip-flop contains high possibility to generate a timing violation. Thus, the flip-flop is required to be changed to monitoring circuit for error detection and correction. Therefore, the flip-flop with smaller slack is changed first to the DSTB.

As shown in Fig. 3-3, the 40% monitoring case is not able to detect errors compared to the other swapping cases (i.e., 60~100% monitoring cases). When lower than 40% of all flip-flops are replaced to DSTBs in the design, the circuit is not able to detect timing errors correctly and underestimates the error occurrence at the same voltage. Also, though there exist small discrepancies from the 100% monitoring case, error detection ability seems quite similar though reducing up to 54% monitoring case.

As explained before, switching activities of the flip-flops are major parameters to analyze the functional errors of the flip-flops. For selecting suitable flip-flops which are more sensitive to the functional error, we propose the sensitivity-based sorting methodology.





Fig. 3-3 AES Cipher: Error rate for various number of flip-flops swapping with the sorted flip-flops by slack-based methodology



Fig. 3-4 AES Cipher: Error rate for various number of flip-flops swapping with the sorted flip-flops by sensitivity-based methodology



Simulation results with sensitivity-based sorting methodology are shown in Fig. 3-4. Flip-flops are sorted in an ascending order of the sensitivity. Lower sensitivity indicates that the slack is smaller, while the flip-flop switches more often. Thus, swapping the flip-flops with lower sensitivity will show better result of an optimal design.

As shown in Fig. 3-4, error rates are the same as the 100% monitoring case until 54% monitoring case. Therefore, 54% monitoring swapping case shows the most area and power savings without any error detection ability loss compared to 100% monitoring case. Thus, it becomes possible to reduce up to 46% of flip-flop area with the sensitivity-based methodology with an optimal number of flip-flop swapping.

Table 3-2 shows the underestimation of the error rate at a 0.80V supply voltage when sorting by slack and sorting by sensitivity for the benchmark, AES Cipher. As shown, by using sensitivity-based sorting methodology, it is possible to lose less underestimation than using slack-based sorting methodology.

	Number of flip-flop swapping								
Swapping percentage	100%	60%	54%	50%	40%				
Slack-based	0%	0%	5.97%	17.24%	64.28%				
Sensitivity-based	0%	0%	0%	12.36%	41.01%				

# Table 3-2 AES Cipher: Underestimation withtwo different sorting methodology at 0.80V

Simulated error rates of the benchmark, JPEG Encoder, are plotted in Fig. 3-5 and Fig. 3-6 for slackbased methodology and sensitivity-based methodology, respectively. As shown in Fig. 3-5, until reducing the number of flip-flop swapping to 70%, the design provides the same error detection ability with 100% monitoring case. However, the system cannot detect timing errors properly when less than 50% of flip-flops are swapped to DSTBs.

As shown in Fig. 3-6, sensitivity-based methodology shows similar trend with slack-based methodology in this benchmark circuit. With the sensitivity-based methodology, up to 70% monitoring case shows the same error detection ability with 100% monitoring case in almost every voltage. However, the system also cannot detect timing error properly when less than 50% of flip-flops are replaced to DSTBs like slack-based sorting methodology. Therefore, it is possible to reduce up to 30% in flip-flop area by the proposed method with an optimal number of flip-flop swapping.





Fig. 3-5 JPEG Encoder: Error rate for various number of flip-flops swapping with the sorted flip-flops by slack-based methodology



Fig. 3-6 JPEG Encoder: Error rate for various number of flip-flops swapping with the sorted flip-flops by sensitivity-based methodology



### Table 3-3 JPEG Encoder: Underestimation with

	Number of flip-flop swapping								
Swapping percentage	100%	70%	60%	56%	50%				
Slack-based	0%	0%	10.46%	22.18%	36.17%				
Sensitivity-based	0%	0%	9.82%	22.04%	24.07%				

### two different sorting methodology at 0.82V

Table 3-3 shows the underestimation of the error rate at a 0.82V supply voltage when sorting by slack and sorting by sensitivity for the benchmark, JPEG Encoder. As shown, it is possible to reduce up to 70% case without compromising an error detection ability for both slack-based and sensitivity-based sorting methodology. However, when some underestimation is allowed, sensitivity-based sorting methodology shows less underestimation compared to slack-based sorting methodology (i.e. under 60% cases).



## **Chapter IV**

### Modified Quine-McCluskey Methodology [25]

When we design, and synthesize the circuits at lower voltage, all timing-critical flip-flops are replaced to error-resilient registers (DSTBs) to detect any errors occurred in each clock cycle. However, some of them are redundant with only causing power and area overheads. In this chapter, we propose a new design flow coupled with a method which finds an optimal set of required error-resilient registers. The optimization method to find the minimum set of error-resilient flip-flops is similar to previous Q-M Boolean function minimization [21, 22].

Algorithm 4-1 describes the pseudo-code of the modified Q-M method to find the minimum set of required DSTBs. Also, Fig. 4-1 shows a simple example for the functional flow of the algorithm. When the gate-level simulation with the Verilog file, in which all the flip-flops are replaced to DSTBs, is finished, errors of each flip-flop are obtained as binary patterns at every cycle (e.g., '1' means error and '0' means non-error). Then, we generate a *#CYCLE*×*#REG* table, where *#CYCLE* is the number of simulation cycles and *CYCLE* is a set of clock cycles. *#REG* is the number of error-occurred registers, while *REG* is a set of the registers. We mark each cell of the table with a corresponding error signal (Line 1-6).

Generated error table is denoted by error[cycle][reg] and each table cell has corresponding error signal data ('0' or '1' value) (Line 7, ①). Then, sum of the values in error[cycle][reg] for each clock cycle (cycle) and register (reg) are saved to  $total\_error[cycle]$  and  $total\_error[reg]$ , respectively. They are used to sort and compare the data (Line 8-15, ②) later. After completing the summation, we sort the *CYCLE* by the  $total\_error[cycle]$  in an ascending order using Insertion-Sort Algorithm [23] (Line 16, ③).

Whenever *total\_error[cycle*] is equal to 1, we search the error-occurred registers (i.e., columns marked '1' in the table, *reg*) in that cycle. Then, we generate the output set *sol* and save the *reg* (R5 in Figure 2, ④) to *sol* (Lines 17-23, ④). If the *total\_error[cycle*] is bigger than 1, we check whether at least one error of the cycle is covered by error-resilient registers in the output. This step starts from the bottom cycle (a cycle with the highest error rate) (Lines 24-39, ⑤). If covered, we skip the current cycle and move to the next cycle (Lines 28-30, ⑦). If not covered, we then compare the *total\_error[cycle*] for every error-resilient register with *error[cycle][reg]* is '1' at that cycle (Line 32, ⑤). Finally, we save the error-resilient register with the highest *total\_error[reg]* to *sol* and move to the next cycle (Lines 31-36, ⑥). When all the algorithms are conducted, the final list in *sol* becomes the set of required error-resilient registers.



Al	gorithm 4-1: Modified Q-M based Minimization
1	Generate a table of <i>#CYCLE</i> × <i>#REG</i> .
2	for each cycle in CYCLE do
3	for each reg in REG do
4	Set every <i>error</i> [ <i>cycle</i> ][ <i>reg</i> ] $\leftarrow 0$ ;
5	end
6	end
7	Put the error data to each <i>error</i> [ <i>cycle</i> ][ <i>reg</i> ];
8	Generate a set: total error[reg] $\leftarrow 0$ ;
3	
10	for each cycle in CYCLE do
11	Calculate <i>total_error</i> [ <i>cycle</i> ];
12	end for each reg in <i>BEC</i> do
13	lor each reg in REG do
14	Calculate <i>total_error</i> [ <i>reg</i> ];
15	end
16	Sort the CYCLE in an ascending order of total_error[cycle] using Insertion-Sort
	Algorithm;
17	for each cycle in CVCLE do
10	if $total error[cycle] = 1$ then
17	$\Gamma$ 1 d $\Gamma$ 1 d $\Gamma$
20	Find the <i>reg</i> with 1 in the <i>cycle</i> ;
21	$sol \leftarrow sol \cup leg bleak,$
22	and
23	
24	for each cycle in CYCLE do
25	$\int \frac{1}{10} \frac{10}{10} 1$
20	
27	Check whether at least one error of the cycle is covered by <i>sol</i> ;
28	break:
30	end
31	else
32	Compare <i>total_error</i> [ <i>reg</i> ] for <i>reg</i> with <i>error</i> [ <i>cycle</i> ][ <i>reg</i> ] =
33	Find the <i>reg</i> with the highest <i>total error</i> [ <i>reg</i> ];
34	$sol \leftarrow sol \cup reg$
35	break;
36	end
37	end
38	end
39	end



SCIENCE AND TECHNOLOGY

					RI	EG			2	R1	R2	R3	R4	R5	R6	RT
	í		R1	R2	R3	R4	R5	R6	C1	1	0	0	1	0	1	3
	C	1	1	0	0	1	0	1	C2	1	1	1	0	0	1	4
	C	2	1	1	1	0	0	1	C3	0	0	1	1	0	0	2
E	C	3	0	0	1	1	0	0	C4	0	0	0	0	1	0	1
CL	C	4	0	0	0	0	1	0	C5	0	0	0	1	0	1	2
C	C	5	0	0	0	1	0	1	C6	1	1	1	0	0	0	3
	C	6	1	1	1	0	0	0	C7	0	0	1	1	0	0	2
	C	7	0	0	1	1	0	0	СТ	3	2	4	4	1	3	
		R1	R2	R3	R4	R5	R6	RT	4	<b>R1</b>	R2	R3	R4	R5	R6	RT
С	4	0	0	0	0	1	0	1	C4	0	0	0	0	1	0	1
С	3	0	0	1	1	0	0	2	C3	0	0	1	1	0	0	2
C	5	0	0	0	1	0	1	2	C5	0	0	0	1	0	1	2
C	7	0	0	1	1	0	0	2	C7	0	0	1	1	0	0	2
C	1	1	0	0	1	0	1	3	C1	1	0	0	1	0	1	3
C	6	1	1	1	0	0	0	3	C6	1	1	1	0	0	0	3
C	2	1	1	1	0	0	1	4	C2	1	1	1	0	0	1	4
C	T	3	2	4	4	1	3		СТ	3	2	4	4	1	3	
_	_			_	_											
(		R1	R2	R3	R4	R5	R6	RT	6	<b>R1</b>	R2	R3	R4	R5	R6	RT
C	4	0	0	0	0	1	0	1	C4	0	0	0	0	1	0	1
C	3	0	0	1	1	0	0	2	C3	0	0	1	1	0	0	2
C	5	0	0	0	1	0	1	2	C5	0	0	0	1	0	1	2
C	7	0	0	1	1	0	0	2	C7	0	0	1	1	0	0	2
C	1	1	0	0	1	0	1	3	C1	1	0	0	1	0	1	3
C	6	1	1	1	0	0	0	3	C6	1	1	1	0	0	0	3
C	2	1	1	1	0	0	1	4	C2	1	1	1	0	0	1	4
C	T	3	2	4	4	1	3		CT	3	2	4	4	1	3	
6		D1	Da	<b>D2</b>	D4	D.	D(	DT		D1	D2	D2	D4	D.5	D	рт
			R2 0	К <b>Э</b>	R4 0	К5 1	K0 0	1			R2 0	R3 0	R4	1 K5	Ro	
	3	0	0	1	1	1	0	2	C3	0	0	1	1	0	0	2
$\frac{c}{c}$	5	0	0	0	1	0	1	2	C5	0	0	0	1	0	1	2
$\overline{c}$	7	0	0	1	1	0	0	2	C7	0	0	1	1	0	0	2
c	1	1	0	0	1	0	1	3	C1	1	0	0	1	0	1	3
C	6	1	1	1	0	0	0	3	C6	1	1	1	0	0	0	3
С	2	1	1	1	0	0	1	4	C2	1	1	1	0	0	1	4
C	т	3	2	4	4	1	3		СТ	3	2	4	4	1	3	
_		_														
(		R1	R2	R3	R4	R5	R6	RT	(10)	R1	R2	R3	R4	R5	R6	RT
C	4	0	0	0	0	1	0	1	C4	0	0	0	0	1	0	1
C	3	0	0	1	1	0	0	2	C3	0	0	1	1	0	0	2
C	=	0	0	0	1	0	1	2	C5	0	0	0	1	0	1	2
and the second se	3	-				_	_		C7		0					
C	7	0	0	1	1	0	0	2	 01	0	0	1	1	0	0	2
C C	5 7 1	0	0	1 0	1	0	0	2	C1	0 1	0	1 0	1	0	0	2
C C C	5 7 1 6	0 1 1	0 0 1	1 0 1	1 1 0	0 0 0	0 1 0	2 3 3	C1 C1	0 1 1	0 0 1	1 0 1	1 1 0	0 0 0	0 1 0	2 3 3
C C C C	5 7 1 6 2	0 1 1 1	0 0 1 1	1 0 1 1	1 1 0 0	0 0 0 0	0 1 0 1	2 3 3 4	C1 C6 C2	0 1 1 1	0 0 1 1	1 0 1 1	1 1 0 0	0 0 0 0	0 1 0 1	2 3 3 4

Fig. 4-1 Steps of the proposed design flow with modified Q-M method



#### 4.1 Experimental Setup

Fig. 4-2 shows the implementation and simulation flow of the proposed method. In order to validate our proposed idea, each design has been implemented in Verilog and synthesized with commercial 65-nm cell library using *Synopsys Design Compiler* [26]. After synthesis, timing-critical flip-flops have been changed to DSTBs. P&R stage has been conducted to get rid of hold violations with insertion of hold buffers by using *Cadence SoC Encounter* [27].



Fig. 4-2 Simulation flow of the experiment

With replaced standard parasitic exchange format (SPEF) and Verilog netlist of each design, we extract the standard delay format (SDF) using *Synopsys PrimeTime* [29]. SDF files are extracted at various voltage range from 0.5 V to 1.2 V in 10-mV increments. Then, these files are used in the gate-level simulations to calculate the gate delays of each voltage.



Gate-level simulations are performed using *Cadence NC-Sim* [28]. Each simulation is conducted for 10,000 cycles to find the optimum set of flip-flops by applying our proposed method. To evaluate the performance of the proposed method, three benchmarks are used, which are Advanced Encryption Standard (AES) Cipher, Joint Photographic Experts Group (JPEG) Encoder and Moving Picture Expert Group (MPEG2). The information of each benchmark is shown in Table 4-1.

Benchmark	# of cells	# of flip-flops	Area (µm <sup>2</sup> )
AES Cipher	21,612	530	42,727
JPEG Encoder	86,858	4,332	194,336
MPEG2	13,042	2,948	54,130

<b>Table 4-1</b>	Benchmark	inform	ation
------------------	-----------	--------	-------

For considering the global process variations, we characterize three different corners for each voltage, which are worst corner (WC), typical corner (TC) and best corner (BC). WC library is used in the synthesis stage and placement stage to satisfy all conditions when conducting a gate-level simulation. Then, we generate SDF files with typical corner library for nominal case.

By adopting various SDF files to each benchmark design, the error rate is measured at each operation voltage. The error signals of each flip-flop are extracted to binary format, which will be used for our modified Q-M method. From the Q-M method, we find an optimal set of error-resilient registers, and update the initial Verilog file which has been generated by synthesis. To control the voltage, by using the detected errors, a single error signal is required. So, we insert an OR tree before the place and route stage.

Place and route stage is performed again with a replaced Verilog file. Then, SPEF and SDF files are updated for the new design. Finally, to validate the correctness and efficiency of the new design, gate-level simulations are performed for 1,000,000 cycles to measure the error rate. Underestimation of the errors is compared between the design which are implemented with DSTBs for every error-occurred flip-flop and the optimized design in which only the selected flip-flops are changed to DSTBs.

#### 4.2 Experimental Results

We use AES Cipher, JPEG Encoder and MPEG2 as benchmark circuits, and DSTB is used as an errortolerant register to validate benefits of our proposed method. The clock period is set to 0.92 ns, 1.10 ns, and 0.84 ns for AES Cipher, JPEG Encoder and MPEG2, respectively. Clock duty cycle needs to be optimized to remove the errors at the nominal voltage (0.90 V) for DSTB technique, in which the latch transfers the input data to the next pipeline stage. Thus, we analyze the clock and find an optimal duty cycle by reducing the high duration, while maintaining the clock period.



Q-M minimization method may become meaningless at higher error rate, because errors at some flipflops may cover almost all errors occurrence at specific cycles. Therefore, we need to specify the reasonable region for accurate analysis. We analyze every simulation result for each benchmark and determine that the reasonable error rate is lower than 40% of for both AES Cipher and JPEG Encoder and less than 5% for MPEG2 module.

#### 4.2.1 Simulation results with voltage scaling

Fig. 4-3, 4-4 and 4-5 show total number of DSTBs and reduced number of DSTBs from the proposed Q-M method as the supply voltage scales for each benchmark, AES Cipher, JPEG Encoder and MPEG2, respectively. Error rates are shown in blue lines for each operation voltage. As expected, more DSTBs are required and the proposed Q-M method eliminates more redundant DSTBs at lower voltage without affecting error detection ability.

In AES Cipher, we can remove about 30% of flip-flops without compromising an error detection ability at low voltage (e.g., 0.83 V). However, at the higher voltage with reasonably small error rate (e.g., <10%), it is impossible to reduce the number of flip-flops by Q-M method as the number of error-occurred flip-flops is small (less than 10). For JPEG Encoder and MPEG2, 44% and 70% of flip-flops, respectively, do not need to be replaced to DSTBs without compromising an error detection ability at low voltage (0.83 V for JPEG Encoder and 0.71 V for MPEG2).



Fig. 4-3 AES Cipher: Simulation result with voltage scaling





Fig. 4-4 JPEG Encoder: Simulation result with voltage scaling



Fig. 4-5 MPEG2: Simulation result with voltage scaling



#### 4.2.2 Simulation results with underestimation of error detection rate

By using of the proposed Q-M method, we can reduce the number of DSTBs significantly for JPEG Encoder and MPEG2 modules. To make a further reduction of DSTBs, we allow small underestimation of error detection rate. We sort the list of flip-flops in an order of the number of errors and replace flip-flops which have more errors to DSTBs.

We sweep the percentage of replacements from 100% to 60% with 10% steps. The amount of underestimation, area, and power are measured from 1,000,000 cycles of simulation. For the simulation, we choose voltage levels which show the best results with reasonable error rates from the Fig. 4-4 and 4-5. 0.83 V for JPEG Encoder and 0.71 V for MPEG2 module are used.

Fig. 4-6, 4-7 and 4-8 show the underestimation, normalized area, and power for various Q-M replacement ratios. The 'without Q-M' means that all error occurred flip-flops are replaced to DSTBs and the 'with Q-M' means that all the flip-flops selected by the proposed Q-M reduction method are replaced.

Fig. 4-6 shows the simulation results of the benchmark, AES Cipher. As shown, by reducing the number of DSTBs, underestimation of the errors increases. For example, when we reduce the number of DSTBs to 80% of Q-M method, we can reduce 0.6% of area and 9.0% of power with 1.4% of underestimation increment compared to the original design (i.e., withoug Q-M). At 60% of Q-M, we can save 1.1% of area and 10.0% of power with 4.0% of underestimation from the initial design.



Fig. 4-6 AES Cipher: Underestimation by various Q-M reduction ratio



For JPEG Encoder shown in Fig. 4-7, the underestimation is increased by reducing the number of DSTBs. When we reduce the number of DSTBs to 80% of Q-M method, 0.9% of area and 3.4% of power can be reduced with 1.5% of underestimation compared to the original design. At 60% of Q-M, we can reduce 1.1% of area and 4.9% of power with 3.2% of underestimation from the original design.



Fig. 4-7 JPEG Encoder: Underestimation by various Q-M reduction ratio



Fig. 4-8 MPEG2: Underestimation by various Q-M reduction ratio



Fig. 4-8 represents the simulation results of the benchmark, MPEG2. As the Q-M replacement ratio is reduced according to the sorting, more benefit in area and power reduction is obtained. It means that all candidate flop-flops from the proposed method have more significance than the other benchmarks. So by reducing the number of DSTBs, the amount of the underestimation increases rapidly. We can reduce up to 2.4% of area and 5.2% of power with 10.2% of underestimation compared to the original design, when we reduce the number of DSTBs to 80% of Q-M. Therefore, in case of MPEG2, as the underestimation of error rate is too high even for only 90% Q-M, it is not acceptable to reduce the number of DSTBs anymore.

#### 4.2.3 Simulation results with considering process variability

Up to above experiments, we have implemented the circuits with a worst corner library and conducted simulations with a typical corner library. However, as the circuit does not necessary operate in the typical corner always, it is required to consider process variations and timing uncertainties. Thus, Monte-Carlo analysis is conducted to validate the correctness and the efficiency of the proposed method.

To insert the randomized delays for each cell, we use Gaussian random function with a three-sigma rule by reading the nominal delay values of the corners (i.e., best and worst corner). A thousand of SDF files are generated with the Gaussian random function for each voltage level and used in the gate-level simulations explained in the previous section. The distribution of the error rates for each voltage is investigated.

When we include the Monte-Carlo delays in the simulations, it is impossible to operate at the voltage that we have used in the previous simulations. So, it is required to raise the operation voltage to make the circuits operate under the variations. As we use a dynamic voltage control function and a voltage (or frequency) regulator, the operation voltage (or frequency) can be dynamically controlled based on the operating conditions and the error rates of the circuits.

Fig. 4-9 shows the gate-level simulation results of AES Cipher with 1,000 random SDF files. The minimum (Min), average (Avg), and maximum (Max) error rates for each operation voltage range from 1.02 V to 1.05 V are shown. The number of DSTBs and the reduction by the proposed methodology are summarized in Table 4-2. The results of MPEG2 are shown in Fig. 4-10 and Table 4-3. We sweep the voltage from 0.85 V to 0.88 V for this benchmark. With our proposed method, we can achieve up to 7.2% and 78.1% of DSTB reduction for AES Cipher and MPEG2, respectively, without any additional design cost. From the results, we can see that our proposed methodology is acceptable under huge process uncertainty.





Fig. 4-9 AES Cipher: Monte-Carlo simulation results



Fig. 4-10 MPEG2: Monte-Carlo simulation results



Operation		1.02		1.03				
Voltage (V)		1.02						
Casa	Total	DSTB from	Reduction	Total	DSTB from	Reduction		
Case	DSTB	Q-M	(%)	DSTB	Q-M	(%)		
Min	77	73	5.2	40	39	2.5		
Avg	87	83	4.6	72	68	5.6		
Max	97	90	7.2	88	83	5.7		
Operation		1.04			1.05			
Operation Voltage (V)		1.04			1.05			
Operation Voltage (V)	Total	1.04 DSTB from	Reduction	Total	1.05 DSTB from	Reduction		
Operation Voltage (V) Case	Total DSTB	1.04 DSTB from Q-M	Reduction (%)	Total DSTB	1.05 DSTB from Q-M	Reduction (%)		
Operation Voltage (V) Case Min	Total DSTB 14	1.04 DSTB from Q-M 14	Reduction (%) 0.0	Total DSTB 4	1.05 DSTB from Q-M 4	Reduction (%) 0.0		
Operation Voltage (V) Case Min Avg	Total DSTB 14 52	1.04 DSTB from Q-M 14 49	Reduction (%) 0.0 5.8	Total DSTB 4 19	1.05 DSTB from Q-M 4 18	Reduction (%) 0.0 5.3		

## Table 4-2 AES Cipher: DSTB reduction with proposed algorithm under process variation

## Table 4-3 MPEG2: DSTB reduction with proposed algorithm under process variation

Operation Voltage (V)		0.85		0.86				
voltage (v)								
Case	Total	DSTB from	Reduction	Total	DSTB from	Reduction		
Case	DSTB	Q-M	(%)	DSTB	Q-M	(%)		
Min	88	22	75.0	64	14	78.1		
Avg	106	39	63.2	53	27	49.1		
Max	126	51	59.5	64	20	68.7		
Operation		0.87			0.99			
Voltage (V)		0.87		0.88				
Case	Total	DSTB from	Reduction	Total	DSTB from	Reduction		
Case	DSTB	Q-M	(%)	DSTB	Q-M	(%)		
Min	7	5	28.6	0	0	-		
Δυα			<b>a a a</b>	22	11	50.0		
Avg	17	13	23.5	22	11	30.0		



# Chapter V Conclusion

The main goal of designing VLSI system is high performance with low energy consumption. Actually, to realize the human-related techniques, such as internet of things (IoTs) and wearable devices, efficient power management techniques are required. Though near threshold computing (NTC) helps to find the optimal solution, this technique may cause timing errors. In order to detect and correct timing errors, error-resilient circuit is required. However, as error-resilient circuit causes area and power overhead, it is required to reduce the number of flip-flops to be swapped to error-resilient flip-flop.

So, in this thesis, we propose a low-overhead error-resilient system and various design methodologies to find an optimal number of error-resilient circuits, such as DSTB in NTV operations. We propose two different sorting methodologies, sensitivity-based sorting algorithm and modified Quine-McCluskey (Q-M) sorting algorithm.

We use DSTB as an error-resilient circuit for timing error detection to find an optimal design of the system. When we use sensitivity-based sorting algorithm, this algorithm is able to reduce about 46% of flip-flop area compared to the conventional error-resilient design which every flip-flop has been changed to DSTB. Also, by using modified Quine-McCluskey sorting algorithm, we can reduce up to 72% of error-resilient flip-flops without compromising an error detection ability. The elimination of the redundant circuits leads to the power and area savings without any additional design changes.

In addition, when we admit some underestimation in error detection rate, it is possible to reduce more area and power with slight increase of the underestimation of error detection ability. Monte-Carlo analysis shows the reliability and efficiency of the proposed methods under the large process variations.



## REFERENCES

- R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester and T. Mudge, "Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits," in Proc. IEEE, pp. 253-266, 2010.
- D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner and T. Mudge, "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," in Proc. IEEE/ACM Symposium on Microarchitecture (MICRO-36), pp. 7-18, 2003.
- K. A. Bowman, J. W. Tschanz, N. S. Kim, J. C. Lee, C. B. Wilkerson, S.-L. L. Lu, T. Karnik and V. K. De, "Energy-Efficient and Metastability-Immune Resilient Circuits for Dynamic Variation Tolerance," IEEE Journal of Solid-State Circuits (JSSC) 44(1) (2009), pp. 49-63.
- M. Choudhury, V. Chandra, K. Mohanram and R. Aitken, "TIMBER: Time Borrowing and Error Relaying for Online Timing Error Resilience," in Proc. IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1554-1559, 2010.
- S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull and D. T. Blaauw, "Razor2: In Situ Error Detection and Correction for PVT and SER Tolerance," in Proc. IEEE International Solid-State Circuits Conference (ISSCC), pp. 400-401, 2008.
- J. Crop, R. Pawlowski and P. Chiang "Regaining Throughput Using Completion Detection for Error-Resilient, Near-Threshold Logic," in Proc. IEEE/ACM Design Automation Conference (DAC), pp. 974-979, 2012.
- C-H. Chen, Y. Tao and Z. Zhang, "Efficient In Situ Error Detection Enabling Diverse Path Coverage," in Proc. IEEE International Symposium on Circuits & Systems (ISCAS), 2013, pp. 773-776.
- T. Sato and Y. Kunitake, "A Simple Flip-Flop Circuit for Typical-Case Designs for DFM," in Proc. International Symposium on Quality Electronics Design (ISQED), pp. 539-544, 2007.
- M. Agarwal, B. C. Paul, M. Zhang and S. Mitra, "Circuit Failure Prediction and its Application to Transistor Aging," in Proc. VLSI Test Symposium (VTS), pp. 277-286, 2007.



- 10. K. Bowman, J. Tschanz, C. Wilkerson, S.-L. Lu, T. Karnik, V. De, S. Borkar, "Circuit techniques for dynamic variation tolerance," in Proc. Design Automation Conference (DAC), pp. 4–7, 2009.
- M. Kurimoto, H. Suzuki, R. Akiyama, T. Yamanaka, H. Ohkuma, H. Takaha and H. Shinohara, "Phase-Adjustable Error Detection Flip-Flops with 2-Stage Hold Driven Optimization and Slack Based Grouping Scheme for Dynamic Voltage Scaling," in Proc. Design Automation Conference (DAC), pp. 884-889, 2008.
- K. Hirose, Y. Manzawa, M. Goshima and S. Sakai "Delay-Compensation Flip-Flop with In-situ Error Monitoring for Low-Power and Timing-Error-Tolerant Circuit Design," Japanese Journal of Applied Physics, 47(4S), 2008, pp. 2779-2787.
- 13. M. R. Choudhury and K. Mohanram, "Masking Timing Errors on Speed-Paths in Logic Circuits," in Proc. IEEE Design, Automation & Test in Europe (DATE), pp. 87-92, 2009.
- L.Wan and D. Chen, "DynaTune: Circuit-Level Optimization for Timing Speculation Considering Dynamic Path Behavior," in Proc. IEEE International Conference on Computer-Aided Design (ICCAD), pp. 172-179, 2009.
- 15. Y. Liu, F. Yuan and Q. Xu, "Re-synthesis for cost-efficient circuit-level timing speculation," in Proc. IEEE/ACM Design Automation Conference (DAC), pp. 158-163, 2011.
- F. Yuan and Q. Xu, "InTimeFix: A Low-Cost and Scalable Technique for In-Situ Timing Error Masking in Logic Circuits," in Proc. IEEE/ACM Design Automation Conference (DAC), pp. 1-6, 2013.
- A. B. Kahng, S. Kang, R. Kumar and J. Sartori, "Recovery-Driven Design: A Methodology for Power Minimization for Error Tolerant Processor Modules," in Proc. IEEE/ACM Design Automation Conference (DAC), pp. 825-830, 2010.
- A. B. Kahng, S. Kang, R. Kumar and J. Sartori, "Slack Redistribution for Graceful Degradation Under Voltage Overscaling," in Proc. Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 825-831, 2010.
- A. B. Kahng, S. Kang and J. Li, "A New Methodology for Reduced Cost of Resilience," in Proc. Great Lakes Symposium on VLSI (GLSVLSI), pp. 157-162, 2014.



- 20. M. Karnaugh, "The Map Method for Synthesis of Combinational Logic Circuits," American Institute of Electrical Engineers (AIEE), 72(5) (1953) pp. 593-598.
- W. V. Quine, "A Way to Simplify Truth Functions," American Mathematical Monthly, 62(9) (1955) pp. 627-631.
- E. J. McCluskey, "Minimization of Boolean Functions," J. Bell System Technical, 35(6) (1956) pp. 1417-1444.
- R. L. Rivest and C. E. Leiserson, "Introduction to Algorithm," McGraw-Hill, Inc., New York, NY, USA, 1990.
- 24. J. Lee, S. Kim, Y. Kim and S. Kang, "An Optimal Operating Point By Using Error Monitoring Circuits with An Error-Resilient Technique," in Proc. IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), pp. 69-73, 2015
- J. Lee, S. Kim, Y. Kim and S. Kang, "A Novel Design Methodology for Error-Resilient Circuits in Near-Threshold Computing," in Proc. IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), pp. 1-4, 2016
- 26. "Synopsys Design Compiler User Guide"
- 27. "Cadence Encounter User Guide"
- 28. "Cadence NCVerilog User Guide"
- 29. "Synopsys PrimeTime User Guide"



## Acknowledgement

I really appreciate Prof. Seokhyeong Kang and Prof. Youngmin Kim for completion of my master thesis. Their advice, teaching and encouragement guided for better research result and quality. Also, I really appreciate the members in System-on-Chip Design Lab. Without them, it might be impossible to enjoy researching in the lab. I also appreciate committee members of my thesis defense, Prof. Jaehyouk Choi and Prof. Seong-Jin Kim.

Also without my loving EA United members and my friends, I couldn't enjoy the life in UNIST. Finally I really appreciate my parents and my sister. Their cheer and belief always made me to be encouraged.

The materials in this thesis is based on the following publications

- Chapter 3 is based on:
  - Jaemin Lee, Seungwon Kim, Youngmin Kim and Seokhyeong Kang, "An Optimal Operating Point By Using Error Monitoring Circuits with An Error-Resilient Technique," in Proc. IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), pp. 69-73, 2015
- Chapter 4 is based on:
  - Jaemin Lee, Sunmean Kim, Youngmin Kim and Seokhyeong Kang, "A Novel Design Methodology for Error-Resilient Circuits in Near-Threshold Computing," in Proc. IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), pp. 1-4, 2016