Low-Power and Error-Resilient VLSI Circuits and Systems

by

Chia-Hsiang Chen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering)
in the University of Michigan
2014

Doctoral Committee:

Assistant Professor Zhengya Zhang, Chair
Associate Professor Achilleas Anastasopoulos
Professor David Blaauw
Assistant Professor Junjie Zhu

© Chia-Hsiang Chen 2014
All Rights Reserved

To my family and "TEA TIME" friends for their love and support

ACKNOWLEDGEMENTS

Many people have contributed to the completion of this dissertation to whom I will always be grateful. My adivisor, Professor Zhengya Zhang, provided me the exceptional scientific resources and guidance, and allowed me to grow intellectully and to explore exciting researches over the years. I would also like to thank Professor David Blaauw, Achilleas Anastasopoulos, and Junjie Zhu, for participating in my dissertation committee and providing valuable feedback.

I appreciate my research sponsors, including Intel, DARPA under cooperative agreement HR0011-13-2-0006, NASA, and the University of Michigan.

I would like to thank all individuals who have inspired and mentored me throughout the doctoral training. Professor Michael Flynn provided me insightful discussions over different ideas. Dr. Keith Bowman, Dr. Farhana Sheikh, Dr. Charles Augustine, and Jim Tschanz from Intel Circuit Research Labs, they inspired me and broadened my knowledge. Another group of people that assisted me through this journey were those in my research group, including Jungkuk Kim, Youn Sung Park, Phil Knag, Yaoyu Tao, Siming Song, Shuanghong, Thomas Chen, and Wei Tang. They shared their expertise and effort to accelerate my understanding on several researching topics. In addition to those in my research group, I would also like to thank Dr. Wei-Hsiang Ma, Dr. Zhiyoong Foo, Tai-Chuan Ou, and Ye-Sheng Kuo for bring enjoyable memories as circuit designers at the University of Michigan .

Finally, I want to express my deepest appreciation to my parents and family in Taiwan for their unconditional love and support.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF FIGURES vii

LIST OF TABLES xiv

ABSTRACT xvi

CHAPTER

I. Introduction 1

1.1 Error characterization and impact analysis 2
1.1.1 Data-retention failures and timing faults 2
1.1.2 Soft errors 3
1.2 Design and evaluation methodology for error detection techniques 4
1.2.1 In situ error detection circuits 4
1.2.2 FPGA-based transient simulator 5
1.3 Confidence-driven computing architecture 5
1.4 Low-power and error-resilient system with algorithm and architecture co-design 5
1.5 Dissertation outline 6

II. Timing Faults and Data-Retention Failures 7

2.1 Statistical circuit simulation methodology 7
2.2 Data-retention failure and Vmin analysis 10
2.3 Hold-time violation and Vmin analysis 13
2.4 Vmin of sequential logic circuits 17
2.5 Summary 19

III. Heavy-Ion Induced Soft Error Characterization 20

3.1 Test chip designs and test setup 21

3.1.1 Test chip 1 21
3.1.2 Test chip 2 24
3.1.3 Ion beam testing 25
3.2 Circuit design considerations 26
3.2.1 Radiation hardening by redundancy 27
3.2.2 Increasing critical charge by upsizing 28
3.2.3 Depth and sizing of combinational circuits 29
3.2.4 Data dependence 29
3.2.5 Latchup and total ionization dose 31
3.3 Voltage and frequency dependence 32
3.3.1 Supply voltage scaling 32
3.3.2 Clock frequency effects 33
3.4 Summary 35

IV. Error-Detection Circuits and Simulation Methodology 36

4.1 Transient error detection circuits 37
4.2 Flexible cross-edge checking 38
4.2.1 Cross-edge circuitry 40
4.2.2 Pre-edge circuitry 43
4.2.3 Comparisons 43
4.3 FPGA-based transient error simulator 44
4.4 Transient error simulator 45
4.4.1 Delay model and error model 46
4.4.2 Transient simulation 48
4.5 Evaluation of error-resilient designs 50
4.5.1 Pre-edge technique 51
4.5.2 Post-edge technique 52
4.6 Summary 54

V. Confidence-Driven Architecture for Error-Resilient Computing 55

5.1 Related work 55
5.2 Confidence-driven computing 57
5.2.1 Lockstep synchronization 58
5.2.2 Speculative synchronization 59
5.2.3 Spatial redundancy 61
5.3 Reliability and performance evaluation using sample-based emulation 61
5.3.1 Reliability evaluation 62
5.3.2 Performance evaluation 63
5.3.3 Implementation results and case study 67
5.4 Early checking 68
5.4.1 Early checking 68
5.4.2 Functional analysis and comparison 69

5.5 Reliability and performance evaluation using event-based simulation . . . 70
5.5.1 Error simulation . . . 71
5.5.2 Experimental evaluation . . . 71
5.5.3 Implementation results and case study . . . 75
5.6 Summary . . . 76

VI. Low-Power Error-Resilient Systems for Wireless Communication . . . 78

6.1 Background . . . 78
6.1.1 MIMO system model . . . 79
6.1.2 Non-iterative MIMO detection algorithms . . . 80
6.1.3 Iterative MIMO detection algorithms . . . 82
6.2 Iterative detection-decoding design overview . . . 83
6.3 SISO MMSE algorithm and optimization . . . 85
6.3.1 Reduced-complexity MMSE-PIC algorithm . . . 86
6.3.2 Algorithmic optimizations for efficient MMSE detector implementation . . . 87
6.4 Architecture and circuit optimization . . . 90
6.5 Nonbinary interface and optimization . . . 97
6.5.1 Detector to decoder: LLR computation with L1-norm 97
6.5.2 Decoder to detector: Efficient soft-symbol and variance computation . . . 100
6.6 Measurements . . . 104
6.7 Conclusion . . . 105

VII. Conclusion and Outlook . . . 109

7.1 Conclusion . . . 109
7.2 Outlook . . . 110

BIBLIOGRAPHY . . . 112

LIST OF FIGURES

Figure

2.1 Master-slave flip-flop (MSFF) schematic. 8

2.2 Probability density distributions from MC simulations with 10^4 samples for (a) normalized hold time at Vnorm and (b) normalized data-retention Vmin with Rg. The MC (a) hold time and (b) data-retention Vmin corresponding to a 3σ WID-variation probability are compared to MPP results. 8

2.3 Data-retention analysis for the slave latch of the MSFF while holding a logic 1. The PMOS process variation (circled) and gate-dielectric soft breakdown (Rg) on node S_I limit the data-retention Vmin. . . 10

2.4 (a) Butterfly curves for the static DC analysis of data retention. (b) Description of the retention windows for the dynamic transient analysis of data retention. 12

2.5 Normalized data-retention Vmin with the individual and combined contributions of WID variation and Rg for the conventional static DC analysis and the dynamic transient analysis that captures the cycle time effect. 12

2.6 Hold time is defined as the CK-to-D delay where a 50% Vcc glitch occurs on node M_O. 13

2.7 Dynamic transient simulation description for the worst-case data-dependent hold-time analysis. 14

2.8 Hold time versus normalized Vcc with different placement of Rg while not including WID variations. 15

2.9 Normalized hold time with and without Rg for a 5σ WID-variation target and an FO4 inverter chain delay versus the normalized Vcc. . 15

2.10 Hold time as a percentage of cycle time versus normalized Vcc. Hold-time Vmin is defined as the Vcc in which the hold time exceeds 10% of the cycle time. 16

2.11 Breakdown of the 5σ WID variation across all the transistors in the MSFF from the MPP simulations. 17

2.12 Vmin for data retention and hold time versus different combinations of Rg and 5σ WID variation. 18

2.13 Hold-time Vmin for the original MSFF in Fig. 2.1 and for the MSFF with a $2\times$ larger first clock inverter. 18

3.1 Shift register chains implemented on the test chip 1. Each chain consists of 500 flip-flops. Chain 6 to 11 each has a varying number and size of inverters. 22

3.2 Three types of D flip-flops: (a) DICE flip-flop [1], (b) TMR flip-flop [2], and (c) unprotected standard D flip-flop. 23

3.3 (a) Test chip 1 layout, and (b) microphotograph. 24

3.4 Radiation test setup. 26

3.5 Automated testing of chip 2. 27

3.6 Cross section per bit with ion energy for standard, DICE and TMR flip-flops. 28

3.7 Cross section per bit after adding combinational circuits. 30

3.8 Latch design results in unequal upsets. 31

3.9 Cross section per bit of D flip-flop, DICE and TMR flip-flops at supply voltage of 1.0V and 0.7V. 32

3.10 Cross section per bit of standard and radiation-hardened correlator cores at supply voltage of 1.0V and 0.7V (clock frequency of 100MHz). 33

3.11 Cross section per bit of standard and radiation-hardened correlator cores at 100MHz and 500MHz (1.0V supply voltage). 34

4.1 (a) Timing margin allocated for static and dynamic variations, and (b) pre-edge and post-edge error checking window. 37

4.2 (a) Checking window positions depending on path criticality, (b) fixed post-edge checking window causing race conditions, and (c) fixed pre-edge checking window leading to performance degradation. 39

4.3 (a) Cross-edge checking design based on a conventional transmission gate flip-flop [3], and (b) error detector design. 40

4.4 Waveforms illustrating the operations of the cross-edge circuitry. . . 42

4.5 (a) Pre-edge checking design, and (b) transition detector design. . . 42

4.6 Waveforms illustrating the operations of the pre-edge circuitry. . . . 43

4.7 FPGA-based transient simulation platform. A multi-stage simulation can be constructed by cascading single stages with individual delay profiles and models. 45

4.8 Timing charts of the FPGA-based hardware emulation: (a) pipeline clock, (b) extracted nominal data path delay, (c) injected transient events (errors and noise), and (d) data transition as the result of (b) and (c). 46

4.9 Transient event generator. 47

4.10 Transient simulation state (a) withe the datapath delay profile and transient error model, and (b) the state with pre-edge error detection and recovery by stalling the pipeline. 49

4.11 Pre-edge error detection and recovery through pipeline stall [4]. . . . 50

4.12 Simulated delay distribution from extracted CORDIC processor. . . 50

4.13 Reliability improvement with pre-edge error detection and correction (CW: checking window, ED: error duration, T_{clk}: clock period). . . 51

4.14 Effective throughput using pre-edge error detection and correction (CW: checking window, ED: error duration, T_{clk}: clock period). . . 52

4.15 Post-edge error detection circuits [5]. 52

4.16 Post-edge error detection rate. 53

4.17 Effective throughput using post-edge error detection and correction. 53

5.1 Proposed confidence-driven computing model incorporating confidence estimators. 56

5.2 Combination of spatial and temporal redundancy for an efficient error-resilient computing system. 57

5.3 Block diagram of confidence estimator. 57

5.4 Flow chart of lockstep synchronization. 58

5.5 Flow chart of speculative synchronization. 59

5.6 FPGA-based system emulation platform for quantitative evaluation. 60

5.7 (a) Error injection mechanism, and (b) error generator. 60

5.8 System error rate based on FPGA emulation results (solid line) and extrapolation (dash line), and (b) average delay per computation as the confidence threshold is adjusted. 62

5.9 (a) System error rate and (b) average delay of the CORDIC processor using temporal redundancy only and DMR (assume a confidence threshold of 2). 64

5.10 Average delay per computation improves with more fine-grained placement of confidence estimators: (a) at confidence threshold = 2, and (b) at confidence threshold = 3 when only temporal redundancy is considered. 65

5.11 (a) System error rate and (b) average delay of the CORDIC processor using lockstep and speculative synchronization. 66

5.12 Normalized energy overhead of three schemes for a highly reliable general purpose processor that requires a MTBF of 2 years. 67

5.13 Conceptual timing diagram illustrating early checking. 69

5.14 Timing charts of the FPGA-based event simulation: (a) main clock, (b) sampling clock, and (c) error generation. 72

5.15 System error rate without any error protection (error duration T_{err} is normalized to $T_{\texttt{clk}}$). 73

5.16 System error rate as the confidence threshold is adjusted (assume a transient or soft error rate of 10^{-5} and $T_{\texttt{sclk}} = 0.05T_{\texttt{clk}}$. Error duration T_{err} is normalized to $T_{\texttt{clk}}$). FPGA simulation results (solid line) and extrapolation (dotted line). 73

5.17 System error rate versus confidence threshold (assume a transient (or soft) error rate of 10^{-5}). FPGA simulation results (solid line) and extrapolation (dotted line). 74

5.18 Normalized throughput versus confidence threshold (assume a transient (or soft) error rate of 10^{-5}). 74

5.19 Adjustment of confidence threshold and sampling clock period to deliver the required system error rate of 10^{-20}. 75

6.1 MIMO system overview 79

6.2 Conventional MIMO detection algorithm (hard outputs) BER performance tradeoffs 80

6.3 MMSE block diagram with iterative detection-decoding design. . . . 82

6.4 PER comparison between an MMSE-LDPC IDD design and MMSE-NBLDPC GF(16) IDD design under a 4x4 256-QAM MIMO system in the TGn type C channel model. The code length is 640 bits with 1/2 code rate for each decoder. Each 256-QAM symbol is mapped to two 4-bit symbols for GF(16) decoding. 83

6.5 PER comparison for MMSE-NBLDPC GF(256) under a 4x4 256-QAM MIMO system in the TGn type C channel model. The code length is 1280 bits with 1/2 code rate for each decoder. Each 256-QAM symbol is mapped to an 8-bit symbol for GF(256) decoding. NBLDPC decoders are implemented based on EMS with n_m=32 and n_m=16. 84

6.6 Soft information exchanged in MMSE-LDPC IDD design: a detector computes input soft symbols $\overleftarrow{s}$ and variances $\overleftarrow{\sigma}^2$ based on a-posteriori LLR delivered from a decoder, and generates a-prior LLR for the decoder from its output soft symbols $\overrightarrow{s}$ and variances $\overrightarrow{\sigma}^2$. 85

6.7 Simplified computation for the bias term required for the output SNRs and output soft symbols. 89

6.8 Design of the MMSE detector in 4 task-based coarse pipeline stages. Stage 2 and 3 operate at a 2X slower clock frequency, and the remaining stages operate at the base clock frequency. 91

6.9 Lower-upper decomposition for an $N_t \times N_t$ matrix $\mathbf{A}$. 91

6.10 Implementation of LUD and forward substitution in stage 2. Left figure indicates the data dependency throughout the algorithm, and highlights the critical latency; right figure annotates the cycles when the result is completed. 92

6.11 Design of reciprocal based on two LUTs. 93

6.12 Least significant bit (LSB) error from the proposed reciprocal design. 94

6.13 Optimization of the critical paths stage 2 and stage 3 in the MMSE detector. Performance-optimized single path is the baseline design. . 95

6.14 Backward substitution and MMSE filtering in stage 3. Left figure shows the data hazards from 2-cycle complex multipilers and the structural hazard from 2 processing units; right figure annotates the cycles when the entry in matrix A^{-1} is completed. MMSE filtering is processed sequentially but is parallelly to the backward substitution. 96

6.15 Data switching activity factors in MMSE detector. 97

6.16 Bit LLR computation (before SNR scaling) for the soft detector output $\vec{s_1}$ in 256-QAM constellations. 98

6.17 Symbol LLR computation (before SNR scaling) for the soft detector output $\vec{s_1}$ in 256-QAM constellation. (Note that the x and y entries are cross added to obtain the symbol LLRs for a GF(256) decoder.) 99

6.18 Computation flow for input soft symbols and variances in (a) binary scheme, and (b) proposed nonbinary scheme. 100

6.19 PER comparison among three cases: (1) 16 GF elements for decoding and 16 GF elements for soft symbol estimation, (2) 32 GF elements for decoding and 32 GF elements soft symbol estimation, and (3) 16 GF elements for decoding and only top 2 GF elements for soft symbol estimation. 101

6.20 PER performance comparison of MMSE-NBLDPC GF(256) IDD system with n_m=12 and 412-b code length. 105

6.21 Microphotograph of the MMSE-NBLDPC iterative detector-decoder chip in 65nm CMOS. 107

6.22 Measured throughput and energy consumption of the MMSE detector and the NBLDPC decoder. The throughput is measured at the maximum clock frequency for each supply voltage. 108

LIST OF TABLES

Table

2.1 Comparison of MPP and MC simulations. Table shows percentage difference between MPP and MC (10^4samples) simulations for both hold time and data-retention Vmin, at two WID variation targets (2.5σ and 3.0σ). Hold time is performed at two voltages and data-retention Vmin is performed with and without Rg. 9

2.2 Normalized data-retention Vmin with Rg and a 5σ WID-variation target across various retention windows for the dynamic transient simulations. 12

3.1 Ions applied in radiation testing and their nominal LET 25

4.1 Comparison of in situ error detection techniques 37

4.2 Implementation complexity of in situ error detection techniques . . 44

4.3 Comparison of simulators 46

5.1 Area and energy of CDC with temporal redundancy only (CE: confidence estimator, CT: confidence threshold 66

5.2 Area and energy of CDC with DMR 66

5.3 Comparisons of error detection techniques with short checking durations 70

5.4 Normalized throughput of early checking in error-free operations . . 72

5.5 Normalized energy of system based on early checking 75

6.1 MIMO detection algorithm (hard outputs) design tradeoffs 81

6.2 SISO MIMO detection algorithm tradeoffs 82

6.3 MIMO detection algorithm (hard outputs) design tradeoffs 104

6.4 Comparison with state-of-the-art MIMO detector. 106

ABSTRACT

Low-Power and Error-Resilient VLSI Circuits and Systems
by
Chia-Hsiang Chen

Chair: Zhengya Zhang

Efficient low-power computation is critically important for the success of the next-generation very-large-scale integration (VLSI) applications. Device and supply voltage (Vcc) have been continuously pushed smaller and lower to meet a more constrained power envelope with each generation, but device and voltage scaling have created resiliency challenges, including increasing timing faults, data-retention failures, and soft errors. This work aims at designing low-power and robust circuits and systems by drawing circuit, architecture, and algorithm approaches. Throughout the work, new simulation methodologies are developed for fast and accurate analysis and evaluation.

This dissertation explores the low-voltage limits of sequential logics. The minimum supply voltage (Vmin) for sequential logic circuits is analyzed by statistically simulating the impact of within-die process variations and gate-dielectric soft breakdown on data retention and hold time. As Vcc scales, statistical circuit simulations demonstrate that hold time increases faster than circuit delay or cycle time, consequently the required number of minimum-delay buffers increases. For this reason, we formulate a new hold-time violation metric that defines Vmin as the Vcc in which the hold time exceeds a target percentage of the cycle time. The hold-time Vmin metric is adopted by Intel as a new design metric.

Device scaling increases soft errors, affecting circuit reliability. Two 65nm bulk CMOS test chips were characterized in heavy-ion radiation environment in Texas A&M University K500 superconducting cyclotron facility in 2012. Through extensive soft error characterization, we observe the soft error mechanisms and its dependence

on supply voltage and clock frequency. The study laid the foundation of the first 65nm digital signal processing (DSP) chip design for a NASA spaceflight project.

To detect timing faults under low Vcc or soft errors due to particle strikes, we propose a new online circuit technique to improve upon the conventional pre-edge and post-edge error detection techniques. This new error detection technique can be used to protect combinational logic paths while minimizing the performance penalty and implementation cost. In particular, a cross-edge technique is developed and its special pre-edge and post-edge versions for fast, moderate, and slow paths, respectively. This technique utilizes the inherent redundancy in a flip-flop design, thereby keeping the cost to only 31 transistors. Furthermore, the error checking window can be tuned by duty cycling the clock signal, offering more flexibility for diverse levels of protection.

As error-resilient designs have become more important with the continued device scaling, a critical challenge of designing error-resilient circuits and systems is the lack of tools to quickly and accurately evaluate their effectiveness and performance. We demonstrate an FPGA-based transient error simulator to accelerate transient error simulations incorporating accurate datapath delay models and realistic error models. Compared to conventional FPGA-based digital error simulators, the FPGA-based transient error simulator captures fine-grained interactions between errors and datapath. This error simulator is constructed based on parameterized models, making it general-purpose and widely applicable. We demonstrate the capability of this simulator in the evaluation of pre-edge and post-edge error detection techniques, using synthesized processors that operate under soft error, transient noise and voltage droop models.

In addition to timing faults and soft errors, future post-CMOS devices could introduce nondeterministic errors. We present a confidence-driven computing (CDC) architecture to provide an adaptive protection for post-CMOS devices. The CDC model employs fine-grained temporal redundancy and confidence checking for a faster adaptation and tunable reliability. The CDC model can be extended to deeply scaled CMOS circuits, where an early checking (EC) technique can be used to perform independent error checking for more flexibility and better performance. A sample-based FPGA emulation along with real-time error injection is proposed to evaluate the CDC model. The CDC model is shown to adapt to fluctuating error rates to enhance the system reliability by effectively trading off performance. FPGA-based transient error simulator is applied to evaluate the EC technique at a finer time scale. The EC technique improves the system reliability by more than four orders of magnitude when errors are of short duration.

Designing low-power resilient systems can effectively leverage application-specific algorithmic approaches. To explore design opportunities in the algorithmic domain, an application-specific detection and decoding processor for multiple-input multiple-output (MIMO) wireless communication is investigated. A reliable and low-power implementation of the MIMO processor is critical for future mobile communication systems. We propose an implementation of a joint detection and decoding technique by enclosing detection and decoding in an iterative loop to enhance both interference cancellation and error reduction. We present new co-optimizations through algorithm, architecture, and low-power circuit techiniques. These optimizations enable our chips to achieve a better throughput, energy efficiency and error performance from the previous work.

This dissertation studies the low-power circuit design issues, and proposes new circuit, architecture, and algorithm solutions to improve the system reliability and to enhance the energy efficiency. The new techniques developed in this work, including hold-time Vmin metric, FPGA-based transient simulator, confidence-driven architecture, and iterative detection-decoding system, provide the opportunities to optimize low-power and error-resilient VLSI circuits and systems.

CHAPTER I

Introduction

Scaling of CMOS device geometries and supply voltage (Vcc) significantly improves the energy efficiency of digital integrated circuits, but also leads to the growing challenges in system reliability and variability [6, 7]. As Vcc reduces, setup and hold times lengthen and data retention degrades. Within-die (WID) process variations and gate-dielectric soft breakdown can even further exacerbate the adverse effect, creating more timing faults and data-retention failures. Meanwhile, as charge reduces inside each logic node, CMOS transistors are becoming more susceptible to external transient noise and particle strikes [8, 9], increasing soft errors and presenting a major design problem especially for space and flight systems.

In parallel with CMOS device scaling, a variety of nanodevices, such as carbon nanotube, graphene, spin, nanoelectromechanical relay, and memristor, have been proposed to sustain Moore's law of scaling for years to come [10, 11]. Although these post-CMOS devices boast potentially much higher integration density and substantially lower energy consumption compared to CMOS, some of them exhibit nondeterministic behavior. For example, memristor switching is a stochastic process that depends on the probabilistic filament formation [12, 13]. Such stochastic devices yield unpredictable operations that are manifested as erratic errors of arbitrary duration.

Since sustaining the scaling and applying new devices are critically important to reduce energy consumption, improving system resiliency is one of the primary goals and challenges in today's very-large-scale integration (VLSI) circuits and systems.

This dissertation is focused on the design of error resilient circuits and systems, and it contributes to four research areas: (1) transistor-level study of the timing faults under low supply voltage and soft errors due to particle strikes, (2) low-cost circuit-level error detection and evaluation methodology, (3) error-resilient computing architecture, and (4) error-resilient algorithm and low-power system implementation

for multiple-antenna wireless communications. The four areas from transistor level to system level are elaborated in the following subsections.

1.1 Error characterization and impact analysis

1.1.1 Data-retention failures and timing faults

As Vcc is reduced, the storage elements, such as static random-access memory (SRAM) and register file, may have internal data corrupted and more errors may also occur during read and write operation. Traditionally, we consider that SRAM and register file limit the minimum supply voltage (Vmin) scaling due to read, write, or data-retention failures [14, 15]. Recent circuit-assist techniques [16, 17] and multiple Vcc power domains [18] have improved Vmin for both SRAM and register file designs. Looking forward, sequential circuits (i.e., flip-flops and latches), which contain feedback circuitry for storing data similar to SRAMs and register files, may start to limit Vmin scaling for the logic portion of the processor design. Since sequential logic circuits do not have the regularity of array structures, the circuit-assist techniques for SRAM and register file designs may not be applicable or incur impracticable overheads for the sequential circuits.

Three fundamental metrics for a sequential circuit are data retention, setup time, and hold time. As Vcc reduces with the presence of process variations and gate-dielectric soft breakdown, the data retention degrades, and the setup and hold times lengthen. Although setup-time violations at low Vcc can be avoided by reducing the clock frequency (Fclk) [6] in post-silicon testing, hold-time or data-retention failures cannot be resolved by changing Fclk. Rather, post-silicon data-retention or hold-time violations are only resolved by increasing Vcc. Furthermore, adding min-delay buffers and upsizing transistors in the sequential circuit are necessary to prevent potential data-retention and hold-time failures at low Vcc. These approaches, however, incur an expensive power overhead when worst-case variations are assumed in pre-silicon design. For this reason, we consider the data retention and hold time as the determining factor for the Vmin of sequential circuits.

As hold time increases faster than the cycle time while lowering Vcc, we introduce a new hold-time violation metric to define Vmin as the Vcc in which the hold time exceeds a target percentage (10%) of the cycle time [19]. As a result, the hold-time Vmin is found at 0.73Vnorm and is primarily affected by within-die (WID) variations in Intel 22nm tri-gate CMOS technology. Our circuit analysis show that the data-retention Vmin is highly sensitive to the gate-dielectric soft breakdown and

the variations on the latch. Hold-time Vmin is most sensitive to the variations on the first clock inverter. Upsizing the first clock inverter in the master-slave flip-flop (MSFF) by 2× can effectively reduce hold-time failures and improve Vmin.

1.1.2 Soft errors

Soft errors are nondestructive, nonpermanent, and nonrecurring errors. They were first observed in DRAM due to alpha particles emitted by lead-based package in the 70s [20]. Neutrons in cosmic rays were found as another important source of soft errors [21–23]. These energetic particles travel through the silicon substrate and create minority carriers. When enough minority carriers are collected by a nearby transistors drain diffusion node, it will result in a potential disruption of the stored 0 or 1 state, or a voltage transient, resulting in soft errors [24–26]. Soft errors belong to the broader class of single-event effects (SEE), defined as any measurable or observable change in state of performance of a device resulting from a single energetic particle strike [27]. Soft errors include single-event upset (SEU), i.e., a soft error caused by a single energetic particle strike [24, 25], and single-event transient (SET), i.e., a momentary voltage spike at a circuit node caused by a single energy particle strike [28].

To evaluate single-event effects, we designed two 65nm bulk CMOS application-specific integrated circuits (ASIC) test chips and characterized them in heavy-ion radiation environment in Texas A&M University K500 superconducting cyclotron facility in 2012. We observe that hardened flip-flops become less effective under heavy-ion testing, and approaches that increase the critical charge by upsizing and increasing supply voltage are also not as effective. The results suggest that the charge conveyed by heavy ion strikes has far exceeded the critical charge needed to cause an upset, thus tuning of the critical charge has little effect. However, multiple bit upsets are more likely with heavier ion strikes, causing radiation-hardened flip-flops to fail more frequently. We also observe the unequal 0-to-1 and 1-to-0 upsets in all shift register tests, which are attributed to the latch design, sizing, and the cross section of P- and N-diffusion. Circuit design and sizing can be used to balance or bias the upset rates for specific applications. The dependence on supply voltage and clock frequency are demonstrated in the dissertation.

1.2 Design and evaluation methodology for error detection techniques

Transient faults and soft errors discussed in section 1.1 often last for a short duration. The impact on the system varies depending on when and where the error occurs, known as timing masking and logic masking, respectively. Sometimes these errors do no harm to a system [29], while at other times they can propagate and accumulate, causing system failures. These problems can become more complicated, and therefore require new research on error detection techniques and efficient evaluation methods.

1.2.1 In situ error detection circuits

To enhance the robustness of deep-submicron designs against occasional delay errors and soft errors, online circuit techniques have been proposed to detect error occurrences. These techniques can be classified to three groups based on how the checking is performed: post-edge checking that detects error in a window after the sampling edge [5, 30–32], pre-edge checking that detects error in a window prior to the sampling edge [4, 33], and multi-edge checking that detects errors by upsampling using multiple clock phases [34, 35]. However, each existing technique has its own advantage and limitation. The post-edge technique incurs no performance penalty as checking occurs after the sampling edge, but each path under post-edge protection must be carefully tuned to avoid race conditions. The pre-edge technique is free of any hold time constraints, and it can use the delay slack in fast paths for error detection, however it prolongs the clock period if it is used in slow (critical) paths. Note that none of the above techniques alone is well suited to providing coverage of all types of datapaths.

To overcome these challenges, we propose a diverse error detection technique to protect combinational logic paths while minimizing the performance penalty and implementation cost [36]. In particular, we use a new cross-edge technique and its special pre-edge and post-edge versions for moderate, fast, and slow paths, respectively. Our method utilizes the inherent redundancy in a flip-flop design, thereby keeping the cost at only 31 transistors. Furthermore, it can also be tuned by duty cycling the clock signal, offering more flexibility for diverse levels of protection.

1.2.2 FPGA-based transient simulator

Error-resilient circuit designs require massive verification and simulation to find the optimal design parameters. Traditionally, the simulation is performed by using either slow software-based transient simulator or inaccurate field-programmable gate array(FPGA)-based digital simulator. The challenges with slow or inaccurate simulation process motivate us to design an FPGA-based transient error simulator to aid the design and evaluation of error-resilient techniques. In the proposed paradigm, we detach circuit characterization from error simulation: circuit characterization will be done efficiently using standard computer-aided design (CAD) tools, while the lengthy error simulation will be done on a fast FPGA-based error simulator. Fast FPGA platforms allow the delay models and error models to be fully exercised, and the interactions between errors and circuits to be captured for a good coverage. The proposed FPGA-based transient simulator is comprised of three main parts: (1) delay profile of the datapath under test, (2) transient error models, and (3) error-resilient designs to be evaluated. All three parts are programmable, producing a versatile and general-purpose error simulation platform.

1.3 Confidence-driven computing architecture

We propose a confidence-driven computing (CDC) architecture for protection against nondeterministic errors over a wide range of rate and duration. The key concept of the proposed computing model is to employ fine-grained temporal redundancy with tunable threshold for a faster adaptation and an adjustable reliability. The CDC model is suitable for designs using nondeterministic post-CMOS devices. It allows systems to adapt to large runtime variations and reduces excessive design margins for an efficient computing system [37].

1.4 Low-power and error-resilient system with algorithm and architecture co-design

Algorithm techniques provide more opportunities for designing a low-power and error-resilient system. Targeting a reliable detection and decoding processor design for multiple-input multiple-output (MIMO) wireless communication system, we explore design methods from algorithm and architecture to circuits. In state-of-the-art MIMO designs, iterative detection and decoding (IDD) has been proposed to reduce the signal-to-noise ratio required for a reliable transmission. An IDD system consists of a

soft-in soft-out (SISO) detector to cancel interference, and a forward error correction (FEC) decoder to remove errors. The two blocks exchange soft information to improve the SNR iteratively. In this dissertation, we demonstrate an IDD design in 65nm CMOS technology with a minimum mean square error (MMSE) detection and a nonbinary low-density parity-check (LDPC) decoding targeting 4x4 256 quadrature amplitude modulation (QAM) MIMO systems. Through study of the algorithmic properties, and the co-optimization algorithm with architecture and circuit designs, our MMSE detector achieves the highest throughput and energy efficiency among the latest published SISO detector designs.

1.5 Dissertation outline

Chapter II describes the study of low-voltage limits of sequential gates. In addition to conventional data-retention Vmin metric, we propose a new hold-time Vmin metric. This study was conducted with an Intel 22nm tri-gate CMOS technology.

Chapter III presents extensive soft error characterization results from our two 65nm bulk CMOS test chips that underwent heavy-ion radiation testing. The soft error mechanisms and their dependence on supply voltage and clock frequency are analyzed.

Chapter IV discusses error detection circuits and simulation methodologies to evaluate error resilient designs. We present an efficient in situ error detection techniques to exploit datapath characteristics for monitoring circuit errors in a more efficient manner. A new simulation tool is constructed based on FPGA to quickly and accurately evaluate error-resilient designs and their effectiveness and performance. We demonstrate the capability of this simulator in the evaluation of error-resilient circuit design techniques.

Chapter V explores a confidence-driven computing (CDC) architecture and its adaptive protection against nondeterministic errors.

Chapter VI demonstrates an ASIC circuit design to improve communication reliability for a multiple-input multiple-output (MIMO) system. We presents new techniques that improve the energy efficiency and enable a higher throughput based on the co-optimization of the algorithm with the architecture and circuit design.

Chapter VII summarizes the contributions of this dissertation and describes the future directions .

CHAPTER II

Timing Faults and Data-Retention Failures

In this chapter, we explore the Vmin for sequential logic circuits in a 22nm tri-gate CMOS technology [38] by statistically simulating the impact of within-die (WID) process parameter variations and gate-dielectric soft breakdown on data retention and hold time for over 10^6 standard-cell master-slave flip-flops (MSFF), as illustrated in Figure 2.1, to represent the sequential circuits in a high-performance microprocessor or SoC design. Section 2.1 describes the statistical circuit analysis. Sections 2.2 and 2.3 explain the circuit simulation methodologies for the data-retention Vmin and hold-time Vmin, respectively. Section 2.4 compares the data-retention Vmin and the hold-time Vmin values while providing insight for reducing the overall Vmin for the sequential logic circuits. Section 2.5 summarizes the key results.

2.1 Statistical circuit simulation methodology

The Monte Carlo (MC) simulation is the most common statistical methodology for capturing the effects of process variations in circuits. The MC simulation performs many MC samples. Based on the input device-level parameter distributions and spatial correlations, each MC sample assigns variations to the device parameters in the circuit. These device-level parameters include channel length, channel width, and threshold voltage. After simulating the circuit with the assigned parameter variations, the circuit output (e.g., delay) corresponds to one MC sample. The MC output distribution is generated after performing a sufficient number of samples.

In a high-performance microprocessor or DSP design, the number of flip-flops can reach or exceed 10^6 [39]. Quantifying the impact of WID variations on this number of flip-flops in a design requires a statistical analysis corresponding to a cumulative probability of 5 standard deviations (σ) from the mean. To accurately capture the 5σ WID-variation probability in the tail of the MC output distribution, more than

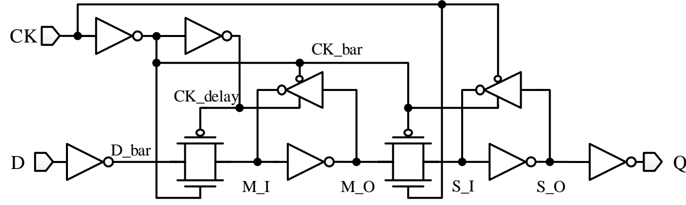
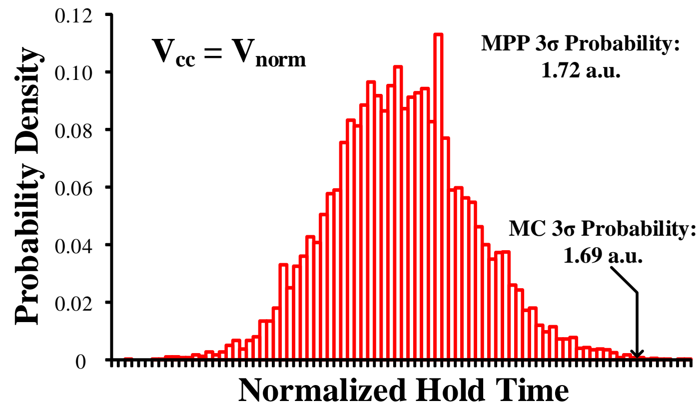


Figure 2.1: Master-slave flip-flop (MSFF) schematic.



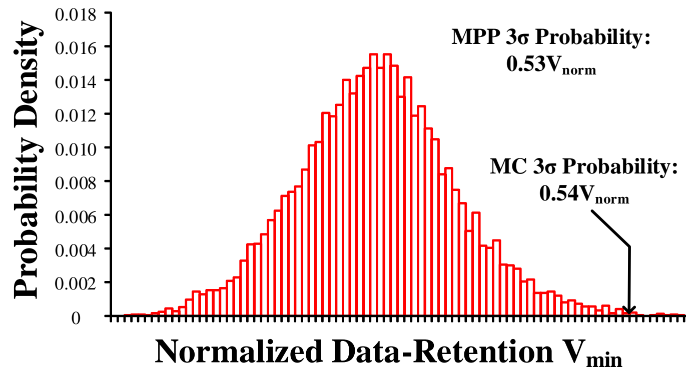


Figure 2.2: Probability density distributions from MC simulations with 10^4 samples for (a) normalized hold time at Vnorm and (b) normalized data-retention Vmin with Rg. The MC (a) hold time and (b) data-retention Vmin corresponding to a 3σ WID-variation probability are compared to MPP results.

10^7 samples are needed, resulting in excessive simulation time, and consequently, rendering the MC approach impracticable as a statistical analysis approach.

Table 2.1: Comparison of MPP and MC simulations. Table shows percentage difference between MPP and MC (10^4samples) simulations for both hold time and data-retention Vmin, at two WID variation targets (2.5σ and 3.0σ). Hold time is performed at two voltages and data-retention Vmin is performed with and without Rg.

	Hold time		Data-retention Vmin	
WID Variation	Vnorm	0.75Vnorm	w/ Rg	w/o Rg
2.5σ	2.1%	0.2%	3.6%	1.6%
3.0σ	1.8%	4.4%	1.9%	0.5%

The most probable point (MPP) simulation [40] provides an exponentially faster alternative to an MC simulation for cumulative probabilities larger than 4σ. In contrast to the MC approach, the MPP only generates a single output value that corresponds to a specific cumulative probability (e.g., a 5σ probability in a normal distribution). The MPP first performs a sensitivity analysis to identify the most sensitive parameters in the circuit that are susceptible to variations. Then, the WID-variations are distributed to either maximize or minimize the circuit response, depending on the output function, for an input cumulative probability corresponding to a target number of σ values (e.g., 5σ). As an example of the MPP hold-time simulation, the WID parameter variations in channel length, channel width, and threshold voltage are distributed among the most sensitive transistors in the MSFF, as described in Fig. 2.1, to maximize the hold-time delay for a cumulative probability corresponding to a 5σ target in a normal distribution. In an MC simulation, the required number of samples increases exponentially as the target σ number increases. In contrast, MPP only requires a fixed number of samples for the sensitivity analysis and for calculating the maximum or minimum circuit output, which depends on the number of transistors in the circuit and is independent of the target σ number. For this reason, MPP is a highly practical statistical simulation methodology for evaluating a circuit response for a target of 4σ or higher. Furthermore, the MPP simulation provides key insight to the most vulnerable transistors in the circuit by specifying the assignment of the device-level parameter variations.

Recent statistical SRAM and register file circuit simulations employ the MPP methodology [41]. Table 2.1 and Fig. 2.2 provide a comparison of the MPP and MC simulations for validating the accuracy of the MPP approach. The MC simulations consist of 10^4 samples to enable a highly accurate analysis of the distribution tail

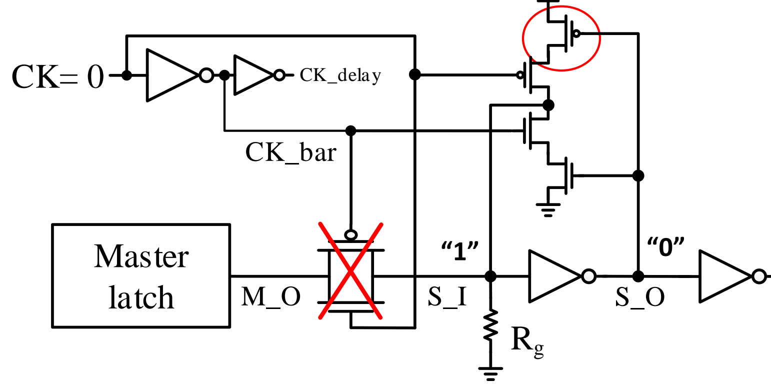


Figure 2.3: Data-retention analysis for the slave latch of the MSFF while holding a logic 1. The PMOS process variation (circled) and gate-dielectric soft breakdown (Rg) on node S_I limit the data-retention Vmin.

for cumulative probabilities corresponding to a 3σ target and below. As described in Sections 2.2 and 2.3, separate statistical circuit simulations quantify the hold time and data-retention Vmin for cumulative probabilities targeting a 2.5σ and a 3.0σ of WID variation. For the hold-time simulations, Vcc equals Vnorm and 0.75Vnorm, where Vnorm represents a normalized voltage for the process technology node. From Table 2.1, the MPP error as compared to MC is less than 5% for all four hold-time statistical simulations. Fig. 2.2(a) highlights one of the four comparisons by plotting the probability density from the MC simulation with Vcc at Vnorm. From Fig. 2.2(a), the MPP normalized delay of 1.72 agrees closely (i.e., 1.8% error) with the MC normalized delay of 1.69 for a cumulative probability corresponding to a 3σ WID-variation target. For the data-retention Vmin, simulations are performed with and without the Rg model that captures the gate-dielectric soft breakdown. In Table 2.1, the MPP error in data-retention Vmin is less than 4%. From Fig. 2.2(b), the MPP output error is 1.9% of the MC simulation value. In summary, the MPP methodology provides a highly accurate result as compared to an MC approach while exponentially reducing the simulation time for cumulative probabilities targeting 4σ and beyond.

2.2 Data-retention failure and Vmin analysis

As described in Fig. 2.1, an MSFF consists of a master latch followed by a slave latch. An MSFF retains data in both the master and slave latches. Since the data retention for the master latch and the slave latch are similar, the data-retention simulation focuses on the slave latch to simplify the analysis. While the number of

latches to consider for the data-retention analysis is twice the number of MSFFs, the change in the number of standard deviations for the WID variations is negligible. Fig. 2.3 zooms-in on the schematic of the slave latch for describing the data-retention analysis. The two primary sources of data-retention degradation are WID process variations and gate-dielectric soft breakdown [42]. Gate-dielectric soft breakdown is modeled by adding a resistor (Rg) between the gate and source of a transistor as illustrated in Fig. 2.3 at node S_I. The value of Rg is empirically extracted from device measurements. Although soft breakdown can occur in any transistor, the most probable node for soft breakdown in the slave latch is either S_I or S_O in Fig. 2.3. These two nodes receive the longest time of DC stress while the transistors on the clock path receive less DC stress because the clock nodes transition twice every cycle [43]. Thus, the data-retention analysis for the MSFF is similar to the SRAM [42]. The conventional SRAM data-retention analysis is based on a static DC simulation [44]. The conventional SRAM circuit analysis breaks the feedback inverter loop to simulate the DC response for the voltage transfer curve (VTC). Similarly for the MSFF, the simulation varies the input voltage from 0V to Vcc on S_I to generate one VTC and S_O to generate another VTC as illustrated in Fig. 2.4(a). From this butterfly curve which is formed by the two VTCs, the static noise margin (SNM) is calculated as the voltage corresponding to the smallest side of the two largest squares bounded inside the curve. The data-retention Vmin is defined as the Vcc in which the SNM collapses to zero.

From MPP simulations, drive-current degradation in the top PMOS of the tri-state inverter, as circled in Fig. 2.3, with an Rg connected between S_I and ground limits the data-retention Vmin. This occurs for two reasons. First, the tri-state inverter is designed with minimum width transistors to minimize the impact on the CK-to-Q delay and area. In comparison to the S_O node which is driven by an inverter, the S_I node is weakly driven by the tri-state or stacked inverter, resulting in greater susceptibility to the gate leakage from soft breakdown as modeled by Rg. Second, the PMOS drive current is slightly weaker than the NMOS drive current for iso-sized transistors [38], thus retaining a logic 1 on S_I is more difficult than holding a logic 0. For these reasons, the worst-case simulations for soft breakdown occur while placing an Rg between S_I and ground while retaining a logic 1. In addition to the location of Rg, the inverter drive current is more sensitive to the top PMOS of the tri-state inverter, as circled in Fig. 2.3, as compared to the bottom PMOS

In contrast to an SRAM or register file design, the MSFF refreshes the data in both latches every cycle. Thus, the slave latch only needs to retain the data for

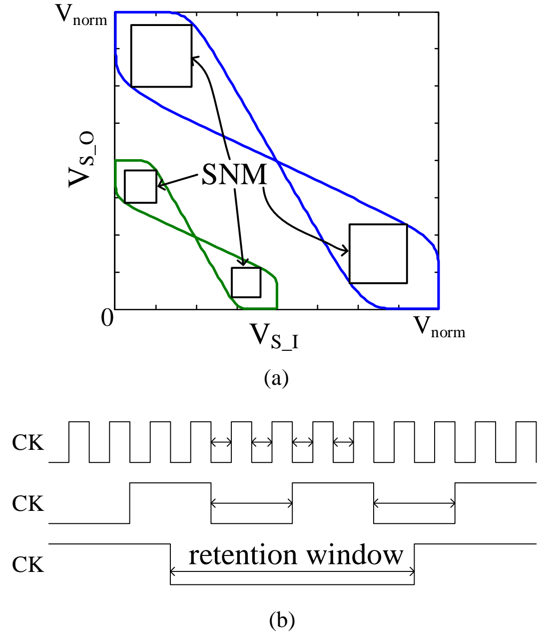


Figure 2.4: (a) Butterfly curves for the static DC analysis of data retention. (b) Description of the retention windows for the dynamic transient analysis of data retention.

Table 2.2: Normalized data-retention Vmin with Rg and a 5σ WID-variation target across various retention windows for the dynamic transient simulations.

Retention Window	0.01μs	0.1μs	1μs	10μs
Norm. Data-Retention Vmin	0.607	0.608	0.609	0.609

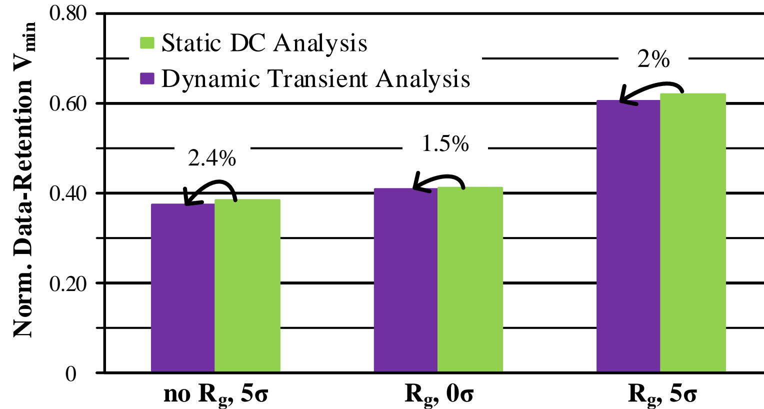


Figure 2.5: Normalized data-retention Vmin with the individual and combined contributions of WID variation and Rg for the conventional static DC analysis and the dynamic transient analysis that captures the cycle time effect.

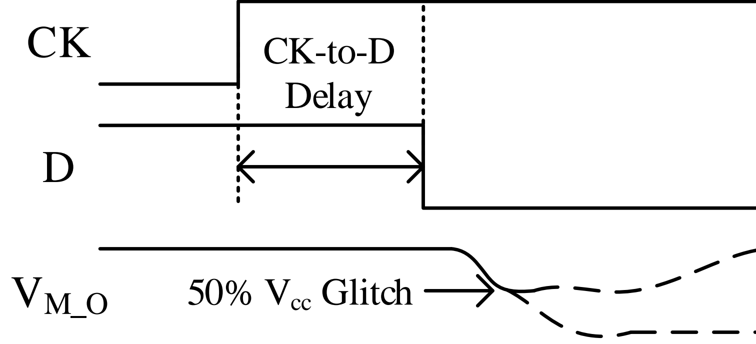


Figure 2.6: Hold time is defined as the CK-to-D delay where a 50% Vcc glitch occurs on node M_O.

half of the clock cycle (i.e., low phase of the clock). The retention time is inversely proportional to Fclk. The traditional static DC analysis assumes an infinite retention window, thus failing to capture the interaction between the data retention and the clock cycle time. To investigate the impact of cycle time (or retention window) on the data-retention Vmin, a transient simulation is performed while varying the retention window as illustrated in Fig. 2.4(b). Table 2.2 lists the normalized data-retention Vmin simulation results for retention windows ranging from 0.01μs to 10μs. From this data, the clock cycle time has a negligible influence on the data-retention Vmin over the cycle time range of interest.

Fig. 2.5 quantifies the individual and combined impact of WID process variations and gate-dielectric soft breakdown on data-retention Vmin for the static DC analysis and the dynamic transient analysis. First, the static DC analysis agrees closely (i.e., within 2%) with the more rigorous and accurate dynamic transient analysis. Since the static DC analysis is significantly faster than the dynamic transient analysis, the conventional static DC simulation is the recommended approach for the data-retention Vmin analysis in sequential logic circuits. Second, the results in Fig. 2.5 indicate that the WID variations at a 5σ target have a similar effect as the gate-dielectric soft breakdown on the data-retention Vmin. The combination of both WID variations and gate-dielectric soft breakdown limits the data-retention Vmin to 0.61Vnorm for the 22nm technology.

2.3 Hold-time violation and Vmin analysis

Referring to Fig. 2.1, hold time is the minimum delay that the MSFF input (D) needs to be held after the rising edge of the clock (CK) to ensure the data is sampled correctly. The hold-time simulation sweeps the transition of D relative to the

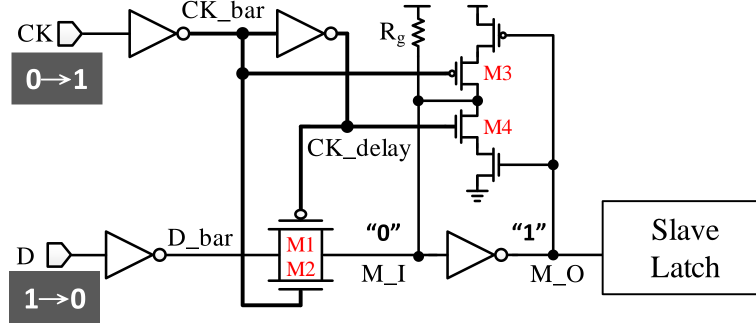


Figure 2.7: Dynamic transient simulation description for the worst-case data-dependent hold-time analysis.

rising CK edge until the CK-to-D delay results in a 50% Vcc glitch at node M_O as described in Fig. 2.6. Hold time is influenced by the same factors considered in the data-retention analysis, including gate-dielectric soft breakdown and WID process variations. Hold time is also data dependent, as the hold time for a logic 1 at the input D is longer than the hold time for a logic 0 for a positive-edge-triggered MSFF as illustrated in Fig. 2.7. This phenomenon is attributed to the misalignment of clock signals to the transmission gate (i.e., transistors M1 and M2) and the tri-state inverter (i.e., transistors M3 and M4). The internally generated CK_delay and CK_bar nodes are separated by an inverter delay. As a result, the transmission-gate PMOS (M1) is always turned off after the transmission-gate NMOS (M2), thus creating a longer transparency window for a logic 1 on D_bar (i.e., 0 on D) as compared to a logic 0 on D_bar (i.e., 1 on D). The longer transparency window directly increases the hold time to prevent the high-to-low transition on D from entering the master latch and corrupting the desired state. In parallel, the M4 NMOS turns on after the M3 PMOS, thus the pull down to maintain 0 at node M_I (i.e., the 1 from D) is weakened. Thus, the hold time for a logic 1 at the input D is significantly longer than the hold time for a logic 0.

Initial hold-time simulations consider the effect of gate-dielectric soft breakdown by placing the Rg at different nodes in the MSFF. The hold time for a logic 1 degrades by either inserting Rg between Vcc and node M_I or placing Rg between node M_O and ground. Fig. 2.8 compares these two scenarios by simulating the impact of Rg on hold time without considering WID variations. Fig. 2.8 demonstrates that the worst-case hold time for a logic 1 occurs for an Rg between Vcc and M_I.

Fig. 2.9 plots the impact of WID process variations for a 5σ target on the hold time with and without inserting Rg between Vcc and M_I. From Fig. 2.9, the impact of WID variations dominates the hold time as Vcc scales while the gate-dielectric soft

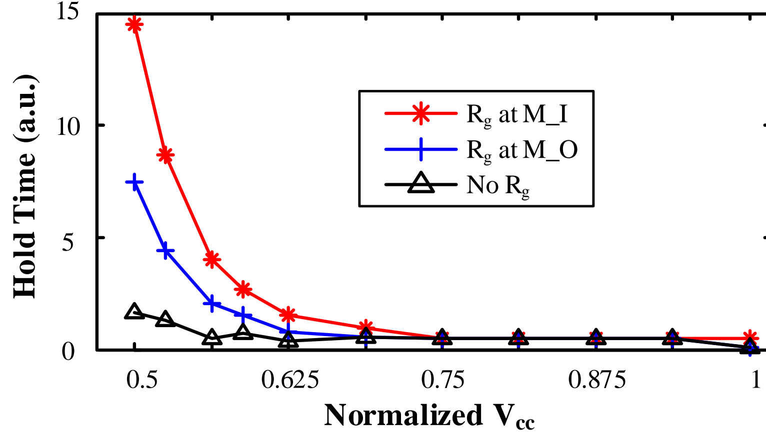


Figure 2.8: Hold time versus normalized Vcc with different placement of Rg while not including WID variations.

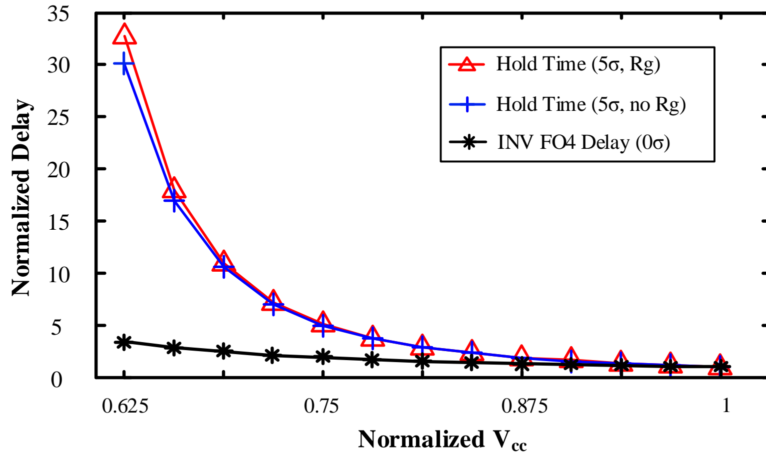


Figure 2.9: Normalized hold time with and without Rg for a 5σ WID-variation target and an FO4 inverter chain delay versus the normalized Vcc.

breakdown has a negligible effect for a 5σ WID-variation target. Fig. 2.9 also plots the normalized fan-out of 4 (FO4) inverter chain delay as a representative of logic path delay as Vcc reduces. From Fig. 2.9, the hold time increases at a much faster rate as compared to the FO4 inverter delay as Vcc decreases. As Vcc reduces from Vnorm to 0.625Vnorm, the normalized FO4 inverter chain delay increases by $3.3\times$ while hold time increases by more than $30\times$. This large discrepancy between the hold time and the inverter chain delay amplifies the susceptibility of sequential logic circuits to min-delay race conditions, thus limiting Vcc scaling.

To avoid the hold-time violations, logic circuit designs must insert additional buffers to allow further Vcc scaling, which negatively affects the logic area and power at the high-performance mode. A critical step for evaluating the hold-time Vmin

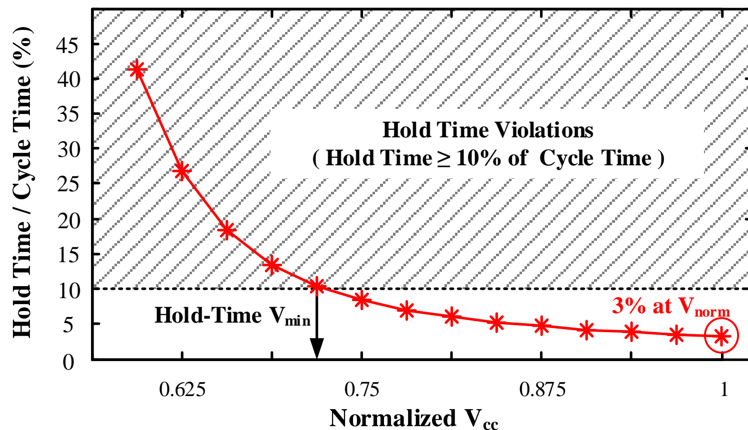


Figure 2.10: Hold time as a percentage of cycle time versus normalized Vcc. Hold-time Vmin is defined as the Vcc in which the hold time exceeds 10% of the cycle time.

is establishing the maximum hold-time delay for a given Vcc. In the simulations, hold time equals 3% of the cycle time at Vnorm. The normalized clock cycle time is assumed to scale as the inverter chain delay. A practical definition of hold-time Vmin is determined by normalizing the hold time to the cycle time at each Vcc value as plotted in Fig. 2.10. This data demonstrates that the hold time increases as a larger fraction of the available cycle time as Vcc reduces. Consequently, the number of buffers for min-delay protection must become an increasing fraction of the total cycle time. The increasing cost of buffer insertion diminishes the energy benefits of reducing Vcc. From Fig. 2.10, a practical limit for hold time is 10% of the cycle time. Beyond this point, the hold time increases exponentially, thus requiring an exponential increase in the number of min-delay buffers to avoid hold-time violations. By defining the maximum hold time as 10% of the cycle time, the hold-time Vmin equals 0.73Vnorm in the 22nm technology.

Fig. 2.11 describes the assignment of device-level parameter variations from the MPP simulation to maximize the hold time for a 5σ WID variation across four Vcc values. This data provides key insight to the most sensitive transistors in the MSFF for hold time. From Fig. 2.11, the MPP simulation places the vast majority of the WID variation on the NMOS of the first clock inverter of the MSFF (i.e., the inverter with input CK and output CK_bar in Fig. 2.1). As Vcc reduces, this NMOS transistor receives a larger portion of the WID variation. The variation in the NMOS of the first clock inverter changes the falling delay of CK_bar and the rising delay of CK_delay, which controls the NMOS and PMOS transistors in the master transmission gate, respectively. A longer channel length, shorter channel width, and/or higher threshold

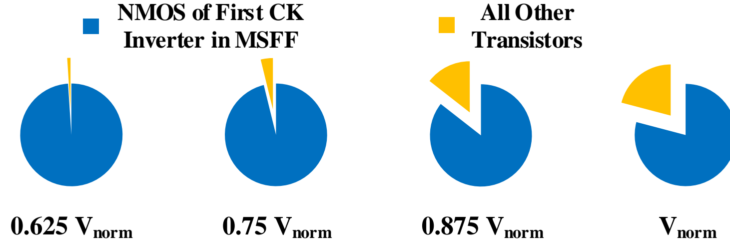


Figure 2.11: Breakdown of the 5σ WID variation across all the transistors in the MSFF from the MPP simulations.

voltage on the NMOS of the first clock inverter weakens the drive strength of this inverter during a rising clock edge, thus increasing the delays for both clock inverters in the MSFF. The longer delays for the two clock inverters expand the transparency window, thus degrading the hold time for the MSFF. In summary, the hold time is most sensitive to the NMOS of the first clock inverter in the MSFF.

2.4 Vmin of sequential logic circuits

Fig. 2.12 compares the data-retention and hold-time Vmin values while considering the individual and combined effects of WID variations and gate-dielectric soft breakdown. When neither WID variations nor gate-dielectric soft breakdown are considered, the data-retention and hold-time Vmin equals the fundamental Vcc scaling limit for CMOS circuits [45,46]. When only accounting for the 5σ WID variation, the data-retention and hold-time Vmin values increase to 0.39Vnorm and 0.72Vnorm, respectively. When only considering the gate-dielectric soft breakdown, the data-retention and hold-time Vmin values are 0.41Vnorm and 0.5Vnorm, respectively. As discussed previously in Sections 2.2 and 2.3, WID variation and gate-dielectric breakdown affect the data retention similarly while the WID variation dictates the hold time. When combining the effects of both WID variation and gate-dielectric soft breakdown, the data-retention and hold-time Vmin values rise to 0.61Vnorm and 0.73Vnorm, respectively. From this analysis, the hold-time Vmin limits the Vcc scaling for sequential circuits in a high-performance microprocessor or DSP in a 22nm technology.

Since the WID variation is the dominant contributor to the hold-time Vmin, reducing the hold-time sensitivity to WID variations enables an overall lower Vmin for the sequential circuits. As described in Section 2.3, the hold time is most sensitive to variations on the NMOS of the first clock inverter in the MSFF. Increasing the transistor width allows more averaging of the random uncorrelated WID variations,

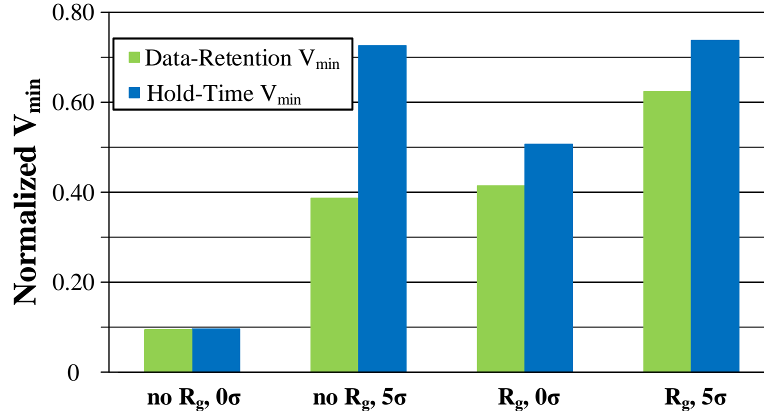


Figure 2.12: Vmin for data retention and hold time versus different combinations of Rg and 5σ WID variation.

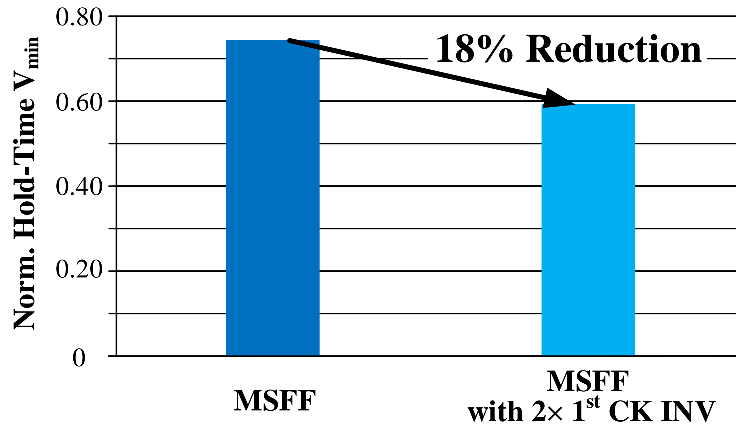


Figure 2.13: Hold-time Vmin for the original MSFF in Fig. 2.1 and for the MSFF with a 2× larger first clock inverter.

consequently reducing the drive current sensitivity to WID variations. Fig. 2.13 reveals an 18% reduction in hold-time Vmin by doubling the size of the first clock inverter. This design change in the MSFF results in an overall 16% Vmin reduction since the data-retention Vmin of 0.61Vnorm now limits the Vcc scaling. Although the larger clock inverter width increases the capacitive load on the clock network and the dynamic power at a given Vcc value, this analysis highlights the opportunity for optimizing the sequential circuit design for enhancing the energy efficiency of a high-performance microprocessor or DSP design.

2.5 Summary

Data-retention Vmin and hold-time Vmin are studied to avoid logic failures on sequential circuits while capturing the effect of WID process variations and gate-dielectric soft breakdown. Statistical circuit simulations demonstrate that the data-retention Vmin depends on both WID variations for a 5σ target and gate-dielectric soft breakdown, which limit the data-retention Vmin to 0.61Vnorm. As hold time increases faster than the cycle time while lowering Vcc, a new hold-time violation metric is introduced to define Vmin as the Vcc in which the hold time exceeds a target percentage (10%) of the cycle time. As a result, the hold-time Vmin is found at 0.73Vnorm and is primarily affected by WID variations. Furthermore, a detailed circuit analysis reveals that the data-retention Vmin is highly sensitive to the gate-dielectric soft breakdown and the variations on the top PMOS of the tri-state inverter. Hold-time Vmin is most sensitive to the variations on the NMOS of the first clock inverter. Upsizing the first clock inverter in the MSFF by $2\times$ reduces the hold-time Vmin by 18% and the overall Vmin by 16%.

CHAPTER III

Heavy-Ion Induced Soft Error Characterization

In this chapter, we present our heavy-ion radiation testing results and analyze the impacts on circuit designs. Heavy-ion radiation testings were carried out in March and August 2012 at the Texas A&M University K500 superconducting cyclotron facility [47]. Our measurements cover an array of heavy ions from neon to gold for chip 1 and from helium to silver for chip 2. The two test chips allow us to characterize SEE at both circuit and system levels with different supply voltages and clock frequencies. We observe that the upset rate saturates with increasing linear energy transfer (or LET, the amount of energy deposited as the ion travels through the material, measured in MeV-cm^2/mg). The results shed light on the effectiveness of radiation hardening approaches over a wide range of ion energies. In particular, conventional radiation-hardened designs that rely on redundancy or increasing critical charge by upsizing and increasing supply voltage are less effective at high LET levels. Our results also suggest that SEU in sequential circuits is dominant compared to SET in combinational circuits in 50MHz and 100MHz testing. Increasing the clock frequency to 500MHz increases upsets due to higher SET, which renders radiation hardening of sequential circuits alone ineffective. In summary, the technical contributions of this work are: (1) provide both circuit and system-level evidence to support earlier findings [48] that hardened flip-flops are becoming less effective at high LET, and clock frequency has a strong impact on the SET in combinational circuits [49]; (2) extend the study on supply voltage [49] by showing that the supply voltage does not have a strong impact in high-LET heavy-ion testing; (3) extend the study on clock frequency [49] by showing that the hardened designs can be more sensitive to clock frequency than standard designs; and (4) provide system-level ASIC testing results that are consistent with the findings from the circuit-level test structures.

3.1 Test chip designs and test setup

Two test chips were designed and fabricated in a TSMC 65nm bulk CMOS process. Chip 1 was dedicated to characterizing SEE on basic ASIC chip building blocks, including standard and hardened flip-flops, and to isolate the effects of supply voltage, sizing, combinational and sequential circuits. The testing was done at a 50MHz clock frequency to allow for probing of each individual test structure.

Chip 2 is a synthesized ASIC chip based on standard cells. The purpose of chip 2 is to characterize SEE of a practical ASIC chip at the system level. The testing was done at two clock frequencies, 100MHz and 500MHz, and two supply voltages, 1.0V and 0.7V, to study the effects of clock frequency and supply voltage on the SEE-induced upsets in a standard DSP core and a radiation-hardened DSP core. The results from both test chips are related to draw conclusions on the protection against heavy ion impact by redundancy and to identify voltage and frequency dependency.

3.1.1 Test chip 1

Test chip 1 measures 1.2mm×1.5mm in size, and it contains 11 independent shift register chains, each consisting of 500 stages of standard D flip flops as shown in Figure 3.1. Chain 1 and 2 are built using DICE [1, 50] and TMR flip-flops [2], respectively. DICE, shown in Figure 3.2(a), is a dual redundant flip-flop that prevents an upset on any one node from propagating and corrupting the stored bit. TMR flip-flop, shown in Figure 3.2(b), uses three copies of storage and majority vote to enhance the protection against any single upset. These hardened flip-flops are evaluated against commercial standard-cell flip-flops that make up chain 3, 4 and 5, where the flip-flops in chain 3 are minimum sized (DFF1X) and those in chain 4 and 5 are upsized using DFF2X and DFF4X cells respectively. Upsizing increases Q_{crit} but also increases charge collection area.

To investigate SET in combinational circuits, we insert inverters in chain 6 to 11. Specifically, in chain 6, 7 and 8, minimum sized inverters (INV1X) are used, and the combinational logic depth is varied by having 4 inverters per shift register stage in chain 6, 8 inverters in chain 7, and 16 inverters in chain 8. Deeper combinational logic increases the collection of SET. When an SET is propagated and sampled by a flip-flop, it is turned into an SEU. In chain 9, 10 and 11, upsized INV2X inverters are used. Upsizing combinational circuits increases Q_{crit}, but also increases SET collection. These test structures allow us to investigate SET in combinational circuits and SEU in sequential circuits, as well as the impact of sizing and depth of combinational

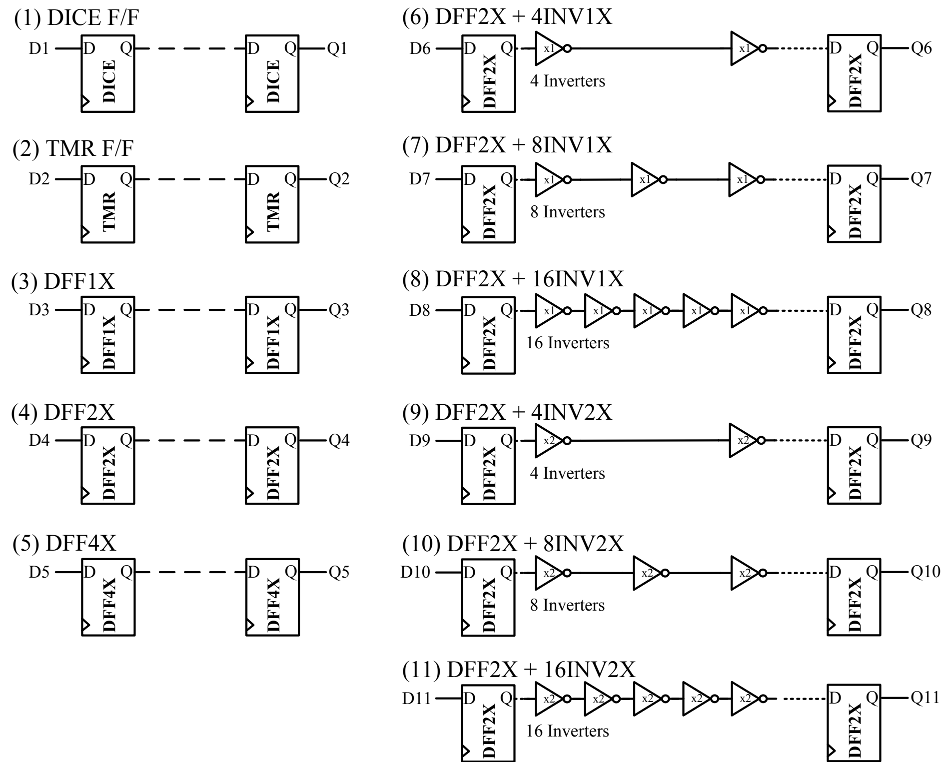


Figure 3.1: Shift register chains implemented on the test chip 1. Each chain consists of 500 flip-flops. Chain 6 to 11 each has a varying number and size of inverters.

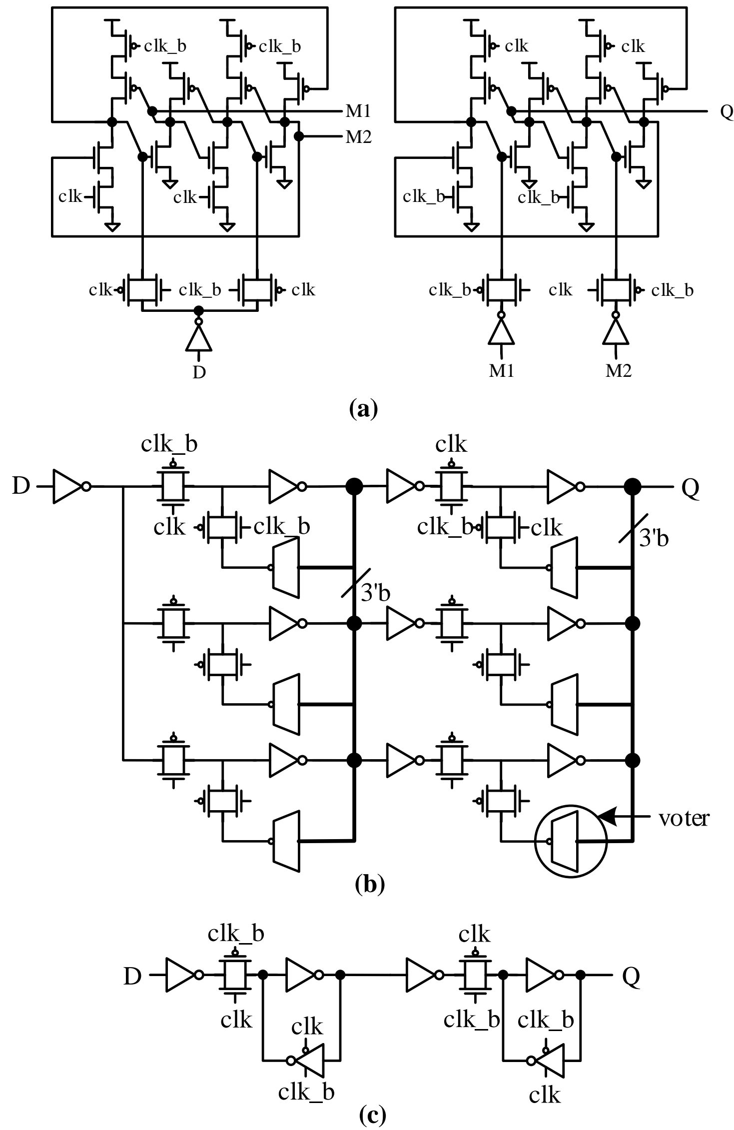


Figure 3.2: Three types of D flip-flops: (a) DICE flip-flop [1], (b) TMR flip-flop [2], and (c) unprotected standard D flip-flop.

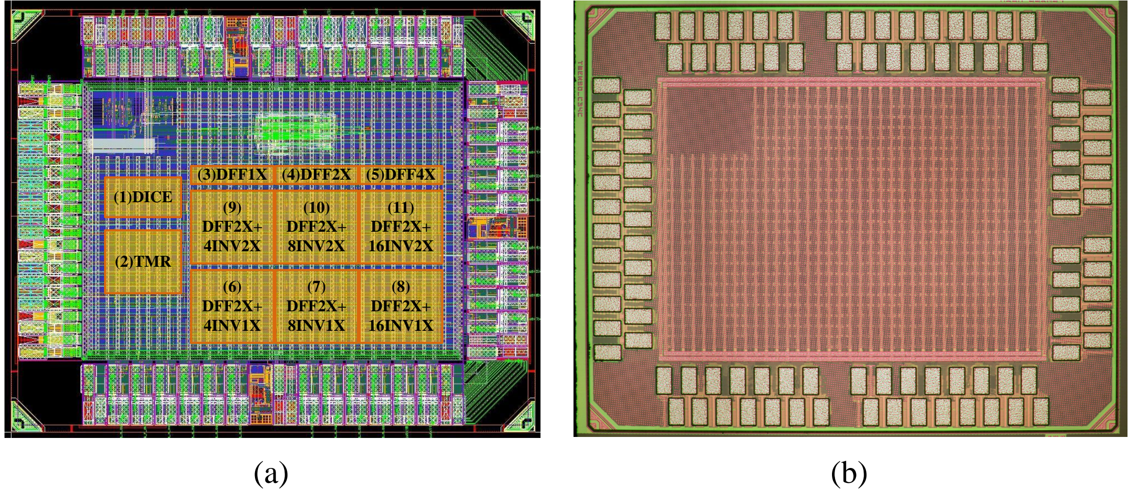


Figure 3.3: (a) Test chip 1 layout, and (b) microphotograph.

circuits.

The layout and microphotograph of test chip 1 are shown in Figure 3.3. In implementing the test structures, we first construct chain 11 as the baseline; then replace the INV2X cells with INV1X cells to make chain 8; remove every other inverter to make chain 10; and so on. In this way, we construct chain 6 to 11 using an identical footprint, thus the impact due to layout difference is minimized. Similarly, chain 3 to 5 also share an identical footprint.

3.1.2 Test chip 2

Test chip 2 measures 1.5mm×1.0mm in size, and it consists of two DSP cores, a standard core and a radiation-hardened core, that compute cross-correlations. Test chip 2 was developed as part of the geostationary synthetic thinned aperture radiometer (GeoSTAR) project [51, 52] led by the NASA Jet Propulsion Laboratory. Each DSP core computes the cross-correlations of 5 inputs with another set of 5 inputs every clock cycle, and accumulates the correlations for 10ms. Following each 10ms integration cycle, the correlation values are read out, and the values are reset for the next integration cycle.

Test chip 2 was synthesized using standard cells of logic gates and flip-flops. The standard core uses commercially available standard flip-flops, while the radiation-hardened core incorporates custom-designed radiation-hardened DICE flip-flops for datapath and TMR flip-flops for control to provide stronger SEE protection. Both

Table 3.1: Ions applied in radiation testing and their nominal LET

Ion	LET(MeV-cm^2/mg)	Ion	LET(MeV-cm^2/mg)
He	0.106	Kr	36.2
N	1.4	Ag	44.5
Ne	2.8	Xe	54.7
Ar	8.9	Au	88.4

cores are placed and routed using commercial CAD software. Test chip 2 provides self test capability by generating test vectors on chip using linear feedback shift registers (LFSR). The standard core measures 0.28mm×0.28mm and the hardened core is larger at 0.33mm×0.33mm due to the larger DICE and TMR flip-flops.

3.1.3 Ion beam testing

In ion beam testing, a test chip is mounted on a test board that is connected to an FPGA board, which generates control signals, test vectors, and collects results for analysis. During the radiation testing, the lids of the test chips are removed and the chips are fully uncovered as shown in Figure 3.4. We ran test chip 1 at 50MHz, a relatively low clock frequency, to probe each individual test structure. Constant data 0 and constant data 1 are fed to the inputs of the 11 shift register chains, and the outputs of these chains are recorded and analyzed by the FPGA. Results are expressed in cross section per bit across a range of LET values. Cross section represents the upset susceptibility, or more specifically, the number of upsets per unit ion fluence (fluence is the flux integrated over time in ions/cm^2). Cross section per bit can be interpreted as the upset rate per unit ion fluence, i.e.,

$$\text{cross section per bit} = \frac{\text{number of upsets}}{\text{number of flip flops} \times \text{fluence}}. \tag{3.1}$$

The average flux applied in our tests ranges from 1.17×10^5 to 2.63×10^5 ions/cm^2·s for chip 1, and from 2.76×10^5 to 1.42×10^6 ions/cm^2·s for chip 2. The ions used and their LET values are listed in Table 3.1.

Chip 2 was tested at a 100MHz and a 500MHz clock frequency using random input vectors generated by on-chip LFSRs. Each test run consists of 10,000 10ms integration cycles, each followed by a readout. The continuous testing requires frequent readouts from the ASIC. To automate the testing, we used a Python script to pre-compute the outputs in each run and store them in the memory on FPGA before each run. The

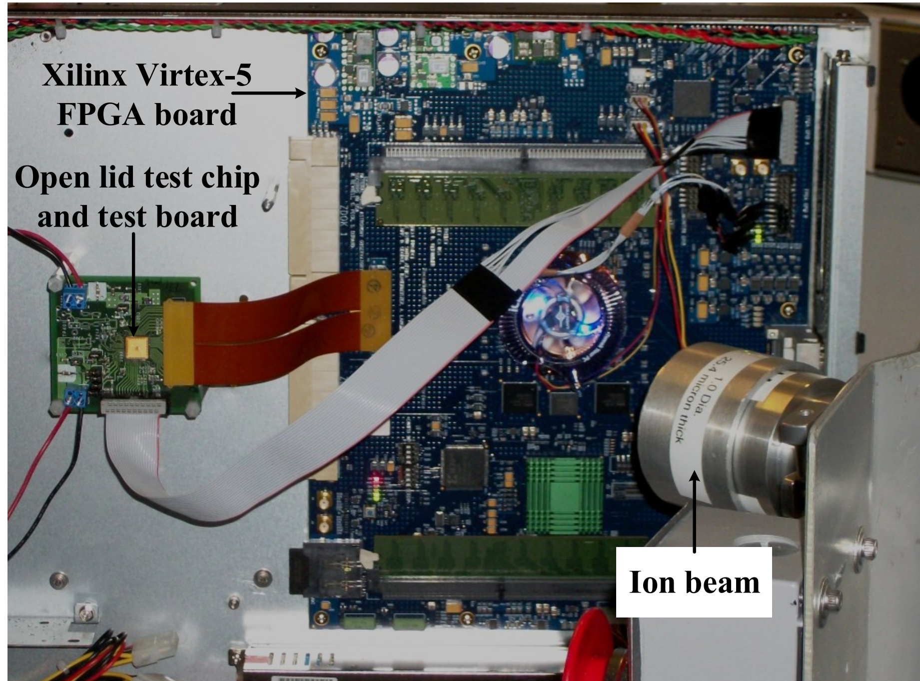


Figure 3.4: Radiation test setup.

Python script also controls the supply voltage and clock frequency of the ASIC. The FPGA is connected to the ASIC and it activates its control signals to start the run. At the end of each integration cycle, the FPGA checks the ASIC outputs for upsets by comparing with the pre-stored outputs. As we are testing the effect of SEE on the system, an “upset” is recorded if any bit in a set of cross-correlation values is wrong. The automated test setup is illustrated in Figure 3.5.

Note that an upset recorded in the radiation testing of chip 2 can be caused by a single SEE occurrence or multiple occurrences during a 10ms integration cycle. The upset count is an indication of the effect of SEE on this particular DSP core over a given time period, rather than a measure of the number of SEE occurrences. Although we report the test chip 2 results in cross section per bit by normalizing the number of upsets by the number of flip-flops in the design and the fluence over the integration cycle as in equation (3.1), the reported cross section per bit is in fact the lower bound of the number of SEE occurrences.

3.2 Circuit design considerations

Circuit design choices, including circuit topology, sizing, and logic depth, determine the circuit’s radiation tolerance. The test structures in chip 1 are subject to

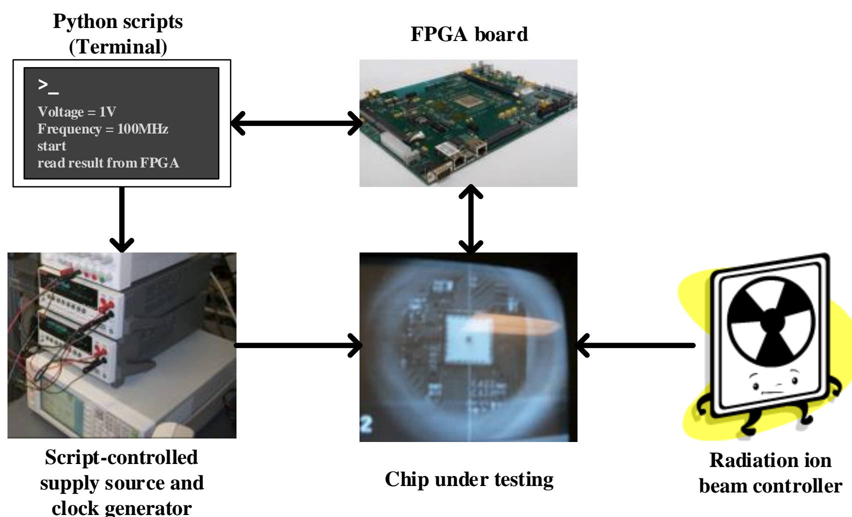


Figure 3.5: Automated testing of chip 2.

the same radiation environment, and the SEE-induced upset rates are compared in Figure 3.6. Note that in plotting chip 1 test results, each data point is a result of several test runs, and the effective fluence is used in equation (1) that factors in the angle of the ion beam with respect to the chip surface.

3.2.1 Radiation hardening by redundancy

Radiation-hardened DICE [1, 50] and TMR flip-flops [2] are commonly used in spaceflight systems to offer better protection against SEE. In low-LET neon (2.8 MeV-cm^2/mg) and argon (8.9 MeV-cm^2/mg) testing of chip 1, DICE and TMR flip-flops are shown to provide at least one order of magnitude improvement in upset rate compared to the standard, unprotected D flip-flops as in Figure 3.6. At higher LET levels (above 50 MeV-cm^2/mg), DICE and TMR become less effective, which is partly due to the lack additional layout spacing between redundant storage nodes [48] and partly due to the increasing multiple bit upsets. Heavier ions such as xenon (54.7 MeV-cm^2/mg) and gold (88.4 MeV-cm^2/mg) deliver much more energy and likely induce more multiple bit upsets [53], making DICE and TMR less effective at high LET levels.

In general, scaling makes DICE and TMR less effective because scaling shrinks the circuit layout and redundant copies in DICE and TMR are physically placed closer to the primary copy [48]. Therefore it becomes more likely for both the redundant and primary copies to be affected by a particle strike, especially at high LET levels.

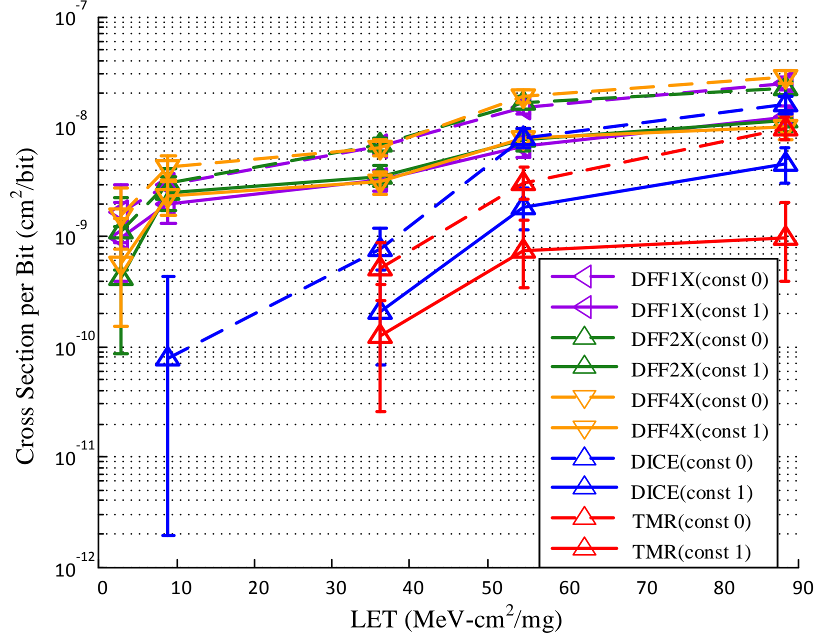


Figure 3.6: Cross section per bit with ion energy for standard, DICE and TMR flip-flops.

To make DICE and TMR more effective, redundant copies need to be placed further apart for isolation, making it less area efficient and partially defeating the purpose of scaling.

3.2.2 Increasing critical charge by upsizing

Scaling reduces device sizes and the critical charge, or Q_{crit}, required to hold a logic level, making circuits more vulnerable to SEE [54]. Previous work suggests that upsizing increases Q_{crit} and immunity against soft errors because of larger capacitance on storage nodes. Upsizing also increases the device drive strength, which helps error recovery [55]. However, we observe in the testing of chip 1 that the difference in upset rates between DFF1X, DFF2X and DFF4X is negligible, as shown in Figure 3.6, which is contrary to previous beliefs. One explanation is that increasing Q_{crit} and drive strength to improve upset immunity is counteracted by the larger drain areas to collect charge.

The weak dependence of upset rate on Q_{crit} is also a result of the amount of charge injected by heavy ions that is much higher than the Q_{crit} even with moderate upsizing. When the charge injected is comparable to Q_{crit}, the minimum charge needed for an

upset, we expect increasing Q_{crit} by upsizing to play a stronger role. However, in a deep submicron bulk CMOS design where Q_{crit} is very low, upsizing is ineffective and inefficient.

3.2.3 Depth and sizing of combinational circuits

Combinational circuits also contribute to upsets. Particle strikes cause glitches in combinational circuits, known as single-event transients (SET). If a SET of a large enough magnitude happens to be sampled, an upset is registered. A SET in combinational circuits often does not lead to upsets, as the SET can be electrically attenuated along the path (electrical masking), or is blocked from propagating due to off-path inputs (logical masking), or the SET arrives too late to be sampled by the flip-flop (temporal masking). For these reasons, the upset rate due to SET depends on logic design and topology, sizing, and timing. With all the masking effects, it is unclear whether SET is an important factor in determining the upset rate.

We evaluate the effect of SET using shift register chains incorporating inverters of various sizes and stages in test chip 1. A longer and upsized chain increases the SET collection area, but also allows the SET to be more electrically attenuated due to higher capacitance and longer path. The results in Figure 3.7 show that the upset rate is almost independent of the depth and sizing of combinational circuits, and adding combinational circuits results in no significant increase in upset rate. Upsets in flip-flops still dominate in the 50MHz testing of chip 1.

Note that as a simple combinational circuit, an inverter chain does not offer any logical masking. In a realistic combinational circuit with logic masking, the upset rate due to SET will be lower. Second, in a relatively low frequency 50MHz testing, the temporal masking [29] has downplayed the importance of SET in combinational circuits, as the slow sampling misses most of the SETs of a short duration. In Section IV.B, we will compare the chip 1 results to the high frequency chip 2 test results to evaluate the effects of temporal masking and also account for logical masking in realistic combinational circuits.

3.2.4 Data dependence

The upset rate measured in cross section per bit is dependent on the test pattern: more upsets occur in constant data 0 testing compared to constant data 1 testing as shown in Figure 3.6, suggesting 0-to-1 upsets are more likely than 1-to-0 upsets. This result is consistent among all shift registers chains and across LET levels higher than

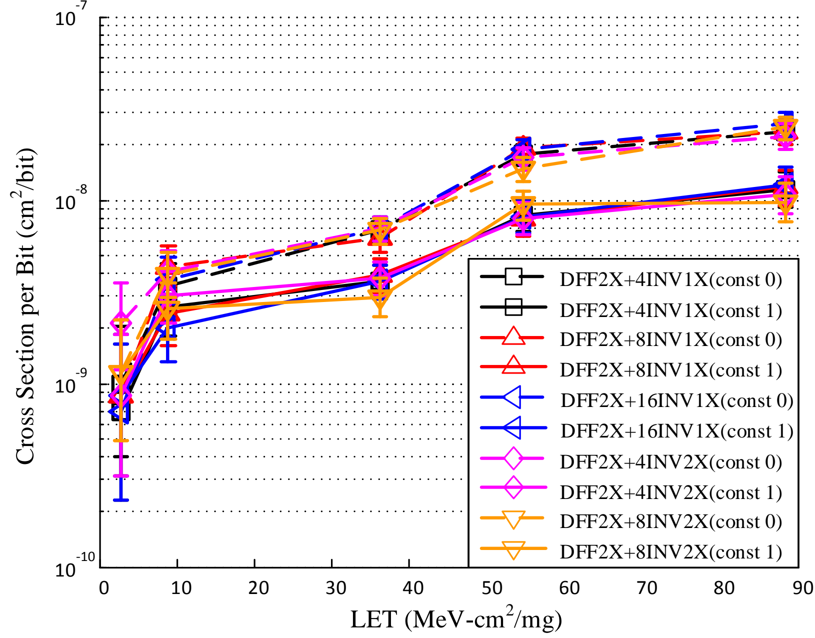


Figure 3.7: Cross section per bit after adding combinational circuits.

30 MeV-cm^2/mg. Some previous work has also recorded this behavior in the testing of standard flip-flops [56–58].

The unequal upsets are found to be a result of the latch design and sizing in the master-slave flip-flops. The two cross-coupled inverters in the latch schematic illustrated in Figure 3.8 are sized differently, resulting in unequal drive strength. The inverter in the primary path (INV1) is sized to have strong and balanced pull-up and pull-down, while the tristate inverter in the feedback path (INV2) is stacked and weaker. As a result, the output of INV2, node NM, which holds the inverted input, is more likely to be affected by charge injection compared to node M. Further, INV2 has a relatively stronger pull-down than pull-up and it holds 0 at its output node NM better than 1, making 1-to-0 upsets on NM more likely and causing more data 0 upsets for the latch.

The upset susceptibility of P- and N-diffusion is another factor behind the unequal upsets. In the latch schematic of Figure 3.8, the total P-diffusion area (transistors T1, T5 and T6) connected to node NM is equal to the total N-diffusion area (transistors T2, T7 and T8). Previous study shows that P-diffusion has a lower cross section, or upset susceptibility, than N-diffusion [54], as P-diffusion resides in an N-well that has a smaller volume to collect charge compared to the P-substrate where N-diffusion

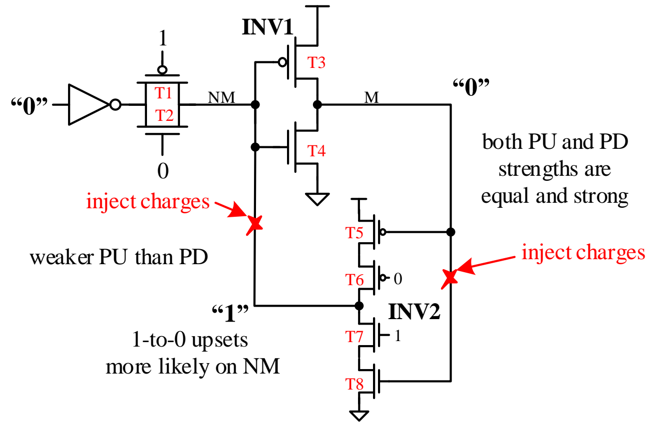


Figure 3.8: Latch design results in unequal upsets.

resides. Therefore, N-diffusion will collect more charge, causing NMOS devices to accidentally turn on and contributing to 1-to-0 upsets on NM. This effect is less pronounced at node M as T3 is sized larger than T4, resulting in a larger P-diffusion area connected to node M that offsets the cross section difference between P- and N-diffusion.

Circuit design and sizing can be used to balance the probability of transitions to have equal 1 and 0 upsets. However, balancing 1 and 0 upsets is not always desirable. In special applications, one is often preferred over the other. Suppose in a controller finite-state machine design, 0-to-1 upset moves the system from an idle state to an active state that causes the stored data to be overwritten; while a 1-to-0 upset moves the system from an active to an idle state, which can be considered safer. Circuit design and sizing can be used to bias the upset rates if necessary.

3.2.5 Latchup and total ionization dose

No latchup has occurred in testing, which confirms that latchup is of less concern in a deeply scaled technology [57]. Our results also indicate that the two 65nm test chips built in a bulk CMOS process are immune to total ionization dose (TID) effects above 100krad (Si) TID. TID effects such as thresholds shifts, latchup events or permanent damages have been a problem in older CMOS technology nodes. Chip 1 and chip 2 were tested up to a TID of 634krad (Si) and 1950krad (Si), respectively with no noticeable degradation in the chip functionality, performance or power consumption.

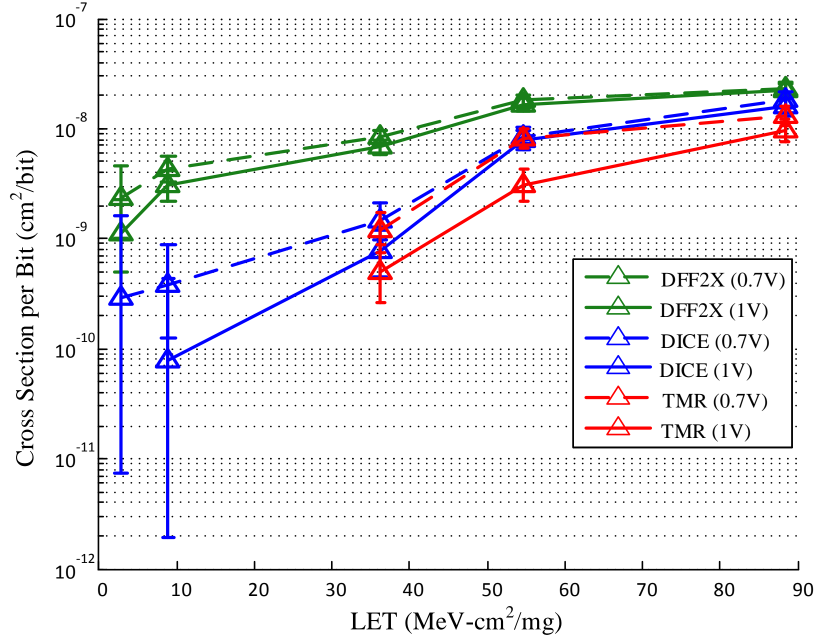


Figure 3.9: Cross section per bit of D flip-flop, DICE and TMR flip-flops at supply voltage of 1.0V and 0.7V.

3.3 Voltage and frequency dependence

Supply voltage and clock frequency are two primary knobs to adjust the performance and power consumption of an ASIC chip. Voltage and frequency scaling also have direct implications on SEE. We evaluate the supply voltage effect using the test structures in chip 1 and the two DSP cores in chip 2, and the frequency effect by comparing chip 2 results at 100MHz and 500MHz.

3.3.1 Supply voltage scaling

Supply voltage scaling reduces Q_{crit} and makes circuits more vulnerable to upsets. Figure 3.9 shows a consistent increase in cross section per bit for the standard, DICE and TMR flip-flops when the supply voltage is reduced from 1.0V to 0.7V. The effect of reducing supply voltage is more noticeable at low LET levels and in DICE and TMR flip-flops, but the difference becomes much narrower at high LET levels.

Attempts to increase Q_{crit} by increasing the supply voltage have little effect at high LET levels because the injected charge by the heavy ions is already much higher than Q_{crit}.

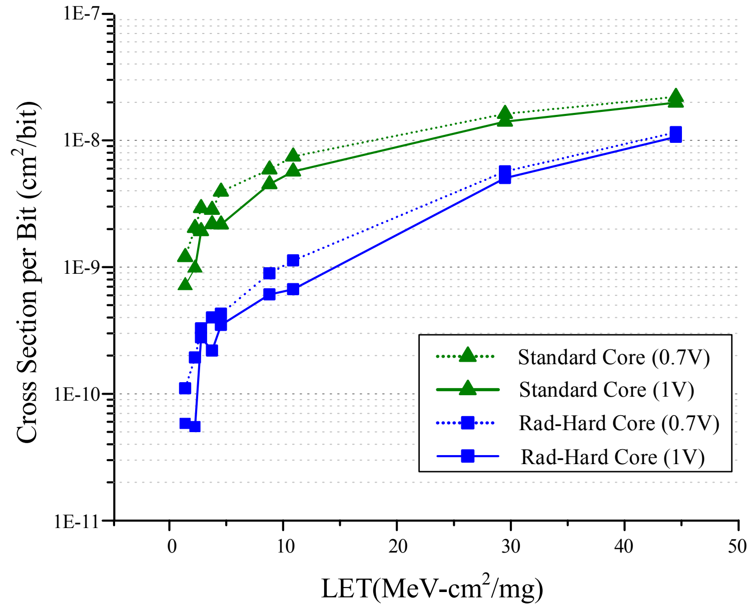


Figure 3.10: Cross section per bit of standard and radiation-hardened correlator cores at supply voltage of 1.0V and 0.7V (clock frequency of 100MHz).

The results of 100MHz dynamic testing of chip 2 at 1.0V and 0.7V are illustrated in Figure 3.10. The radiation-hardened DSP core equipped with DICE flip-flops for datapath and TMR flip-flops for control exhibits an order of magnitude lower upset rate than the standard DSP core at low LET levels, but the difference is diminished at high LET levels. Voltage scaling makes a less pronounced difference, and the difference also becomes negligible at high LET levels, which agrees with the test chip 1 results above. The insight confirms that supply voltage scaling does not always lead to a large increase in upset rate, making it a viable option for power reduction in spaceflight ASIC chips if a small increase in upset rate is acceptable.

3.3.2 Clock frequency effects

Clock frequency affects the upset rate following two mechanisms: at a high frequency, frequent sampling causes more SETs to be registered; at a lower frequency, fewer SETs are registered (known as temporal masking), but flip-flops need to retain data for a longer period, which makes them more vulnerable to SEUs. Frequency shifts the relative importance of SET and SEU: SEU in sequential circuits dominate at a lower frequency, and more SET-induced upsets are expected at a higher frequency.

The results of chip 2 frequency testing are shown in Figure 3.11. In the standard

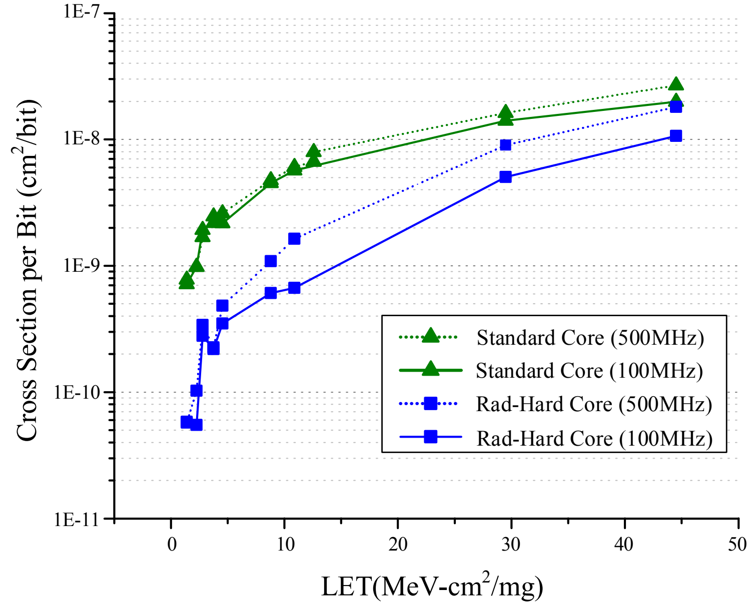


Figure 3.11: Cross section per bit of standard and radiation-hardened correlator cores at 100MHz and 500MHz (1.0V supply voltage).

DSP core, increasing the clock frequency from 100MHz to 500MHz has little effect at low LET levels, indicating the dominance of SEU at low LET, a phenomenon also observed in the 50MHz testing of chip 1. At high LET levels, the upset rate at 500MHz is slightly higher than at 100MHz, which is attributed to the combined effect of more SETs under high energy particle impact and high frequency sampling that causes more SETs to be registered as upsets.

The radiation-hardened DSP core shows a stronger frequency dependence than the standard core across a wide range of LET levels. This observed frequency dependence can be explained by two factors. First, the DICE and TMR flip-flops in the radiation-hardened DSP core offer a better protection against SEUs, thus the SEU rate is noticeably lower than in the standard core, especially at low LET levels. Increasing the clock frequency in the radiation-hardened DSP core causes SET-induced upsets to become relatively more significant. Second, the DICE and TMR flip-flops in the radiation-hardened core have longer setup and hold times relative to the standard flip-flops, which leads to a longer sampling window to capture SETs. Therefore, the frequency dependence of SET-induced upsets is more apparent in the radiation-hardened DSP core.

The high frequency test results suggest that hardening flip-flops alone is insufficient for ASIC chips operating at a 100MHz or higher clock frequency. SET-induced upsets play an important role at high clock frequency, and it is necessary to incorpo-

rate techniques to detect and overcome SET in combinational circuits for complete protection.

3.4 Summary

We evaluate single-event effects using two 65nm bulk CMOS ASIC test chips. We observe that hardened flip-flops become less effective under heavy-ion testing, and approaches that increase the critical charge by upsizing and increasing supply voltage are also not as effective. The results suggest that the charge conveyed by heavy ion strikes has far exceeded the critical charge needed to cause an upset, thus tuning of the critical charge has little effect. However, multiple bit upsets are more likely with heavier ion strikes, causing DICE and TMR flip-flops to fail more frequently. We also observe the unequal 0-to-1 and 1-to-0 upsets in all shift register tests, which are attributed to the latch design, sizing, and the cross section of P- and N-diffusion. Circuit design and sizing can be used to balance or bias the upset rates for specific applications.

Our tests suggest that single-event transients in combinational circuits are less critical than single-event upsets in sequential circuits at a low clock speed of 50MHz or 100MHz due to temporal masking. Increasing the clock frequency to 500MHz increases the relative importance of single-event transients in combinational circuits, which increases the upset rate in a radiation-hardened core and renders the sequential-only protection ineffective.

The radiation test data of two chips are translated to the expected upset rate in the geosynchronous orbit. The radiation-hardened DSP core is expected to incur 1 upset every 70 years in the geosynchronous orbit, while the standard DSP core is expected to incur 1 upset every year. The standard DSP core offers significant advantages in power and area cost. Even without any hardening, the error rate of the standard DSP core is considered acceptable for our project, as long as the system can recover from an upset and reset its state.

This work sheds light on various design aspects that can potentially affect the robustness of deeply scaled ASIC chips under heavy-ion impacts. The 65nm bulk CMOS technology permits acceptable designs for spaceflight systems. Taking advantage of the single-event effect characterization and design considerations to build more efficient and resilient systems remains our future work.

CHAPTER IV

Error-Detection Circuits and Simulation Methodology

To enhance the robustness of deep-submicron designs against occasional delay errors and soft errors, online circuit techniques have been proposed to detect error occurrences. These techniques can be classified to three groups based on how the checking is performed: post-edge checking that detects error in a window after the sampling edge [5, 30–32], pre-edge checking that detects error in a window prior to the sampling edge [4, 33], as in Figure 4.1(b), and multi-edge checking that detects errors by upsampling using multiple clock phases [34, 35].

Each existing technique has its own advantage and limitation. The post-edge technique incurs no performance penalty as checking occurs after the sampling edge, but each path under post-edge protection must be carefully tuned to avoid race conditions. The pre-edge technique is free of any hold time constraints, and it can use the delay slack in fast paths for error detection, however it prolongs the clock period if it is used in slow (critical) paths. Note that none of the above techniques alone is well suited to providing coverage of all types of datapaths. The implementation cost is also prohibitive, as each of the above techniques costs more than 40 transistors [5,30,33] or requires special clock controls [35], making it very expensive to equip every datapath for a full coverage.

To overcome these challenges, this chapter propose a diverse error detection technique to protect combinational logic paths while minimizing the performance penalty and implementation cost. In particular, a new cross-edge technique is applied and its special pre-edge and post-edge versions for moderate, fast, and slow paths, respectively. This method utilizes the inherent redundancy in a flip-flop design, thereby keeping the cost at only 31 transistors. Furthermore, it can also be tuned by duty cycling the clock signal, offering more flexibility for diverse levels of protection.

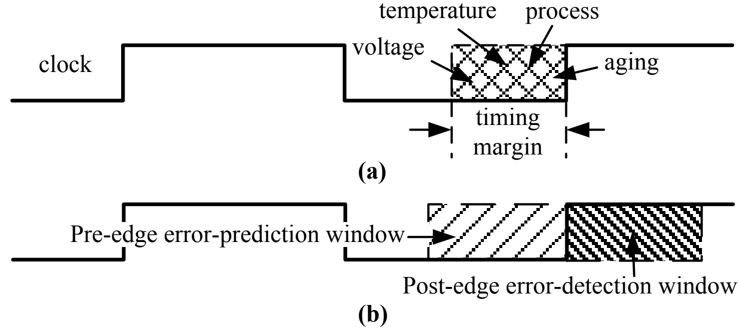


Figure 4.1: (a) Timing margin allocated for static and dynamic variations, and (b) pre-edge and post-edge error checking window.

Table 4.1: Comparison of in situ error detection techniques

Technique	Post-edge checking	Pre-edge checking	Multi-edge checking
Implementations	RAZOR [5, 30], DSTB [31], TDTB [59]	BISER [33], aging sensors [4]	PEDFF [34], TIMBER [35]
Detection mechanism	Transition [5, 31] Duplication [30, 59]	Transition [4] Duplication [4, 33]	Duplication [34, 35]
Clock domains	1	1	2 [34] and 4 [35]
Race conditions	Yes	No	Yes
Checking window	Limited by fast path	Limited by max clock period	Flexible by multi-sampling

4.1 Transient error detection circuits

Pre-edge, post-edge and multi-edge are three classes of in situ timing error detection techniques. Their unique features are listed in Table 4.1 for comparison. The post-edge technique performs error detection after the sampling edge, thus eliminating the timing margin that would otherwise be necessary for the occasional variation-induced delay errors. The post-edge technique has been applied in high-performance, low-power designs. It allows the clock period to be reduced for a higher performance or the supply voltage to be reduced for a lower power consumption, as the post-edge checking acts as a safety net to detect the resulting delay errors. Post-edge checking is often implemented by a duplicate shadow latch that holds a post-edge sample that is compared with the primary sample for error detection [30, 59]. Transition detection was recently proposed as an alternative for a more area-efficient implementation. Successful designs including the second-generation RAZOR [5, 32] and DSTB [31] have been demonstrated. However, it requires each path under post-edge protection to be

carefully adjusted to avoid race conditions.

The pre-edge technique has been presented in [4,33] to monitor errors by detecting transitions in the checking window before the data is registered. The pre-edge technique is free of hold time constraints and is proven to be effective, e.g., in detecting slow changing events such as transistor aging [4], but the checking window occupies a portion of the clock period and degrades the performance in order to guarantee error-free operations for critical paths.

The multi-edge technique in [34, 35] offers more flexibility as the sampling edges for error detection are not necessarily aligned with the primary sampling edge of the datapath. Even though multi-edge improves the performance by adaptive time borrowing, the multi-edge implementation costs more than the transition detection introduced in post-edge and pre-edge checking and an additional clock domain also adds significant design complexity.

Each of the three techniques has its advantages and disadvantages in handling paths of different delays. The output of a critical path makes its final transition close to the sampling edge, while a non-critical path makes its final transition early. In the case of a fast path, the transition can happen shortly after the launching edge, resulting in a tight hold time constraint and a significant idle period prior to the end of the clock period. The ideal error checking window in each case should be placed right after the transition point as any delayed transition is an indication of possible errors. The desirable checking window for each type of path is annotated in Figure 4.2(a). The figure also shows the difficulty of designing one checking window that fits all cases: a post-edge checking window causes race conditions from fast path, as in Figure 4.2(b); a pre-edge checking window leads to performance penalty due to critical path as the clock period needs to be lengthened, as in Figure 4.2(c).

Besides the location of the checking window, the length of the checking window is also important. A fixed length, as in most of the existing pre-edge and post-edge implementations, offers only a fixed protection against non-deterministic errors with variable durations. A soft error can last from a few pico seconds to hundreds of pico seconds [60]. The large variation calls for a tunable scheme to support different levels of protection as needed.

4.2 Flexible cross-edge checking

We exploit the diverse path delay distribution and design a new cross-edge checking to improve the performance while minimizing race conditions. We implement

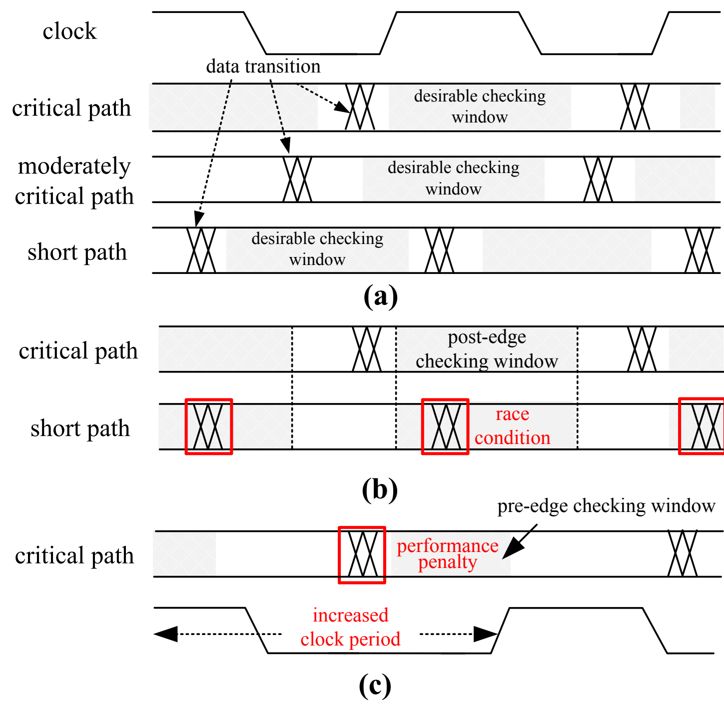


Figure 4.2: (a) Checking window positions depending on path criticality, (b) fixed post-edge checking window causing race conditions, and (c) fixed pre-edge checking window leading to performance degradation.

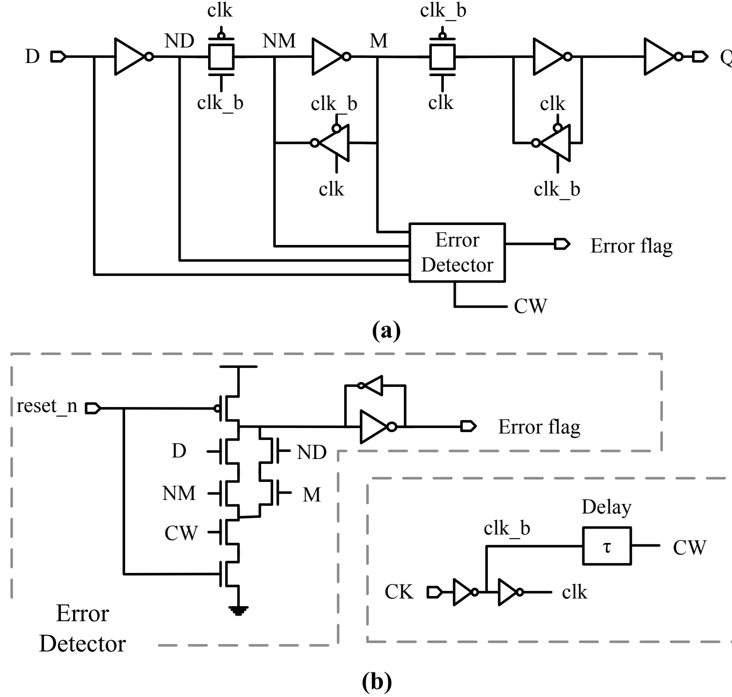


Figure 4.3: (a) Cross-edge checking design based on a conventional transmission gate flip-flop [3], and (b) error detector design.

cross-edge checking based on a low-cost transition detection using a flip-flop. The cross-edge circuitry can be tuned depending on path criticality: pre-edge for fast paths, post-edge for critical paths, and cross-edge for the optimal trade-off between performance and race conditions in moderate paths. We assume that the path criticality is determined in design time for tuning the cross-edge circuitry.

A cross-edge checking window starts prior to the sampling edge and extends after the sampling edge, thus it crosses the sampling edge. A flexible and efficient circuit implementation is shown in Figure 4.3(a) based on a conventional transmission gate flip-flop [3]. The flip-flop naturally keeps four copies of an input: D, ND, NM and M ("N" indicates logic inversion) that are phased apart by inverters and a transmission gate. We rely on the inherent redundancy instead of creating duplications, which is the key to achieving lower power and area.

4.2.1 Cross-edge circuitry

The cross-edge error detector circuit is shown in Figure 4.3(b). The checking window CW is based on the inverted clock clk_b phased by an amount τ that is controlled by a local delay element. The delay element can be shared by a group

of cross-edge flip-flops to amortize its cost. We can increase τ to push the checking window forward in time towards a post-edge checking for critical paths, or shrink the delay towards a pre-edge checking for fast paths. To reduce the design effort, a number of checking windows can be made based on different local delays. We will select one for a cluster of paths to suit their criticality. In addition to the location-tunability of the checking window, the duration of CW is controlled by duty cycling the clock signal. Thus, the duration is a tunable fraction of the clock cycle, and is dictated by one of the clock phase (i.e., low phase of the clock in Figure 4.4).

The error detector circuit in Figure 4.3(b) is made of a dynamic gate followed by a latch [5]. The dynamic gate first pre-charges. When CW is high, it checks the agreement of (D and NM) and (M and ND) as an indication of an erroneous transition. More specifically, the cross-edge checking window can be divided into two parts: a pre-edge part and a post-edge part. During the pre-edge part, clock is low and the master latch is transparent (see Figure 4.3(a)). Early samples in NM and M are checked against late arriving samples D and ND to accomplish pre-edge checking. The dynamic gate is designed to respond much quicker than the propagation delay between D and NM and the delay between M and ND to guarantee the proper functionality. During the post-edge part of the checking window, the master latch is holding and the stored samples in NM and M are checked against post-edge samples D and ND to accomplish post-edge checking. If an erroneous transition is detected, the dynamic gate pulls down and an error flag is generated to trigger appropriate actions. An error could be corrected through rollback recovery [61], architectural recovery [62] or cross-layer recovery [63]. Once the error is resolved, the detector is reset by the controller.

We implemented the cross-edge circuitry in a 65nm CMOS technology and the SPICE simulation waveforms are provided in Figure 4.4. The checking window is generated by delaying clk_b. In the first clock cycle, a clean input D is shown to be correctly registered. While in the second clock cycle, the input D makes an erroneous transition during the checking window and it is detected as an error. The proposed cross-edge checking circuitry adds a detector and extra wiring to the conventional flip-flop design. Our 65nm CMOS circuit simulation shows that the design consumes 13% more static power and 25% more total power than the conventional flip-flop at the highest switching activity of 1.

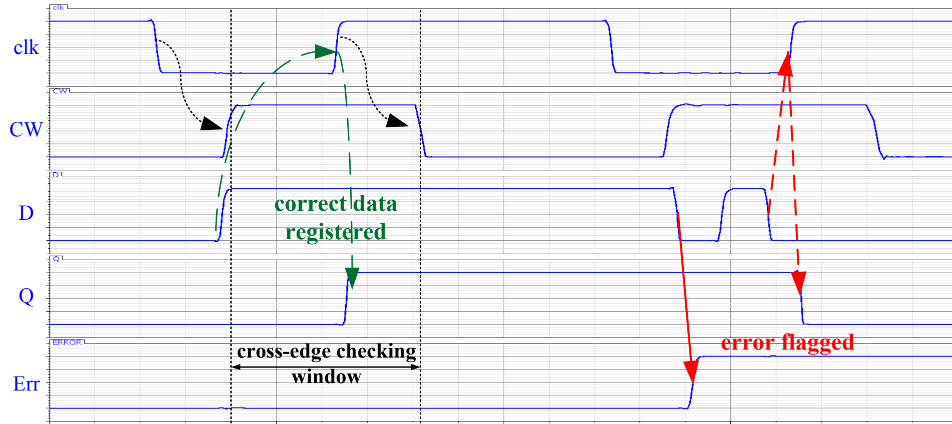


Figure 4.4: Waveforms illustrating the operations of the cross-edge circuitry.

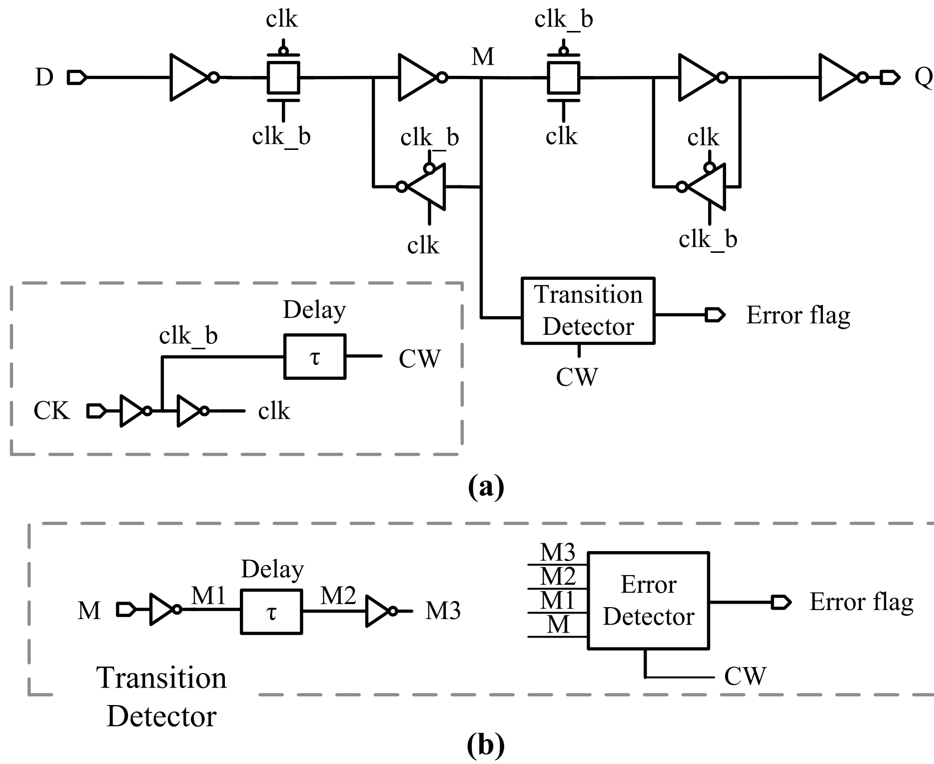


Figure 4.5: (a) Pre-edge checking design, and (b) transition detector design.

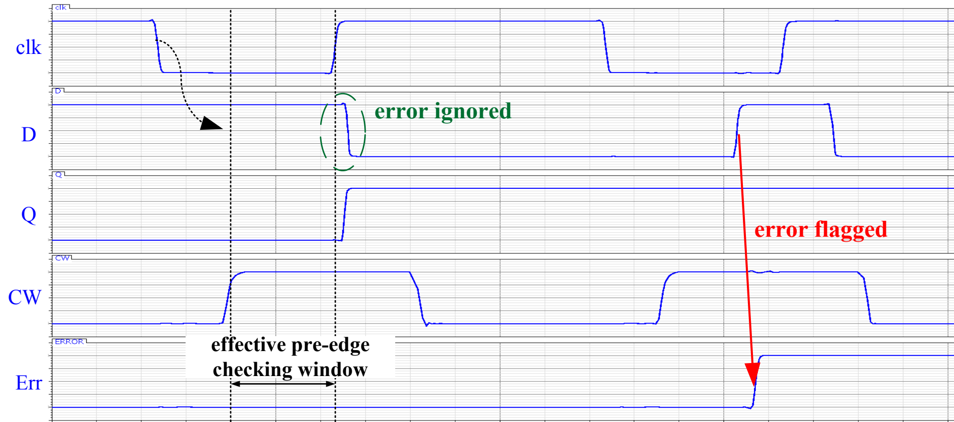


Figure 4.6: Waveforms illustrating the operations of the pre-edge circuitry.

4.2.2 Pre-edge circuitry

Pre-edge checking can be made by shortening or eliminating the local delay τ in the cross-edge design, but an alternative design is possible for a guaranteed alignment between the sampling edge and the checking window for a zero hold time. Figure 4.5(a) shows the alternative design. The same error detector circuit is used but it taps only the master latch internal node M and its delayed copies. During the pre-edge part of checking window, the master latch is transparent and the error detector monitors (M and M2) and (M1 and M3) for transition detection, which implements pre-edge checking. When the master latch is holding during the post-edge part of the checking window, M can no longer change and post-edge checking is essentially turned off.

The SPICE simulation waveforms of this circuitry are illustrated in Figure 4.6. An erroneous transition made by input D after the sampling edge is blocked by the master latch, thus no error is detected. In this way, an effective pre-edge checking window is created as shown in Figure 4.6, and the alignment of this effective pre-edge window with the sampling edge is guaranteed. The design costs 15% more static power and 33% more total power than a conventional flip-flop at the highest switching activity of 1. The slight increase in power compared to the cross-edge design is due to the extra inverters to generate M1, M2 and M3 as in Figure 4.5(b).

4.2.3 Comparisons

We compare the implementation complexity of the proposed cross-edge technique with state-of-the-art in situ error detection techniques in Table 4.2. All error detection techniques use one delay element to generate the checking window. The transistor

Table 4.2: Implementation complexity of in situ error detection techniques

Flip-flop type	Standard [3]	Post-edge [5, 30]	Pre-edge [33]	Multi-edge [35]	Cross-edge (this work)
Transistor count	22	47	52	36	31
Delay elements	0	1	1	1	1
Clock domains	1	1	1	≥ 2	1

count of the proposed cross-edge flip-flop is only 31, much lower than other designs. In particular, the post-edge RAZOR technique uses detection clock generator and transition detector, which increase its transistor count to 47. The pre-edge BISER technique uses two conventional flip-flops and a C-element as a filter, resulting in a transistor count of 52. The transistor count of multi-edge TIMBER flip-flop is 36, but a specially designed clock control circuit is required to generate input signals to the flip-flop, costing more than 30 extra transistors.

4.3 FPGA-based transient error simulator

Even though these error-resilient design techniques improve circuit reliability and reduce design margins, it is often difficult to precisely evaluate these techniques in design time. Conventional software-based transient circuit simulation using commercial CAD tools becomes very slow with the addition of transient error effects to reach a good coverage [64]. FPGA has been used to accelerate error simulations, but past work has been limited to cycle-based "digital" simulations with coarse delay and error models [65, 66].

The challenges with slow software-based transient simulator and inaccurate FPGA-based digital simulator motivate us to design an FPGA-based transient error simulator to aid the design and evaluation of error-resilient techniques. In the proposed paradigm, we detach circuit characterization from error simulation: circuit characterization will be done efficiently using standard CAD tools, while the lengthy error simulation will be done on a fast FPGA-based error simulator. Fast FPGA platforms allow the delay models and error models to be fully exercised, and the interactions between errors and circuits to be captured for a good coverage.

The proposed FPGA-based transient simulator is comprised of three main parts as shown in Figure 4.7: (1) delay profile of the datapath under test, (2) transient

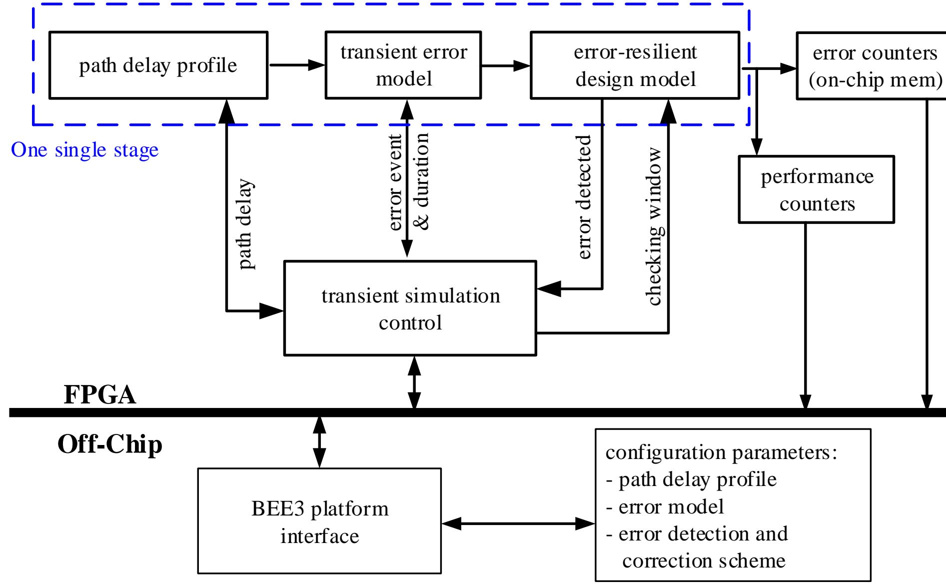


Figure 4.7: FPGA-based transient simulation platform. A multi-stage simulation can be constructed by cascading single stages with individual delay profiles and models.

error models, and (3) error-resilient designs to be evaluated. All three parts are programmable, producing a versatile and general-purpose error simulation platform.

4.4 Transient error simulator

Transient simulation is usually carried out at a much finer time step to mimic as closely as possible the circuits' continuous time behavior including voltage and current. For error simulation, the continuous monitoring can be simplified to the monitoring of events, such as data propagating to the end of a datapath, an error upsetting an output node, etc. In the past, "digital" error simulations have made the assumptions that error events such as transient faults only occur at clock cycle boundaries [65]. This simplifying assumption allows error simulations to be done very quickly, but it also has two intrinsic problems: (1) transient timing effects are neglected, e.g., a transient fault from a soft error is sometimes masked without introducing an eventual error due to timing masking [29]; and (2) error-resilient circuit designs cannot be fully captured by a cycle-by-cycle digital simulation, e.g., the RAZOR technique [5] of double sampling and correction cannot be simulated.

The proposed FPGA-based transient simulation operates at finer time steps and allows events to occur at these finer time steps. We use the FPGA clock period T_{step}

Table 4.3: Comparison of simulators

	Transient sim (software)	Digital sim (FPGA)	Transient sim (FPGA)
Simulation throughput	1	$> 10^6$ [65]	$> 10^3$
Time resolution	fine step size	one clock period	sub clock period

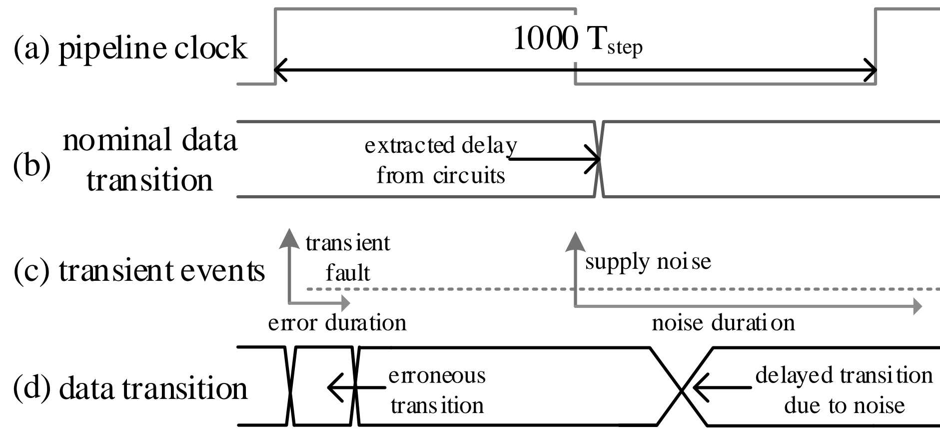


Figure 4.8: Timing charts of the FPGA-based hardware emulation: (a) pipeline clock, (b) extracted nominal data path delay, (c) injected transient events (errors and noise), and (d) data transition as the result of (b) and (c).

as the unit time step. For example, if one chooses a simulation time step of 1ps and a clock cycle period of 1ns, the 1ps time step is mapped to one T_{step} on FPGA, and the 1ns cycle period is mapped to $1000T_{step}$. This setup permits a simulation throughput of $1/(1000T_{step})$. Figure 4.8 presents an illustration of the timing. Note that the quality of transient simulations depends on the time step size: smaller time steps yield more accurate results at a lower simulation throughput. A comparison between simulators is presented in Table 4.3, showing the advantage of FPGA-based transient simulator in providing more accurate simulations at a much higher throughput than software.

4.4.1 Delay model and error model

The FPGA-based transient error simulator contains three parts: circuit (or datapath) delay models, error models and error-resilient design. Unlike a conventional

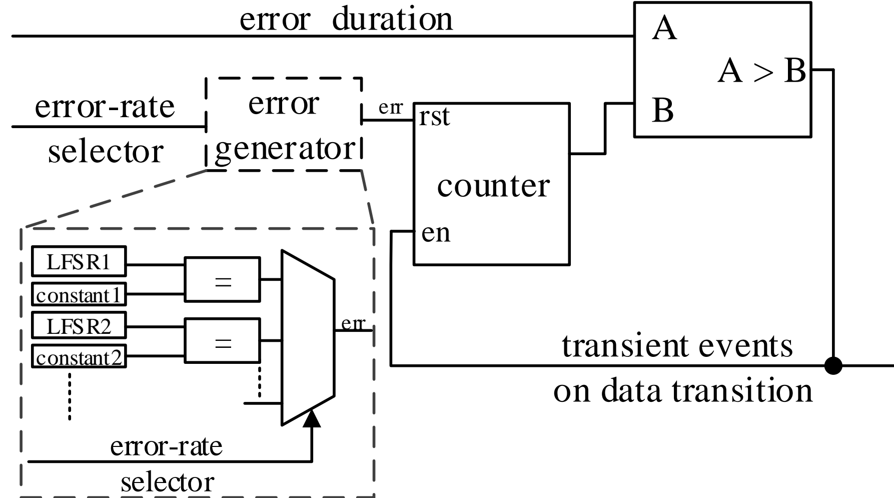


Figure 4.9: Transient event generator.

FPGA emulation, the datapath under test is not directly implemented on FPGA, because FPGA is used as a simulator rather than a prototyping platform. The datapath delay will be characterized through circuit simulators, and the delay model is programmed on FPGA for error simulation.

The datapath delay model is stored on FPGA in two possible formats: either a histogram or an application trace. In the histogram format, datapath delays are binned and the probability of each delay being exercised is associated with each bin. Such a delay histogram can be obtained by circuit characterizations (using SPICE or back-annotated RTL simulation) of common applications that represent realistic workloads or standard benchmarks. On the other hand, the application trace is a list of instruction by instruction delays, which are specific to one application or benchmark. When running the transient simulation on FPGA, we will select datapath delays from the histogram based on the probability of each bin, or run through the application trace. The histogram format is compact and can be used for long error simulations, and the application trace is used for specific tests and corner cases.

Error models are programmed on FPGA alongside the datapath delay model for error simulation. In our proof-of-concept simulator, we implement models of most common errors: soft error (single even transient), coupling noise and voltage droop. Soft error is known for its random occurrence and a soft error causes a transient upset to a circuit node for a period of time as shown in Figure 4.8. A soft error model can be implemented based on a linear feedback shift register (LFSR): LFSR generates pseudo-random numbers, and an error is generated when the random number matches a given constant, as illustrated in Figure 4.9. In this design, the error rate is 2^{-n} where

n is the adjustable bit width of the constant. Once a soft error is generated, it lasts for a duration described by a statistical distribution. Coupling noise effect is similar to soft error and it is modeled as a transient fault in the same way. Voltage droop is modeled as a sinusoidal voltage fluctuation around a nominal value [67, 68]. The voltage droop model is implemented as a randomly generated event that changes the datapath delay for the duration of the droop. The delay change is described by a sinusoid of a select period and peak magnitude. Additional models can be added to capture more error sources of relevance. The models can be tuned to adjust the error rate, duration and distribution.

4.4.2 Transient simulation

The transient simulator keeps a time step counter. At the start of a transient simulation, the time step counter is reset to 0 and the datapath delay model picks a delay t_{path} (in units of T_{step}) indicating the input is launched with an expected propagation delay t_{path} as illustrated in Figure 4.10(a). The output data is initialized to invalid and remains invalid until the time step counter reaches t_{path}. The transient simulation proceeds in steps of T_{step} and the time step counter increments by 1 every step. In each simulation step, each error model decides whether to generate an error based on a selected error rate. From Figure 4.10(a), if a transient event is generated, the output is invalidated for the duration of the transient event. If a voltage droop is generated, t_{path} is lengthened or shortened according to a selected sinusoidal function. The transient simulation controller keeps track of the time step counter, datapath delay, error states, and makes updates to the output indicator. When the time step counter reaches the clock period T_{clk}, the controller inspects the output indicator and records an error if the output is invalid. The transient simulation then moves to the next clock period. The time step counter resets to 0 and a new path delay is picked and the process continues.

The transient simulation can be used to evaluate error-resilient designs. An error detection and correction technique specifies an error checking window CW (duration in terms of number of T_{step} and position relative to the clock cycle boundary) along with an error correction mechanism. To simulate error detection and correction, in each T_{step} inside CW, the simulation controller inspects the output and an error is flagged if the output is changed when the checking window turns on as described in Figure 4.10(b). The error detection triggers error correction mechanism, e.g., by stalling the pipeline (moving the current cycle boundary to $2T_{clk}$) or reissuing the instruction (purging the pipeline and reissuing the current t_{path}). The simulator uses

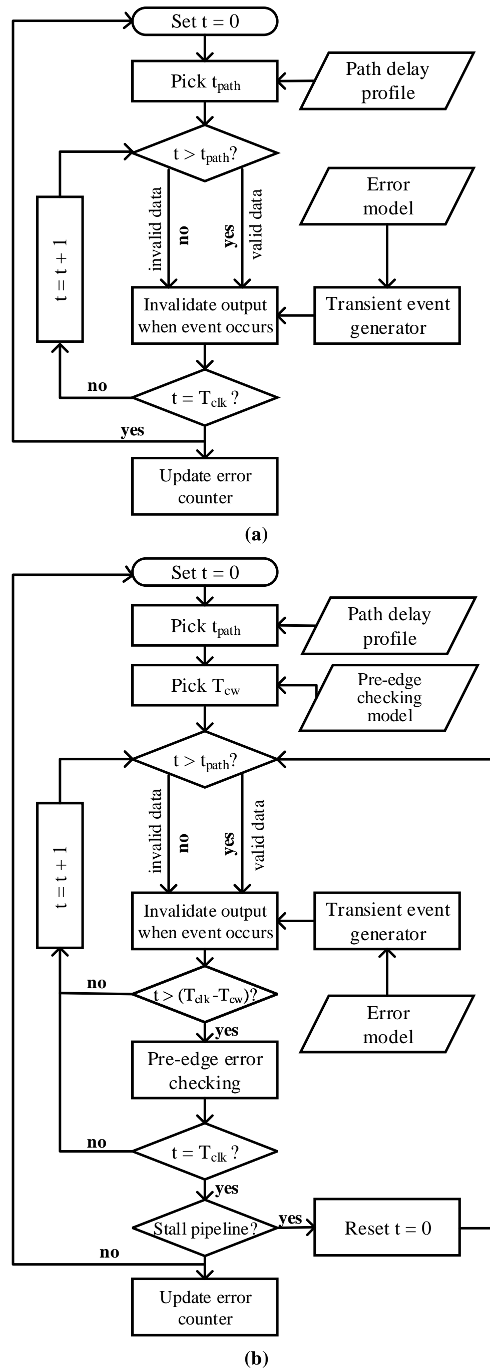


Figure 4.10: Transient simulation state (a) withe the datapath delay profile and transient error model, and (b) the state with pre-edge error detection and recovery by stalling the pipeline.

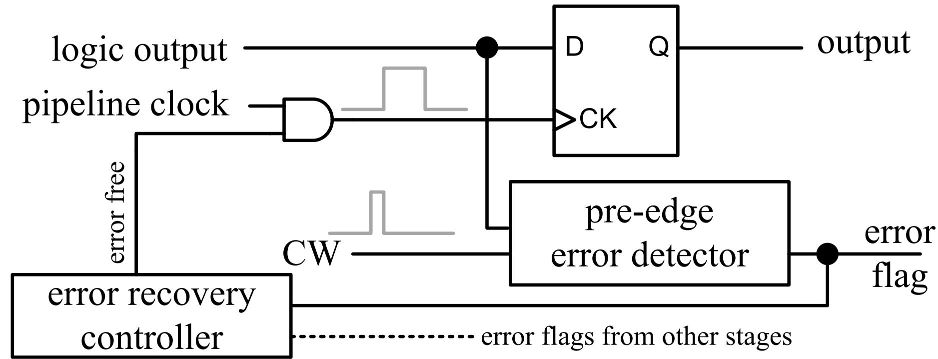


Figure 4.11: Pre-edge error detection and recovery through pipeline stall [4].

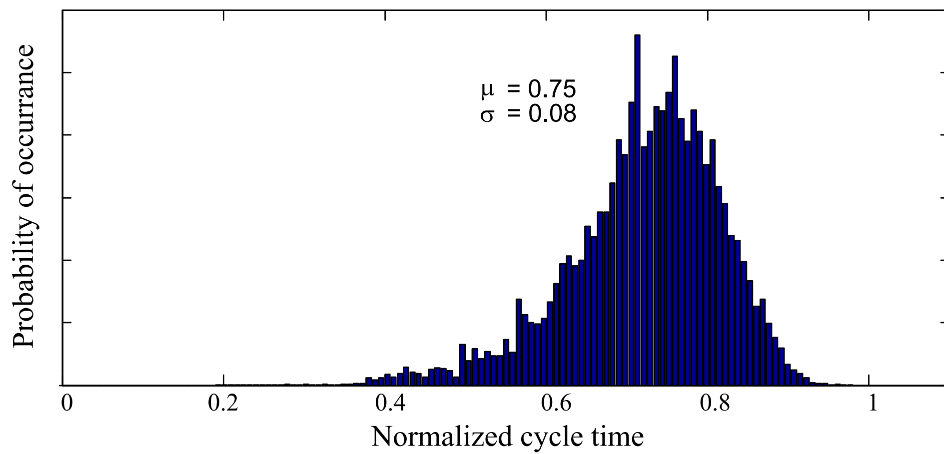


Figure 4.12: Simulated delay distribution from extracted CORDIC processor.

a performance counter to keep track of the number of outputs produced, and an error counter to track the number erroneous outputs. Together with the time step counter, the simulator measures the performance and error rate.

4.5 Evaluation of error-resilient designs

The complete transient error simulator is implemented on a BEE3 platform [69] for a multi-stage pipeline. Each pipeline stage is modeled using a separate datapath delay model. Soft error, coupling noise and voltage droop are added in the simulation. We perform experiments on representative datapaths and error models for evaluating common error detection and correction circuit techniques.

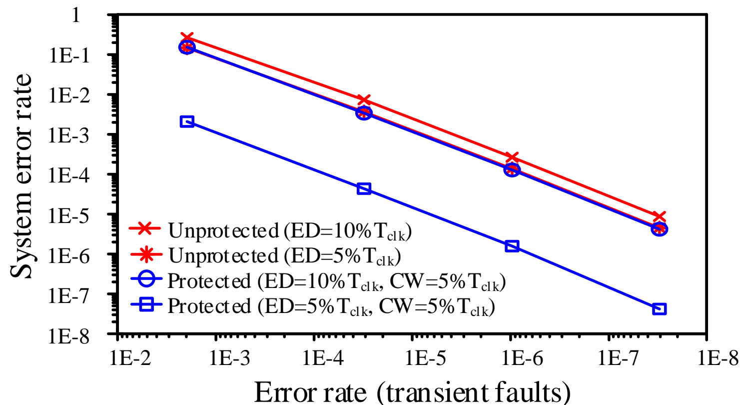


Figure 4.13: Reliability improvement with pre-edge error detection and correction (CW: checking window, ED: error duration, T_{clk}: clock period).

4.5.1 Pre-edge technique

The pre-edge error detection and correction, shown in Figure 4.11, has been proposed in [4, 33] to monitor errors by detecting glitches in the checking window before the output is registered. The pre-edge technique is effective in detecting slow changing NBTI-induced PMOS aging and random transient faults. A longer checking window CW provides better protection against errors at the cost of lengthening the clock period and degrading performance. The pre-edge error detection is accompanied by a pipeline stall to correct the errors [62].

We evaluate the pre-edge error detection and correction on a coordinate rotation digital computer (CORDIC) processor that is synthesized in a 45nm CMOS technology. The processor consists of three identical pipeline stages and the delay profile of each stage is shown in Figure 4.12. We adjust the transient error rate and duration, and measure the effect on the CORDIC processor while tuning the length of CW. The FPGA-based transient simulator captures the CORDIC processor's error rate due to transient failures as shown in Figure 4.13. The CORDIC processor's reliability decreases with higher error rate and longer error duration (labeled as ED in the figures). One important observation uncovered by the transient simulation is that the pre-edge technique is only effective when CW is comparable or longer than the soft error duration. Two orders of magnitude of reliability improvement over the unprotected case can be achieved when CW is appropriately chosen.

Pre-edge error detection triggers pipeline stalls to correct errors, leading to throughput degradation. Transient simulation shows that the throughput of the CORDIC processor is primarily determined by the error rate and the length of CW, as shown in Figure 4.14. Lengthening CW enhances the error protection, but also increases

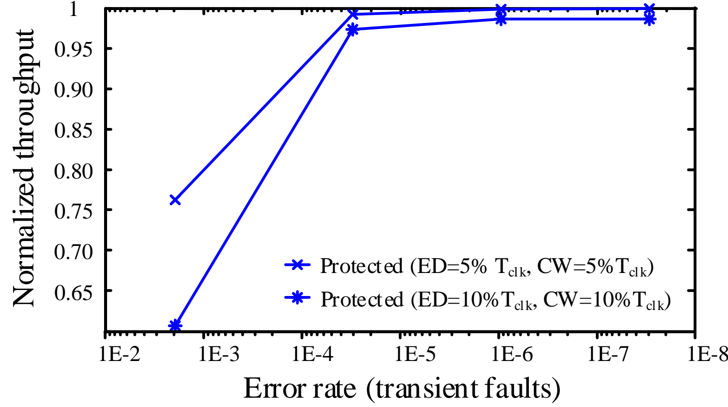


Figure 4.14: Effective throughput using pre-edge error detection and correction (CW: checking window, ED: error duration, T_{clk}: clock period).

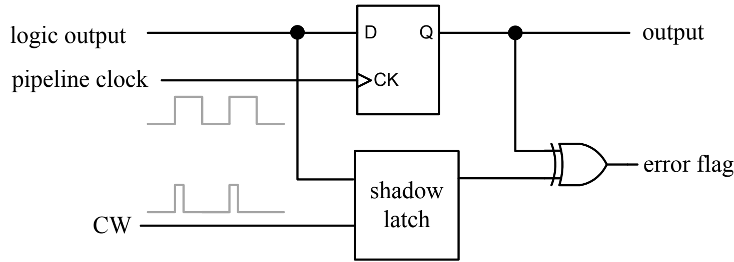


Figure 4.15: Post-edge error detection circuits [5].

the chance of detecting data transition in the critical paths and therefore degrades the throughput. The trade-off between reliability and performance obtained from the transient simulation can be used to guide practical designs.

4.5.2 Post-edge technique

The post-edge error detection and correction technique has been very popular in high-performance, low-power designs. The post-edge technique, illustrated in Figure 4.15, detects errors after the clock edge, allowing correction of delay errors from long paths that exceed the clock cycle time [70]. The technique is often applied in conjunction with dynamic voltage and frequency scaling to increase the clock frequency for a higher performance, or reduce the supply voltage for a lower power consumption.

We evaluate the post-edge error detection and correction on an Alpha processor [71] that is synthesized in a 45nm CMOS technology. Application traces were obtained from the execution stage of the Alpha processor running the instructions of a recursive Fibonacci number generation. We consider the effects of coupling noise and voltage droop on the Alpha processor. The coupling noise is randomly occurring and its

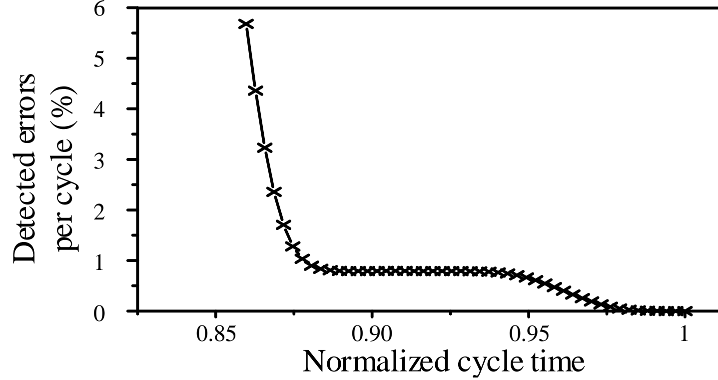


Figure 4.16: Post-edge error detection rate.

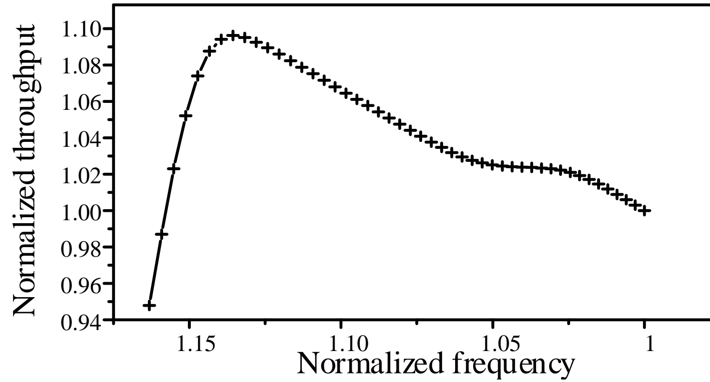


Figure 4.17: Effective throughput using post-edge error detection and correction.

duration is modeled with a Gaussian distribution of a standard deviation of 1%-3% of T_{clk}. The voltage droop usually lasts for a longer period and we assume it increases the circuit delay by up to 15% of T_{clk} [67, 68].

The Alpha processor is simulated with post-edge error detection. A fixed post-edge CW is selected based on the fastest path delay to avoid hold time issues. Dynamic frequency scaling is applied to take advantage of post-edge error detection and correction. As the clock period decreases with increasing frequency, more errors are detected as shown in Figure 4.16. Errors can be corrected using instruction flush [5] or system nuke [31] that are aided by the micro-architecture and operating system. However, error correction usually introduces a few cycles of penalty and the throughput takes a hit. Considering an average 5 cycles of penalty to flush the Alpha processor's pipeline and reissue the instruction, the effective throughput using dynamic frequency scaling can be measured as in Figure 4.17. The peak throughput is achieved at 1.13 times the nominal frequency. Such a design exploration involving reliability and performance can be quickly obtained using the FPGA-based transient error simulator.

4.6 Summary

This chapter presents a new efficient and in situ error detection technique to enhance system reliability against delay and soft errors. This technique exploits datapath criticality by appropriately adjusting the checking window for a higher performance while minimizing race conditions. The error detection technique is implemented in a flexible cross-edge checking circuitry that relies on the inherent redundancy without resorting to additional storage or duplication, thus the implementation cost is kept low. By duty cycling the clock signal, the checking window can be adjusted to provide different levels of protection. An alternative pre-edge circuitry is also proposed to guarantee the alignment of the checking window and the sampling edge for a zero hold time. The flexible and low-cost technique provides diverse path coverage for a fully protected error-resilient system. To rapidly evaluate error detection techniques, an FPGA-based transient simulator for error-resilient circuit and system design and evaluation. The general-purpose simulator consists of configurable datapath delay models and error models to be widely applicable. Error-resilient circuit design techniques, including pre-edge and post-edge error detection and correction, are simulated on this platform based on a synthesized CORDIC processor and an Alpha processor. The simulations shed light on key design choices, such as the length of pre-edge checking window and its impact on reliability and performance. The FPGA-based transient simulation complements circuit simulation and system emulation as a useful tool for resilient circuit and system designs.

CHAPTER V

Confidence-Driven Architecture for Error-Resilient Computing

In this chapter, we propose a confidence-driven computing (CDC) architecture for protection against nondeterministic errors over a wide range of rate and duration. The key concept of the proposed computing model is to employ fine-grained temporal redundancy with tunable threshold for a faster adaptation and an adjustable reliability. The CDC model is suitable for designs using nondeterministic post-CMOS devices. It allows systems to adapt to large runtime variations and reduces excessive design margins for an efficient computing system.

5.1 Related work

Error tolerance and error correction are the primary solutions to designing resilient systems. Error tolerance can be provided by the algorithm itself. For instance, digital signal processing algorithms like image and video processing naturally accommodate imprecision in the computation [72]. Low-power circuits have been designed to take advantage of the algorithmic noise tolerance (ANT) [73], where errors incurred due to supply voltage over scaling can be tolerated by the algorithm. Similarly, the scalable stochastic processor [74] applies error-scaling friendly circuits to make full use of algorithmic error tolerance. Error tolerance can also be provided at the architecture level, as done in ERSA [75] that employs multiple cores of lower reliability to execute probabilistic applications. These recent works improve the robustness, thus the supply voltage and clock frequency margins can be reduced to lower power consumption and to improve performance. However, error tolerance techniques are often limited in their applicability. ANT, stochastic processor and ERSA all rely on the characteristics of the target algorithms and they need to be tailored to each.

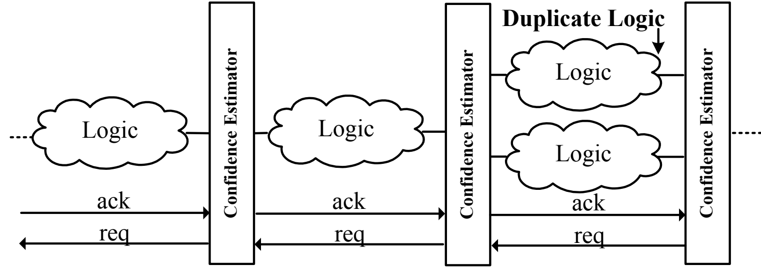


Figure 5.1: Proposed confidence-driven computing model incorporating confidence estimators.

Error detection and correction is general purpose and it is usually aided by spatial duplication or temporal redundancy. For example, feed-forward recovery uses spatial duplication to detect and correct errors with no interruption to the computation. The approach is commonly known as N-modular redundancy or NMR. However, its cost increases to support a wide range of device error rates [76]. The BulletProof design [77] incorporates adaptive sparing using routers in a defect-tolerant architecture, but the area and energy overhead are still significant.

We classify different error recovery techniques based on their target error duration from short transient to permanent. NMR is capable of correcting both permanent errors and transient errors. If an error is known to last for a short duration, a more area- and energy-efficient alternative is through temporal redundancy [78] that can be implemented in a checkpoint and rollback scheme. The RAZOR technique [30] performs rollback at the circuit level by a parallel shadow latch to detect and correct errors. In this way, RAZOR only incurs a small performance overhead, but the error detection window is short and fixed, limiting its effectiveness for errors that last for a larger fraction of a cycle. An alternative circuit-level rollback and recovery technique [79] requires a duplicate datapath for error detection and rollback buffers for correction. The rollback recovery technique has a smaller area overhead compared to NMR, but the protection level cannot be adjusted either.

The confidence-driven computing model proposed in this chapter can be applied with temporal redundancy to extend the error detection window to multiple clock cycles. It also allows temporal redundancy to be applied in conjunction with spatial redundancy to improve throughput and latency. Its key feature is that it enforces a confidence threshold to be met by looking for agreements, either through repeated computation over the same datapath (temporal redundancy) or duplicate datapaths (spatial redundancy), as shown in Figure 5.1. The confidence threshold can be adjusted based on the device error rate and the application requirement. For example,

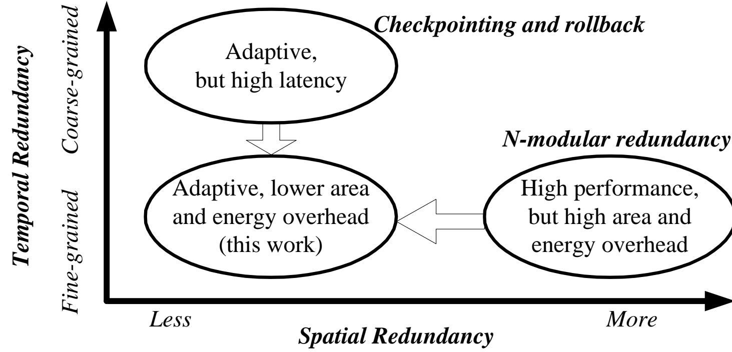


Figure 5.2: Combination of spatial and temporal redundancy for an efficient error-resilient computing system.

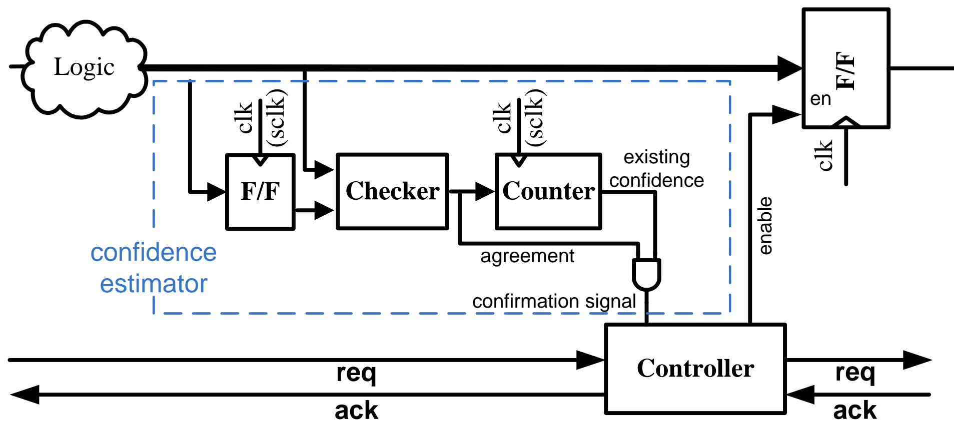


Figure 5.3: Block diagram of confidence estimator.

when the application-required reliability is high and the device error rate is high, the confidence threshold is raised to allow more repeated computations. The combination of spatial and temporal redundancy produces an adaptive resilient computing illustrated in Figure 5.2.

5.2 Confidence-driven computing

In the confidence-driven computing (CDC) model, a datapath is partitioned into segments where the probability of error of each segment is bounded. A confidence estimator is placed at the end of each segment to harden the output for forward propagation to the next stage. A confidence estimator consists of four components: a flip-flop, a checker, a counter and a small controller as shown in Figure 5.3. The confidence estimator samples the output of the datapath in every clock cycle and compares it with the previous sample stored in the flip-flop (through temporal redun-

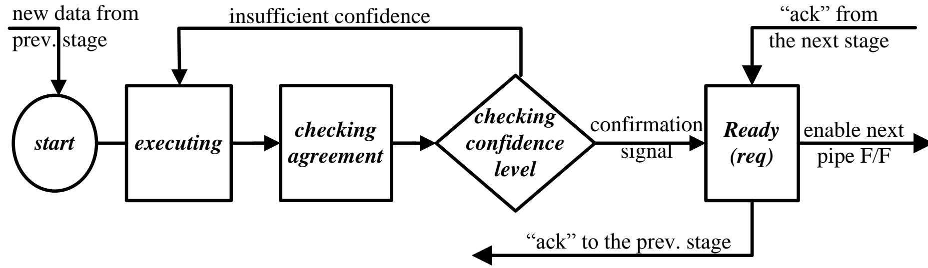


Figure 5.4: Flow chart of lockstep synchronization.

dancy). The checker performs the comparison and looks for either an agreement or a disagreement. The counter keeps track of the confidence level: the confidence level is raised upon an agreement and reset upon a disagreement. The controller ensures that the confidence level reaches a required threshold before allowing the output to be propagated through the main flip-flop by enabling the clock.

An n-bit counter is capable of tracking 2^n distinct confidence levels. The lowest confidence level indicates bypass. The controller generates handshaking signals to synchronize with its neighboring stages. Upon confirming the confidence level meeting the required confidence threshold, the controller sends a request (`req`) to the following stage. Once the following stage becomes ready to accept a new input, an acknowledgment (`ack`) is sent back and the controller then enables the main flip-flop to pass the current output to the next stage. We develop two synchronization schemes: lockstep synchronization and speculative synchronization, and a method to incorporate spatial redundancy. The operations are elaborated in the following subsections.

5.2.1 Lockstep synchronization

Lockstep synchronization guarantees the output of one stage to be hardened before it is propagated to the next stage. A slow stage, due to errors, holds back the output and leaves the neighboring stages waiting. The flow chart of lockstep synchronization is shown in Figure 5.4. The output is sampled and checked for agreement. The counter accumulates the confidence level until it reaches the confidence threshold, at which point the current stage enters the ready state and sends a `req` to the following stage. It waits until the following stage signals an `ack` and then enables the output to be propagated to the following stage. After the output exits the current stage, an `ack` is also passed to the previous stage.

The confidence threshold is the primary knob to tune the output reliability level.

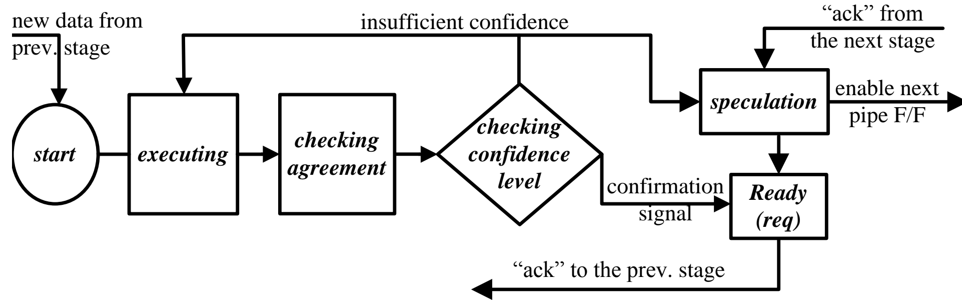


Figure 5.5: Flow chart of speculative synchronization.

A confidence threshold of 2 requires 2 agreements and a threshold 3 requires 3 agreements. Note that the confidence estimator looks for consecutive agreements in time, which is different from the majority voting scheme used in NMR. A disagreement resets the confidence level and restarts the confidence accumulation process. By setting higher threshold in the less reliable stages, we can avoid having less reliable stages dominate the overall system error rate and therefore improve the system reliability more effectively.

5.2.2 Speculative synchronization

Speculative synchronization allows an output to proceed to the next stage even if the confidence level has not reached the confidence threshold. Compared to the lockstep scheme, speculative execution shortens the latency and permits a higher throughput. The flow chart of speculative synchronization is shown in Figure 5.5. Under this scheme, the confidence estimators act as transparent gate keepers: tentative output is passed to the next stage when the next stage is ready, while the tentative output is still being hardened by the current stage. When the current stage finally reaches the confidence threshold, an `ack` is sent to the previous stage, indicating that it is ready to accept a new input. Note that to ensure data being hardened in the correct sequential order using the speculative synchronization, the confidence threshold in each stage needs to be set to no lower than the one in the previous stage.

Speculative synchronization cuts the idle cycles when one stage is complete and waiting for the previous stage to finish accumulating confidence. The scheme assigns tentative work to otherwise idle stages. Therefore, it improves the throughput and latency compared to the lockstep scheme. The speculative synchronization provides almost identical protection as the lockstep scheme, but the energy consumption is higher due to higher switching activity in speculative execution.

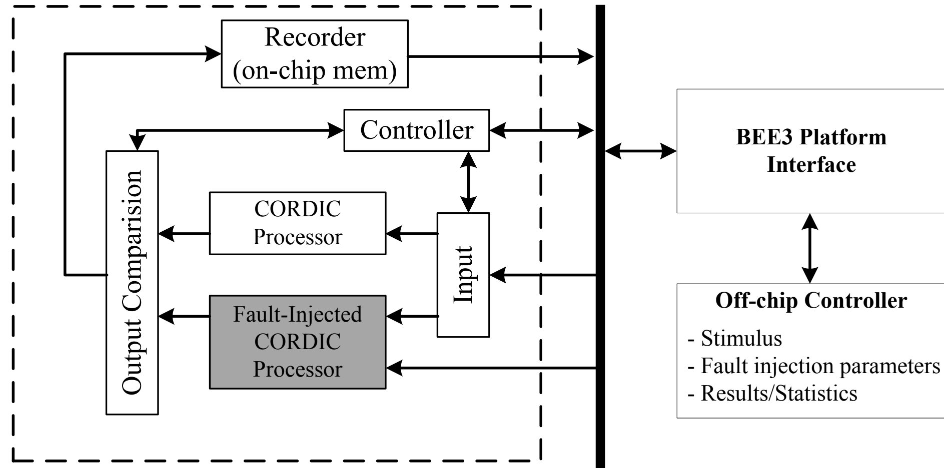


Figure 5.6: FPGA-based system emulation platform for quantitative evaluation.

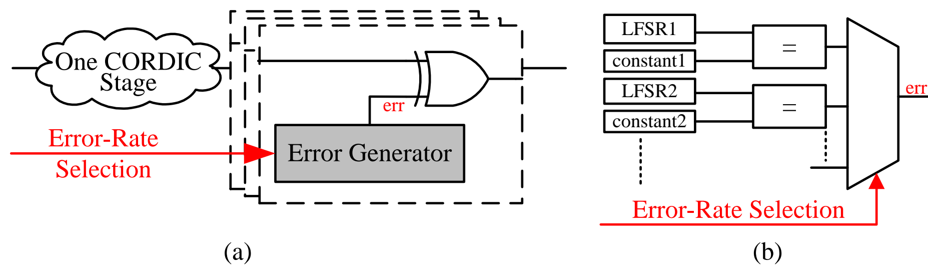


Figure 5.7: (a) Error injection mechanism, and (b) error generator.

5.2.3 Spatial redundancy

To speed up the accumulation of the confidence level for a higher performance, spatial redundancy can be incorporated in conjunction with temporal redundancy. The simplest way to include spatial redundancy is by providing a duplicate datapath in the form of dual modular redundancy (DMR). A duplicate datapath allows the confidence to be accumulated quickly, thus increasing throughput and minimizing latency. DMR is less expensive than TMR. Errors detected in DMR trigger recomputations for error correction in an iterative DMR method. The inter-stage synchronization is performed using either the lockstep or the speculative scheme described above.

5.3 Reliability and performance evaluation using sample-based emulation

To evaluate the reliability and performance of CDC, we propose an FPGA-based system emulation, which has been demonstrated to accelerate error simulation by up to six orders of magnitude [65]. Fast emulation permits reliability measurement down to low error rates that are relevant in practical systems, and bit- and cycle-accurate emulation also provides good performance measurements. Compared to other hardware emulators proposed in the past, our emulator uses real-time, on-FPGA error generation, instead of pre-stored error vectors or scan chains, hence a much higher emulation throughput is possible.

We use a coordinate rotation digital computer (CORDIC) processor as the test vehicle and the emulation platform is illustrated in Figure 5.6. The platform consists of an error-free 16-bit, 12-stage CORDIC processor as the reference and an identical, but fault-injected CORDIC processor protected by CDC. The entire system is mapped to a Xilinx Virtex-5 FPGA on a BEE3 platform [69]. The platform offers ample resources for evaluating complex designs.

An error injection block is added to the end of each CORDIC stage to simulate runtime errors, as shown in Figure 5.7(a). The error injection block inverts the output bits using XOR gates based on the error rate that is selected. Random errors are generated using a set of linear feedback shift registers (LFSR) by comparing their values to constants: if the LFSR values match the constants, bit errors are injected by inverting the bits. Since each LFSR produces 1 and 0 with nearly equal likelihood, the probability of error being generated is 2^{-x}, where x is the length of the constant. A set of maximal-length LFSRs of various lengths is used, each of which produces

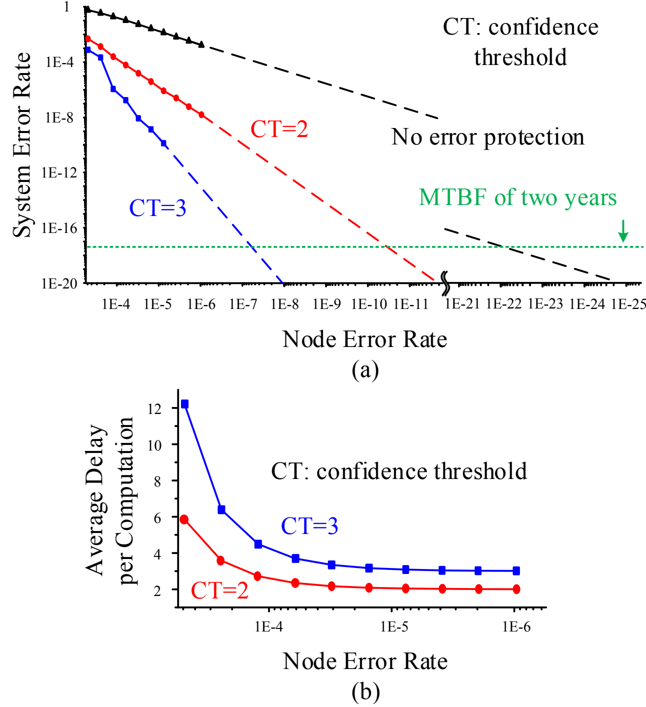


Figure 5.8: System error rate based on FPGA emulation results (solid line) and extrapolation (dash line), and (b) average delay per computation as the confidence threshold is adjusted.

a different error probability. An error rate selector, shown in Figure 5.7(b), sets the error rate by choosing one of the LFSRs. Using this platform, we were able to collect more than 1000 errors for a reliable estimate of the system error rate at 10^{-10} in one week.

5.3.1 Reliability evaluation

Emulation proves the effectiveness of CDC. The system error rate (the probability of an incorrect system output) as a function of the circuit node error rate (the probability of error of any circuit node) is plotted in Figure 5.8(a). Without any protection, the error rate of the CORDIC processor is three orders of magnitude higher than the node error rate due to the large number of circuit nodes that are subject to errors. With confidence estimators placed at the end of 6th stage and 12th stage of the CORDIC processor and as we increase the confidence threshold, the system error rate decreases by at least four orders of magnitude when the node error rate is at 10^{-5} or lower.

To translate the error rates to practical terms, if we were to guarantee a mean time between failure (MTBF) of two years for a 1GHz CORDIC processor, the required

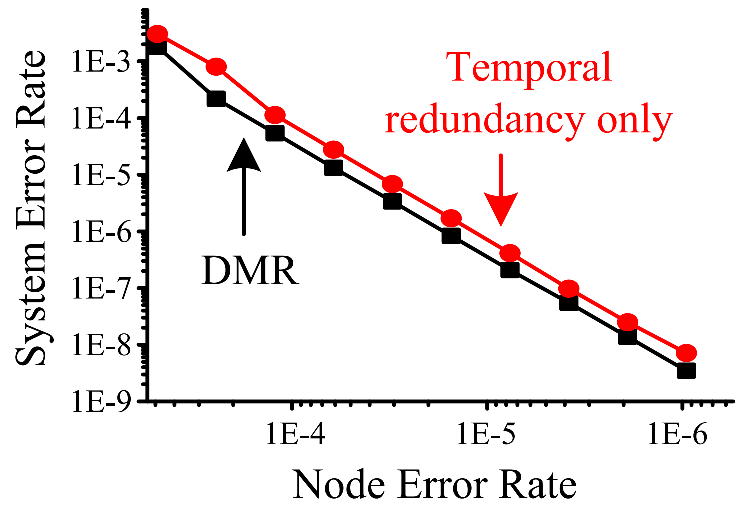
node error rate is 10^{-22} if no protection is used. This extremely low node error rate is possible with current mainstream CMOS technology but will likely become difficult with continued device scaling and the new generation of nano devices. By inserting confidence estimators, the required node error rate is relaxed to 10^{-11} with a confidence threshold of 2 and 10^{-8} with a threshold of 3. The threshold can be adjusted at runtime based on the underlying circuit error rate and the application requirement.

For a better protection, the DMR method can be applied. The improved reliability shown in Figure 5.9(a) is due to the difference between confidence accumulation policies: an error in DMR invalidates one pair of computations from both the primary and the duplicate datapath, thus the number of agreements needed on average to reach the confidence threshold is slightly higher. The DMR method is also capable of detecting permanent errors when a confidence estimator consistently reports disagreements and fails to meet the confidence threshold over a large number of clock cycles. An adaptive sparing method can be used to dynamically allocate additional spatial redundancy to overcome permanent errors.

5.3.2 Performance evaluation

A high confidence threshold provides better protection but it also decreases the throughput. Figure 5.8(b) shows the average delay per computation (the inverse of throughput): when the node error rate is moderate to low, a confidence threshold of 2 requires at least 2 clock cycles per computation and a threshold of 3 requires at least 3 cycles per computation. When the circuit node error rate is high, the delay becomes significant using a higher threshold. Frequent node errors slow down the process of gathering agreements. It is therefore more advantageous to operate CDC at the lowest confidence threshold that provides the necessary reliability. The runtime configurable confidence threshold accommodates device fluctuations and different reliability requirements among applications.

Confidence estimators can be placed in finer-grained intervals to reduce the throughput penalty. A fine-grained placement of confidence estimators shortens the path and bounds the probability of error, contributing to a faster convergence towards the required reliability. The sampling clock frequency can also be increased due to the shortened delay per stage. Figure 5.10 shows the improved average delay per computation as more stages of confidence estimators are inserted to the same CORDIC processor. The improvement becomes more significant at high circuit node error rates, thanks to the faster convergence towards the confidence threshold. The delay



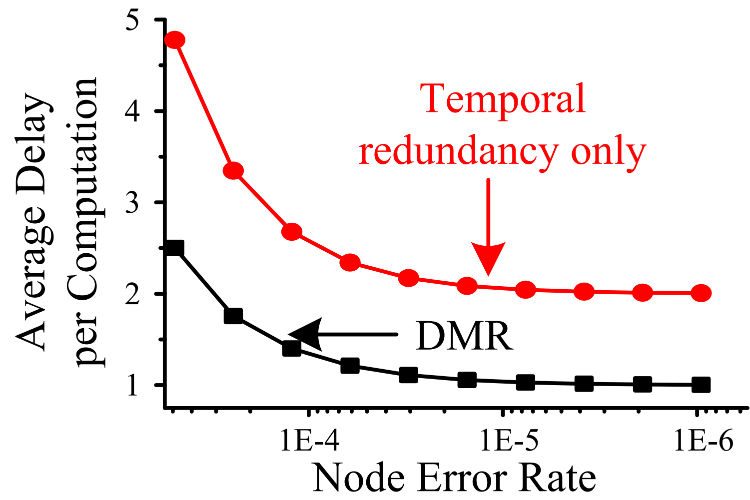


Figure 5.9: (a) System error rate and (b) average delay of the CORDIC processor using temporal redundancy only and DMR (assume a confidence threshold of 2).

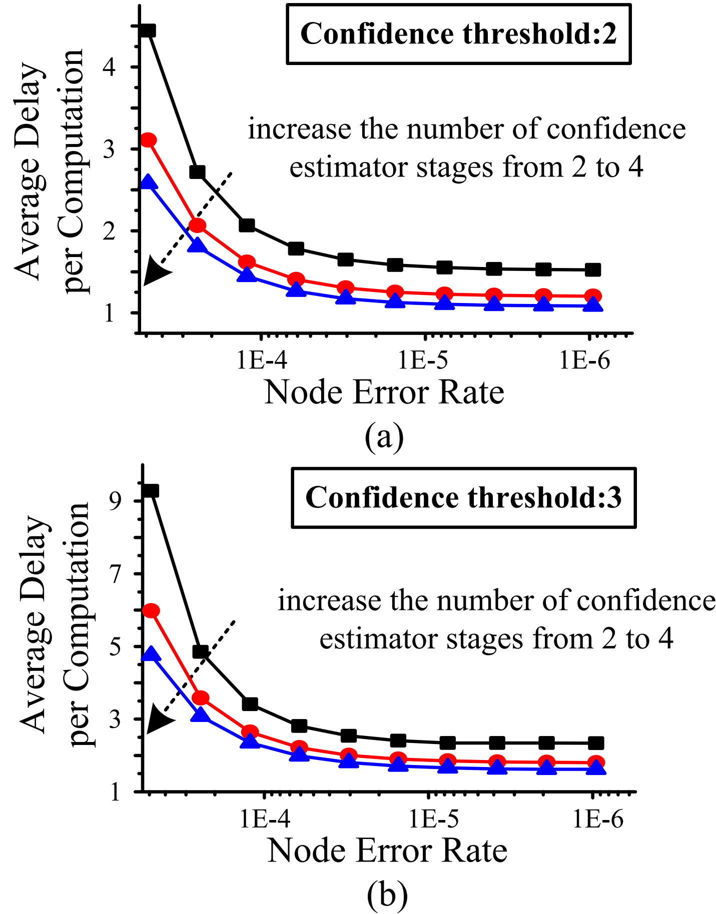
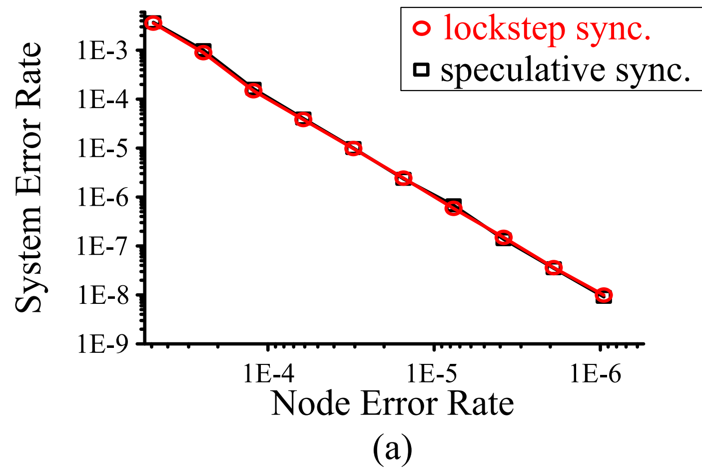


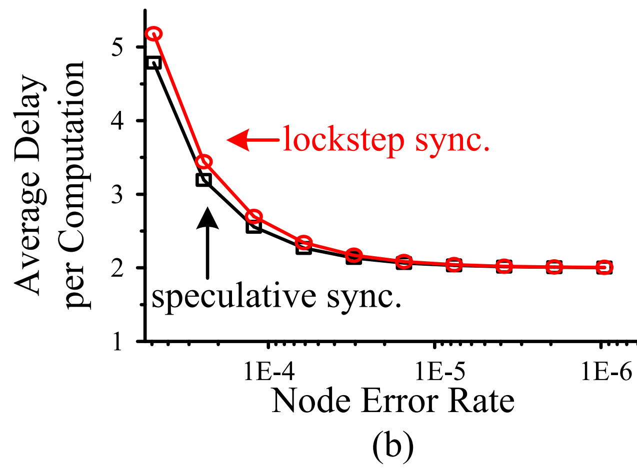
Figure 5.10: Average delay per computation improves with more fine-grained placement of confidence estimators: (a) at confidence threshold = 2, and (b) at confidence threshold = 3 when only temporal redundancy is considered.

improvement is also attributed to the increased clock frequency as more confidence estimator stages are inserted.

The above discussions are based on lockstep synchronization. The speculative synchronization further improves the throughput when the circuit node error rate is high, as shown in Figure 5.11(b). Moreover, the speculative scheme shortens the latency of computation: when the circuit node error rate is moderate to low, the latency of computation is almost independent of the confidence threshold because a tentative output is passed along without waiting, followed by parallel checking performed at all the stages. Therefore, the speculative synchronization is especially attractive for latency-sensitive applications.



(a)



(b)

Figure 5.11: (a) System error rate and (b) average delay of the CORDIC processor using lockstep and speculative synchronization.

Table 5.1: Area and energy of CDC with temporal redundancy only (CE: confidence estimator, CT: confidence threshold

	1-stg. CE	2-stg. CE	3-stg. CE	4-stg. CE
Norm. Clk Period	1.30	0.76	0.60	0.54
Norm. Area	1.08	1.21	1.34	1.46
Norm. Energy (CT=2)	1.14	1.20	1.36	1.53
Norm. Energy (CT=3)	1.15	1.23	1.46	1.64

Table 5.2: Area and energy of CDC with DMR

	1-stg. CE	2-stg. CE	3-stg. CE	4-stg. CE
Norm. Area	2.03	2.15	2.28	2.40
Norm. Energy	2.12	2.17	2.27	2.42

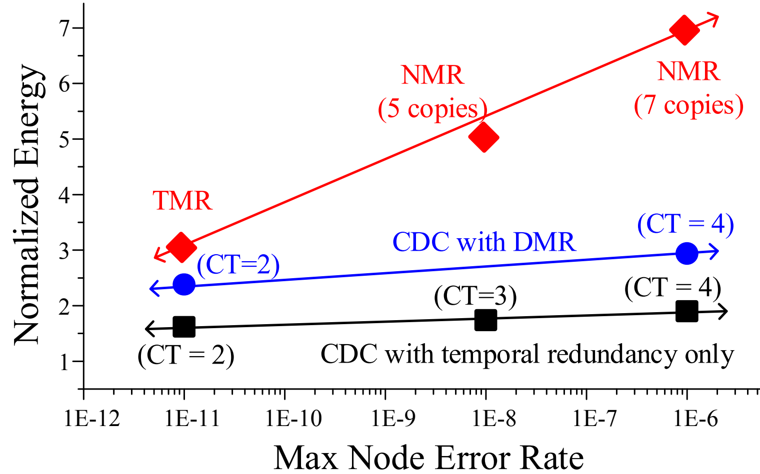


Figure 5.12: Normalized energy overhead of three schemes for a highly reliable general purpose processor that requires a MTBF of 2 years.

5.3.3 Implementation results and case study

We estimate the performance, area and energy of the CDC design by synthesizing it along with a CORDIC processor using a 45nm CMOS technology. Table 5.1 presents the comparison. Having one stage of confidence estimator increases the clock period by 30% and introduces an 8% area overhead. Additional stages of confidence estimators reduces the clock period, but the area overhead rises. However, there is a limit to how fine-grained the confidence estimators can be efficiently placed due to the diminishing improvement in throughput, and the escalating cost of area and energy. Design time decisions need to be made based on the expected range of circuit error rates along with the area and energy constraints imposed by the design. DMR provides better throughput, but the energy and area are 58% and 64% higher respectively by comparing the results in Table 5.2 and Table 5.1 (when using four stages of confidence estimators).

We show a case study based on a highly reliable general-purpose processor that requires two years of MTBF. Figure 5.12 shows the trade-off between the energy and the system reliability of CDC compared to the conventional NMR scheme. CDC guarantees a given system error rate of 10^{-17}, even when the underlying circuit node error rate fluctuates between 10^{-11} and 10^{-6}. The protection can be accomplished in several ways: temporal redundancy only or DMR with either lockstep or speculative synchronization. When circuit node error rate increases, the reliable system is achieved by setting a higher confidence threshold with a gradual increase of energy. In comparison, the energy of N-module redundancy is much higher.

5.4 Early checking

In CDC, error checking holds back the datapath. Errors increase the delay to reach the required confidence, resulting in an inefficient checking for short-duration errors, such as soft errors and errors due to coupling noise and voltage droop. To overcome this challenge, we propose an early checking (EC) technique, built on top of the CDC model. The EC technique exploits the delay slack in the vast majority of the datapaths for fast error checking and confidence accumulation.

We show in Figure 4.12 a realistic example of path delay profile of a synthesized 16-bit 12-step CORDIC processor assuming uniform random inputs. The delay profile can be modeled using a Gaussian distribution with mean of 0.75 and standard deviation of 0.08, normalized to the longest path delay. The clock cycle boundary is drawn at 1 to contain the longest path. Extra margins are often added in practice to accommodate process variation and runtime fluctuation. A study in [35] also shows that 70% of the datapaths in a commercial processor have at least 20% timing slack. Therefore, the confidence estimator can start sampling early, well before the clock edge, thus the name early checking.

5.4.1 Early checking

Using EC, glitches due to transient errors and soft errors will be detected and invalidated and a required confidence can be met before the clock edge without using an extra clock cycle. If the required confidence cannot be met by the clock edge due to long path delay and (or) errors, an additional clock cycle is needed. However, long path delay is statistically less likely, thus the performance penalty will remain low.

To implement EC, we decouple datapath from error checking and place them on separate clocks. Error checking will operate on a possibly faster sampling clock (`sclk`), so the confidence level can start accumulating early and quickly. The datapath samples the confidence level at the main clock (`clk`) and makes the propagation decision. EC can be illustrated using the same block diagram in Figure 5.3, except that the checking path runs on `sclk` instead of `clk`. The interface between the datapath and error checking is the confidence level – error checking accumulates the confidence level at the `sclk` frequency and the datapath inspects the confidence level at the `clk` frequency. It is in fact not necessary to have `sclk` and `clk` synchronized or phase-aligned, thus the `sclk` generation can be simplified or self timed using a small ring oscillator. The `sclk` of different stages do not need to be synchronized either, and `sclk` can also be shared among multiple stages to save cost.

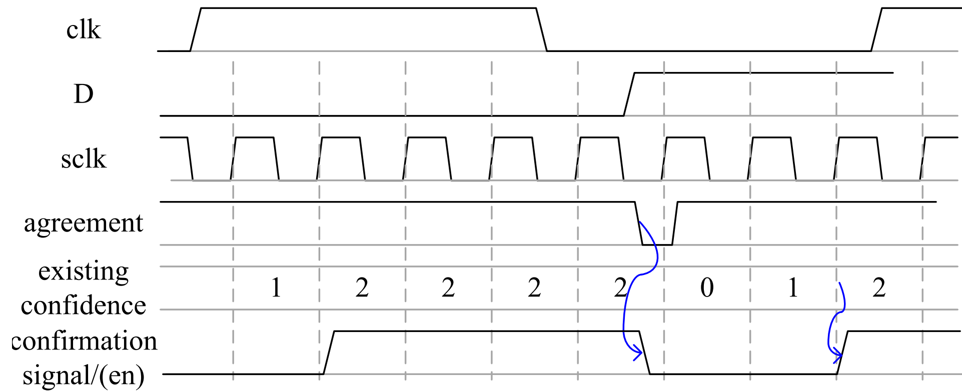


Figure 5.13: Conceptual timing diagram illustrating early checking.

5.4.2 Functional analysis and comparison

The conceptual timing diagram of EC is shown in Figure 5.13, where the sampling clock (`sclk`) runs independently of the main clock (`clk`) and the two are not necessarily phase-aligned. The confidence estimator starts accumulating confidence at the first possible `sclk` edge. If `sclk` is fast, the confidence level quickly reaches the threshold and saturates. After some time, D makes the final transition to the correct value. The transition is detected by the confidence estimator as a disagreement, which resets the confidence level to 0. The confidence estimator continues to sample D and looks for additional agreements. On the next edge of `clk`, the confidence level is inspected and as it matches the confidence threshold, the output of the current stage is allowed to propagate through the main flip-flop.

In an error-free operation, a late arriving output will be treated as tentative if an insufficient confidence level is accumulated before the `clk` edge, and the output will be held in the current pipeline stage for one more cycle. The probability of holding depends on how often the long paths are exercised. This probability can be very low for delay distributions with long tails, as in the case of deep submicron circuits running at a reduced supply voltage [73]. We expect that the performance loss will be low if the critical paths are only infrequently exercised and the confidence level is relatively low.

The effect of a transient fault or soft error depends on its arrival time and duration. If an error arrives before the output makes the final transition, it is not different from the error-free case discussed above. If an error arrives at the same time or after the output makes the final transition, it will prolong the final confidence accumulation and increase the probability of holding. A high error rate or long error duration will

Table 5.3: Comparisons of error detection techniques with short checking durations

Technique	RAZOR [5]	BISER [33]	CDC with Early Checking
Mechanism	post-edge checking	pre-edge checking	multiple checkings before clock edge
Protection level	fixed in design time	fixed in design time	tunable by changing confidence threshold
Area overhead	small	small	moderate
Performance impact	small	larger impact with longer checking window	depends on confidence threshold and runtime delay distribution

further increase the holding probability, leading to performance loss.

The EC technique can be compared to other common approaches such as RAZOR and BISER that also target errors of short duration. The comparison is shown in Table 5.3. RAZOR double samples and looks for an agreement. It introduces a small area overhead, and the impact on performance is small. However, the protection level is dictated by the checking duration and it is fixed during design time. BISER has been presented to monitor errors by detecting suspicious transitions. Its area overhead is small but the impact on performance is decided by the checking duration. A longer checking duration results in more performance degradation. Similar to RAZOR, the checking duration of BISER is fixed and cannot be changed to adapt to various protection levels. In contrast, the EC technique offers a tunable protection level by setting the confidence threshold. The impact on performance depends on the confidence threshold and the runtime delay distribution.

5.5 Reliability and performance evaluation using event-based simulation

The effectiveness of a resilient design is highly dependent on the complex interplay of error events (error occurrence and duration), path events (path start and end) and the error protection mechanism. The FPGA-based emulation is cycle based and events can only occur at clock cycle boundaries. Such a platform is sufficient for a sample-based technique such as CDC described above. However, the EC technique is more fine-grained, and it requires a transient simulator for the quantitative evaluation.

To speed up the transient simulation, we propose an FPGA-based event simulator. The simulator consists of path delay model, error model and the EC technique to be tested. The simulation operates on the FPGA system clock. The main clock period, sampling clock period, path delays and error durations are in units of the FPGA system clock period T_{sys}. For example, we can set the main clock period $T_{\texttt{clk}}$ =

$1000\mathrm{T}_{sys}$, and the sampling clock (`sclk`) period $\mathrm{T}_{\texttt{sclk}}$ is set independently, e.g., $\mathrm{T}_{\texttt{sclk}} = 100\mathrm{T}_{sys}$. The path delay model is implemented as a random number generator based on its statistical distribution (e.g., Figure 4.12). The transient and soft error models are implemented similarly.

The simulation is conducted in steps of T_{sys}. In each T_{sys}, the simulation controller determines whether the path under test gives a valid output based on the time elapsed, delay generated by the delay model, and error occurrence and duration given by the error model. The confidence estimator samples the output at `sclk`, and increments or resets the confidence level. The confidence level is sampled at `clk` to decide whether to propagate the output or to hold it for one more cycle.

The FPGA simulation platform operates at a frequency of 100MHz or higher, thus a 100kHz simulation throughput is possible if $\mathrm{T}_{\texttt{clk}}$ is set to $1000\mathrm{T}_{sys}$. This setup allows us to collect about 20 errors in two weeks for an estimate of the system error rate at 10^{-10}. A finer time resolution is possible, but the simulation throughput will be reduced proportionally.

5.5.1 Error simulation

A transient error flips the validity of the path output (from valid to invalid) and causes an error. Both the error arrival time and duration are tunable in T_{sys} steps. Figure 4.9 shows the construction of the error model. In each T_{sys} step, it produces a conditional error using an LFSR. To model error duration, a counter is started upon error assertion to keep the error for the error duration. The error duration can also be randomly generated.

Figure 5.14 illustrates the timing of the FPGA-based event simulation platform. All events are lined up with the FPGA system clock. Error events can be programmed to model their natural occurrence and duration. The simulation platform is general purpose and not tied to any circuit implementation. It provides a high simulation throughput and can be easily extended to other related work.

5.5.2 Experimental evaluation

To set up the experiment, we set $\mathrm{T}_{\texttt{clk}} = 1000\mathrm{T}_{sys}$, and created a Gaussian path delay model following the path delay distribution shown in Figure 4.12, with a mean and standard deviation set to $0.8\mathrm{T}_{\texttt{clk}}$ and $0.0625\mathrm{T}_{\texttt{clk}}$. The tail of the distribution beyond 3.2σ is saturated to meet $\mathrm{T}_{\texttt{clk}}$.

We first evaluate the performance of EC under error-free operations. Table 5.4

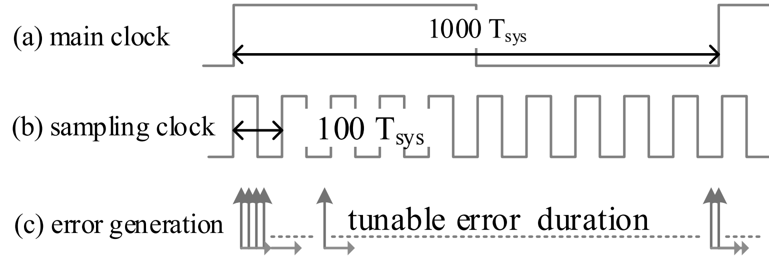


Figure 5.14: Timing charts of the FPGA-based event simulation: (a) main clock, (b) sampling clock, and (c) error generation.

Table 5.4: Normalized throughput of early checking in error-free operations

	CT=1	CT=2	CT=3	CT=4
$T_{\mathtt{sclk}}$=0.05$T_{\mathtt{clk}}$	0.997	0.965	0.841	0.683
$T_{\mathtt{sclk}}$=0.1$T_{\mathtt{clk}}$	0.981	0.723	0.510	0.503

shows the normalized throughput. In an error-free operation, the throughput is determined by the length of the error detection window, or the product of the sampling clock period $T_{\mathtt{sclk}}$ and the confidence threshold. Increasing the confidence threshold prolongs the error detection window, degrading the throughput due to long paths that need extra time to meet a required threshold. Nevertheless, the throughput can be improved by shortening $T_{\mathtt{sclk}}$. For example, at a confidence threshold of 3, shortening $T_{\mathtt{sclk}}$ from $0.1T_{\mathtt{clk}}$ to $0.05T_{\mathtt{clk}}$ improves the throughput by 65%.

As errors are injected to the system, the reliability starts to degrade. Figure 5.15 shows that the CORDIC processor's error rate increases with transient and soft error rate and duration. Figure 5.16 shows the effect of adjusting the confidence threshold while holding $T_{\mathtt{sclk}}$ and the transient error rate constant. As the confidence threshold is raised by 1, the system reliability is improved by three orders of magnitude or more. However, it is important to notice from Figure 5.16 that the sampling clock period $T_{\mathtt{sclk}}$ should be chosen to match or exceed the error duration T_{err} to guarantee a minimum error detection window for an appreciable improvement in reliability.

Figure 5.17 shows the effect of errors of long and short duration. If errors last for a relatively long duration, e.g., $T_{err} = 0.1T_{\mathtt{clk}}$, it is necessary to select the sampling clock period $T_{\mathtt{sclk}} \geq T_{err}$ for a more effective improvement of the system reliability. On the other hand, if errors are known to be very short, e.g., $T_{err} = 0.01T_{\mathtt{clk}}$, the length of $T_{\mathtt{sclk}}$ has little impact on the system reliability. The throughput is given in Figure 5.18. A long $T_{\mathtt{sclk}}$ or a higher confidence threshold degrades the through-

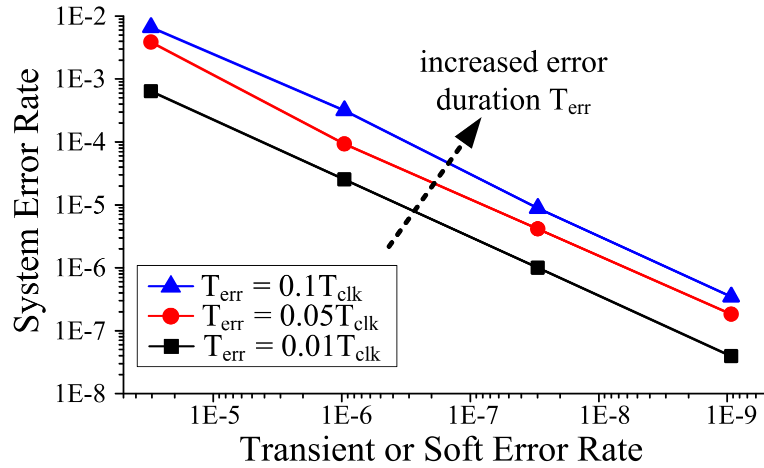


Figure 5.15: System error rate without any error protection (error duration T_{err} is normalized to $T_{\texttt{clk}}$).

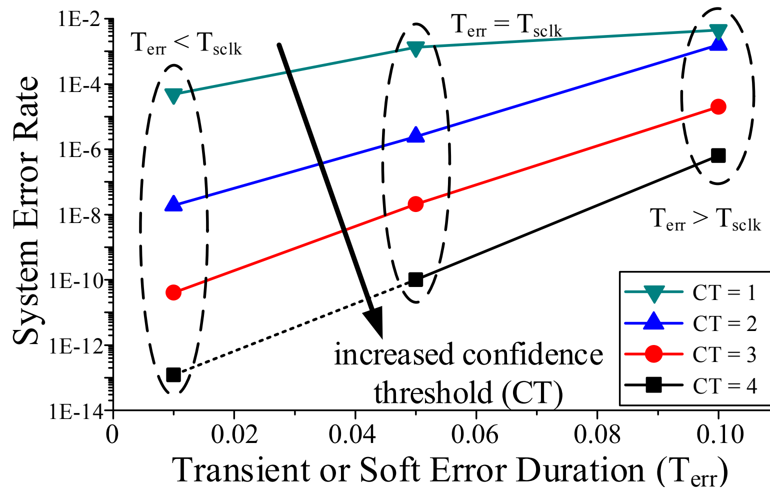


Figure 5.16: System error rate as the confidence threshold is adjusted (assume a transient or soft error rate of 10^{-5} and $T_{\texttt{sclk}} = 0.05T_{\texttt{clk}}$. Error duration T_{err} is normalized to $T_{\texttt{clk}}$). FPGA simulation results (solid line) and extrapolation (dotted line).

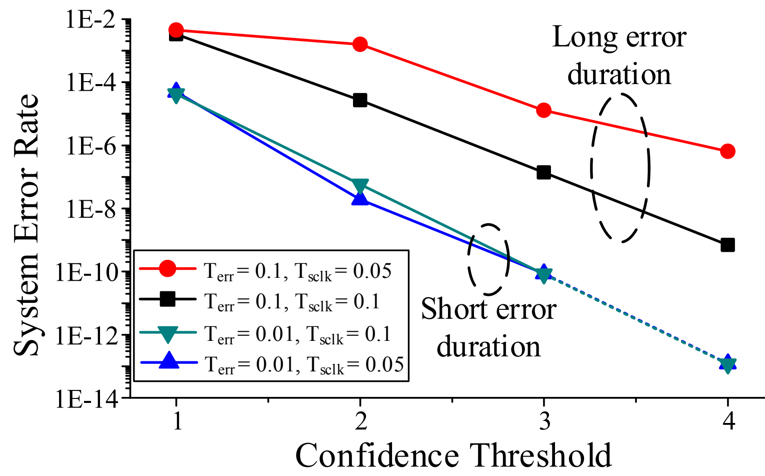


Figure 5.17: System error rate versus confidence threshold (assume a transient (or soft) error rate of 10^{-5}). FPGA simulation results (solid line) and extrapolation (dotted line).

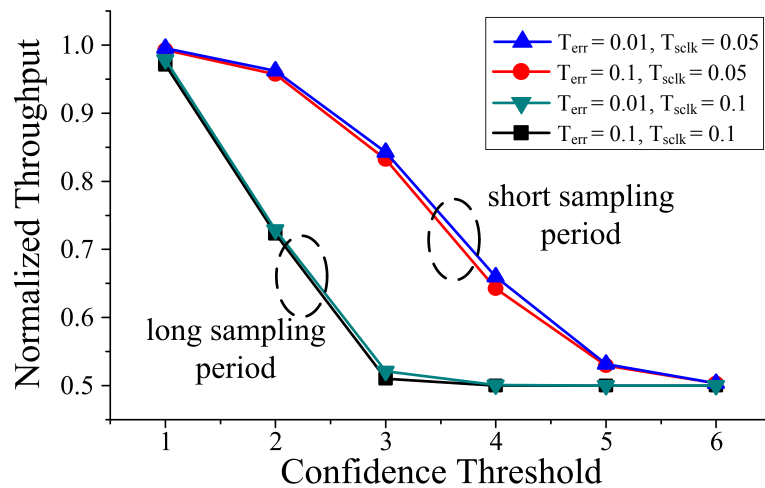


Figure 5.18: Normalized throughput versus confidence threshold (assume a transient (or soft) error rate of 10^{-5}).

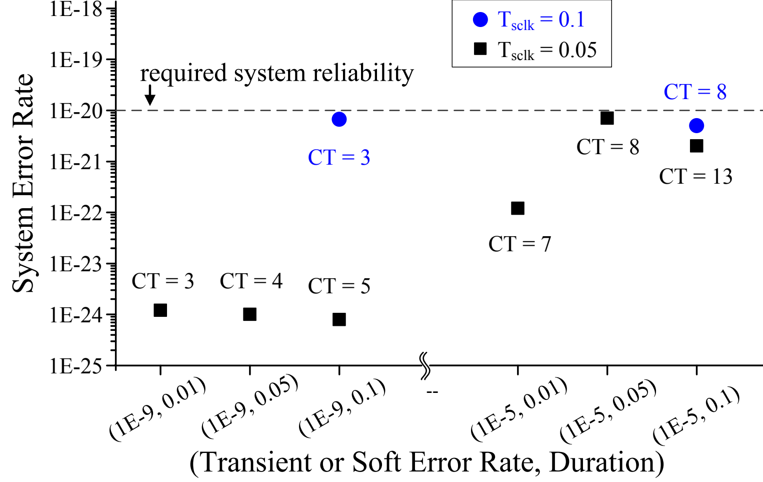


Figure 5.19: Adjustment of confidence threshold and sampling clock period to deliver the required system error rate of 10^{-20}.

Table 5.5: Normalized energy of system based on early checking

	CT=1	CT=2	CT=3	CT=4
Norm. Energy	1.191	1.203	1.210	1.222

put. Interestingly, the normalized throughput saturates to approximately 0.5, as one additional (main) clock cycle provides more than sufficient time for accumulating a high enough confidence using EC.

To summarize the results, the throughput of a system is determined by the length of the error detection window, or the product of $T_{\texttt{sclk}}$ and confidence threshold. A short window enables a higher throughput, as less time is needed to attain the required confidence. For a good protection, $T_{\texttt{sclk}}$ needs to match or exceed T_{err}. Therefore, we formulate a two-step design strategy: (1) tune $T_{\texttt{sclk}}$ to match the expected error duration, and (2) set the confidence threshold to obtain the required system error rate.

5.5.3 Implementation results and case study

A 16-bit, 12-step CORDIC processor is synthesized in a 45nm CMOS technology. The area overhead of implementing EC is only 12%. The energy overhead is as low as 20% at a low confidence threshold, and it increases marginally with a higher confidence threshold as seen in Table 5.5.

We formulate a case study based on a general purpose processor that requires a very low system error rate of 10^{-20}. Transient and soft error rates are assumed

to vary between 10^{-9} and 10^{-5}, and the error duration varies between $0.01\text{T}_{\texttt{clk}}$ and $0.1\text{T}_{\texttt{clk}}$. Resilient computing can be accomplished by tuning the confidence threshold and sampling clock period $\text{T}_{\texttt{sclk}}$, the results of which are shown in Figure 5.19. For example, when the transient (or soft) error rate is 10^{-9}, the confidence threshold can be set to 3, 4 or 5 at $\text{T}_{\texttt{sclk}} = 0.05\text{T}_{\texttt{clk}}$ to accommodate error durations ranging from $0.01\text{T}_{\texttt{clk}}$ to $0.1\text{T}_{\texttt{clk}}$. The long error duration of $0.1\text{T}_{\texttt{clk}}$ warrants an increase of $\text{T}_{\texttt{sclk}}$ to $0.1\text{T}_{\texttt{clk}}$ and the confidence threshold can be reduced to 3 while still satisfying the system error rate requirement. A higher transient (or soft) error rate of 10^{-5} requires a higher confidence threshold of 7, 8 and 13 for error durations ranging from $0.01\text{T}_{\texttt{clk}}$ to $0.1\text{T}_{\texttt{clk}}$ to meet the system reliability.

5.6 Summary

We present a confidence-driven computing (CDC) model to protect the circuits built on highly variable and nondeterministic devices. The reliability is enhanced by confidence estimators that rely on either temporal redundancy only or the combination of temporal and spatial redundancy. The area and energy cost can be kept low by using temporal redundancy only, while the combined temporal and spatial redundancy provides a higher throughput. The two methods follow one of the two synchronization schemes: lockstep or speculative. The speculative scheme permits a lower latency and higher throughput with a small increase in energy.

The CDC model uses cycle-based checking targeting non-deterministic nano devices like memristors. To achieve a better performance for deeply scaled CMOS circuits that are mostly affected by transient faults and soft errors, we present a variation of CDC using early checking (EC). The EC technique recycles unused cycle time to accumulate confidence level, allowing fast response to randomly occurring errors of short duration.

The CDC model is emulated on an FPGA platform with real-time error injection to prove its effectiveness. Quantitative evaluations of reliability and performance shed light on the choice of confidence threshold, placement of confidence estimators, and synchronization.

To evaluate the performance of the EC technique with transient errors, an FPGA-based event simulator is presented to incorporate both path delay model and error model at much finer time scale. The simulation captures complex interactions between path delays, errors and protection scheme. The EC technique is demonstrated to be highly effective against short transient errors. It improves the system reliability by

more than four orders of magnitude if the errors are of short duration, while the performance loss is kept as low as 0.3%.

The CDC model and the EC technique are promising solutions to bridge the gap between different applications and fluctuating device behavior in deeply scaled CMOS and future device technologies. The design insights will allow us to construct a reliability diverse computer architecture with computing elements that provide a range of reliability levels at appropriate energy cost to deliver the required performance.

CHAPTER VI

Low-Power Error-Resilient Systems for Wireless Communication

Designing low-power resilient systems can effectively leverage application-specific algorithmic approaches. To explore design opportunities in the algorithmic domain, an application-specific detection and decoding processor for multiple-input multiple-output (MIMO) wireless communication is investigated. In this chapter, we utilize the algorithmic resilience and co-designing with circuit and architecture to improve the area, throughput, and energy efficiency. The concept is demonstrated on an application-specific detection and decoding processor for the MIMO system.

We review the MIMO system and conventional detection algorithms in Section 6.1, and evaluate the state-of-the-art iterative detection-decoding algorithms in Section 6.2. From Section 6.3 to 6.5, we focus on the algorithmic optimization and hardware implementation. Silicon measurement from the fabricated test chip in 65nm CMOS is summarized in Section 6.6. Our design achieves higher throughput and energy efficiency over the state-of-the-art, and it maintains a good packet error rate performance, demonstrating the advantage of multilevel optimization techniques.

6.1 Background

Advanced wireless communication standards such as IEEE 802.11n/ac and 3GPP LTE Advanced Release 10/11 [80] adopt MIMO communication to increase spectral efficiency and data rate. For example, IEEE 802.11n allows for up to a 4x4 antenna configuration (4 transmit and 4 receive antennas); IEEE 802.11ac calls for up to an 8x4 configuration and 3GPP LTE Advanced release 10 [80] specifies up to an 8x8 antenna configuration. The enhancement in spectral efficiency and higher data rate comes at a significant computational cost: work load profiling indicates that MIMO

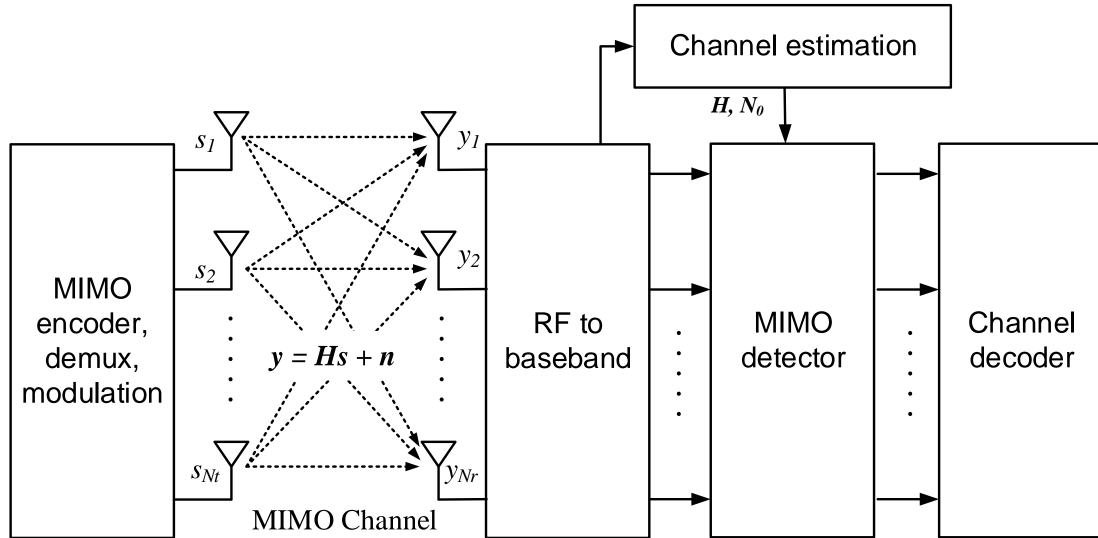


Figure 6.1: MIMO system overview

detection at the receiver can consume a large amount of the computing cycles in the physical layer baseband processing [81].

Figure 6.1 shows the system architecture for a generic MIMO system. At the transmitter, a coded data stream is demultiplexed onto multiple transmit antennas. The data streams are modulated onto constellations, such as quadrature amplitude modulation (QAM). The use of multiple transmit channels allows the MIMO system to achieve high data rates using only a limited spectrum. At the receiver, the signals are processed by the analog frontend at each receive antenna. The digital frontend performs carrier frequency offset compensation, synchronization, and equalization. The MIMO detector cancels the interference between different receive channels before further processing by the channel decoder.

6.1.1 MIMO system model

A MIMO system with N_t transmit antennas and N_r receive antennas, operating in a symmetric M-QAM scheme, with $log_2 M$ bits per symbol is modeled by:

$$\mathbf{y} = \mathbf{HS} + \mathbf{n}, \tag{6.1}$$

where $\mathbf{y}$ is the received signal vector, $\mathbf{H}$ is the channel matrix, $\mathbf{S}$ is the transmitted symbol vector, and $\mathbf{n}$ is channel noise.

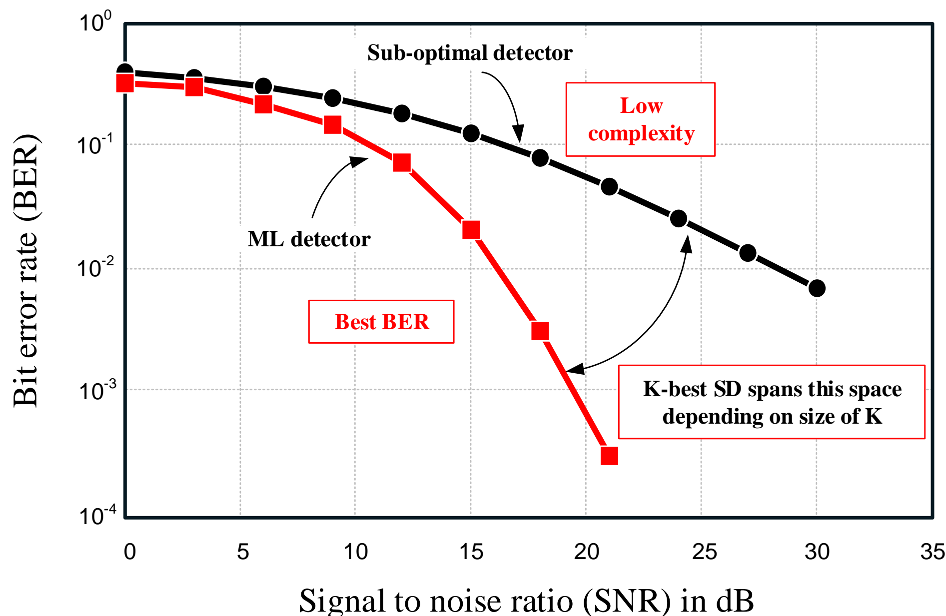


Figure 6.2: Conventional MIMO detection algorithm (hard outputs) BER performance tradeoffs

6.1.2 Non-iterative MIMO detection algorithms

In a conventional non-iterative detecting-decoding scheme in Figure 6.1, a maximum likelihood (ML) detection algorithm achieves the minimum bit-error rate (BER) [82]. The ML detector estimates the transmitted signal by solving

$$\widehat{\mathrm{D}} = \min_{\widetilde{S} \in Z_{N_t}} ||\mathbf{y} - \mathbf{H}\widetilde{\mathbf{S}}||^2 \tag{6.2}$$

where $\widehat{\mathrm{D}}$ represents the Euclidean (L2) distance that needs to be minimized, and $\widetilde{\mathbf{S}}$ is the estimated transmitted symbol vector. Eq. 6.2 is generally non-deterministic polynomial hard (NP-hard) and represents the closest point problem [83] which is a search through the set of all possible lattice points for the global best in terms of distance between the received signal $\mathbf{y}$ and $\mathbf{H}\widetilde{\mathbf{S}}$. Effectively, eq. 6.2 describes a tree search problem where each of the constellation points in Z_{N_t} is evaluated to find the path through the tree that minimizes the total error accumulated from each estimated symbol. Thus, the goal is to find the closest vector $\widetilde{\mathbf{S}}$ to the original transmitted symbol vector $\mathbf{S}$ for a received vector $\mathbf{y}$.

Several algorithms have been proposed to address the complexity of MIMO detection in the receiver, offering different tradeoffs between power and performance. Table 6.1 gives an overview of conventional MIMO detection techniques and qualita-

Table 6.1: MIMO detection algorithm (hard outputs) design tradeoffs

Detector type	BER performance	Computational complexity	In a large MIMO system
Optimal detectors Maximum likelihood (ML) Sphere decoder (SD)	Optimal	Exponential	Complex and expensive
Sub-optimal detectors Zero forcing (ZF) MMSE V-BLAST SIC	Suboptimal	Linear, polynomial	Efficient but inaccurate
Near-ML detectors SD with termination K-best SD Markov chain Monte Carlo	Near optimal	Polynomial	Feasible

tive tradeoffs. Among the MIMO detection techniques listed in Table 6.1 and shown in Figure 6.2, the ML detector minimizes the uncoded BER performance through exhaustive search, but the complexity grows exponentially with increasing number of antennas and the modulation order [84]. In contrast, linear detectors such as zero-forcing and MMSE detectors and successive interference cancellation (SIC) detectors have polynomial complexity, but they suffer from poor BER. Markov chain Monte Carlo methods [85] perform well in low SNR channel conditions but exhibit poor performance in high SNR channels. The sphere decoding (SD) based detector using depth-first search [86] without termination criteria results in optimal BER as in the case of an ML detector, but the complexity and power are prohibitive, and it may never reach its solution in bounded time. SD using breadth-first search with termination criteria can achieve near-optimal performance with polynomial computational complexity; however, such detectors may still require high area and power as antenna configuration increase. In [81], we present an implementation of SD detection with breadth-first search and limit the number of candidates evaluated in each step to predifined K candidates with minimum partial distance. The adaptive K-Best detection achieves a very high throughput of 3.2 Gb/s in an Intel 22nm tri-gate CMOS technology and a low energy per bit ranging from 14 to 44 pJ/bit.

K-Best detection achieves high throughput by only producing hard output, constraining itself to be used in conjunction with a hard decoder, the performance of which is much inferior to a soft decoder. While most non-iterative algorithms provide

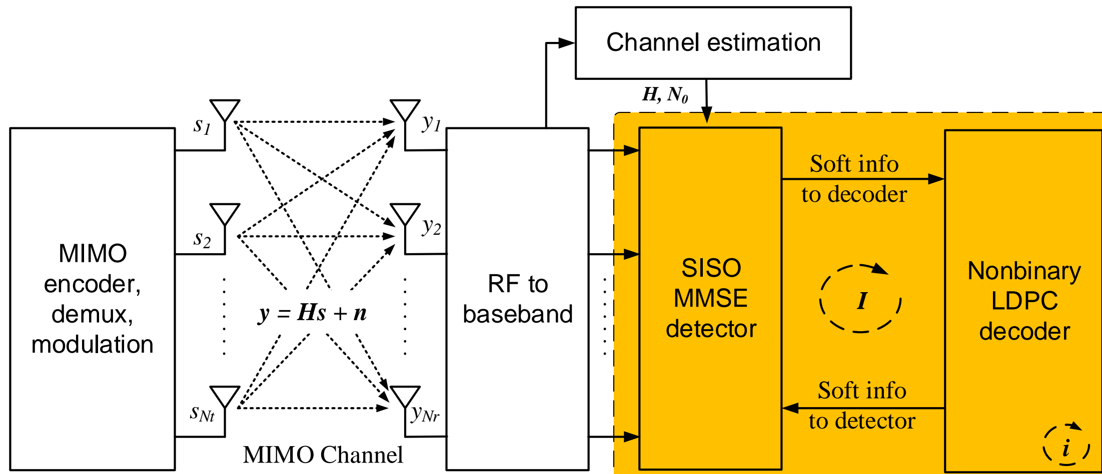


Figure 6.3: MMSE block diagram with iterative detection-decoding design.

Table 6.2: SISO MIMO detection algorithm tradeoffs

Detector type	PER performance	Computational complexity
SISO SD	Better	Increase with modulation order Exponential to antenna config.
SISO MMSE	Close to SD with more iter.	Constant with modulation order Polynomial to antenna config.

only hard outputs, SD with depth-first search can be re-designed to generate soft outputs at the cost of high complexity and low throughput [87].

6.1.3 Iterative MIMO detection algorithms

The latest MIMO wireless systems have adopted iterative detection and decoding (IDD) as shown in Figure 6.3. Compared to the hard output MIMO detectors mentioned earlier, an IDD system requires a soft-in soft-out (SISO) detector to cancel interference, and a SISO forward error correction (FEC) decoder to remove errors. The two blocks exchange soft information to improve the packet-error rate (PER) of packets of coded information iteratively. State-of-the-art IDD circuit designs based on SISO sphere decoding (SD) and low-density parity-check (LDPC) FEC have been demonstrated in [88,89] for up to 4x4 64-QAM MIMO systems. Compared to an SISO SD detector [88–90], an SISO minimum mean square error (MMSE) detector [91,92] has been demonstrated to have a lower complexity and can be efficiently scaled to

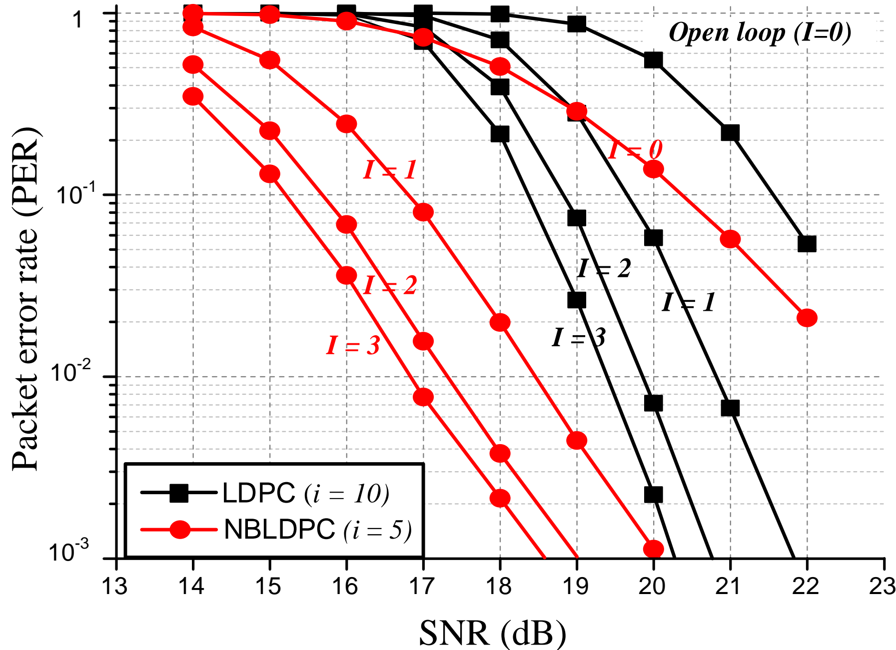


Figure 6.4: PER comparison between an MMSE-LDPC IDD design and MMSE-NBLDPC GF(16) IDD design under a 4x4 256-QAM MIMO system in the TGn type C channel model. The code length is 640 bits with 1/2 code rate for each decoder. Each 256-QAM symbol is mapped to two 4-bit symbols for GF(16) decoding.

support a larger antenna array and higher modulation order. Performance and complexity tradeoffs of SISO SD detection and SISO MMSE detection are summarized in Table 6.2.

6.2 Iterative detection-decoding design overview

Expanding antenna configuration and scaling modulation order are the most effective techniques to enhance the data rate and spectral efficiency. As antenna configuration and modulation order increase, however, noise and interference effects worsen, degrading PER and reliability. The fundamental challenges in designing an IDD detector is how to scale to large antenna configuration and modulation order while maintaining a good PER performance with each additional outer iteration ($\boldsymbol{I}$). As the complexity of SISO SD detector scales exponentially to the antenna configuration and modulation order, SISO MMSE detector has been demonstrated to have a higher throughput over SISO SD detectors and can be more easily scaled for large MIMO

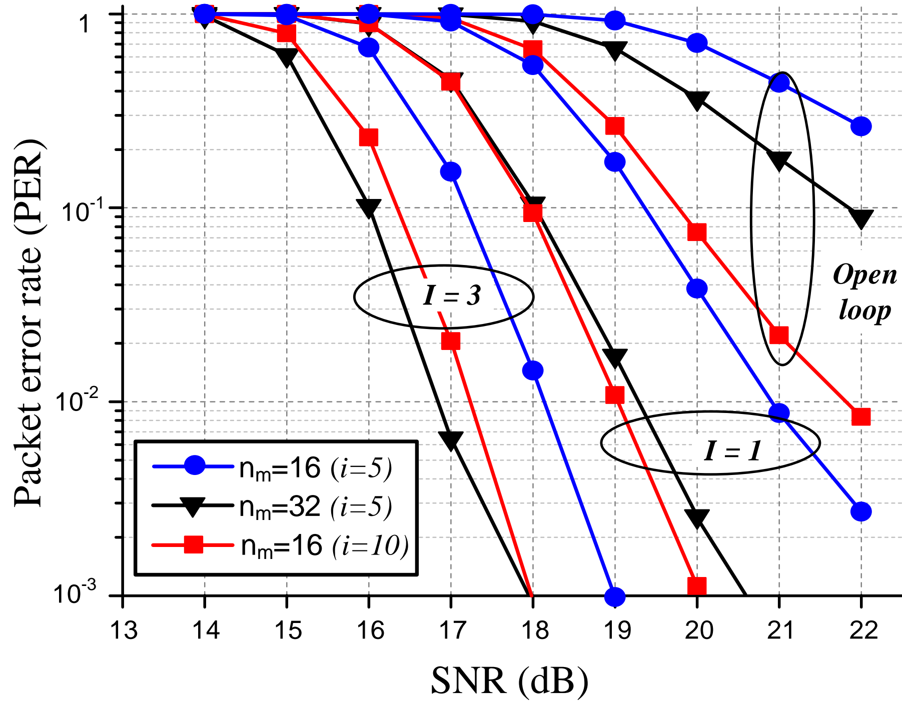


Figure 6.5: PER comparison for MMSE-NBLDPC GF(256) under a 4x4 256-QAM MIMO system in the TGn type C channel model. The code length is 1280 bits with 1/2 code rate for each decoder. Each 256-QAM symbol is mapped to an 8-bit symbol for GF(256) decoding. NBLDPC decoders are implemented based on EMS with n_m=32 and n_m=16.

system [92, 93].

A SISO decoder is an equally important part of an IDD design. Recent IDD designs have adopted LDPC codes and used LDPC decoders [88–90]. Compared to binary LDPC codes, nonbinary LDPC (NBLDPC) codes defined over Galois field (GF) outperform binary LDPC codes of comparable block length in coding gain [94], and it has been shown that NBLDPC codes significantly improve the detection-decoding performance over binary LDPC codes [95].

In this work, we propose a novel interface between an SISO MMSE detector and an NBLDPC decoder, and connect them together in an iterative MMSE-NBLDPC IDD design. The PER performance and the coding gain are evaluated in Figure 6.4 for a MMSE-NBLDPC IDD and a MMSE-LDPC IDD. The NBLDPC code and the binary LDPC code share the same block length of 640 bits and the same code rate of 1/2. The NBLDPC code is defined over GF(16). In a 4x4 256-QAM MIMO system operating under the TGn type C channel model [96], Figure 6.4 suggests that the

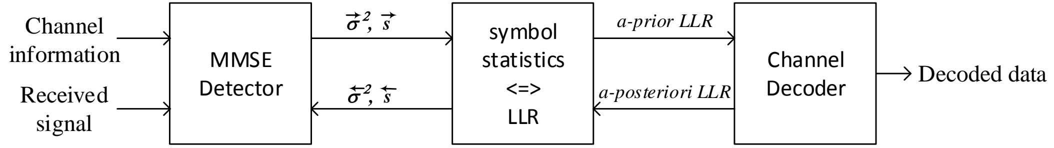


Figure 6.6: Soft information exchanged in MMSE-LDPC IDD design: a detector computes input soft symbols $\overleftarrow{s}$ and variances $\overleftarrow{\sigma}^2$ based on a-posteriori LLR delivered from a decoder, and generates a-prior LLR for the decoder from its output soft symbols $\overrightarrow{s}$ and variances $\overrightarrow{\sigma}^2$.

MMSE-NBLDPC (5 inner iterations) scheme has clear advantage in PER in both open loop and close loop, even with fewer decoding iterations (i=5 for the NBLDPC decoder compared to i=10 for the LDPC decoder).

Despite NBLDPC decoder's higher complexity, efficient approximate decoding [94] is well suited for a high GF computation. For instance, using the truncated extended min-sum (EMS) decoding algorithm, only the top n_m candidates, $n_m < q$ in GF(q), with high likelihood, instead of all q GF elements, are processed, eliminating those less likely elements during each single inner iteration. The complexity reduction has a direct consequence on area efficiency and energy consumption in the decoder. We show in Figure 6.5 the simulation results of a 4x4 256-QAM point mapped to a GF(256) symbol. As the results indicate the loss of PER performance due to the reduction of n_m can be compensated, or even better, by increasing the number of inner iterations.

In the rest of this chapter, we first review the latest SISO MMSE algorithm, and then demonstrated our proposed architecture and algorithm optimizations for the SISO MMSE detector and the nonbinary interface.

6.3 SISO MMSE algorithm and optimization

SISO MMSE interference cancellation algorithm was first proposed in [91]. In the first open loop operation, an SISO MMSE detector performs conventional MMSE filtering but generates a-prior log-likelihood ratio (LLR) for a SISO decoder from its output soft symbols and variances as shown in Figure 6.6. The SISO decoder uses a-prior LLR and generates a-posteriori LLR. In an IDD design, the detector first the a-posteriori LLR from the decoder as input, and generates input soft symbols and variances. These symbol statistics are used to cancel the interference on the received

data from multiple antennas. After interference cancellation, the residuals are then equalized using an MMSE filter based on the symbol variances. In the end, after the output soft symbols are computed, the SISO MMSE detector calculates the a-prior LLR for the decoder. As the iteration increases, if the interference is successfully removed and noise is minimized, the input and output soft symbols will converge, while input and output variances will be very small. In this section, we review the SISO MMSE algorithm and provide methods to further optimize such algorithm as well as its hardware implementation.

6.3.1 Reduced-complexity MMSE-PIC algorithm

In the first demonstrated ASIC chip, SISO MMSE algorithm is implemented with parallel interference cancellation (PIC) algorithm. Referring to the MIMO model in eq. 6.1 with $N_t \times N_r$ antenna configuration, the MMSE PIC algorithm is carried out in six steps [92], including the conversions between LLR and symbol statistics in step (i) and step (vi).

(i) Symbol statistics: Based on the assumption of Gaussian distributions, estimate the soft symbol $\overleftarrow{s}_t$ and variance $\overleftarrow{\sigma}_t^2$ for every symbol $t = 1..N_t$. Convert the input (a-prior) LLRs from a decoder into binary probabilities (more details in section 6.5).

(ii) Gram matrix and matched filter: Compute the Gram matrix $\mathbf{G} = \mathbf{H}^H\mathbf{H}$, and the matched filter output $\mathbf{y}^{mf} = \mathbf{H}^H\mathbf{y}$. The Gram matrix and the matched filter output can be stored in the first iteration, and reused in the subsequent iterations to reduce the computations.

(iii) MMSE filter matrix: Generate the matrix $\mathbf{A}$ and compute the MMSE filter matrix $\mathbf{A}^{-1}$.

$$\mathbf{A} = \mathbf{G}\Lambda + \mathbf{N}_0\mathbf{I}, \tag{6.3}$$

where Λ=diag$(\overleftarrow{\sigma}_t^2, .., \overleftarrow{\sigma}_{N_t}^2)$, and N_0 is estimated noise variance from the channel estimation block in Figure 6.3.

The matrix inversion is implemented based on lower-upper decomposition (LUD) algorithm $\mathbf{A}$=$\mathbf{LU}$, with a forward substitution to find $\mathbf{L}^{-1}$ by solving $\mathbf{Lx}$=$\mathbf{I}$, and it is followed by a backward substitution to find $\mathbf{A}^{-1}$ by solving $\mathbf{Ux}$=$\mathbf{L}^{-1}$.

(iv) Parallel interface cancellation: Perform parallel interface cancellation on $\mathbf{y}_t^{mf}$ from step (ii) by the estimated soft symbols from step (i). The result consists of the contribution by t-th symbol, $t = 1..N_t$, along with noises and possibly errors from incorrect symbol estimations

$$y_t^{PIC} = y_t^{mf} - \sum_{j \neq t} g_j \overleftarrow{S}_j, \tag{6.4}$$

where g_j is the j-th column of matrix $\mathbf{G}$.

(v) MMSE filtering: Compute the bias term for each stream

$$\mu_t = a_t^H g_t, \text{ for } t = 1, .., N_t \tag{6.5}$$

where a_t^H denotes the t-th row of A^{-1}. Using the reciprocals $\mu_t^{-1} = 1/\mu_t$, update the output soft symbols

$$\overrightarrow{s}_t = \mu_t^{-1} a_t^H y^{PIC}, \tag{6.6}$$

and the corresponding SNRs

$$\rho_t = \frac{1}{\overrightarrow{\sigma}_t^2} = \frac{\mu_t}{1 - \overleftarrow{\sigma}_t^2 \times \mu_t}. \tag{6.7}$$

(vi) Outputs: Generate output LLRs for each bit from an output soft symbol

$$\widetilde{L}_{t,b}^{(bin)} = \rho_t (\min_{a \in Z_b^{(0)}} |\overrightarrow{s}_t - a|^2 - \min_{a \in Z_b^{(1)}} |\overrightarrow{s}_t - a|^2). \tag{6.8}$$

where $Z_b^{(0)}$ and $Z_b^{(1)}$ are the subsets of Z with the b-th bit as zero or one respectively.

6.3.2 Algorithmic optimizations for efficient MMSE detector implementation

We present new algorithmic optimizations that reduce the computation and improve the numerical stability for a fixed-point MMSE detector implementation.

Dynamic scaling: The matrix inversion in setp (iii) above requires high arithmetic

precision for numerical stability. In a high SNR regime (small $\mathbf{N}_0$) or after the first outer iteration when the prior information has started to converge (small variance for an input soft symbol), the entries of matrix $\mathbf{A}$ in eq. 6.3 can be very small in value, and the inversion of $\mathbf{A}$ results in very large numerical values. In order to reduce the numerical range needed in matrix inversion and the required word length in a fixed-point hardware, we apply dynamical scaling of the matrix $\mathbf{A}$ by Γ according to the symbol variances.

$$\mathbf{A}' = \mathbf{A}\Gamma = (\mathbf{G}\Lambda + \mathbf{N}_0 I)\Gamma, \tag{6.9}$$

where Γ is a real-valued $N_T \times N_T$ diagonal matrix. A d_{min} is chosen as a design parameter to ensure the diagonal entries in A' are large enough after the scaling, which means $\Gamma_{t,t}\times \overleftarrow{\sigma}_t^2 \geq d_{min}$. The MMSE filtering matrix is found by rescaling eq. 6.9

$$\mathbf{A}^{-1} = (\mathbf{G}\Lambda + \mathbf{N}_0\mathbf{I})^{-1} = \Gamma\Gamma^{-1}(\mathbf{G}\Lambda + \mathbf{N}_0\mathbf{I})^{-1} = \Gamma(\mathbf{A}')^{-1}. \tag{6.10}$$

In addition to scaling the numerical range, we also lower bound the symbol variance to $\overleftarrow{\sigma}_{min}^2$ and the noise density to $\mathbf{N}_{0,min}$ to limit the resolution and ensure numerical stability. In our 4×4 MIMO detector design, we set $\overleftarrow{\sigma}_{min}^2 = 2^{-7}$, $\mathbf{N}_{0,min}$=1, and $d_{min} = 2^{-4}$. With these choices, a 21-bit×21-bit multiplier in matrix inversion is sufficient for computing the matrix inversion under regular channel conditions.

Interference cancellation and MMSE filtering reformation: We reformulate eq. 6.4 by subtracting the $g_t\overleftarrow{s}_t$ term on the right hand side to get y^{IC}.

$$y_t^{IC} = y_t^{mf} - \sum_{j=1}^{j=N_t} g_j \overleftarrow{s}_j \tag{6.11}$$

for $t = 1, .., N_t$. Since the term $g_i\overleftarrow{S}_i$ is the i-th symbol's contribution to y_t^{mf}, this y_t^{IC} term consists of only the noise and possible errors from incorrect symbol estimates. The output soft symbol can be updated using y^{IC} instead of y^{PIC} in eq. 6.6.

$$\overrightarrow{s}_t = \overleftarrow{s}_t + \mu^{-1} a_t^H \mathbf{y}^{IC}. \tag{6.12}$$

Eq. 6.6 and eq. 6.12 are mathematically equivalent. However, in hardware implementation, numerical errors are introduced by fixed point computation when the number

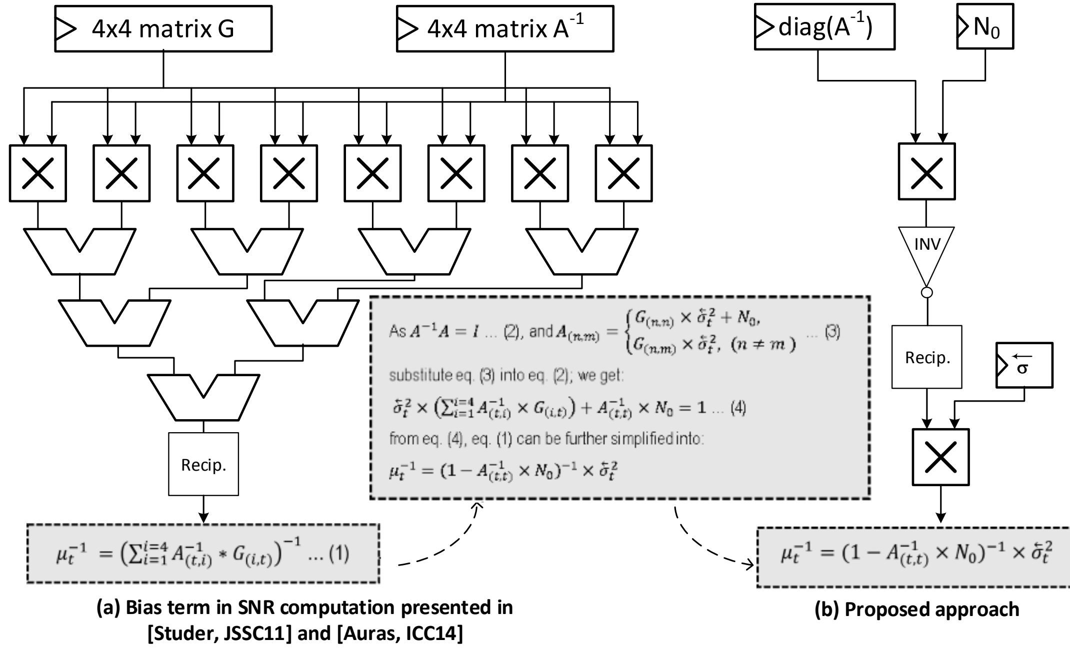


Figure 6.7: Simplified computation for the bias term required for the output SNRs and output soft symbols.

of bits are not sufficient enough. Because μ_t^{-1} is a result after several multiplication, addition, and reciprocal computation, the multiplication of $\mu_t^{-1}y^{PIC}$ amplifies the numerical errors compared to $\mu_t^{-1}y^{IC}$ because y^{PIC} contains an extra soft symbol term. Thus, using eq. 6.12 is more numerically stable.

Bias term computation: The bias term computation $\mu_t = a_t^H g_t$ requires 4 complex multiplications, 4 complex additions, and a large memory to store both matrix $\mathbf{A}^{-1}$ and $\mathbf{G}$. By reformulating the equation, we find a new approach to cut down computational complexity without degrading the performance.

First, based on eq. 6.3, the matrix $\mathbf{A}$ can be computed as

$$\mathbf{A}_{n,m} = \begin{cases} \mathbf{G}_{n,m} \times \vec{\sigma}_n^{\,2}, \, n \neq m \\ \mathbf{G}_{n,m} \times \vec{\sigma}_n^{\,2} + \mathbf{N}_0, \, n = m \end{cases} , \tag{6.13}$$

where $\mathbf{A}_{n,m}$ represents the entry of n-th row and m-th column in matrix $\mathbf{A}$. Secondly,

since $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$, we have

$$\sum_{i=0}^{i=N_t} \mathbf{A}_{t,i}^{-1}\mathbf{A}_{i,t} = 1 \tag{6.14}$$

Substituting eq. 6.13 into eq. 6.14, we have

$$\sum_{i=0}^{i=N_t} \mathbf{A}_{t,i}^{-1}\mathbf{A}_{i,t} = \overleftarrow{\sigma}_t^{2} \times (\sum_{i=0}^{i=N_t} \mathbf{A}_{(t,i)}^{-1}\mathbf{G}_{i,t}) + \mathbf{A}_{t,t}^{-1} \times \mathbf{N}_0 = 1. \tag{6.15}$$

Comparing eq. 6.15 to the bias term in eq. 6.5, we find

$$\overleftarrow{\sigma}_t^{2} \times (\sum_{i=0}^{i=N_t} \mathbf{A}_{t,i}^{-1}\mathbf{G}_{i,t}) + \mathbf{A}_{t,t}^{-1} \times \mathbf{N}_0 = \overleftarrow{\sigma}_t^{2} \mu_t + \mathbf{A}_{t,t}^{-1} \times \mathbf{N}_0 = 1. \tag{6.16}$$

Thus, the computation μ_t can be simplified as

$$\mu_t = (1 - A_{t,t}^{-1} \times N_0) \times (\overleftarrow{\sigma}_t^{2})^{-1}. \tag{6.17}$$

This simplification reduces more than 70% of computation area, including 6 multipliers, 7 adders, and input memory of 792 bits as shown in Figure 6.7, resulting in 65% power reduction.

6.4 Architecture and circuit optimization

Our MMSE ASIC is implemented in 4 coarse pipeline stages as depicted in Figure 6.8. Channel information and input symbol LLRs from the decoder are processed in the first stage to generate matrix $\mathbf{A}$. The matrix is then inverted using LUD and substitutions for MMSE filtering in the second and third stage, while interference cancellation is done in parallel. The final stage computes SNR and symbol LLRs as the input to the NBLDPC decoder.

The LUD in the second stage contains the critical paths and requires a long latency, causing a bottleneck in throughput. The LUD pseudo code in Figure 6.9 shows the data dependency for each loop. We identify the critical path in each top loop is "reciprocal (line 3)→column update (line 6)→ diagonal entry update (line 9)". As the reciprocal unit dictates the inner loop of the LUD and is accessed sequentially due to data dependency, we propose the reciprocal computation by Newton-Raphson algorithm in a parallel structure to shorten the number of cycles in the second stage

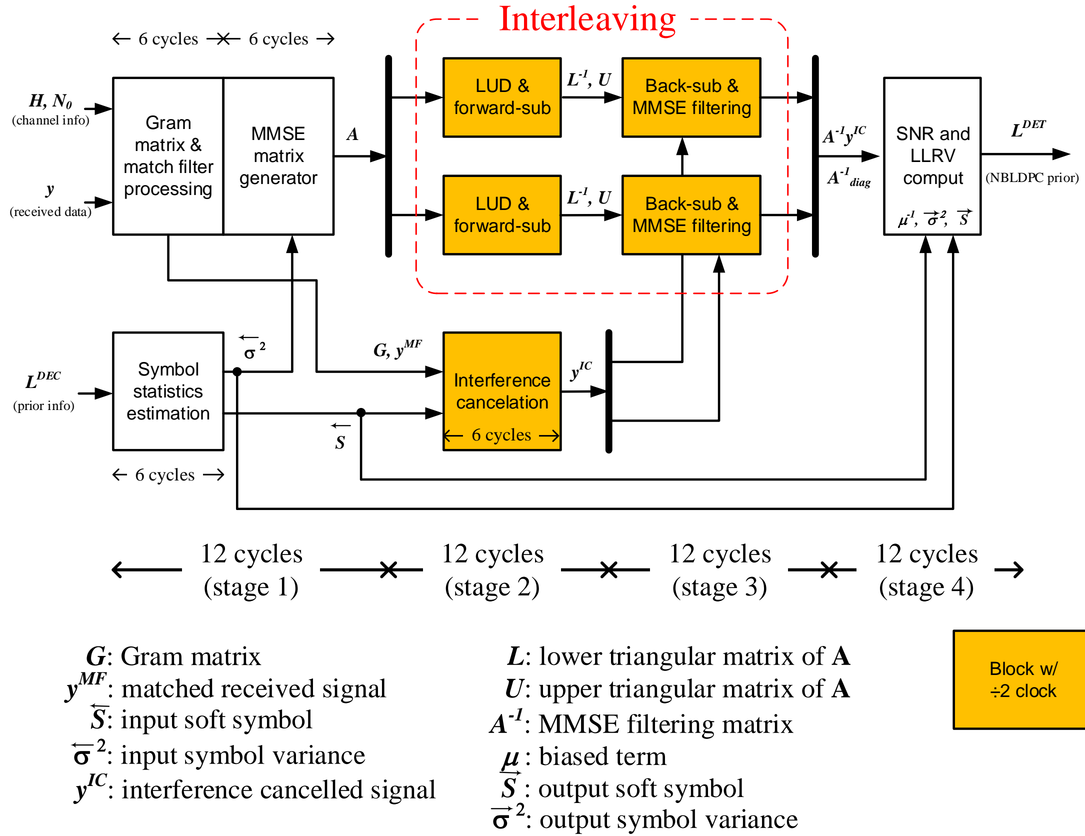


Figure 6.8: Design of the MMSE detector in 4 task-based coarse pipeline stages. Stage 2 and 3 operate at a 2X slower clock frequency, and the remaining stages operate at the base clock frequency.

LU Decomposition Pseudo Code

1 **for** $i = 1$ **to** N_t
2 $\quad \mathbf{U}_{i,i} = \mathbf{A}_{i,i}$
3 $\quad \mathbf{R}_{i,i} = 1/\mathbf{U}_{i,i}$ (reciprocal)
4 $\quad$ **for** $j = i + 1$ **to** N_t
5 $\quad\quad \mathbf{U}_{i,j} = \mathbf{A}_{i,j}$
6 $\quad\quad \mathbf{L}_{j,i} = \mathbf{A}_{j,i} \times \mathbf{R}_{i,i}$ (scaling)
7 $\quad$ **for** $j = i + 1$ **to** N_t
8 $\quad\quad$ **for** $k = j + 1$ **to** N_t
9 $\quad\quad\quad \mathbf{A}_{j,k} = \mathbf{A}_{j,k} - \mathbf{L}_{j,i}\mathbf{U}_{i,k}$ (update)
10 **return L** and **U**

Figure 6.9: Lower-upper decomposition for an $N_t \times N_t$ matrix $\mathbf{A}$.

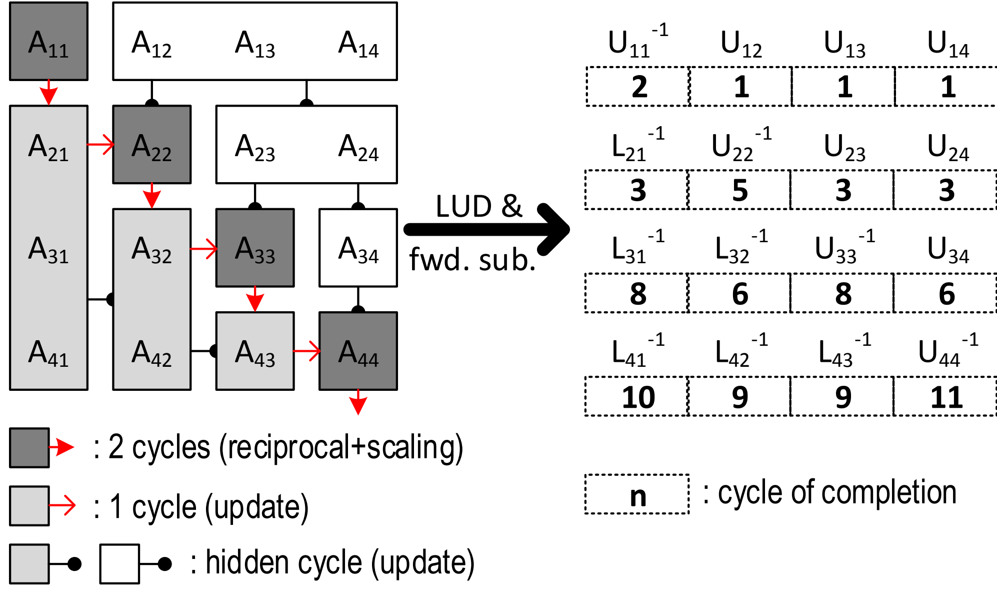


Figure 6.10: Implementation of LUD and forward substitution in stage 2. Left figure indicates the data dependency throughout the algorithm, and highlights the critical latency; right figure annotates the cycles when the result is completed.

to 12 cycles as shown in Figure 6.10, compared to 18 cycles in the prior work [92,93]. However, cycle reduction also tightens the timing constraint on the complex multipliers in the second and third stages, requiring longer cycle time and resulting in unbalanced pipeline stages. To achieve a better throughput, we propose to process the matrix inversion and MMSE filtering in a interleaving scheme shown in Figure 6.8. In addition to throughput enhancement, we propose additional methods to improve area and power reduction by just-in-time sequential computation in stage 3 and clock gating circuit technique.

Reciprocal: We redesign the reciprocal based on Newton-Raphson algorithm from [92]. The reciprocal is obtained iteratively using the following equation

$$\tilde{x}_{k+1} = 2\tilde{x}_k - x\tilde{x}_k^2. \tag{6.18}$$

Given $1 \leq x < 2$, $\tilde{x}$ converges to x^{-1} through iterative update. However, as the reciprocal is accessed sequentially due to the data dependency in LUD processing,

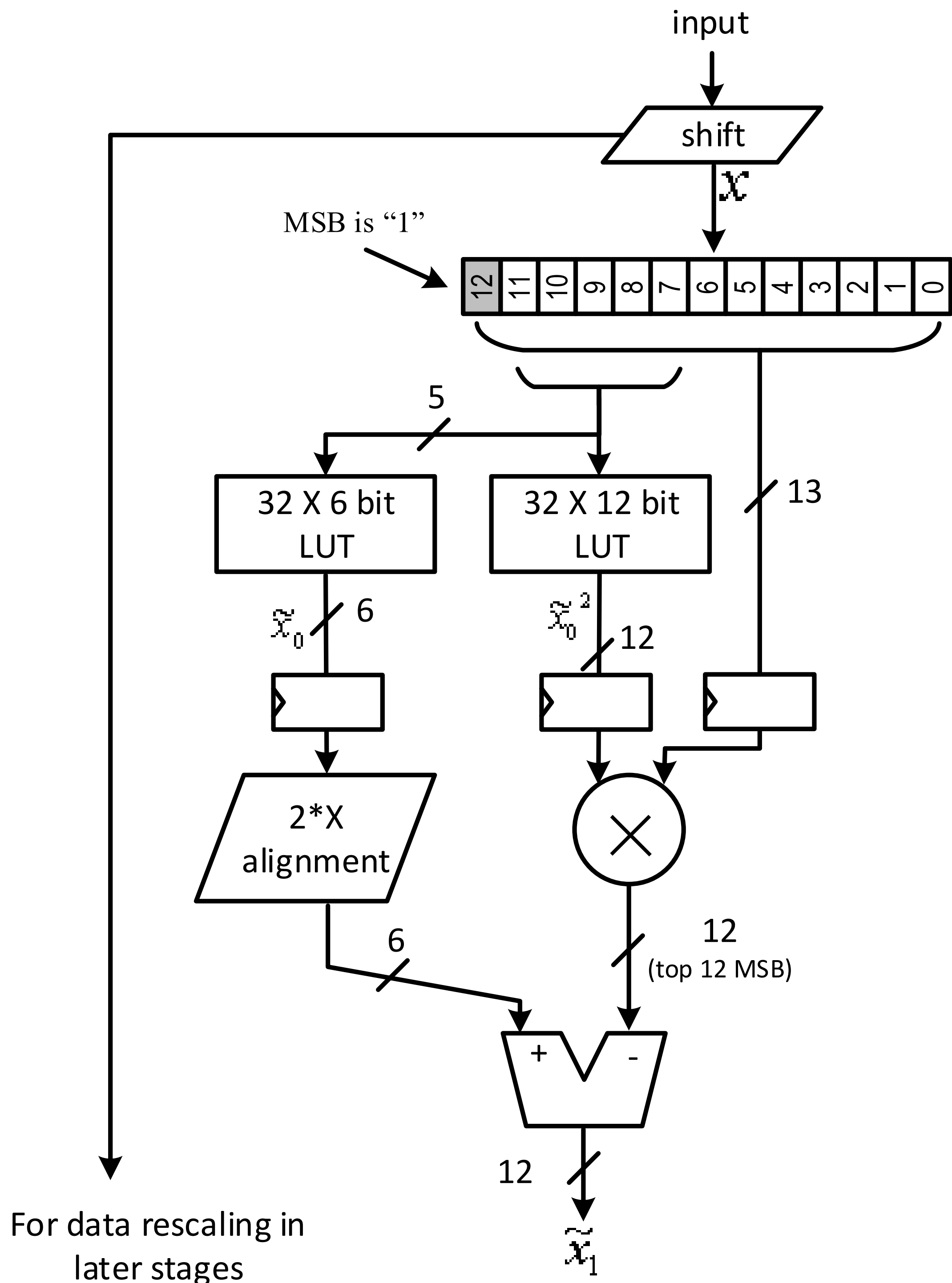


Figure 6.11: Design of reciprocal based on two LUTs.

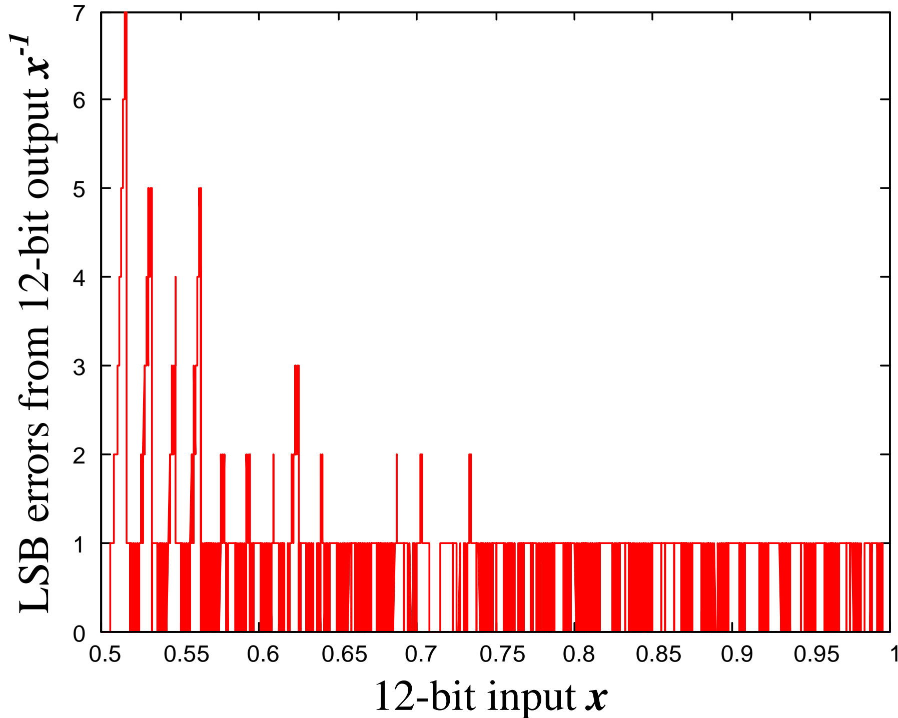


Figure 6.12: Least significant bit (LSB) error from the proposed reciprocal design.

the longer latency caused by iterative approach will directly degrade latency as well as system throughput. To avoid the throughput degradation, we implement this algorithm in a feed forward scheme. The key novelty in our design is the use of the top few MSB bits as the inputs of LUT1 and LUT2 to estimate $\tilde{x}_0$ and $\tilde{x}_0^2$ as the initial inputs to eq. 6.18. The block diagram of the lookup table (LUT) based hardware design is presented in Figure 6.11. The input is initially shifted to have the leading "1" as the most significant bit (MSB). This shift ensures output x to be in the range between 1 and 2. The output of LUT1 is scaled by 2, which is a bit alignment and does not require any extra circuitry. The output from LUT2 is multiplied by the input x. The aligned output of LUT1 and the product from the multiplication are added together to obtain the output. Using 5-bit input LUT1 and LUT2 and the word length labeled in Figure 6.11, an average precision of $2^{-11.14}$ can be achieved with a maximum error of $2^{-9.2}$ as shown in Figure 6.12, which is sufficient for the MMSE detector based on our simulation and what is reported in [93].

Interleaving: The 12-cycle LUD is achieved by the 2-cycle reciprocal, and one cycle complex multiplier in the second stage as labeled in Figure 6.10. In the third stage,

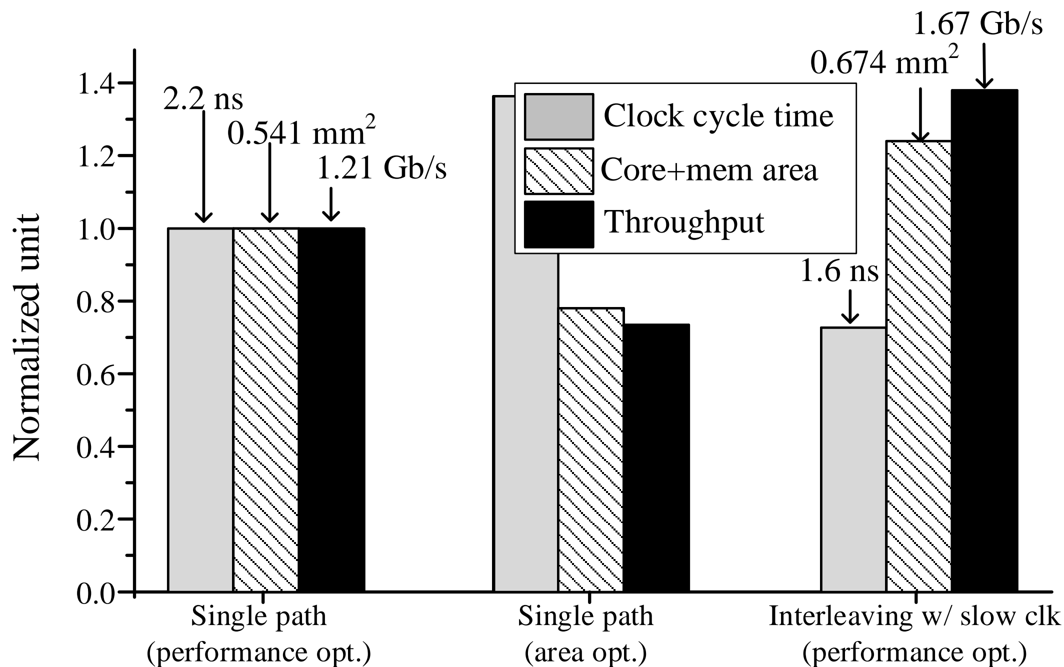


Figure 6.13: Optimization of the critical paths stage 2 and stage 3 in the MMSE detector. Performance-optimized single path is the baseline design.

the complex 21-bit×21-bit multiplier for the back substitution is done in 2 cycles. However, the long reciprocal and multiplier in the second and third stages require a long cycle time, diminishing the throughput. To loosen the timing constraint on these long critical paths and balance all stages, we use a simple clock divider and create a 2X slow clock domain for the two stages, and recoup the throughput by interleaving between two copies of the datapaths without stalling the pipeline as shown in Figure 6.8. The 2X slow clock provides additional timing slacks to allow the gates to be downsized. As a result, the duplication costs only 24% additional area over the baseline, as depicted in Figure 6.13, but the throughput is increased by 38%. The downsized gates also reduce the load capacitance, and thus improving the energy efficiency.

Just-in-time sequential MMSE filtering: Before generating the output soft symbol, MMSE filtering computes $a_t^H y^{IC}$ (eq. 6.12). This operation can be done concurrently as the filtering matrix A^{-1} is being generated during backward substitution. With our pipeline scheduling illustrated in Figure 6.14, the completion of each entry $A^{-1}_{t,m}$ triggers an immediate update of the corresponded filtering output. For instance,

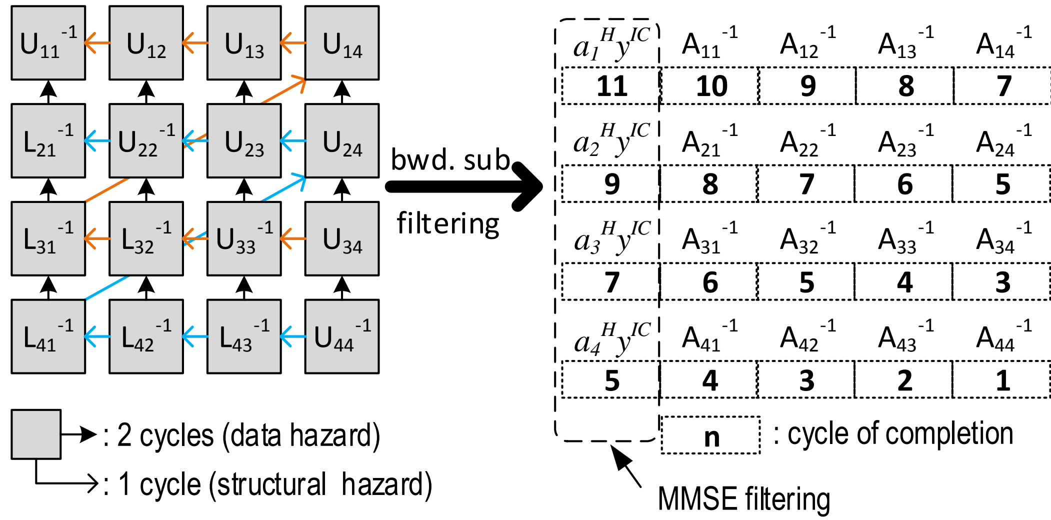


Figure 6.14: Backward substitution and MMSE filtering in stage 3. Left figure shows the data hazards from 2-cycle complex multipilers and the structural hazard from 2 processing units; right figure annotates the cycles when the entry in matrix A^{-1} is completed. MMSE filtering is processed sequentially but is parallelly to the backward substitution.

when $A_{4,4}^{-1}$ entry is available, $a_4^H y^{IC}$is updated by the term of $A_{4,4}^{-1} y_4^{IC}$. Therefore,all entries of A^{-1} but the diagonal ones are consumed right away, reducing the boundary registers for the filtering matrix by 85%. Only the diagonal entries are buffered for the fourth stage for the bias term calculation.

Clock gating: To lower the power consumption, automatic clock gating is applied to stage boundary and buffer registers to save 53% power of the detector. A total of 9.1kb registers are used for buffering data in and between stages of the detector. Registers are used in place of memory arrays to support high access bandwidth and the flexibility of placing small memory blocks. Registers are power hungry, but we recognize a power reduction opportunity as most of the registers used in our design are infrequently updated due to the coarse pipelining, e.g., 1 update every 12 cycles for the 7.6kb stage boundary registers in the detector. We exploit the access pattern to reduce power by enabling clock gating of the registers when they are idling, saving 53% power in total.

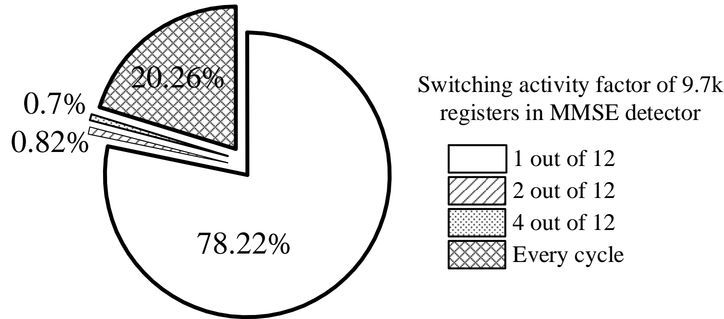


Figure 6.15: Data switching activity factors in MMSE detector.

6.5 Nonbinary interface and optimization

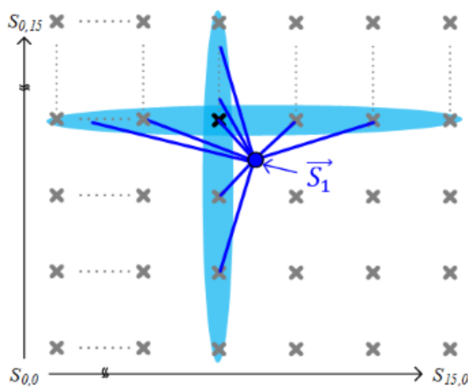
In this section, we propose our methods to efficiently calculate the soft information exchanged between the decoder and the detector iteratively. In particular, our computation is designed to support nonbinary LDPC codes defined over GF.

6.5.1 Detector to decoder: LLR computation with L1-norm

In a conventional IDD system with a binary coding, a detector computes the output LLR for each bit by the Euclidean distance to all constellation points. The max-log approximation [97] can be applied to significantly reduce the implementation complexity at the cost of a small performance loss. With the max-log approximation, the binary LLR computation requires only the Euclidean distances to the two constellation points. One of the points has the highest probability of data 1 while the other one carries the most likely data 0, as described by eq. 6.8.

In a low-order modulator, such as BPSK or 4-QAM, the LLR computation is relatively simple. However, it becomes more complicated in a higher-order modulation. As an M-QAM modulation is the combination of two $\sqrt{M}$-pulse amplitude modulations (PAM),The max-log approximation reduces the search space from M candidates into $2\sqrt{M}$ candidates as highlighted in Figure 6.16. A further simplification can be achieved by piecewise linear function approximated based on Gray mapping but it suffers from a loss of performance in a higher modulation scheme. However, even with these simplifications, the LLR calculation for a large constellation is still complex, e.g., it requires 18 LLR calculations for a 64-QAM constellation [98].

Compared to binary LLR computation, the proposed MMSE detector and the NBLDPC decoder exchange symbol LLRs. Conventionally, the symbol LLRs are constructed from the binary LLRs [95]. For instance, a GF(256) element K has a



bit	LLR
b0	$\lvert\vec{S_1}-s_{14,3}\rvert^2-\lvert\vec{S_1}-s_{12,3}\rvert^2$
b1	$\lvert\vec{S_1}-s_{13,3}\rvert^2-\lvert\vec{S_1}-s_{12,3}\rvert^2$
b2	$\lvert\vec{S_1}-s_{12,3}\rvert^2-\lvert\vec{S_1}-s_{11,3}\rvert^2$
b3	$\lvert\vec{S_1}-s_{12,3}\rvert^2-\lvert\vec{S_1}-s_{7,3}\rvert^2$
b4	$\lvert\vec{S_1}-s_{12,1}\rvert^2-\lvert\vec{S_1}-s_{12,3}\rvert^2$
b5	$\lvert\vec{S_1}-s_{12,2}\rvert^2-\lvert\vec{S_1}-s_{12,3}\rvert^2$
b6	$\lvert\vec{S_1}-s_{12,3}\rvert^2-\lvert\vec{S_1}-s_{12,4}\rvert^2$
b7	$\lvert\vec{S_1}-s_{12,8}\rvert^2-\lvert\vec{S_1}-s_{12,3}\rvert^2$

Figure 6.16: Bit LLR computation (before SNR scaling) for the soft detector output $\vec{s}_1$ in 256-QAM constellations.

normalized symbol LLR $\widetilde{L}_K^{(GF)}$ accumulated from 8 binary LLRs

$$\widetilde{L}_K^{(GF)} = \sum_{b=1}^{b=8} f(K_b)\widetilde{L}_{K,b}^{(bin)} \qquad f(K_b) = \begin{cases} 1, K_b = 1 \\ -1, K_b = 0 \end{cases}, \tag{6.19}$$

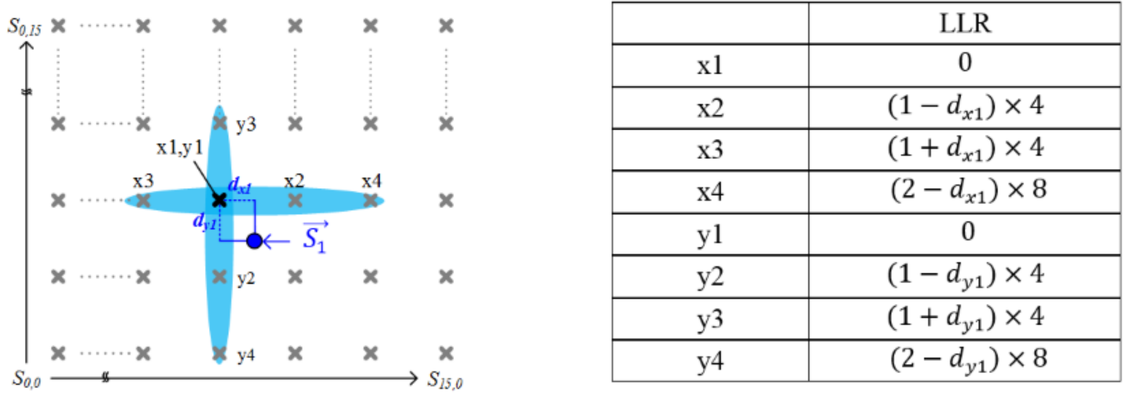
where K_b is the b-th binary bit of symbol K, and $\widetilde{L}_{K,b}^{(bin)}$ is derived from eq. 6.8. In a high GF coding, e.g., GF(256), eq. 6.19 needs to be evaluated for each of the 256 GF elements or constellation points, making it a very expensive operation.

We exploit a new approach to simplify the symbol LLR calculation by directly computing from the distance between the soft symbol and the constellation points. To begin, given a soft symbol $\vec{s}_t$, the constellation point K's log likelihood (LL) is

$$\widetilde{LL}_K^{(GF)} = -\rho_i|\vec{s_t} - K|^2. \tag{6.20}$$

Eq. 6.20 is non-positive, meaning that the likelihood value of $\widetilde{LL}_K^{(GF)}$ in the linear domain is less than or equal to 1. In NBLDPC decoding, each constellation point is mapped to a symbol, and the symbol LLR is calculated by dividing the symbol's likelihood by reference symbol's likelihood. Conventionally, the reference symbol is chosen as the 0 GF element [94,95]. The symbol LLRs are sorted to support NBLDPC decoding [94].

Since the symbol LLRs are calculated by normalization and then sorted for NBLDPC decoding, we propose to simplify the symbol LLR calculation by using the most likely constellation point as the reference symbol, labeled as $s_{x1,y1}$ in Figure 6.17. That is,



	LLR
x1	0
x2	$(1 - d_{x1}) \times 4$
x3	$(1 + d_{x1}) \times 4$
x4	$(2 - d_{x1}) \times 8$
y1	0
y2	$(1 - d_{y1}) \times 4$
y3	$(1 + d_{y1}) \times 4$
y4	$(2 - d_{y1}) \times 8$

Figure 6.17: Symbol LLR computation (before SNR scaling) for the soft detector output $\vec{s_1}$ in 256-QAM constellation. (Note that the x and y entries are cross added to obtain the symbol LLRs for a GF(256) decoder.)

we calculate the symbol LLR by normalizing the $\widetilde{LL}_K^{(GF)}$ of the symbol K by the $\widetilde{LL}_{x1,y1}^{(GF)}$ of the most likely symbol $s_{x1,y1}$. Based on eq. 6.8 and eq. 6.20, symbol LLR is calculated using the following equation

$$\widetilde{L}_K^{(GF)} = -(\widetilde{LL}_K^{(GF)} - \widetilde{LL}_{x1,y1}^{(GF)}) = \rho_i(|\vec{s_t} - s_K|^2 - |\vec{s_t} - s_{x1,y1}|^2). \tag{6.21}$$

Note that we scale the normalized LLRs by -1, resulting in non-negative LLRs to simplify the computation in a NBLDPC decoder.

We project the Euclidean distance onto real and imaginary axes as shown in Figure 6.17. Assuming the constellation points are spaced by 2, if the distance between the soft symbol and the closest constellation point is d_{x1} on the real axis, the most likely symbol has LLR value of 0, and the second most likely symbol $x2$ has

$$\widetilde{L}_{x2}^{(GF)} = \rho_i(|2 - d_{x1}|^2 - |d_{x1}|^2) = \rho_i(4 - 4d_{x1}). \tag{6.22}$$

The square terms in eq. 6.21 are canceled out, and the evaluation requires only L1-norm and no multiplication. This L1-norm can be further extended to the rest of the symbol candidates, as illustrated in Figure 6.17. The output symbol LLRs are monotonically increasing with the first LLR being 0, resulting in sorted symbol LLRs. In an M-QAM constellation, the symbol LLRs for the real and imaginary axes can be calculated independently as two sets of GF($\sqrt{M}$) LLRs and LLRs are cross added to get GF(M) LLRs.

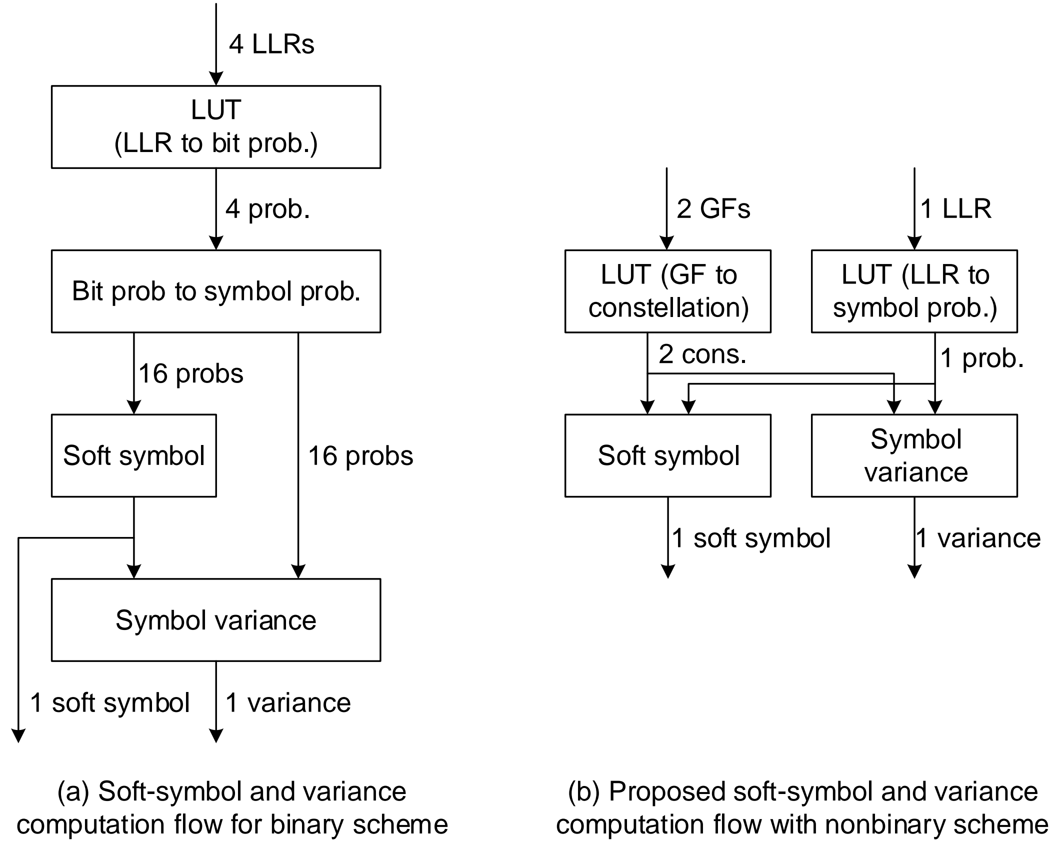


Figure 6.18: Computation flow for input soft symbols and variances in (a) binary scheme, and (b) proposed nonbinary scheme.

In our chip prototype implementation, the designed NBLDPC decoder requires the 12 most reliable GF elements in GF(256). Thus, we only need the 4 most likely symbols in each of the real and imaginary axis, and then cross add the LLRs to form the 12 most reliable GF elements along with the LLRs. Finally, as Figure 6.17 illustrates, the LLRs are computed using bit invert, shift and add in our design, and the nearest points are determined directly based on the output soft symbols, eliminating costly search and multiply.

6.5.2 Decoder to detector: Efficient soft-symbol and variance computation

The computation of soft-symbol $\overleftarrow{s}_t$ and variance $\overleftarrow{\sigma}_t^2$ is the first step of the SISO MMSE detection, as shown in Figure 6.8. Since the real axis and imaginary axis are

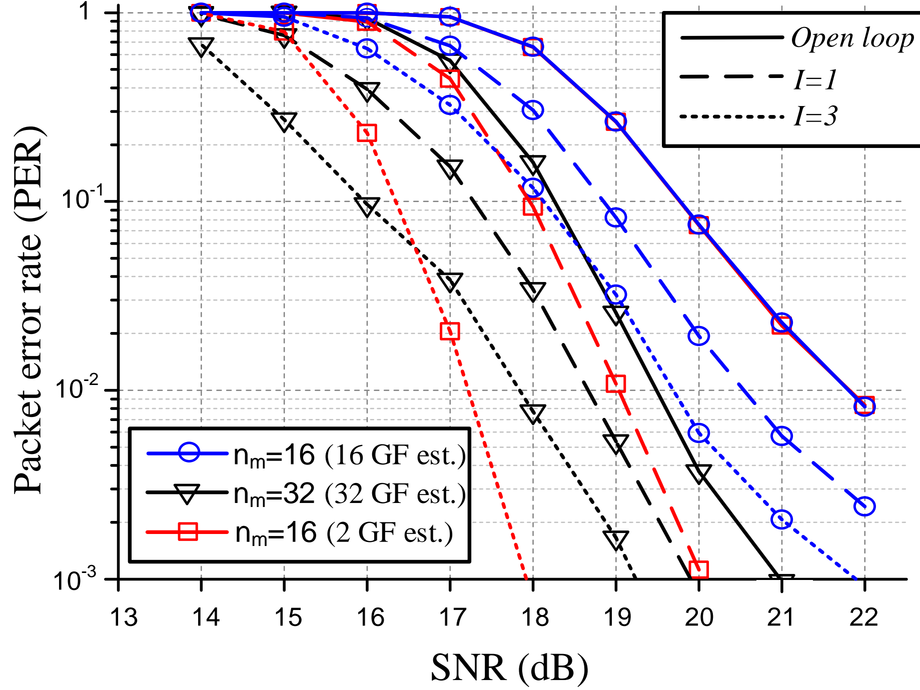


Figure 6.19: PER comparison among three cases: (1) 16 GF elements for decoding and 16 GF elements for soft symbol estimation, (2) 32 GF elements for decoding and 32 GF elements soft symbol estimation, and (3) 16 GF elements for decoding and only top 2 GF elements for soft symbol estimation.

independent as described in section 6.5.1, we only discuss the soft symbol calculation for the real axis. The real axis for 256-QAM represents a 16-PAM constellation.

In a conventional binary coding scheme [92] as illustrated in Figure 6.18(a), the binary LLRs are first translated into linear probability P$[s_{t,b}]$ by

$$P[s_{t,b} = x] = \frac{\mathrm{e}^{(0.5xL_{t,b}^{(bin)})}}{\mathrm{e}^{(0.5L_{t,b}^{(bin)})} + \mathrm{e}^{(-0.5L_{t,b}^{(bin)})}},\ x = \{+1, -1\}. \tag{6.23}$$

With the linear probabilities, a symbol probability can be found by

$$P[s_t = a] = \prod_{b=1}^{b=4} P[s_{t,b} = a_b],\ \forall t \in [1, .., N_t]\ \text{and}\ \forall a \in [1, .., \sqrt{M}]. \tag{6.24}$$

Finally, the input soft symbol is calculated by

$$\overleftarrow{s}_t = \sum_{a=1}^{a=\sqrt{M}} P[s_t = a]a, \tag{6.25}$$

and the variance of the soft symbol is

$$\overleftarrow{\sigma}_t^2 = \sum_{a=1}^{a=\sqrt{M}} P[s_t = a](a - \overleftarrow{s}_t)^2, \tag{6.26}$$

Soft symbol and variance computations can be simplified if certain mapping systems are assumed, e.g. Gray mapping. In [99], the soft symbol and variance computations are done using 3 LLR-to-probability look-up tables (LUTs), 7 additions, and 5 multiplications are required for a 64-QAM modulation. The complexity scales quadratically with the constellation sizes, and the same approach can hardly be extended into a 256-QAM or a higher-order modulation due to the prohibitive complexity.

We propose a new methodology that computes soft symbol and variance efficiently from the nonbinary LLRs. A truncated EMS nonbinary LDPC decoder produces n_m candidates for an output symbol [100]. In our implementation, we use $n_m = 12$ for a GF(256) output symbol. Because all the n_m candidates for a nonbinary symbol are normalized to the most reliable candidate α_1, which has a numeral LLR value of 0, the normalized probability of a constellation point that corresponds to symbol α_i is calculated as

$$\frac{P[s_t = Z(\alpha_i)]}{P[s_t = Z(\alpha_1)]} = \frac{1}{2}(1 - \tanh(\frac{1}{2}L_{\alpha_i}^{(GF)})), \forall i \in [1, .., n_m]. \tag{6.27}$$

where function $Z(\alpha_i)$ maps a GF element to a constellation point. The linear probability of the constellation point is then calculated by

$$P[s_t = Z(\alpha_i)] = \frac{(P[s_t = Z(\alpha_i)]/P[s_t = Z(\alpha_1)])}{\sum_{j=1}^{j=n_m} \frac{P[s_t = Z(\alpha_j)]}{P[s_t = Z(\alpha_1)]}}, \tag{6.28}$$

which is very complicated to compute, especially the denominator term. In this work, we propose a much simpler approach to calculate soft symbols and variances using only the two most reliable candidate constellaion points for a nonbinary symbol shown in Figure 6.18(b). In the first step, the top two candidates α_1 and α_2 are mapped to constellation points s_1 and s_2 by $Z(\alpha_1)$ and $Z(\alpha_2)$, respectively. The probabilities of

these two points are calculated as P_1 and P_2.

$$P_2 = \frac{1}{1+\mathrm{e}^{L_2^{(GF)}}}, \text{ and } P_1 = 1 - P_2 = \frac{\mathrm{e}^{L_2^{(GF)}}}{1+\mathrm{e}^{L_2^{(GF)}}} \tag{6.29}$$

Assume that the top two candidates' probabilities dominate, so that P_1 can be simplified to $(1-P_2)$, and the division in eq. 6.28 is no longer needed. As P_2 is always equal or lower than 0.5, $(1-P_2)$ can be approximately calculated by bit-wise inverting, i.e., $\sim P_2$. Note that we use 1's complement as opposed to 2's complement, but it has a negligible impact on the performance [101]. Input soft symbol computation on the real axis now becomes

$$\overleftarrow{s}_{t,real} = P_1 s_{1,real} + P_2 s_{2,real} = (\sim P_2) s_{1,real} + P_2 s_{2,real}, \tag{6.30}$$

and the equivalent variance is

$$\overleftarrow{\sigma}^2_{t,real} = P_1 (s_{1,real} - \overleftarrow{s}_{t,real})^2 + P_2 (s_{2,real} - \overleftarrow{s}_{t,real})^2. \tag{6.31}$$

Substituting eq. 6.30 into eq. 6.31, we simplify the variance computation to

$$\overleftarrow{\sigma}^2_{t,real} = (\sim P_2) P_2 (s_{1,real} - s_{2,real})^2. \tag{6.32}$$

As the dependency of soft symbol in variance computation is eliminated, the soft symbol and variance can be calculated in parallel to reduce the latency as shown in Figure 6.18.

Using the two dominant candidates, our approach only requires 1 LLR-to-probability LUTs, 2 additions, and 5 multiplications for one axis of a soft symbol calculation in a 256-QAM modulation. The variances from two axes are summed up to calculate the input symbol variance by $\overleftarrow{\sigma}^2_t = \overleftarrow{\sigma}^2_{t,real} + \overleftarrow{\sigma}^2_{t,imag}$. However, as only 2 candidates are considered, the input symbol variance is usually underestimated when P_2 is large. Therefore, we add a compensation term based on P_2 to improve the PER performance, as done in qe. 6.33

$$\overleftarrow{\sigma}^2_t = \overleftarrow{\sigma}^2_{t,real} + \overleftarrow{\sigma}^2_{t,imag} + C P_2, \tag{6.33}$$

where C is chosen as 2^n for a convenient implementation that does not require a multiplier.

In Figure 6.19, we compare the PER performance using three prior probability

Processing Block	Mult	Add	LUT	Recip	Cycles	Area (%)
Gram matrix & matched filter	16	16	0	0	6	13%
Symbol statistics	3	3	5	0	6	1.1%
SISO matrix **A**	4	1	0	0	6	3.9%
LUD & forward substitution	8	12	2	1	12 (÷2)	14.3%
Back-sub & MMSE filtering	10	8	0	0	12 (÷2)	22.3%
Interference cancelation	4	4	0	0	6 (÷2)	2.2%
LLRV	4	4	8	1	12	4.7%
Miscellaneous						1.9%

Table 6.3: MIMO detection algorithm (hard outputs) design tradeoffs

computations: (1) using $n_m = 16$ case and all 16 GF elements to calculate the soft symbol and variance following eq. 6.27 and eq. 6.28, (2) using $n_m = 32$ and all 32 GF elements to calculate the soft symbol and variance, and (3) using $n_m = 16$ and only the top 2 GF elements to calculate the soft symbol and variance. In an open loop operation, $n_m = 32$ provides better performance over $n_m = 16$. However, in an iterative loop, it is clear that using only the 2 most reliable GF elements provides an excellent PER performance, suggesting that it is a viable approach for a much reduced complexity.

6.6 Measurements

We summarize the design and area breakdown of each block of the MMSE detector in Table 6.3. Due to the limitation of silicon area, we choose a NBLDPC code with block length of 412 bits and implement EMS decoding with $n_m = 12$ in the test design. The PER performance of this setup for a 4x4 256-QAM MIMO is provided in Figure 6.20.

The MMSE-NBLDPC iterative detector-decoder test chip was fabricated in TSMC 65nm CMOS technology. The chip dimension is 2.04mm×2.2mm, and the MMSE detector core and the NBLDPC decoder core occupy 0.7mm^2 and 1.7mm^2, respectively. The fabricated test chip is fully functional. At room temperature and 1.0V supply, the MMSE detector runs at a maximum frequency of 517MHz for a throughput of 1.38Gb/s, the highest reported throughput of a SISO MMSE detector [92]. The MMSE detector consumes 26.5mW, which translate to 19.2pJ per bit, an order of magnitude lower than the previous SISO detector designs [88, 89, 92], demonstrating

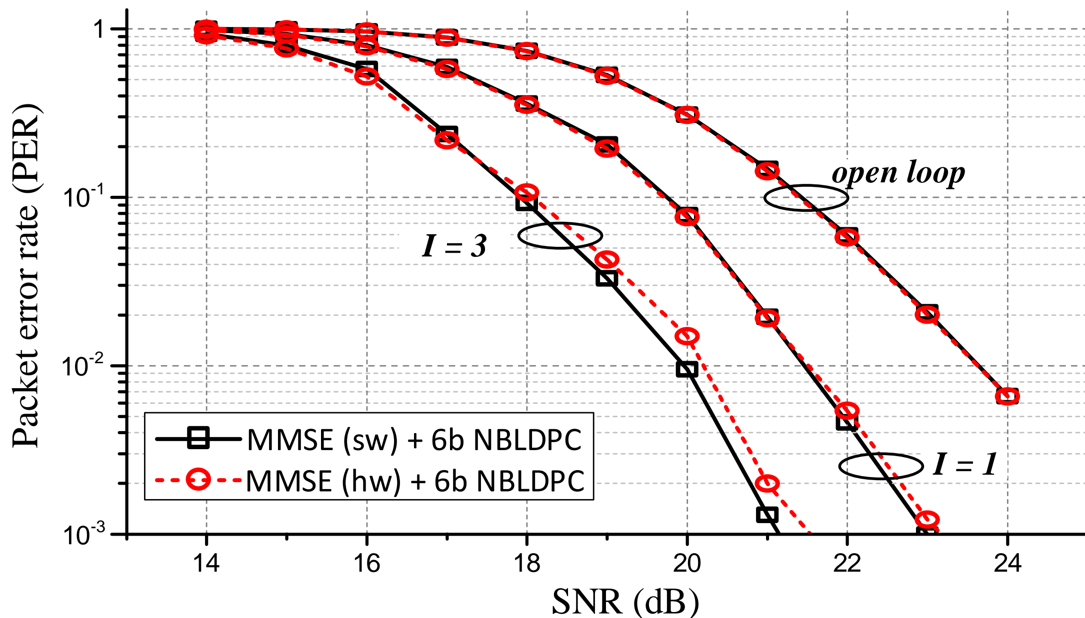


Figure 6.20: PER performance comparison of MMSE-NBLDPC GF(256) IDD system with n_m=12 and 412-b code length.

the advantage of our optimized MMSE detection for IDD.

The energy efficiency can be further improved by voltage and frequency scaling, as shown in Figure 6.22. At a 500mV supply, the MMSE detector. Our work is compared with state-of-the-art MIMO detector and decoder designs in table 6.4.

6.7 Conclusion

We develop the first MMSE-NBLDPC IDD system that achieves a higher coding gain compared to the conventional binary IDD systems [88, 89, 92]. In our MMSE ASIC implementation, we first present algorithmic optimizations to enhance the numerical stability and reduce the computational complexity. In particular, we use an algorithmic property to greatly reduce the area of the output soft symbol and SNR computation by 40%. Then, we demonstrate architecture and circuit techniques to improve throughput and energy efficiency, including interleaving and two clock domains, a new reciprocal block, just-in-time sequential MMSE filtering, and clock gating. To efficiently exchange nonbinary soft information, we present a technique to compute NBLDPC's a-prior LLRs using only the L1-norm, and another technique to compute MMSE's input soft information base on only the two most reliable GF

Detector	Noethen ISSCC14	Borlenghi ESSRC12	Winter ISSCC12	Studer JSSC11	This work
IDD design	yes	yes	no	yes	yes
Algorithm	SD SISO	SD SISO	SD SO	MMSE SISO	MMSE NB-SISO
MIMO system	≤ 4x4	≤ 4x4	≤ 4x4	4x4	4x4
Modulation	≤ 64	≤ 64	≤ 64	≤ 64	256
Technology [nm]	65	65	65	90	65
Core area [mm^2]	-	2.78	0.31	1.5	0.7
Preprocessing area [kGE] Detection area [kGE]	383[a]	-[b] 872	-[b] 215	410	347[c]
Frequency [MHz]	445	135	333	568	517
Power [mW]	87	-	38	189	26.5
Throughput [Mb/s]	396	194	296-807	757	1379
Area efficiency [Mb/s/kGE]	1.03	0.22	1.37-3.75	1.85	3.68
Energy efficiency [pJ/b]	220	920	48	250	19.2

[a]: memory for data exchange included
[b]: data pre-processing block (QRD) not included
[c]: total area is 264 kGE if no interleaving processing

Table 6.4: Comparison with state-of-the-art MIMO detector.

elements from NBLDPC's a-posteriori LLRs. Our SISO MMSE implementation supports 4x4 256-QAM MIMO system, and it achieves the highest throughput and energy efficiency among the latest published SISO detector designs.

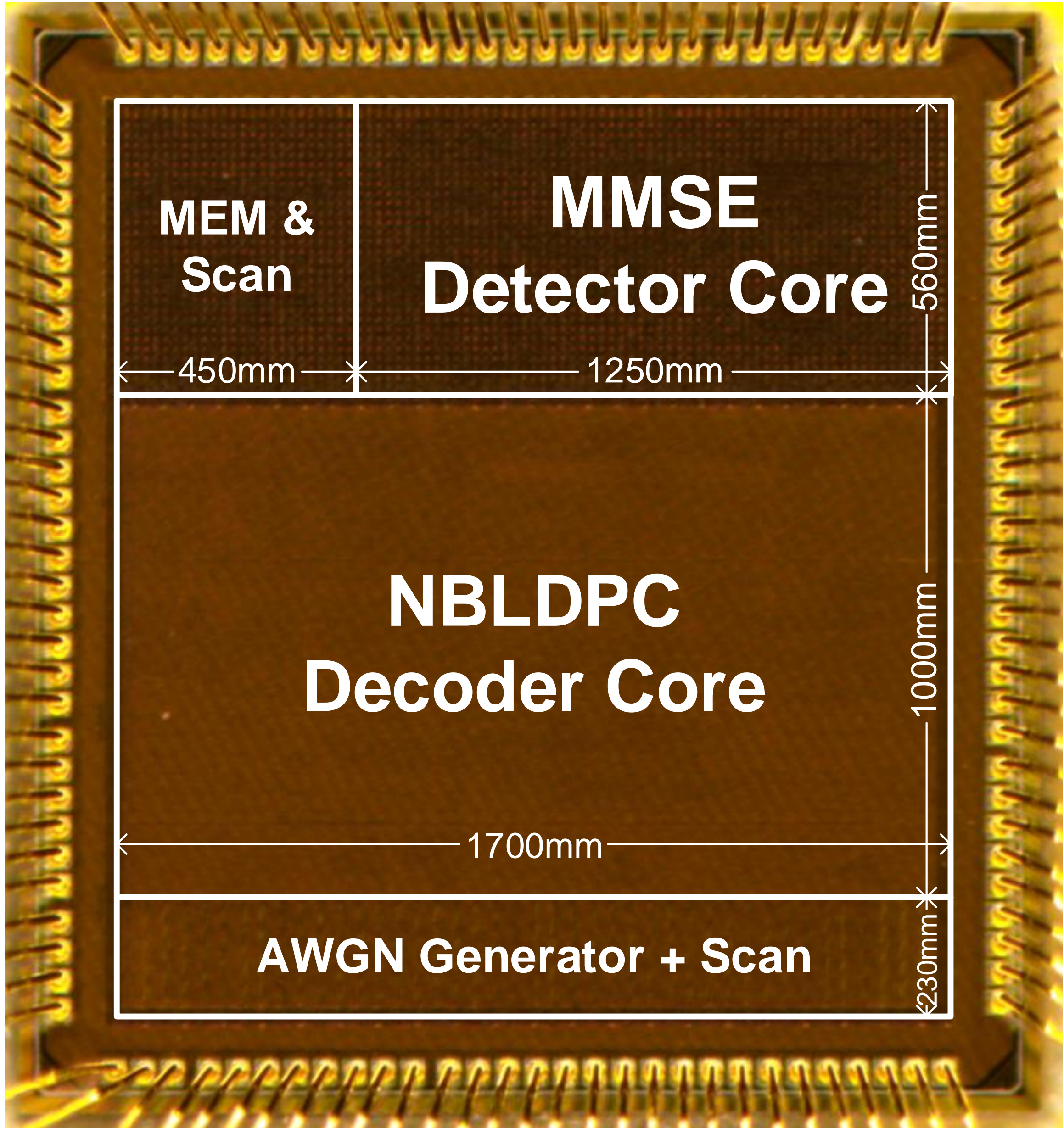


Figure 6.21: Microphotograph of the MMSE-NBLDPC iterative detector-decoder chip in 65nm CMOS.

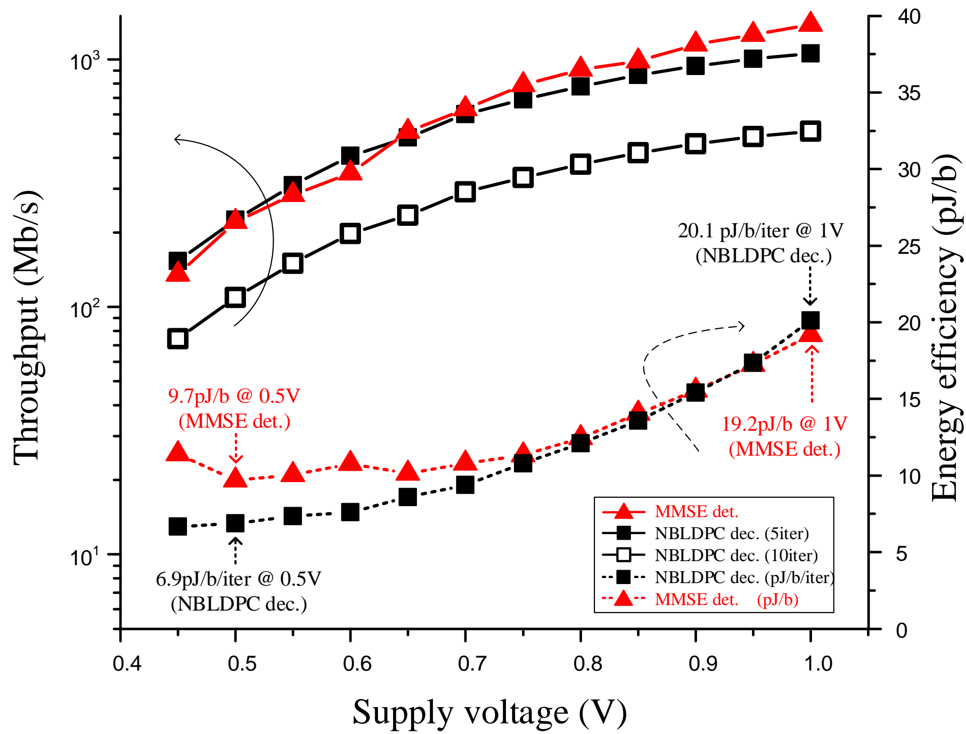


Figure 6.22: Measured throughput and energy consumption of the MMSE detector and the NBLDPC decoder. The throughput is measured at the maximum clock frequency for each supply voltage.

CHAPTER VII

Conclusion and Outlook

7.1 Conclusion

This dissertation presents the study of low-power and error-resilient VLSI design issues, and new techniques to improve the reliability, throughput, and energy efficiency. We summarize the work from a device-level perspective and a system-level perspective.

Device-level perspective: study of transistor and circuit failure
Hold time violation presents a major issue for Vcc scaling and and is a design challenge especially for a low-power system. We performed detailed analysis on Intel's 22nm tri-gate technology. A new hold-time violation metric is introduced to define Vmin as the Vcc in which the hold time exceeds a target percentage (10%) of the cycle time. In the commonly used master-slave flip-flop, our circuit analysis reveals that the hold time Vmin is most sensitive to the variations on the NMOS of the first clock inverter, which needs to be carefully designed for a low-power (low-voltage) system.

Meanwhile, we characterize the impact of soft errors on VLSI DSP systems through test chips exposed under heavy-ion radiation. At a low supply voltage of 0.7 V and low beam energy, the error rates of flip-flops and DSP cores increase by a factor of 2 to 5. At high particle energy, the increase in error rate is almost negligible, suggesting that the charge conveyed by heavy ion strikes has far exceeded the critical charge and tuning the supply voltage is ineffective. Increasing the clock frequency increases the relative importance of transient errors from combinational circuits. The effect is more pronounced in hardened circuit designs. Our results show that implementing hardened circuits for a low-power and high-performance system under heavy-ion radiation is not as effective as previously expected.

System-level perspective: algorithm and architecture co-design
Many VLSI DSP systems are resilient to errors, providing opportunities for more aggressive voltage scaling or algorithm simplification to reduce power. We propose a confidence-driven architecture, which is an architectural technique that can be used to adjust the tradeoff between reliability and performance. In addition, we present an FPGA-based emulation platform to provide a rapid evaluation of the performance and reliability of error-resilient circuit designs.

Algorithm and architecture co-optimization provides more opportunities to improve energy efficiency and performance. Targeting MIMO wireless communications, we propose co-optimization methods to design iterative detection-decoding processor. We demonstrate new methods to improve the detection algorithm, and simplify the interface between the detector and decoder. To enhance the throughput and reduce the energy consumption, we use low-power circuit techniques, including interleaving for the critical blocks, clock gating on the idle registers, and new circuit architecture for reciprocal unit. Our design fabricated in 65nm CMOS technology achieves a high energy efficiency of 19.2 pJ/b and a high throughput of 1.38Gb/s.

7.2 Outlook

Based on the research throughout this dissertation, we briefly outline future research topics.

Device-level perspective: circuit design to monitor errors
(1) Sensors and error detectors are proposed to detect the failure point and indicate the Vmin point for online adjustment, such as temperature sensor and cannery circuit for delay monitoring. However, no hold-time violation sensor has been designed to indicate the hold-time Vmin. Since hold time has become a major concern for low-power design, it will be necessary to have a hold-time monitor on chip to prevent such failure occurrence.
(2) Soft errors can silently corrupt a data storage node without being detected even on radiation hardened circuits. It is therefore necessary to have a more sensitive error detector to signal suspicious behavior especially on the most important part, such as a controller or a finite state machine.

System-level perspective: Algorithm and architecture co-designed DSP

for wireless communication:

(3) No complete iterative MMSE-NBLDPC designs have yet been done because of insufficient algorithm study of the scaling effects between two cores. Hence, researching the efficient scaling, i.e., by shifting, and completely integrating two cores are interesting tasks.

(4) NBLDPC decoder processing with different number of GF elements provides the opportunity to dynamically trade PER performance for a lower power consumption and a higher energy efficiency. Designing an MMSE-NBLDPC IDD system with tunable number of GF elements will make a more efficient and adaptive IDD system for MIMO wireless communication in the future.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] T. Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron CMOS technology," *IEEE Trans. Nuclear Science*, vol. 43, no. 6, pp. 2874–2878, Dec. 1996.

[2] Single-event effect mitigation in RTAX-DSP space-flight FPGAs. [Online]. Available: http://www.actel.com/documents/rtaxdsp_see_wp.pdf

[3] G. Gerosa, S. Gary, C. Dietz, P. Dac, K. Hoover, and J. Alvarez, "A 2.2 W, 80 MHz superscalar RISC microprocessor," *IEEE J. Solid-State Circuits*, vol. 29, no. 12, pp. 1440 –1454, Dec. 1994.

[4] M. Agarwal, B. Paul, M. Zhang, and S. Mitra, "Circuit failure prediction and its application to transistor aging," in *VLSI Test Symposium*, May 2007, pp. 277–286.

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

[6] K. Bowman, S. Duvall, and J. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE J. Solid-State Circuits*, vol. 37, no. 2, pp. 183–190, Feb. 2002.

[7] S. Borkar, "Parameter variations and impact on circuits and microarchitecture," in *ACM/IEEE Design Automation Conf.*, 2003, pp. 338–342.

[8] P. Hazucha and C. Svensson, "Impact of CMOS technology scaling on the atmospheric neutron soft error rate," *IEEE Trans. Nuclear Science*, vol. 47, no. 6, pp. 2586–2594, 2000.

[9] Y. Komatsu, Y. Arima, T. Fujimoto, T. Yamashita, and K. Ishibashi, "A soft-error hardened latch scheme for SoC in a 90 nm technology and beyond," in *Custom Integrated Circuits Conf.*, Oct. 2004, pp. 329–332.

[10] D. Frank, R. Dennard, E. Nowak, P. Solomon, Y. Taur, and H.-S. P. Wong, "Device scaling limits of Si MOSFETs and their application dependencies," *Proc. IEEE*, vol. 89, no. 3, pp. 259–288, Mar. 2001.

[11] International technology roadmap for semiconductors. [Online]. Available: http://public.itrs.net/

[12] S. H. Jo, K.-H. Kim, and W. Lu, "Programmable resistance switching in nanoscale two-terminal devices," *Nano Letters*, vol. 9, no. 1, pp. 496–500, 2009.

[13] S. E. Savelev, A. S. Alexandrov, A. M. Bratkovsky, and R. Stanley Williams, "Molecular dynamics simulations of oxide memristors: Crystal field effects," *Applied Physics Letters*, vol. 99, no. 5, pp. 053 108–053 108–3, Aug. 2011.

[14] H. Qin, Y. Cao, D. Markovic, A. Vladimirescu, and J. Rabaey, "Sram leakage suppression by minimizing standby supply voltage," in *IEEE Intl. Symp. Quality Electronic Design*, 2004, pp. 55–60.

[15] R. Heald and P. Wang, "Variability in sub-100nm sram designs," in *IEEE/ACM Intl. Conf. Computer Aided Design*, 2004, pp. 347–352.

[16] S. Ohbayashi, M. Yabuuchi, K. Nii, Y. Tsukamoto, S. Imaoka, and Y. Oda, "A 65 nm SoC embedded 6T-SRAM design for manufacturing with read and write cell stabilizing circuits," *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 820–829, Apr. 2007.

[17] E. Karl, Y. Wang, Y.-G. Ng, Z. Guo, F. Hamzaoglu, and U. Bhattacharya, "A 4.6GHz 162Mb SRAM design in 22nm tri-gate cmos technology with integrated active Vmin-enhancing assist circuitry," in *IEEE Intl. Solid-State Circuits Conf.*, 2012, pp. 230–231.

[18] O. Hirabayashi, A. Kawasumi, A. Suzuki, Y. Takeyama, K. Kushida, and K. Sasaki, "A process-variation-tolerant dual-power-supply SRAM with 0.179m2 cell in 40nm CMOS using level-programmable wordline driver," in *IEEE Intl. Solid-State Circuits Conf.*, 2009, pp. 458–459.

[19] C.-H. Chen, K. Bowman, C. Augustine, Z. Zhang, and J. Tschanz, "Minimum supply voltage for sequential logic circuits in a 22nm technology," in *IEEE Int. Symp. Low Power Electronics and Design*, 2013, pp. 181–186.

[20] T. C. May and M. H. Woods, "Alpha-particle-induced soft errors in dynamic memories," *IEEE Trans. Electron Devices*, vol. 26, no. 1, pp. 2–9, Jan. 1917.

[21] D. Binder, E. C. Smith, and A. B. Holman, "Satellite anomalies from galactic cosmic rays," *IEEE Trans. Nuclear Science*, vol. 22, no. 6, pp. 2675–2680, Dec. 1975.

[22] J. F. Ziegler and W. A. Lanford, "Effect of cosmic rays on computer memories," *Science*, vol. 206, no. 4420, pp. 776–788, 1979.

[23] E. Normand, "Single event upset at ground level," *IEEE Trans. Nuclear Science*, vol. 43, no. 6, pp. 2742–2750, Dec. 1996.

[24] P. E. Dodd and L. W. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics," *IEEE Trans. Nuclear Science*, vol. 50, no. 3, pp. 593–602, Jun. 2003.

[25] T. Karnik and P. Hazucha, "Characterization of soft errors caused by single event upsets in cmos processes," *IEEE Trans. Dependable and Secure Computing*, vol. 1, no. 2, pp. 128–143, Apr. 2004.

[26] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Dev. Mat. Rel.*, vol. 5, no. 3, pp. 305–316, Sept. 2005.

[27] "Measurement and reporting of alpha particle and terrestrial cosmic ray-induced soft errors in semiconductor devices," *JESD89A*, Aug. 2001.

[28] P. E. Dodd, "Production and propagation of single-event transients in high-speed digital logic ics," *IEEE Trans. Nuclear Science*, vol. 51, no. 6, pp. 3278–3284, Dec. 2004.

[29] P. Shivakumar, M. Kistler, S. Keckler, D. Burger, and L. Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," in *Int. Conf. Dependable Systems and Networks*, 2002, pp. 389–398.

[30] 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 *IEEE/ACM Int. Symp. Microarchitecture*, Dec. 2003, pp. 7–18.

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

[32] S. Valadimas, Y. Tsiatouhas, and A. Arapoyanni, "Cost and power efficient timing error tolerance in flip-flop based microprocessor core," in *IEEE European Test Symposium*, 2012, pp. 8–13.

[33] M. Zhang, S. Mitra, T. M. Mak, N. Seifert, N. J. Wang, Q. Shi, K. S. Kim, N. R. Shanbhag, and S. J. Patel, "Sequential element design with built-in soft error resilience," *IEEE Trans. VLSI System*, vol. 14, no. 12, pp. 1368–1378, Dec. 2006.

[34] M. Kurimoto, H. Suzuki, R. Akiyama, T. Yamanaka, H. Ohkuma, H. Takata, and H. Shinohara, "Phase-adjustable error detection flip-flops with 2-stage hold driven optimization and slack based grouping scheme for dynamic voltage scaling," in *Design Automation Conference, ACM/IEEE*, Jun. 2008, pp. 884–889.

[35] M. Choudhury, V. Chandra, K. Mohanram, and R. Aitken, "TIMBER: Time borrowing and error relaying for online timing error resilience," in *Design, Automation and Test in Europe Conf.*, Mar. 2010, pp. 1554–1559.

[36] C.-H. Chen, Y. Tao, and Z. Zhang, "Efficient in situ error detection enabling diverse path coverage," in *IEEE Int. Symp. Circuits Syst.*, 2013, pp. 773–776.

[37] C.-H. Chen, D. Blaauw, D. Sylvester, and Z. Z., "Design and evaluation of confidence-driven error-resilient systems," *IEEE Trans. VLSI System*, vol. 22, no. 8, pp. 1727–1737, Aug. 2014.

[38] C. Auth, C. Allen, A. Blattner, D. Bergstrom, M. Brazier, and M. Bost, "A 22nm high performance and low-power CMOS technology featuring fully-depleted tri-gate transistors, self-aligned contacts and high density MIM capacitors," in *IEEE Symp. VLSI Technology*, 2012, pp. 131–132.

[39] Xilinx 7 series FPGAs overview. [Online]. Available: http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf

[40] X. Du and W. Chen, "A most probable point based method for uncertainty analysis," *J. Design and Manufacturing Automation*, vol. 4, no. 1, pp. 47–65, Feb. 2001.

[41] D. Khalil, M. Khellah, N.-S. Kim, Y. Ismail, T. Karnik, and V. De, "SRAM dynamic stability estimation using MPFP," in *Init. Microelectronics*, 2007, pp. 167–170.

[42] M. Agostinelli, J. Hicks, J. Xu, B. Woolery, K. Mistry, and K. Zhang, "Erratic fluctuations of sram cache vmin at the 90nm process technology node," in *IEEE Intl. Electron Devices Meeting*, 2005, pp. 655–658.

[43] S. Kumar, C. Kim, and S. Sapatneker, "Impact of NBTI on SRAM read stability and design for reliability," in *IEEE Intl. Symp. Quality Electronic Design*, 2006, pp. 213–218.

[44] E. Seevinck, F. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *IEEE J. Solid-State Circuits*, vol. 22, no. 5, pp. 748–754, Oct. 1987.

[45] e. A. W. Burks., *Theory of self-reproducing automata [by] Neumann, J. von.* Univ. of Illinois Press, 1966.

[46] E. Seevinck, F. List, and J. Lohstroh, "Ion-implanted complementary MOS transistors in low-voltage circuits," *IEEE J. Solid-State Circuits*, vol. 7, no. 2, pp. 146–153, Apr. 1972.

[47] The cyclotron institute at TAMU. [Online]. Available: http://cyclotron.tamu.edu/ref/

[48] N. J. Gaspard, S. Jagannathan, Z. J. Diggins, M. P. King, S.-J. Wen, R. Wong, T. D. Loveless, K. Lilja, M. Bounasser, T. Reece, A. F. Witulski, W. T. Holman, B. L. Bhuva, and L. W. Massengill, "Technology scaling comparison of flip-flop heavy-ion single-event upset cross sections," *IEEE Trans. Nuclear Science*, vol. 60, no. 6, pp. 4368–4373, 2013.

[49] N. N. Mahatme, N. J. Gaspard, S. Jagannathan, T. D. Loveless, B. L. Bhuva, W. H. Robinson, L. W. Massengill, S.-J. Wen, and R. Wong, "Impact of supply voltage and frequency on the soft error rate of logic circuits," *IEEE Trans. Nuclear Science*, vol. 60, no. 6, pp. 4200–4206, 2013.

[50] P. Hazucha, T. Karnik, S. Walstra, B. Bloechel, J. Tschanz, J. Maiz, K. Soumyanath, G. Dermer, S. Narendra, V. De, and S. Borkar, "Measurements and analysis of SER-tolerant latch in a 90-nm dual-Vt CMOS process," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1536–1543, Sep. 2004.

[51] B. Lambrigtsen, W. Wilson, A. Tanner, T. Gaier, C. Ruf, and J. Piepmeier, "GeoSTAR - a microwave sounder for geostationary satellites," in *IEEE Int. Geoscience and Remote Sensing Symp.*, vol. 2, 2004, pp. 777–780.

[52] A. Tanner, W. Wilson, B. Lambrigsten, S. Dinardo, S. Brown, P. Kangaslahti, T. Gaier, C. Ruf, S. Gross, B. Lim, S. Musko, S. Rogacki, and J. Piepmeier, "Initial results of the geostationary synthetic thinned array radiometer (GeoSTAR) demonstrator instrument," *IEEE Trans. Geoscience and Remote Sensing*, vol. 45, no. 7, pp. 1947–1957, 2007.

[53] J. Black, A. Sternberg, M. Alles, A. Witulski, B. Bhuva, L. Massengill, J. Benedetto, M. Baze, J. Wert, and M. Hubert, "HBD layout isolation techniques for multiple node charge collection mitigation," *IEEE Trans. Nuclear Science*, vol. 52, no. 6, pp. 2536–2541, Dec. 2005.

[54] P. Hazucha, C. Svensson, and S. A. Wender, "Cosmic-ray soft error rate characterization of a standard 0.6um CMOS process," *IEEE J. Solid-State Circuits*, vol. 35, no. 10, pp. 1422–1429, Oct. 2000.

[55] V. Joshi, R. R. Rao, D. Blaauw, and D. Sylvester, "Logic SER reduction through flip flop redesign," in *Int. Symp. Quality Electronic Design*, Mar. 2006, pp. 611–616.

[56] D. Rennie, D. Li, M. Sachdev, B. L. Bhuva, S. Jagannathan, S. Wen, and R. Wong, "Performance, metastability, and soft-error robustness trade-offs for flip-flops in 40 nm CMOS," *IEEE Trans. Circuits and Systems I*, vol. 59, no. 8, pp. 1626–1634, 2012.

[57] P. Roche, G. Gasiot, S. Uznanski, J.-M. Daveau, J. Torras-Flaquer, S. Clerc, and R. Harboe-Sorensen, "A commercial 65 nm CMOS technology for space applications: heavy ion, proton and gamma test results and modeling," *IEEE Trans. Nuclear Science*, vol. 57, no. 4, pp. 2079–2088, Aug. 2010.

[58] G. Gasiot, M. Glorieux, S. Clerc, D. Soussan, F. Abouzeid, and P. Roche, "Experimental soft error rate of several flip-flop designs representative of production chip in 32nm CMOS technology," *IEEE Trans. Nuclear Science*, vol. 60, no. 6, pp. 4226–4231, Dec. 2013.

[59] K. Bowman, J. Tschanz, N. S. Kim, J. Lee, C. Wilkerson, S.-L. Lu, T. Karnik, and V. De, "Energy-efficient and metastability-immune timing-error detection and recovery circuits for dynamic variation tolerance," in *Integrated Circuit Design and Technology and Tutorial*, June 2008.

[60] P. Dodd, M. Shaneyfelt, J. Felix, and J. Schwank, "Production and propagation of single-event transients in high-speed digital logic ICs," *IEEE Trans. Nuclear Science*, vol. 51, no. 6, pp. 3278–3284, Dec. 2004.

[61] T. Sakata, T. Hirotsu, H. Yamada, and T. Kataoka, "A cost-effective dependable microcontroller architecture with instruction-level rollback for soft error recovery," in *Int. Conf. Dependable Systems and Networks*, June 2007, pp. 256–265.

[62] C.-H. Chen, Y. Kim, Z. Zhang, D. Blaauw, D. Sylvester, H. Naeimi, and S. Sandhu, "A confidence-driven model for error-resilient computing," in *Design, Automation Test in Europe Conference Exhibition*, Mar. 2011.

[63] N. Carter, H. Naeimi, and D. Gardner, "Design techniques for cross-layer resilience," in *Design, Automation Test in Europe Conf. Exhibition (DATE)*, Mar. 2010, pp. 1023–1028.

[64] C. Lopez-Ongil, M. Garcia-Valderas, M. Portela-Garcia, and L. Entrena, "Autonomous fault emulation: A new FPGA-based acceleration system for hardness evaluation," *IEEE Trans. Nuclear Science*, vol. 54, no. 1, pp. 252–261, Feb. 2007.

[65] A. Pellegrini, K. Constantinides, D. Zhang, S. Sudhakar, V. Bertacco, and T. Austin, "CrashTest: A fast high-fidelity FPGA-based resiliency analysis framework," in *IEEE Int. Conf. Computer Design*, Oct. 2008, pp. 363–370.

[66] E. Johnson, M. Caffrey, G. P., R. N., and W. M., "Accelerator validation of an FPGA SEU simulator," *IEEE Trans. Nuclear Science*, vol. 50, no. 6, pp. 2147–2157, Dec. 2003.

[67] A. Raychowdhury, B. Geuskens, K. Bowman, J. Tschanz, S.-L. Lu, T. Karnik, M. Khellah, and V. De, "Tunable replica bits for dynamic variation tolerance in 8T SRAM arrays," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, pp. 797–805, Apr. 2011.

[68] K. Bowman, C. Tokunaga, T. Karnik, V. De, and J. Tschanz, "A 22nm dynamically adaptive clock distribution for voltage droop tolerance," in *IEEE VLSI Symposium*, June 2012, pp. 94–95.

[69] J. D. Davis, C. P. Thacker, and C. Chang, “BEE3: Revitalizing computer architecture research,” *Technical Report MSR-TR-2009-45, Microsoft Research*, 2009.

[70] K. Bowman and J. Tschanz, “Resilient microprocessor design for improving performance and energy efficiency,” in *IEEE/ACM Computer-Aided Design*, Nov. 2010, pp. 85–88.

[71] Alpha architecture handbook. [Online]. Available: http://h18000.www1.hp.com/cpq-alphaserver/technology/literature/alphaahb.pdf

[72] M. Breuer, S. Gupta, and T. Mak, “Defect and error tolerance in the presence of massive numbers of defects,” *IEEE Design Test of Computers*, vol. 21, no. 3, pp. 216–227, Jun. 2004.

[73] R. Hegde and N. Shanbhag, “Energy-efficient signal processing via algorithmic noise-tolerance,” in *Int. Symp. Low Power Electronics and Design*, 1999, pp. 30–35.

[74] S. Narayanan, J. Sartori, R. Kumar, and D. Jones, “Scalable stochastic processors,” in *Design, Automation and Test in Europe Conf.*, Mar. 2010, pp. 335–338.

[75] H. Cho, L. Leem, and S. Mitra, “ERSA: Error resilient system architecture for probabilistic applications,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 31, no. 4, pp. 546–558, Apr. 2012.

[76] K. Nikolic, A. Sadek, and M. Forshaw, “Fault-tolerent techniques for nanocomputers,” *Nanotechnology*, vol. 13, no. 3, pp. 357–362, May 2002.

[77] K. Constantinides, S. Plaza, J. Blome, B. Zhang, V. Bertacco, S. Mahlke, T. Austin, and M. Orshansky, “BulletProof: a defect-tolerant CMP switch architecture,” in *Int. Symp. High-Performance Computer Architecture*, Feb. 2006, pp. 5–16.

[78] M. Nicolaidis, “Time redundancy based soft-error tolerance to rescue nanometer technologies,” in *IEEE VLSI Test Symp.*, 1999, pp. 86–94.

[79] H. Naeimi and A. DeHon, “Fault-tolerant sub-lithographic design with rollback recovery,” *Nanotechnology*, vol. 19, no. 11, pp. 357–362, Feb. 2008.

[80] 3GPP standards website. [Online]. Available: http://www.3gpp.org

[81] F. Sheikh, C.-H. Chen, D. Yoon, B. Alexandrov, K. Bowman, A. Chun, A. Hossein, and Z. Zhang, “3.2Gbps channel-adaptive configurable MIMO detector for multi-mode wireless communication,” in *IEEE Intl. Signal Processing Systems*, 2014.

[82] M. Damen, H. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Comm.*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

[83] E. Agrell, T. Eirksson, A. Vardy, and K. Zeger, "Closest point searh in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[84] T. D. Chiueh, P. Y. Tsai, and I. W. Lai, "Baseband receiver design for wireless MIMO-OFDM communications," *2nd Eddition, John Wiley and Sons*, 2012.

[85] S. A. Laraway and B. Farhang-Boroujeny, "Implementation of a Markov Chain Monte Carlo based multiuser/MIMO detector," *IEEE Trans. Circuits and Systems - I*, vol. 56, no. 1, pp. 246–255, May 2009.

[86] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, Jul. 2005.

[87] E. M. Witte, F. Borlenghi, G. Ascheid, R. Leupers, and H. Meyr, "A scalable vlsi architecture for soft-input soft-output single tree-search sphere decoding," *IEEE Trans. Circuits Systems II*, vol. 57, no. 9, pp. 706–710, Sept. 2010.

[88] B. Noethen, O. Arnold, E. Adeva, T. Seifesr, E. Fischer, S. Kunze, E. Matus, G. Fettweis, H. Eisenreich, G. Ellguth, S. Hartmann, S. Hoppner, S. Schiefer, J.-U. Schlubler, S. Scholze, D. Walter, and R. Schuffny, "A 105GOPS 36mm2 heterogeneous SDR MPSoC with energy-aware dynamic scheduling and iterative detection-decoding for 4G in 65nm CMOS," in *IEEE Intl. Solid-State Circuits Conf.*, 2014, pp. 188–189.

[89] F. Borlenghi, E. M. Witte, G. Ascheid, H. Meyr, and A. Burg, "A 2.78 mm2 65nm CMOS gigabit MIMO iterative detection and decoding receiver," in *IEEE European Solid-State Circuits Conf.*, 2012, pp. 65–68.

[90] M. Winter, S. Kunze, E. Adeva, B. Mennenga, E. Matus, G. Fettweis, H. Eisenreich, G. Ellguth, S. Hoppner, S. Scholze, R. Schuffny, and T. Kobori, "A 335Mb/s 3.9mm2 65nm CMOS flexible MIMO detection-decoding engine achieving 4G wireless data rates," in *IEEE Intl. Solid-State Circuits Conf.*, 2012, pp. 216–217.

[91] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, Jul. 1999.

[92] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output mimo detection using MMSE parallel interference cancellation," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1754–1765, Jul. 2011.

[93] D. Auras, R. Leupers, and G. H. Ascheid, “A novel reduced-complexity soft-input soft-output MMSE MIMO detector: algorithm and efficient VLSI architecture,” in *Proc. IEEE Conf. Commun.*, 2014, pp. 4722–4728.

[94] Y. S. Park, Y. Tao, and Z. Zhang, “A 1.15 Gb/s fully parallel nonbinary LDPC decoder with fine-grained dynamic clock gating,” in *IEEE Intl. Solid-State Circuits Conf.*, 2013, pp. 422–423.

[95] S. Pfletschinger and D. Declercq, “Getting closer to MIMO with non-binary codes and spatial multiplexing,” in *Proc. IEEE Globecom*, 2010, pp. 1–5.

[96] V. Erceg *et al.*, “TGn channel models,” *IEEE 802.11 document 03/940r4*, May 2004.

[97] J. Erfanian, S. Pasupathy, and G. Gulak, “Reduced complexity symbol detectors with parallel structures for ISI channels,” *IEEE Trans. Comm.*, vol. 42, no. 2, pp. 1661–1671, Feb. 1994.

[98] R. Yazdani and M. Ardakani, “Efficient llr calculation for non-binary modulations over fading channels,” *IEEE Trans. Comm.*, vol. 59, no. 5, pp. 1236–1241, May 2011.

[99] A. Tomasoni, M. Ferrari, D. Gatti, F. Osnato, and B. S., “A low complexity Turbo MMSE receiver for W-LAN MIMO systems,” in *Proc. IEEE Conf. Commun.*, 2006, pp. 4119–4124.

[100] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, “High-throughput architecture and implementation of regular (2, dc) nonbinary LDPC decoders,” *IEEE Trans. Comm.*, vol. 58, no. 5, pp. 1365–1375, May 2010.

[101] H. Kaul, M. Anders, S. Mathew, S. Hsu, A. Agarwal, F. Sheikh, R. Krishnamurthy, and S. Borkar, “A 1.45GHz 52-to-162GFLOPS/W variable-precision floating-point fused multiply-add unit with certainty tracking in 32nm CMOS,” in *IEEE Intl. Solid-State Circuits Conf.*, 2012, pp. 182–183.