6-1-2018

Power efficient resilient microarchitectures for PVT variability mitigation

Shady Agwa

THE AMERICAN UNIVERSITY IN CAIRO
الجـامـعـة الأمـريكـيـة بالقـاهـرة

Power Efficient Resilient Microarchitectures for PVT Variability Mitigation

A Thesis submitted in partial fulfillment of the requirement for the degree of PhD.

Electronics & Communications Engineering ECNG, School of Science & Engineering SSE, The American University in Cairo AUC.

By:

Shady Onsey Haleem RizkAlla Agwa

shady_agwa@aucegypt.edu

January 2018

The American University in Cairo

School of Science & Engineering

Power Efficient Resilient Microarchitectures for PVT Variability Mitigation

A thesis submitted by

Shady Onsey Haleem RizkAlla Agwa

Submitted to

The Department of Electronics and Communications Engineering

In Partial Fulfillment of the requirements for

The Degree of Doctor of Philosophy

Dr. Yehea Ismail Date
Thesis Supervisor
Professor at the American University in Cairo, CND Director

Dr. Eslam Yahya Date
Thesis Co-Advisor
Visiting Professor at the Ohio State University

Dr. Ahmed Abou-Auf Date
Internal Examiner
Professor at the American University in Cairo

Dr. Mohamed Shalan Date
Internal Examiner
Associate Professor at the American University in Cairo

Dr. Hani Fikry Ragai Date
External Examiner
Emeritus Professor at Ain Shams University

Dr. Mohamed Riad El-Ghoneimy Date
External Examiner
Professor at Cairo University

Dr. Karim Seddik Date
Committee Observer
Associate Professor at the American University in Cairo

*To all my family members, especially my **Mother**, for their endless support and care: Your shining love guides my steps.*

*To my lovely **JOLLY** who gives me happiness: Thanks for being here.*

Shady Agwa

Jan. 2018

Acknowledgements

I would like to express my gratitude to my professor, Dr. Yehea Ismail for giving me the chance to work in a professional research environment and to gain more academic and professional experience.

I would like to thank my co-advisor, Dr. Eslam Yahya for his endless support and guidance.

Special thanks to Eng. Dalia Ahmed for her non-stopping support and her amazing efforts.

This research was partially funded by Zewail City of Science and Technology, AUC, the STDF, Intel, Mentor Graphics, ITIDA, SRC, ASRT and MCIT.

Table of Contents:

Chapter 1: Introduction. 1

1.1 Introduction. 1

1.2 Synchronous Sequential Circuits. 4

1.3 Research Objective. 8

1.4 Research Contribution. 8

1.5 Thesis Description. 9

Chapter 2: Literature Review. 12

2.1 PVT Variability Mitigation Approaches. 12

2.1.1 Time Relaxation Techniques. 12

2.1.2 Error Detection Techniques. 18

2.2 Power Saving Approaches. 23

2.2.1 Runtime Techniques. 23

2.2.2 Design-time Techniques. 27

Chapter 3: PVT Variability Mitigation. 31

3.1 Correction Function Technique. 31

3.1.1 Microarchitecture Overview. 31

3.1.2 Case Study, 4x4 bit Multiplier. 34

3.1.3 Experimental Results. 37

3.1.4 Conclusion. 38

3.1.5 Recommendations. 38

3.2 Error Recovery System using Taps, ERSUT. 40

3.2.1 ERSUT Microarchitecture Overview. 40

3.2.2 ERSUT Timing Constraints. 45

3.2.3 Meta-stability Analysis. 50

3.2.4 Case Study, 16x16 bit MAC Unit. 51

3.2.5 Experimental Results. 53

3.2.6 Conclusion. 55

3.2.7 Recommendations. 55

3.3 ERSUT-Based Resilient Microarchitecture. 57

3.3.1 Resilient Microarchitecture Overview. 58

3.3.2 Error Detection Circuit. 61

3.3.3 Meta-stability Analysis. 62

3.3.4 Timing Constraints. 63

3.3.5 Case Study, 16x16 bit MAC Unit. 67

3.3.6 Experimental Results. 70

3.3.7 Conclusion. 74

3.3.8 Recommendations. 74

Chapter 4: Power Efficient Design. 76

4.1 ERSUT-Based Microarchitecture and DVS. 76

4.1.1 Power vs. Throughput. 77

4.1.2 Low Power/ Same Throughput. 79

4.1.3 Case Study, 16x16 bit MAC Unit. 80

4.1.4 Experimental Results. 80

4.1.5 Conclusion. 82

4.2 Double-Edge-Triggered Microarchitecture. 83

4.2.1 Microarchitecture Overview. 84

4.2.2 Shared-Edge Detector, SED. 87

4.2.3 PVT Variability Mitigation. 90

4.2.4 Case Studies, 16x16 bit MAC Unit and ISCAS’89. 94

4.2.5 Experimental Results. 94

4.2.6 Conclusion. 99

Chapter 5: ERSUT-based CAD Automation. 101

5.1 ERSUT-based CAD Algorithm. 101

5.1.1 ERSUT-based Approach Implementation Flow. 101

5.1.2 Algorithm General Overview. 103

5.1.3 ERSUT-based Algorithm Details. 106

5.2 Validation Study of ERSUT-based Approach. 112

5.3 Conclusion. 117

Chapter 6: Conclusion and Future Directions. 118

6.1 Conclusion. 118

6.2 Future Directions. 121

References. 122

List of Figures:

Figure 1.1: Moore's law: Intel processors as an example. 1

Figure 1.2: The yield of the fabricated chips based on the maximum power budget and the time delay constraint. 3

Figure 1.3: A General architecture of a synchronous sequential circuit. 4

Figure 1.4: A General overview of the clock skewing for a synchronous sequential circuit. 5

Figure 1.5: The components of the clock period including the worst case of the critical path and the safety margins dedicated for clock skewing and jitters. The induced delays caused by PVT Variations should be taken into consideration to avoid timing violations. 7

Figure 2.1: Conventional (a) and proposed (b) Synchronizing Mechanism for n-Stage Pipeline Circuit. Cascaded delay chains are used to locally compensate the delay. 13

Figure 2.2: SACTA Self-Adjusting Clock Tree Architecture. Temperature adjustable buffers (white triangles) are used to mitigate the thermal induced time variability. 14

Figure 2.3: Time borrowing using pulsed-latches with different pulse widths. 15

Figure 2.4: The delayed master clock (CLKM) relative to the slave clock (CLKS) to create a window of transparency (t = transparent, o = opaque). 16

Figure 2.5: Skewing the clock signal to mitigate the PVT variations and the circuitry to change the delay of the clock signal. 17

Figure 2.6: A general overview of the Canary-based approach. 18

Figure 2.7: A general overview of TIMBER and its error relay logic. 19

Figure 2.8: A general overview of Razor-based approach. 20

Figure 2.9: A generic overview of the relation between the power, the delay, and the supply voltage of the logic gates. 24

Figure 2.10: An example of clock gating for register files. 25

Figure 2.11: Two power supply systems. (a) Normal power supply system, (b) Power gating system. 26

Figure 2.12: The transistor level design of the Double-Edge-Triggered Flip-Flop DETFF with two different implementations. 28

Figure 2.13: The implementation of the Double-Edge-Triggered Flip-Flop DETFF using three multiplexers. 29

Figure 2.13: The implementation of the Double-Edge-Triggered Flip-Flop DETFF using D-latch and pulse generator. 29

Figure 3.1: AND gate and its correction function: (a) Truth table, (b) Logic gates. 32

Figure 3.2: General architecture of the correction function approach for a two stages pipeline. 33

Figure 3.3: Two stages pipelined 4x4 bit multiplier. 35

Figure 3.4: A general overview of ERSUT microarchitecture. (a) Conventional pipeline architecture. (b) ERSUT architecture: Delayed copy of the data D is stored in PLS and compared to the data saved by FFM. In case of error, the error signal E is activated and propagates to make pulsed latches PL1, PL2, and PL3 transparent, equivalently stealing slack from STG1, STG2, STG3, and STG4. 41

Figure 3.5: The internal mechanism of ERSUT. (a) Truth table of the multiplexer selection signal S based on the error signal E. (b) Tap architecture: If (E = 1) → T = 1 (independent of Clk); as a result, the related pulsed latch becomes transparent. 42

Figure 3.6: The timing diagram of ERSUT main signals in case of error detection. 44

Figure 3.7: An example of pipeline with a critical stage illustrating the timing constraints. Symbol (*) refers to time violation. 46

Figure 3.8: A pipeline with a critical stage and Taps illustrating the short paths constraints where A, B and C are the values of the short paths. 48

Figure 3.9: The meta-stability resolution regions for ERSUT. 50

Figure 3.10: The general architecture of the 16x16 bit MAC unit. 51

Figure 3.11: The simplified architecture of the first critical stage, showing examples for a critical path (in red) and potential short paths (in blue). 52

Figure 3.12: A simplified architecture of the first two critical stages where the critical path in one stage borrows slack from a non-critical path of the other stage. 53

Figure 3.13: A general overview of the proposed self-healing resilient microarchitecture. The shadow flip-flop samples Dout at Clk_Edge and the main latch ML samples Dout from Clk_Edge to Clk_Edge + Tcp/2, where Tcp is the clock period. 59

Figure 3.14: The architecture of the Configurable-Buffer-Flip-Flop CBF, including its functionality table and the timing diagram. 60

Figure 3.15: Error Detection: (a) The conventional circuit or ERSUT. (b) The proposed circuit. 61

Figure 3.16: An example of a self-healing resilient microarchitecture with potential data racing through the short paths. Short paths (TSH1, TSH2, and TSH3) must be readjusted not to race with the long paths (Tm1, Tm2, and Tm3). 66

Figure 3.17: The light ERSUT-based resilient microarchitecture. The shadow flip-flop samples Dout at Clk_Edge and the main latch ML samples Dout from Clk_Edge to Clk_Edge + Tcp/2, where Tcp is the clock period. 68

Figure 3.18: The architecture of the Configurable-Latch-Flip-Flop CLF, including its functionality table and the timing diagram. 69

Figure 3.19: The micrograph of the chip including both the conventional and the resilient MAC units. 71

Figure 3.20: The operating frequency versus the supply voltage for the conventional MAC unit and the ERSUT-based resilient MAC unit. 73

Figure 4.1: The tradeoff between power saving and throughput for the Razor-based approaches and the proposed approach depending on the error recovery power. Reducing the error recovery power, due to recovery without pipeline flushing, shift the optimal point to save more power. 78

Figure 4.2: Low power/ same throughput flow using the voltage scaling and the ERSUT-based resilient microarchitecture. 79

Figure 4.3: The normalized power consumption versus the operating frequency for the conventional MAC unit and the ERSUT-based resilient MAC unit. 81

Figure 4.4: The Double-Edge-Triggered Pulsed-Latch DETPL using Edge-Detector circuit. The frequency of CLK signal is half the frequency of CLK_P signal. 84

Figure 4.5: The timing diagram of the internal signals of the Edge-Detector circuit including the clock signal CLK, the delayed clock signal CLK_D, and the pulsed-latch clock signal CLK_P which equals the double frequency of the main clock signal CLK.. 85

Figure 4.6: The double-edge-triggered microarchitecture using pulsed-latches and edge-detector circuits. The clock signal CLK propagates at frequency F/2 while the double-edge-triggered pulsed-latches operate at frequency F. 86

Figure 4.7: The 2-bit Double-Edge-Triggered Pulsed-Latch DETPL using Shared-Edge-Detector SED circuit 88

Figure 4.8: The double-edge-triggered microarchitecture using pulsed-latches with shared-edge-detector SED circuits. Each two adjacent pulsed-latches use only one edge-detector to reduce area overhead and power consumption. 89

Figure 4.9: The double-edge-triggered resilient microarchitecture examining PVT variations for all stages. The maximum accumulated induced delay for the whole pipeline that can be tolerated is up to 6 ns [2 ns for Stg1, 1 ns for Stg2, 2 ns for Stg3, and 1 ns for Stg4]. 91

Figure 4.10: The comparison of the power reduction between Double-Edge-Triggered Pulsed-latches with separate edge-detector (DETPL) and Double-Edge-Triggered Pulsed-latches with Shared-Edge-Detector (DETPL & SED). 98

Figure 4.11: The comparison of the area overhead between Double-Edge-Triggered Pulsed-latches with separate edge-detectors (DETPL) and Double-Edge-Triggered Pulsed-latches with Shared-Edge-Detectors (DETPL & SED). 98

Figure 4.12: The toleration windows of the ISCAS’89 benchmark circuits and the MAC unit in comparison to their clock periods. The toleration window represents the PVT variations that can be tolerated at runtime without generating errors. 99

Figure 5.1: The flowchart of applying the ERSUT-based resilient microarchitecture for a RTL design. 102

Figure 5.2: The flowchart of the ASIC cell-based flow. (a) The conventional flow, (b) The ERSUT-based flow. 104

Figure 5.3: A general overview of the inputs and outputs of the ERSUT-based C++ Algorithm. 105

Figure 5.4: The main functional blocks of the ERSUT-based C++ Algorithm. 106

Figure 5.5: The components of the struct used by the ERSUT-based Algorithm. 107

Figure 5.6: An example of RTL design to be mapped by the ERSUT-based algorithm. 108

Figure 5.7: The nodes of the bidirectional linked list representing the RTL design of Figure 5.6. 108

Figure 5.8: The areas of the different ISCAS’89 benchmark circuits including the MAC unit. 112

Figure 5.9: The numbers of flip-flops of the different ISCAS’89 benchmark circuits including the MAC unit. 113

Figure 5.10: The numbers of all paths of the different ISCAS’89 benchmark circuits including the MAC unit. 114

Figure 5.11: The percentage of critical paths in comparison to the number of all paths for the different ISCAS’89 benchmark circuits including the MAC unit. 114

Figure 5.12: The percentages of the toleration ratio for the different ISCAS’89 benchmark circuits including the MAC unit. 115

List of Tables:

Table 3.1. Results of Applying correction functions on 4x4 bit multiplier. 37

Table 3.2. Results of Area and Power for the Conventional MAC unit and the MAC unit with ERSUT. 54

Table 3.3. Results of the Conventional MAC unit and the ERUST-based resilient MAC unit including area and power, at the typical supply voltage 1.2 V, and the critical frequency 375 MHz. 70

Table 3.4. The operating frequencies of the conventional and ERSUT-based resilient MAC units at different supply voltages. 72

Table 4.1. The normalized power figures of the conventional and ERSUT-based resilient MAC units and their operating frequencies at different supply voltages. 82

Table 4.2. The results of area and power delay product of the MAC unit and ISCAS'89 benchmark circuits for the double-edge-triggered microarchitecture with separate edge-detector. 95

Table 4.3. The results of area and power delay product of the MAC unit and ISCAS'89 benchmark circuits for the double-edge-triggered microarchitecture with shared-edge-detector SED. 97

Table 5.1. The distribution of different circuits of ISCAS’89 including the MAC unit for the different toleration ratios. 116

List of Abbreviations:

PVT: Process, Voltage, and Temperature.

DVS: Dynamic Voltage Scaling.

DVFS: Dynamic Voltage and Frequency Scaling.

CAD: Computer-Aided Design.

ERSUT: Error Recovery System Using Taps.

DETFF: Double-Edge-Triggered Flip-Flop.

CBF: Configurable-Buffer-Flip-Flop.

CLF: Configurable-Latch-Flip-Flop.

DETPL: Double-Edge-Triggered Pulsed-Latch.

SED: Shared-Edge-Detector.

List of Publications:

[1] S. Agwa, E. Yahya and Y. Ismail, "Design techniques for variability mitigation", Int. J. Circuits and Architecture Design, Vol. 1, No. 1, 2013.

[2] S. Agwa, E. Yahya and Y. Ismail, "Variability Mitigation Using Correction Function Technique", 2013 IEEE International Conference on Electronics, Circuits, and Systems, ICECS, 2013.

[3] S. Agwa, E. Yahya and Y. Ismail, "ERSUT: A Self-Healing Architecture for Mitigating PVT Variations without Pipeline Flushing", IEEE Transactions on Circuits and Systems II, TCASII Express Briefs, Volume: 63, Issue: 11, 2016, pp. 1069 - 1073.

[4] S. Agwa, E. Yahya and Y. Ismail, "Power efficient AES core for IoT constrained devices implemented in 130nm CMOS," 2017 IEEE International Symposium on Circuits and Systems, ISCAS 2017, Baltimore, MD, 2017, pp. 1-4.

[5] S. Agwa, E. Yahya and Y. Ismail, "A Low Power Self-healing Resilient Microarchitecture for PVT Variability Mitigation ", Accepted by IEEE Transaction on Circuits And Systems I, TCAS I Regular Paper, 2017.

[6] S. Agwa, E. Yahya and Y. Ismail, "Low Power Double-Edge-Triggered Microarchitecture Using Pulsed-Latches", 2018 [In Preparation].

[7] S. Agwa, E. Yahya and Y. Ismail, "Validation Study on Time Relaxation Benefits using ERSUT-based CAD Algorithm", 2018 [In Preparation].

Abstract

Nowadays, the high power density and the process, voltage, and temperature variations became the most critical issues that limit the performance of the digital integrated circuits because of the continuous scaling of the fabrication technology. Dynamic voltage and frequency scaling technique is used to reduce the power consumption while different time relaxation techniques and error recovery microarchitectures are used to tolerate the process, voltage, and temperature variations. These techniques reduce the throughput by scaling down the frequency or flushing and restarting the errant pipeline. This thesis presents a novel resilient microarchitecture which is called ERSUT-based resilient microarchitecture to tolerate the induced delays generated by the voltage scaling or the process, voltage, and temperature variations. The resilient microarchitecture detects and recovers the induced errors without flushing the pipeline and without scaling down the operating frequency. An ERSUT-based resilient 16 × 16 bit MAC unit, implemented using Global Foundries 65 nm technology and ARM standard cells library, is introduced as a case study with 18.26% area overhead and up to 1.5x speedup. At the typical conditions, the maximum frequency of the conventional MAC unit is about 375 MHz while the resilient MAC unit operates correctly at a frequency up to 565 MHz. In case of variations, the resilient MAC unit tolerates induced delays up to 50% of the clock period while keeping its throughput equal to the conventional MAC unit's maximum throughput. At 375 MHz, the resilient MAC unit is able to scale down the supply voltage from 1.2 V to 1.0 V saving about 29% of the power consumed by the conventional MAC unit.

A double-edge-triggered microarchitecture is also introduced to reduce the power consumption extremely by reducing the frequency of the clock tree to the half while preserving the same maximum throughput. This microarchitecture is applied to different ISCAS'89 benchmark circuits in addition to the 16x16 bit MAC unit and the average power reduction of all these circuits is 63.58% while the average area overhead is 31.02%. All these circuits are designed using Global Foundries 65nm technology and ARM standard cells library.

Towards the full automation of the ERSUT-based resilient microarchitecture, an ERSUT-based algorithm is introduced in C++ to accelerate the design process of the ERSUT-based microarchitecture. The developed algorithm reduces the design-time efforts dramatically and allows the ERSUT-based microarchitecture to be adopted by larger industrial designs. Depending on the ERSUT-based algorithm, a validation study about applying the ERSUT-based microarchitecture on the MAC unit and different ISCAS'89 benchmark circuits with different complexity weights is introduced. This study shows that 72% of these circuits tolerates more than 14% of their clock periods and 54.5% of these circuits tolerates more than 20% while 27% of these circuits tolerates more than 30%. Consequently, the validation study proves that the ERSUT-based resilient microarchitecture is a valid applicable solution for different circuits with different complexity weights.

Index Terms— Error Detection, Error Recovery, Time Relaxation, Low Power, Power Management, Pipeline Recovery, Process Voltage Temperature (PVT) Variations, Resilient Microarchitecture, Self-healing Architecture, Double-Edge-Triggered, Throughput, Dynamic Voltage Scaling, CAD, Static Timing Analysis, ISCAS'89, MAC unit.

Chapter 1: Introduction

1.1 Introduction

Nowadays, the use of digital integrated circuits is extended to be a part of our daily life. Processors, GPUs, Internet of Things IoTs, and other mobile-based applications are still hungry for higher performance with lower cost. This essential demand shows the urgent need not only for high speed digital integrated circuits but also low power circuits with better performance per watt. The performance demands are extended to be concerned by higher speed, smaller size, longer battery life, better performance per watt and lower cooling costs. As mentioned by Moore's law, number of transistors are doubled every two years and this is verified by the continuous decreasing of transistors' size.

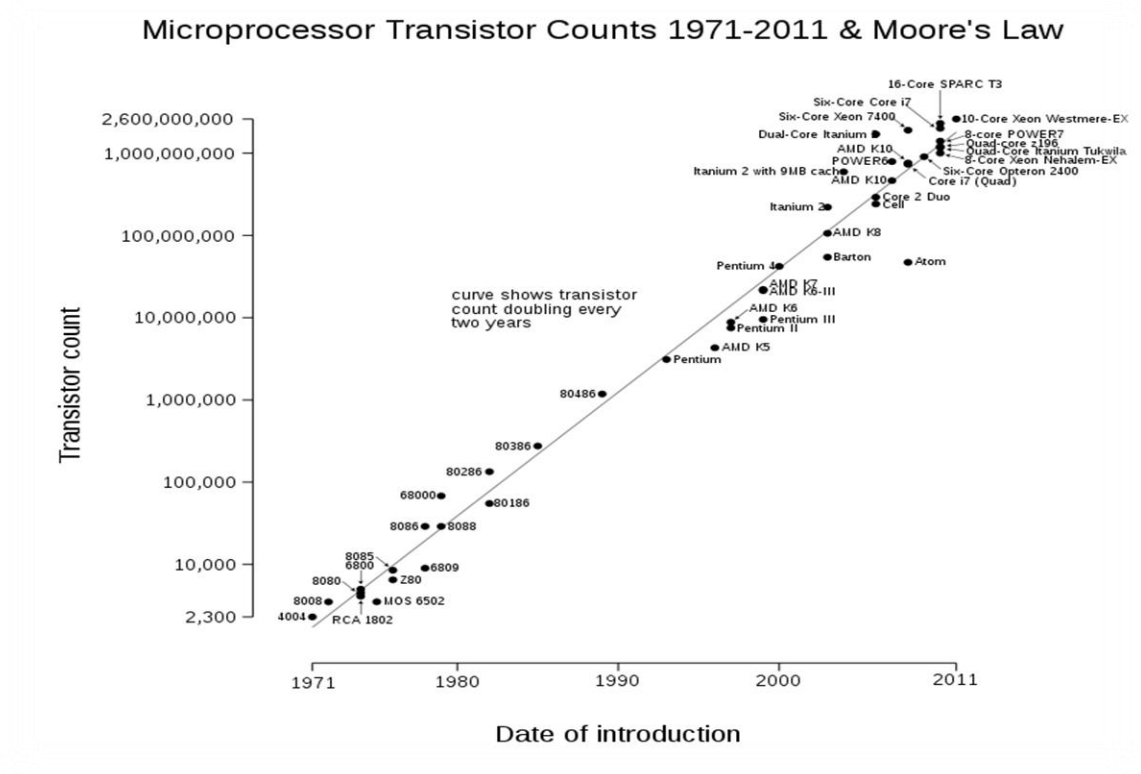


Figure 1.1: Moore's law: CPU transistor counts against dates of introduction © [1].

Figure 1.1 shows the increasing number of transistors used by the different CPUs during the last decades as an example. As the fabrication technology migrates towards nanometer scale, billions of transistors are integrated together into one chip to do more tasks with higher speed and lower costs. On the other hand, many serious issues have arisen just like process, voltage and temperature (PVT) variations, high power density, and yield loss. Process, voltage and temperature (PVT) variations have become a serious problem against the synchronous digital circuit design due to the deep nanometer scale. Different doping concentrations, different gate-lengths, unexpected voltage drops in the power supply grid and temperature fluctuations cause temporary and permanent defects. These defects are the major cause of the unpredicted induced delays which lead to a miss-synchronization. Because of these delay variations, the registers receive corrupted data especially at the end of the critical stages. So, the timing constraints of the digital chips must have enough safety margins to avoid any timing violations and they have to operate at lower than the maximum frequencies to avoid the impacts of any potential variation.

Operating billions of transistors with very high frequencies consumes a catastrophic amount of power and increases the probability of malfunctioning. The power budget is an important matter regarding the high performance applications especially for mobile devices and IoT constrained devices. Increasing the power density of the chip does not only increase the probability of malfunctioning due to the thermal induced delays, but also accelerates the degradation and aging issues. PVT variations and high power density threaten Moore's law and put an end to the trend of increasing the operating frequency. These two major issues have a considerable negative impact on the reliability of the fabricated chips any many chips can be excluded due to the lack of reliability.

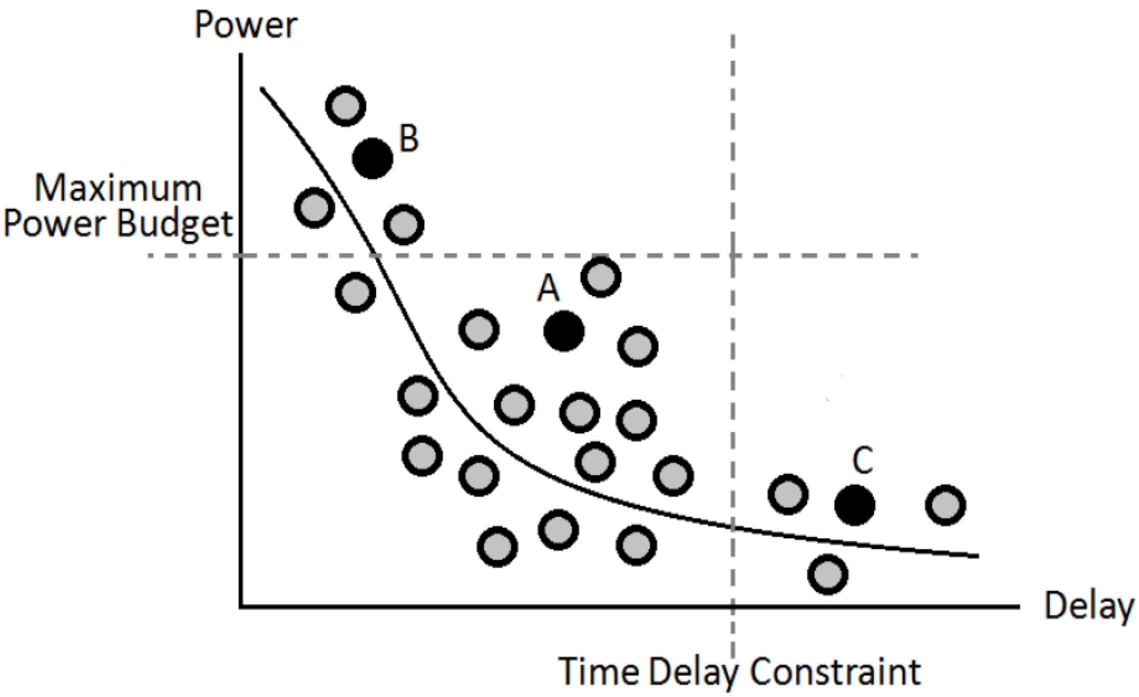


Figure 1.2: The yield of the fabricated chips based on the maximum power budget and the time delay constraint © [3].

The yield of the manufactured chips is simply defined as the amount of working chips that are satisfying the predesigned performance conditions and specifications compared to the total number of developed chips [2]. It depends on two major constraints: the maximum power budget and the time delay constraint. Figure 1.2 shows the classification of the fabricated chips based on these two major constraints [3]. Only chips of zone A are considered as working chips because they meet the maximum power budget and the time delay constraint of the designed chip. Chips of zone C are excluded due to violating the time delay constraint while chips of zone B are also excluded due to exceeding the maximum power budget of the chip. PVT variations drag the chips from zone A to zone C due to the induced delay variations while the high power consumption drags the chips from zone A to zone B by violating the power budget. Consequently, the yield of the fabricated chips is decreased by both PVT variations and the high power consumption. This shows the urgent demands of novel approaches that reduce the power consumption and tolerate the PVT induced delays while the main target is to achieve the maximum throughput by using the minimum amount of power.

1.2 Synchronous Sequential Circuits

The sequential digital integrated circuits can be classified into two major categories: Synchronous circuits and Asynchronous circuits. The synchronous sequential circuits are synchronized by a periodic clock signal based on which the data is transferred through the pipeline. On the other hand, the asynchronous circuits are clock-less circuits that are based on a handshaking protocol signals to transfer data from one stage to another.

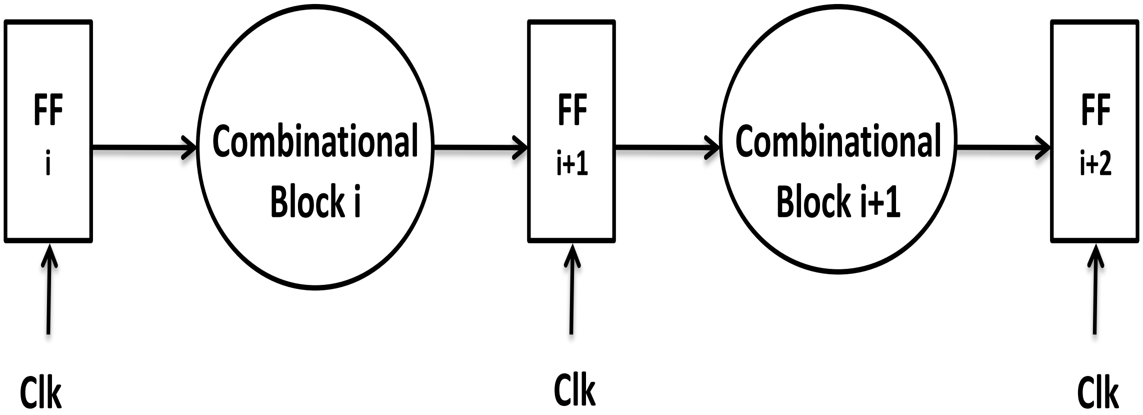


Figure 1.3: A General architecture of a synchronous sequential circuit © [4].

Figure 1.3 shows a general architecture of the synchronous sequential circuits. These circuits consist of combinational logic blocks that are surrounded by memory elements like flip-flops. The flip-flops isolate the combinational logic blocks and transfer data between the adjacent stages according to a common periodic clock signal. At the rising or falling edge of the clock signal, every flip-flop captures the output data of its previous stage and transfers the data to the next stage. The clock period of the sequential circuit is chosen according to the worst path of the most critical stage. Every combinational logic stage has to deliver its output data within the predetermined clock period. Setup time, hold time and clock skewing should be taken into consideration to ensure the correctness of transferring data between stages. The timing of synchronous sequential circuits is verified by two constraints to prevent receiving incorrect early data or missing correct late data. The first constraint is the short path constraint, illustrated by (1.1).

$$T_{i+1} + T_H < T_i + \mathrm{d(i, i+1)} \qquad (1.1)$$

While T_i is the arrival time of the clock at flip-flop i. T_{i+1} is the arrival time of the clock at the next flip-flop i+1. T_H is the hold time which is the amount of time needed after the clock edge for the input data to be held stable to make sure that the master latch of the flip-flop has been fully disabled. d(i, i+1) is the minimum propagation delay of the combinational logic stage i. Satisfying this constraint guarantees enough time for flip-flop i+1 to save the data before the new data, from flip-flop i, propagates through the short path of the combinational logic stage and damages the old correct data. The second constraint is the long path constraint, illustrated by (1.2).

$$T_{i+1} + T_{CP} > T_i + \mathrm{D(i, i+1)} + T_S \qquad (1.2)$$

While T_{CP} is the clock period. D(i, i+1) is the maximum propagation delay of the combinational logic stage i. T_S is the setup time which is the amount of time needed before the clock edge for the input data to be stable to make sure that the master latch of the flip-flop has enough time to save the correct data. Satisfying this constraint guarantees enough time for flip-flop i+1 not to save an early incorrect version of the propagating data.

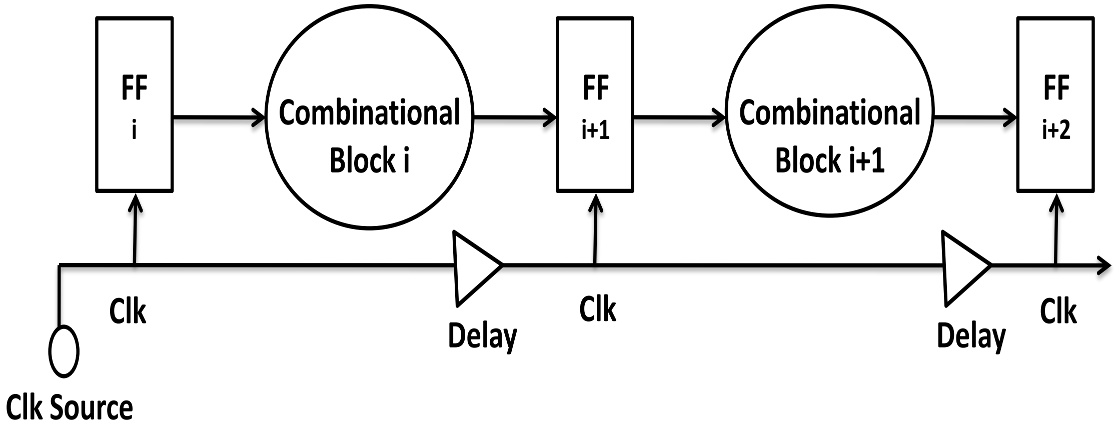


Figure 1.4: A General overview of the clock skewing for a synchronous sequential circuit © [5].

Clock signals reach the flip-flops through the clock distribution networks which are called clock trees. These signals propagate through interconnection wires and buffers with different delays. The flip-flops within the same pipeline may receive the same clock signal at different times with variant delays. So, clock skewing means that the clock signal reaches different flip-flops at different times through the clock tree.

Figure 1.4 shows a general overview of the clock skewing that affects the synchronous circuits. For example, the clock signal Clk reaches the flip-flop FF_{i+1} earlier than FF_{i+2} and later than FF_i due to the delays of the clock tree interconnects and buffers. Many algorithms are developed to synthesize a zero skew clock trees to avoid any timing violations. However, any unpredicted positive or negative skewing can corrupt the data. The positive skew occurs when the clock signal Clk reaches FF_{i+1} after it reaches FF_i so the combinational block i has more slack to deliver its output but on the other hand this positive skew can violate the short path constraint as the new data of FF_i may have enough time to propagate through the short path and corrupt the correct previous data of FF_{i+1}. The negative skew occurs when the clock signal Clk reaches FF_{i+1} before it reaches FF_i so the combinational block i does not have enough time to deliver its output data and the long path constraint may be violated. So safety margins should be added to the clock period to tolerate the clock skewing and to ensure the robustness of the pipeline stages by keeping the long path and short path constraints verified.

Figure 1.5 shows the major components of the clock period that have to satisfy the long path constraint. The clock period is greater than the average propagation delay of all stages and it is adjusted based on the worst case of the most critical path belongs to the critical stage. The worst case is also data dependent as the input data pattern should propagate through the longest path to meet this case. In addition to the worst case margin, another safety margin is added to the clock period for the clock skewing and clock jitters.

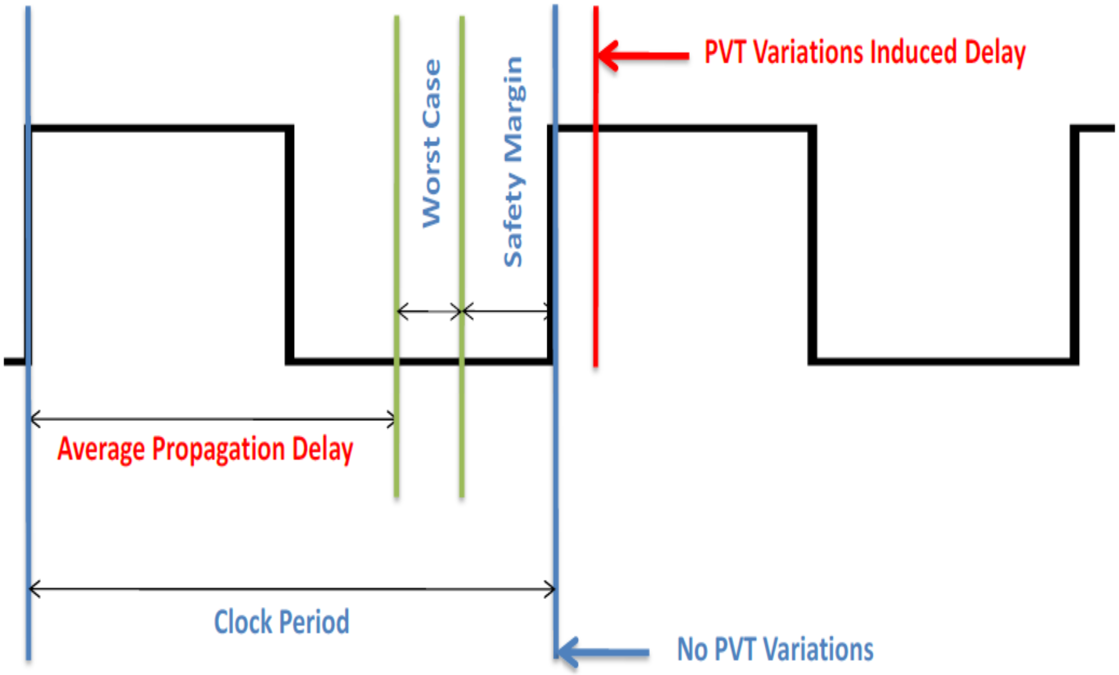


Figure 1.5: The components of the clock period including the worst case of the critical path and the safety margins dedicated for clock skewing and jitters. The induced delays caused by PVT Variations should be taken into consideration to avoid timing violations.

Process, voltage and temperature (PVT) variations lead to unpredicted run-time induced delays. If there is no enough safety margin, the critical stages violate the long path constraint and fail to deliver their output data before the active edge of the clock signal. Due to PVT induced delays, critical stages deliver incorrect data to their successive stages and the errant data propagates through the pipeline. Then, the corrupted pipeline should be flushed and restarted from the previous correct state. Flushing the pipeline does not only reduce the overall throughput of the digital circuits but also increases the energy consumption of a specific operation. Increasing the safety margins, to cover the PVT variations, increases the robustness and the reliability of the digital circuits but it also decreases the throughput by decreasing the operating frequency.

1.3 Research Objective

The objective of this research is to increase the yield and the reliability of the digital integrated circuits by tolerating the PVT delay variations and reducing the power consumption without decreasing the maximum throughput to keep the high performance demand satisfied. The major contribution of this work is to investigate the potential solutions for these two major issues and to develop a power efficient resilient microarchitecture with an error recovery mechanism. The proposed microarchitecture eliminates the safety margins of the timing constraints to operate at the maximum critical frequency, while the error recovery mechanism tolerates the induced errors generated by the PVT variations without flushing the faulty pipeline stages. In case of no PVT variations, this resilient microarchitecture makes use of the hardware overhead to reduce the power consumption through Dynamic Voltage Scaling (DVS) technique without reducing the throughput. If the power budget is not the targeted priority, the proposed resilient microarchitecture increases the operating frequency beyond the maximum critical frequency at the cost of the power.

1.4 Research Contribution

The major contributions of this research are summarized as follows:

- Developing a generic resilient microarchitecture that tolerates the PVT induced delays at runtime without flushing the pipeline or reducing the operating frequency.
- Using the developed resilient microarchitecture to reduce the power consumption by scaling down the supply voltage without scaling the operating frequency. So the digital circuits are able to decrease the power consumption without affecting the maximum throughput.
- Developing a low power double-edge-triggered microarchitecture with an acceptable area overhead, which reduces the power of the clock tree components dramatically without reducing the maximum throughput.

- Developing a CAD algorithm in C++ that automates and accelerates the design process of the developed resilient microarchitecture.
- Introducing a validation study about the potential benefits of using the developed resilient microarchitecture to increase the performance for different ISCAS'89 benchmark circuits.

1.5 Thesis Description

This thesis consists of six chapters and the following paragraphs introduce a general overview of each chapter's content.

Chapter 1: Introduction.

A general overview is introduced about the digital integrated circuits and the targeted problem is discussed including PVT variations and high power consumption. The synchronous sequential circuits and their timing constraints are discussed to clarify the timing issues threatening the reliability of these circuits. The objective and the contribution of this research are also listed.

Chapter 2: Literature Review.

The literature review chapter is divided into two sections. The first section is dedicated for the PVT variability mitigation techniques including time relaxation techniques and error detection techniques. The second section discusses the power saving approaches used to design low power digital integrated circuits.

Chapter 3: PVT Variability Mitigation.

The different developed techniques, tolerating the PVT variations, are discussed. The Correction Function technique is explained and its experimental results are shown. The Error Recovery System Using Taps ERSUT and its experimental results are discussed in detail. The ERSUT-based resilient microarchitecture, extracted from ERSUT, is introduced as a novel solution for the PVT variations while its experimental results are shown.

Chapter 4: Power Efficient Design.

Integrating dynamic voltage scaling DVS with the developed ERSUT-based resilient microarchitecture is introduced, in addition to its experimental results, as a novel power management solution to reduce the power consumption without reducing the throughput. The Double-Edge-Triggered microarchitecture is also introduced with its experimental results as a promising solution to reduce the power consumption of the clock tree with acceptable area overhead.

Chapter 5: ERSUT-based CAD Automation.

Automating the process of applying the ERSUT-based approach is discussed and an algorithm in C++ is introduced to be integrated with the CAD Static Timing Analysis tool. The ERSUT-based algorithm is applied on different circuits of the ISCAS’89 benchmark circuits. A validation study about the potential benefits of using the ERSUT-based resilient microarchitecture is discussed.

Chapter 6: Conclusion and Future Directions.

In this chapter, the final conclusion is introduced for all results obtained by the different developed approaches for both PVT variability mitigation and power saving. The future directions of this research are also discussed to investigate the potential improvements can be applied.

Chapter 2: Literature Review

2.1 PVT Variability Mitigation Approaches

Process, Voltage and Temperature (PVT) variations have negative impacts on the performance of the manufactured chips. Their effect has been increased since the fabrication technology went into the deep nanometer scale. Process variations may be existing due to variations in channel length or doping concentration. Environmental variations are related to ambient conditions including voltage and temperature. Voltage variations are caused by unexpected voltage drops in the power supply network or variations in the supply voltage itself. Temperature variations arise due to temperature fluctuations and other environmental impacts. Many approaches were developed and introduced as potential solutions to mitigate the PVT variations [3]. These approaches are targeting different level of solutions from device level to microarchitecture level. For device level approaches, the body biasing is widely used to mitigate the PVT variations [6] [7]. This research is interested only in the microarchitecture approaches which target any device for any technology. The microarchitecture approaches are classified into two main categories: Time Relaxation Techniques, and Error Detection Techniques.

2.1.1 Time Relaxation Techniques

The Time Relaxation Techniques aim to mitigate the negative impacts of the PVT variations by relaxing the timing conditions among the different pipeline stages to prevent the potential timing violations and to avoid any kind of error generation.

2.1.1.1 Correlated Clock Skewing [8]

Designing clock distribution network with zero clock skew is beneficial in case of ignoring PVT variations but nowadays it is useful to use clock skew intentionally not only to improve performance but also to compensate the effects of PVT variations. At design time, safety margins are considered to ensure the robustness of the chip in the presence of the PVT dependent skew variations. A compensation mechanism for environmental parameters fluctuation (voltage and temperature) was introduced as shown in Figure 2.1.

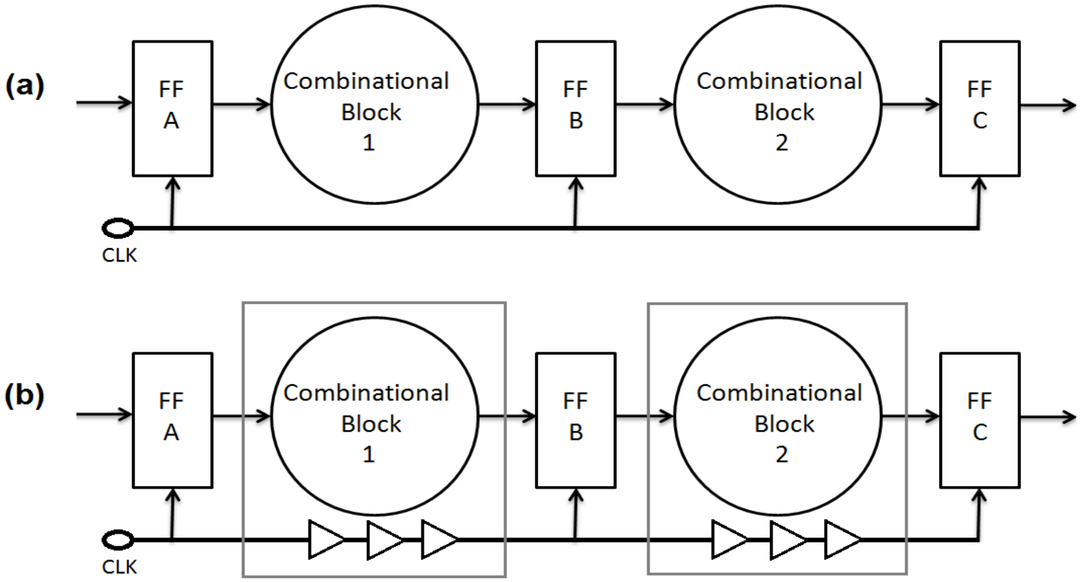


Figure 2.1: Conventional (a) and proposed (b) synchronizing mechanism for n-stage pipeline circuit. Cascaded delay chains are used to locally compensate the delay © [8].

The proposed compensation mechanism depends on the correlation between the combinational logic stages and the skewing buffers. As shown in Figure 2.1.b, the clock signals of the different registers propagate through the delay chains of buffers. Both combinational logic stages and skewing buffers are assumed to experience the same delay variations because of spatial correlation which is assumed to offer the same voltage and temperature fluctuations.

2.1.1.2 Self-Adjusting Clock Tree Architecture, SACTA [9]

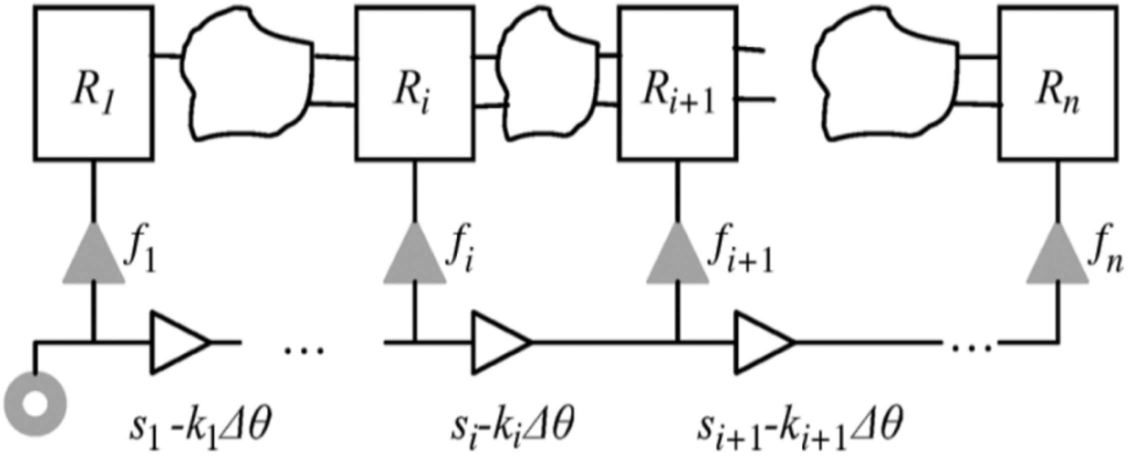


Figure 2.2: SACTA Self-Adjusting Clock Tree Architecture. Temperature adjustable buffers (white triangles) are used to mitigate the thermal induced time variability © [9].

This approach is more concerned about thermal-induced delays generated by temperature fluctuations. Its objective is to keep the performance in the presence of variations with minimum hardware overhead. The problem was defined as to design a clock tree that changes skew values dynamically and these values are linear functions of the temperature. Figure 2.2 shows a pipeline with Self-Adjusting Clock Tree Architecture. The white triangles are automatic temperature adjustable skew buffers while the gray triangles are fixed skew buffers which are designed to be temperature insensitive. These fixed buffers have a base delay F_i. The relationship between the delay and temperature is expressed as s_i- $K_i \Delta\theta$, where s_i is the delay of the skew buffer at the worst-case temperature; K_i is the temperature sensitivity coefficient; Δθ is the difference between the worst-case temperature and the operating temperature. SACTA has a powerful advantage over the static clock skew scheduling techniques as static techniques only satisfy some temperature profiles selected during design time while SACTA changes skew values dynamically.

2.1.1.3 Pulsed-Latches [10]

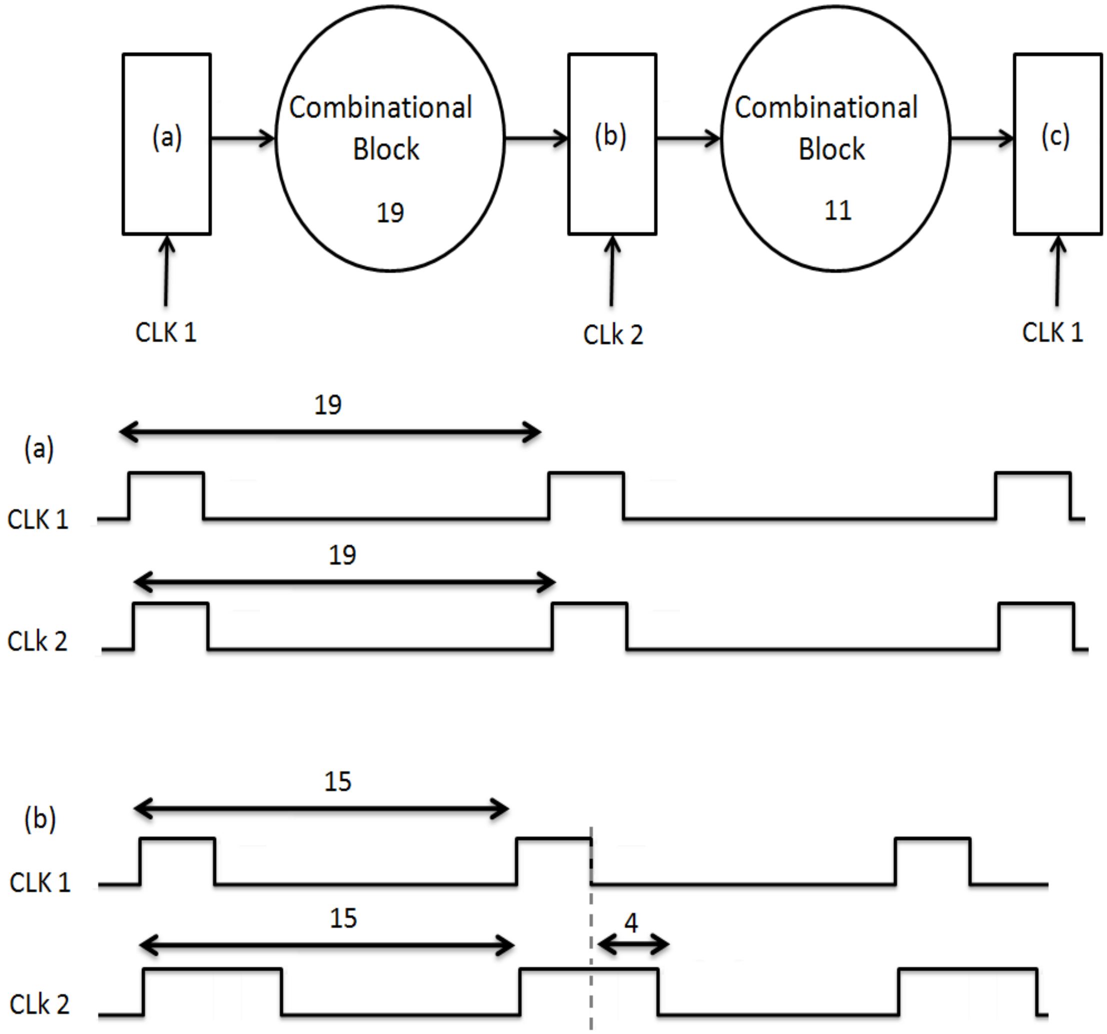


Figure 2.3: Time borrowing using pulsed-latches with different pulse widths © [10].

Pulsed-latches were introduced as an ideal sequencing element for high performance integrated circuits due to its reduced sequencing overhead. Pulsed-latch circuits with regulating pulse-width give enough flexibility to manage timing issues. The transparency windows of the pulsed-latches are used to borrow time from adjacent stages by generating different clock pulse widths as shown in Figure 2.3. The pulsed-latches a, b, and c, driven by clock signals CLK1 and CLK2, are shown in Figure 2.3. The maximum delay of the combinational logic stage between a, and b is assumed to be 19 time units and that between b and c to be 11. The clock period has to be 19 time units to preserve timing constraints. If b is driven by a new clock with wider pulse by 4 time units, then the clock period can be reduced to 15 time units because the combinational logic stage between a and b can borrow 4 time units from the adjacent combinational logic stage between b and c. The same concept can be used to tolerate the PVT induced delay variations by relaxing the timing among the adjacent stages. However, using different clock signals with different pulse widths is a serious problem that should be avoided.

2.1.1.4 Soft-Edge Flip-Flops [11]

The soft-edge flip-flop is designed with two different clocks, one for the master latch and the other for the slave latch. The clock of the master latch is delayed with respect to the slave clock to create a transparency window as illustrated in Figure 2.4.

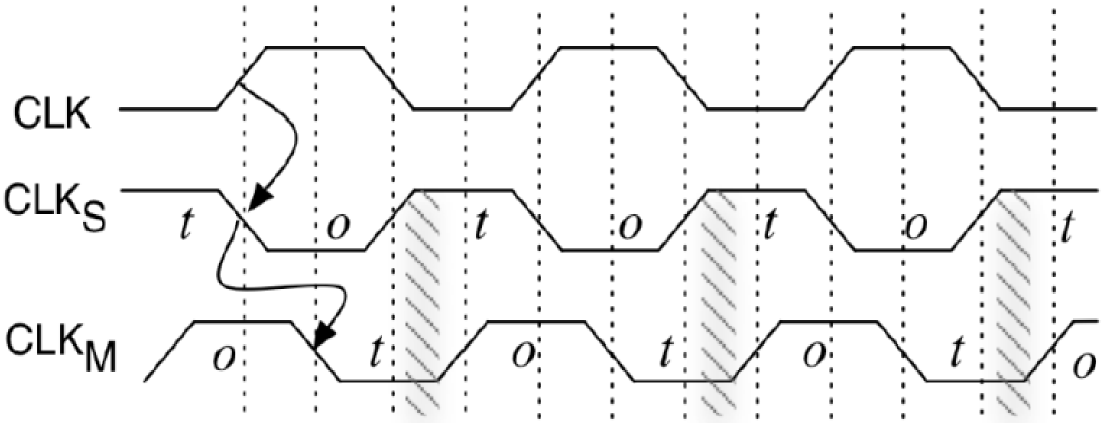


Figure 2.4: The delayed master clock (CLKM) relative to the slave clock (CLKS) to create a window of transparency (t = transparent, o = opaque) © [11].

This transparency window allows delayed data to be captured and transferred to the slave latch. So, the soft-edge flip-flop keeps the data synchronization at the clock edge, in addition to a transparency window around this edge. This transparency window reduces the sensitivity to clock skew and jitter and allows time borrowing among pipeline stages to mitigate the PVT variability induced delays.

2.1.1.5 ReCycle [12]

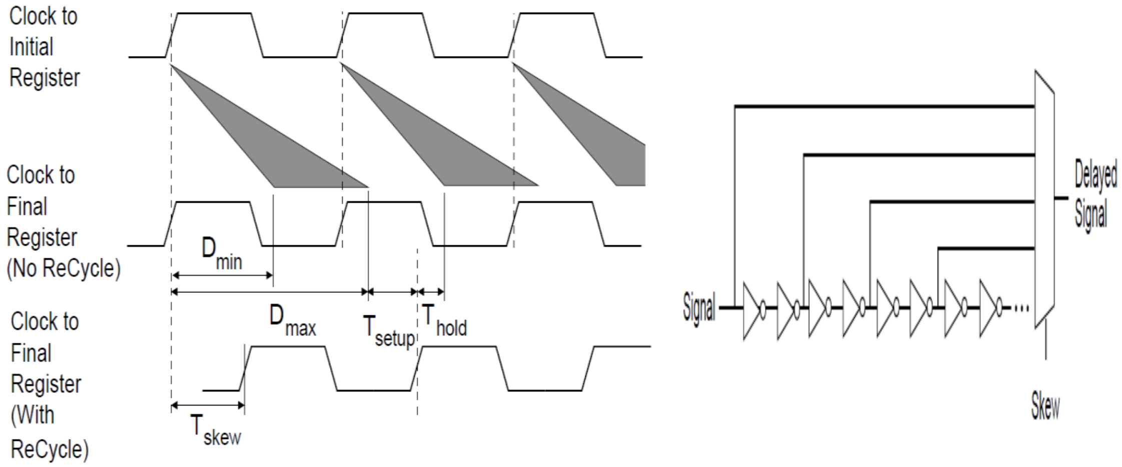

Clock to Initial Register
Clock to Final Register (No ReCycle)
Clock to Final Register (With ReCycle)
D_{min}
D_{max}
T_{setup}
T_{hold}
T_{skew}
Signal
Delayed Signal
Skew

Figure 2.5: Skewing the clock signal to mitigate the PVT variations and the circuitry to change the delay of the clock signal © [12].

The ReCycle approach was developed to relax the timing of the adjacent stages by using clock skewing. Figure 2.5 shows the programmable clock skewing used to re-adjust the clock skew according to the requirements of the critical stages. Tunable Delay Buffers (TDB) are used to allow critical stages borrowing time slack from the non-critical stages. Donor stages, which are empty stages, were also added to the critical loop of the pipeline to relax the timing and to increase the throughput of the digital integrated circuit. However, the ReCycle approach needs a complicated testing setup with BIST vectors at startup to tune the clock skewing in addition to the overhead of the donor stages.

Although most of the time relaxation techniques have a relatively low hardware overhead, they depend on the design-time safety margins. These safety margins make the design more conservative by reducing the operating frequency which means that a real chance to get better performance is lost. The targeted designs have to operate at lower than the maximum frequency to avoid generating errors. If the safety margins are reduced and the PVT induced delay variations exceed these margins, errors are generated and corrupted data propagates through the pipeline stages. In case of error generation due to these unexpected variations, the digital integrated circuits are malfunctioning without any chance of error detection and recovery. This shows the importance of the error detection techniques that are able to detect and recover the generated errors.

2.1.2 Error Detection Techniques

The Error Detection Techniques aim to predict or detect the generated errors due to the PVT induced delay variations and then to feedback the system to take actions to recover these errors. Consequently, the error detection techniques are able to reduce the safety margins of both frequency and voltage till the error rate exceeds a certain threshold by which the recovery cost is greater than the achieved benefits.

2.1.2.1 Canary-based Approaches [13]

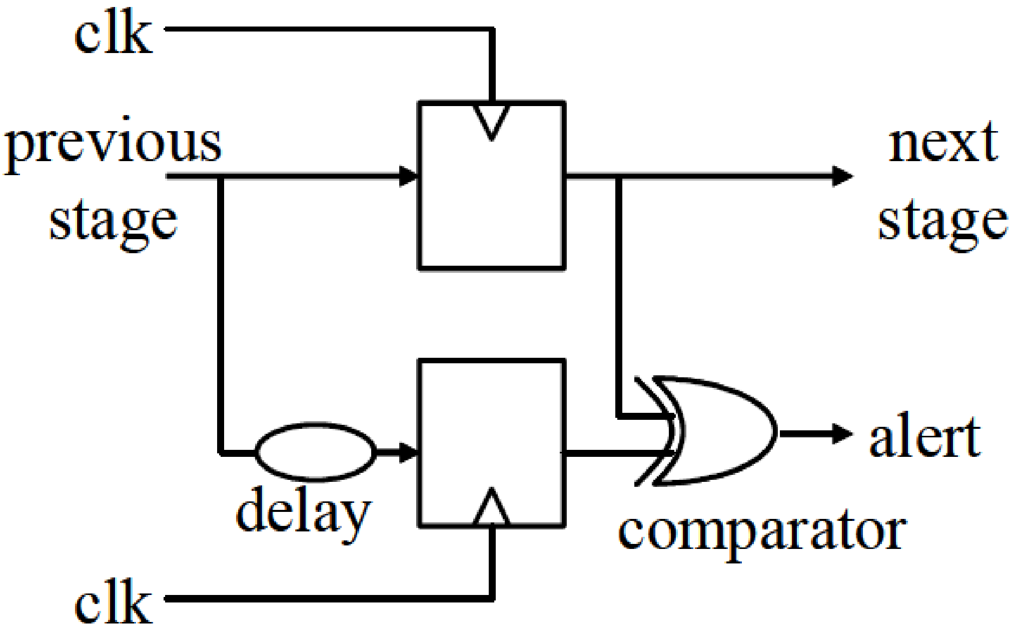


Figure 2.6: A general overview of the Canary-based approach © [13].

Canary-based approaches give an early indication about the potential errors due to the PVT variations. Figure 2.6 shows the basic architecture of the Canary-based approaches in which two flip-flops are used to save the output data; the main flip-flop saves the data in-time while the other flip-flop saves the data after a certain delay. This delay is the minimum safety margin which is predetermined at design time. The two values are compared and if they are identical, the design is still safe and there are no potential errors. If the two values are not identical, the alert is activated to show the possibility of a potential error. The alert signal is used to feedback the system to re-adjust the operating frequency or the supply voltage to increase the safety margin. Although Canary-based approaches predict the potential occurrence of errors to be avoided, safety margins are still required which means that lower operating frequency should be applied.

2.1.2.2 Time Borrowing and Error Relaying, TIMBER [14]

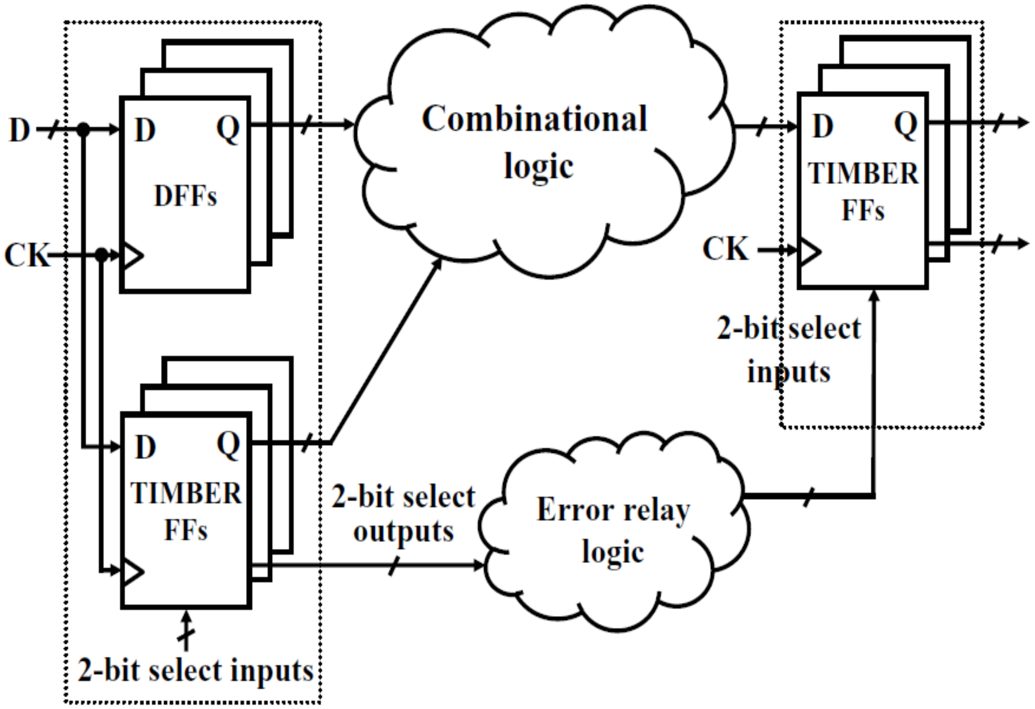


Figure 2.7: A general overview of TIMBER and its error relay logic © [14].

Time Borrowing and Error Relaying (TIMBER) approach allows the critical stages to borrow slack from their next stages and recover the error by using discrete time interval margins and scaling down the operating frequency. Figure 2.7 shows a general overview about the main components of the TIMBER microarchitecture. TIMBER flip-flop has two master latches with a normal clock and a delayed one to detect errors. It also has a programmable clock skewing circuitry to be adjusted at runtime according to the output of the error relay logic. The error relay-logic units are responsible for choosing the mode of operation. The modes of operation include: tolerating error by time borrowing without raising an error flag, and error detection with raising an error flag. If there are two-stage timing errors they are masked by borrowing a time-borrowing (TB) interval and an error-detection ED interval, then the timing error is flagged to a central error control unit and the operating frequency is reduced without flushing the pipeline. Although this approach avoids flushing the pipeline, the overhead of the error relay logic and TIMBER flip-flops is relatively high in comparison to the small time intervals offered for time borrowing. Finally, TIMBER has to scale down the operating frequency and reduce the throughput to prevent the error from propagating through multi-stages.

2.1.2.3 Razor-Based Approaches [15-21]

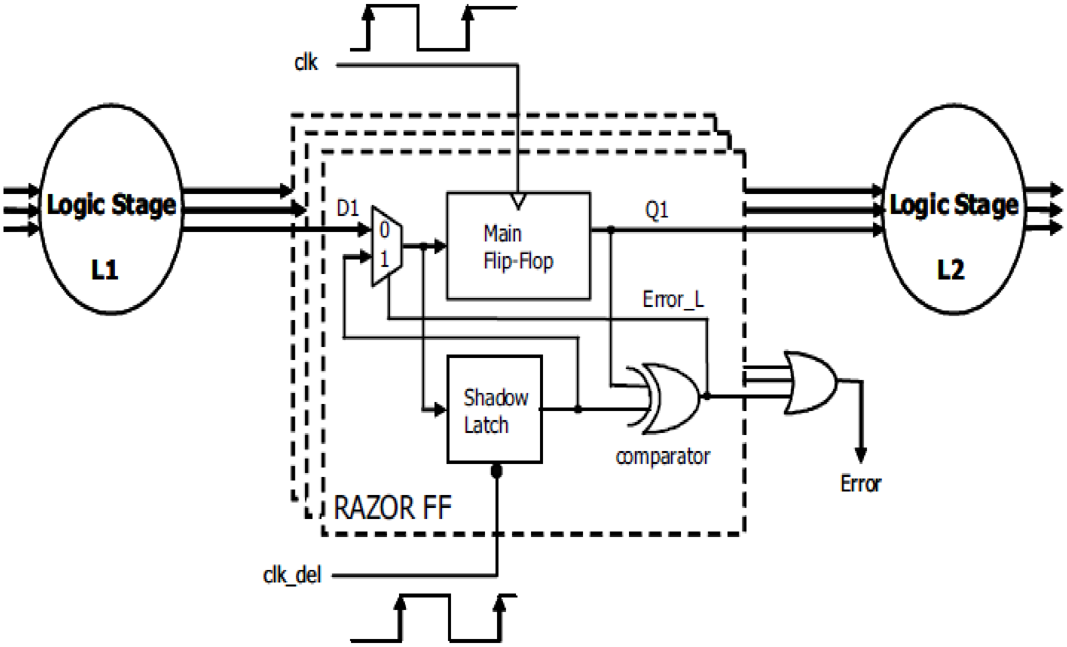


Figure 2.8: A general overview of Razor-based approach © [15].

Razor is one of the most common techniques used to tolerate the PVT variations and to reduce the safety margins for both supply voltage and operating frequency. The basic idea behind Razor is to remove the safety margins while monitoring the error rate. If the error rate exceeds a certain threshold the operating frequency or the supply voltage must be re-adjusted at runtime. As Figure 2.8 shows, the main flip-flop captures the data according to the normal clock while a delayed clock is used to save a delayed version of the same data by a shadow latch. The two versions are compared and if they are not identical the error signal is activated. In contrast to conservative designs, which need conservative safety margins, Razor eliminates the safety margins and it uses the error signals dynamically to readjust the operating conditions with low hardware overhead. However, Razor and Razor II have to flush and restart the pipeline stages in case of error detection. Flushing the pipeline decreases the throughput and increases the energy consumption of the error recovery. Flushing and restarting the pipeline is a costly solution for the time and energy budget especially for modern processors using multi-instructions issuing and superscalar architectures with deep pipelines. On the other hand, Bubble Razor stalls the pipeline for a whole clock cycle, in case of error detection, by inserting a bubble to compensate the error. Inserting bubbles also decreases the throughput and may export a problem to the outer systems that expect a real data while receiving a bubble.

Better-than-worst-case designs, based on Razor approach, are suffering from the high cost of error recovery process including losses in throughput and energy. Razor-based approaches also suffers from meta-stability issues that reduce the total throughput and the power saving. The meta-stability of a signal, due to a setup violation, is an unstable state in which the signal is not resolved at a certain logic level (neither 0 nor 1). As the meta-stability resolution is time independent, all better-than-worst-case designs (suffering from meta-stability) are not reliable enough to be adopted by the industry which is the main limitation of these approaches to be widely used [22].

Although the error detection techniques predict or detect the errors, they have to keep the safety margins and/or scale down the operating frequency to tolerate the induced delay variations. They do not make use of the potential benefits could be achieved by the time relaxation concept. Integrating the concept of "Error Detection" with the concept of

"Time Relaxation" can offer a great chance to tolerate the PVT induced delay variations without flushing the pipeline nor reducing the operating frequency. The objective of this research is to develop an error detection technique that uses the time relaxation concept to tolerate the PVT induced delay variations without flushing the pipeline nor reducing the operating frequency.

2.2 Power Saving Approaches

Due to the continuous scaling of the fabrication technology, the high power density became one of the most critical issues that limit the performance of the synchronous sequential circuits. Many approaches are introduced to manage the power consumption and to develop low-power designs. These approaches are targeting different levels of solutions from the device level to the architecture and system levels. This research is focused only on the microarchitecture and the circuit-level approaches which are suitable for any device or technology. The targeted approaches are classified into two main categories: runtime techniques which manage the power consumption at runtime, and design-time techniques which develop low-power digital integrated circuits.

2.2.1 Runtime Techniques

The power management runtime techniques are introduced to reduce the power consumption of the digital integrated circuit at runtime. Their main objective is to control the power consumption according to the functionality and the performance requirements while avoiding any chance to waste power.

2.2.1.1 Dynamic Voltage and Frequency Scaling (DVFS)

Dynamic Voltage and Frequency Scaling (DVFS) is widely used to reduce the power consumption. Figure 2.9 shows an illustrating example for the tradeoff among the supply voltage, the power consumption, and the gate delay. Increasing the supply voltage decreases the delay while increases the power consumption. On the other hand, decreasing the supply voltage increases the gate delay and decreases the power consumption. At runtime, the digital circuit scales down its operating frequency and then its supply voltage to reduce its power consumption without generating errors [23].

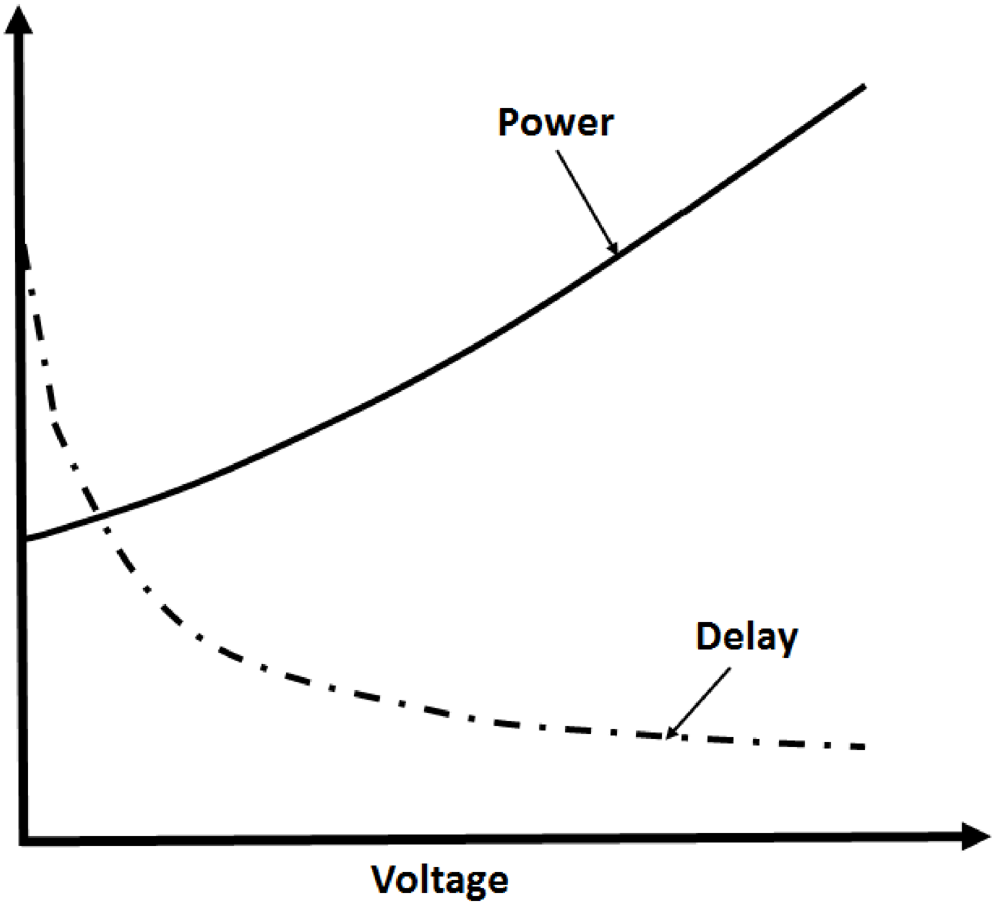


Figure 2.9: A generic overview of the relation between the power, the delay, and the supply voltage of the logic gates.

This approach reduces the power consumption efficiently especially for the dynamic power which is proportional to the operating frequency linearly and to the supply voltage quadratically. However, DVFS approach has to reduce the throughput of the integrated circuits to save power and it presents a tradeoff between a high throughput with a high power consumption against a low power consumption with a low throughput [15].

2.2.1.2 Clock Gating [24]

The main concept behind the clock gating is preventing the clocking of the registers if these registers do not need to write new data. The clock signal is gated not to reach these idle registers to save a large amount of power. The clock gating can be implemented using a gate which is called gate-based style or using a latch which is called latch-based style. While the latch-based style is more robust against glitches, the gate-based style is more efficient for the area and power perspectives.

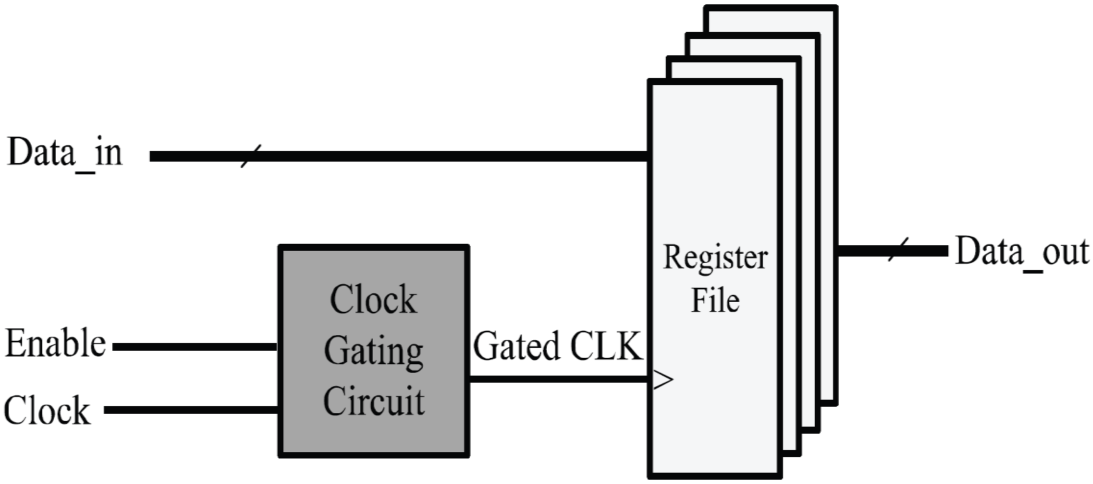


Figure 2.10: An example of clock gating for register files © [24].

For the synchronous sequential circuits, the clock signal has the highest switching activity which contributes significantly in the total power consumption. Figure 2.10 shows an example of the clock gating for some register files. The clock signal is enabled only when the register files are used and then it is disabled again while these register files are idle. As, the clock gating becomes one of the most common techniques used to manage the power consumption at runtime, the CAD tools integrate this technique for the ASIC cell-based design flow.

2.2.1.3 Power Gating [25]

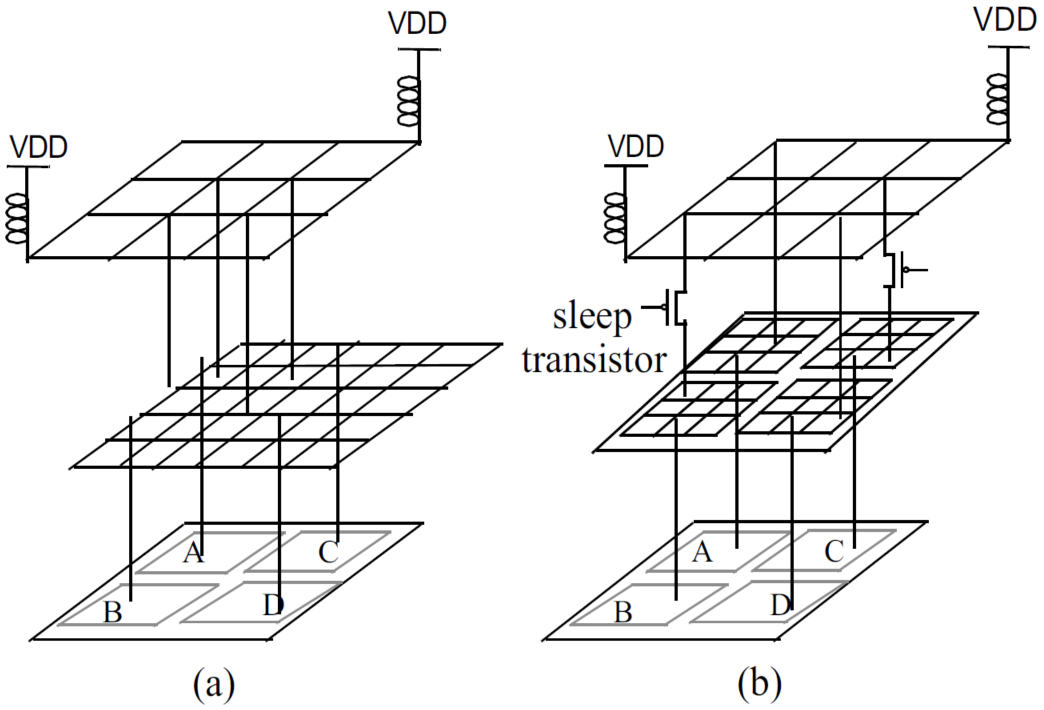


Figure 2.11: Two power supply systems. (a) Normal power supply system, (b) Power gating system © [25].

The power gating is used to reduce the leakage power of the idle blocks of the chip. As shown in Figure 2.11.b, a sleep transistor is used to control the power supply system. If the related block is idle, the sleep transistor is turned off to save the leakage power. However, turning this transistor on and off causes a considerable power supply current to be drawn in a short period of time which may lead to voltage fluctuations on the power distribution network.

2.2.2 Design-time Techniques

The objective of the low-power design-time techniques is to reduce the power budget of the synchronous sequential circuits during the design phase without affecting the final throughput. Unlike the power management runtime techniques, the low power design techniques are not controllable at runtime.

2.2.2.1 Multiple-Power Domains [26]

Multiple-power domains technique is one of the common solutions to reduce the dynamic power especially for the many cores chips. The chip is divided into different voltage domains, and every domain feeds a certain block of the chip which is compatible with the applied voltage. The different supply voltages are distributed according to the real demand of each block while different clocks are used to feed each block (domain) with the suitable frequency. The main objective of this approach is to assign a lower voltage supply as possible to every block of the chip while applying its efficient operating frequency. Consequently, voltage-level shifters and synchronizers are required to interface between the different voltage and clock domains.

2.2.2.2 Low-Power Clock Tree

The dynamic power of the synchronous circuits is dissipated through three major components: combinational logic clouds, registers, and clock tree. The clock tree consumes a large amount of the total power due to the high frequency and the switching activity of the clock signal which propagates through the clock tree buffers. According to [27], the clock tree consumes up to 40% of the total power of the synchronous sequential circuits. Also registers, which consist of flip-flops or latches, consume a large amount of the power budget that may reach 30% [28]. Reducing the power consumption of both clock tree and registers enhances the overall performance significantly and increases the reliability and lifetime of the fabricated chips.

The main idea is to decrease the clock frequency of the clock tree without decreasing the throughput. The Double-Edge-Triggered Flip-Flop (DETFF) was introduced as a potential solution to reduce the power consumption of the clock tree [29-37]. The DETFF captures data at the rising and falling edges of the clock cycle, so the clock frequency of the clock tree is reduced to the half (F/2) while the combinational logic stages still work at the maximum frequency (F). As a result, the power of the clock distribution network is reduced significantly. Many implementations were introduced for the DETFF trying to optimize its power delay product PDP, as shown in Figure 2.12 [29].

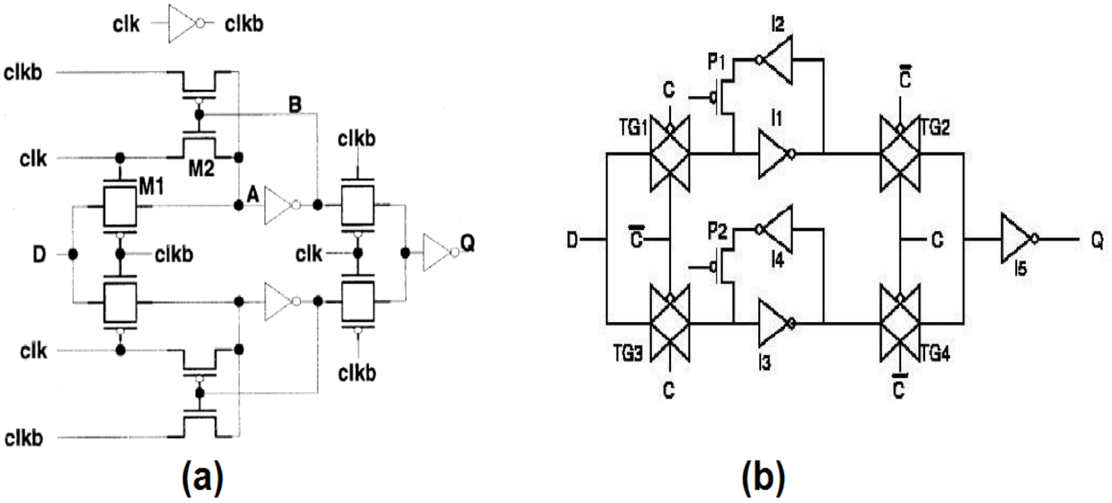


Figure 2.12: The transistor level design of the Double-Edge-Triggered Flip-Flop DETFF with two different implementations © [29].

Three multiplexers with feedbacks were used to implement the DETFF with about 50% area overhead, as shown in Figure 2.13 [30]. However, the latching operation of the multiplexer-based flip-flop is weak and it does not allow this flip-flop to drive large loads [31].

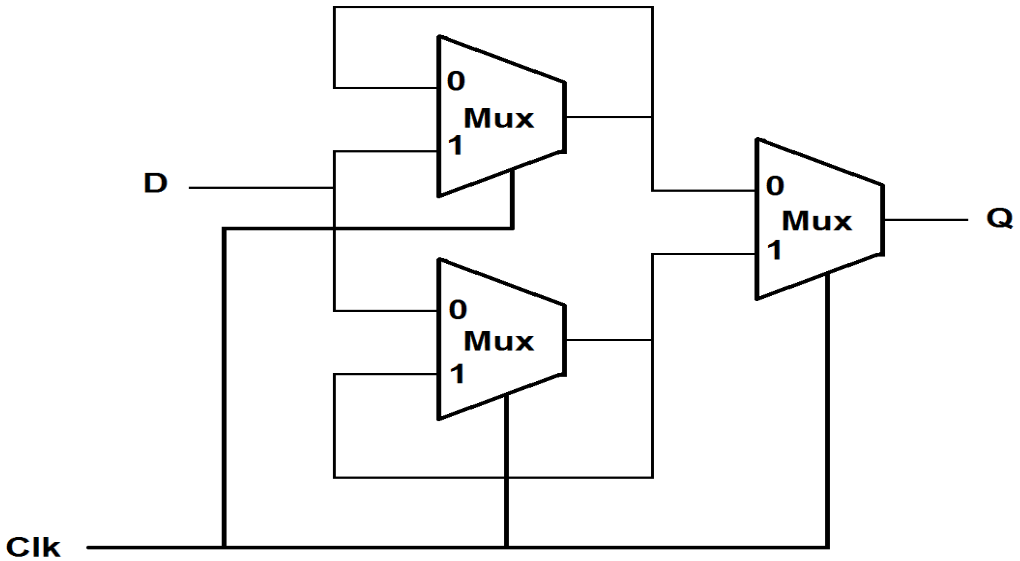


Figure 2.13: The implementation of the Double-Edge-Triggered Flip-Flop DETFF using three multiplexers © [29].

For another implementation, a single D-latch was also used to implement the DETFF in addition to a pulse generator [32] [33]. At each rising edge and falling edge of the clock signal, the pulse generator generates a small pulse to activate the D-latch and the input data is captured by the latch. Using a single latch instead of the flip-flop is better for the area and power perspectives [34].

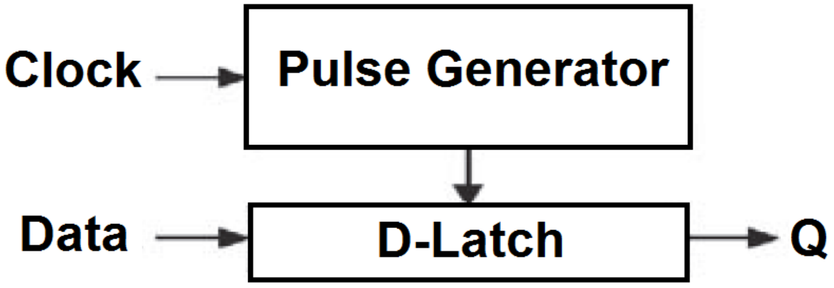


Figure 2.13: The implementation of the Double-Edge-Triggered Flip-Flop DETFF using D-latch and pulse generator © [32].

Although the double-edge-triggering concept reduces the power consumption of the synchronous designs, it has a huge area overhead that reaches about 70% of the total area [35]. In the literature, many papers mentioned the power reduction which is achieved by the DETFF against the common flip-flop, while ignoring the area overhead figures. This huge area overhead is a real obstacle against the adoption of the DETFF by the large synchronous sequential circuits. In another direction, the double-edge-triggering concept is not supported by the ASIC cell-based tools and on-the-shelf standard cells libraries. This means that using the double-edge-triggering concept for a large design needs a lot of work around to integrate the custom DETFF cell with the standard cells library. Implementing the double-edge-triggering concept using the common ASIC cell-based flow while reducing its area overhead is a good target that leads to a significant power saving.

Chapter 3: PVT Variability Mitigation

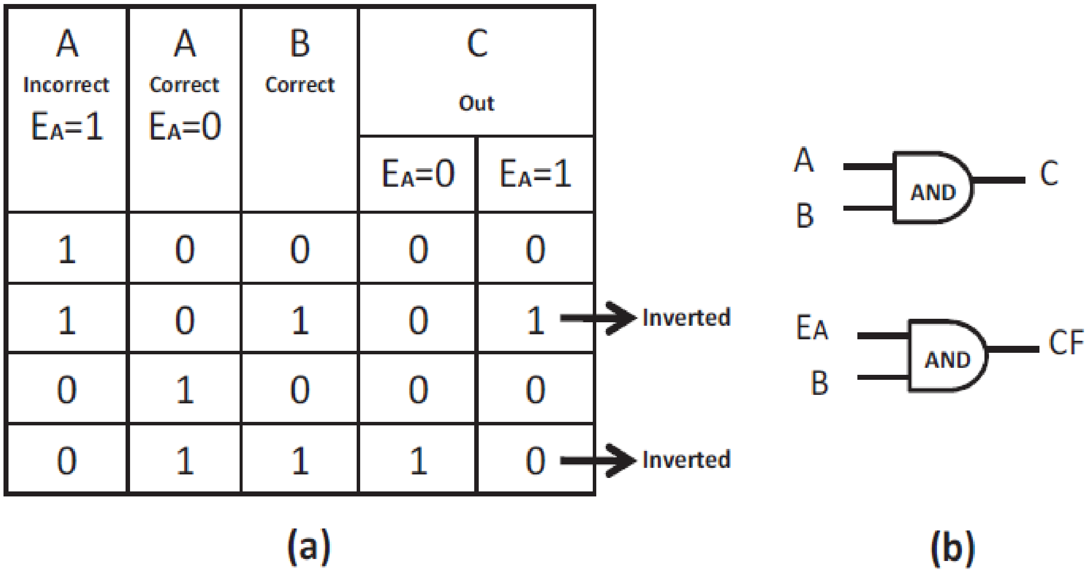
This chapter discusses our developed novel approaches for tolerating the PVT induced delay variations. The main objective is to tolerate the induced delays without flushing and restarting the pipeline. The other objective is to preserve the maximum throughput by keeping the maximum operating frequency.

3.1 Correction Function Technique

Because of PVT variations, critical stages are expected to examine time violations more than normal non-critical stages which have enough slack. According to Razor-based approaches, if an error is detected the faulty stages of the pipeline are flushed and the pipeline is restarted after the faulty state. The correct data should be restored from a shadow latch. According to this approach, many clock cycles are wasted through detecting the error, flushing the pipeline stages and restoring the data. Flushing the pipeline decreases the overall throughput of the circuit and wastes the power. In case of high error rates, flushing the pipeline is very costly and it should be avoided. If the pipeline stage has enough time to change its output after detecting an error, the pipeline will be able to continue its work without flushing.

3.1.1 Microarchitecture Overview

Correction function technique is able to detect the error at the input of the combinational logic stage and correct the output of the stage according to the effect of the propagated incorrect input. The combinational logic stages are based on the binary system which depends on two logic states: high and low. The incorrect input is the inverted logic state of the correct input. The incorrect input may lead to inverting the output of the combinational stage according to its functionality. Inverting the output of any combinational logic stage, due to the error at one of its inputs, can be determined by what we call: Correction Function.



A Incorrect E_A=1	A Correct E_A=0	B Correct	C Out		
			E_A=0	E_A=1	
1	0	0	0	0	
1	0	1	0	1	Inverted
0	1	0	0	0	
0	1	1	1	0	Inverted

Figure 3.1: AND gate and its correction function: (a) Truth table, (b) Logic gates © [5].

Figure 3.1 shows an AND gate, as a very simple example of a combinational logic, and its correction function. The AND gate has an input A from a critical unreliable stage. If an error is detected and the propagated value of A was wrong, the output C of the AND gate should be inverted if the output of the correction function CF is high. In case of error detection EA = 1, the output C of AND gate should be inverted if the other input B =1 and the output will only depend on the value of input A which means that every change in A should result in a change in the output C.

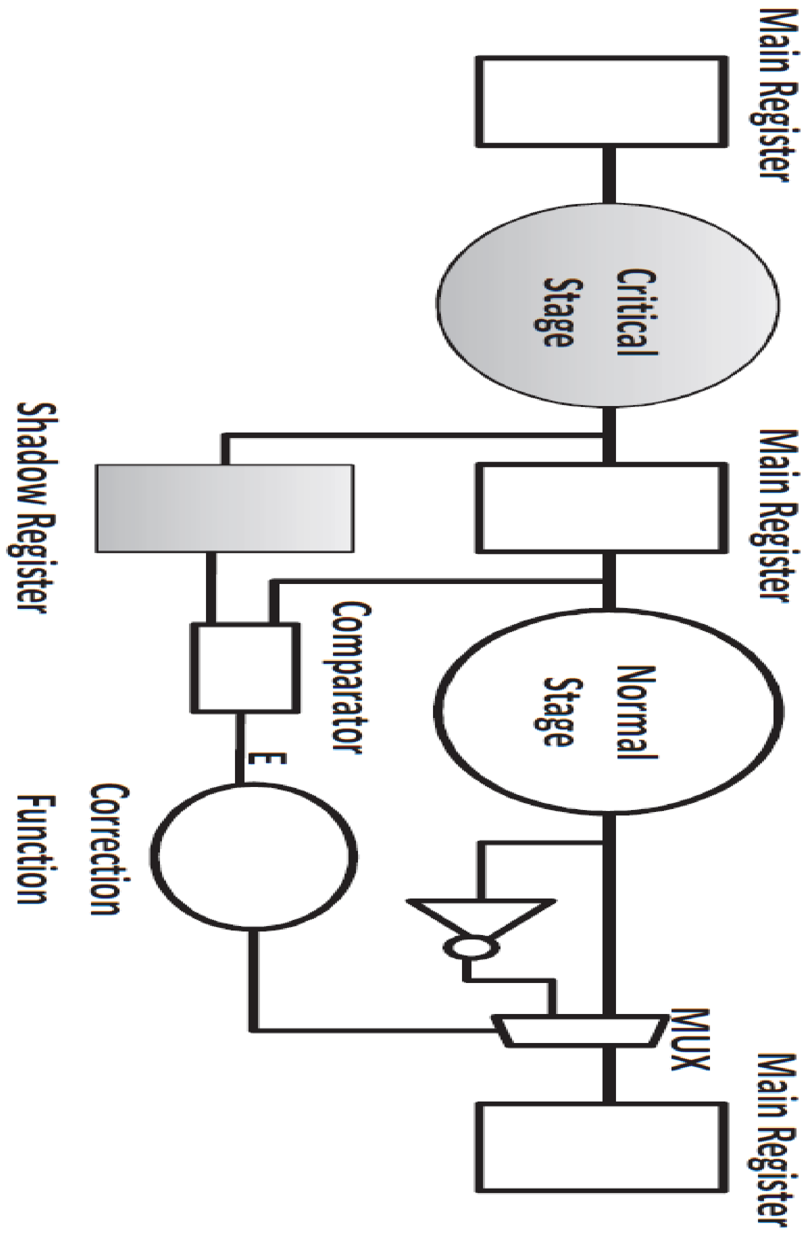


Figure 3.2: General architecture of the correction function approach for a two stages pipeline © [5].

Figure 3.2 shows the general architecture of the Correction Function approach. At the output of the critical stage, there are two registers with two different clocks. The main register captures the data at the rising edge of the normal clock while the shadow register captures a delayed version of the data at the rising edge of a delayed clock. The time interval between the two clocks is determined at design time according to the predicted induced delay. The data captured by the shadow register is compared to the data of the main register. If the two values are not identical, the error signal is activated (E=1), and the correction function determines if the output of the current stage, which is a normal stage, should be inverted or not by adjusting the selection signal of the MUX. Only critical stages with potential variations should be treated so that the correction function is only added to the normal stage which is the successor of the faulty critical stage.

$$T_{Delay} + T_{Error\ Detection} + T_{Correction\ Function} + T_{Selection} < T_{CP} \quad (3.1)$$

Constraint (3.1) illustrates that the total time consumed by the induced delay variation, detecting the error, correction function, and selecting the new output must not exceed the clock period, where T_{Delay} is the induced delay by the variations. $\mathrm{T}_{Error\ Detection}$ is the propagation delay of the error detection circuit. $\mathrm{T}_{Correction\ Function}$ is the propagation delay of the correction function. $\mathrm{T}_{Selection}$ is the time needed by the MUX to choose the output and T_{CP} is the clock period of the design.

3.1.2 Case Study, 4x4 bit Multiplier

To validate the correction function approach, a 4x4 bit multiplier is selected to be a case study. The multiplier was designed to be pipelined in two stages, as shown in Figure 3.3. It has two 4-bit input busses and one 8-bit output bus. One of the two input busses is assumed to be unreliable because of potential variations, which means that there are 4 critical stages and 4 potential errors. This approach considers a single-bit error detection and correction for simplicity, so that only one error can be detected and corrected each cycle. Isolating the faulty input within the first stage of the multiplier reduces the cost of the correction functions. Stage 0 is assumed to be the critical stage and the inputs B0, B1, B2 and B3, from Stage 0, form the unreliable bus. The correction functions control the outputs of Stage 1. Only three types of correction functions are used to make decisions about inverting the outputs because the multiplier consists of only three main components: AND gates, Half Adders and Full Adders.

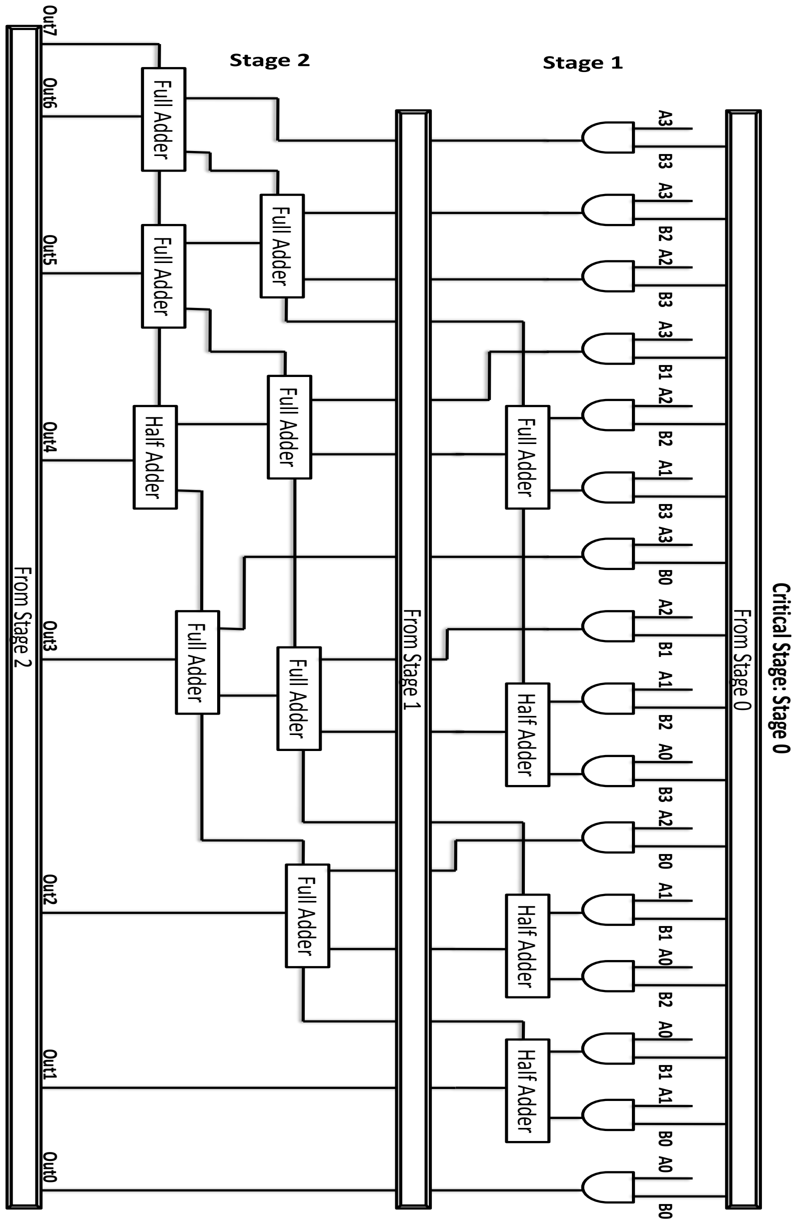


Figure 3.3: Two stages pipelined 4x4 bit multiplier © [5].

The correction function of the AND gate is shown in (3.2).

$$\mathrm{CF_{AND}} \;=\; \mathrm{E_{Bj}.\,A_i} \qquad (3.2)$$

For example: i=0 and j=0 in Figure 3.3, while A_i and B_j are the inputs and E_{Bj} is the error signal of the input B_j.

The correction functions of the Half Adder outputs are shown in (3.3) and (3.4).

$$\mathrm{CF_{Half\,Adder\,Sum}} \;=\; \mathrm{E_{Bj}.\,A_i + E_{Bl}.\,A_k} \qquad (3.3)$$

$$\mathrm{CF_{Half\,Adder\,Carry}} \;=\; \mathrm{E_{Bj}.\,A_i.\,A_k.\,B_l + E_{Bl}.A_i.B_j.A_k} \qquad (3.4)$$

For example: i=1, j=0, k=0 and l=1 in Figure 3.3, while $(A_i.B_j)$ and $(A_k.\ B_l)$ are the inputs. E_{Bj} is the error signal of the input B_j and E_{Bl} is the error signal of the input B_l.

The correction functions of the cascaded Full and Half Adders outputs are shown in (3.5) and (3.6).

$$\mathrm{CF_{Full/Half\,Adder\,Sum}} = \mathrm{E_{Bj}.\,(A_m \wedge (A_i.\,A_k.\,B_l)) + E_{Bl}.A_k.!(A_i.B_j)} \qquad (3.5)$$

$$\mathrm{CF_{Full/Half\,Adder\,Carry}} = \mathrm{E_{Bj}.(A_i{+}A_m).(\,A_k.\,B_l) + E_{Bl}.(A_i{+}A_m).(A_k.B_j)} \quad (3.6)$$

For example: i=0, j=2, k=1, l=3 and m=2 in Figure 3.3, while A_i, A_k, A_I, B_j and B_l are the inputs. E_{Bj} is the error signal of the input B_j and E_{Bl} is the error signal of the input B_l. These three types are used to correct the outputs of Stage 1 shown in Figure 3.3, While the symbol (.) means AND, (+) means OR, (!) means NOT and (^) means XOR.

3.1.3 Experimental Results

The pipelined 4x4 bit multiplier and the correction functions were designed in Verilog using Synopsys Design Vision targeting TSMC 90 nm technology. Table 3.1 shows that the area overhead of the correction function technique is 45.9% compared to the normal multiplier. The consumed power is also increased by 42.7% compared to the normal multiplier. The clock period is assumed to be 2 ns according to the worst case of the critical stage 0 which is not included by this design. The long path of the normal stage 1 is increased because of adding the detection and correction components. The new longest path became 1.28 ns instead of 1.01 ns. Stage 1 and Stage 2 are non-critical stages so that they do not need the full clock cycle to finish their computations.

The tolerated induced delay of the critical stage T_{Delay} can be up to 0.72 ns while the rest of the clock period is consumed by the error detection circuit and the correction function. So the correction function technique has the flexibility to tolerate the induced delays up to 35% of the clock cycle without flushing the pipeline.

Table 3.1. Results of Applying correction functions on 4x4 bit multiplier.

4x4 bit Multiplier	Clock Period = 2 ns		
	Area (um^2)	**Power (uW)**	**Delay (ns)**
No Correction Function	1349.227	132.249	1st Stage: 1.01 2nd Stage: 1.72
Correction Function	1967.827	188.669	1st Stage: 1.28 2nd Stage: 1.72

3.1.4 Conclusion

Using correction functions mitigates the potential impacts of PVT variations at the cost of power and area. If an error is detected, the pipeline is able to correct the error at run-time. The correction function approach does not have to flush the pipeline and restore the correct data. The overhead of area (45.9%) and power (42.7%) is related to the internal architecture of the multiplier which is only two pipeline stages. The overhead of power and area is expected to be reduced for larger designs with many pipeline stages because only critical stages need detection and correction functions. However, the complexity of the correction function is expected to be increased dramatically for bigger logic stages. The more unreliable inputs feeding the non-critical stage, the more complexity of the correction functions which leads to higher area and power overhead.

3.1.5 Recommendations

The correction function is not a generic solution because it depends on the functionality of the combinational stages and its complexity increases for more complicated designs. The area and power overhead is relatively high and does not support the objective of power saving. The results of the Correction Function Technique lead to some recommendations which are listed as follows:

- The new proposed approach should preserve the concept of no pipeline flushing to preserve the maximum throughput.

- The new proposed approach should be a generic microarchitecture that does not depend on the functionality of the combinational logic stages.

- The new proposed microarchitecture should integrate the concept of error detection with the concept of time relaxation.

- Instead of tolerating the induced delay variations during one clock cycle, the new proposed microarchitecture should tolerate the induced delays through multi clock cycles to get more flexibility.

3.2 Error Recovery System Using Taps (ERSUT)

The previous technique, Correction Function, detects the induced errors of the critical stage and corrects the error without flushing the pipeline by inverting the output of the next non-critical stage depending on a combinational logic which is called correction function. The area and power overhead of the correction function approach is relatively high due to the combinational logic overhead. The correction function is also dependent on the functionality of the combinational stage and its complexity increases for more complicated designs. So, the objective of the new approach is to solve the problem of time violations, caused by PVT variations, with a generic microarchitecture that does not depend on the functionality of the combinational logic stages. The proposed microarchitecture detects the error and restores the correct data without flushing the pipeline by borrowing time from next stages. The critical stage is defined as a stage that violates the timing conditions due to the effects of PVT variations. Non-critical stages are the stages affected by PVT variations but still have enough slack to tolerate their PVT variations and to compensate the timing violation of the critical stage. As a result, even if the non-critical stages are affected by the PVT variations, there is still enough slack to be borrowed for error recovery.

3.2.1 ERSUT Microarchitecture Overview

The target of this microarchitecture is to avoid the high penalty of flushing the pipeline in case of error detection because the modern processor architectures, using deep pipelines, increase the demand of flexible self-healing pipelines. The new approach depends on using pulsed latches instead of flip-flops for the critical stage and its successive non-critical stages, while the rest of the design is still based on flip-flops as illustrated in Figure 3.4. Pulsed latches preserve data while the clock signal is low and transfer data while the clock signal is high.

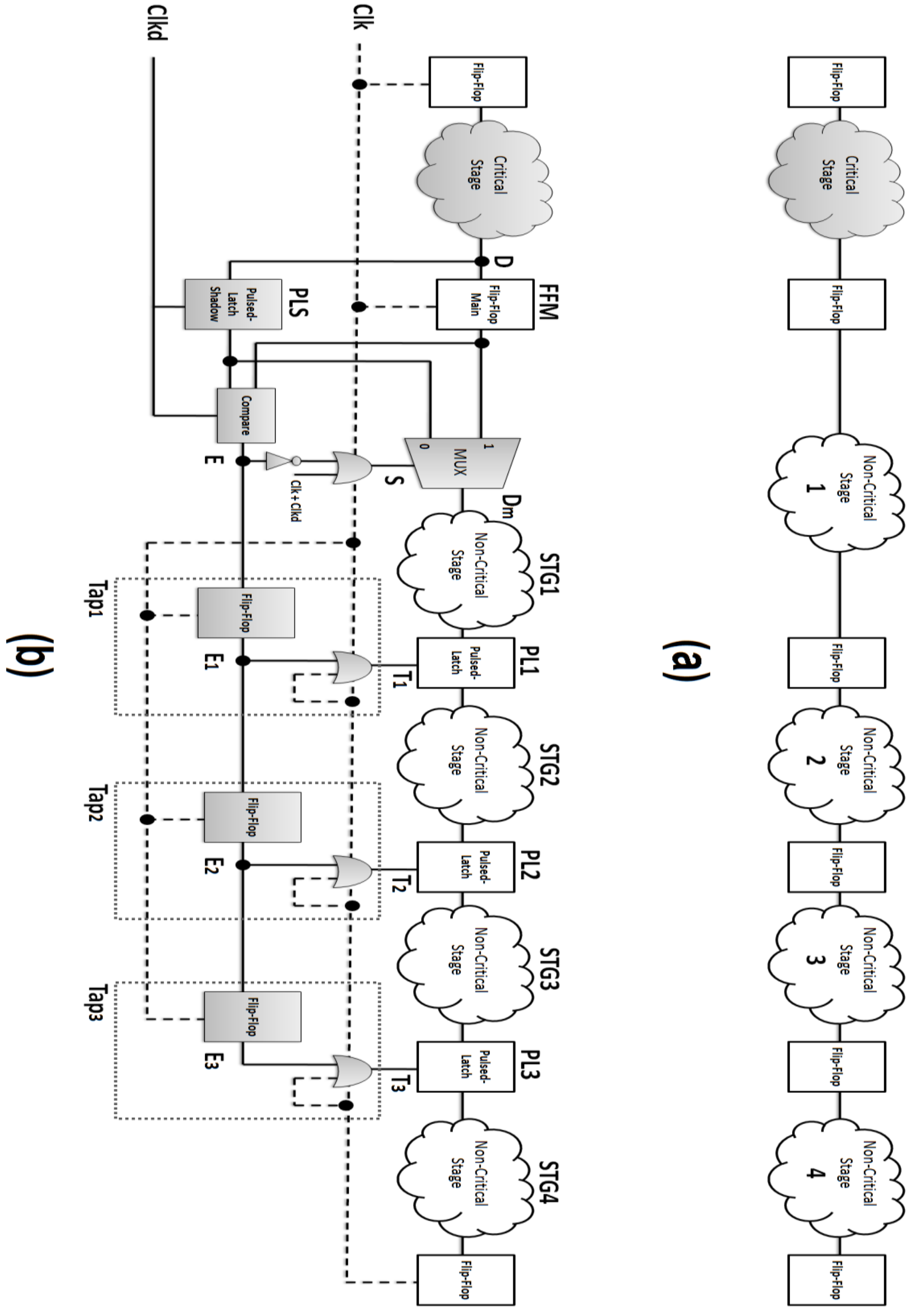
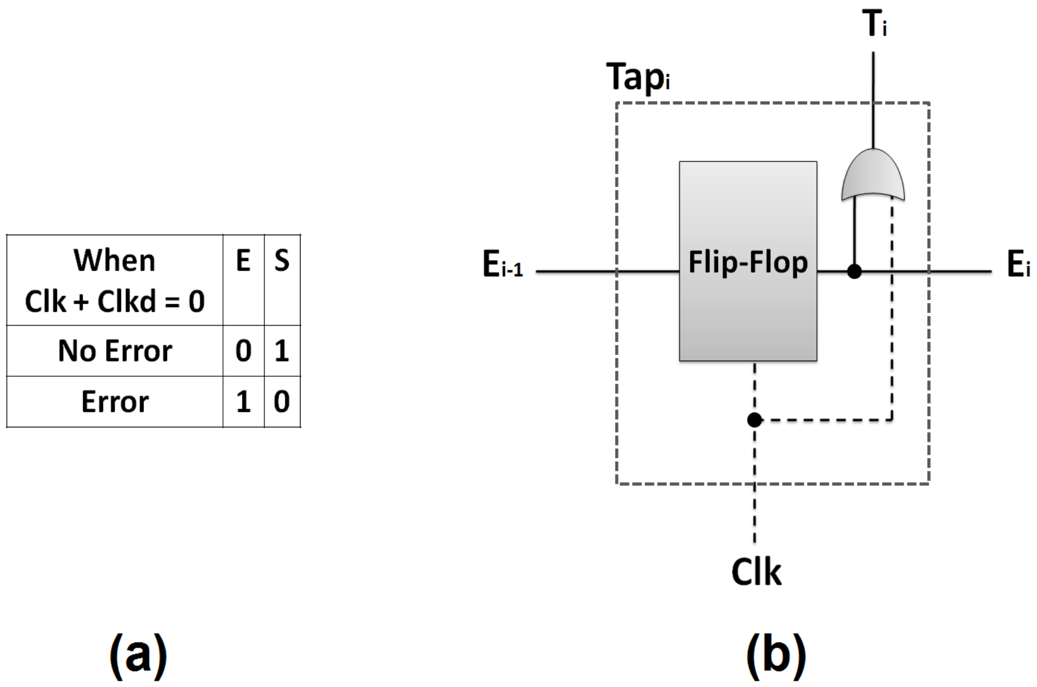


Figure 3.4: A general overview of ERSUT microarchitecture © [38]. (a) Conventional pipeline microarchitecture. (b) ERSUT microarchitecture: Delayed copy of the data D is stored in PLS and compared to the data saved by FFM. In case of error, the error signal E is activated and propagates to make pulsed latches PL1, PL2, and PL3 transparent, equivalently stealing slack from STG1, STG2, STG3, and STG4.

The conventional pipeline consists of cascaded combinational stages surrounded by flip-flops as shown in Figure 3.4.a. The ERSUT, shown in Figure 3.4.b, has two clocks: the normal clock Clk and another delay-clock Clkd. Clkd is the same normal clock Clk with an extended pulse width PW to cover the potential PVT induced delay. To avoid the cost of the additional clock domain, Clkd is locally generated for the targeted stages, and the two clock signals are affected by the same variations and circumstances of routing from the same source. The output data D, computed by the critical stage, is stored by the main flip-flop (FFM) at the rising edge of Clk. Based on the clock Clkd, a delayed copy of D is stored by the shadow pulsed latch (PLS). The two values are compared, and if they are not identical, the error signal E is activated.



When Clk + Clkd = 0	E	S
No Error	0	1
Error	1	0

Figure 3.5: The internal mechanism of ERSUT © [38]. (a) Truth table of the multiplexer selection signal S based on the error signal E. (b) Tap architecture: If (E = 1) → T = 1 (independent of Clk); as a result, the related pulsed latch becomes transparent.

As shown by the truth table in Figure 3.5.a, the error signal E controls the multiplexer to determine which of the two copies of the data is transferred. If there is no error, the copy of the main flip-flop FFM propagates to the next stage. In the case of an error, the correct data which are stored by the PLS propagates.

OR-ing clock signals with the inverse of the error signal restores the selection signal S of the multiplexer to the normal state at every new clock cycle to allow data from the main flip-flop FFM to propagate instead of the PLS. To allow stealing time from the next stages, the error signal E propagates through controlling elements which we call “Taps”. As shown in Figure 3.5.b, the tap consists of a flip-flop to capture the error signal E_i for the next clock cycle in addition to an OR gate which is used to generate the tap-controlled clock signal T_i. If an error E_1 is propagated to the first tap, the related pulsed latch PL1 is open (transparent) during the whole clock cycle. This means that STG1 and STG2, shown in Figure 3.4.b, are combined together for only one clock cycle. In this way, the slack of the two stages is used to compensate the PVT induced delay that occurred in the critical stage. If this slack is not enough, the error signal E_1 propagates to the next tap. The second tap controls the clock signal of the next pulsed latch PL2 and this scenario continues for the rest of the taps in the design. The number of these taps is determined in the design phase. Whenever the error signal E returns low, the taps switch off, and then, PL1, PL2, and PL3 progressively return to their normal operation as pulsed latches (transparent only during the active period of Clk).

Figure 3.6 shows the timing diagram of the proposed approach in case of error detection. The data D is captured at the main flip-flop FFM by the signal Clk. The value of D is changed due to variations, so the correct value is missed by FFM and it is not transferred to STG1. The shadow pulsed latch PLS captures the correct value of D since it is controlled by the clock Clkd. The two values are compared, and the error signal E goes high. The multiplexer selection signal S is changed from high to low. It allows the multiplexer to pass the delayed version of D from the shadow pulsed latch. Dm is the output of the multiplexer that is delivered to STG1. The error signal E is transferred to the first tap at the beginning of the next clock cycle Clk. E_1 is changed from low to high and turns the tap-controlled clock signal T_1 high for a complete clock cycle. The error signal propagates through the taps as an error wave (E, E_1, E_2, and E_3) at the beginning of each clock cycle, controlling the tap-controlled clock signals (T_1, T_2, and T_3). Each tap combines the two adjacent stages for one clock cycle and then splits them at the next clock cycle. As the tap-controlled pulsed latches are open for one clock cycle, some timing constraints are necessary to avoid racing through short paths.

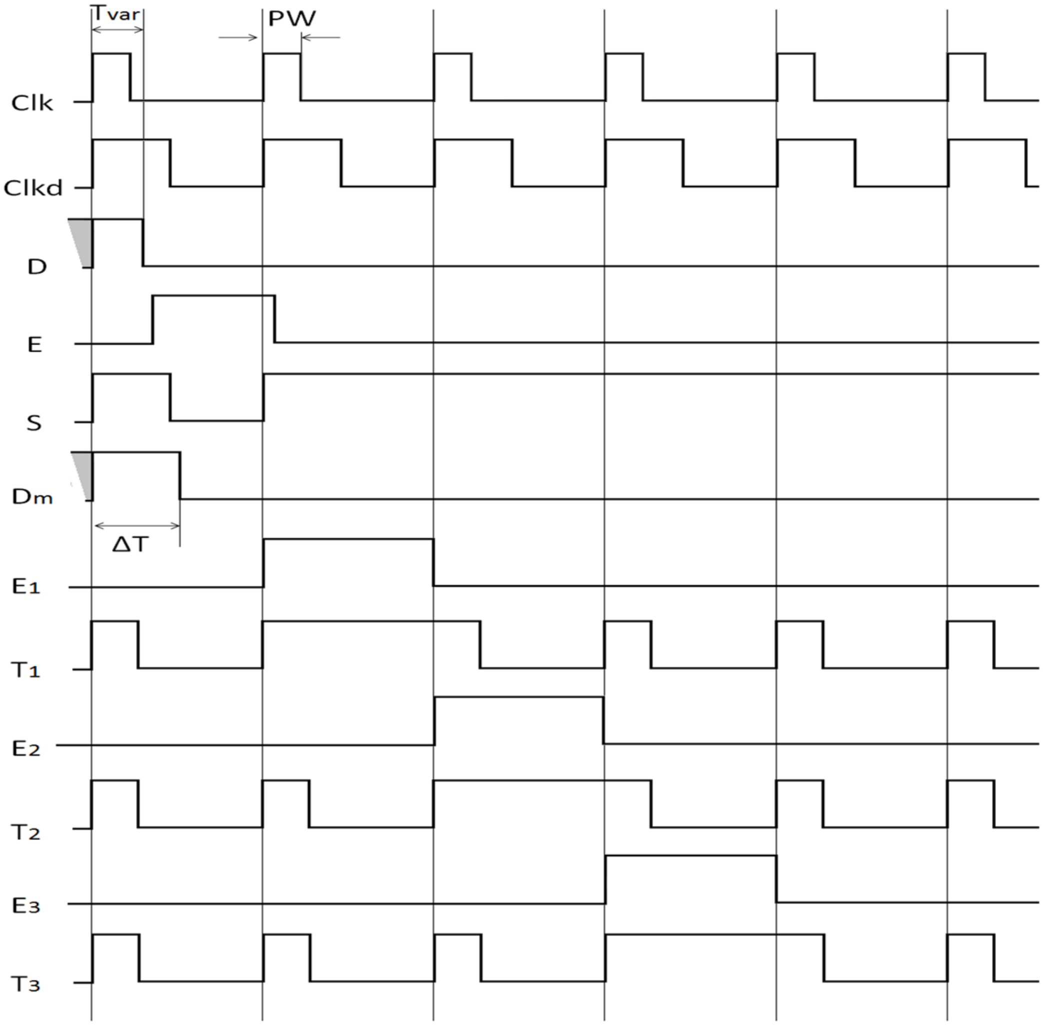


Figure 3.6: The timing diagram of ERSUT main signals in case of error detection © [38].

3.2.2 ERSUT Timing Constraints

Turning the pipeline with potential variations into a flexible self-healing pipeline needs some timing constraints. These constraints are classified into two categories: timing constraints to determine the minimum number of Taps, and short-paths constraints to guarantee a correct data flow.

3.2.2.1 Determining the Number of Taps

To determine the number of Taps for a certain design, the total induced delay is calculated by equation (3.7).

$$\Delta T = T_{var} + T_{detection} + T_{Selection} \qquad (3.7)$$

The total induced delay ΔT includes the induced delay of PVT variations T_{var}, the error detection delay $T_{detection}$ which is the maximum propagation delay of the comparator, and the valid data selection delay $T_{Selection}$ which is the maximum propagation delay of the multiplexer. The slack of the non-critical stage is given by equation (3.8), where S_i is the slack of the non-critical stage i (STGi). T_{cp} is the clock period and Tm_i is the maximum propagation delay of STG_i.

$$S_i = T_{cp} – Tm_i \qquad (3.8)$$

As shown by the simplified version of ERSUT microarchitecture in Figure 3.7, a pipeline is used as an illustrating example. The clock period is assumed according to the worst-case of the critical stage to be 7 time units. In normal operating conditions, the data is supposed to reach PL1 after 7 time units at T7, PL2 at T14, PL3 at T21, and then the last flip-flop at T28. The induced delay is assumed to be 4 time units due to PVT variations.

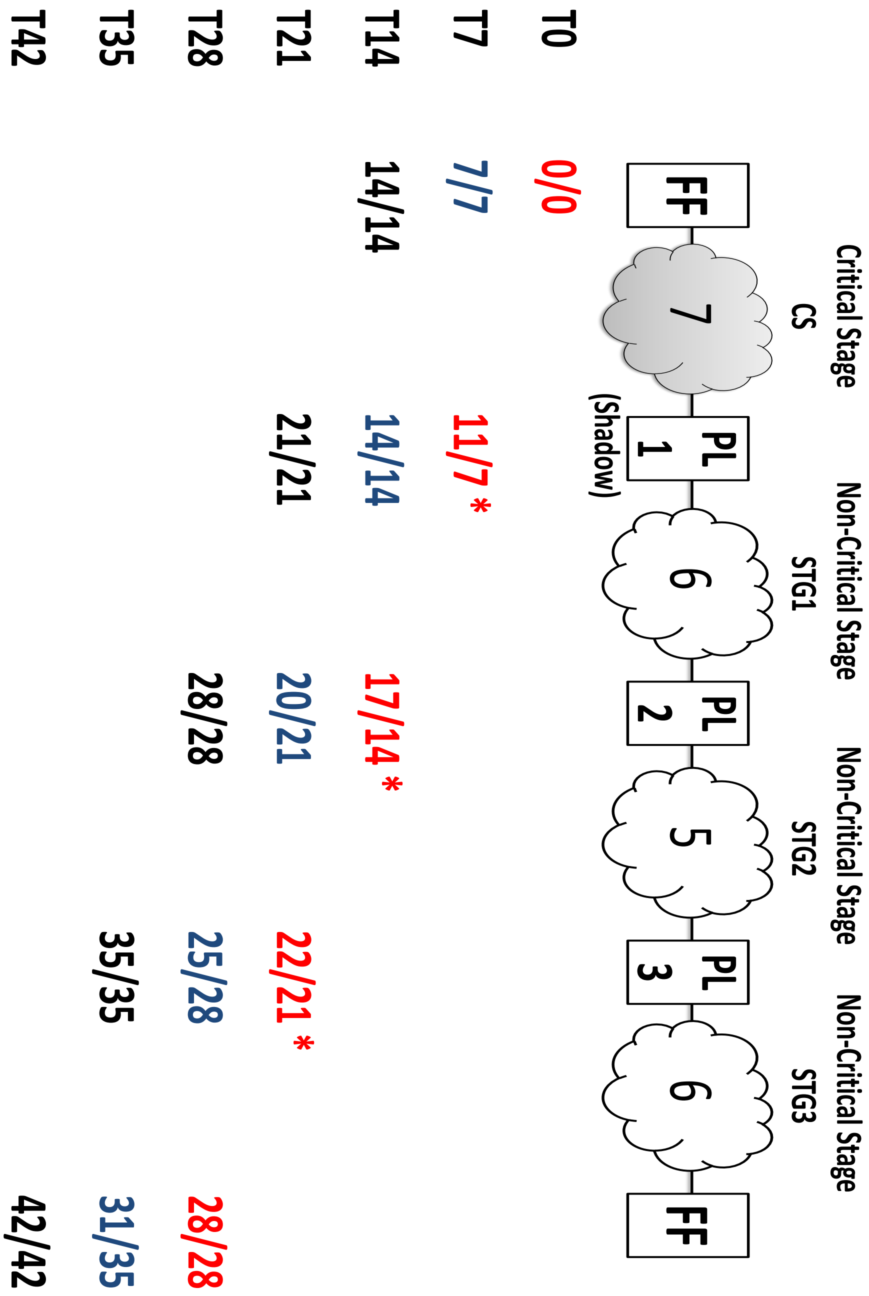


Figure 3.7: An example of a pipeline with a critical stage illustrating the timing constraints. Symbol (*) refers to time violation © [38].

The available slack per each successive stage determines the number of non-critical stages used to compensate the induced delay. For more balanced stages, it is still possible to steal time because the critical path in one stage may steal slack from non-critical paths of the next stages even if these stages have balanced critical paths. The probability of cascaded critical paths through successive stages is very low.

As shown in Figure 3.7, to compensate the induced delay, we need to borrow 4 time units from the next stages. STG1 has 1 time unit as a slack, STG2 has 2 time units, and STG3 has 1 time unit. The summation of all these slacks, which is 4 time units, is used to compensate the induced delay. In this example, we suppose (for simplicity) that the PVT variations affect only the critical stage. In the implementation, a worst-case of PVT variations is applied to all stages and then the slack of the non-critical stages is determined based on the definition of the non-critical stage. The delayed correct data reaches PL1 at T11 (T7 + 4 time units). The propagated data is able to go through the next pulsed-latch PL2 and it arrives at T17 (T14 + 3 time units). Then it propagates through the non-critical stage STG2 to reach PL3 at T22 (T21 + 1 time unit). It is observable that the induced delay is reduced due to the available slack of the non-critical stages STG1 and STG2. At the end of STG3, the correct data reaches the output flip-flop in time without delay at T28. Each tap-controlled pulsed-latch at the output of each non-critical stage is opened for a whole clock cycle and it allows two successive correct data values to be transferred. For example, the pulsed-latch PL2 is open for a clock cycle from T14 to T21. Therefore, it allows two successive correct values to be transferred at T17 and T20. This pulsed-latch PL2 is turned back to its normal operation during the next clock cycle beginning at T21.

$$\Delta \mathrm{T} < \mathrm{S}_{\max}\text{, where } \mathrm{S}_{\max} = \sum_{i=1}^{N} S_i \qquad (3.9)$$

Formula (3.9) is the main timing constraint, where ΔT is the total induced delay. $\mathrm{S}_{\max}$ is the maximum summation of the available slacks and N is the number of non-critical stages used to compensate ΔT. The timing constraints (3.7), (3.8), and (3.9) are important to determine the number of stages and Taps needed to tolerate the potential variations. Based on these constraints, the error recovery system is built and the short paths of the controlled stages are re-adjusted.

3.2.2.2 Short Paths Constraints

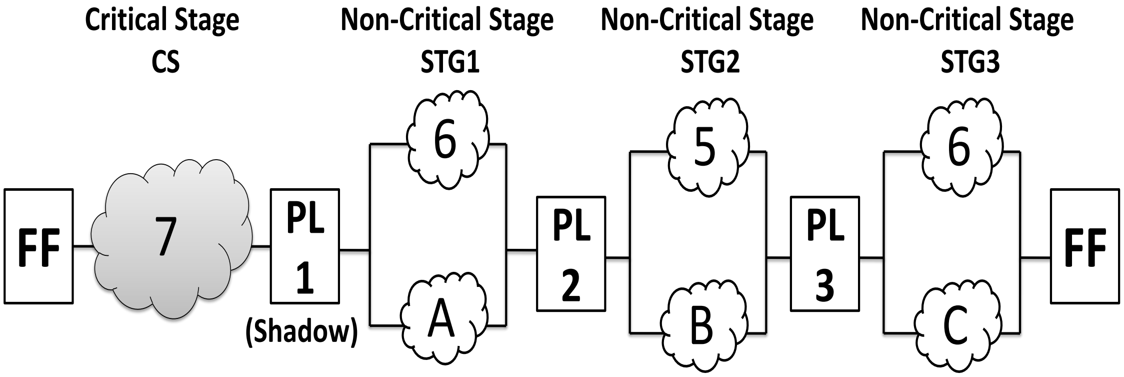


Figure 3.8: A pipeline with a critical stage and Taps illustrating the short paths constraints where A, B and C are the values of the short paths © [38].

As shown in Figure 3.8, a pipeline with a critical stage is used (as a simple example) to illustrate the short paths constraints. We assume that there are two cascaded versions of data D_T and D_{T+1} propagating through the pipeline, where D_T is the delayed data and D_{T+1} is the Data of the next clock cycle. The worst case scenario is the racing between the two versions when D_T propagates through the long paths of STG1, STG2, and STG3 while D_{T+1} propagates through the short paths A, B, and C. The time separation between the two events at PL1 is (T_{cp} - S_{max}) by assuming that S_{max} is the upper bound of ΔT. The target of the short paths constraints is to avoid racing between the two cascaded data. If the clock period related to the critical stage is 7 time units and the induced delay is assumed to be 4 time units, the valid data reaches PL1 at T11 (T7 + 4 time units) and this correct data reaches PL2 at T17.

A new valid data reaches PL1 at T14 by assuming that the critical stage does not generate more errors (worst case scenario). If this new data propagates through the short path A, it must reach PL2 after T17, so that the minimum value of A has to be more than 3 time units. The delayed correct data reaches PL2 at T17, PL3 at T22 and the output flip-flop at T28 through the long paths of the non-critical stages STG1, STG2, and STG3. The next valid data reaches PL1 at T14 and then propagates by the transparent pulsed-latches

PL2 and PL3 through the short paths A, B and C. This means that the next valid data may propagate and damage the previous delayed correct data.

$$Where|_{n=1}^{N}\{ \sum_{i=1}^{n} T_{Shi} > \sum_{i=1}^{n} Tm_i - (T_{cp} - S_{max})\} \qquad (3.10)$$

$$T_{Shi} < Tm_i \qquad (3.11)$$

To keep the two cascaded data without losing any of them, the lower bound constraint (3.10) and the upper bound constraint (3.11) are applied for the short paths of the non-critical stages, where N is the number of the non-critical stages used to get slack and T_{Shi} is the propagation delay of the short path of stage i.

$$A > S_{max} - T_{cp} + Tm_1 \qquad (3.12)$$

$$A + B > S_{max} - T_{cp} + Tm_1 + Tm_2 \qquad (3.13)$$

$$A + B + C > S_{max} - T_{cp} + Tm_1 + Tm_2 + Tm_3 \qquad (3.14)$$

Formulas (3.12), (3.13) and (3.14) extracted from (3.10) are used to determine the values of A, B and C, where $Tm_1 = 6$, $Tm_2 = 5$, $Tm_3 = 6$, $S_{max} = 4$ and $T_{cp}= 7$. Based on the previous three inequalities, (3.15), (3.16) and (3.17) can be extracted.

$$A > 3 \qquad (3.15)$$

$$A + B > 8 \qquad (3.16)$$

$$A + B + C > 14 \qquad (3.17)$$

$$A < 6, B < 5, C < 6 \qquad (3.18)$$

Inequalities (3.18) prevent turning the short path to long path that may affect the total slack used to tolerate the induced delay. For example if A is selected to be 4 time units and C is selected to be 5 time units then B should be 6 Time units to satisfy (3.16) and (3.17), but this value turns B from short to long path and it reduces the total slack from 4 to 3 time units. This means that the values of short paths A, B, and C must be selected to preserve all constraints: (3.15), (3.16), (3.17), and (3.18).

3.2.3 Meta-stability Analysis

Meta-stability is a challenging issue that affects the reliability of all Razor-based approaches [22]. As the meta-stability resolution is not time-deterministic, we assume that the resolution of the data saved by the main flip-flop FFM (in Figure 3.4.b) can be achieved during one of three regions through the clock cycle. The three regions of meta-stability resolution A, B, and C are illustrated by Figure 3.8. If the main flip-flop FFM resolves its meta-stability during region A which includes the pulse width of the normal clock Clk and the slack of the next non-critical stage, a delayed copy of the data propagates to the next stage without raising an error flag. This delay is tolerated by the default relaxation of using pulsed-latches with a transparent window equals to the pulse width of the normal clock Clk.

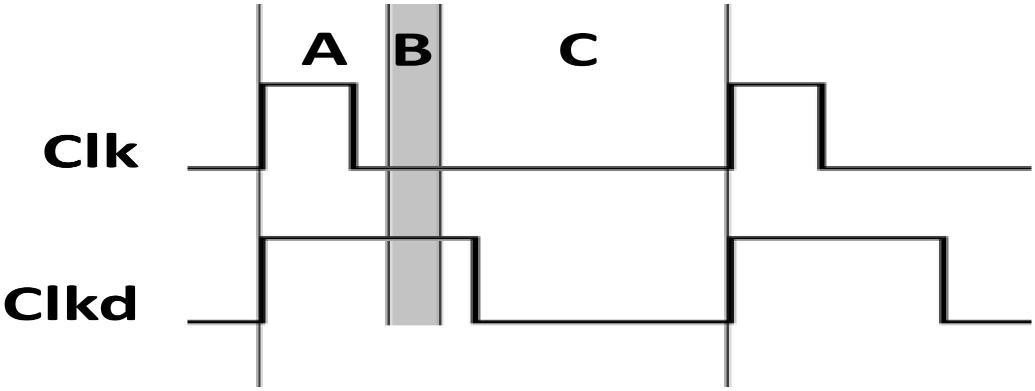


Figure 3.9: The meta-stability resolution regions for ERSUT © [38].

If the meta-stability is resolved during region B, a delayed copy of data propagates to the next stage and the error signal E has enough time to be deactivated. Based on the final version of the deactivated error signal, the Taps do not work to compensate the induced delay of the meta-stability and the next non-critical stage may violate the timing conditions. If the meta-stability of the main flip-flop FFM is resolved during region C, the error signal E is activated and it has no enough time to be deactivated and the signal E is latched by Clkd. As a result, the Taps are activated to compensate the induced delay. Based on this analysis, ERSUT reduces the probability of the meta-stability violations because it tolerates the late meta-stability resolution for the most of the clock cycle.

3.2.4 Case Study, 16x16 bit MAC Unit

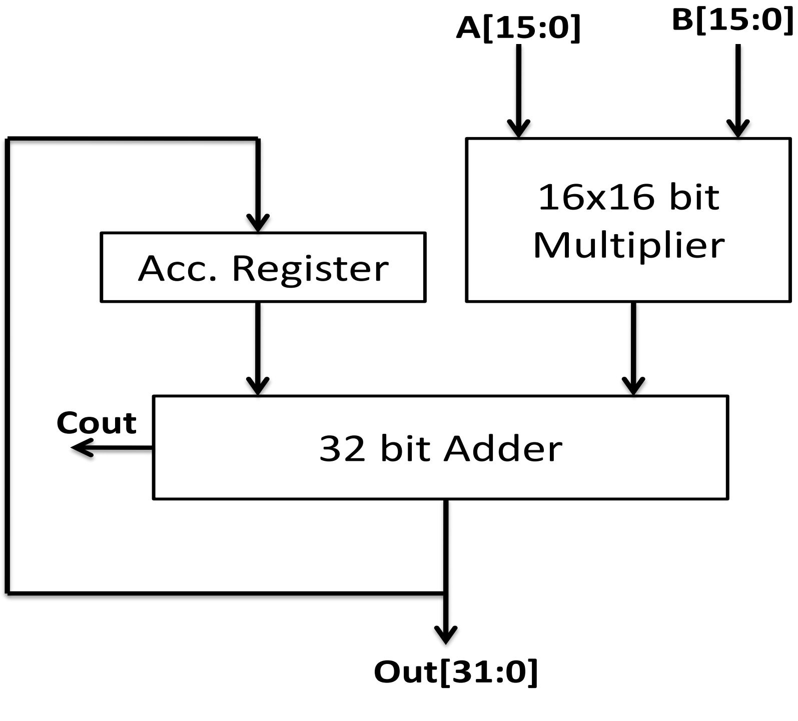


Figure 3.10: The general architecture of the 16x16 bit MAC unit © [39].

The MAC unit is an important building block in many applications such as deep machine learning and digital signal processing. A 16x16 bit MAC unit, shown in Figure 3.10, was selected to be the case study. This 16x16 bit MAC unit has about 854 flip-flops and the total number of its critical and non-critical paths is about 4846. Although the area of the selected MAC unit is greater than most of the ISCAS'89 benchmark circuits, its functionality (multiply and accumulate) is easy to be traced and verified which facilitates the functional verification of the proposed approach. In another direction, the 16x16 bit MAC unit was selected carefully as a worst-case scenario to examine the proposed approach for two main reasons: 1) The number of short paths is much greater than the number of the critical paths healed by the proposed approach, which tests the efficiency of our solution from the area overhead point of view, and 2) The ratios between the clock period and the readjusted short paths are very high and the maximum ratio is up to 8.7x which is considered as an aggressive test for the cost of the proposed approach.

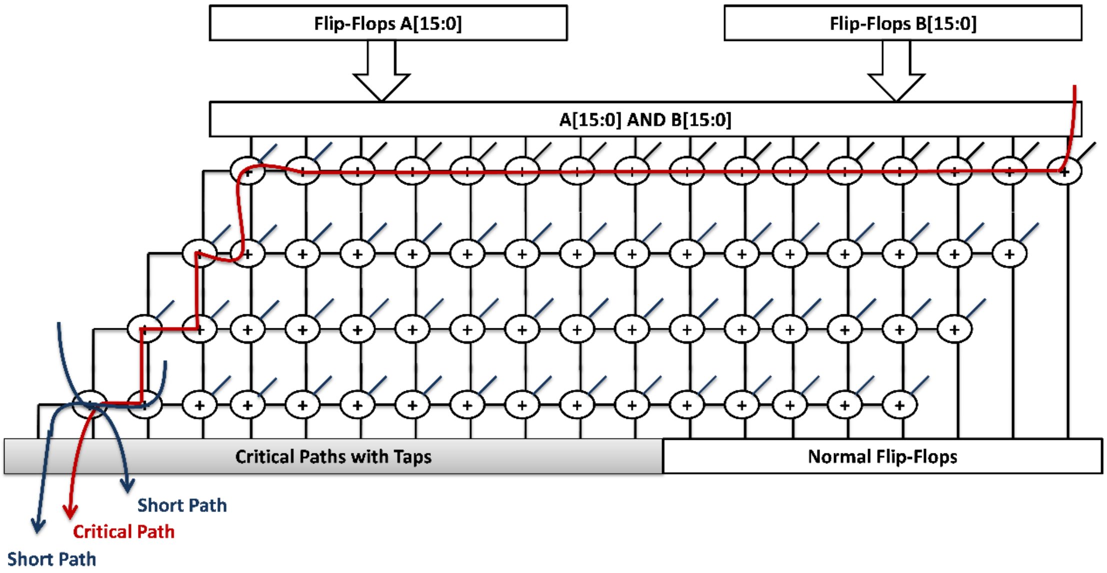


Figure 3.11: The simplified architecture of the first critical stage, showing examples for a critical path (in red) and potential short paths (in blue) © [39].

The first reason indicates that many short paths need to be readjusted while the second reason shows that most of the readjusted short paths need more delay elements to satisfy the timing constraints because their delays are much smaller than the clock period of the MAC unit as illustrated in Figure 3.11. These two factors aggressively test the efficiency of the proposed microarchitecture from area and power point of view. For bigger designs, just like industrial processors with more complicated designs, the area and power overhead of the proposed approach is expected to be reduced. For these complicated designs, only critical stages, which are the minority, need to be healed while the rest of the design operates normally.

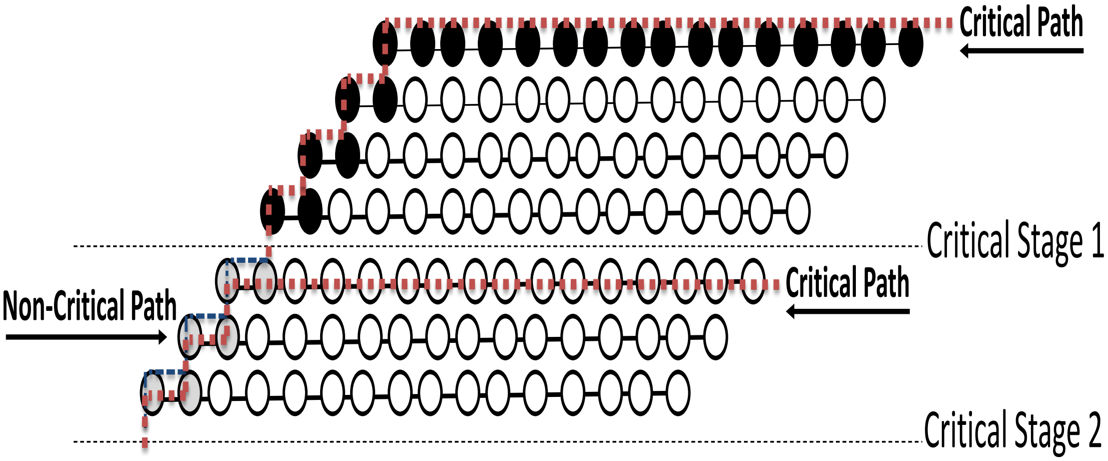


Figure 3.12: A simplified architecture of the first two critical stages where the critical path in one stage borrows slack from a non-critical path of the other stage © [39].

Even the critical stages are cascaded, the critical path of each stage still borrows enough slack from the non-critical paths of the next critical stage as shown in Figure 3.12. For more complex designs, it is expected to get better results as the number of the critical stages are much less than the non-critical stages compared to the case of MAC unit. Our future direction is to investigate the validity of this resilient approach for ISCAS’89 benchmark which contains a wide variety of digital circuits.

3.2.5 Experimental Results

The RTL of a conventional MAC unit and a MAC unit with ERSUT are designed in Verilog using the structural type of coding. The two MAC units are identical and they have the same specifications to get a fair comparison. They are implemented using Synopsys synthesis tool, Design Vision, with TSMC 90 nm as the target technology. The synthesis tool is forced not to touch the hierarchy of the two MAC units to keep them identical. The place and route is done by using Synopsys IC Compiler. The clock period of the MAC unit is 12.5 ns according to the worst case of the critical path based on the static timing analysis of Synopsys Design Vision. The maximum PVT induced delay variation that can be tolerated is up to 20% of the clock period. All critical paths affected by the induced delay variations are healed by ERSUT for the two cases.

Table 3.2. Results of Area and Power for the Conventional MAC unit and the MAC unit with ERSUT.

MAC UNIT	Clock Period = 12.5 ns				
	Conventional	**ERSUT**			
		Tolerate 10%	Overhead	Tolerate 20%	Overhead
Area (um^2)	48835.076	55638.436	13.93%	59054.590	20.93%
Power (mW)	1.425	1.670	17.21%	1.791	25.7 %

Table 3.2 shows the area and power figures for both the conventional MAC unit and the MAC unit with ERSUT. MAC unit with ERSUT tolerates induced delays up to 20% of its clock period while the area overhead of the new approach is 20.93% compared to the conventional design. Most of this cost is due to re-adjusting the short paths of the non-critical stages controlled by the Taps. The power overhead of the new approach is 25.7% to tolerate 20% of the clock period. The area and power overhead is not relatively high in comparison to Correction Function techniques. However, the MAC unit has many short paths to be readjusted per each healed critical path which increases the hardware overhead.

3.2.6 Conclusion

Error Recovery System Using Taps (ERSUT) is a promising approach to tolerate the PVT induced delays. This approach detects the error and reconfigure the pipeline stages for recovery without flushing the pipeline. At typical operating conditions, ERSUT can be used to increase the operating frequency (in case of no PVT variations) and then the ERSUT hardware can be used to tolerate the timing violations of the frequency scaling up. Unlike Correction Function technique, ERSUT is not dependent on the complexity of the combinational logic stages functions. However, the complexity of error detection and valid data selection circuits is consuming the potential gain of ERSUT because these circuits add a delay overhead to the induced delay that should be tolerated. The relatively high-power overhead is acting as an obstacle against one of the two major objectives which is power saving. Although ERSUT decreases the probability of the meta-stability, the issue is still existing and it threatens the chance of adopting ERSUT by the industry.

3.2.7 Recommendations

The Error System Using Taps ERSUT is a generic solution because it does not depend on the functionality of the combinational stages. Unlike Correction Function, its complexity does not increase if the complexity of the combinational logic stages increases. However, the results of ERSUT lead to some recommendations which are listed as follows:

- The new proposed approach should reduce the complexity of the error detection and valid data selection circuits to increase the tolerated PVT induced delay variations.

- The new proposed approach should reduce the overhead of area and power in comparison to the tolerated induced delays.

- The new proposed microarchitecture should solve the meta-stability issue.

- The new microarchitecture should be able to save power by using Dynamic Voltage Scaling DVS.

3.3 ERSUT-Based Resilient Microarchitecture

ERSUT approach was presented to detect the induced errors and recover the faulty stages without flushing the pipeline. ERSUT eliminates the safety margins and scale up the frequency at the typical supply voltage if there are no PVT variations. However, ERSUT has a relatively high power and area overhead in comparison to the tolerated induced delay which limits its adoption for more complex designs. The error-detection circuit of ERSUT adds a huge delay overhead which decreases the flexibility of this approach to tolerate larger PVT induced delays. ERSUT also suffers from meta-stability issues which threatens the reliability of the whole approach.

The current target is to develop an ERSUT-based resilient microarchitecture that eliminates the safety margins of the timing constraints without affecting its reliability. This new microarchitecture tolerates the induced delays, caused by PVT variations, without flushing and restarting the pipeline. If there are no PVT variations, the proposed ERSUT-based microarchitecture uses the hardware overhead to introduce a tradeoff between lower power for the same throughput and higher throughput at the cost of power. The error recovery system supports the microarchitecture to scale down the voltage without reducing the operating frequency and tolerates the induced errors without pipeline flushing.

In comparison to ERSUT approach, the new proposed microarchitecture reduces the complexity of the error detection circuit to permit the approach tolerating larger PVT induced delays. The new microarchitecture solves the meta-stability issue of the ERSUT and the Razor-based approaches which increases the reliability of this resilient microarchitecture. The area overhead is also decreased in comparison to the tolerated induced delay to make the new approach applicable for more complex designs. While ERSUT consumes more power than the conventional designs, the proposed resilient microarchitecture is able to save power without decreasing the throughput.

3.3.1 Resilient Microarchitecture Overview

The main idea behind the proposed ERSUT-based resilient microarchitecture is to detect the generated errors at the output of the critical stages while it allows the delayed correct data to propagate through the pipeline. The induced delay caused by either PVT variations or DVS is compensated by borrowing the available slack of the successive non-critical stages.

Figure 3.13 shows the general overview of the proposed microarchitecture. In case of PVT variations or voltage scaling down, the induced delays of the critical stages are expected to violate the timing conditions of the pipeline. The critical stage fails to deliver its output data D_{out} in time. The shadow flip-flop captures the errant data D at the active edge of the clock signal while the main latch ML permits the delayed correct value D_d to propagate. Because the two copies are not identical, the error signal E is activated (active low). The correct data D_d reaches the non-critical stage NS1 after a certain delay, so that the pipeline needs to be flexible enough to compensate this delay. At the next clock cycle, E propagates through the flip-flop EF1 to control a special memory element which we called Configurable-Buffer-Flip-Flop CBF. When E1 is active low, CBF1 is transparent for a whole clock cycle to allow the delayed data to propagate through the successive non-critical stage NS2. At the next clock cycle and if there is no more errors, E1 is deactivated and CBF1 returns back to its normal operation as a flip-flop. Then, E2 is activated and CBF2 turns its operation from a normal flip-flop to a buffer to allow the delayed data from NS2 to propagate to NS3, etc. While the delayed data is propagating through the non-critical stages, the induced delay is compensated by borrowing the available slack of these non-critical stages. We assume that even if the non-critical stages examine induced delays due to voltage scaling or PVT variations, they still have enough slack to compensate the total induced delay through the pipeline. The number of CBFs and EFs are determined at design time to be sufficient for compensating the maximum expected induced delay. The Main Latch can be replaced by CBF to get a transparency window greater than a half clock cycle. A clock signal with more than 50 % duty cycle can be used as a potential alternative.

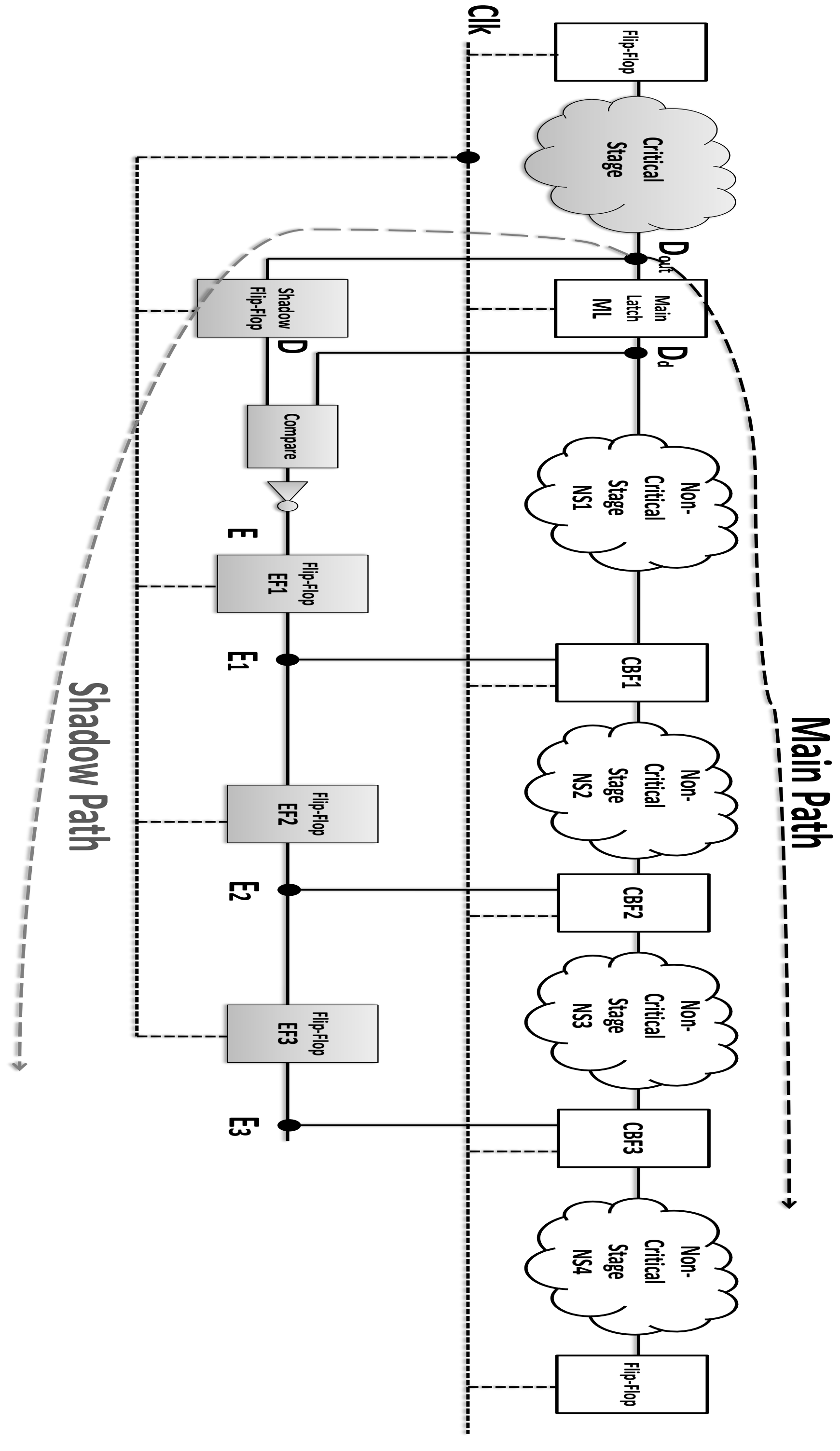
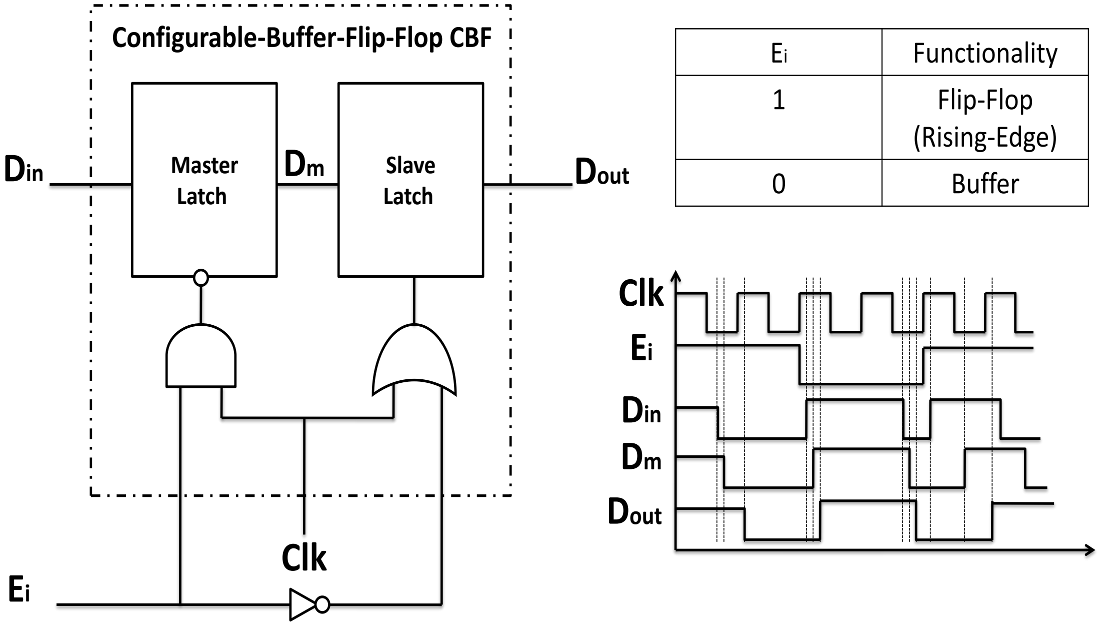


Figure 3.13: A general overview of the proposed self-healing resilient microarchitecture. The shadow flip-flop samples D_{out} at Clk_Edge and the main latch ML samples D_{out} from Clk_Edge to Clk_Edge + $T_{cp}/2$, where T_{cp} is the clock period © [39].



Ei	Functionality
1	Flip-Flop (Rising-Edge)
0	Buffer

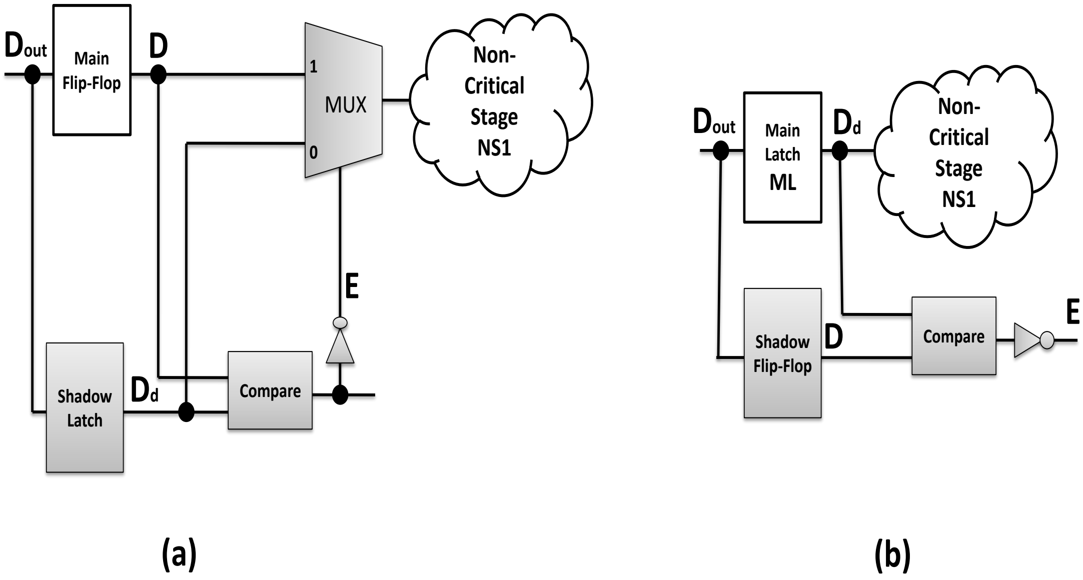
Figure 3.14: The architecture of the Configurable-Buffer-Flip-Flop CBF, including its functionality table and the timing diagram © [39].

Figure 3.14 shows the internal design of the CBF and its timing diagram. If the error signal Ei is active low, CBF is transparent and it acts as a buffer because the clocking signals of the master latch and the slave latch are always active regardless the value of the clock signal Clk. If Ei is high (deactivated), CBF acts as a normal flip-flop because the clocking signals of the master latch and the slave latch are dominated by Clk.

This design allows the non-critical stages controlled by CBFs (for example: NS1 and NS2 in Figure 3.13) to be combined together as a one stage for only one clock cycle (when Ei is activated) and then separated at the next clock cycle (When Ei is deactivated) to continue their normal operation till the next error. As the CBFs should be open for a whole clock cycle to compensate the induced delay, some timing constraints must be preserved to assure the robustness of the pipeline operation.

3.3.2 Error Detection Circuit

The error detection circuit is a main component of the proposed microarchitecture. Increasing the complexity of the error detection circuit has negative impacts on area and power overhead as shown by ERSUT. Figure 3.15.a shows the conventional error detection circuit used by ERSUT, in which the main flip-flop captures data in time while the shadow latch detects the delayed correct value. After that, the two values are compared and if they are not identical the error signal E is activated. When E is active, the multiplexer switches its output from D to D_d to allow the correct delayed value to propagate. Redirecting the correct data from the shadow latch to the multiplexer increases the total induced delay that should be tolerated. This means that more non-critical stages and CBFs are needed to tolerate the total induced delay (generated by error detection and data selection circuits in addition to PVT variations or DVS induced delays).



(a)

(b)

Figure 3.15: Error Detection © [39]: (a) The conventional circuit or ERSUT. (b) The proposed circuit.

The error detection circuit shown in Figure 3.15.b is adopted by the new microarchitecture which saves the incorrect in-time data by the shadow flip-flop while the delayed correct value propagates through the main latch. Because of using a latch instead of the normal flip-flop, the delayed correct data propagates directly to the next stage and it does not need to wait for the decision of the comparator. The delay overhead of the comparison and multiplexing is eliminated to avoid extra power and area overhead consumed by inserting MUXs and more CBFs.

3.3.3 Meta-stability Analysis

As mentioned before, the meta-stability of a signal is an unstable state in which the signal is not resolved at a certain logic level (neither 0 nor 1). The main cause of the meta-stability is the setup violations in which the data is changed late enough while the master latch is closing so it is not able to deliver the data correctly to the slave latch. As the meta-stability resolution is time independent, all better-than-worst-case designs suffer from meta-stability which limits the adoption of these designs by the industry [22]. The used error detection circuit, shown in Figure 3.15.b, solves the meta-stability issue by transferring it from the data path to the shadow path. If the meta-stability is resolved correctly and early enough, the error signals are not activated and the circuit continues its normal operation correctly because the data path does not experience the meta-stability. If the meta-stability is not resolved early enough, the error signal is activated, controlling the CBF to be transparent for a whole clock cycle while the pipeline does not need to borrow this slack. This false alert does not affect the functionality of the pipeline because the data continues propagating correctly through the pipeline stages. Unlike the other better-than-worst-case approaches, the meta-stability issue is controlled and its effect is limited not to cause a pipeline flushing or data corruption.

3.3.4 Timing Constraints

The timing constraints of the ERSUT-based resilient microarchitecture introduce more generic and realistic constraints than the mentioned in section 3.2.2. The new constraints discussed in this section is suitable for both PVT induced delay variations and Dynamic Voltage Scaling. Unlike ERSUT timing constraints, the induced delays are applied for all stages, including critical and non-critical stages. The effect of PVT variability or DVS is considered for all stages instead of the critical stages only.

Regarding the Main Latch ML, its short paths should be readjusted to prevent any hold violations due to its transparency window. In case of error detection, each CBF is open for a whole clock cycle and racing between the current data and its successive version leads to data damage and pipeline corruption. To avoid data racing through the transparent CBFs, timing constraints are preserved to assure that every version of data propagates safely without any potential damage. The design methodology is based on the Static Timing Analysis STA tool of Synopsys, by which all delays are extracted based on the used technology and the standard cells library. To avoid any data racing or unexpected timing violations, the design follows a conservative methodology to satisfy all timing constraints. For the hold time constraint, all delays should be extracted at the fast corner; while the positive slacks and the propagation delays should be estimated at the slow corner. The timing constraints of the proposed approach are classified into two groups: the first group is related to the minimum number of CBFs used to tolerate the expected induced delay while the second group concerns about re-adjusting the short paths controlled by CBFs to eliminate the data racing.

$$\Delta T = \Delta T_C + \sum_{i=1}^{N} \Delta T_i \qquad (3.18)$$

$$\Delta T_C < \min \{T_{CP}, S_{max}\} \qquad (3.19)$$

The number of CBFs is determined based on the total induced delay ΔT illustrated by (3.18), where ΔT_C is the induced delay of the critical stage, ΔT_i is the induced delay of the non-critical stage i, and N is the number of non-critical stages used to compensate ΔT. Inequality (3.19) shows the upper limit of ΔT_C, which is the minimum value of either the maximum available slack S_{max} or the clock period T_{CP}.

$$S_i = T_{CP} - [T_{mi} + \Delta T_i] \qquad (3.20)$$

$$S_{max} = \sum_{i=1}^{N} S_i = N*T_{CP} - \sum_{i=1}^{N}[T_{mi} + \Delta T_i] \qquad (3.21)$$

$$\Delta T_C < N*T_{CP} - \sum_{i=1}^{N}[T_{mi} + \Delta T_i] \qquad (3.22)$$

The slack S_i, available by the non-critical stage i, is calculated by (3.20) where T_{mi} is the maximum propagation delay of the non-critical stage i. The maximum slack S_{max} is the summation of the slacks offered by the non-critical stages, as illustrated by (3.21). Based on equation (3.21), the induced delays of the non-critical stages, due to PVT variations or power saving, are taken into consideration, so that S_{max} is the net available slack used to tolerate errors generated by the critical stage. From (3.19) and (3.21), the upper limit of ΔT_C is calculated through inequality (3.22) by which the minimum number of CBFs is determined at design time. Even for a pipeline with more balanced stages, the critical stages are still able to steal time from the next stages because every stage has many internal paths and most of them are non-critical. The critical path of a critical stage may be able to steal slack from the non-critical paths of the next critical stages. The probability of cascaded critical paths through all successive stages is very low and the realistic limitation of this approach is to be stuck with a self-looping critical path.

To assure the robustness of the pipeline operation while the CBFs are transparent, the short paths should be readjusted to eliminate the racing between the successive data. Figure 3.16 shows an example of a self-healing resilient pipeline that uses two CBFs to steal time from three non-critical stages. The short paths constraint prevents the racing between the delayed data propagating through the long paths (T_{m1}, T_{m2}, and T_{m3}) and its successive data propagating through the short paths (T_{SH1}, T_{SH2}, and T_{SH3}). This means that the data at the next clock cycles should be delayed while propagating through these short paths till the previous delayed data reaches CBF1, CBF2 and the output flip-flop safely.

For the worst case scenario, it is assumed that the induced delay is equal to the maximum slack. Inequality (3.23) shows that the short paths T_{SH1} after a new clock period T_{cp} should be greater than the long path Tm1 and the worst induced delay which is equal to S_{max}. Based on (3.23), inequalities from (3.24) to (3.26) can be extracted.

$$T_{SH1} + T_{CP} > [T_{m1} + \Delta T_1] + S_{max} \qquad (3.23)$$

$$T_{SH1} > 2T_{CP} - [T_{m2} + \Delta T_2] - [T_{m3} + \Delta T_3] \qquad (3.24)$$

$$T_{SH1} + T_{SH2} > 2T_{CP} - [T_{m3} + \Delta T_3] \qquad (3.25)$$

$$T_{SH1} + T_{SH2} + T_{SH3} > 2T_{CP} \qquad (3.26)$$

From (3.24), (3.25), and (3.26), a general inequality (3.27) is extracted to be applied for all short paths affected by CBFs where N is the number of non-critical stages used to offer slack.

$$\text{Where}\Big|_{n=1}^{N}\{$$

$$\sum_{i=1}^{n} T_{SHi} > (N\text{-}1)*TCP - \sum_{i=1}^{N}[T_{mi} + \Delta T_i] + \sum_{j=1}^{i}[T_{mj} + \Delta T_j]$$

$$\} \qquad (3.27)$$

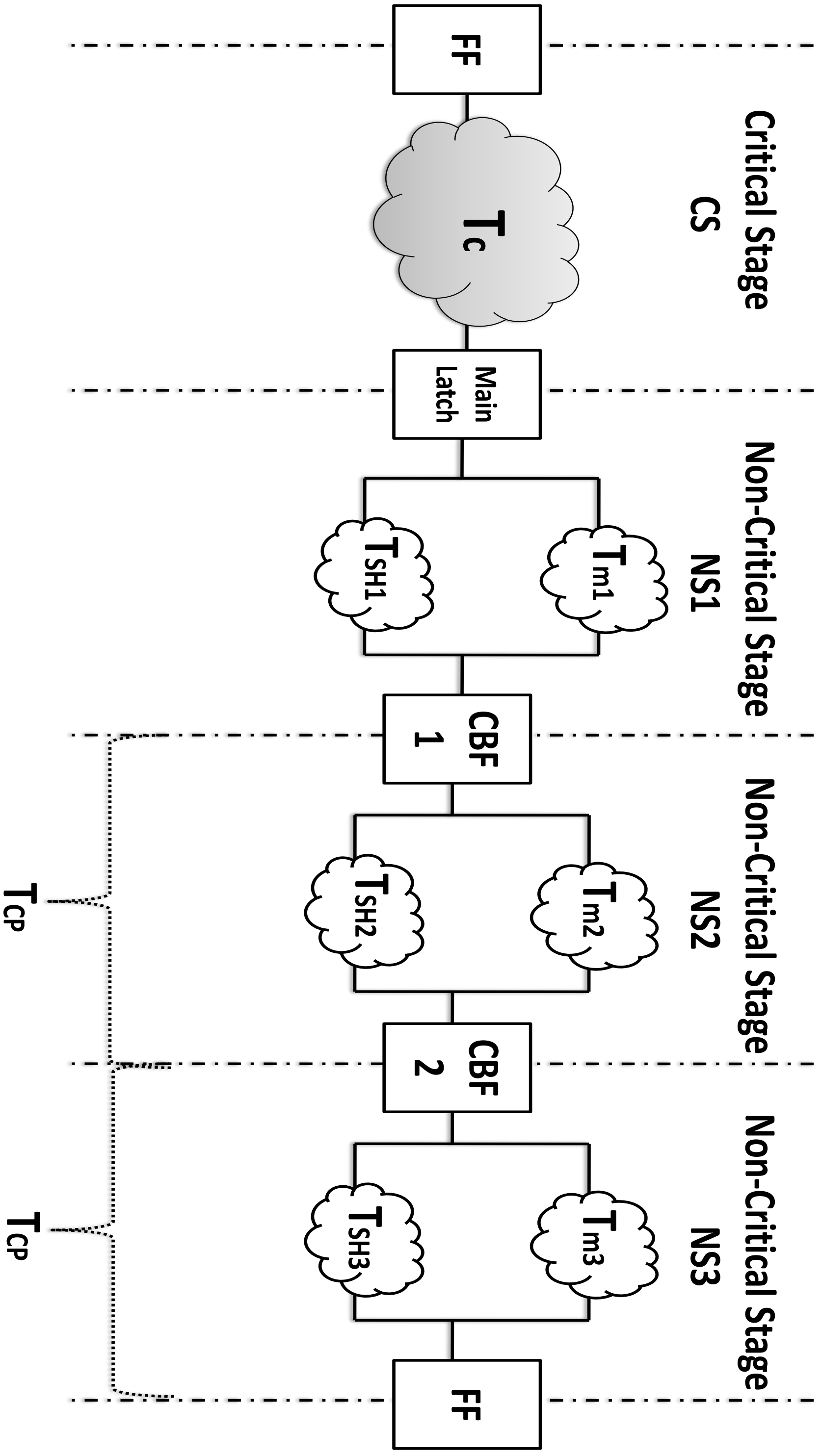


Figure 3.16: An example of a self-healing resilient microarchitecture with potential data racing through the short paths. Short paths (T_{SH1}, T_{SH2}, and T_{SH3}) must be readjusted not to race with the long paths (T_{m1}, T_{m2}, and T_{m3}) © [39].

The common technique for readjusting the short paths is to insert delay elements through these paths to satisfy the previous constraint. The added delay elements are the major reason to increase the area and power overhead of the proposed microarchitecture. Therefore, increasing the number of short paths, which need readjustment, increases the power and area overhead to be a critical concern. This concern was taken into consideration while choosing the test case circuit to represent a worst case scenario

3.3.5 Case Study, 16x16 bit MAC Unit

In case of the 16x16 bit MAC unit, the critical stages do not need the registers of their successive stages to be transparent more than 50% of the clock period. Thus, the ERSUT-based resilient microarchitecture was modified to make the registers (CBFs) of the non-critical stages transparent for only half clock cycle. Based on this special case, a light microarchitecture is introduced in Figure 3.17 to be used for the MAC unit. This light version decreases the overall costs of the proposed ERSUT-based resilient microarchitecture. The only difference between the full microarchitecture shown in Figure 3.13 and the light microarchitecture in Figure 3.17 is replacing the Configurable-Buffer-Flip-Flops CBFs by the Configurable-Latch-Flip-Flops CLFs. In case of error detection, CLFs act as latches and they are transparent for only half clock cycle. When the error signal is deactivated, CLFs return back to the normal operation as flip-flops.

Instead of combining the two adjacent stages for a whole clock cycle, the CLF is used to combine them for only half clock cycle. This reduces the complexity of the short path constraints and also reduces the total overhead of the short paths readjusting in the case of MAC unit. On the other hand the generic version of the ERSUT-based resilient microarchitecture is still valid to be implemented for extreme cases in which the critical stages (or the cascaded critical stages) need to borrow more slack and the total induced delays need a whole clock cycle to be recovered.

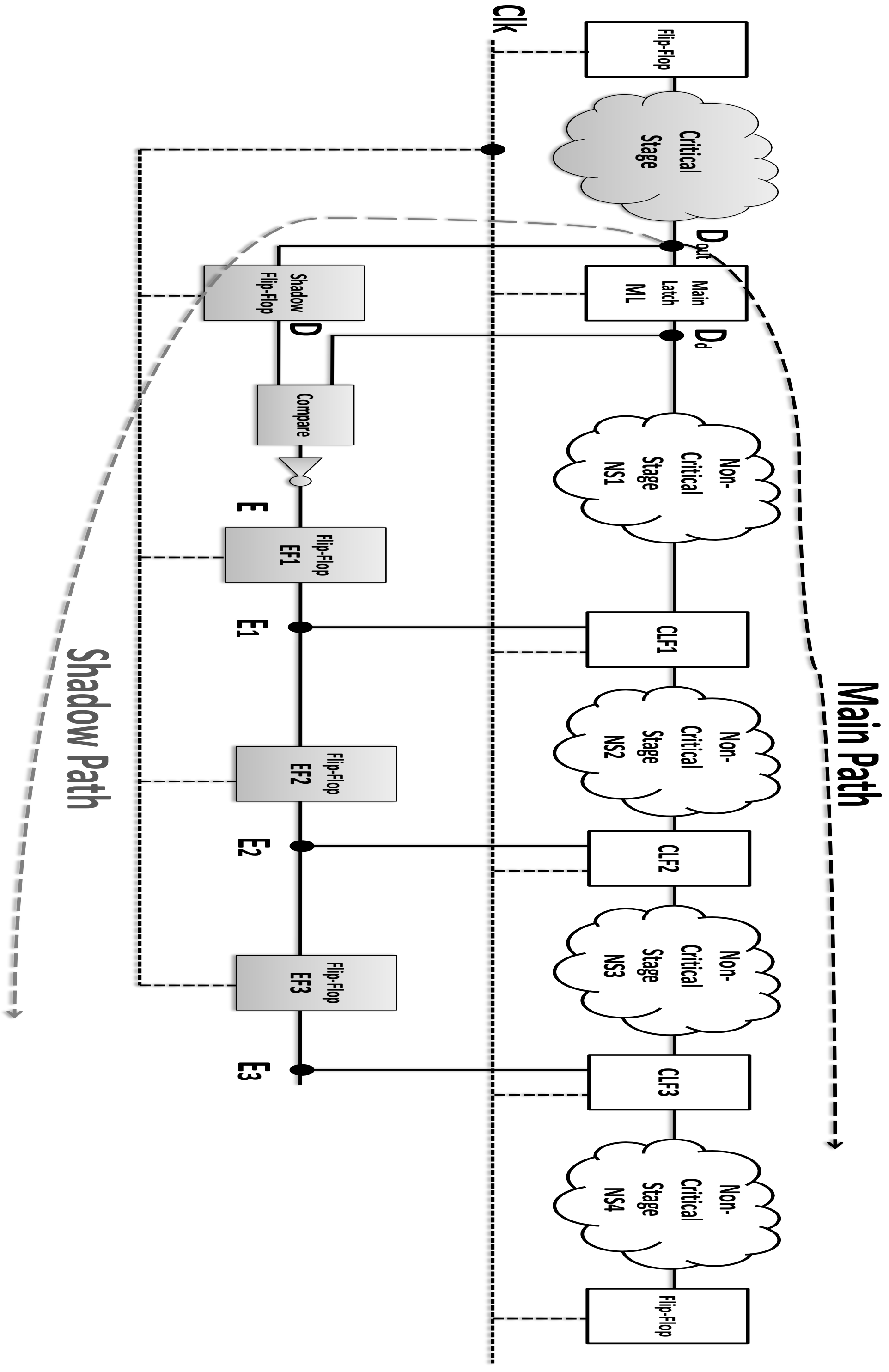
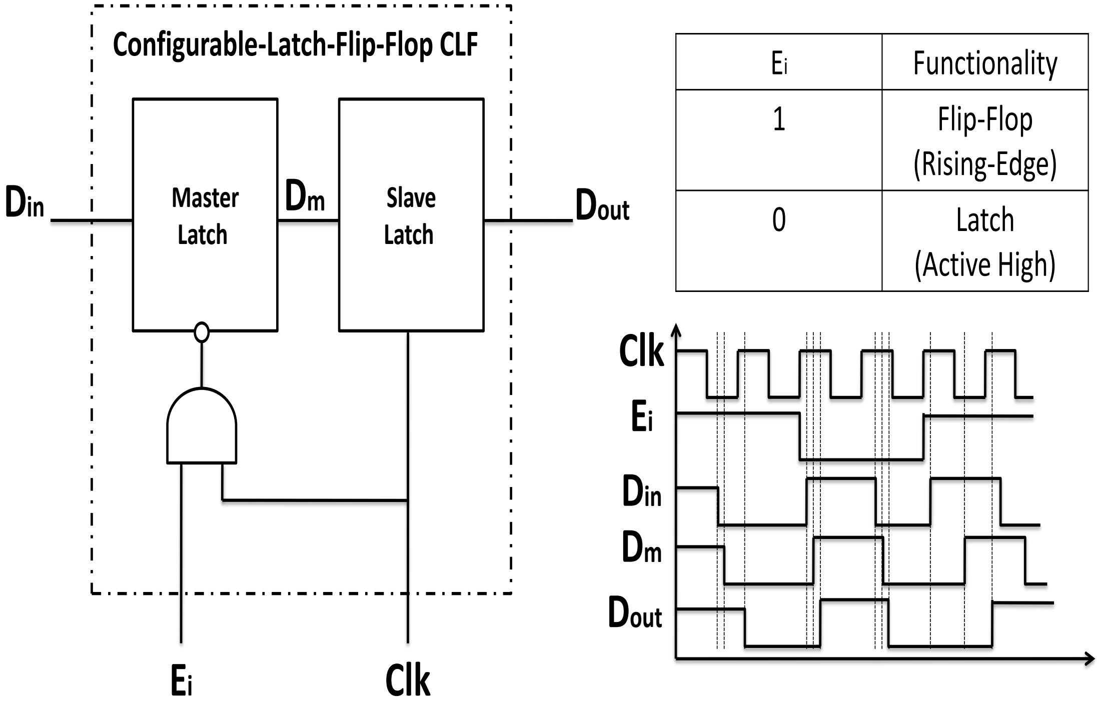


Figure 3.17: The light ERSUT-based resilient microarchitecture. The shadow flip-flop samples D_{out} at Clk_Edge and the main latch ML samples D_{out} from Clk_Edge to Clk_Edge + $T_{cp}/2$, where T_{cp} is the clock period © [39].



Ei	Functionality
1	Flip-Flop (Rising-Edge)
0	Latch (Active High)

Figure 3.18: The architecture of the Configurable-Latch-Flip-Flop CLF, including its functionality table and the timing diagram © [39].

Figure 3.18 shows the internal architecture, the functionality table, and the timing diagram of the Configurable-Latch-Flip-Flop CLF. It consists of a master latch (negative) and a slave latch (positive) while the clock of the master latch is gated by the error signal Ei. If Ei is activated (low), the clock controlling the master latch is always low and the master latch is transparent for the whole clock cycle. It allows the CLF to transfer data while the clock signal is high (acting as a latch). If Ei is not activated, CLF acts as a normal flip-flop transferring data only at the rising edge of the clock signal.

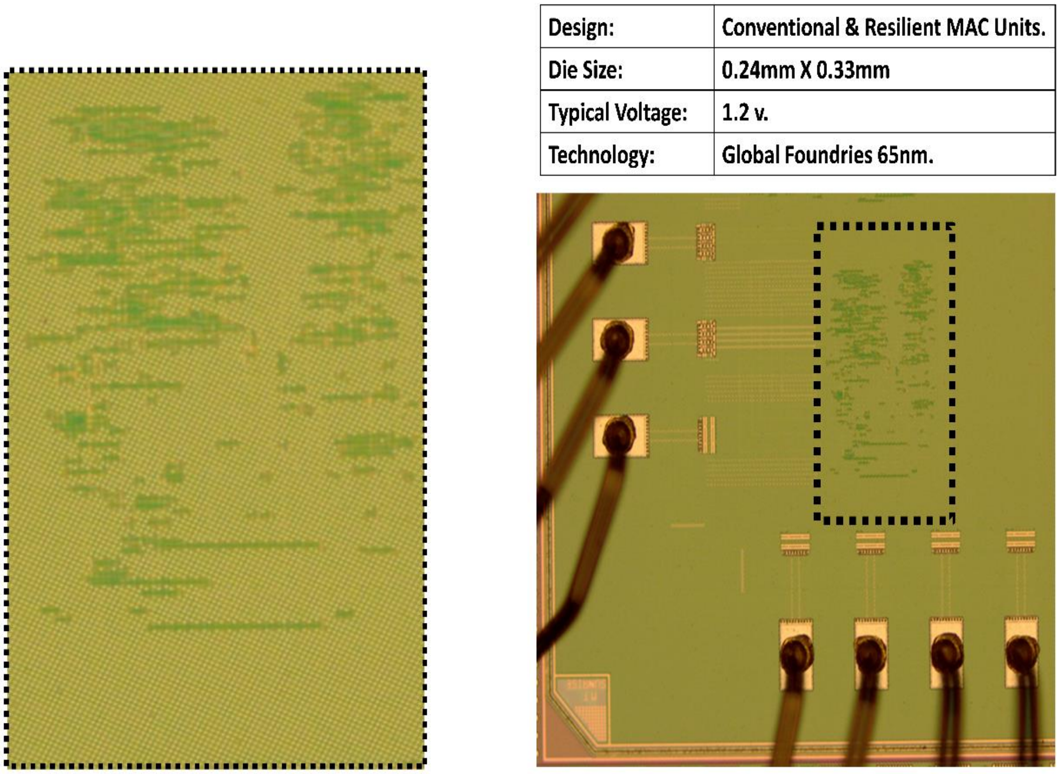
3.3.6 Experimental Results

The test chip was designed in Verilog and implemented using Global Foundries 65 nm technology and ARM standard cells library. It consists of a conventional MAC unit and a resilient MAC unit with the same specifications. The architecture of the two MAC units was designed as binary array and Synopsys Design Vision was forced not to touch the RTL design hierarchy to keep the two MAC units identical while IC Compiler was used for the place and route. Both conventional and resilient MAC units have nearly balanced nine pipeline stages. For the two MAC units, their critical and non-critical stages have the same delays. Although the first five stages are critical, each critical stage has many internal non-critical paths which afford enough slack for the critical paths in the other stages. This means that the critical paths in one critical stage borrow time slack from the non-critical paths of the next critical stage even these stages are nearly balanced.

Table 3.3. Results of the Conventional MAC unit and the ERUST-based resilient MAC unit including area and power, at the typical supply voltage 1.2 V, and the critical frequency 375 MHz.

MAC Unit	Frequency = 375 MHz, Voltage = 1.2 V.		
	Conventional	**ERSUT**	**Overhead**
Area (um^2)	15754.680	18632.160	18.26%
Power (mW)	2.740	2.796	2.055%

The critical frequency for the conventional design is about 375 MHz at the typical voltage 1.2 V. The new resilient microarchitecture has about 18.26% area overhead and 2.05% power overhead at the typical voltage as shown by Table 3.3. This area overhead is mostly due to the delay elements used for the short paths readjustment. This overhead is used to tolerate PVT induced delays or to save power by scaling down the supply voltage.



Design:	Conventional & Resilient MAC Units.
Die Size:	0.24mm X 0.33mm
Typical Voltage:	1.2 v.
Technology:	Global Foundries 65nm.

Figure 3.19: The micrograph of the manufactured chip including both the conventional and the resilient MAC units © [39].

The micrograph of the chip including both the conventional and the resilient MAC units is shown in Figure 3.19. The chip, shown by the micrograph, also includes the testing hardware circuitry which is based on the scan chain registers. The scan-based testing is one of the most common techniques of the Design For Testability DFT which increases the controllability and observability of the targeted design.

In case of PVT variations, the resilient MAC unit tolerates the induced delays up to 50% of its clock period without flushing the pipeline while the conventional approach has to reduce its throughput to operate correctly. These tolerated induced delays represent the temperature changes, the voltage instability or the process variations. The resilient MAC unit is able to operate correctly for the conventional critical frequency 375 MHz while scaling the voltage down to 1.0 V instead of the typical supply voltage 1.2 V saving about 29% of the power consumed by the conventional microarchitecture. This voltage degradation from 1.2 V to 1.0 V is considered as an extreme case of voltage variations.

Table 3.4. The operating frequencies of the conventional and ERSUT-based resilient MAC units at different supply voltages.

Voltage (V)	**Frequency (MHz)**	
	Conventional MAC Unit	**Resilient MAC Unit**
1.2	375	565
1.14	335	505
1.09	295	445
1.03	255	385
1.0	235	355
0.97	215	320
0.91	175	265
0.86	135	205
0.8	100	150

Table 3.4 shows the operating frequencies for both the conventional and the ERSUT-based resilient microarchitectures for different supply voltages. At the typical voltage 1.2 V, the conventional MAC unit operates at 375 MHz while the resilient MAC unit operates at 565 MHz. A safety margin of 10% was examined and selected for both the conventional and resilient MAC units to avoid any errors generated by clock skewing or

jitters while the supply voltage is scaled down. This safety margin dedicates the variation problem only to the voltage variations, and the same safety margin is applied to the two MAC units to get a fair comparison.

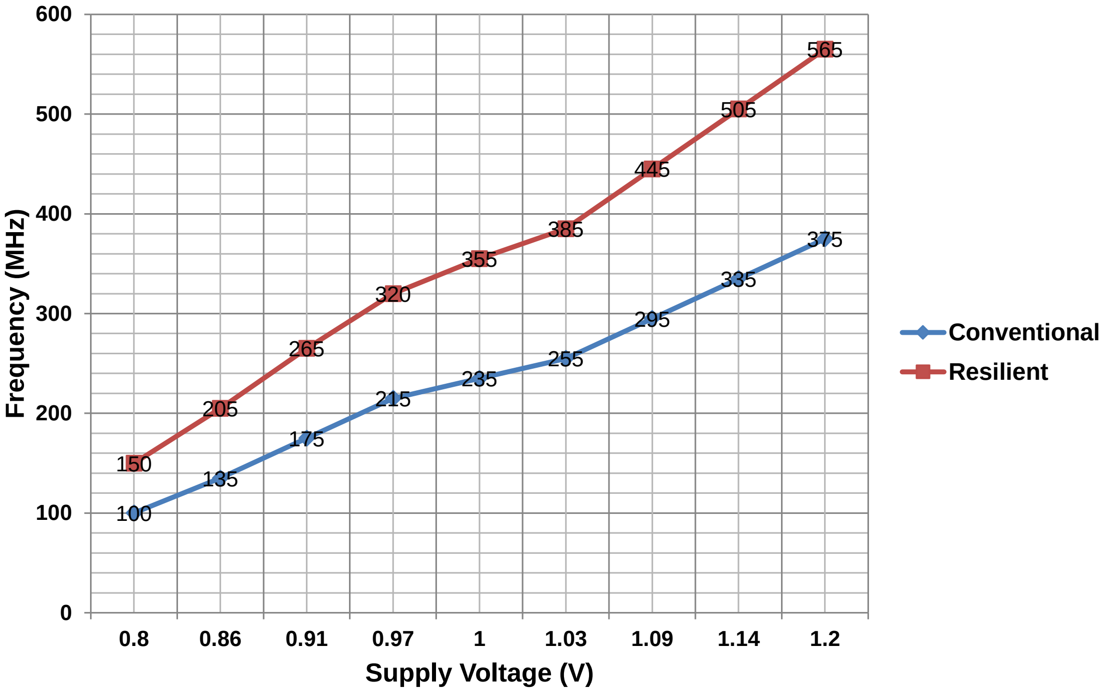


Figure 3.20: The operating frequency versus the supply voltage for the conventional MAC unit and the ERSUT-based resilient MAC unit © [39].

Figure 3.20 shows the operating frequencies at different supply voltages for the conventional and the resilient MAC units. These results show that the ERSUT-based resilient microarchitecture increases the throughput for the same operating conditions in comparison to the conventional design. For the same supply voltage, the resilient approach is able to operate correctly at higher frequencies by violating the timing conditions and tolerating the generated errors using the internal error recovery system. Increasing the operating frequency (which increases the throughput gain) increases also the total power consumption because the power is proportional to the operating frequency. The power figures of the ERSUT-based resilient microarchitecture is discussed in detail by the next chapter.

3.3.7 Conclusion

The new ERSUT-based resilient microarchitecture is a promising generic solution to tolerate the PVT induced delays and also to save the power consumption. The proposed resilient microarchitecture succeeded to tolerate the induced PVT variations or increase the throughput according to the priority of the digital circuit. Unlike other approaches, the new ERSUT-based microarchitecture does not recover the generated errors at the cost of throughput by scaling down the frequency or flushing the pipeline. Unlike the previous ERSUT approach, the error detection circuit is simplified and it does not add any overhead to the total induced delay. The meta-stability issue is solved by the new resilient microarchitecture which keeps the maximum throughput and increases the reliability of the healed design. The power and area overhead is reduced while larger induced delay is tolerated in comparison to ERSUT. The resilient MAC unit tolerates PVT induced variations up to 50% of its clock period with 18.26 % area overhead and 2.05% power overhead at the typical supply voltage and the conventional critical frequency.

For deeper pipelines, there is a better chance for this approach to borrow more slack from the successive stages. The overhead of the proposed approach is expected to be reduced as the number of non-critical stages is expected to be increased in comparison to the critical stages. In case of the architectural branching (non-linear pipelining), our approach calculates the maximum tolerated induced delay based on the minimum available slack in all branches. Looping is a special case of branching and our approach might fail only when a critical path is looping on itself and there is no available slack to be borrowed.

3.3.8 Recommendations

The ERSUT-based resilient microarchitecture needs more elaboration to investigate its potential benefits. Integrating Dynamic Voltage Scaling DVS with the ERSUT-based approach is an important step to investigate the power gain figures of the healed integrated circuits. In another direction, the process of applying this new approach needs to be automated to reduce the time and efforts of the design phase. Automating the design phase allows the ERSUT-based approach to be implemented on different complex digital circuits. A validation study is also required to show the probability of applying the ERSUT-based resilient microarchitecture on different digital circuits with different complexities. Some recommendations are listed as follows:

- Applying DVS for the ERSUT-based resilient MAC unit to investigate the potential power saving could be achieved by using the two approaches.

- Design an ERSUT-based CAD algorithm to facilitate the automation of ERSUT-based resilient approach.

- Apply the ERSUT-based CAD algorithm on different circuits of the ISCAS’89 benchmark to setup a validation study of the ERSUT-based resilient microarchitecture.

Chapter 4: Power Efficient Design

This chapter is concerned about developing power efficient integrated circuit designs by integrating different microarchitecture level techniques. For the runtime power management, the digital integrated circuit should be able to configure its power consumption according to its performance priority. On the other hand, the design-time power saving is targeting to design integrated circuits with low power budget for the maximum performance requirements. In section 4.1, the proposed runtime power management technique integrates the ERSUT-based microarchitecture and the Dynamic Voltage Scaling DVS technique to reduce the power consumption without reducing the throughput. Section 4.2 introduces the Double-Edge-Triggered microarchitecture as a promising solution to develop low power designs.

4.1 ERSUT-Based Microarchitecture and DVS

As previously mentioned, the Dynamic Voltage Scaling is one of the most common techniques used to manage the power consumption of the integrated circuits at runtime. When the power budget is the top priority, the digital circuit has to scale down its supply voltage to reduce the power consumption. As the supply voltage decreases, the delay of the logic gates and registers increases which leads to time violations and the corrupted data propagates through the pipeline stages of the synchronous sequential circuits. To avoid error generation, the digital integrated circuit has to scale down its operating frequency and reduce its throughput. Although the DVS is a common effective technique, it introduces a tradeoff between the throughput and the power consumption because reducing power consumption comes at the cost of the throughput.

4.1.1 Power vs. Throughput

Better-than-worst-case designs, like Razor-based approaches, try to increase the benefits of the DVS by using their internal error detection circuits. Razor-based approaches scale down the supply voltage without decreasing the operating frequency while monitoring the error rate. In case of error generation, the error is recovered and the pipeline is restarted to continue its operation. If the total power consumed by the error recovery becomes greater than the total power saving, the supply voltage is scaled up to reduce the error rate. This adaptive tuning technique has a main challenge which is the error recovery cost.

Figure 4.1 shows the tradeoff among the error recovery power, the power saving and the throughput. For the Razor-based approaches, scaling down the supply voltage reduces the operational power of the digital integrated circuit while the error rate is increased and the power consumed to recover these errors is also increased. Beyond the optimal point, the power consumed to recover the errors is greater than the power saving and the throughput of the digital integrated circuit is reduced dramatically. The cost of power recovery is mainly due to the pipeline flushing and restarting. Flushing the pipeline, or even halting it for a clock cycle, per each error reduces the throughput dramatically after a certain point.

Reducing the overhead of the error recovery power allows the digital circuit to scale down its supply voltage to lower levels and to save more power by shifting the optimal point as shown by Figure 4.1. Unlike Razor-based approaches, the ERSUT-based resilient microarchitecture does not flush and restart the faulty pipeline, so the chance to save more power is existing. Reducing the power consumption of the error recovery system used by ERSUT allows the new resilient microarchitecture to scale down the supply voltage to lower levels, while recovering the errors (during the normal operation) does not decrease the throughput.

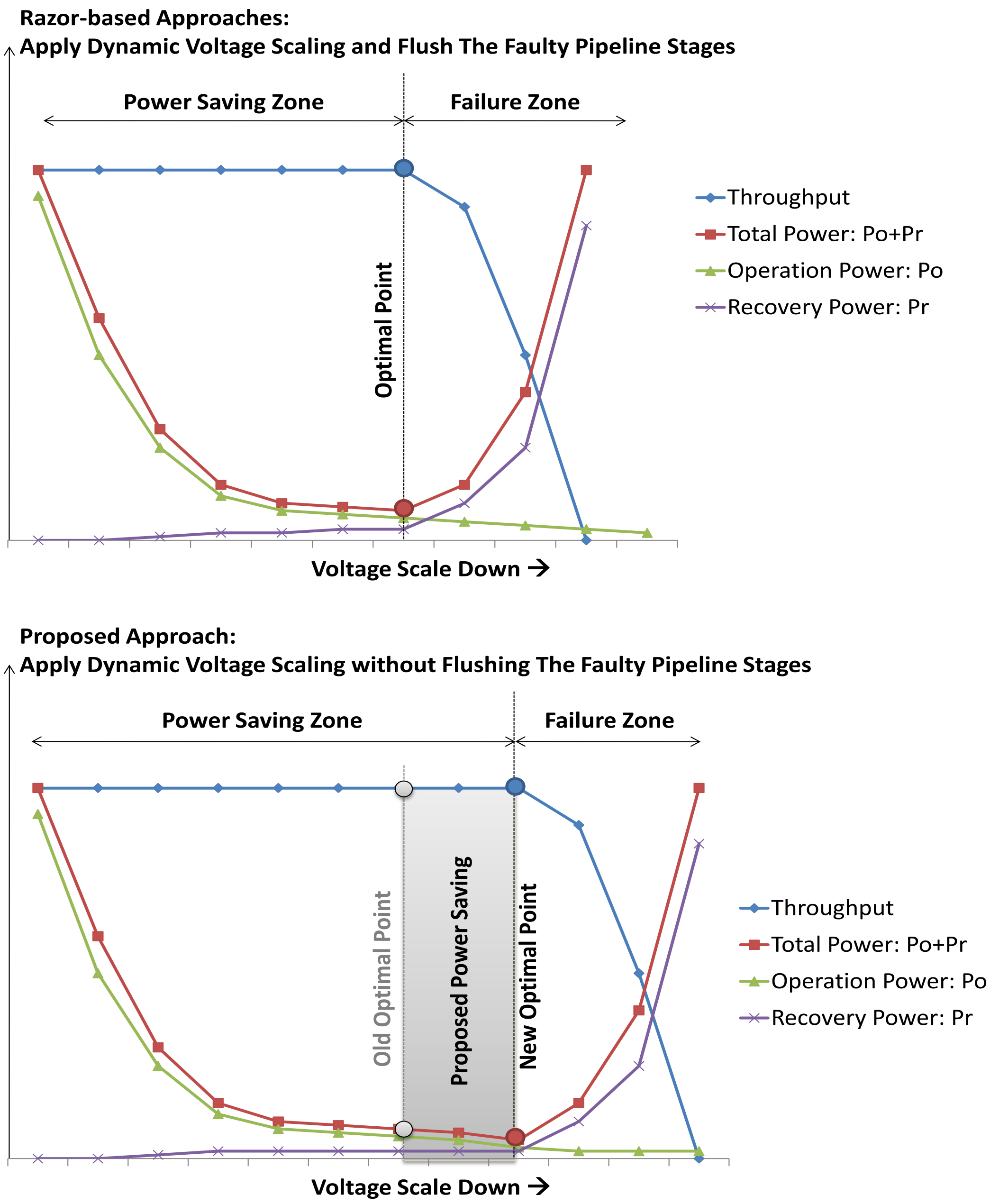


Figure 4.1: The tradeoff between power saving and throughput for the Razor-based approaches and the proposed approach depending on the error recovery power. Reducing the error recovery power, due to recovery without pipeline flushing, shift the optimal point to save more power © [39].

4.1.2 Low Power/ Same Throughput

The ERSUT-based resilient microarchitecture is able to tolerate the PVT induced delays and continue the normal operation of the pipeline without reducing the throughput. Results, shown in Section 3.3.6, illustrates that the resilient microarchitecture is able to increase its throughput at the cost of power if there are no PVT variations. The voltage scaling is considered to be an extreme case of voltage variation which is intentionally applied to reduce the power consumption. The induced delays generated by voltage scaling are detected by the error recovery system of the resilient microarchitecture and then tolerated by borrowing slacks from the successive stages. The toleration ratio of the induced delays is determined at design time according to the ERSUT-based microarchitecture design. Within this toleration ratio, the resilient design is able to scale down its supply voltage without decreasing its throughput.

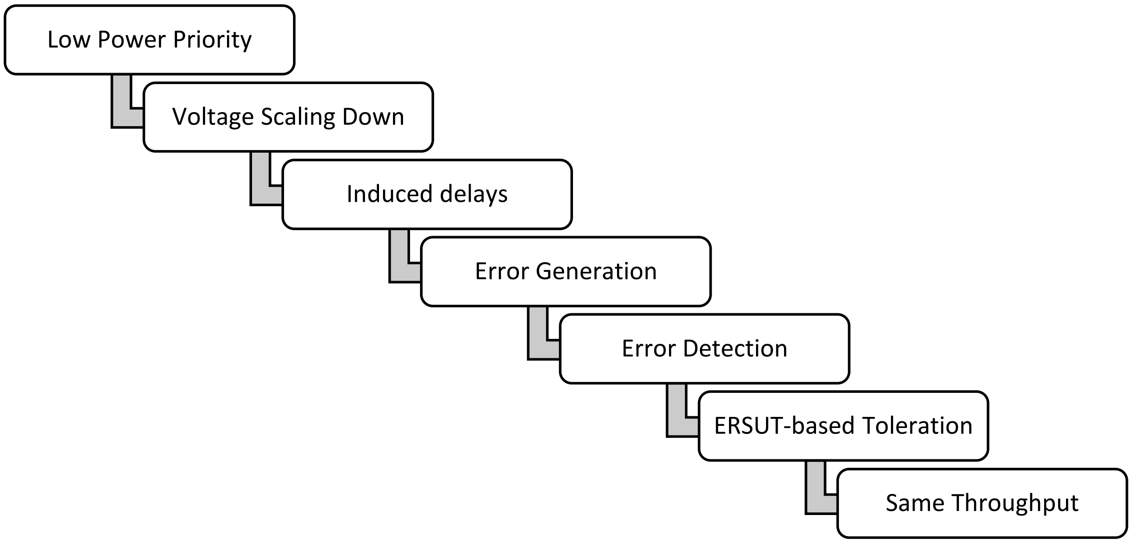


Figure 4.2: Low power/ same throughput flow using the voltage scaling and the ERSUT-based resilient microarchitecture.

Figure 4.2 shows the flow of keeping the throughput while reducing the power consumption using the DVS and the ERSUT-based resilient microarchitecture. The integration of these two techniques offers a new relation between the throughput and the power consumption which is totally different from the old tradeoff.

4.1.3 Case Study, 16x16 bit MAC Unit

The same case study of the conventional and resilient MAC units, shown in Section 3.3.5, is used to investigate the potential power benefits of applying DVS. As previously mentioned, the 16x16 bit MAC unit borrows up to 50% of its clock cycle. Consequently, the light ERSUT-based microarchitecture is applied to the MAC unit and the Configurable-Latch-Flip-Flops CLFs are used instead of the Configurable-Buffer-Flip-Flops CBFs to reduce the overhead. All critical paths, which are expected to violate the timing constraints during the DVS, are treated with CLFs.

4.1.4 Experimental Results

The test chip, shown in Figure 3.19, was designed in Verilog and implemented using Global Foundries 65 nm technology and ARM standard cells library. It consists of the conventional and the resilient MAC units which have the same specifications and the same pipeline stages. The critical frequency for the conventional design is about 375 MHz at the typical voltage 1.2 V. The ERSUT-based resilient microarchitecture has about 18.26% area overhead and 2.05% power overhead at the typical voltage, and it tolerates up to 50 % of its clock period. This allows the ERSUT-based resilient MAC unit to adopt an extreme level of voltage scaling to save power in case of no PVT variations.

The supply voltage can be scaled extremely down to 1.0 V while the resilient MAC unit still operates correctly for the conventional critical frequency 375 MHz, saving about 29% of the power consumed by the conventional microarchitecture. While the proposed resilient microarchitecture increases the throughput for the typical voltage (1.2 V) at the cost of power consumption, it also reduces the power consumption at the conventional maximum throughput by decreasing the supply voltage without decreasing the operating frequency. These modes of operations can be used dynamically according to the runtime priority of the digital circuit.

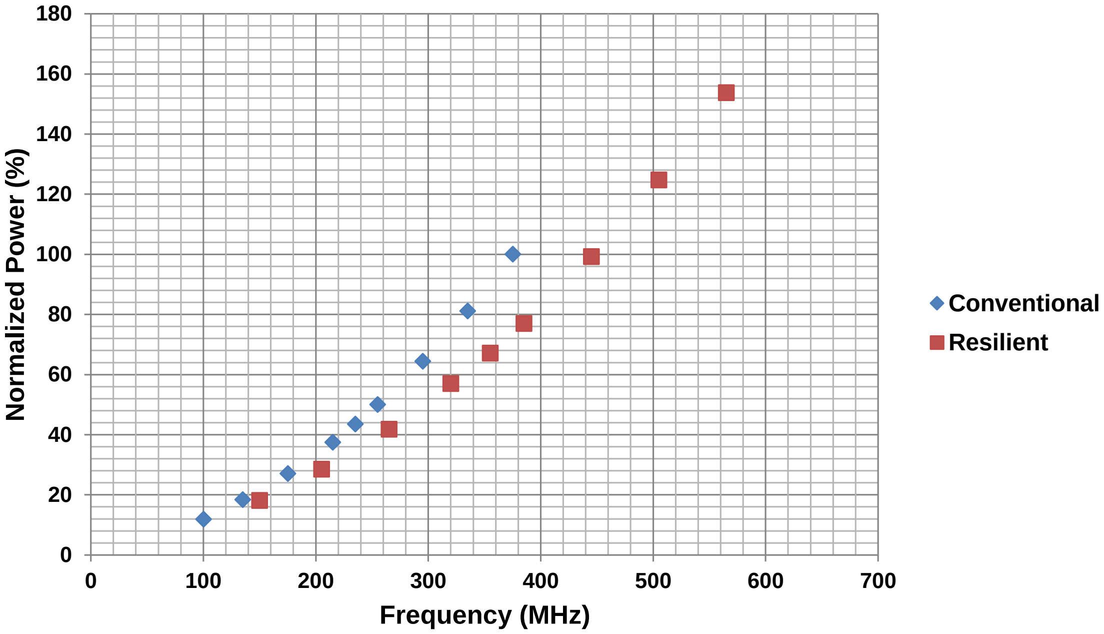


Figure 4.3: The normalized power consumption versus the operating frequency for the conventional MAC unit and the ERSUT-based resilient MAC unit © [39].

Figure 4.3 shows the normalized power consumption at different operating frequencies for the conventional and the resilient MAC units. These results show that the resilient MAC unit consumes less power than the conventional for the same operating frequency. From another point of view, the same amount of power used by the conventional microarchitecture to operate at 375 MHz, is used by the ERSUT-based resilient microarchitecture to operate nearly at 445 MHz. These results show the improvement of the power efficiency in case of using the ERSUT-based resilient microarchitecture.

Table 4.1 shows the normalized power figures for the conventional and the resilient MAC units and their operating frequencies for different supply voltages. At the typical voltage 1.2 V and the conventional maximum frequency 375 MHz, the power of the conventional MAC unit is considered to be the reference power consumption. At 1.09 V, the resilient MAC unit consumes about 99% of this reference power to operate for a frequency of 445 MHz which is greater than the conventional maximum frequency.

Table 4.1. The normalized power figures of the conventional and ERSUT-based resilient MAC units and their operating frequencies at different supply voltages.

Voltage (V)	**Conventional MAC Unit**		**Resilient MAC Unit**	
	Power (Normalized)	**Frequency (MHz)**	**Power (Normalized)**	**Frequency (MHz)**
1.2	1x	375	1.537x	565
1.14	0.811x	335	1.247x	505
1.09	0.644x	295	0.992x	445
1.03	0.500x	255	0.770x	385
1.0	0.435x	235	0.671x	355
0.97	0.375x	215	0.570x	320
0.91	0.271x	175	0.418x	265
0.86	0.184x	135	0.285x	205
0.8	0.119x	100	0.181x	150

4.1.5 Conclusion

Integrating ERSUT-based resilient microarchitecture with DVS approach is a promising technique to save power without losing the maximum throughput. The induced delays generated by the DVS are tolerated by the ERSUT-based resilient microarchitecture while the recovery process does not force the faulty pipeline to be flushed and then restarted. The Resilient MAC unit has only 18.26 % area overhead but on the other hand it saves up to 29% of the conventional MAC unit power consumption for the same throughput. In comparison to Razor-based approaches, the ERSUT-based resilient approach scales down the supply voltage to extreme levels which are considered unrecovered failure zone by these approaches. Unlike ERSUT, the ERSUT-based resilient approach has less power and area overhead with larger toleration ratio which permits the targeted design to save power using the DVS.

4.2 Double-Edge-Triggered Microarchitecture

Another direction to get power efficient digital integrated circuits is to reduce the power budget of the maximum throughput at design-time by using low power design techniques. The double-edge-triggering concept, mentioned in Chapter 2, is a promising approach to reduce the power consumption of the synchronous sequential circuits. However, this technique has a huge area overhead that reaches about 70% of the total area [35]. In the literature, many papers mentioned the power reduction which is achieved by using the Double-Edge-Triggered-Flip-Flops DETFFs against the common flip-flops, while ignoring the area overhead figures. This huge area overhead is a real obstacle against the adoption of this technique by the large synchronous sequential circuits. In another direction, the double-edge-triggering concept is not supported by the ASIC cell-based tools and the on-the-shelf standard cells libraries. This means that using the double-edge-triggering for a large synchronous design needs a lot of work around to integrate the custom DETFF cells with the standard cells library into one large ASIC design.

The target of the proposed Double-Edge-Triggered microarchitecture is to introduce a low power resilient microarchitecture that is based on Double-Edge-Triggered Pulsed-Latches (DETPL). The proposed microarchitecture should solve the area overhead issue by using the hardware sharing concept. The transparency window of the pulsed-latches is useful to tolerate the PVT induced delay variations or other environmental setup variations by relaxing the timing constraints among the adjacent pipeline stages. To test the validation of this approach, the common ASIC cell-based flow should be used with the on-the-shelf standard cells library to apply the Double-Edge-Triggered microarchitecture for a wide variety of different synchronous circuits.

4.2.1 Microarchitecture Overview

Latches are considered to be superior to flip-flops in terms of area and power dissipation. However, the CAD tools do not support the latch-based circuits for the Static Timing Analysis (STA) due to the transparency window of the normal D-latches [40]. On the other hand, the pulsed-latches are used to act just like the soft-edge flip-flops and they are friendly to the STA tools [27]. These pulsed-latches quite fit with the low power designs because of its smaller size and lighter clock load [41]. For these reasons, the pulsed-latch is adopted by the proposed microarchitecture to be double-edge-triggered instead of the flip-flop. Using pulsed-latches does not only reduce the power and area overhead of the double-edge-triggered registers but also mitigates the PVT induced delay variations through the transparency window. This window permits the pulsed-latch to pass the correct data even if it comes a little bit late.

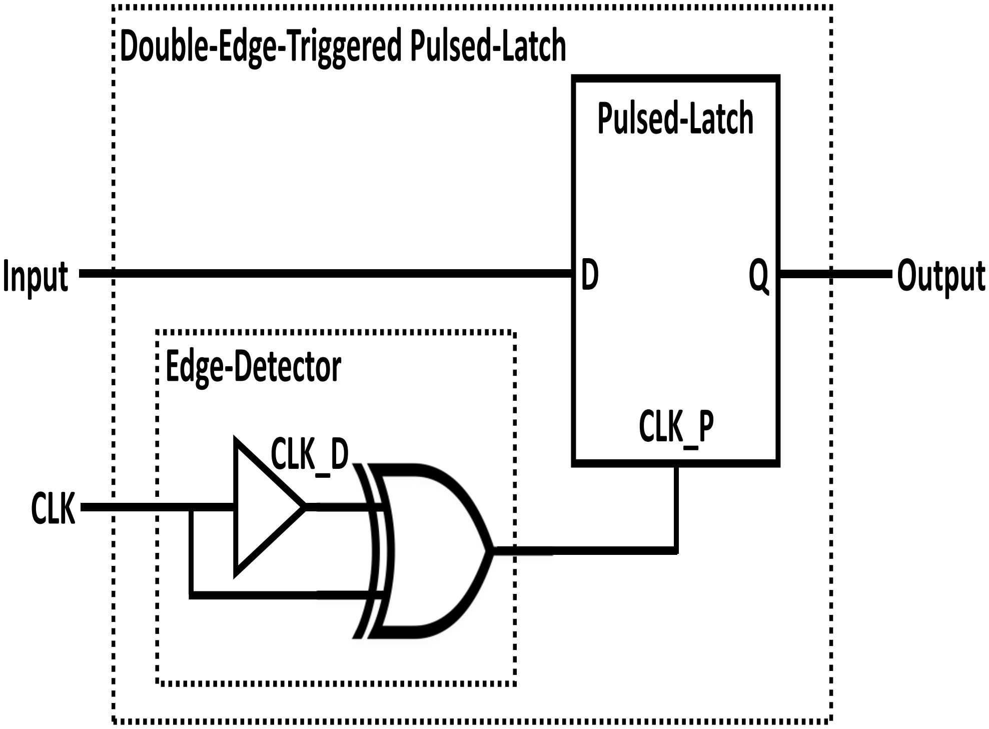


Figure 4.4: The Double-Edge-Triggered Pulsed-Latch DETPL using Edge-Detector circuit. The frequency of CLK signal is half the frequency of CLK_P signal.

Figure 4.4 shows the internal architecture of the Double-Edge-Triggered Pulsed-Latch (DETPL) which is used to replace the common flip-flop. The pulsed-latch is a normal d-latch which is transparent or open while the clock signal is high, so it passes its input data, then it is opaque or closed when the clock signal goes low. The pulse width of the clock signal is generated locally at the rising edge and the falling edge of the clock signal by using the edge-detector circuit. The clock signal CLK is fed to the clock tree at the half maximum frequency (F/2), then CLK reaches the Xor gate and the delay element of the edge-detector at the same time. A delayed version of the clock signal CLK_D is Xor-ed with the in-time signal CLK to generate the pulsed-latch clock signal CLK_P with the maximum frequency F, as illustrated by the timing diagram in Figure 4.5. By placing the edge-detector locally for the pulsed-latch, the proposed microarchitecture is able to locally multiply the frequency of the clock tree by two.

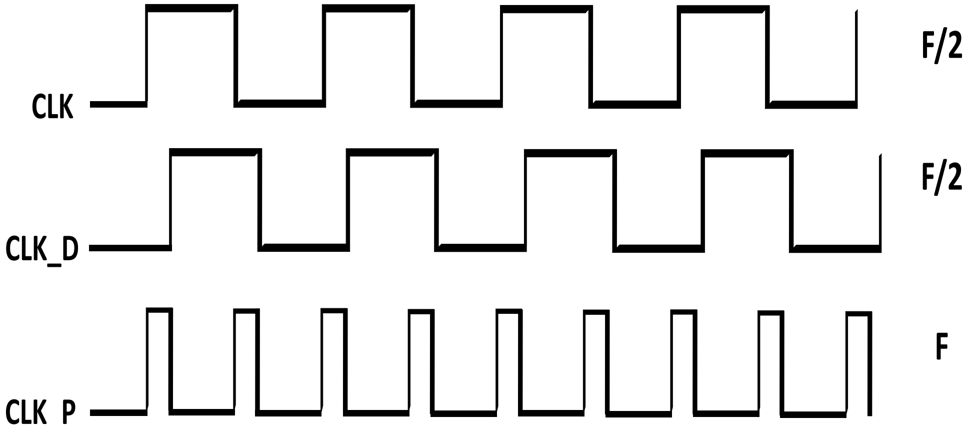


Figure 4.5: The timing diagram of the internal signals of the Edge-Detector circuit including the clock signal CLK, the delayed clock signal CLK_D, and the pulsed-latch clock signal CLK_P which equals the double frequency of the main clock signal CLK..

Figure 4.6 shows the double-edge-triggered microarchitecture using the double-edge-triggered pulsed-latches. As the pulsed-latches have a transparency window, the hold time constraint of the synchronous circuits is re-adjusted to be immune against any kind of data racing.

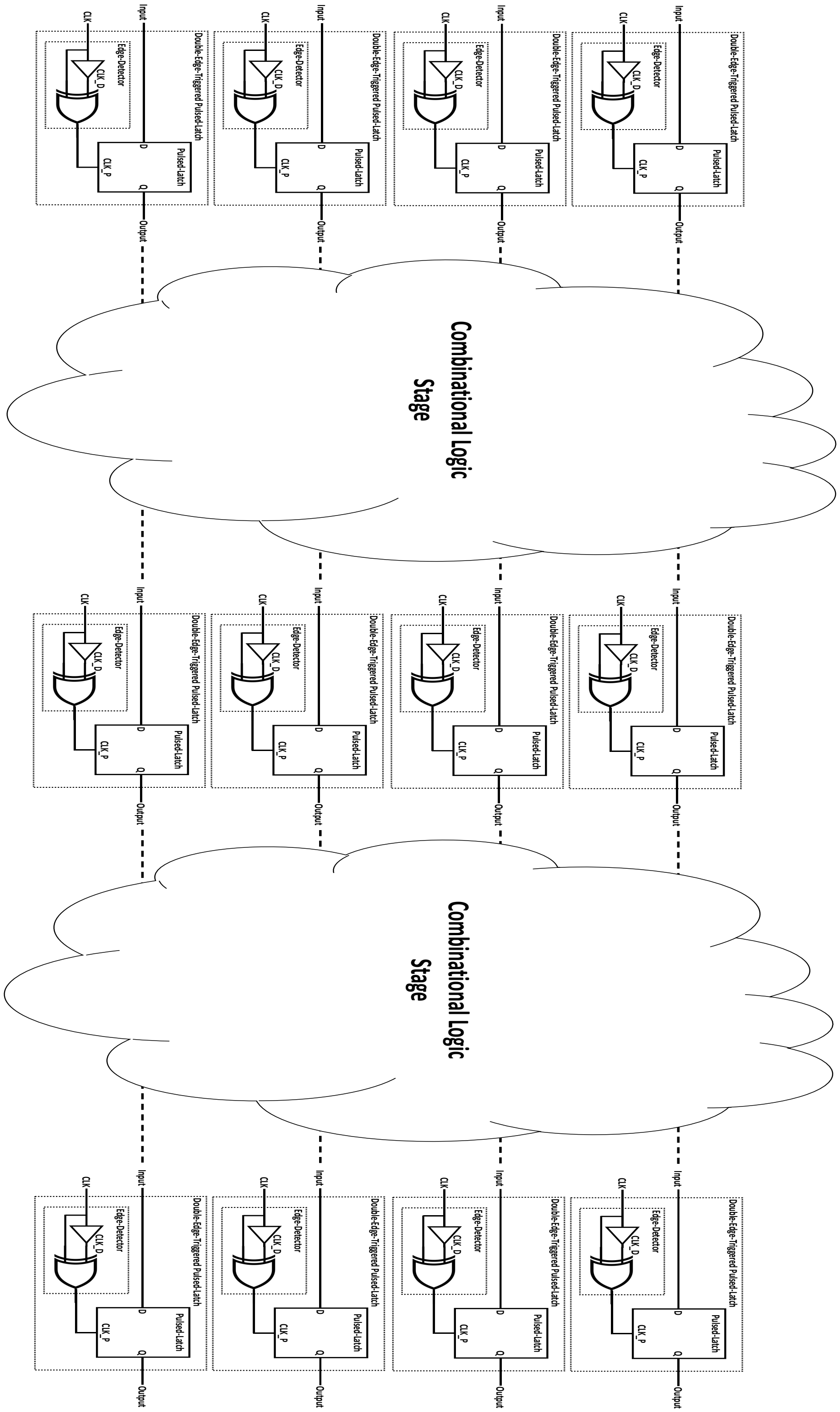

Double-Edge-Triggered Pulsed-Latch
Edge-Detector
CLK_D
Pulsed-Latch
D
Q
CLK_P
CLK
Input
Output
Combinational Logic Stage
Combinational Logic Stage

Figure 4.6: The double-edge-triggered microarchitecture using pulsed-latches and edge-detector circuits. The clock signal CLK propagates at frequency F/2 while the double-edge-triggered pulsed-latches operate at frequency F.

The short paths of all stages are extended to be greater than the pulse width of the clock signal as shown in inequality (4.1). This pulse width is determined at design time according to the used delay element. AS shown in equation (4.2), the pulse width covers both the minimum pulse width of the latch which is based on the used technology and the toleration window which is dedicated to mitigate the PVT variations. Increasing the transparency window increases the toleration and flexibility of the microarchitecture but on the other hand increases the area overhead due to the buffers should be used to re-adjust the short paths.

$$T_{SHi} > T_{PW} \qquad (4.1)$$

$$T_{PW} = T_{MPW} + \Delta T \qquad (4.2)$$

Formulas (4.1) and (4.2) are added to the common setup and hold time constraints, where T_{SHi} is the delay of the shortest path of stage i, T_{PW} is the pulse width of the clock signal, T_{MPW} is the minimum pulse width required by the latch to pass the data correctly and ΔT is the toleration window within which the data can be delayed and passed correctly.

4.2.2 Shared-Edge Detector (SED)

The double-edge-triggered microarchitecture reduces the power consumption of the clock tree by halving the frequency of the clock signal which propagates through the clock tree buffers. However, the huge area overhead is the main drawback of this proposed approach. The area overhead, which may exceed 70% of the total synchronous design area, is not acceptable for adoption by the industry. In this thesis, a Shared-Edge-Detector SED is presented as a solution not only to reduce the area overhead dramatically but also to reduce the total power consumption of the whole design. As shown in Figure 4.7, a single Edge-Detector is shared between only two adjacent pulsed-latches to reduce the area overhead. The Shared-Edge-Detector SED is placed spatially correlated to the adjacent pulsed-latches. Increasing the number of pulsed-latches that share the same SED creates sub-trees for the clock signal. The clock signal propagates through these sub-trees at the maximum frequency (F) not the half maximum frequency (F/2). This means that more power is consumed by the clock sub-trees and the gained power reduction may be vanished. Thus, only two adjacent pulsed-latches are permitted to share one SED to avoid any demand to create a sub-tree for the clock signal.

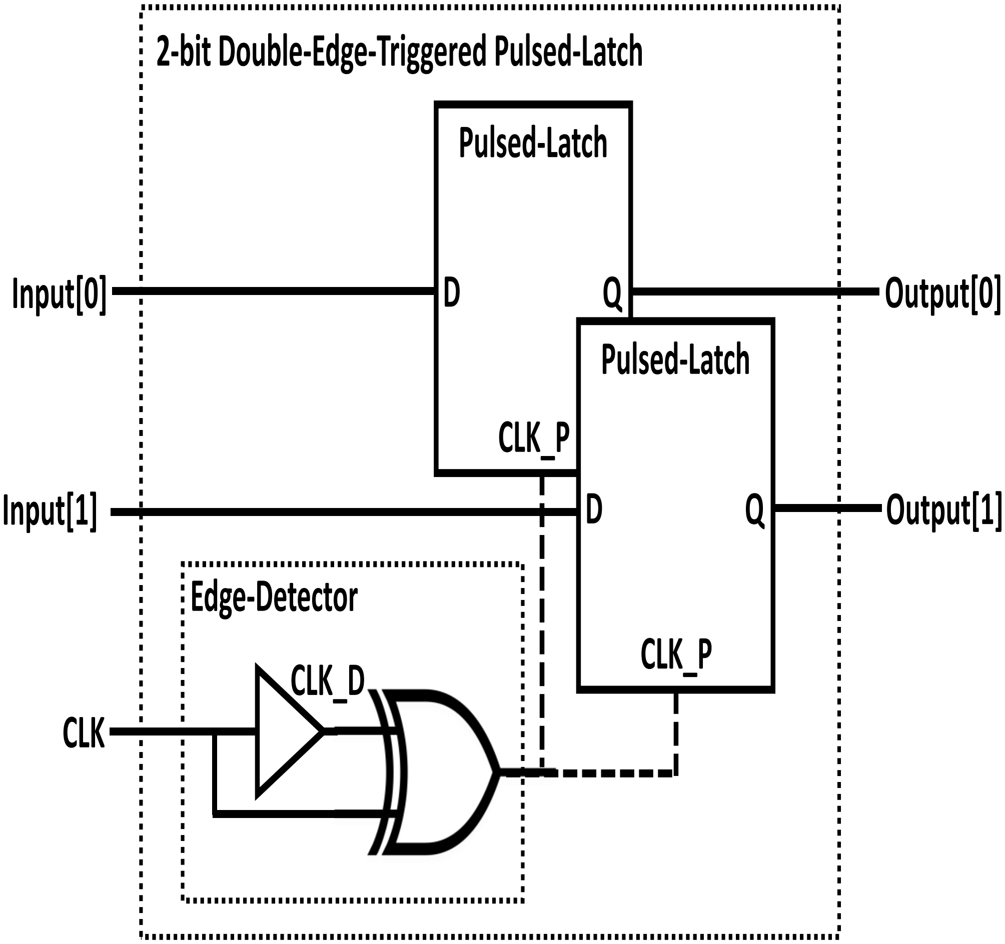


Figure 4.7: The 2-bit Double-Edge-Triggered Pulsed-Latch DETPL using Shared-Edge-Detector SED circuit

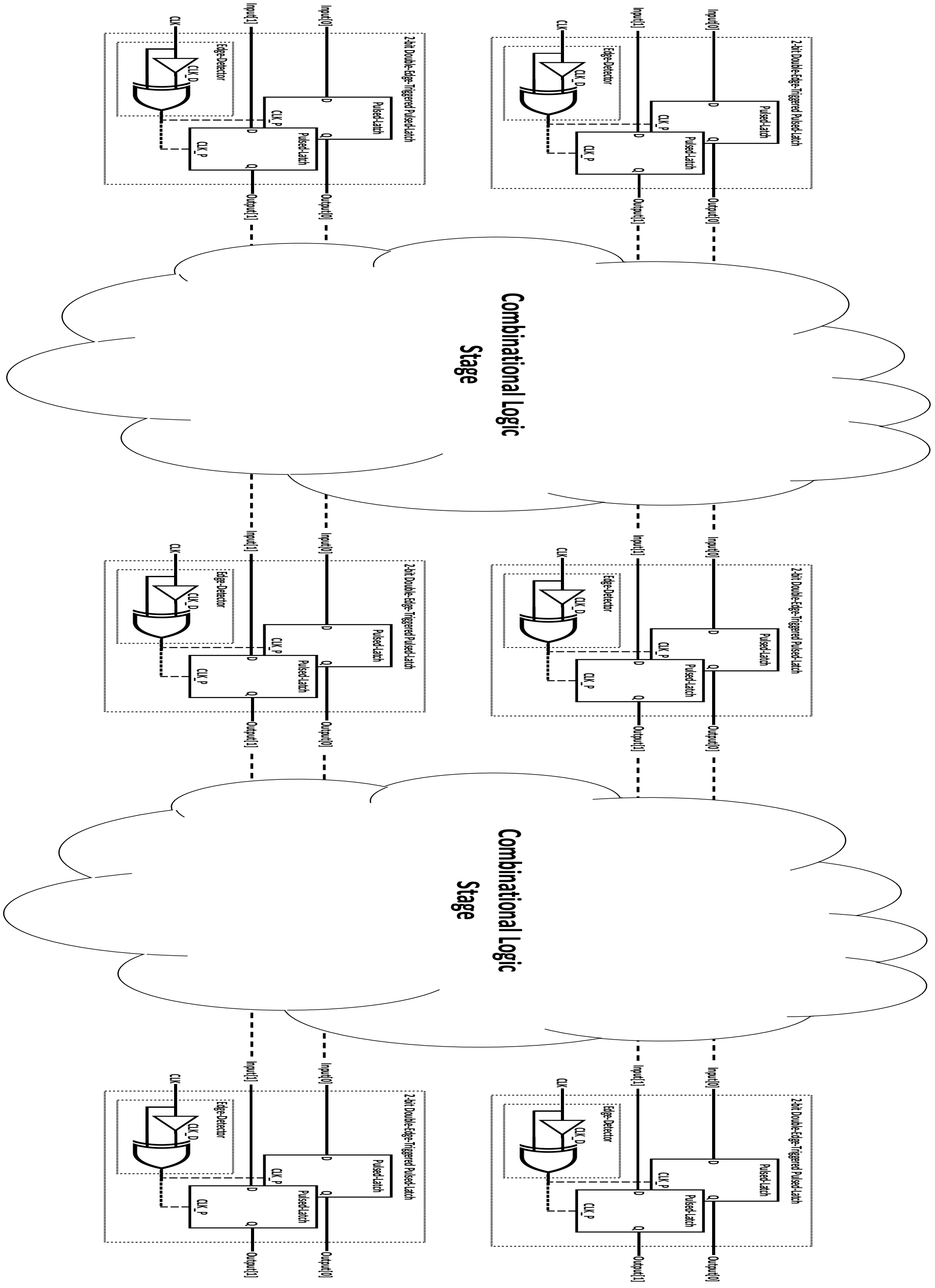


Figure 4.8: The double-edge-triggered microarchitecture using pulsed-latches with shared-edge-detector SED circuits. Each two adjacent pulsed-latches use only one edge-detector to reduce area overhead and power consumption.

Figure 4.8 shows the modified double-edge-triggered microarchitecture with the shared-edge-detector SED circuits. The memory unit of this microarchitecture is the 2-bit double-edge-triggered pulsed-latch. If there is a single pulsed-latch that is not adjacent/near to another pulsed-latch, a separate edge-detector circuit is used as shown in Figure 4.4. The placement constraint of the edge-detector is preserved to avoid any sub-tree generation which consumes more power. The transparency window of the pulsed-latches tolerates any potential negative clock skewing while the hold time constraints is conservative enough to tolerate any positive clock skewing. The proposed microarchitecture, shown in Figure 4.8, allows the industry to adopt the double-edge-triggering concept for large synchronous designs because it is expected to reduce the power consumption significantly with an acceptable area overhead.

4.2.3 PVT Variability Mitigation

Although the target of this Double-Edge-Triggered microarchitecture is to reduce the power consumption of the synchronous sequential circuits, it also belongs to the category of the time relaxation techniques that are able to mitigate the negative impacts of PVT variations.

For this microarchitecture, two scenarios are assumed regarding the PVT induced delay variations. The first scenario is that the PVT variations affect only some specific stages without affecting the other pipeline stages due to spatial correlated defects. If the defective stage is critical, it violates the timing constraints and generates runtime errors. For the proposed microarchitecture, this faulty critical stage tolerates the error by borrowing the time slack of its successive stage up to the toleration window ∆T which is determined at design-time.

The second scenario is that the PVT variations affect all stages and the whole pipeline is considered to be defective. In this pessimistic scenario, all critical and non-critical stages examine the induced delay variations. While only the critical stages violate the timing constraints and generate errors, the non-critical stages still offer time slacks that can be used to tolerate the errors by the proposed microarchitecture.

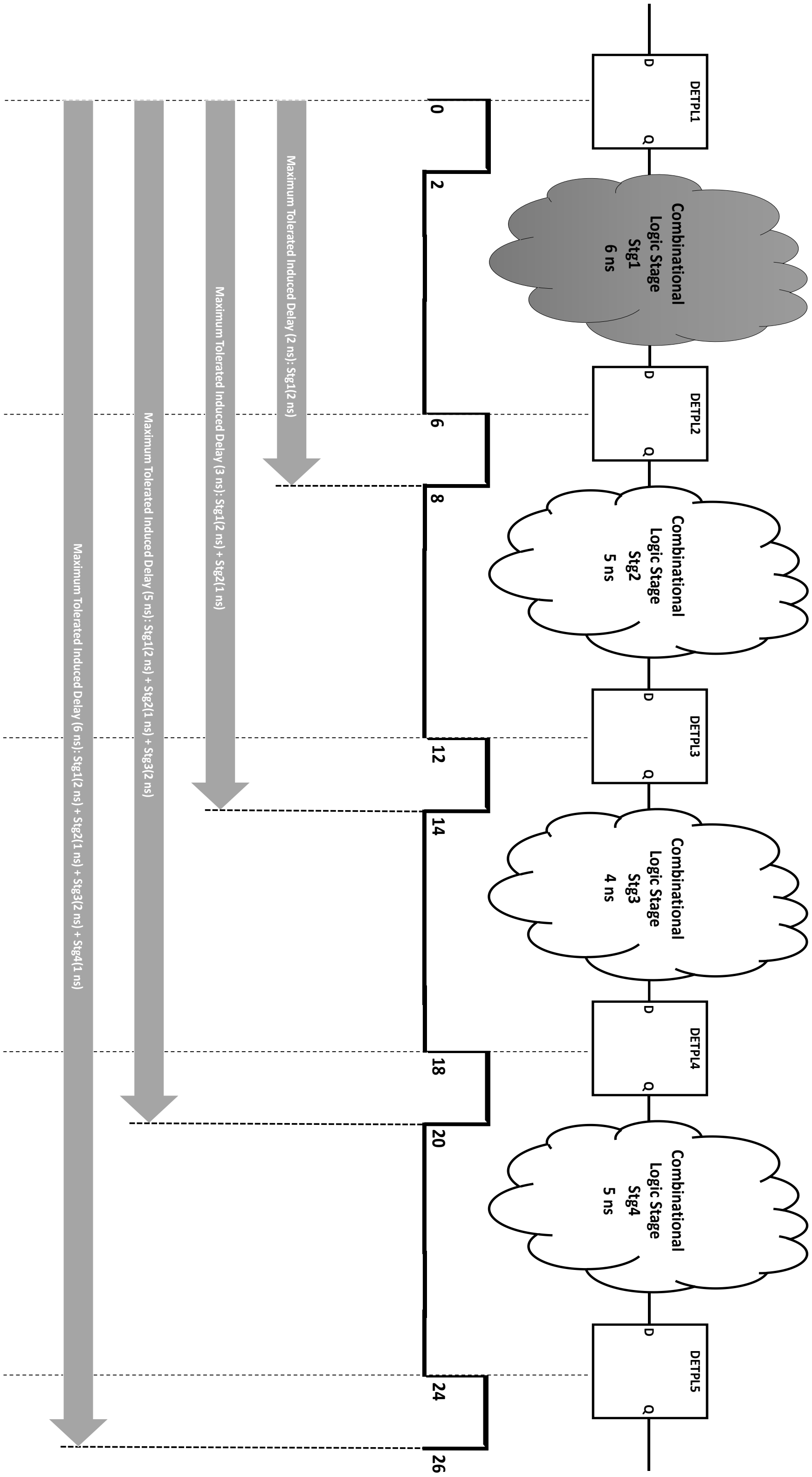

DETPL1
D
Q
Combinational Logic Stage Stg1 6 ns
DETPL2
Combinational Logic Stage Stg2 5 ns
DETPL3
Combinational Logic Stage Stg3 4 ns
DETPL4
Combinational Logic Stage Stg4 5 ns
DETPL5
0
2
6
8
12
14
18
20
24
26
Maximum Tolerated Induced Delay (2 ns): Stg1(2 ns)
Maximum Tolerated Induced Delay (3 ns): Stg1(2 ns) + Stg2(1 ns)
Maximum Tolerated Induced Delay (5 ns): Stg1(2 ns) + Stg2(1 ns) + Stg3(2 ns)
Maximum Tolerated Induced Delay (6 ns): Stg1(2 ns) + Stg2(1 ns) + Stg3(2 ns) + Stg4(1 ns)

Figure 4.9: An Example of a double-edge-triggered resilient microarchitecture examining PVT variations for all stages. The maximum accumulated induced delay for the whole pipeline that can be tolerated is up to 6 ns [2 ns for Stg1, 1 ns for Stg2, 2 ns for Stg3, and 1 ns for Stg4].

Figure 4.9 shows an example of the resilient double-edge-triggered microarchitecture for the second scenario in which all stages examine PVT variations. For simplicity, the T_{MPW} is ignored assuming that the toleration window ΔT is equal to the pulse width T_{PW}. As shown in Figure 4.9, the critical stage is assumed to be 6 ns which is equal to the clock period. Regarding the synchronous designs, the number of critical stages is small in comparison to the non-critical stages. So, it is assumed that only Stg1 is critical while the other stages Stg2, Stg3, and Stg4 are non-critical. The toleration window is assumed to be 2 ns so the critical stage Stg1 can deliver its output data later up to 2 ns. If the non-critical stage Stg2 receives its input after 8 ns instead of 6 ns, it still has enough time to deliver its output at 14 ns instead of 12 ns. This means that Stg2 has 3 ns slack: 2 ns to compensate the induced delay of Stg1 and only 1 ns for its induced delay. Consequently, Stg3 receives its input data at 14 ns instead of 12 ns and it has 2 ns for its potential induced delay. Then, Stg4 can receive its input late at 20 ns instead of 18 ns to compensate the previous induced delay and it still has 1 ns slack to compensate its own induced delay. At DETPL5, the maximum accumulated induced delay through the pipeline stages is up to 6 ns which is distributed as follows: 2 ns for Stg1, 1 ns for Stg2, 2 ns for Stg3, and 1 ns for Stg4.

From the previous example, a generic formula (4.3) is extracted to calculate the maximum accumulated induced delay (D_{Maxi}) that is tolerated at each double-edge-triggered pulse-latch DETPL. This tolerated delay is dependent on the pulse width of the clock period and the borrowed slack from the non-critical stages.

$$D_{Maxi} < (i-1).T_{CP} + T_{PW} - \sum_{j=1}^{i-1} Tm_j \ , \qquad \textit{Where } i = 1, 2, 3, \ldots, N \tag{4.3}$$

From (4.3), the maximum induced delay D_{Maxi} accumulated at the DETPLi is calculated at design-time to estimate the flexibility limits of the proposed design, where T_{CP} is the clock period, Tm_j is the maximum propagation delay of the stage j and N is the number of the pipeline stages. By substituting in formula (4.3) for the previous example shown in Figure 4.9, D_{Max1}, D_{Max2}, D_{Max3}, D_{Max4}, and D_{Max5} are calculated in (4.4), (4.5), (4.6), (4.7), and (4.8).

$$\begin{gathered} D_{Max1} < (0\ X\ 6) + (2) - (0), \\ D_{Max1} < 2 \text{ ns} \end{gathered} \tag{4.4}$$

$$\begin{gathered} D_{Max2} < (1\ X\ 6) + (2) - (6), \\ D_{Max2} < 2 \text{ ns} \end{gathered} \tag{4.5}$$

$$\begin{gathered} D_{Max3} < (2\ X\ 6) + (2) - (6 + 5), \\ D_{Max3} < 3 \text{ ns} \end{gathered} \tag{4.6}$$

$$\begin{gathered} D_{Max4} < (3\ X\ 6) + (2) - (6 + 5 + 4), \\ D_{Max4} < 5 \text{ ns} \end{gathered} \tag{4.7}$$

$$\begin{gathered} D_{Max5} < (4\ X\ 6) + (2) - (6 + 5 + 4 + 5), \\ D_{Max5} < 6 \text{ ns} \end{gathered} \tag{4.8}$$

4.2.4 Case Studies, 16x16 bit MAC Unit and ISCAS’89

To verify the Double-Edge-Triggered microarchitecture approach, a 16x16-bit MAC unit is selected to be a case study in addition to ISCAS’89 benchmark circuits with different weights. The MAC unit is a major building block for the machine learning accelerators and the digital signal processing units. Reducing the power consumption of the MAC unit reduces the total power consumption of these ASIC chips dramatically. ISCAS’89 benchmark circuits are also selected to examine the proposed microarchitecture for different circuit weights from the area perspective. After normalizing the area of all case study circuits, the smallest circuit is S27 (1x) and the largest one is S38417 (164.25x) while the MAC unit is about 59.3x.

4.2.5 Experimental Results

All benchmark circuits, including the MAC unit, are designed in Verilog and implemented through the normal ASIC Cell-based flow using Synopsys Tools for Global Foundries 65 nm technology and ARM standard cells library. The RTL code was synthesized using Synopsys Design Vision while the place and route was done by Synopsys IC Compiler. The power reduction is calculated by the Power Delay Product PDP concept while the overall performance is calculated for the Power Area Delay Product PADP.

Table 4.2 shows the results of the benchmark circuits using a separate edge-detector circuit for each DETPL. The average power reduction of all these circuits is 32.92% while the average area overhead is 47.37%. The MAC unit has the maximum power reduction 50.12% at the cost of 64.62% area overhead. The S35932 circuit has the minimum power reduction 15.83% with the largest area overhead 78.94%. This huge area overhead is due to the large number of flip-flops replaced by the double-edge-triggered pulsed-latches and the edge-detector circuits which increases the area dramatically and degrades the power saving offered by the proposed microarchitecture.

Table 4.2. The results of area and power delay product of the MAC unit and ISCAS'89 benchmark circuits for the double-edge-triggered microarchitecture with separate edge-detector.

	Normal Flip-flops		**Double-Edge-Triggered Microarchitecture**		**Performance Analysis**		
	Area (um^2)	**PDP (mW x nSec)**	**Area (um^2)**	**PDP (mW x nSec)**	**Area Overhead (%)**	**Power Reduction (%)**	**PADP Gain (%)**
S27	295.68	0.1867	375.72	0.1738	27.07	30.18	11.28
S420	1023.36	0.7350	1780.8	0.5746	74.02	29.64	-22.44
S838	2098.603	1.2765	2682.6	0.9311	27.83	27.06	6.76
S5378	7378.32	5.8406	9623.52	3.8499	30.43	34.08	14.02
MAC Unit	17534.773	6.0076	28866.0	3.1632	64.62	50.12	17.89
S35932	36825.36	18.5990	65894.64	15.6556	78.94	15.83	-50.61
S38417	48565.125	27.5702	62399.4	16.5348	28.5	43.55	27.46
Average					**47.34**	**32.92**	**0.623**

From Table 4.2, it is noticeable that if the registers occupy the majority of the synchronous design area, the area overhead increases significantly as in S420, MAC unit, and S35932. This costly penalty shows the importance of sharing the SED circuits to reduce the area overhead and not only to keep the power gain but also to increase this gain.

Table 4.3 shows the results of the benchmark circuits using a shared-edge-detector SED for each two adjacent DETPLs. The average power reduction for all these circuits is increased to 63.58% which is about 1.93x of the previous reduction while the average area overhead is reduced to 31.02% which is about 0.66x of the previous area overhead. The S38417 circuit has the maximum power reduction 67.2% at the cost of only 12.27% area overhead. The S35932 circuit still has the minimum power reduction but with 58.23% at the cost of 52.78% area overhead.

Figure 4.10 shows the comparison between the double-edge-triggered microarchitectures of the benchmark circuits with separate edge-detectors and the same circuits with shared-edge-detectors from the power reduction perspective. The power reduction is noticed to be increased even for the benchmarks with registers majority. Sharing the edge-detector between every two adjacent pulsed-latches removes the unneeded hardware redundancy and saves a lot of power consumed by the Xor gate and the delay element of the edge-detector circuit.

Figure 4.11 shows the comparison for the benchmark circuits from the area overhead perspective. It is costly and worthless to apply the double-edge-triggered microarchitecture with separate edge-detectors for the benchmark circuits in which the registers are the majority like S420 and S35932. After applying the shared-edge-detector concept, the area overhead for every design is reduced and the total performance gain becomes worthy. Thus, the proposed double-edge-triggered microarchitecture with shared-edge-detectors is now applicable for all synchronous circuits with an acceptable area overhead and a magnificent power gain.

Table 4.3. The results of area and power delay product of the MAC unit and ISCAS'89 benchmark circuits for the double-edge-triggered microarchitecture with shared-edge-detector SED.

	Normal Flip-flops		**Double-Edge-Triggered Microarchitecture (SED)**		**Performance Analysis**		
	Area (um^2)	**PDP (mW x nSec)**	**Area (um^2)**	**PDP (mW x nSec)**	**Area Overhead (%)**	**Power Reduction (%)**	**PADP Gain (%)**
S27	295.68	0.1867	368.28	0.08703	24.55	59.21	49.20
S420	1023.36	0.7350	1536.0	0.2653	50.09	63.90	45.82
S838	2098.603	1.2765	2420.64	0.4419	15.35	65.38	60.07
S5378	7378.32	5.8406	8353.32	2.0962	13.21	64.11	59.37
MAC Unit	17534.773	6.0076	26106.797	1.9796	48.89	67.05	50.94
S35932	36825.36	18.5990	56263.2	7.7693	52.78	58.23	36.18
S38417	48565.125	27.5702	54521.76	9.0416	12.27	67.20	63.18
Average					**31.02**	**63.58**	**52.11**

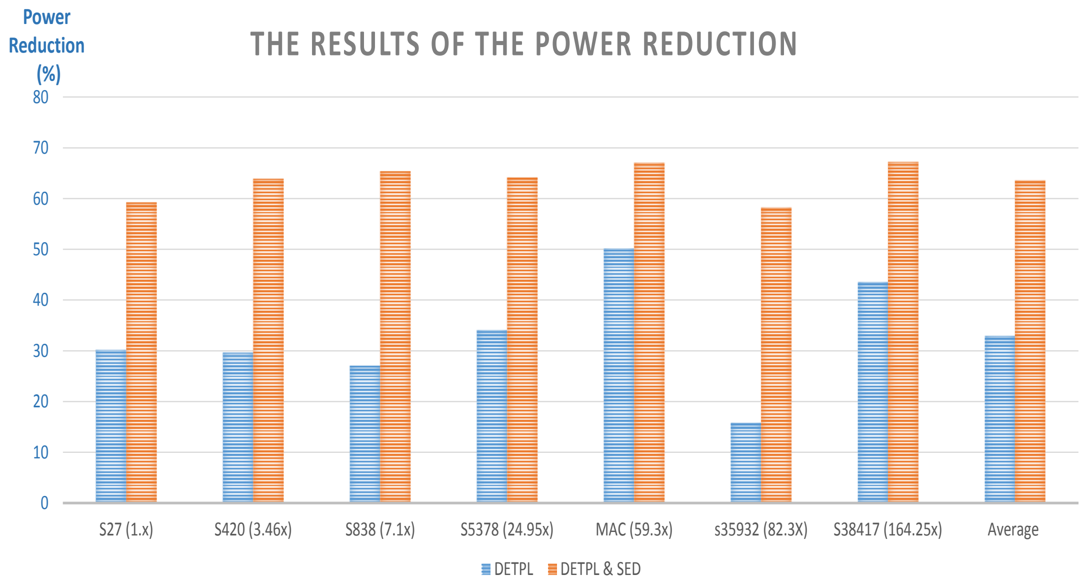


Figure 4.10: The comparison of the power reduction between Double-Edge-Triggered Pulsed-latches with separate edge-detector (DETPL) and Double-Edge-Triggered Pulsed-latches with Shared-Edge-Detector (DETPL & SED).

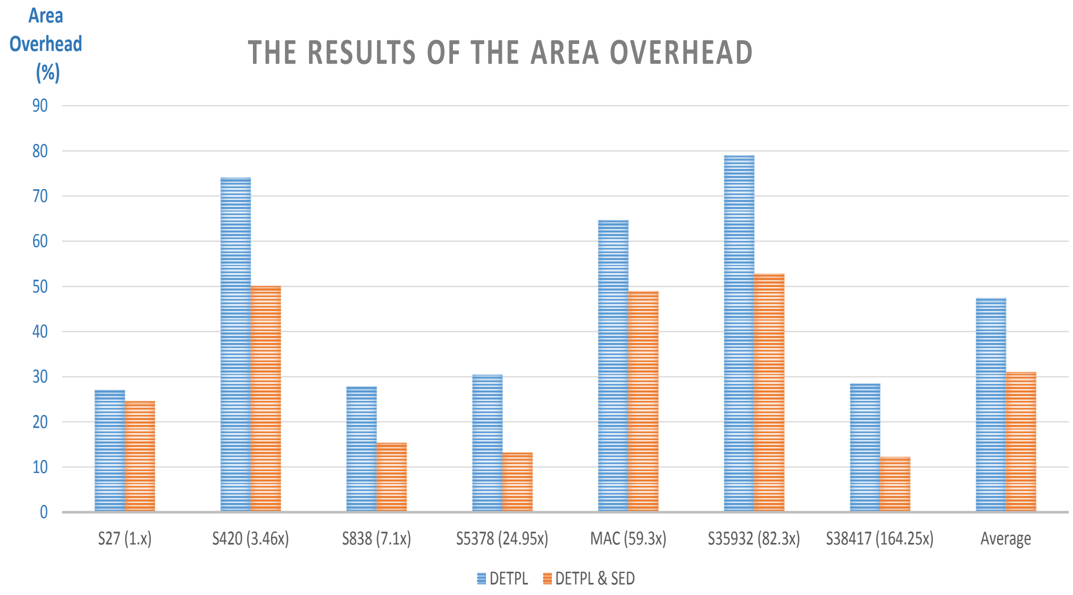


Figure 4.11: The comparison of the area overhead between Double-Edge-Triggered Pulsed-latches with separate edge-detectors (DETPL) and Double-Edge-Triggered Pulsed-latches with Shared-Edge-Detectors (DETPL & SED).

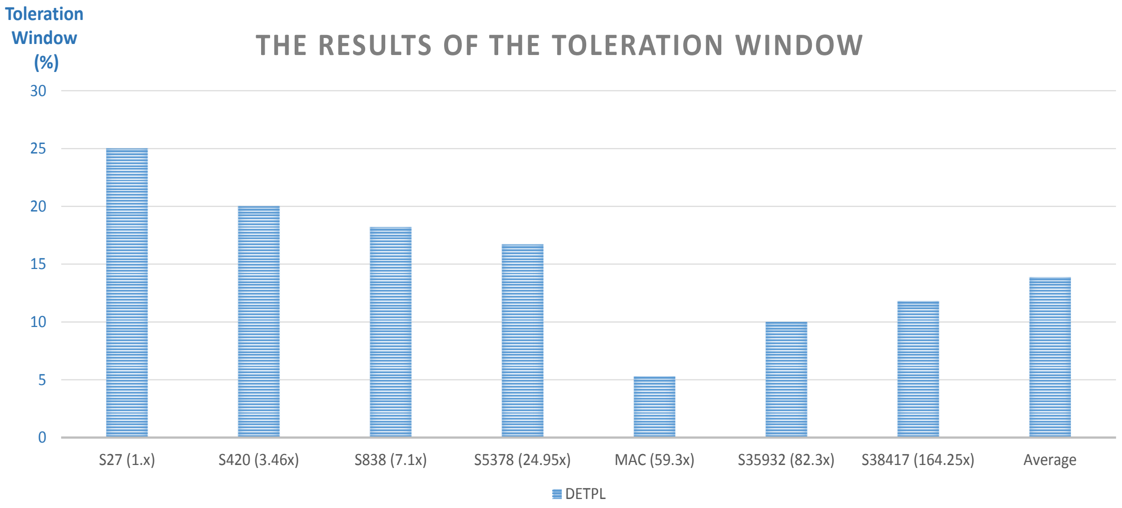


Figure 4.12: The toleration windows of the ISCAS’89 benchmark circuits and the MAC unit in comparison to their clock periods. The toleration window represents the PVT variations that can be tolerated at runtime without generating errors.

Regarding the PVT variations mitigation, Figure 4.12 shows the results of the toleration windows ΔTs of the ISCAS’89 benchmark circuits and the MAC unit in comparison to their clock periods. S27 circuit has the largest toleration window which is about 25% of its clock period while the MAC unit has the smallest toleration window which is about 5.3% of its clock period. These windows are mainly dependent on the values of the available delay elements which are offered by the standard cells library. The average toleration window of all designs is about 13.86% which represents the average PVT induced delay variations that are tolerated at runtime. Increasing the toleration window increases the area overhead because more short paths need to be readjusted not to violate the hold timing constraint.

4.2.6 Conclusion

The Double-Edge-Triggered microarchitecture with Shared-Edge-Detector circuit is a promising applicable approach to reduce the power budget of the digital integrated circuits efficiently with an acceptable area overhead. Although it does not tolerate large induced delays, unlike the ERSUT-based microarchitecture, it still mitigates the PVT variations by using the time relaxation concept. The normal ASIC Cell-based flow and the on-the-shelf standard cells library are used to implement the double-edge-triggering concept for the different ISCAS’89 benchmark circuits and the MAC unit. The proposed microarchitecture is very promising for low power designs because its average power reduction is 63.58% with average area overhead 31.02%.

Merging the double-edge-triggered microarchitecture with the runtime power management techniques like the Dynamic Voltage Scaling DVS and the ERSUT-based microarchitecture is a very promising direction to investigate the potential power gain could be achieved. Integrating ERSUT-based microarchitecture with the double-edge-triggering concept is expected to increase the power savings dramatically while increasing the ability of the targeted circuits to tolerate larger induced delays.

Chapter 5: ERSUT-based CAD Automation

This chapter is concerned about automating the ERSUT-based approach to be compatible with the STA CAD tools. This automation allows us to apply the ERSUT-based microarchitecture for different circuits and it facilitates the design process for more complex designs. A validation study about the potential benefits of applying ERSUT-based approach for different ISCAS'89 benchmark circuits is also discussed.

5.1 ERSUT-based CAD Algorithm

The ERSUT-based resilient microarchitecture is a promising approach to increase the performance of the synchronous sequential circuits. The main concept of the ERSUT-based approach is to extract the benefits of the slack available by the non-critical stages. This neglected slack is reused to compensate the induced delay variations related to PVT variability or Dynamic Voltage Scaling DVS. Applying the ERSUT-based resiliency permits the operating pipeline to save power without reducing the throughput or to increase the throughput at the cost of power. Therefore, there is a chance to break the old tradeoff between power consumption and throughput.

5.1.1 ERSUT-based Approach Implementation Flow

Figure 5.1 shows the flowchart of applying the ERSUT-based resilient microarchitecture for a certain design. First step is to do a Static Timing Analysis STA for the whole design while the expected delay variations are taken into consideration. After applying the expected induced delay variations, all critical paths are detected and examined for potential timing violations. For each critical path, the maximum required slack is calculated and its successive non-critical paths are detected and checked for the available slack. If the available slack is sufficient to tolerate the total induced delay for each critical path, the ERSUT-based resilient microarchitecture is valid to be applied. If there is no sufficient slack, or the critical path loops on itself, the ERSUT-based approach is not valid.

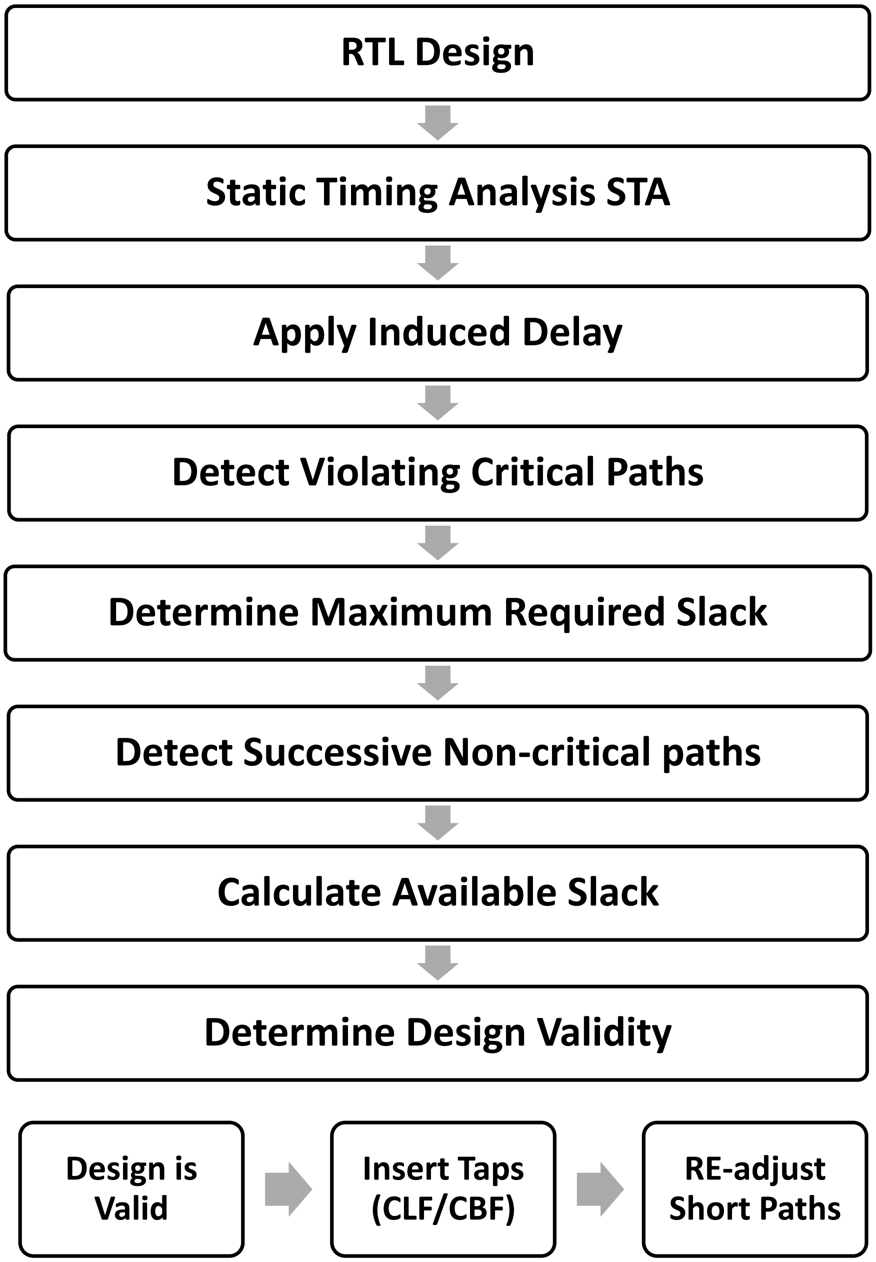

RTL Design
Static Timing Analysis STA
Apply Induced Delay
Detect Violating Critical Paths
Determine Maximum Required Slack
Detect Successive Non-critical paths
Calculate Available Slack
Determine Design Validity
Design is Valid
Insert Taps (CLF/CBF)
RE-adjust Short Paths

Figure 5.1: The flowchart of applying the ERSUT-based resilient microarchitecture for an RTL design.

After checking the validation of this design, the Taps including Configurable-Latch-Flip-Flops CLFs or Configurable-Buffer-Flip-Flops CBFs are inserted in their appropriate places. After ensuring that all setup violations are healed by inserting Taps, all short paths connected to CBFs or CLFs are readjusted to preserve the short path constraints eliminating any chance of hold violations or data racing.

In the case study of 16x16 bit MAC unit, the number of detected paths is about 4846 and the number of flip-flops is about 854. These huge numbers of paths and flip-flops need exhausting efforts to apply the ERSUT-based microarchitecture manually. The well-known internal architecture and functionality of the MAC unit is an important helpful factor that facilitates the manual work efforts. For larger designs, with greater number of flip-flops and paths, applying ERSUT-based approaches manually could be impossible. Even for designs smaller than the MAC unit with unknown architectures, applying ERSUT-based microarchitecture is still exhaustive and it consumes a lot of time. Time to market is an important factor that should be taken into consideration while developing a new approach. Thus, the ERSUT-based resiliency needs to reduce the efforts of its design phase. Developing an ERSUT-based CAD algorithm that is able to facilitate and accelerate the design process is an important achievement.

5.1.2 Algorithm General Overview

The proposed automated algorithm is an important step to accelerate the design process of the ERSUT-based approach and make a chance for more complicated designs to adopt it. This proposed algorithm is also useful to make a validation study about the potential benefits of applying the ERSUT-based resilient microarchitecture for different synchronous sequential circuits like ISCAS’89 benchmark circuits. The ERSUT-based algorithm should be compatible with the STA design tools like Synopsys Design Vision to be able to extract the information of the STA timing reports. As it is recommended to develop a portable algorithm that is compatible with different operating systems, C++ is used to develop the code. The final objective is to integrate the ERSUT-based algorithm with the common ASIC cell-based flow. Figure 5.2.a shows the common ASIC cell-based flow while Figure 5.2.b shows the new proposed ASIC cell-based flow after inserting the ERSUT-based Algorithm.

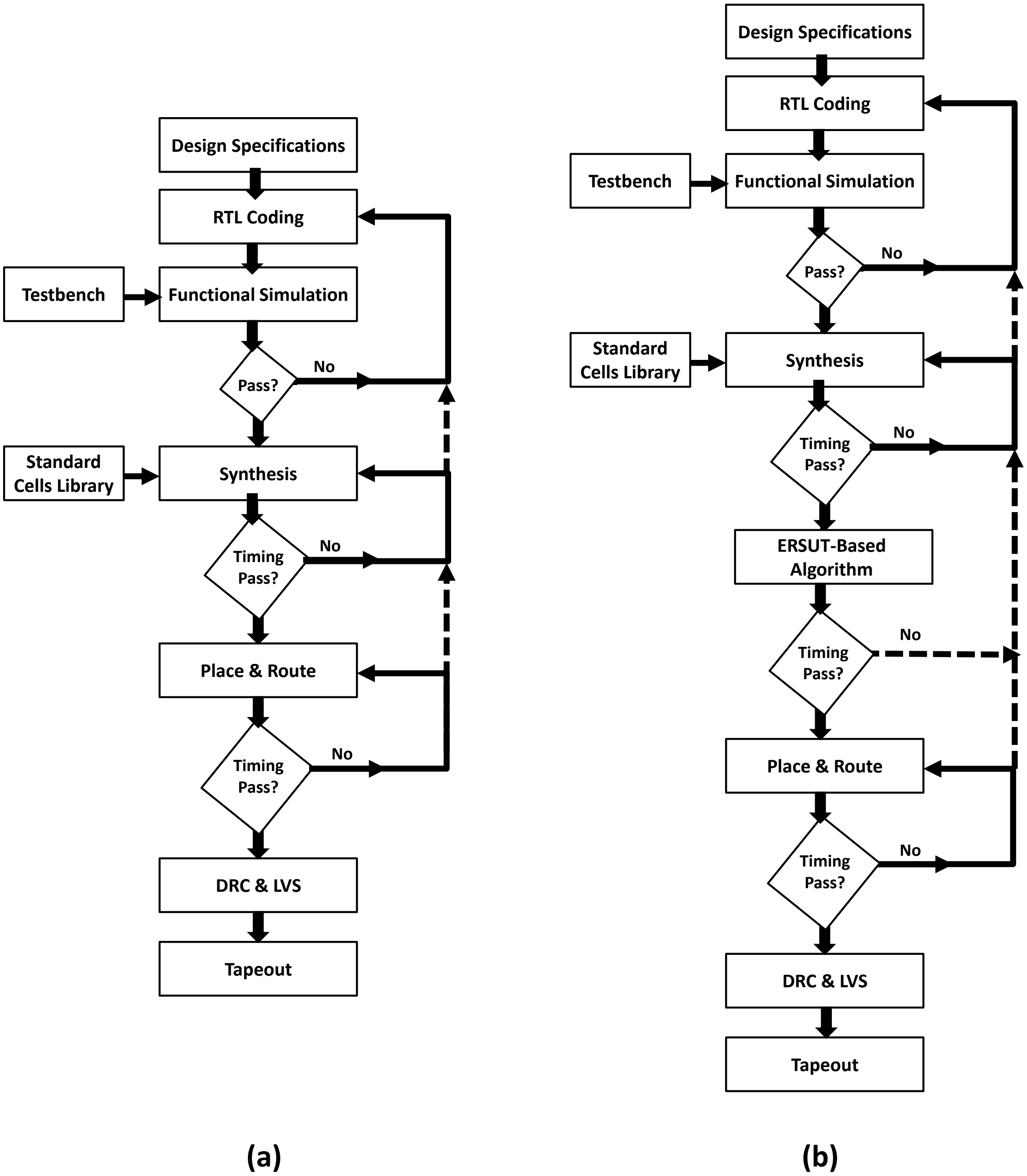


Figure 5.2: The flowchart of the ASIC cell-based flow. (a) The conventional flow, (b) The ERSUT-based flow.

The step of ERSUT-based algorithm is inserted after the Synthesis step to get the STA reports and it modifies the design turning it into a resilient one. Then the timing is checked again for any violations before proceeding through the Place and Route step.

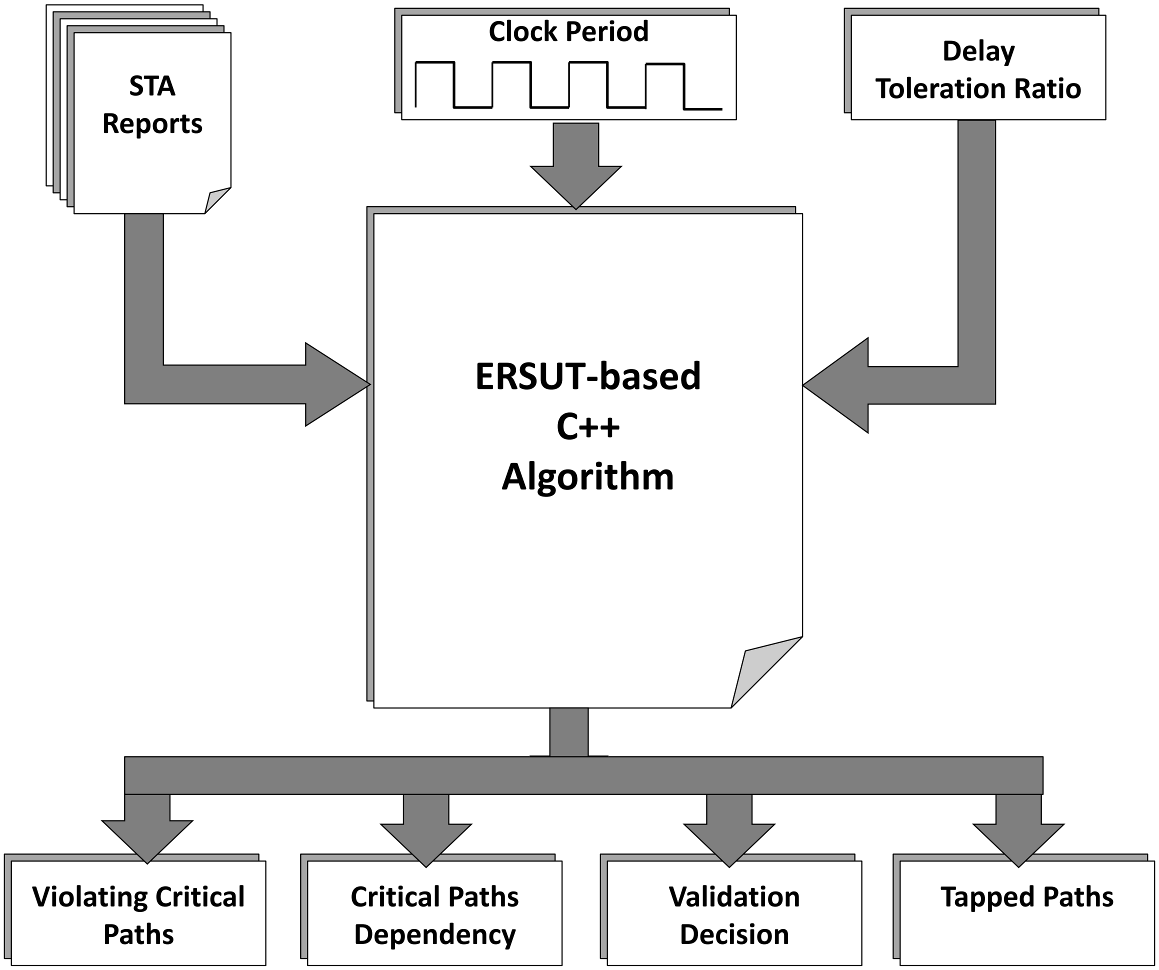


Figure 5.3: A general overview of the inputs and outputs of the ERSUT-based C++ Algorithm.

Figure 5.3 shows a general overview of the targeted algorithm including its inputs and outputs. This algorithm takes the reports of the static timing analysis as a text file in addition to the critical clock period used to generate these reports. The ERSUT-based algorithm also needs the expected induced delays should be tolerated in comparison to the clock period. The toleration ratio is calculated as a percentage of the clock period which covers the maximum expected induced delay. This delay toleration ratio is applied on all paths including critical and non-critical paths. The first output of the proposed algorithm is a list of the violating critical paths that are not able to tolerate the delay toleration ratio. The dependency among these critical paths is extracted and listed to check if there are successive critical paths that eliminate any chance of slack borrowing. The third output is the validation decision which confirms that the ERRSUT-based microarchitecture is applicable for the targeted design or not. The last output is a list of the critical paths and their successive non-critical paths that need Taps. This list shows the start point (flip-flop) and the end point (flip-flop) of each path that needs a Tap so every end point flip-flop should be replaced by a CLF or CBF.

5.1.3 ERSUT-based Algorithm Details

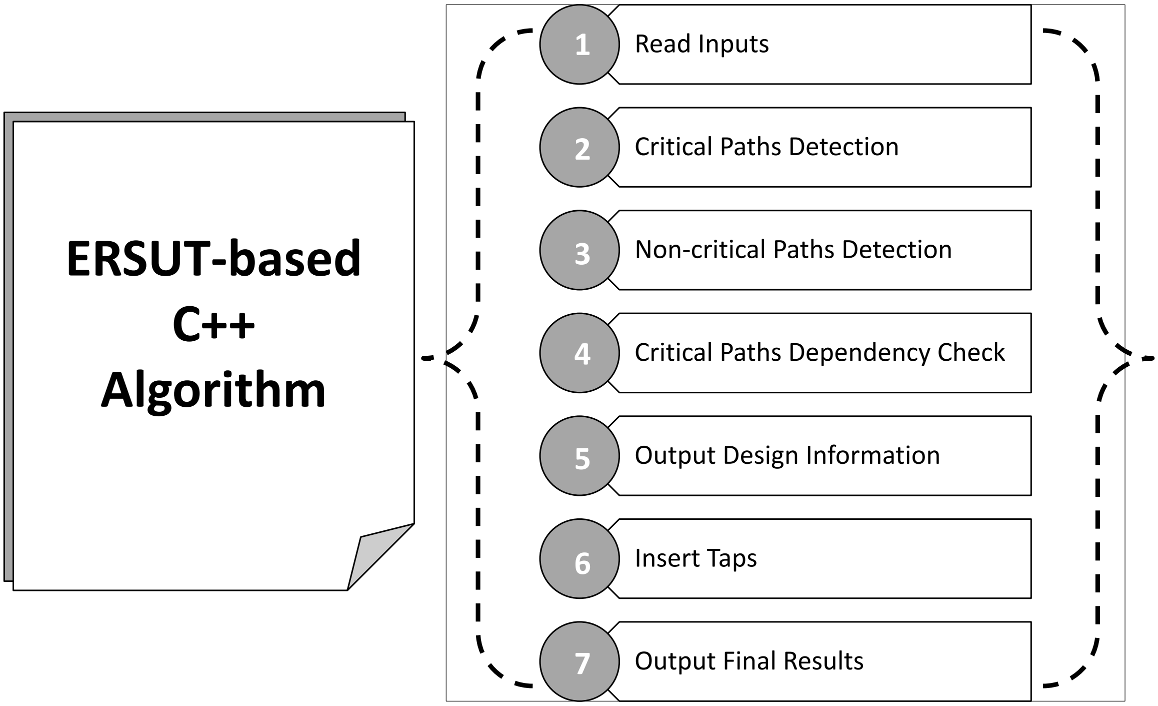

ERSUT-based C++ Algorithm
1
Read Inputs
2
Critical Paths Detection
3
Non-critical Paths Detection
4
Critical Paths Dependency Check
5
Output Design Information
6
Insert Taps
7
Output Final Results

Figure 5.4: The main functional blocks of the ERSUT-based C++ Algorithm.

Figure 5.4 shows the main functional blocks of the ERSUT-based algorithm. The first block is to read the inputs including the STA reporting text file. The function of the second block is to detect the critical paths violating the delay toleration ratio. The third block is dedicated to detect the non-critical paths of the whole design which is the slack borrowing pool for the critical paths. The block number four has a very important task which is tracing the dependency among the critical paths which determines if the algorithm is able to proceed with this design or not. Then, an output function lists the extracted information about the design and the decision of proceeding based on the critical paths dependency. The sixth block cares about tracing the non-critical paths to insert the required Taps, Then, the last block outputs the validation result of implementing the ERSUT-based microarchitecture and the list of the start and end points of the healed paths.

The required data for each path is saved by a struct which is a composite data type in C++ that groups different data variables to be saved under one name. Figure 5.5 shows the internal variables of the proposed struct called Path which saves the start point, the end point, the available slack, the slack of the previous predecessor path, and a flag to distinguish the critical path that needs a Tap. The whole integrated circuit design needs to be mapped by a bidirectional linked list of the Path struct nodes.

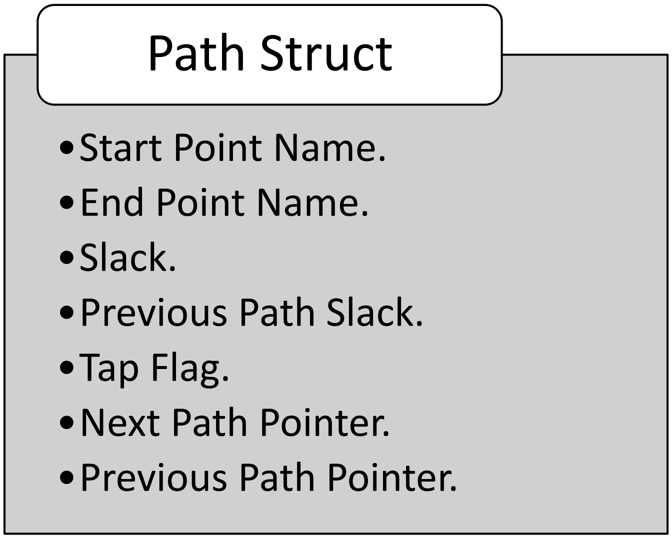


Figure 5.5: The components of the struct used by the ERSUT-based Algorithm.

For example, if the RTL design shown by Figure 5.6 is a potential input for the ERSUT-based algorithm, it should be mapped to a linked list which is shown in Figure 5.7. The design, shown in Figure 5.6, has 8 flip-flops which construct 12 different paths. These paths are represented by 12 nodes in the linked list shown in Figure 5.7.

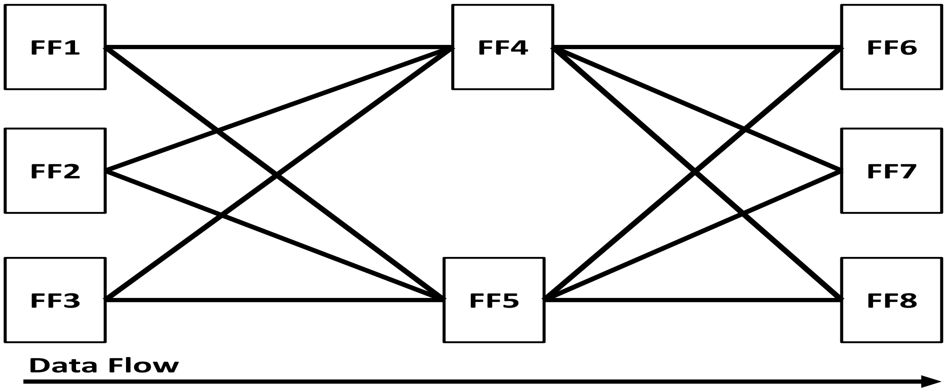
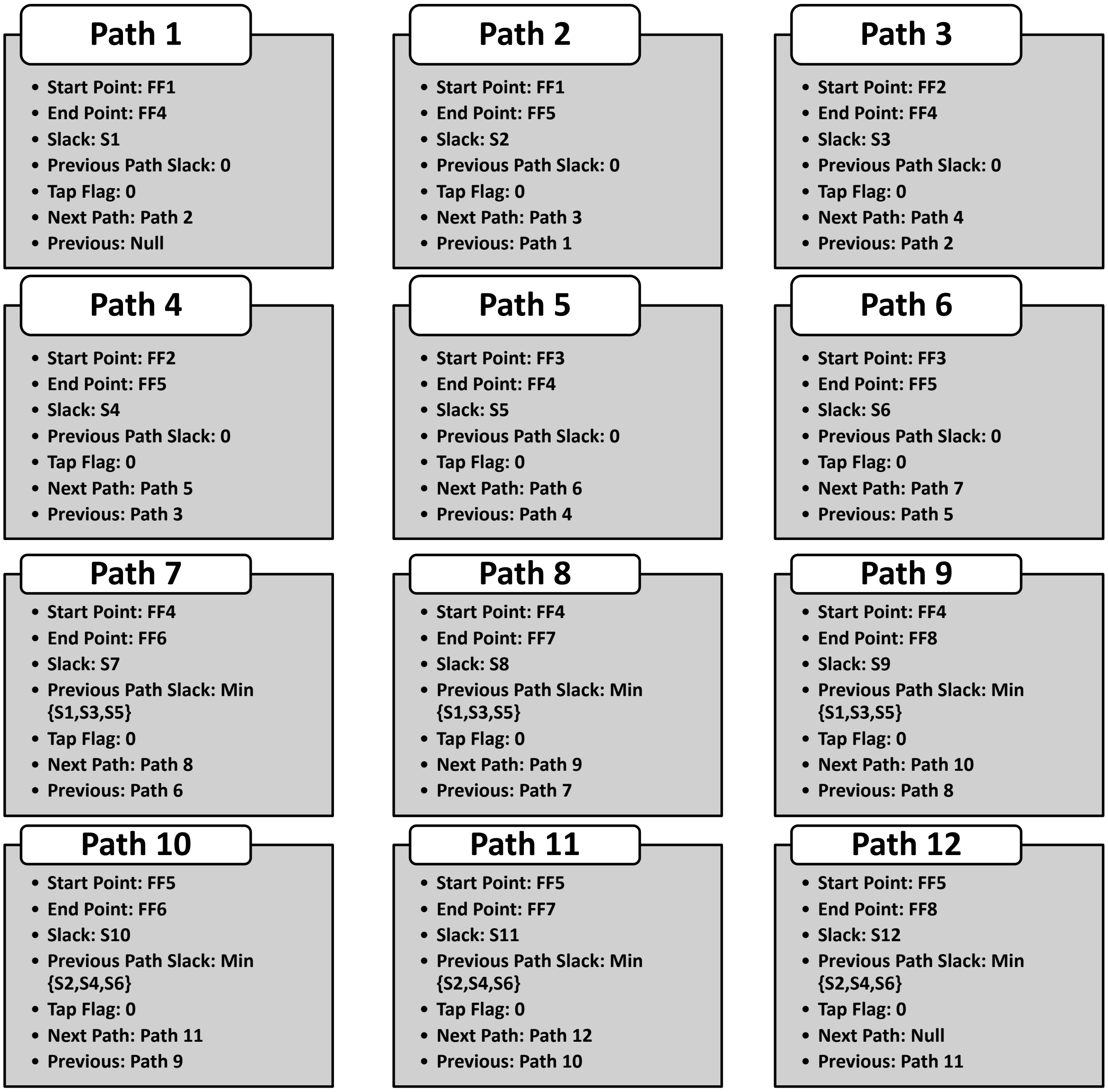


Figure 5.6: An example of RTL design to be mapped by the ERSUT-based algorithm.



Path 1

- Start Point: FF1
- End Point: FF4
- Slack: S1
- Previous Path Slack: 0
- Tap Flag: 0
- Next Path: Path 2
- Previous: Null

Path 2

- Start Point: FF1
- End Point: FF5
- Slack: S2
- Previous Path Slack: 0
- Tap Flag: 0
- Next Path: Path 3
- Previous: Path 1

Path 3

- Start Point: FF2
- End Point: FF4
- Slack: S3
- Previous Path Slack: 0
- Tap Flag: 0
- Next Path: Path 4
- Previous: Path 2

Path 4

- Start Point: FF2
- End Point: FF5
- Slack: S4
- Previous Path Slack: 0
- Tap Flag: 0
- Next Path: Path 5
- Previous: Path 3

Path 5

- Start Point: FF3
- End Point: FF4
- Slack: S5
- Previous Path Slack: 0
- Tap Flag: 0
- Next Path: Path 6
- Previous: Path 4

Path 6

- Start Point: FF3
- End Point: FF5
- Slack: S6
- Previous Path Slack: 0
- Tap Flag: 0
- Next Path: Path 7
- Previous: Path 5

Path 7

- Start Point: FF4
- End Point: FF6
- Slack: S7
- Previous Path Slack: Min {S1,S3,S5}
- Tap Flag: 0
- Next Path: Path 8
- Previous: Path 6

Path 8

- Start Point: FF4
- End Point: FF7
- Slack: S8
- Previous Path Slack: Min {S1,S3,S5}
- Tap Flag: 0
- Next Path: Path 9
- Previous: Path 7

Path 9

- Start Point: FF4
- End Point: FF8
- Slack: S9
- Previous Path Slack: Min {S1,S3,S5}
- Tap Flag: 0
- Next Path: Path 10
- Previous: Path 8

Path 10

- Start Point: FF5
- End Point: FF6
- Slack: S10
- Previous Path Slack: Min {S2,S4,S6}
- Tap Flag: 0
- Next Path: Path 11
- Previous: Path 9

Path 11

- Start Point: FF5
- End Point: FF7
- Slack: S11
- Previous Path Slack: Min {S2,S4,S6}
- Tap Flag: 0
- Next Path: Path 12
- Previous: Path 10

Path 12

- Start Point: FF5
- End Point: FF8
- Slack: S12
- Previous Path Slack: Min {S2,S4,S6}
- Tap Flag: 0
- Next Path: Null
- Previous: Path 11

Figure 5.7: The nodes of the bidirectional linked list representing the RTL design of Figure 5.6.

The previous path slack value is assigned to the minimum slack available by the predecessor path. For example, the starting point of Path 8 is FF4 while its end point is FF7 and its slack value is S8 nsec. Its previous path slack is the minimum value of S1, S3, and S5. The Tap flag is set by default to 0 until the path is detected as a violating path then the flag is set to 1. These nodes are the main data containers used by the different functional blocks which are discussed in detail as follows:

5.1.3.1 Read Inputs

Read Inputs functional block is the first block in the ERSUT-based algorithm. It reads the text file that contains the STA data. As the timing reports generated by the STA tool of Synopsys is widely used, and Synopsys is one of the most common ASIC cell-based flow tools, the format of its timing reports is adopted by the algorithm. This increases the ability of integrating the ERSUT-based algorithm with the synthesis tool of Synopsys.

Read Inputs block also gets the clock period used to generate the static timing analysis and the toleration ratio which is the percentage of the clock period should be tolerated due to the PVT variations or the Dynamic Voltage Scaling DVS. As the induced delay variations are assumed to affect the whole pipeline stages and their paths as in case of DVS, the toleration ratio is applied for all paths including critical and non-critical paths and the new slack values are calculated for all these paths.

5.1.3.2 Critical Paths Detection

This functional block extracts the start and end points of each critical path and its relative slack after applying the toleration ratio. If the value of the modified slack is less than or equal to zero, the path is considered to be critical. These critical paths are saved in a bidirectional linked-list of Path structs. The redundancy among these paths is removed while saving the current critical path information by comparing its start and end points with the previous detected paths. If the current critical path has the start and end points of another existing path, the redundancy is removed by taking the minimum available slack.

This could be happened because the data may propagate through different logic gates for the same start and end points. The output of this block is a list of the critical paths with their minimum slack values.

5.1.3.3 Non-critical Paths Detection

Just like the previous block, the non-critical paths detection block is responsible to extract the start and end points of each non-critical path and its relative slack after applying the toleration ratio. The non-critical paths are saved in another bidirectional linked-list and the redundancy between the non-critical paths is also removed. The linked list of the non-critical paths is concatenated to the list of the critical paths and the final output is a bidirectional linked list of all critical and non-critical paths which represents the whole design.

5.1.3.4 Critical Paths Dependency Check

The functionality of this block is concerned about the dependencies among the different critical paths. The algorithm traces the critical paths saved by the linked list to find if there are some cascaded critical paths. If the algorithm finds some cascaded critical paths, their negative slack values are accumulated and this total negative slack is used to determine the next step of the algorithm. If the absolute value of the total negative slack is greater than the clock period, the algorithm should be stopped because it fails to apply the ERSUT-based microarchitecture for the targeted design. On the other hand, if the absolute value of the accumulated negative slack is less than the clock period, the chance is still existing to apply the ERSUT-based microarchitecture and the algorithm proceeds to the next functional block.

5.1.3.5 Output Design information

At this step, the algorithm outputs some information about the targeted design. The number of detected paths for the whole design and the number of critical paths are displayed. The critical paths violating the toleration ratio are listed including their start points, end points, and required slack values. The dependencies among the different critical paths are listed while the accumulated negative slack is calculated per each dependency. The decision about proceeding to the next block is also displayed.

5.1.3.6 Insert Taps

Reaching this functional block does not mean that ERSUT-based microarchitecture is valid for the targeted design. The validation decision is taken at the end of this functional block. For each critical path, the successive paths are traced to insert Taps (including CLFs or CBFs). The negative slack of this critical path is compensated by the positive slack of its cascaded paths. The algorithm propagates through the whole tree of its successive paths until the negative slack is totally compensated. If a critical path is not able to borrow enough slack from its successive tree of paths (to compensate its negative slack), the ERSUT-based microarchitecture is considered to be invalid for this design.

5.1.3.7 Output Final Results

Finally, the algorithm lists the start point and the end point of each path that needs a Tap. The final decision about applying the ERSUT-based microarchitecture is stated. The list of Tapped paths can be used to replace their end point flip-flops by the Configurable-Latch-Flip-Flops if the relative borrowed slack is less than a half clock cycle or the Configurable-Buffer-Flip-Flops if the borrowed slack is greater than a half clock cycle.

The algorithm was tested and traced manually for many benchmark circuits to ensure the robustness of the algorithm. Then, the algorithm was used to test the validation of the ERSUT-based resilient microarchitecture for different circuits of ISCAS’89 benchmark.

5.2 Validation Study of ERSUT-based Approach

Automating the ERSUT-based microarchitecture design process is an important step to investigate the potential performance benefits could be offered to the different synchronous sequential circuits. Without developing the ERSUT-based algorithm, investigating the validation study is not applicable because even small designs with unknown architectures need exhaustive efforts to trace the critical and non-critical paths and to extract the maximum induced delay could be tolerated.

The ERSUT-based algorithm is applied to different ISCAS’89 benchmark circuits with different weights and complexities. The results extracted from the algorithm give a considerable estimation about the validity of the ERSUT-based microarchitecture for different circuits with different areas and complexities.

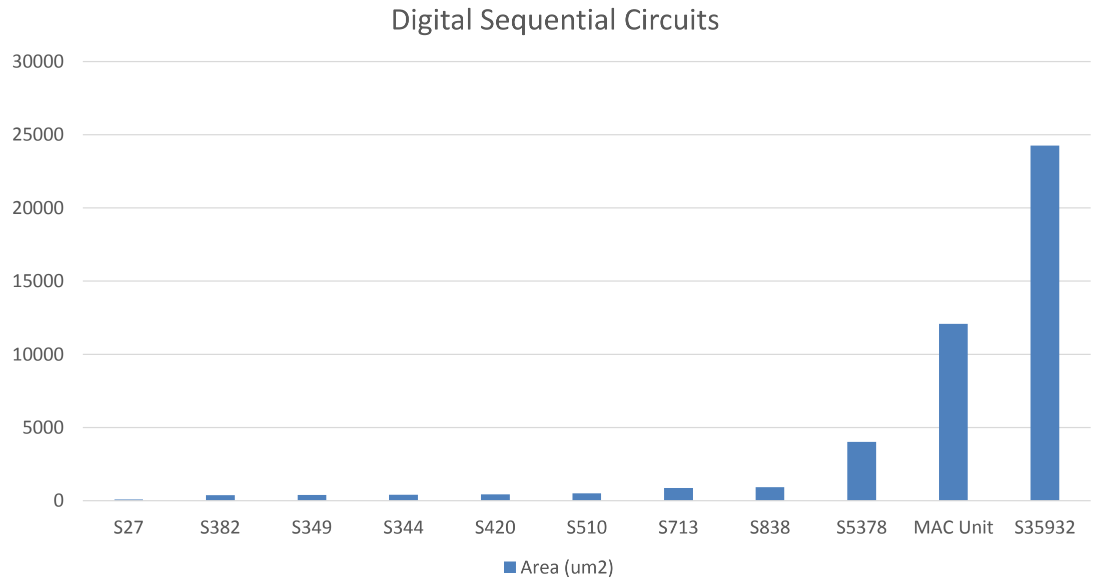


Figure 5.8: The areas of the different ISCAS’89 benchmark circuits including the MAC unit.

Figure 5.8 shows the areas of the ISCAS’89 benchmark circuits used to examine the ERSUT-based approach including the MAC unit. All these circuits are written in Verilog and synthesized using Global Foundries 65 nm technology and ARM standard Cells library. Figure 5.8 shows the wide variety of areas selected to examine the validation of the ERSUT-based microarchitecture. The smallest area is related to S27 which is about 81 um^2 while the largest area is about 24265 um^2 which is related to S35932. The results are extracted by Synopsys Synthesis Tool.

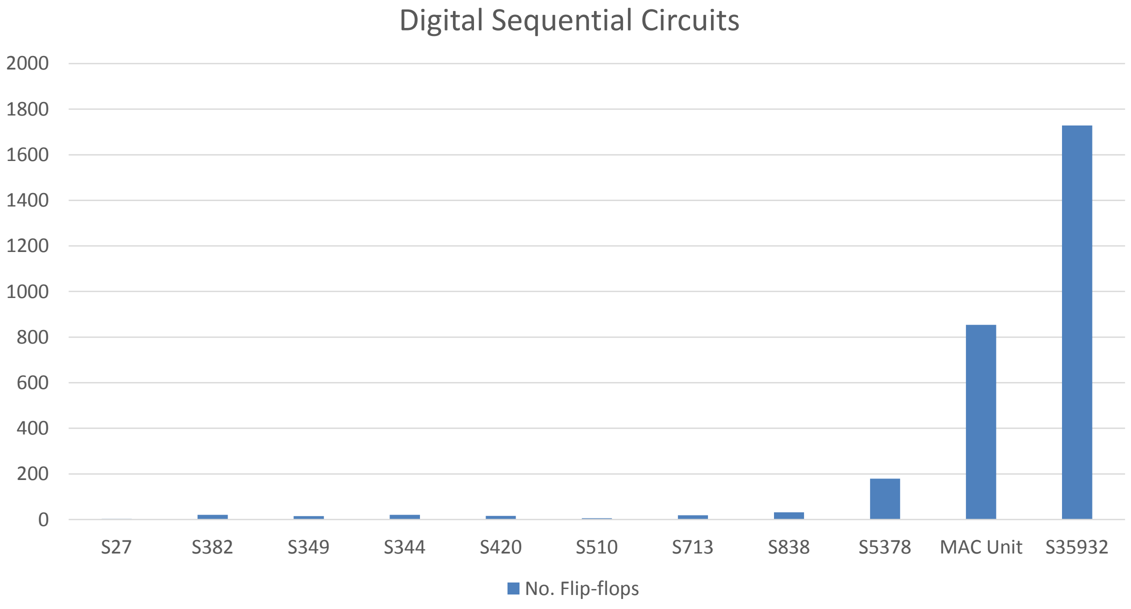


Figure 5.9: The numbers of flip-flops of the different ISCAS’89 benchmark circuits including the MAC unit.

Figure 5.9 shows the number of flip-flops per each design. The S35932 has the largest number of flip-flops which is about 1728. On the other hand, the MAC unit has 854 flip-flops while S27 has only a few number of flip-flops.

After applying the ERSUT-based algorithm for each circuit, the number of the critical paths per each circuit is extracted. Figure 5.10 shows the number of all paths for each circuit, while Figure 5.11 shows the percentage of the critical paths for each circuit in comparison to the number of all paths.

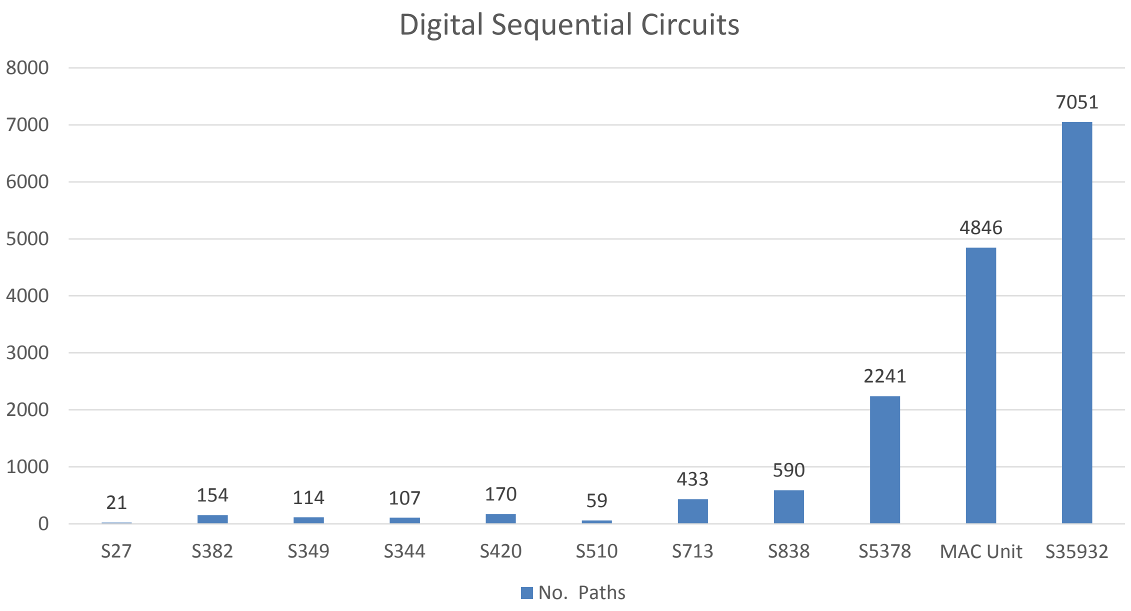


Figure 5.10: The numbers of all paths of the different ISCAS'89 benchmark circuits including the MAC unit.

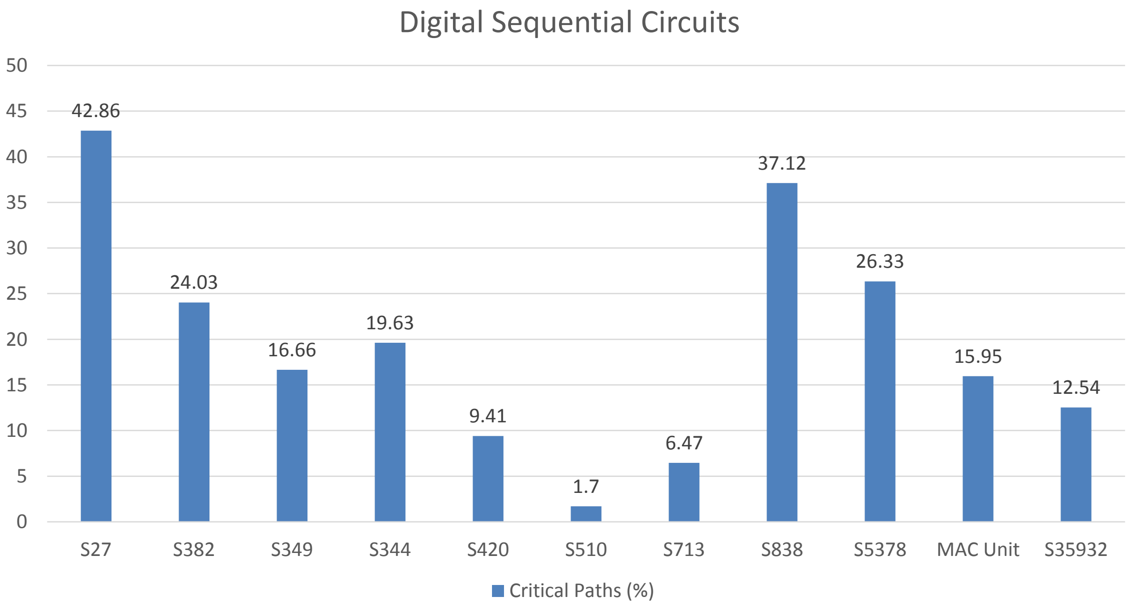


Figure 5.11: The percentage of critical paths in comparison to the number of all paths for the different ISCAS'89 benchmark circuits including the MAC unit.

S35932 has the largest number of paths which is 7051, while the MAC unit has 4846 paths. Although the area of S35932 is larger than the MAC unit, the percentage of its critical paths is less than the MAC unit. On the other hand, although S27 has the smallest area and the smallest number of paths, it has the largest percentage of critical paths which is 42.86%.

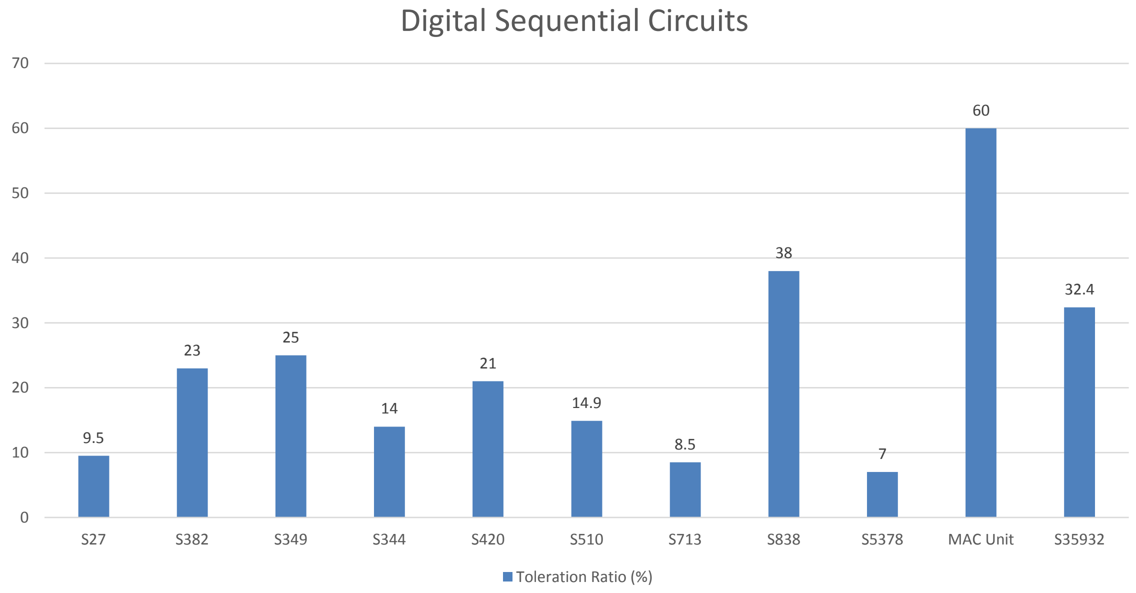


Figure 5.12: The percentages of the toleration ratio for the different ISCAS’89 benchmark circuits including the MAC unit.

Figure 5.12 shows the toleration ratio results extracted by the ERSUT-based algorithm. The algorithm is applied iteratively for each circuit to find the maximum toleration ratio which means the maximum tolerated induced delay in comparison to the clock period. S5378 has the minimum toleration ratio and it tolerates the induced delays up to only 7% of its clock period. On the other hand, the MAC unit has the largest toleration ratio and it tolerates up to 60% of its clock period. S35932, which is the largest circuit, tolerates up to 32.4% of its clock period while S27, which is the smallest circuit, tolerates up to 9.5%.

The results in Table 5.1 show that about 27% of the tested circuits has toleration ratios greater than 30% of their related clock periods, while 54.5% of these circuits has toleration ratios greater than 20%, and more than 72% of these circuits has toleration ratios greater than 14%. The rest 27% of the benchmark circuits (including the MAC unit) has toleration ratios from 7% to 9.5% which are still good values especially for PVT variability mitigation.

Table 5.1. The distribution of different circuits of ISCAS'89 including the MAC unit for the different toleration ratios.

ISCAS'89 Benchmark Circuits and MAC Unit	
Toleration Ratio (%)	**Percentage of Circuits (%)**
>30%	27%
>20%	54.5%
>14%	72%
7% to 9.5%	27%

These results illustrate that the percentage of the tolerated induced delays does not depend on the percentage of the critical paths nor the area of the targeted circuits. The toleration of the induced delays increases when the internal architecture of the digital circuit allows the critical paths to borrow more slack from their successive paths. The major limitation that reduces the percentage of the toleration ratio is the case of critical path self-looping. When the critical path feeds itself, there is no chance to borrow slack from itself to tolerate the induced delays.

5.3 Conclusion

The ERSUT-based algorithm is introduced as a promising solution to accelerate the design process of ERSUT-based resilient microarchitectures. Thanks to the developed algorithm, the exhaustive design efforts and time are saved by automating the process of tracing critical and non-critical paths to find the violating points that need Taps. The validation study of applying ERSUT-based resilient microarchitecture for different ISCAS'89 benchmark circuits is developed by using the ERSUT-based algorithm. It is so difficult to develop the validation study manually without using the automation step. The validation study on the MAC unit and the different ISCAS'89 benchmark circuits with different areas and complexities shows that 72% of these circuits tolerates more than 14% of their related clock periods while 27% of them tolerates more than 30%. This shows that the ERSUT-based resilient microarchitecture is applicable for these circuits. These different circuits can use the ERSUT-based approach to tolerate the PVT variations, to save power by scaling down the supply voltage without reducing the throughput, or even to increase the operating frequency at the typical supply voltage in case of no PVT variations.

The ERSUT-based algorithm needs more enhancement to increase its performance and to reduce its complexity. This algorithm needs to be fully automated as more functional blocks should be added to trace and re-adjust the short paths. Integrating the ERSUT-based algorithm with the STA tool and the ASIC-cell based flow is another future requirement. Afterwards, the ERSUT-based resilient microarchitecture will be more applicable for more complex industrial designs.

Chapter 6: Conclusion and Future Directions

6.1 Conclusion

High power consumption and PVT variability are the most urgent issues that affect the performance of the synchronous sequential circuits. Many approaches were developed to mitigate the PVT variations and to save the power at the cost of the throughput. Flushing the faulty pipeline or reducing the operating frequency are widely used to tolerate the PVT induced delays. On the other hand, DVS is widely used to reduce the power consumption but at the cost of reducing the operating frequency. The Correction Function technique was introduced to tolerate the PVT induced delays without flushing or restarting the faulty pipeline. It detects and corrects the error during one clock cycle. However, the overhead of area (45.9%) and power (42.7%) is high and the complexity of the correction function is proportional to the complexity of the non-critical logic stages which limits the adoption of the Correction Function technique by bigger designs.

The first version of ERSUT was introduced as a promising approach to tolerate the PVT induced delays. This approach detects the error and reconfigures the pipeline stages for recovery without flushing the pipeline. The 16x16 bit MAC unit tolerates PVT induced variations up to 20% of its clock period with 20.93% area overhead and 25.7% power overhead. However, the complexity of the error detection circuit decreases the chance to tolerate larger PVT induced delays. The relatively high power overhead does not support the power efficiency perspective. Just like Correction Function technique and the other Razor-based approaches, ERSUT suffers from the meta-stability issue which limits the adoption of all these better-than-worst-case approaches by the industry.

This research presents the ERSUT-based resilient microarchitecture as a promising solution to tolerate the PVT induced delays with low area and power overhead. In case of no PVT variations, the ERSUT-based resilient microarchitecture saves power by scaling down the supply voltage without reducing the operating frequency to preserve the maximum throughput. It is also able to increase the operating frequency at the cost of power for the typical supply voltages. The ERSUT-based resilient microarchitecture inherits the concept of detecting and recovering the error without flushing the pipeline stages or reducing the operating frequency. In contrast to ERSUT, the new approach reduces the complexity of the error detection circuit which allows to tolerate larger induced delays and to reduce the power and area overhead. Unlike ERSUT and all Razor-based approaches, the ERSUT-based resilient microarchitecture solves the meta-stability issue which affects the reliability and the throughput of the synchronous sequential circuits. The resilient 16x16 bit MAC unit tolerates PVT induced variations up to 50% of its clock period with 18.26 % area overhead and 2.05% power overhead at the typical supply voltage and the conventional critical frequency. In case of no PVT variations, the resilient MAC unit increases the operating frequency to 565 MHz (1.5x speedup). For the conventional critical frequency 375 MHz, the resilient MAC unit operates down to 1.0 V instead of the typical voltage 1.2 V, saving about 29% of the power consumed by the conventional MAC unit.

In another direction, the Double-Edge-Triggered microarchitecture using Shared-Edge-Detectors is introduced to reduce the power budget of the synchronous sequential circuits at design time. This approach reduces the power consumption of the clock tree components by reducing the frequency of the clock tree to the half while keeping the same throughput. The normal ASIC Cell-based flow and on-the-shelf standard cells library are used to design the different ISCAS'89 benchmark circuits and the 16x16 bit MAC unit to investigate the potential power gains of this approach. As the developed microarchitecture uses pulsed-latches and shares edge-detector circuits, it solves the main issue of the double-edge-triggering concept which is the unacceptable area overhead. The proposed microarchitecture is promising for low power designs because its average power reduction is 63.58% with average area overhead 31.02%. The transparency window of the pulsed-latches is used to relax the timing conditions among the adjacent stages and to tolerate the PVT induced delay variations which increases the performance and the reliability of the synchronous designs.

The ERSUT-based algorithm is developed to facilitate the ERSUT-based microarchitecture implementation. It reduces the exhaustive design-time efforts dramatically and it accelerates the process of applying the ERSUT-based microarchitecture on different circuits with different complexities. The developed algorithm explores the design and extracts the critical paths that violate the predetermined toleration ratio. It extracts the dependency among all these critical paths and calculates the accumulative required slack for these cascaded paths. Then, the developed algorithm traces each critical path to find its successive paths and borrows slack from these successive paths. The final results of this algorithm are a decision about the validation of applying the ERSUT-based microarchitecture and a list of the start and end points of the paths that need to be healed by Taps.

The validation study of applying the ERSUT-based microarchitecture on different ISCAS'89 benchmark circuits is built by using the ERSUT-based algorithm. The MAC unit and ISCAS'89 benchmark circuits with different weights and complexities were selected to investigate the potential benefits of applying the ERSUT-based approach. AS 72% of these circuits tolerates more than 14% of their clock periods, and 27% of these circuits tolerates more than 30%, the ERSUT-based resilient microarchitecture has been proved to be a valid promising approach to increase the performance of the synchronous sequential circuits. The ERSUT-based approach is valid to tolerate large PVT induced delay variations (from 7% to 60%) and it is also valid to save power by using DVS without decreasing the operation frequency until the voltage scaling induced delays reach the toleration ratio.

The ERUT-based resilient microarchitecture and its relative algorithm in C++ are introduced as an efficient solution to increase the reliability and performance of the synchronous sequential circuits. The validation study shows that the ERSUT-based approach is a valid approach to tolerate large PVT induced variations and to save power efficiently by integrating it with the DVS approach. The ERSUT-based resilient microarchitecture introduces as new relation between power and throughput as it reduces the power consumption without reducing the throughput. The Double-Edge-Triggered microarchitecture with Shared-Edge-Detectors is a promising solution for the low power designs at the cost of the area. It reduces the power budget significantly by reducing the power consumption of the clock tree components.

6.2 Future Directions

The ERSUT-based resilient microarchitecture reduces the power consumption at runtime by using the dynamic voltage scaling approach while the Double-Edge-triggered microarchitecture with Shared-Edge-Detectors reduces the power budget at design time. Consequently, integrating the two approaches together is a promising future direction as the total power reduction is expected to be huge. After merging the two approaches, the ERSUT-based approach will increase the ability of PVT variability toleration while the double-edge-triggering concept will decrease the power budget of the targeted design at the typical operating conditions.

Investigating the potential solutions to decrease the area overhead of the Double-Edge-triggered microarchitecture and the ERSUT-based microarchitecture is another research direction that improves the chances of these approaches to be adopted by larger industrial designs.

Applying the ERSUT-based microarchitecture for more synchronous sequential designs like industrial processors, GPUs, and IoTs is an important direction that requires more automation for this approach.

Optimizing the performance of the developed ERSUT-based algorithm by reducing its complexity is very important to deal with larger circuits and more complicated designs. In addition to optimizing the existing algorithm, adding more functional blocks is another step towards the full automation of the ERSUT-based approach. Tracing the short-paths and applying the timing constraints of the ERSUT-based approach to readjust these paths are important future functions that should be added to the developed algorithm.

Finally, the full integration of the ERSUT-based algorithm with the common ASIC-cell based flow tools (like Synopsys) is a promising step to extract the potential benefits of the unused slacks available by the non-critical paths which may lead to unexpected performance improvement.

References:

[1] Moore's law, 10 January 2018. [Online]. Available: https://en.wikipedia.org/wiki/Moore%27s_law.

[2] A. Pan, "A Hardware Framework for Yield and Reliability Enhancement in Chip Multiprocessors", MSc thesis, Graduate School of the University of Massachusetts Amherst, September 2009.

[3] S. Agwa, E. Yahya and Y. Ismail, "Design techniques for variability mitigation", Int. J. Circuits and Architecture Design, Vol. 1, No. 1, 2013.

[4] H. Sathyamurthy, S. S. Sapatnekar, and J. P. Fishburn, "Speeding Up Pipelined Circuits Through a Combination of Gate Sizing and Clock Skew Optimization", 1995 IEEE/ACM International Conference on Computer-Aided Design, 1995. ICCAD-95. Digest of Technical Papers, pp.467-470.

[5] S. Agwa, E. Yahya and Y. Ismail, "Variability Mitigation Using Correction Function Technique", 2013 IEEE International Conference on Electronics, Circuits, and Systems, ICECS, pp.293-296, December 2013.

[6] M. Olivieri, G. Scotti, and A. Trifiletti, "A Novel Yield Optimization Technique for Digital CMOS Circuits Design by Means of Process Parameters Run-Time Estimation and Body Bias Active Control", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 13, No. 5, pp.630-638, MAY 2005.

[7] R. Teodorescu, J. Nakano, A. Tiwari, J. Torrellas, "Mitigating Parameter Variation with Dynamic Fine-Grain Body Biasing", 40th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO, pp.27-42, December 2007.

[8] D. Andrade, F. Martorell, A. Calomarde, F. Moll, and A. Rubio, "A new compensation mechanism for environmental parameter fluctuations in CMOS digital Ics", Microelectronics Journal, June 2009, Vol. 40, No. 6, pp.952–957.

[9] J. Long, J. C. Ku, S. O. Memik, and Y. Ismail, "SACTA: A Self-Adjusting Clock Tree Architecture for Adapting to Thermal-Induced Delay Variation", IEEE Transactions on

Very Large Scale Integration (VLSI) Systems, September 2010 ,Vol. 18, No. 9, pp.1323-1336.

[10] S. Paik, S. Lee, and Y. Shin, "Retiming Pulsed-Latch Circuits with Regulating Pulse Width", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 30, No. 8, August 2011, pp.1114-1127.

[11] M. Wieckowski, Y. M. Park, C. Tokunaga, D. W. Kim, Z. Foo, D. Sylvester, D. Blaauw, "Timing Yield Enhancement Through Soft Edge Flip-Flop Based Design", IEEE Custom Integrated Circuits Conference, 2008, CICC 2008, September 2008, pp.543-546.

[12] A. Tiwari, S. R. Sarangi, and J. Torrellas, "ReCycle: Pipeline Adaptation to Tolerate Process Variation", 34th Annual International Symposium on Computer Architecture, pp.323-334, June 2007.

[13] Y. Kunitake, T. Sato, and H. Yasuura, "A Replacement Strategy for Canary Flip-Flops", IEEE 16th Pacific Rim Int. Symp. on Dependable Computing, PRDC, pp.227-228, Dec. 2010.

[14] M. Choudhury, V. Chandra, K. Mohanram, and R. Aitken, "TIMBER: Time Borrowing and Error Relaying for Online Timing Error Resilience", Design, Automation & Test in Europe Conference & Exhibition DATE, pp.1554-1559, March 2010.

[15] 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. of the 36th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-36, December 2003, pp.7-18.

[16] S. Lee, S. Das, T. Pham, T. Austin, D. Blaauw, T. Mudge, "Reducing Pipeline Energy Demands with Local DVS and Dynamic Retiming", In Proc. of the 2004 International Symposium on Low Power Electronics and Design, ISLPED '04, August 2004, pp.319-324.

[17] 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, JSSC, Vol. 41, No. 4, pp.792-804, April 2006.

[18] 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, JSSC, Vol. 44, No. 1, pp.32-48, Jan. 2009.

[19] K. A. Bowman, et al., "A 45 nm Resilient Microprocessor Core for Dynamic Variation Tolerance," IEEE J. Solid-State Circuits, VOL. 46, NO. 1, pp.194-208, Jan. 2011.

[20] 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, JSSC, Vol. 48, No. 1, pp.66-81, Jan. 2013.

[21] S. Kim, et al., "Razor-Lite: A Side-Channel Error-Detection Register for Timing-Margin Recovery in 45nm SOI CMOS," ISSCC Dig. Tech. Papers, pp.264-266, Feb. 2013.

[22] S. Beer, M. Cannizzaro, J. Cortadella, R. Ginosar, and L. Lavagno, " Metastability in Better-Than-Worst-Case Designs", 2014 20th IEEE International Symposium on Asynchronous Circuits and Systems, ASYNC, pp. 101-102, May 2014.

[23] S. Agwa, E. Yahya and Y. Ismail, "Power efficient AES core for IoT constrained devices implemented in 130nm CMOS," 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, 2017, pp. 1-4.

[24] A. Namazi and M. Abdollahi, "PCG: Partially Clock-Gating Approach to Reduce the Power Consumption of Fault-Tolerant Register Files," 2017 Euromicro Conference on Digital System Design (DSD), Vienna, 2017, pp. 323-328.

[25] H. Jiang, M. Marek-Sadowska and S. R. Nassif, "Benefits and costs of power-gating technique," 2005 International Conference on Computer Design, San Jose, CA, USA, 2005, pp. 559-566.

[26] Ivar Håkon Lysfjord, "Multiple Power Domains", MSc thesis, Department of Electronics and Telecommunications, Norwegian University of Science and Technology, June 2008.

[27] H.-T. Lin, Y.-L. Chuang, and T.-Y. Ho, “Pulsed-latch-based Clock Tree Migration for Dynamic Power Reduction,” in Proc. Int. Symp. Low Power Electron. Design, Aug. 2011, pp. 39–44.

[28] E. Consoli, G. Palumbo, J. Rabaey and M. Alioto, "Novel Class of Energy-Efficient Very High-Speed Conditional Push–Pull Pulsed Latches", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 7, pp. 1593-1605, 2014.

[29] I. A. Khan, D. Shaikh and M. T. Beg, "2 GHz Low Power Double Edge Triggered Flip-flop in 65nm CMOS Technology," 2012 IEEE International Conference on Signal Processing, Computing and Control, Waknaghat Solan, 2012, pp. 1-5.

[30] P. Sameni and S. Mirabbasi, "A Fully Differential High-speed Double-edge Triggered Flip-flop (DETFF)," Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513), 2004, pp. 1459-1462 Vol.3.

[31] R. Chandrasekaran, Yong Lian and Ram Singh Rana, "A High-speed Low-power D Flip-flop," 2005 6th International Conference on ASIC, Shanghai, 2005, pp. 82-85.

[32] M. Parsa, M. Aleshams and M. Imanieh, "A New Structure of Low-power and Low-voltage Double-edge Triggered Flip-flop," 2014 International Conference on Advances in Energy Conversion Technologies (ICAECT), Manipal, 2014, pp. 118-124.

[33] Yu-Yin Sung and R. C. Chang, "A Novel CMOS Double-edge Triggered Flip-flop for Low power Applications," 2004 IEEE International Symposium on Circuits and Systems, 2004, pp. II-665-8 Vol.2.

[34] T. A. Johnson and I. S. Kourtev, "A Single Latch, High Speed Doubleedge Triggered Flip-flop (DETFF)," ICECS 2001. 8th IEEE International Conference on Electronics, Circuits and Systems, 2001, pp. 189-192 vol.1.

[35] W. M. Chung, “The Usage of Dual Edge Triggered Flip-flops in Low Power, Low Voltage Applications”, A thesis presented to the University of Waterloo, Master of Applied Science in Electrical and Computer Engineering Waterloo, Ontario, Canada, January 2003.

[36] Ch-Ch. Wang, G-N. Sung, M-K. Chang, and Y-Y. Shen, “Energy-Efficient Double-Edge Triggered Flip-Flop Design,” IEEE Asia Pacific Conference on Circuits and Systems, APCCAS 2006. 4-7 Dec. 2006, pp.1792–1795.

[37] D. L. Oliveira, T. Curtinhas, L. A. Faria, J. L. V. Oliveira and L. Romano, "Design of Low-power Two-hot Finite State Machines Operating on Double edge Clock," 2016 IEEE ANDESCON, Arequipa, 2016, pp. 1-4.

[38] S. Agwa, E. Yahya and Y. Ismail, "ERSUT: A Self-Healing Architecture for Mitigating PVT Variations without Pipeline Flushing", IEEE Transactions on Circuits and Systems II, TCASII Express Briefs, Volume: 63, Issue: 11, pp.1069-1073, November 2016.

[39] S. Agwa, E. Yahya and Y. Ismail, "A Low Power Self-healing Resilient Microarchitecture for PVT Variability Mitigation," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. PP, no. 99, pp. 1-10, 2017.

[40] H. T. Lin, Y. L. Chuang, Z. H. Yang and T. Y. Ho, "Pulsed-Latch Utilization for Clock-Tree Power Optimization," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 4, pp. 721-733, April 2014.

[41] V. Ahuja, P. T. Karule and U. S. Ghodeswar, "Design of High Speed Conditional Push Pull Pulsed Latch," 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, 2016, pp. 1374-1378.