Informatics

Utilizing and Extending the Inherent Fault Tolerance Properties of Asynchronous QDI Circuits

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Zaheer Tabassam, MS
Matrikelnummer 12110385

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Steininger

Diese Dissertation haben begutachtet:

Prof. Hong Chen

Assoc. Prof. Petr Fišer

Wien, 1. Jänner 2024

Zaheer Tabassam

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Informatics

Utilizing and Extending the Inherent Fault Tolerance Properties of Asynchronous QDI Circuits

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Zaheer Tabassam, MS
Registration Number 12110385

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Steininger

The dissertation has been reviewed by:

Prof. Hong Chen

Assoc. Prof. Petr Fišer

Vienna, 1st January, 2024

Zaheer Tabassam

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

TU Bibliothek, WIEN Your knowledge hub

Erklärung zur Verfassung der Arbeit

Zaheer Tabassam, MS

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. Jänner 2024

Zaheer Tabassam

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Acknowledgements

Thanks to Almighty ALLAH, who listens to me whenever I speak to Him, especially guiding me throughout this jorney.

My heartfelt appreciation goes to my mother, Rabia, whose belief in the attainability of my goals and tireless efforts defy expression. I extend sincere thanks to my father, Muhammad Nazir, for his unwavering support in every conceivable manner. To my sisters, Saba Tabassam and Muqadas Noor, your steadfast support has been the foundation upon which I built the courage to embark on this endeavor.

I extend my sincere appreciation to Andreas Steininger, my doctoral studies supervisor, who took a chance on me when I was a novice in this field. His consistent support and weekly interactions over the past three years have been instrumental in my progress and the generation of innovative ideas contributing to my research.

A special acknowledgment is reserved for Syed Rameez Naqvi, my supervisor during my master's studies, whose guidance not only introduced me to the asynchronous world but also paved the way to TUWien. His mentorship instilled in me the courage to take bold steps and strive relentlessly towards my objectives.

I am grateful to my dedicated group members, Robert Najvirt, Florian Huemer, Raghda El Shehaby, and Patrick Behal, whose collaborative efforts laid the groundwork for my research activities. The extensive and insightful meetings we shared, filled with valuable suggestions, significantly contributed to the development of my work.

My appreciation extends to my institute fellows, whose unwavering support has been a constant in my journey.

I am indebted to all those who shared positive words of encouragement, providing a comforting backdrop for me during the course of this transformative journey.

Finally, I owe a debt of gratitude to my wife, Khush Bakht, whose companionship has transformed these intense months into a period of joy and love.

This research was partially supported by the project ENROL (grant I 3485-N31) of the Austrian Science Fund (FWF).

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Kurzfassung

Die schlechte Vorhersagbarkeit des Zeitverhaltens ist eines der Hauptprobleme in getakteten Schaltungen. Die inhärente Selbsttaktung asynchroner Schaltungen kann dieses Problem effizient lösen und bietet Anlass, deren mögliche weitere Vorteile zu erkunden. Die Klasse der Quasi Delay-Insensitive (QDI)-Schaltungen stellt aufgrund ihres im Vergleich zu synchronen Schaltungen sehr flexiblen Zeitverhaltens und ihrer Realisierungsstruktur einen guten Kompromiss dar. Allerdings macht ihre ereignisgesteuerte Natur, so wichtig sie für das gewünschte adaptive Zeitverhalten ist, sie anfällig für andere Umwelteinflüsse wie single event transient (SET)s. Dies liegt daran, dass die einzelnen Module der Schaltung den Zeitpunkt der Generierung sowie der Übernahme von Daten selbst entscheiden, was die Datenübernahmefenster unberechenbar macht und die Schaltung für jegliche Signaländerungen, unabhängig von deren Quelle, empfänglich bleiben lässt. In dieser Arbeit untersuchen wir die drei Hauptteile von QDI-Schaltungen – das Speicherelement (Buffer-Grundstruktur), die kombinatorische Logik und das bedingte Steuerelement – während dieser Zeitfenster unter dem Einfluss von SETs.

Die Buffer-Grundstruktur einer QDI-Schaltung basiert auf einem grundlegenden asynchronen Element namens Muller C-element (MCE) (oder anderen Gattern mit Hysterese). Sein Arbeitsprinzip umfasst zwei Modi: den kombinatorischen Modus, bei dem alle Eingänge übereinstimmen, wodurch der Ausgang unmittelbar von diesen bestimmt wird, und den Speichermodus, bei dem der Ausgang durch die interne Speicherschleife des MCE definiert wird. Im Speichermodus ist das MCE anfällig für SETs, an Eingang wie an Ausgang. Der Wechsel zwischen diesen Modi und folglich die Länge der fehleranfälligen Fenster hängt vom Verhalten der benachbarten Module, Quelle und Senke, ab. Wir analysieren das Verhalten von QDI-Schaltungen in Bezug auf SETs bei variabler Geschwindigkeit von Quelle und Senke. Für ein erstes Verständnis beginnen wir unsere Untersuchungen mit einer leeren Pipeline-Schaltung, da der Buffer der aktive Teil im Handshake ist und für die Umwandlung eines SET in einen single event upset (SEU) verantwortlich ist. Wir schlagen verschiedene Techniken vor, die Widerstandsfähigkeit der Buffer-Grundstruktur gegen SETs zu verbessern. Diese Techniken nutzen im Wesentlichen die Informationen des Handshake-Zyklus, um die empfindlichen Fenster der Buffer zu verkürzen. Die Ergebnisse bestätigen, dass unsere vorgeschlagenen Verbesserungen die Schaltungen widerstandsfähiger gegen SETs machen.

Wir untersuchen separat auch die Auswirkungen von MCEs im kombinatorischen Teil

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

und tragen dabei Sorge, transiente Probleme zu lösen, bevor sie die Buffer-Elemente erreichen. Anhand der Analyse einer als Pipeline realisierten Multiplikatorschaltung untersuchen wir zwei bestehende Techniken und schlagen schließlich einen verbesserten Ansatz vor, der bei Erkennung einer transienten Störung innerhalb der kombinatorischen Logik (1) die Speicheraktivität des Buffers einfriert und (2) die QDI-Natur der Schaltung nutzt, um selektiv alle MCEs rückzusetzen, die während dieses Handshake-Zyklus inaktiv sein sollen. Vielversprechende experimentelle Ergebnisse unterstützen die theoretischen Konzepte unseres Ansatzes.

Die gesteuerten Datenflusselemente von QDI-Schaltungen enthalten ebenfalls MCEs. Wir modifizieren diese, um die aktuellen Handshake-Informationen dazu nutzen zu können, SET-bedingte Flanken im jeweils nicht ausgewählten Pfad zu löschen. Die experimentelle Analyse zeigt, dass diese Verbesserungen zur Widerstandsfähigkeit der Schaltung beitragen.

Generell besteht unsere Strategie darin, die intrinsischen Fehlererkennungsfähigkeiten in QDI-Schaltungen bestmöglich zu nutzen, wobei natürlich auch andere Aspekte abzuwägen sind. Die Neuheit unserer Arbeit liegt in der Einbeziehung der Umgebung der QDI-Schaltung (Quelle/Senke) und einer detaillierten Betrachtung der Effekte, die sich aus deren abnormalem Verhalten ergeben können. Da jede Schaltungsvariante zusätzlich zur gewünschten Minderung der SET-Effekte auch weitere Vorzüge und Probleme mit sich bringt, müssen wir gewählte Ansätze sorgfältig anpassen. Letztendlich besteht unser Ziel darin, ein hohes Maß an Widerstandsfähigkeit gegen SETs zu erreichen, und dabei zur Verringerung der dafür nötigen Redundanz das flexible Zeitverhalten von QDI-Schaltungen bestmöglich zu nutzen. Unsere Ergebnisse bestätigen, dass wir tatsächlich eine bessere Widerstandsfähigkeit als die bestehenden Ansätze erreichen konnten.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Abstract

Timing variation is one of the main concerns in clocked circuits. The inherent self-timed property of asynchronous circuits can efficiently address this problem and convince the community to explore its other veiled benefits. The Quasi Delay-Insensitive (QDI) class of circuits, due to its flexible timing compared to synchronous circuits and its realization structure, represents a good trade-off. However, its event-driven nature, while being instrumental for the desired adaptive timing, makes it prone to other environmental effects such as single event transients (SETs). This is because connected modules are free in their generation and consumption timing of data, which makes waiting windows unpredictable and leaves the circuit remaining open to any change regardless of the source. In this thesis, we investigate the main three parts of QDI circuits – storage element (buffer template), combinational logic and conditional control element – during these windows under the influence of SETs.

The buffer template of a QDI circuit is based on a basic asynchronous element called Muller C-element (MCE) (or other gates with hysteresis). Its working principle comprises two modes, combinational mode when all inputs match, meaning the output is strongly indicated by the inputs, and storage mode when the output is defined by the internal storage loop of the MCE. In storage mode the MCE is susceptible to SETs from input as well as output. The alternation between these modes and consequently its susceptible window length is depended on its connected environment, source and sink. We analyze the behavior of QDI circuits in relation to SETs for variable speed of source and sink. For a first understanding, we start our investigations with an empty pipelined circuit, as the buffer is the active part in the handshaking, and it is responsible for converting an SET into an single event upset (SEU). We propose several techniques to improve the resilience of the buffer template against SETs. Basically these techniques utilize the handshake cycle information to shorten the armed (sensitive) windows of the buffers. The results confirm that our proposed enhancements make the circuits more resilient against SETs.

We also examine the effects of MCEs in the combinational part separately and strive to resolve transient issues before they reach the buffer elements. Using a pipelined multiplier circuit for our analysis we analyze two respective state-of-the-art techniques and finally propose an improved technique which, upon detection of a transient within the combinational logic, (1) freezes the data latching activity of buffer, and (2) leverages the QDI nature of the circuit to selectively flush all those MCEs that are supposed to be

zero during this handshake cycle. Promising experimental results back the theoretical concepts of our approach.

The conditional control elements of QDI circuits also comprise MCEs. We modify these in a way that the current handshake information is used by the unselected path to nullify transitions caused by SETs. The experimental analysis shows that these enhancements contribute well to the resilience of the circuit.

In general, we strive to make best use of the intrinsic fault detection capabilities in QDI circuits while considering other trade-offs. The novelty of our work lies in addressing concerns in detail that arise when the environment behaves abnormally in a QDI circuit. As each variation reveals its own problems and benefits on top of the novel techniques we propose for mitigating SET effects, we need to carefully adjust suitable approaches. The ultimate objective is to attain a high level of resilience against SETs, also utilizing their inherent delay insensitivity while minimizing redundancy. The results confirm that we successfully achieved better resilience than the state-of-the-art approaches.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Contents

Kurzfassung **ix**

Abstract **xi**

Contents **xiii**

1 Introduction **1**
1.1 Research Questions 2
1.2 Thesis Organization and Publications 4

2 Asynchronous QDI Logic and its Fault Tolerant Techniques **7**
2.1 Quasi Delay Insensitive Design: Isochronic Fork Assumption 7
2.2 Pipeline Load Factor 14
2.3 Circuit Description and Fault Injection simulation environment 16
2.4 Target Circuits 19
2.5 Fault-tolerance techniques from literature 21

3 Fault Tolerance Behavior of QDI Buffer Templates **23**
3.1 Weak-Conditioned Half Buffer (WCHB) behavior under SET's 23
3.2 InterlockingWCHB under the influence of SETs 31
3.3 Input/Output-Interlocking WCHB for the mitigation of SET effects . 40
3.4 Input Output Interlocking with SR latch WCHB 50
3.5 Δ: Dual_Completion_Detection Input Output Interlocking WCHB . 60

4 The behavior of QDI Combinational Logic under the influence of SETs **73**
4.1 SET effects in DIMS Combinational Logic with a WCHB buffer template 73
4.2 The SE Tolerance Asynchronous Pipeline I approach for flushing SEUs from DIMS combinational logic with a WCHB buffer template 78
4.3 The SE Tolerance Asynchronous Pipeline II approach for mitigating SET effects in DIMS combinational logic with a WCHB buffer template . . 85
4.4 SETs effects in DIMS Combinational Logic with the Δ buffer template 93
4.5 SETAPI approach for flushing SEUs from DIMS combinational logic with the Δ buffer template 97

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

4.6 ζ approach to flush SEUs from DIMS combinational logic with a modified Δ buffer template . . . 104

5 SET suseptibility of QDI Conditional Control Elements **115**
5.1 SET effects on QDI *Multiplexer* and *De-multiplexer* . . . 115
5.2 Proposed SET resilience enhancements for QDI *Multiplexer* and *De-multiplexer* . . . 121
5.3 Final Comments on Δ_E and $\zeta\_\Delta'\_E$ . . . 128

6 Conclusion and Future Work **131**
6.1 Resilience of Buffer templates . . . 131
6.2 Combinational logic Flushing Techniques . . . 133
6.3 Towards the resilient Conditional Control Elements . . . 134
6.4 Comparing SET Effects in Quasi Delay Insensitive and Synchronous Circuits . . . 135
6.5 Future Work . . . 135

List of Figures **137**

Bibliography **143**

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

CHAPTER 1

Introduction

Most digital circuits are basically composed of two elementary parts: computation (main functional part) and storage. Data computed by combinational logic is latched into storage elements with a special signal transition. In synchronous design these transitions are delivered by a pre-calculated fixed digital pulse called global clock. The estimation of the "clock" signal's period considers the longest combinational delay path. In contrast to that, in asynchronous design [Spa20] the signal transitions required for triggering the latching are locally generated by two connected modules following some special protocol.

As delays continue to become increasingly unpredictable and variable in modern VLSI technologies, the worst-case delay assumption underlying the synchronous design paradigm is getting cumbersome and inefficient. Justifying the abstraction of synchronous design, namely that all flip-flops of the circuit are triggered at precisely the same time instances, demands high design efforts and optimal energy distribution [LKM10, Fri01, DIBM03].

Consequently, asynchronous design techniques, whose closed-loop timing design can flexibly adapt to delay variations, are considered an attractive alternative. Asynchronous design methods are also known for their superior energy efficiency [CRS21] and low power dissipation. Reduced electromagnetic emission, higher operating speed, and better modularity are among a few other traits associated with asynchronous logic design [Spa20]. In context with new architectures [BBN22, LHT+21, NKD+21, DWO+21] asynchronous circuits are also moving into the spotlight.

Their flexible timing makes asynchronous circuits, most notably the so-called QDI ones, also robust against fault effects that impact the circuit's temporal behavior. As mentioned earlier, QDI circuits employ local handshakes instead of a global clock to control the progress of computation and data exchange. These handshakes essentially form closed control loops that precisely adapt the timing to the actual delays, hence QDI circuits can easily accommodate delay variations, and they are robust against delay-related types of faults. However, their operation is event-based (rather than level-based as in synchronous

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

designs), which makes them vulnerable to SETs. Their relatively large data accepting windows [HNS20] make them quite susceptible to environmental effects, as well as other types of transient faults in the value domain, like glitches. Unfortunately, as feature sizes shrink in modern technologies, the rate of SETs is also increasing. Hence, making QDI circuits more resilient against SETs becomes a very relevant issue.

Proven fault-tolerance techniques known from the synchronous domain usually cannot directly be transferred into the asynchronous domain. However, in the literature several specific techniques have already been proposed to make asynchronous designs more resilient against SETs, some of them were discussed by [Hue22]. [Sak21] also investigates QDI circuits in this respect in a detailed manner. This thesis sets out to perform a comprehensive study on how QDI circuits react to transient faults in the value domain (specifically SETs), elaborate suitable and efficient hardening approaches, and finally give experimental evidence for the improvements achieved.

1.1 Research Questions

In this work we investigate the behavior of all parts of a QDI circuit including buffer template, combinational logic and conditional control elements under SETs. Towards this end, our strategy will be to first investigate the root cause of errors, and based on these insights then propose effective methods for their mitigation.

1.1.1 Resilience of QDI circuits against SETs

The environment in which the QDI circuits are operating defines their operating frequency. This is not necessarily under (full) control of the circuit and depends on how fast the data is accepted by the sink (receiver) as well as how fast new data is available from the source (sender). This essentially unknown behavior defines the mode in which the circuit operates; sometimes the source is slow compared to the sink, sometimes it is vice versa and sometime both source and sink respond equally fast or even without any delay.

This leads to the main questions, namely

How can the resilience of a QDI pipeline depend on the speed of its environment, and how can this knowledge be leveraged to improve the resilience?

Can the resilience obtained by existing approaches for SET hardening of QDI pipeline buffers be improved without having to resort to (expensive) replication approaches?

To answer these main questions, these scenarios must be analyzed with the following subquestions.

- While waiting for source and sink, which nodes of a buffer template are susceptible to SETs? What is the resilience without these delays?
- Where are susceptible windows in each mode of operation.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

- What types of error are generated as SET effects?
- Are these error types the same for different modes? Does their probability depend on the mode?
- What are the main causes behind the generation of different types of errors for different modes?
- How do the state of the art fault tolerance approaches addresses this issue?
- Does QDI also offer any inherent fault tolerance properties for transient faults?
- If yes, how much can we benefit from it?
- How can we extend these inherent fault tolerance properties without utilizing full duplication (circuit replica)?
- The principle of causality inherent to QDI circuits seems to prohibit that a glitch, as an effect of an SET, propagates to the output without changing its nature to a stable value. However, literature reported glitches from QDI circuits due to SET strikes. What are the main reasons behind the observed glitch appearance?

With the knowledge of how circuits behave in each mode of operation, and what are the main causes of error generation, we can efficiently deal with the fault effects. We then add combinational logic to the picture with one main research question and several subquestions.

Does specific protection of combinational parts of a QDI circuit significantly contribute to the overall robustness, and if so, how can it be accomplished in an efficient way?

- What is the contribution of QDI combinational logic to the overall error rate?
- What is the main reason of its susceptibility?
- How does the community address this issue?
- How effective are the existing methods? Where are their limitations? Can we do better?

After analyzing the resilience of linear pipelined circuits, we extended our work to non-linear pipeline QDI circuits. Non-linear QDI circuits also include conditional control elements which require extra consideration. So our next research question, along with its associated subquestions relates to these.

Does specific protection of conditional control elements within a QDI circuit significantly contribute to the overall robustness, and if so, how can it be accomplished in an efficient way?

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

- What is the contribution of QDI conditional control elements to the overall error rate?
- Which path of a conditional control element is more susceptible to SET and why?
- How we can enhance the resilience of a more susceptible path?
- Do we have to add extra circuitry, or we can utilize the QDI nature of the circuit to enhance its resilience?
- Is the position of a conditional control element relevant for its susceptibility?

The main approach to analyze the circuits is fault-injection simulation. Through a careful mix of deterministically chosen and randomly varied parameters we try to cover the relevant parameter space. The experimental analysis serves two purposes: Firstly, it provides a good indication of the effects SETs can have in the chosen target circuit and hence points to weak spots. Secondly, it allows a quantitative robustness assessment, which is important for a comparison of approaches and the judgement of improvements attained through circuit enhancements.

1.2 Thesis Organization and Publications

In the next chapter we will start with the basics of asynchronous and specifically the QDI class of circuits. We will discuss the MCE, a key building block of QDI circuits. Most parts of the buffer template, combinational logic and conditional control element are composed of MCEs, so, to get a basic understanding, we first discuss its susceptibility against SETs in isolation. Then we review the encoding scheme and Delay-Insensitive Minterm Synthesis (DIMS) style to implement a QDI combinational logic, we also discuss the WCHB, a baseline buffer template for implementing a QDI pipeline. We also give a detailed view of Pipeline load factor (PLF) and our simulation tool for fault-injection experiments with all settings of the circuit. Finally, in this chapter we also introduce the target circuits for our evaluation study, and, most importantly, the literature review.

In Chapter 3, we thoroughly analyze the behavior of different QDI buffer templates under SETs. We present a detailed view of buffer templates with all important parameters. We published the results of our susceptibility analysis of state-of-the-art fault tolerant buffer templates in

P. Behal, F. Huemer, R. Najvirt, A. Steininger and Z. Tabassam, "Towards Explaining the Fault Sensitivity of Different QDI Pipeline Styles," 2021 27th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC), Beijing, China, 2021, pp. 25-33.

We follow up with the findings from [HNS20] and extend their work with a resilient approach addressing the susceptible windows. We introduce a novel approach called Input Output Interlocking WCHB which we also published as

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Z. Tabassam, P. Behal, R. Najvirt and A. Steininger, "Input/Output-Interlocking for Fault Mitigation in QDI Pipelines," 2021 Austrochip Workshop on Microelectronics (Austrochip), Linz, Austria, 2021, pp. 17-20.

Next, we present the thorough investigations of that approach which suggest refinements to achieve higher (or at least the same) resilience with a simpler architecture. We introduce a further improved the buffer template that was published in

Z. Tabassam and A. Steininger, "Towards Resilient QDI Pipeline Implementations," 2022 25th Euromicro Conference on Digital System Design (DSD), Maspalomas, Spain, 2022, pp. 657-664.

In a next step we elaborate a buffer template we call Δ which shows high resilience against SETs for a variety of PLFs. The behavior is achieved by minimizing the sensitive window of the circuit. The analysis presented is based on the publication

Z. Tabassam and A. Steininger, "SET Hardened Derivatives of QDI Buffer Template," 2022 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Austin, TX, USA, 2022, pp. 1-6.

Chapter 3 not only presents the published SET hardened QDI buffer templates but also provides more elaborate analyses from different angles to understand their behavior under SETs.

After having elaborated a somewhat satisfactory protection for the buffer templates, we start investigating the contribution of combinational logic to the overall error rate in Chapter 4, and its main root cause. Based on a thorough analysis we propose an enhancement for the state-of-the-art combinational logic flushing approach. It comes along without flushing the whole combinational logic and without adding extra computation cycles, but instead carefully identifies the fault and selectively flushes the stored erroneous transitions. This idea was published in

Z. Tabassam, A. Steininger, R. Najvirt and F. Huemer, "ζ: A Novel Approach for Mitigating Single Event Transient Effects in Quasi Delay Insensitive Logic," 2023 28th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC), Beijing, China, 2023, pp. 48-57.

To address the research questions regarding the QDI conditional control elements we investigate their behavior in Chapter 5 in detail and present techniques to enhance their resilience against SETs. More specifically, we propose a method which considers for each path of conditional control elements when it is safe to latch data and when not. These techniques are presented in

Z. Tabassam and A. Steininger, “Towards Resilient Quasi Delay Insensitive Conditional Control Elements”, 2023 26th Euromicro Conference on Digital System Design (DSD).

To generate the QDI circuits (VHDL file) we utilize the tool designed by our group members Florian Huemer [Hue22] and Robert Najvirt. This tool is basically written in Python language where the final file form is then passed to the fault-injection simulation

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

environment developed by Patrick Behal and Robert Najvirt [BHNS21, Beh21]. My contribution is the thorough analysis of QDI circuits under SETs and the proposed enhancements that yield better resilience. In my experiments I also identified the residual sensitive windows for future work. The thesis is summarized in Chapter 6 by highlighting the main contributions.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

CHAPTER 2

Asynchronous QDI Logic and its Fault Tolerant Techniques

This chapter introduces the basics of QDI logic including its timing assumption, basic architecture, its environment and the most important its behavior under variable delays from source and sink. As the focus of thesis is fault tolerance of QDI circuits we discuss the state-of-the-art fault tolerance techniques. This chapter also introduces all terms that are relevant in any context with the work presented in the next chapters.

2.1 Quasi Delay Insensitive Design: Isochronic Fork Assumption

The term QDI comes from delay insensitivity: the circuit operates without any timing assumptions, which means wires and gates can have arbitrary delays. In QDI we constrain a delay-insensitive (DI) circuit with an isochronic fork assumption to allow its realization with basic logic gates [Mar86]. The isochrnoic assumption suggests that the different branches of the fork must observe the same delay. Figure 2.1 explains the isochronic fork assumption: line "F" forks to a NOT and an OR gate, when both branches have the same wire delays, we can guarantee the appearance of the high signal at OR-gate input "B" before input "A" goes low. In that manner OR-gate output "C" is glitch free.

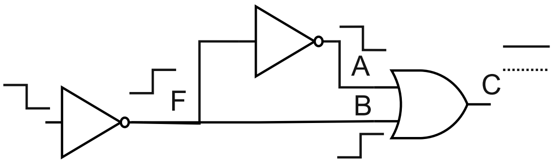


Figure 2.1: Isochronic fork assumption: reprinted from [BOF10]

Asynchronous circuits may also be realized with some other approaches e.g. bundled data, hybrid techniques etc. All approaches can be found in [BOF10].

2.1.1 The MCE

A fundamental building block of QDI circuits is the Muller C-element (MCE). Its functionality is simple: only the logic level of matching inputs is passed to the output and retained until a matching pattern with the opposite logic level arrives, Figure 2.2 shows a symbolic representation of a MCE and its variants.

As Figure 2.2 suggests, a regular MCE's output changes for symmetric inputs after the inertial delay of the gate. This also holds for the MCE with negative input (C-element-); however, here for up transitions (only) the input $NegIN$ has no impact. Likewise, the MCE with positive input (C-element+) ignores $PosIN$ for the down-transition.

The MCE is a storage element that works on a simple matching and latching principle: only matching inputs are latched.

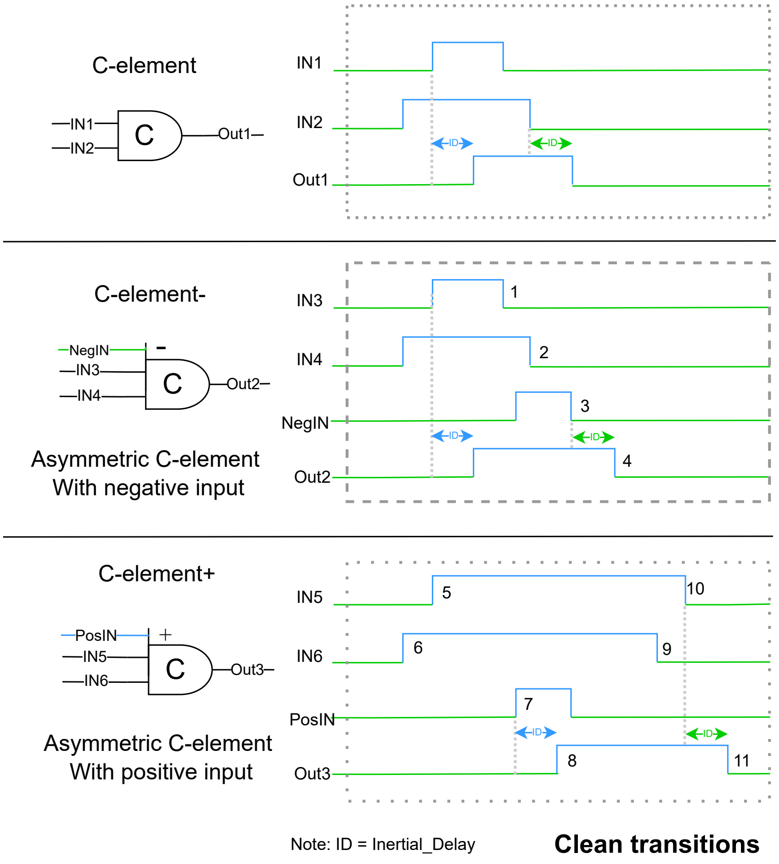


Figure 2.2: MCE and its derivatives normal behavior

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

2.1.2 Modeling a C-element

Algorithm 2.1 is a detailed model of the MCE and is consistently referenced throughout this work. If asymmetric inputs are added, they are treated as normal inputs, subject to validation under one of the respective conditions, as specified. In the case of adding a reset and activating it, the MCE simply flushes the stored value. The special input *sp* has unique functions; if added, it must be checked for *SET* conditions. If *sp* is low and the second input *IN2* is also low, it activates the reset condition. Additionally, if the input *n* is added and goes low, it forcefully resets the MCE when in storage mode.

Algorithm 2.1: MCE Model

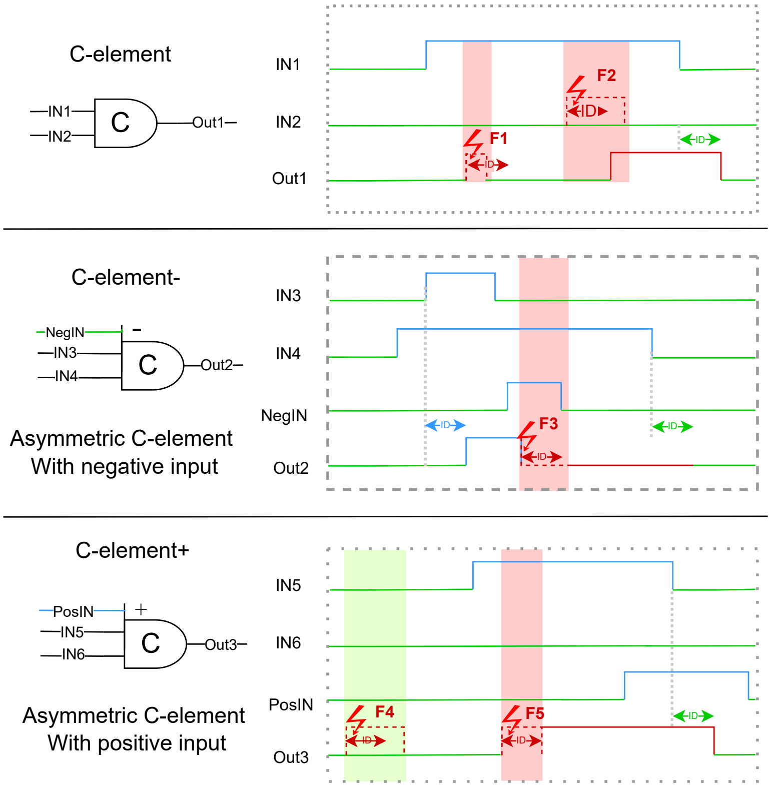
```
Comment: ∧ is logical AND, ∨ is logical OR, and ∼ is logic inversion
Inputs   : IN1, IN2 , Asymmetric+, Asymmetric-, Reset, n, sp
Comment: Only IN1 and IN2 serve as primary inputs; all others are included as
per the requirements
Output   : Out1
if $IN1 \wedge IN2\ (\wedge\ Asymmetric+)\ (\wedge \sim sp)$ then
    Out1 = 1 after (Inertial Delay);
    Comment: Condition 1 – set
else if $(\sim (IN1 \vee IN2\ (\vee\ Asymmetric-))) \vee Reset$ then
    Out1 = 0 after (Inertial Delay);
    Comment: Condition 2 – reset
else if $\sim (IN1 \wedge IN2) \wedge \sim n$ then
    Out1 = 0 after (Inertial Delay);
    Comment: force reset with “n”
else if $\sim IN2 \wedge \sim sp$ then
    Out1 = 0 after (Inertial Delay);
    Comment: force reset with “sp”
else if $Out1$ then
    Out1 = 1 after (Inertial Delay);
    Comment: Condition 3 – hold 1
else if $(\sim Out1)$ then
    Out1 = 0 after (Inertial Delay);
    Comment: Condition 4 – hold 0
```

2.1.3 Sensitive Areas of the $C-element$ and SETs

Technology advancements pose some challenges, one being SET effects, [Kob20, TPES22], and [FCMG13] discussed these effects on digital circuits in comprehensive manners. As our focus is QDI and the main element is the MCE that converts SETs to soft errors, we only highlight the effects of an SET with respect to the MCE. MCEs has a major contribution in error generation and propagation, but also masking. The right part of Figure 2.3 illustrates possible effects on the three MCE types under fault (SET) scenarios.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

These are numbered F1 to F5. Discussing all fault combinations is not feasible here – the purpose of the following discussion is to show some principal effects.



Note: ID = Inertial_Delay **Faulty transitions**

Figure 2.3: MCE and its derivatives under SETs

1. F1: A fault pulse hits the MCE output Out1 while in storage mode (non-matching inputs). In the physical implementation this fault would attempt to flip the MCE's storage loop from the output side, but the one shown here is too short to overcome the element's inertia.
2. F2: The MCE input rail In2 is hit by a fault. While it persists, the inputs match. Since this time the fault pulse is sufficiently long, Out1 is indeed flipped and remains at the erroneous level even after the fault vanished.
3. F3: A negative fault pulse forces Out2 to "0". Note that right before the fault occurred, the C-element- moved to storage mode, by a transition on IN3. However, in contrast to F1, this fault is long enough to flip the state.
4. F4: A positive fault pulse affects Out3 of C-element+. Even though its length is sufficient, it has no effect because the MCE is in combinational mode.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

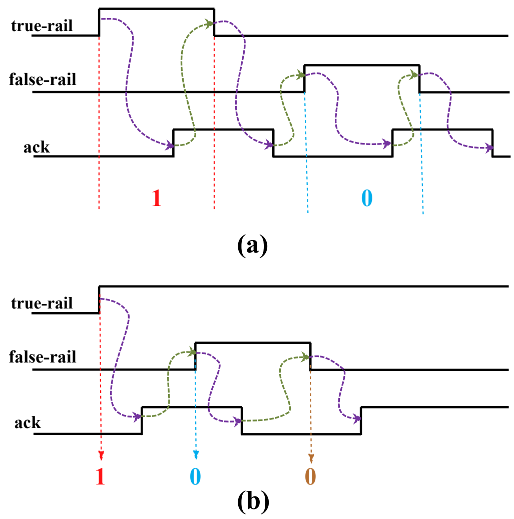


Figure 2.4: Dual rail encoding (a) 4-phase, (b) 2-phase

5. F5: A positive fault pulse hits Out3 while in storage mode. Its length is sufficient to flip the state – even though PosIN is at LO.

2.1.4 Dual Rail Protocol

As QDI circuits fall under the self-timed circuit category, data must be encoded in a way that confirms its validity to connected modules. In this work we mostly used the dual-rail encoding scheme with 4-phase handshake protocol [Spa20]. As in Figure 2.4, for each data bit we have two rails tagged "true-rail" and "false-rail". Figure 2.4-a illustrates the dual rail 4-phase handshaking protocol between source and sink, "true" and "false" rails are from source, while "ack" is an explicit signal from sink that confirms valid data reception. A logical “1” is represented by setting these rails to (1, 0) and “0” by (0, 1). These are code words and called (data) tokens. The code (0, 0) is used as a spacer, as demanded in the 4-phase protocol to separate data tokens. Note that for the considered scheme (1, 1) is an illegal pattern. After receiving a logical high acknowledgement signal (when considering logic low as reset) from the receiver the source changes the data token for a spacer. The handshake is completed with its 4^{th} phase when the receiver responds by resetting the acknowledgment. On the other hand, in the 2-phase handshake protocol as presented in Figure 2.4-b, each transition on a data rail presents new data while each transition on the acknowledgement line presents completion of a handshake cycle.

2.1.5 Completion Detection

As discussed, the data is received as encoded word, and the sink must acknowledge each "token" or "transition" depending on the protocol. Figure 2.5 presents a completion

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

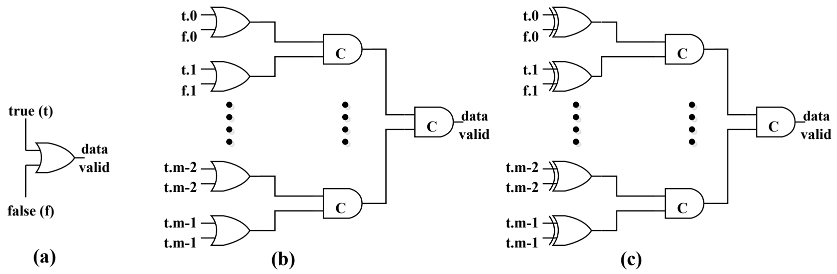


Figure 2.5: Completion detection logics (a) 4-phase signaling with 1-bit message, (b) 4-phase signaling with m-bit message, (c) 2-phase signaling with m-bit message

sensing circuit to establish the command to confirm the reception. Figure 2.5-a generates the acknowledgement signal for dual rail 1-bit data while observing the 4-phase protocol, as only one rail made transitions during each handshake cycle, the OR gate is enough to check the data and spacer. Figure 2.5-b is for m-bit's, here we use MCE's that will ensure the generation of a completion signal once all rails have made their transitions. Last Fig. 2.5-c generates the acknowledgement signal when we have m-bits, and we are running with a 2-phase handshake protocol. [Sri22] also discusses completion detection in asynchronous circuits in more detail.

2.1.6 Dual Rail Combinational Logic

DIMS [SS93], because of its delay insensitive property is one valid choice to implement combinational blocks of QDI circuits. The DIMS 2-input AND gate in Figure 2.6 is comprised of four MCE's which generate minterms of the dual rail bits a, b. With these minterms, the output can be easily generated using an OR gate as shown. As in Figure 2.6, its AND gate so the *out.t* only goes high when both true rail of each bit (a and b) are high and for all other combinations *out.f* is "1". Circuits realized with DIMS are strongly indicating ones: an output is only generated when inputs are complete.

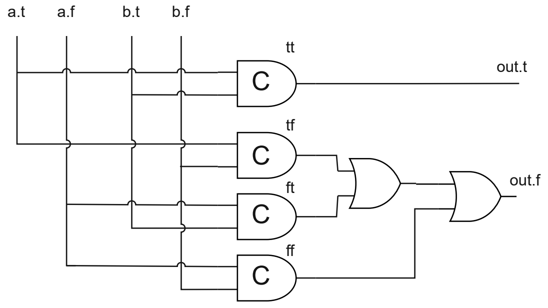


Figure 2.6: DIMS AND Gate

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

The second method we list was presented by [FB96], also known as Null Convention Logic (NCL), to design a logic block with threshold gates. Figure 2.7 presents a NCL AND gate. The presented NCL AND gate is strongly indicating but not all NCL circuits are strongly indicating.

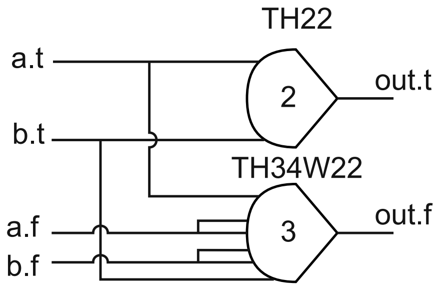


Figure 2.7: NCL AND Gate

Based on NCL, [KL02] introduced the NCL_X flow, the main theme is to separately treat the combinational and completion detection parts. In this way a designer can easily optimize the circuit without violating delay insensitivity of the circuit. [Hue22] also discusses NCL_X [KL02] in a very precise way. The highlighted difference compared to NCL is the extra completion detection from the combinational part of the circuit. There are also other methods available to design dual rail combinational logic block that can be benefited from [Spa20].

2.1.7 Weak-Conditioned Half Buffer

Figure 2.8-A presents a WCHB [BOF10] based 2-bit, 2-stage pipelined QDI [Spa20] circuit, and its waveform in Figure 2.8-B explaining the basic functionality. For the event driven configuration, each bit is encoded by dual-rail protocol using *.T* and *.F* rails. Recall the dual rail configuration: To present a logical *1* from *Buffer_0* to the next stage *Buffer_1*, *Bit-0*'s *Out00.T* is set while keeping *Out00.F* low. Likewise, on *Bit-1* the logical *0* is encoded by setting *Out01.T* low and *Out01.F* high as Figure 2.8-B suggests (with null *Function* assumption). As already discussed, both rails *.T and .F* are not allowed to go high at once; this is considered as illegal code word in dual rail encoding scheme. This array of bits (in our case 2) with one rail set for each are also called data tokens. By sending one, we initiate the first phase of a 4-phase handshake cycle. As shown, when *en1* goes high the data waiting at the inputs of *Buffer 1* is latched and by using an explicit signal named *Acknowledgement* (*Ack1_out*) the receiver stage tells the sender about the successful reception of the data token; in this way we achieved the second phase of our communication cycle. Each pair of subsequent data tokens must be separated by a *Bubble*: here both, *.T* and *.F* rail of each bit are reset by the sender stage on the reception of rising *Ack1_out*, it corresponds to the third phase and in response to this bubble (spacer) *Ack1_out* signal will go back to low which is the 4^{th} and last

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

phase of the handshake cycle. That is why this whole mechanism is known as 4-phase handshake or return to zero protocol, as we go to zero after each data token.

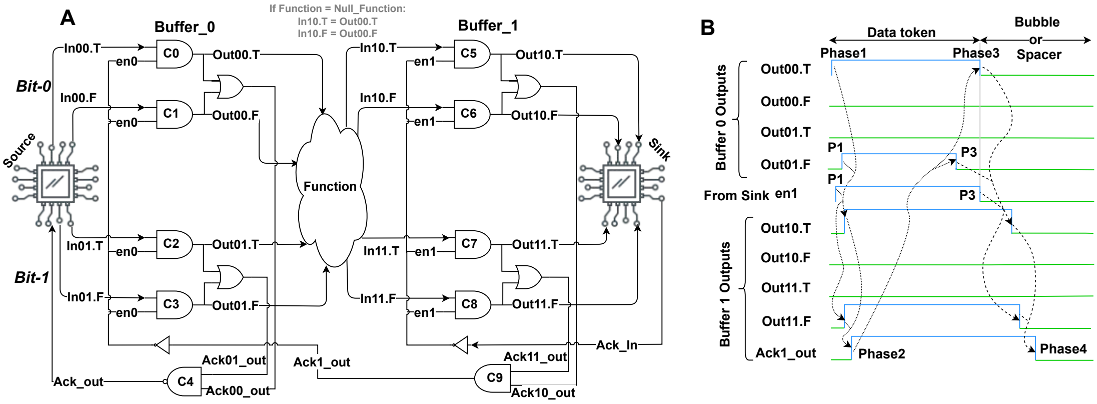


Figure 2.8: A: 2-bit, 2-stage QDI pipelined circuit realized with Weak-Conditioned Half Buffer (WCHB) buffer template, B: Waveform of Circuit in A

The term weak-conditioned comes from weak-conditioned logic [SM80]: validity of input is assumed by the validity of output; these circuits are also known as strongly indicating circuits. Moreover, when pipeline is completely filled, only one "Buffer" from two consecutive ones stores a valid token. That's why it is called half buffer. More information can be found in [BOF10], they also discuss other QDI buffer templates e.g. Pre-Charged half Buffer (PCHB), Pre-Charged full Buffer (PCFB) etc. Throughout this work our focus will be WCHB.

2.2 Pipeline Load Factor

In Section 2.1.7 we discussed how QDI circuits interact with their connected environment. In this section we look how they flexibly adapt their operational speed to the speed at which tokens are provided by the source and consumed by the sink. To illustrate that, we are referring to the Figure 2.9 and Figure 2.10 that demonstrate *Bubble_Limited_Mode* and *Token_Limited_Mode* respectively.

1. Figure 2.9-a: a symbolic representation of a 3-stage QDI pipelined circuit with QDI source and sink. From source to sink data moves on dual-rail (DR) data lines where from sink to source the explicit acknowledgement signal conveys the reception of data tokens. The red color filled file symbols represent the data token while green color empty file symbols are for spacer.
2. Figure 2.9-b: The circuit is in a reset state, all stages contain bubbles, which means all the *ACK* signals are high, demanding the data token from predecessor stages.
3. Figure 2.9-c: 1^{st} token is issued by source and presented to the first stage.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

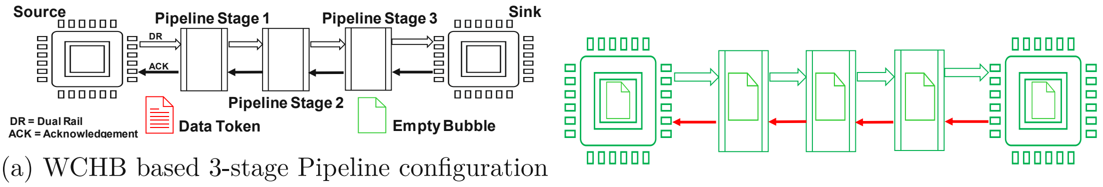
4. Figure 2.9-d: 1^{st} token is successfully latched by the first stage; this is explicitly indicated by the *ACK* signal (bubble) as it goes low (green). In addition, this signal tells the predecessor to replace the data token with a spacer.
5. Figure 2.9-e: 1^{st} token is successfully latched by the second stage, where the first stage now contains the spacer and requests the source for the 2^{nd} token.
6. Figure 2.9-f: 2^{nd} token is latched by the first stage while the 1^{st} token is now available for sink.
7. Figure 2.9-g: Sink is not responding, the data token at the input of the sink experiences a delay in latching. At this stage our pipeline is filled; to request a new token we have to retire the already latched tokens and accept the spacer provided by source. The pipeline needs a new bubble to proceed further. This situation when the sink is slow, is termed *Bubble Limited Mode*.
8. Figure 2.9-h: after some arbitrary delay *ACK* goes low from the sink (bubble), which is the indication that the awaited token is successfully latched by sink.

Figure 2.9 presents the basic data movement within the QDI pipeline with the effects of sink delay. Figure 2.10 is the continuation of the activity from Figure 2.9 but this time we demonstrate the effects of source delay.

1. Figure 2.10-a: a spacer took the place of the 2^{nd} Token in the first pipeline stage as it moved to the second stage.
2. Figure 2.10-b: the first stage requests a new token from the source that will be the 3^{rd} in number, but the source is not responding yet.
3. Figure 2.10-c: 2^{nd} The token proceeds further to the third pipeline stage while we are still waiting for a new token in first pipeline stage.
4. Figure 2.10-d: The 2^{nd} token is successfully latched by the sink; the source did not respond with a new token.
5. Figure 2.10-e: the third pipeline stage replaces the token with a spacer (generated by the source in Figure 2.10-a), still waiting for new a token from the source.
6. Figure 2.10-f: at this point all pipeline stages run out of tokens. We conclude that if the source is slow, we entered *Token Limited Mode* because we require a token to continue the flow of pipeline.

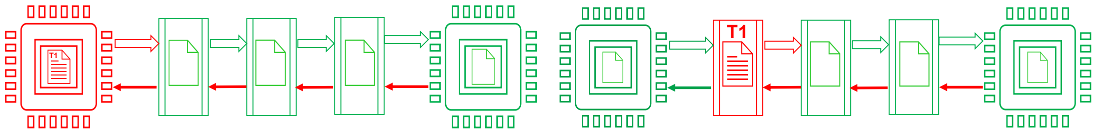
In Figure 2.9 we investigate the *Bubble Limited Mode*, when we need a bubble to proceed further: we conclude that the shortage of bubble stops the pipeline flow at a certain stage. From this point PLF >1 presents the *Bubble Limited Mode* as we have more tokens than bubbles. In contrast, Figure 2.10 concludes that the pipeline flow gets blocked due to the shortage of tokens called *Token Limited Mode*. In the remaining discussion *Token Limited Mode* may represent a PLF <1, as we are out of tokens.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.



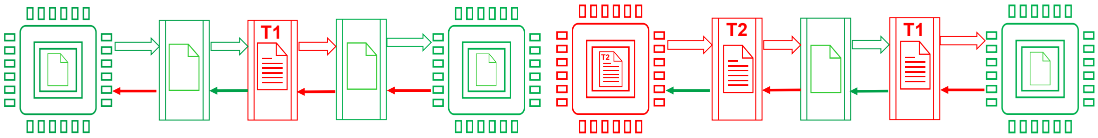
(a) WCHB based 3-stage Pipeline configuration with QDI Source and Sink

(b) Reset State



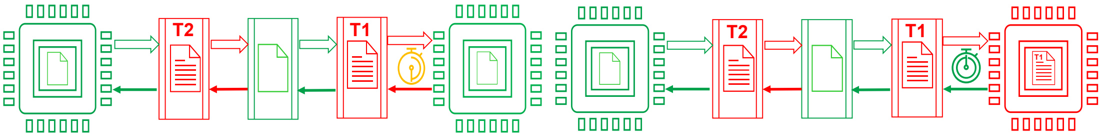
(c) 1^{st} Token from Source

(d) 1^{st} Token latched by 1^{st} Buffer



(e) 2^{nd} Token Requested

(f) 2^{nd} Token from Source



(g) Circuit Experience Delay from Sink in consumtion of 1^{st} Token

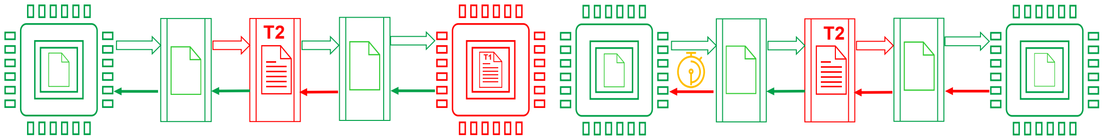
(h) After an arbitrary time, Sink latched the provided Token and responed with Ack Signal

Figure 2.9: Token Movement within QDI Pipeline while experience Sink Delay called Bubble limited mode

2.3 Circuit Description and Fault Injection simulation environment

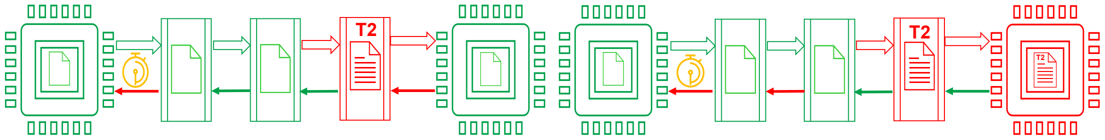
For the generation of QDI circuits we use the Python-based asynchronous circuit description framework by [Hue22]. This tool provides us with a high-level interface to design an asynchronous circuit. We defined our circuits in python description file, and the tool generates the circuit description in Verilog and VHDL format which is then simulated with standard digital simulators. There are multiple steps involved in translation from python description to Hardware Discription Language (HDL) format that is not part of this work, but the interested readers may find it in Chapter 5 of [Hue22]. The HDL file is then passed to the environment [BHNS21] for the fault injection experiments. Figure 2.11 shows our simulation setup. It consists of a QDI source and sink generating and acknowledging DR data with programmable delays to mimic real-world DI scenarios. With these programmable source/sink delays we control the PLF. Monitors are placed

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub



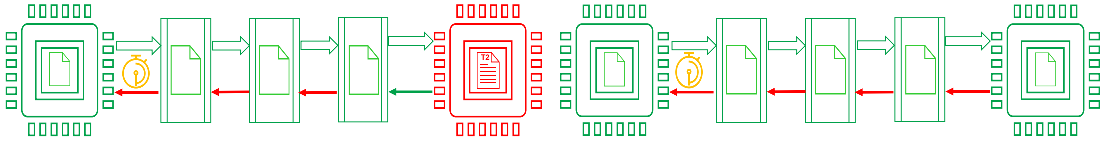
(a) 1^{st} Token is consumed by Sink

(b) Request a 3^{rd} Token



(c) Source is not responding

(d) 1^{st} Two Pipeline stages goes out of Token



(e) 2^{nd} Token is consumed by Sink

(f) All Pipeline Stages goes out of Tokens

Figure 2.10: Token Movements Continue from Figure 2.9, while experience Source Delay called Token limited mode

at the interfaces of the target circuit that check each activity and compare it to a golden run. Please note that only value deviations are considered as error, while timing issues are only considered an observation because circuits are QDI. As our concern are SETs, which occur rarely, we only inject one fault per simulation and observe the behavior. We are excluding input and output signals from the injection list because these are directly observable to the monitors with no chance for mitigation or masking, so these will simply result in higher fault rates. The injection count for the respective circuit variant is chosen directly proportional to the number of nodes in the circuit and the average processing time of the circuit (this is a pre-calculated time for a specified number of tokens successfully passed during the golden run). However, we set a minimum threshold of 10K for the injection count to obtain meaningful statistics in all cases. Injection time and location are randomly chosen, while the injection pulse length is variable, we also add a value higher than the longest inertial delay among all gate delays in our considered buffer templates, and hence electrical masking cannot occur which purely gives insights of logical and temporal masking. Where the injection pulse lower than the shortest inertial delay gives us the estimate of electrical masking.

2.3.1 Gate Delays

For the gate delays we utilized the timings from NanGate_15nm library file with typical conditions [si214]. As an MCE is not part of this library, we considered the MCE model from [SGY^{+}09] in which they propose a combination of simple NAND gates. Gate delays

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

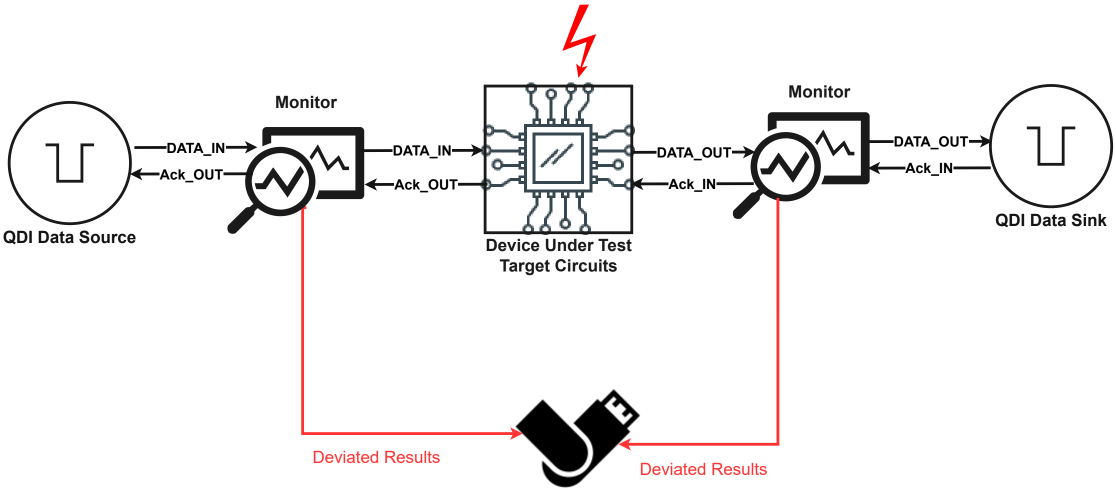


Figure 2.11: Fault Injection Simulation Environment

are computed from a timing matrix of the respective gate using an interpolation method with fixed index_1 (input_net_transition), as in our simulations we are not varying this parameter, while we vary index_2 (total_output_net_capacitance) depending on the fanout of the respective gate.

2.3.2 Fault effects

Deviations from the golden run are classified into two main categories, namely data errors and timing issues. For timing issues, i.e., data arriving earlier or later than expected, data integrity remains safe, because of the circuit's delay insensitivity. Data errors are further classified into four categories

(a) Value error: data is received correctly, but the value is not as expected.

(b) Coding error: both rails of the DR bit go high; this is illegal in our considered DR encoding.

(c) Glitch: during any handshake phase a signal makes more than one transition, or causality of signals is otherwise violated by a wrong sequence, like *acknowledgment* activated before data completion

(d) Deadlock: the circuit stops in a state where no further transition is possible.

2.3.3 SETs Pulse Width

[Kob20] provides a detailed survey on SETs and SEUs in digital circuits. Since we are utilizing delays from a 15nm library, our focus range is from 9nm to 22nm. According to their analysis, the duration of SET for this range is approximately 23ps to 107ps. This

TU WIEN Bibliothek Your knowledge hub

clearly indicates the range of *realistic* pulse widths. Given that our abstraction level is gate level, and we are specifically concerned with logical and temporal masking effects, we ensure that electrical masking cannot occur by selecting an injection pulse length greater than the longest inertial delay among all gate delays in our target circuit. That is why for each target circuit we performed a detailed analysis of the delays of the contained gates and related our injection pulse widths to that. This can be considered the range of *effective* pulse widths. As can be seen in the histograms, it is in the order of some tens of picoseconds. So when choosing different effective pulse widths ranging from the shortest gate delay (approx 10ps) to the longest one (approx 150ps), we automatically also covered the realistic range from [Kob20].

2.3.4 Field Programmable Gate Array (FPGA) Emulation and Real Experiments with Application Specific Integrated Circuit (ASIC)

Our current research primarily involves simulations, providing valuable insights and a controlled environment for initial analysis, which enables us to refine and optimize the proposed approaches. As our circuit description is in VHDL file format, FPGA emulation could be performed as an extension of this work in a straightforward manner. Fault injection experiments with FPGA (using “saboteur gates” to modify signal levels in a targeted way) expedite the process, but this approach does not yield more realistic results than our simulation (while, however, limiting options for controlling and observing signals). To bridge the gap between simulation results and practical deployment, conducting experiments on ASIC is crucial. ASICs offer a level of realism and fidelity beyond what simulations and FPGA emulations can provide. Still, the choice of radiation environment (particle type, energy, distribution, rate etc) is decisive for the outcome. We acknowledge the importance of validating the proposed techniques on an ASIC platform to ensure real-world applicability, a task that may be pursued in postdoctoral research.

2.4 Target Circuits

During the research journey we evaluated our proposed fault-tolerance approaches with different QDI target circuits, but in this thesis our main aim is to convey the knowledge we earned during the investigations and to highlight the contribution in a clear manner. So, we decided to consider an empty pipeline circuit as presented in Figure 2.12 throughout Chapter 3, as this chapter investigates the buffer templates and their resilience against SETs. To this end, this configuration gives us clear insights of buffer templates’ susceptibility. Due to its simplicity backtracking of effects to the root cause promises to be simpler, and potential masking of effects by combinational function blocks can be ruled out in the empty pipeline. Another advantage is that changing from one template to another already changes the whole circuit.

For Chapter 4 our target circuit will be a 16-bit 7 stage pipelined multiplier as presented in Figure 2.13 (to fit the block diagram we cut the circuit into two parts at *Buffer 4*). The

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

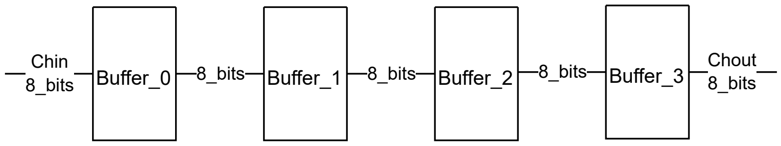


Figure 2.12: Empty Pipeline Circuit Configuration

multiplication is performed using a simple binary multiplication method. This pipelined multiplier is supposed to give insights about how pipelines with relatively complex computational units in between react to SETs, because these units also contribute to error generation.

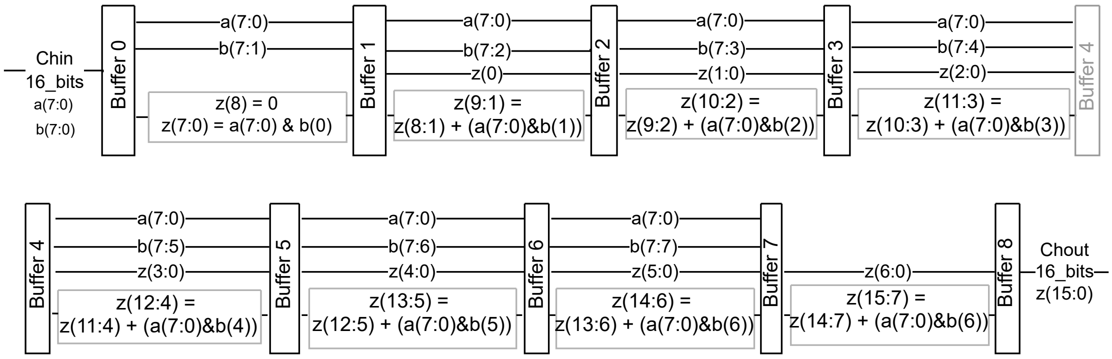


Figure 2.13: 16-bit 7 Stage Pipelined Multiplier

For Chapter 5 our focus will be conditional control elements, so we select a 16-bit iterative multiplier as our target circuit, as presented in Figure 2.14. In the iterative configuration, the shift and add operation are performed on operands until the multiplication is completed. The main purpose of this configuration is to analyze the susceptibility of the conditional control elements *Multiplexer* (Mux) and *De-Multiplexer* (Demux) towards the SET.

2.4.1 Target Circuit for the Future Work

Following the improvement of key components in the circuit, such as buffers, combinational logic, and conditional control elements, we are confident that when integrating our proposed enhancements into larger circuits, like a microprocessor, it will demonstrate the same resilience as observed in simple pipelined and iterative multipliers. This is substantiated by the fact that a microprocessor comprises buffers, combinational logic, and conditional control elements, and the enhanced versions proposed are robust against SETs when evaluated within the context of straightforward architectures. In the case of larger circuits, the primary constraint lies in simulation time. For instance, when simulating a simple 16-bit multiplier, the observed simulation time is approximately 2 hours. This simulation is conducted on a network of 10 physical machines, each equipped

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

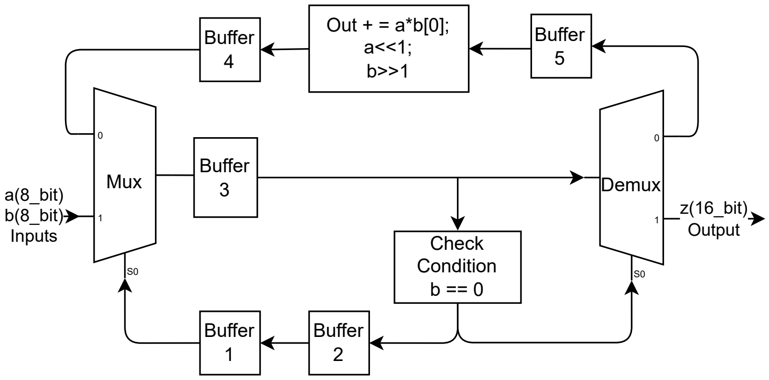


Figure 2.14: Iterative Multiplier

with a 3.5 GHz 7th generation Intel i5 processor and 16 GB RAM. Additionally, each machine operates four workers in parallel, with one worker assigned per core.

2.5 Fault-tolerance techniques from literature

The dependability of the circuits is most often assured by redundancy, most of the existing fault-tolerance techniques leverage hardware redundancy [KK20a]. For QDI templates, [JM05] also introduces double-check of a double-up approach. And this approach shows approximately full resilience towards the SET's as [Hue22] evaluates in a recent study. But the area overhead easily exceeds 3 to 4 times that of to the base circuit with high processing time, compared by [Beh21]. [JM07] also highlights another drawback of modular redundancy: liveness of the QDI circuits may be compromised by the majority voter. A transient may cause a deadlock or change the phase of the current cycle. But with some modification's redundancy can be used: [MRL05a] presents a hardening technique like dual modular redundancy (DMR), duplicating the combinational logic, with a single storage element, but each MCE has an extra input that is connected with the redundant combinational part to tolerate single event transients (SETs). They also introduce a rail synchronizing technique, in which each bit is hardened by synchronizing it with a parallel bit. At last, they use a 1-bit controller that forms an extra check for buffer.

By leveraging redundant topology [MR07], [KMM15], and [KHS$^+$20] proved the resilience of real-word asynchronous circuits like processors and controllers. Few other works like [VM02], and [KK20b] present methods that are inspired from synchronous hardening methods.

[LM04] in their investigation of QDI circuits discuss different types of faults and their out-turns that lead them to detect and correct faulty transitions. Their focus is fail stop operation, converting failures to deadlocks. Continuing in [PM05] without duplicating the whole circuit for the detection of fault for fail stop operation they introduce an extra

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

check rail as validator. That extra rail also provides a second acknowledgement signal. In addition to that there is also a special I-element for the detection of the illegal word. This invention gives efficient failure detection both for hard and soft errors. [MRL04] also analyze and highlights the fault-sensitive parts of asynchronous circuits, in their study MCE's sensitivity regions are explicitly mentioned that are addressed in later research [MRL05b]. Where [GYB07] introduces a hardening technique with time redundancy, the buffer recovers itself after the transient fault vanished out. The first latch accepts data and performs a validity check, and if the check fails the second latch does not respond while the first one is also reset to accept valid data after the transient passed. In [BS09] there are more than 9 fault tolerant techniques, at later stages we consider some of these in our analysis. In most techniques they try to minimize the storage mode of the MCE.

It is difficult to conclude the best approach from all these, but the approach that appeals most is to shorten the storage mode of the MCE's during varying PLF. As QDI circuits are already using more than one rail for each bit, full redundancy becomes expensive to realize. After considering most of the discussed studies in their evaluation [HNS20] presents two very elegant fault-tolerant buffer designs with minor overheads, named Interlocking and deadlocking. Those two proposals are evaluated in [BHN$^+$21] with state-of-the-art approaches in a comprehensive manner. The Interlocking WCHB seems a strong candidate where the Deadlocking WCHB converts other error types to deadlocks as it is designed for fail safe applications. So, we decided to start with their findings. Special approaches dealing with the flushing of combinational logic will be discussed in Chapter 4 in detail.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

CHAPTER 3

Fault Tolerance Behavior of QDI Buffer Templates

The main theme of this chapter is to investigate the susceptibility of the QDI buffer templates towards the SET's. Throughout this chapter we only consider the *Empty Pipeline Circuit* as our target circuit. In this way we can remain focused on the inherent fault tolerance behavior of the buffer without any external effects from the other parts of circuit. We will extend the scope in the subsequent chapters. We start our analysis with the baseline QDI buffer template named WCHB and then its fault tolerant state of the art derivatives, we also discuss the SET hardened templates proposed during our research journey. This chapter also discusses each possible effect of SET on a buffer and then backtraces its root cause. At the end of the chapter, we will highlight the pros and cons of each template.

3.1 WCHB behavior under SET's

The WCHB and its basic functionality is already known from Section 2.1.7. Here we start investigating its fault-tolerance behavior specifically under SETs. The target circuit is an 8-bit, 4-stage *Empty Pipeline* from Section 2.4 with WCHB as buffer template. Figure 3.1 presents the gate delays and gate count of the target circuit. On the x-axis we have a numerical identifier for each gate of our target circuit that counts up to around 125, where the y-axis presents the propagation delay of respective gate. The maximum propagation delay of a gate contained in the WCHB based *Empty Pipeline Circuit* is 49ps, where the minimum is 23ps as presented. Please note that the colors change with the y-axis; they only help the reader to estimate which region is more dense.

Figure 3.2 presents the susceptibility results of our target circuit under the influence of SETs. It presents the number of times the circuit failed to mask the effects of the

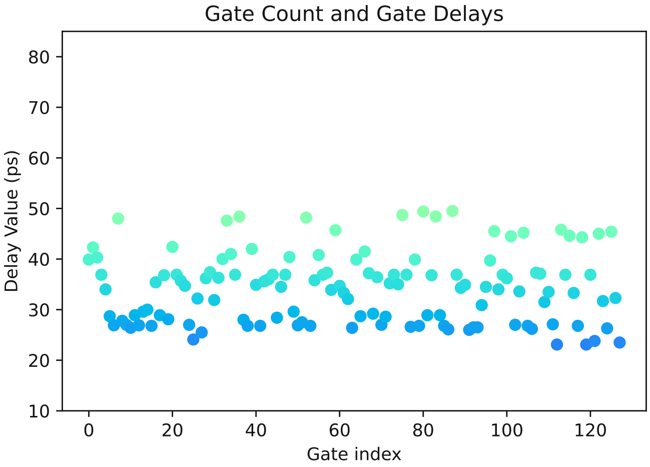


Figure 3.1: Gate Count (128) and Gate Delays of 8-bit, 4-stage Empty Pipeline Circuit with WCHB as Buffer Template

SET. As we are injecting one fault per simulation, the simulation count and the fault injections are same. With our selected target circuit for all PLF configurations the injection count is less than the set threshold 10K, so it is automatically set to 10K (the selection criteria are already discussed in Section 2.3). On the x-axis we vary the injection pulse length. As the minimum propagation delay a gate contained is 23ps, we start from $minimum - 10\%\_of\_minimum$ and end at $maximum + 100\%\_of\_maximum$, that is 99ps in this case. The different injection pulse lengths unfold the resilience behavior of the buffer template in a comprehensive manner. On the y-axis, each row corresponds to a different PLF, from $TokenLimitedMode$ to $BubbleLimitedMode$. This checks the circuits' fault tolerance with dynamic environmental effects.

The first noticeable thing from Figure 3.2 is the most prominent error types: for PLF$\leq$1 it is coding errors, while in extreme $BubbleLimitedMode$, e.g. PLF=4 or 10, the number of deadlocks proportionally increases. Before moving to the main root cause of each error we observed in Figure 3.2, here we first discuss the sensitive windows of the WCHB. We divide our analysis into three sections $TokenLimitedMode$, normal operation and $BubbleLimitedMode$ as presented in Figure 3.3-A, B and C respectively.

In the following discussion we highlight the sensitive windows of the WCHB with respect to SETs. The reference circuit for the waveform is presented in Figure 2.8-A, where the $Buffer\_0$ is selected this time to illustrate the behavior.

1. Figure 3.3-A: Indicated with *Phase 1* "en0" goes high, which means $Buffer\_0$ is armed for the data token. The source generates the data token after some arbitrary delay labeled with *Source Delay*. During the *Source Delay* window MCE-0,1,2 and 3 are in storage mode. As we already know from Section 2.1.3 the MCE is

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

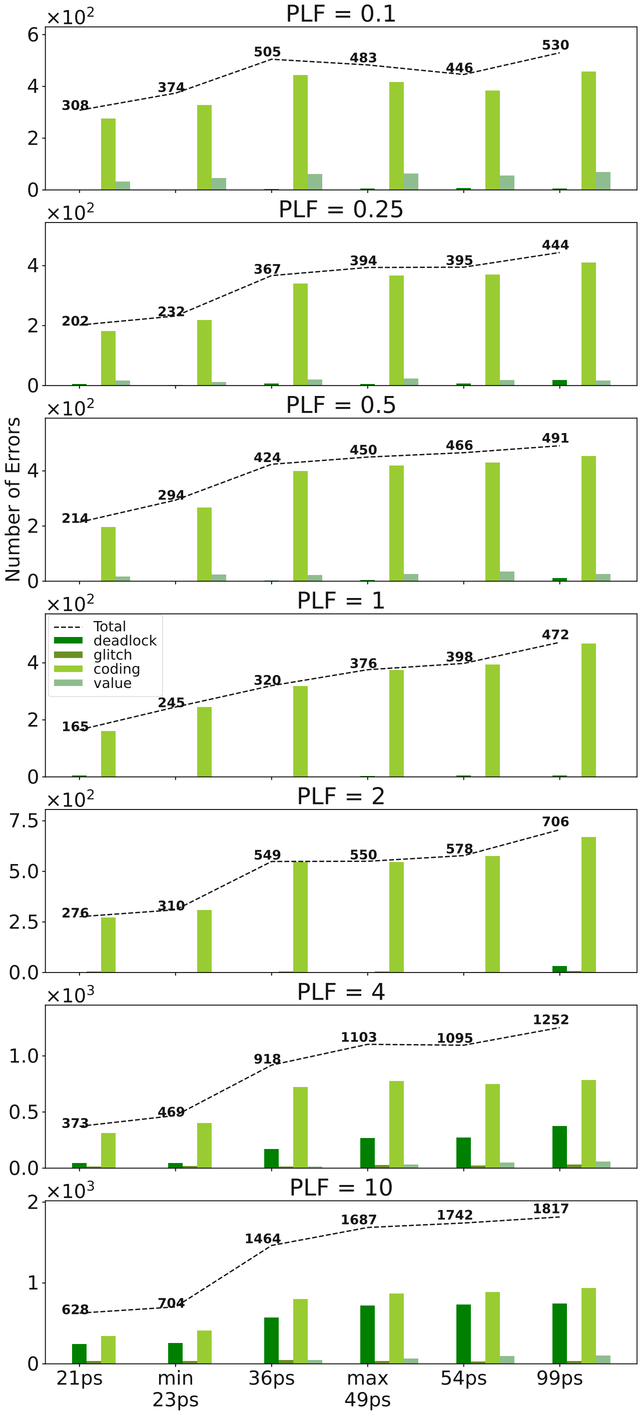


Figure 3.2: Susceptibility analyses of 8-bit, 4-stage Empty Pipeline Circuit with WCHB under the influence of SETs with variable PLF and different injection pulse lengths

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

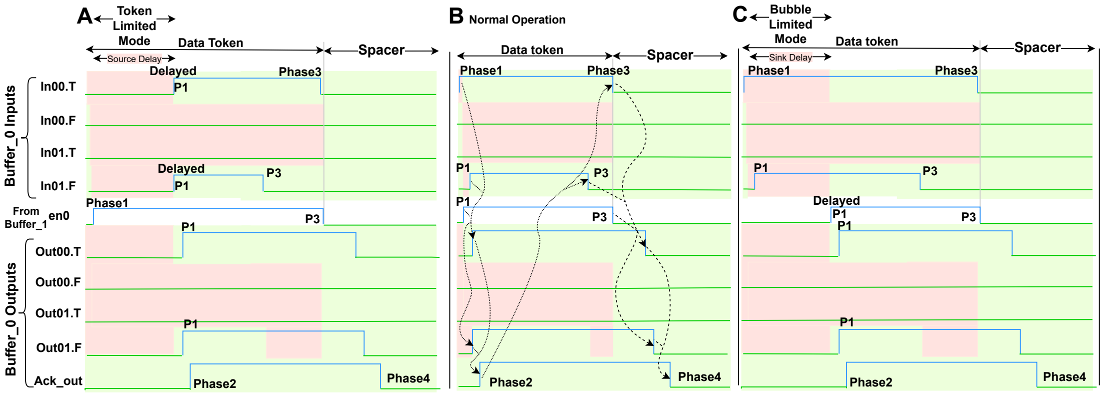


Figure 3.3: Sensitive Windows of WCHB

sensitive to SETs in storage mode. As shown, during this window these MCEs are susceptible to SETs from the input as well as from the output side.

2. Figure 3.3-B: In normal 4-phase dual rail operation, only one MCE of each bit shows a transition, which means the other remains in storage mode, and is susceptible to SETs from both input and output sides. The reason is the activation through the enable signal. In the presented scenario, these are MCE-2 and 3, as their input remains low where “en0” goes high to complete the handshake cycle. Moreover, as all bits are independent from the others in their transition, their timing during the respective handshake cycle also affects their susceptibility. This time *In01.F* shows a transition a bit earlier than the *In00.T* for *Phase 3*, as a result MCE-3 goes into storage mode until “en0” goes low.

3. Figure 3.3-C: This time *en0* arrived some arbitrary time after the data token, as indicated with *Sink Delay*. All MCEs of the respective buffer go into storage mode until *en0* goes high.

3.1.1 Main Causes of Errors in Token Limited Mode

The prominent error types during Token Limited Mode from PLF=0.1 to PLF=0.5 are coding and value errors, as reported in Figure 3.2. We trace back the main causes and discuss them here with the help of Figure 3.4.

1. Figure 3.4-A: An SET appears at *In00.F* while the source is silent. As “en0” is already there the MCE fires, but this MCE is supposed to remain *low* during this handshake cycle. After some arbitrary *Source Delay* the real data token arrives at *In00.T* and the respective MCE also fires, and as a result the coding error is generated. This error propagates to the output because the WCHB is not able to mitigate it.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

2. Figure 3.4-B: This time the SET shows up after the real data token is latched but before it goes retired (acknowledgment from the next stage). In this case the SET is also considered as real transition and latched. This is because the WCHB considers every transition valid during the respective latching window. Here again the coding error is generated and propagated.

3. Figure 3.4-C: During Token Limited Mode, the source may generate data tokens at different time stamps; this time, the expected token at *In00.T* arrives a bit late as compared to the other bits. During this time window as shown the SET appears and is considered as valid transition. As this bit is the final, awaited bit, the SET advances the handshake cycle before the correct transition arrives at *In00.T*. In this case the correct transition is missed because “en0” goes low before it occurs. So, we propagate the wrong value to the next stage and flag the value error.

4. We only consider the scenarios when an SET appears at the inputs of the MCE, but if it appears at the output node during the sensitive window, it will have the same impact, as we already discussed in Section 2.1.3.

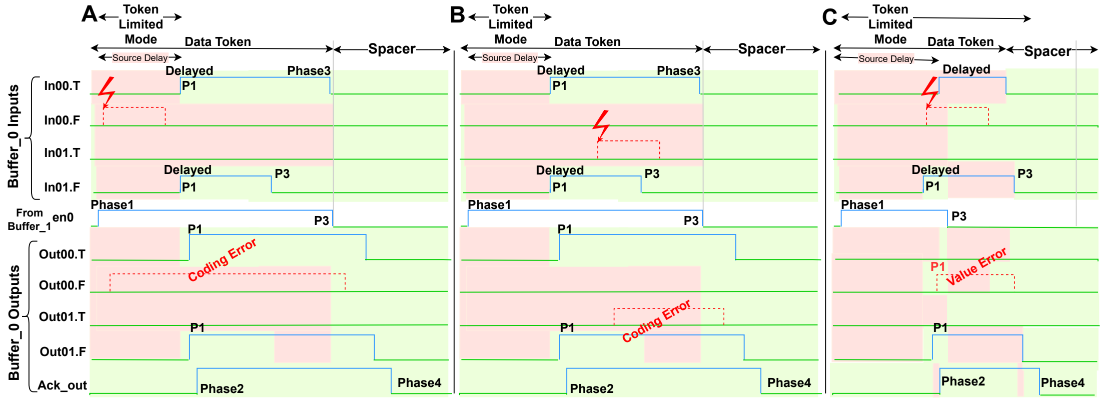


Figure 3.4: Main Causes of Errors During Token Limited Mode

3.1.2 Main Causes of Errors during Normal Operation

The prominent error type for PLF=1 is coding error as Figure 3.2 suggests. We will discuss the main causes behind these here.

1. Figure 3.5-A: As we know for the valid data token only one rail makes a transition, where the other MCE remains in storage mode as indicated with red shade. During this time window if the SET appears at any of the non-transitioned rails it may be captured and propagated, as all MCEs are armed during the valid data phase. In the shown case it appears at *In01.T* and generates a coding error, as the correct data token is already there.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

2. Figure 3.5-B: If the SET appears at the output of the non-transitioned MCE where the length of the pulse is long enough to flip the storage loop, the logic value will be changed, and a coding error be generated as a result.

3. Figure 3.5-C: In this case an SET appears really close to the correct transition that is expected at *In01.F*, and as a result a coding error is generated and propagated.

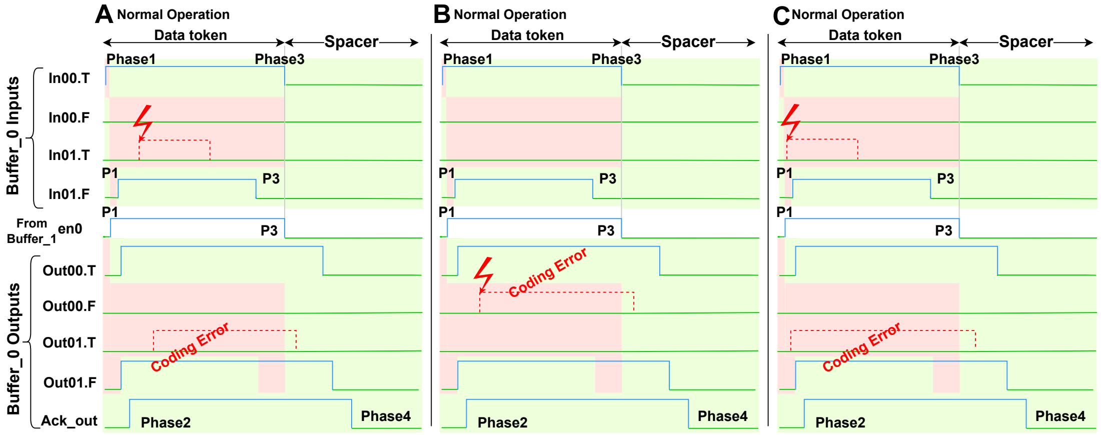


Figure 3.5: Main Causes of Errors during Normal Operation

3.1.3 Main Causes of Errors During Bubble Limited Mode

As Figure 3.2 suggests, during Bubble Limited Mode (PLF = 2 to PLF=10) the prominent error types include coding error and deadlock.

1. Figure 3.6-A: As we know during bubble limited mode the sink is slow in response; as shown “en0” is delayed for the acknowledgment of the data token. This time the SET appears at the input *In01.T* but after the valid data token is removed from the opposite rail of the respective bit, the data token is still stored by MCE-C3, because, due to the sink delay, “en0” is still active. Consequently, the SET is latched by MCE-C2 and generates the coding error as shown, where *Out01.T and Out01.F* both go high.

2. Figure 3.6-B: The handshake cycle is completed between source and $Buffer\_0$ when *Ack_out* goes low indicated by *Phase4*. At this point the new data token is available at the inputs of $Buffer\_0$, but we are not able to latch it because we are still containing the last spacer for the next stage, as long as “en0” is still low (the last spacer is not yet acknowledged). This means the handshake cycle between $Buffer\_0$ and $Buffer\_1$ is not completed yet. During this waiting time window, the SET appears at the output *Out00.T* of MCE-C0. As the latter is in storage mode, the SET is converted into an SEU. As this faulty transition happens before

the last spacer is acknowledged, as a result it blocks the required transition to shift the handshake cycle, in this case we entered into a deadlock situation.

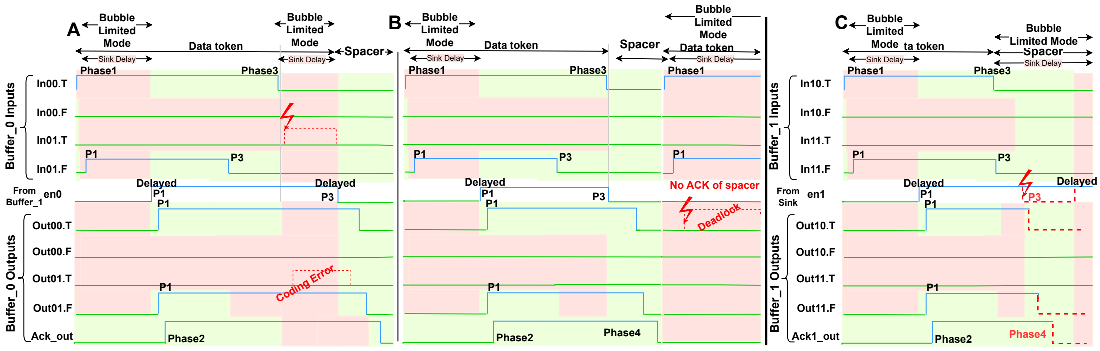


Figure 3.6: Main Causes of Errors During Bubble Limited Mode

3.1.4 Main causes of glitches during Bubble Limited Mode

We define a glitch as a short pulse, i.e., a sequence of two opposing transitions, which are both caused by some external influence rather than being part of the causal chain of events in the regular operation. In particular, in regular operation no signal makes more than one transition during a handshake phase, so any double-transition during a single handshake phase can safely be considered a glitch. Due to the causal behavior of QDI circuits if no transition is expected, a faulty one remains ineffective or at most causes a timing deviation. If a transition is expected, the glitch's leading transition will move the handshake process forward one step in one or the other way. However, in a QDI circuit, a next transition on the same signal will then only be expected once the reaction to the previous one has sufficiently propagated. Therefore, due to the short duration of the glitch (by definition), its trailing transition cannot arrive at an armed MCE and must hence remain ineffective. In conclusion we can state that in a QDI circuit glitches cannot propagate as such. So, the question arises why we observed a glitch in our results. In the following we cover the main reasons behind and how to address these.

We have already seen that during Bubble Limited Mode the leading glitch transition from the successor stage does not arrive at a point in time when a regular transition is expected (MCE is armed). Figure 3.2 shows that for PLF 0.1, to 1 all variants show 0% glitches, so that is why we can eliminate these from our analysis and only consider PLF > 1 in this section.

Backtracking the glitches, we had observed for bubble limited mode in our experimental results gave an amazing insight: faults on only one signal contribute to all glitches we observed.

Having identified the culprit through filtering of the results, we can now use Figure 3.6-C to explain how an SET on "en1" (*enable* signal of the last stage) can generate a glitch at

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

the output. With our choice of PLF > 1 the circuit is running in bubble limited mode as illustrated in Figure 3.6-C. In this mode the last stage spends a lot of time waiting for "en1" (inverted version of *Ack_In*). It is worth noting that for this section we changed our target buffer used in the explanation; for the following discussion our focus is on *Buffer_1* instead of *Buffer_0* in the last sections.

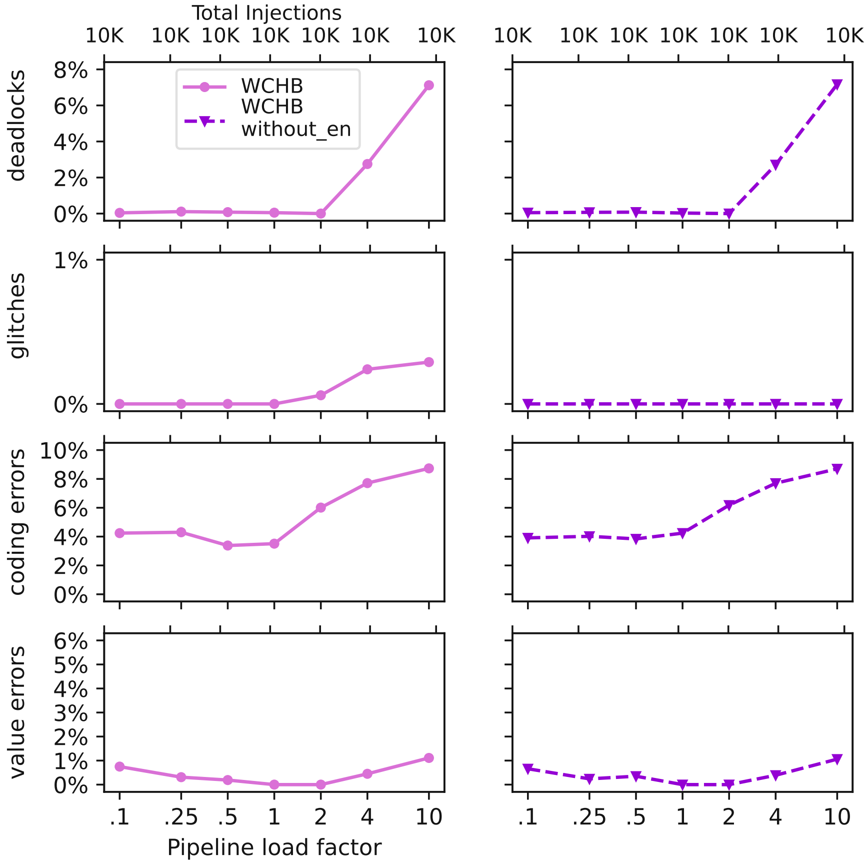


Figure 3.7: Error Rate of WCHB based 8-bit, 4-stage Empty Pipeline Circuit with and without last Stage Enable Signal in Injection List, with 60ps injection pulse

1. On the arrival of "en1" at P1 (more delayed than usual) a data token advances to the circuit output, and as a result, *Ack*1*_out* is activated (Phase 2) as a confirmation to the predecessor stage.

2. This acknowledgement ripples upstream to the *Buffer_0*, and as a result the latter issues a spacer indicated with Phase 3.

3. Now we are waiting for "en1" to go down on the sink side to complete the four-phase handshake.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

4. Accidentally, during this waiting window (highlighted with Sink delay) if a negative fault hits the “en1” signal as shown, it just produces a spacer at the output before the sink responded to the data token.

5. Recall our definition of fault effects, according to which the case where any rail changes more than once during same handshake phase it considered a glitch. So, the main cause of glitches is when a new transition arrives during an ongoing handshake phase.

6. A close look at Figure 2.8 reveals that “en1” is simply the inversion of Ack_In. While the latter, being an input signal, is excluded from fault injection, the former is not. As a result, we have an “internal” signal that is fully under control of an external input (more precisely, our testbench), so obviously it can have any arbitrary behavior without any chance for mitigating that through circuit provisions.

Figure 3.7 presents the resilience of our target circuit in percentage form, where each row is for a different error type. The bottom x-axis is again for the pipeline load factor, where on top we have the corresponding number of injections. In Figure 3.7 we ran fault injection experiments with (left column) and without hitting the last stage “enable” signal (right column). As clearly visible in first column second row, we initially observe glitches where after removing the last stage “enable” signal from the injection list, we can indeed confirm 0% glitches for the Bubble Limited Mode.

3.2 InterlockingWCHB under the influence of SETs

[HNS20] after considering most state of the art SET mitigating techniques in their evaluation study present two very elegant fault-tolerant buffer designs with respect to area overhead named Interlocking buffer and Deadlocking buffer. In our study we are interested in the Interlocking methodology as we want to keep the circuit alive where the deadlocking method simply locks the circuit for further communication on the occurrence of a fault. Figure 3.8 presents our 2-bit, 2-stage QDI pipelined circuit realized with the Interlocking technique, with changes to the original template highlighted in orange color. With the Interlocking methodology, only the first arriving input transition is passed to the output, and that successful output transition blocks any further input transition on the second rail MCE.

They use asymmetric MCEs with positive input, which means that high transitions are passed from the normal inputs to the output only when this extra input is logically high. In this case they connect the asymmetric input with the inverted opposite rail output as shown. So, the MCE only accepts a high transition from its inputs if the opposite rail MCE stores a logical zero. This configuration prevents coding errors to propagate, as from each rail pair only one MCE is allowed to fire. Figure 3.9 presents the sensitive windows of the Interlocking WCHB with respect to SETs. They are the same as for the plain WCHB, except for the non-transitioned MCEs’ sensitivity in the respective

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

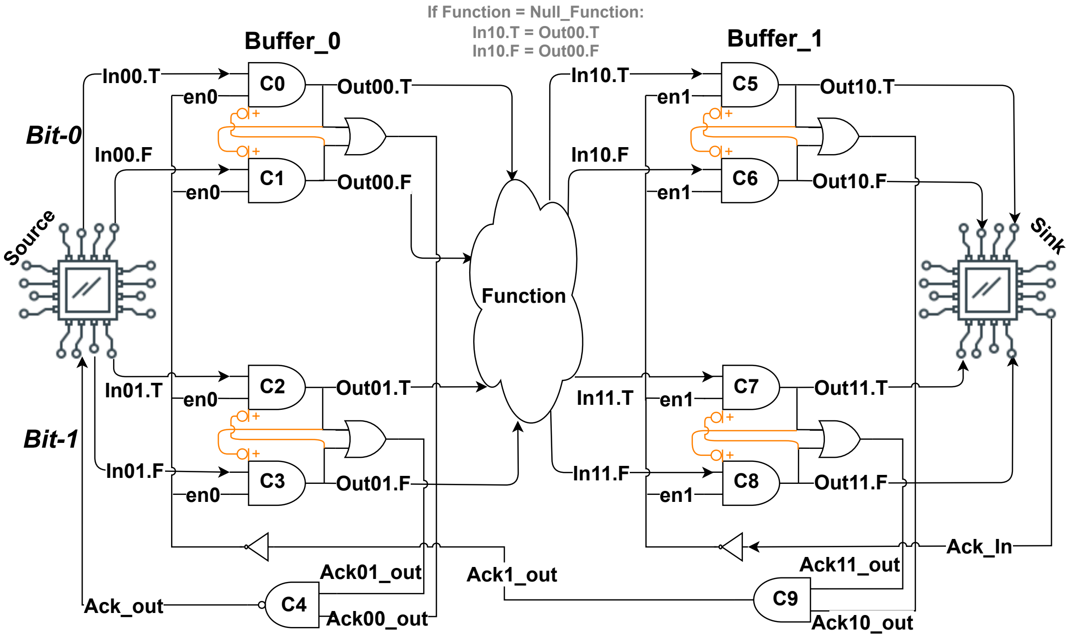


Figure 3.8: 2-bit, 2-stage QDI pipelined circuit realized with Interlocking WCHB buffer template

handshake cycle. The non-transitioned MCEs enter a resilience mode from the input side (*In00.F and In01.T* nodes) once the valid transition is passed and latched by the opposite rail MCE, as visible from comparison of Figure 3.9 and Figure 3.3.

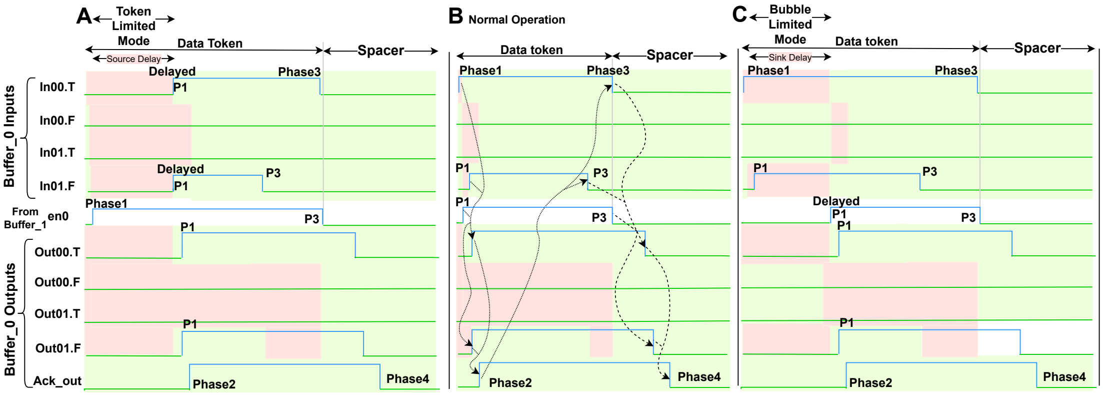


Figure 3.9: Sensitive windows of Interlocking WCHB

For the evaluation of the approach, we stick with our 8-bit, 4-stage Empty Pipeline target circuit while changing the buffer template to the Interlocking WCHB. To evaluate the resilience behavior of the Interlocking WCHB we first extract the gate delays of the target circuit as presented in Figure 3.10. The gate count is the same as for WCHB,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

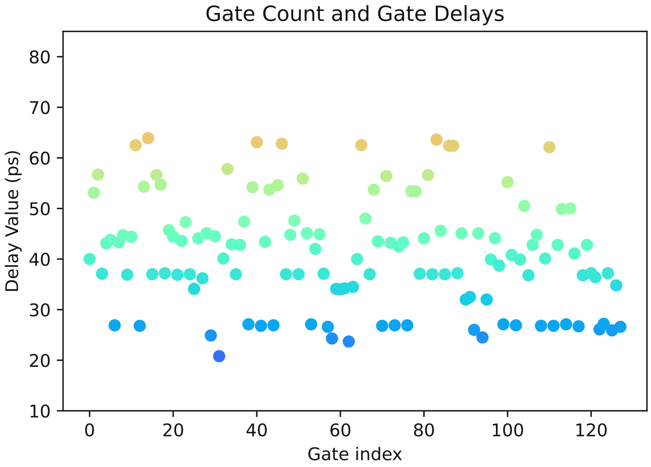


Figure 3.10: InterlockingWCHB gate count (128) and gate delays

because we only change the configuration of the main MCEs without adding any extra gates to the template. The maximum delay a gate contains is 64ps whereas the minimum propagation delay is 21ps. The area between 33ps to 46ps is a bit denser than the other regions as Figure 3.10 suggests, where very few gates exceed the limit of 50ps. In this case these are approximately 25 gates out of 128.

We performed fault-injection simulations with 6 different injection pulses as presented in Figure 3.11. The first thing we noticed from these results is the masking of short injection pulses like 21ps and 18ps. The latter is (minimum - 10% of minimum) and hence smaller than any gate delay, while, as visible in Figure 3.10, only one gate's propagation delay is 21ps either. Apparently, these short pulses are all electrically masked and do not affect the resilience of the target circuit. When we move further to a bit longer pulses like 42ps we start observing errors. While increasing the injection pulse beyond the maximum propagation delay of the gates in a circuit shows a saturation of the resulting error count.

Moving to the result analysis, we only observe value errors in Token Limited Mode. And after crossing the balanced mode PLF=1 deadlocks show up, and in our considered extreme Bubble Limited Mode the deadlocks are even more prominent. In the following sections we discuss the main causes of each error type observed in the respective mode of circuit operation. Please note that the last stage enable signal is deleted from the injection list as we consider it as an input signal.

3.2.1 Main Causes of Errors in Token Limited Mode

During Token Limited Mode the value errors are the only reported errors, as visible in Figure 3.11. In this mode we do not gain a lot in terms of resilience; compared to the baseline buffer template the resilience is approximately the same. Recall the result of WCHB from Figure 3.2, where the prominent errors are coding errors. Due to the fact

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

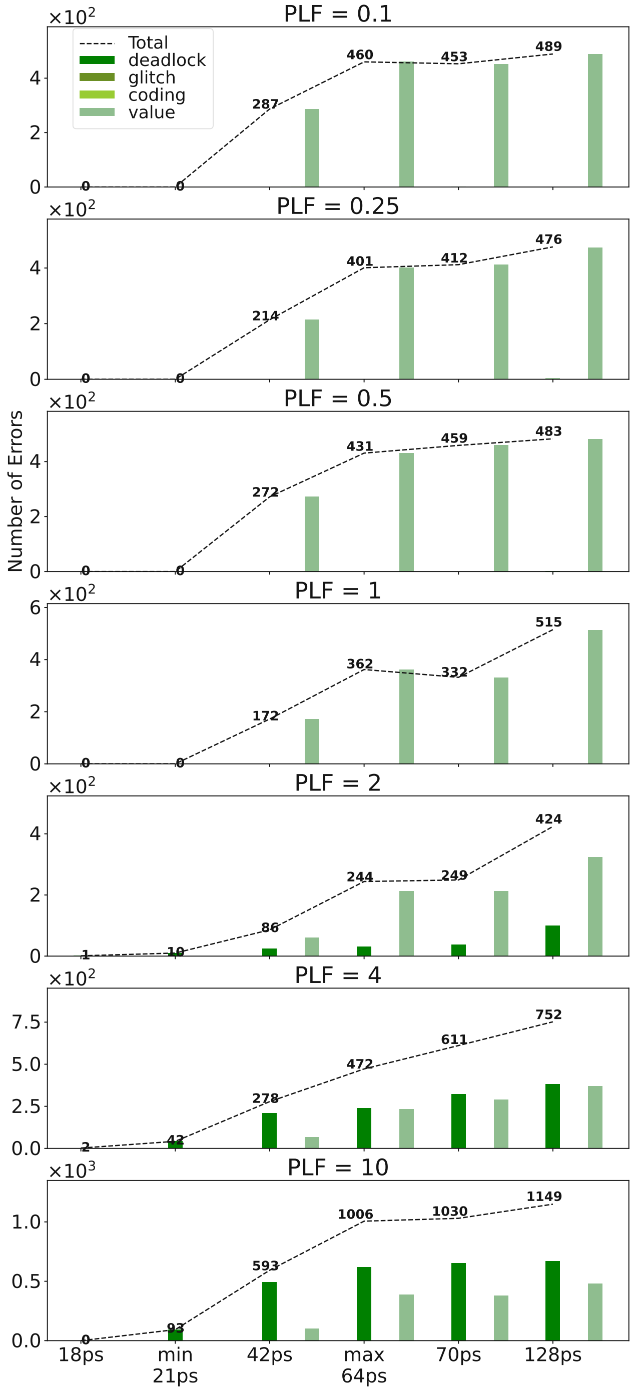


Figure 3.11: Suseptibility analyses of InterlockingWCHB under the influence of SET with variable PLF and different injection pulse lengths

that the Interlocking WCHB buffer does not propagate a coding error from input to output, some of them are corrected because the fault appear after the valid transition is mitigated by the interlocking, while others are converted into value errors by the last stage buffer.

While our simulation results, as presented in the bar graphs, comprise a large number of samples and hence have good coverage, the following discussion, for the sake of brevity, illustrates a selection of cases only, just to give an impression of important effects.

1. Figure 3.12-A: compared to the WCHB Token Limited Mode window in Figure 3.4-B here once rail *In01.F* transitioned and reached *Out01.F* indicated with “P1”, it locks the opposite rail MCE from latching any erroneous input transitions, and so the remaining waiting window of *In01.T* goes green. This is in contrast to Figure 3.4-B where the fault appears during this waiting window at *In01.T* and generates a coding error at the output. Still, a fault creating a transition *before* *Out01.F* transitioned will be effective and create a value error, as shown in the figure.

2. Figure 3.12-A: as “en0” is already active, the fault at *In00.F* is latched and appears at *Out00.F*. As a result the valid transition that arrives later is blocked, and a value error is generated and propagated as shown.

3. Figure 3.12-B: the fault hits *Out00.F* marginally after the valid transition from *In00.T* was accepted by MCE C0 (Figure 3.8). After C0’s propagation delay the valid transition appears at *Out00.T*, but the faulty one just arrived already at *Out00.F* as well. Both these transitions are a bit late to disable the respective opposite rail to fire. As a result, a coding error is generated. But when these transitions reach the inputs of the last buffer stage, in this case MCEs C5 and C6, the leading transition will disable the lagging one, due to timing asymmetries. In this case C6 wins and fires before the valid transition gets latched. As a result, the value error flag is raised. Note that in case the valid transition had won the race, the fault would have been masked and a correct result generated.

4. Figure 3.12-C: the case is the same as presented in Figure 3.12-A, but this time as shown the fault at *Out01.T* blocks the respective opposite rail MCE from firing (correctly), and so instead of a coding error a value error propagates to the next stage.

3.2.2 Main Causes of Errors during Balanced Operation

As Figure 3.11 suggests, in balanced mode PLF=1 we only observe value errors. In the following discussion, with the help of Figure 3.13, we will closely look into the main reasons behind these errors.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

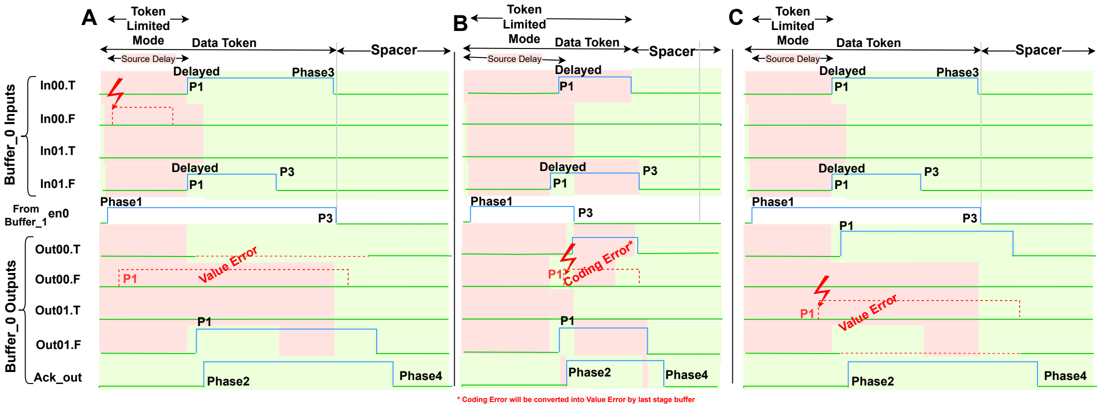


Figure 3.12: Sensitive Windows of Interlocking WCHB during Token Limited Mode

1. Figure 3.13-A: as we know the completion detection of each stage is composed of an OR and an MCE-based tree structure (Figure 3.8), so the final acknowledgment signal *Ack_out* is generated after the delay composed of the propagation delay of the respective gates. In the investigated scenario presented in Figure 3.13-A, the spacer acknowledgment to the source is generated at P4. If we pay attention to P1*, an acknowledgment from the next stage is received *before* P4. This makes the buffer armed for the next data token. During this susceptible window the fault at P1* is considered as valid transition and latched by MCE C2 Figure 3.8. After P4, the valid data token arrives but as MCE C2 already blocks MCE C3 for any transition, the faulty transition at *Out01.T* is considered as valid, and the result is a value error.

2. Figure 3.13-B: in the investigated scenario we observe the role of gate delays in the resilience of the circuit. As we use an interpolation method to decide the gate delays for a simulation run, in this case the propagation delay of C1 is less than the one of C0 for rising transitions. As presented, the fault hits *In00.F* after the valid transition is already at *In00.T* while *en0* is also there. Due to the lower propagation delay with respect to its complementary rail MCE C1 still fires first and locks C0 for any further transition. In that way the faulty transition at *Out00.F* generates a value error.

3. Figure 3.13-C: the investigated case is a bit different than the others in a way that it may only happen in the empty pipeline configuration. To better understand, we consider $Buffer\_1$'s inputs, in contrast to the last presented scenarios where we used $Buffer\_0$'s output for our demonstration purposes. $Ack1\_out$ is low which means our $Buffer\_0$ is armed for the data token ("en0" must low in this case), and $en1$ is high which means $Buffer\_1$ is also armed. The fault hits *In11.T* as shown, and as *In11.T* is directly connected to the asymmetric input of MCE C3, the latter then blocks the transition from appearing at *In01.F* marked with P1.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

In this way the fault that appeared at *In11.T* is converted into a value error and propagated to the next stages.

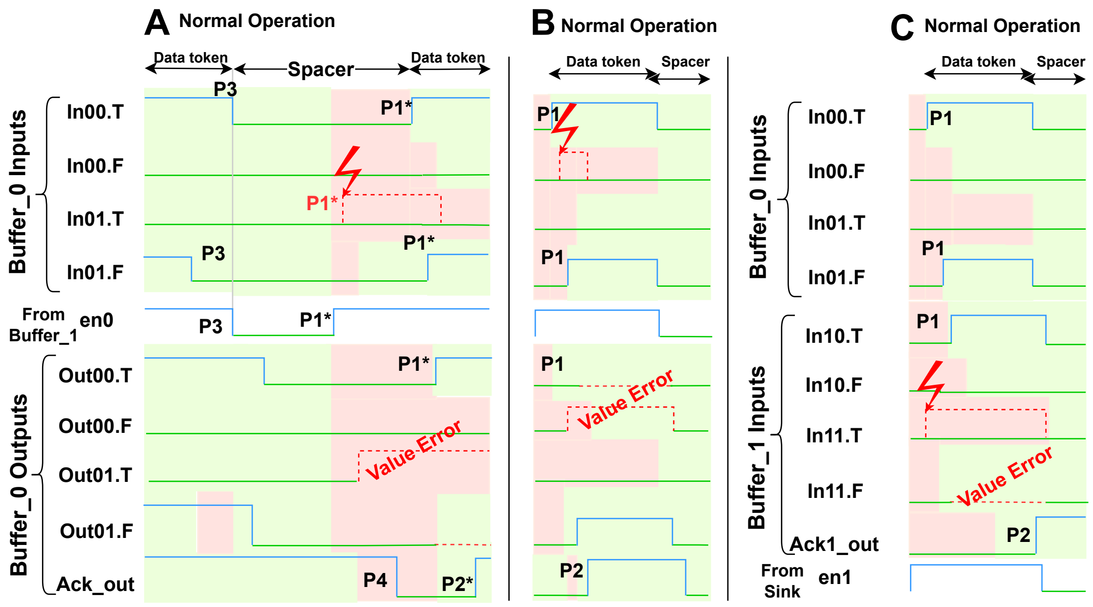


Figure 3.13: Sensitive Windows of Interlocking WCHB during balanced operation

3.2.3 Main Causes of Errors during Bubble Limited Mode

In extreme bubble limited mode we observe deadlocks and value errors, as Figure 3.11 suggests. In the following discussion we will try to figure out the main reasons why the buffer template shows these types of error and fails to mitigate the faults at the respective time instances – again by focusing on selected examples.

1. Figure 3.14-A: the data token is provided by $Buffer\_0$ but $Buffer\_1$ is still not armed for it, as “en1” is delayed. Still, the completion detection of $Buffer\_0$ generates the acknowledgement, and as a result the new spacer is provided by the source. At that point $Buffer\_0$ contains last copy of the data token and $Buffer\_1$ containing a last copy of spacer. Furthermore, as “en0” is high now, $Buffer\_0$ is in storage mode. Consequently, the fault appearing at *In10.T* during the time window labeled with *Sink Delay* (see Figure 3.14-A) is retained by MCE C0 as it is in a storage mode. This time the fault polarity is negative so the respective MCE changes its storage state to 0. As soon as“en1” finally goes high, the data token from “Bit-1” is latched but we do not have any data token from “Bit-0”, which keeps the circuit in a waiting state, corresponding to a deadlock.
2. Figure 3.14-B: This time we advance our handshake cycle to phase 3 with $Buffer\_0$, so $Buffer\_0$ requests a new data token from source. Now “en0” going low again

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

changes $Buffer\_0$'s MCEs into storage mode. To complete handshake phase 3, "en1" responds after some arbitrary time. If during the time window before that the fault appears at *In10.T*, it changes the stored value of the respective MCE. Once the value of MCE C0 is changed we are not able to revert it back, as our MCE is in a storage mode. The output of MCE C5 remains high, and we are not able to generate the acknowledgement of the spacer for $Buffer\_0$, as $Ack1\_out$ is toggling back to "0".

In Figure 3.11 we also observed a value errors, so here we also discuss the main causes behind the generation of these.

1. Figure 3.14-C: compared to Figure 3.14-A and B, here we consider again $Buffer\_0$ for the explanation of the fault scenario. While "en0" is delayed the MCE whose inputs change are sensitive to SETs from input and output side. As the MCE C1 is supposed to remain zero during this handshake cycle, it is not sensitive from the input side. In the shown setting, the fault appears at *In00.F* when it is actually green (according to theory), but on the arrival of "en0" the faulty transition is latched by the respective MCE and blocks the valid transition from *In00.T* to proceed to *Out00.T*. This happens because the propagation delay of MCE C1 is less than the one of C0. Recall that we also observed the same gate delay effects in the last section. Here a value error is generated and propagated to the sink. As a consequence, in the face of significant delay asymmetries, the green shading of intervals actually needs to be adjusted accordingly to account for such effects. However, in the following we will keep our focus on the main effects and hence stick to the slightly simplified shading technique.

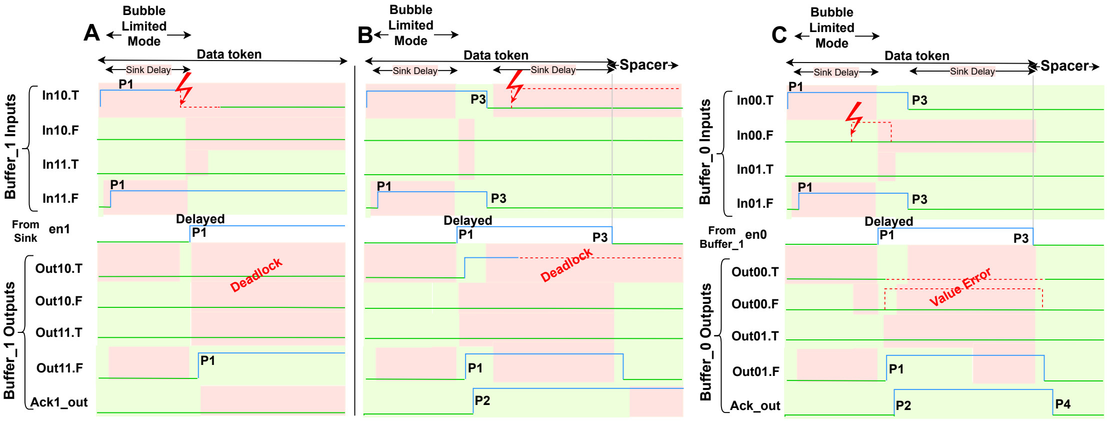


Figure 3.14: Sensitive Windows of Interlocking WCHB during Bubble Limited Mode

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

3.2.4 The resilience of WCHB and Interlocking WCHB

Figure 3.15 presents the error rate of both discussed buffer templates to make a fair comparison. Starting with the deadlocks, in terms of percentage, going from extreme token limited mode to the bubble limited mode we do not achieve noticeable difference in terms of the resilience. In the second row we present the glitches: As the main cause of glitches is identified, and after addressing this we do not experience any glitch as shown. Next is the coding errors: The WCHB, due to its unprotected nature passes any input transitions to the output without any further check, if they occur during a sensitive window. Where the Interlocking WCHB does the cross check before passing any transition, it blocks the second transition and prevents the generation of coding errors. But moving to the last row we realize that this check is not so effective, as the coding errors in WCHB are converted into value errors in Interlocking WCHB.

After the discussions in Section 3.2 and the comparison of this section, we conclude that the Interlocking WCHB provides us a good insight into the sensitive windows and how to minimize them to achieve a higher resilience. We use these insights in our further analysis to make circuits more resilient.

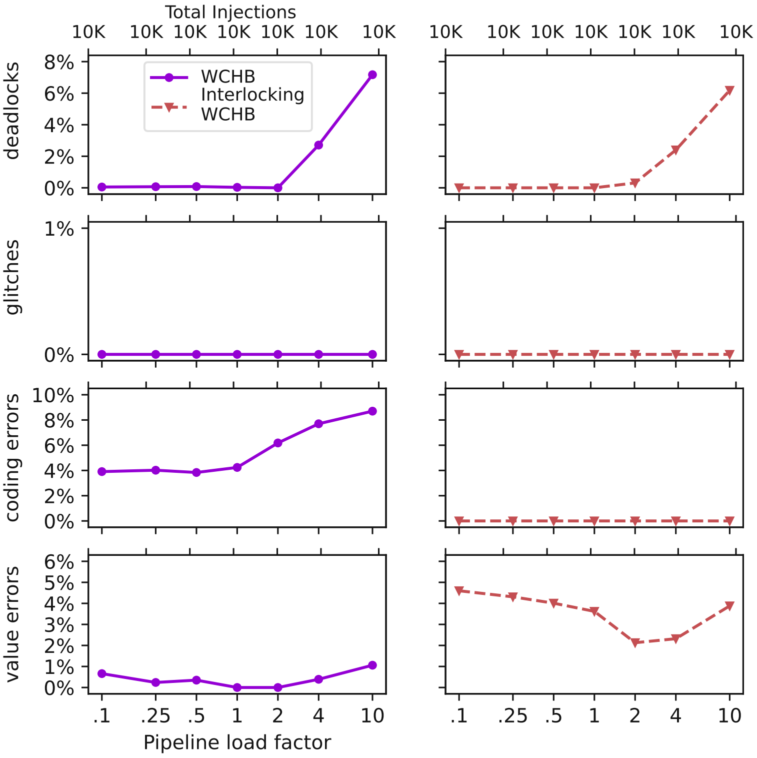


Figure 3.15: Suseptibility analyses of WCHB and Interlocking WCHB under the influence of SET (Injection pulse 60ps)

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

3.3 Input/Output-Interlocking WCHB for the mitigation of SET effects

To enhance the resilience of the buffer template, we propose an Input Interlocking in front of the buffer, in addition to the output Interlocking, drawing inspiration from [HNS20] and building upon the ideas presented in [TBNS21]. Figure 3.16 presents a 2-bit, 2-stage empty pipeline with a buffer template they call InOutInterlock WCHB. The InOutInterlock WCHB is purely designed to stop the propagation of SETs from the buffer input to the next stage. In InOutInterlock WCHB (Figure 3.16) the following modifications are applied:

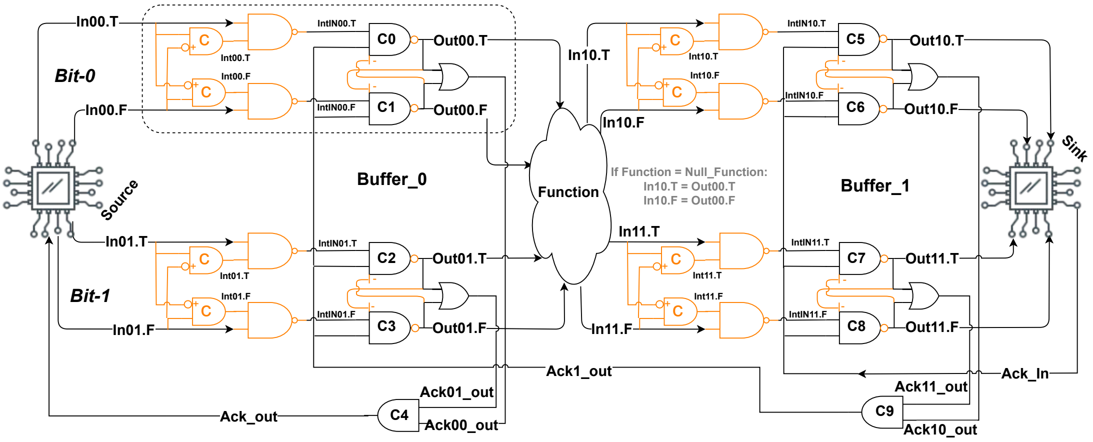


Figure 3.16: InOutInterlock WCHB

1. Asymmetric MCEs with one positive and one normal input are added for a first validity check. A rising transition on one rail on the normal input is only captured if the other rail of that bit is low.
2. NAND gates serving as small pulse filters are added. The input MCEs of the first check serving as delay elements, short positive pulses having a lower pulse width than the propagation delay of the MCEs are filtered from reaching the output MCE.
3. As the NAND gate inverts the logic, there is no need for the acknowledgement signal inversion. This saves some transistors and therefore contributes to compensating for the area overhead of the additional input filter.
4. Revision of the output MCE from two normal and one positive (asymmetric) input used in the Interlocking WCHB to two normal and one negative input, to further reduce area overhead.

With these modifications the first high transition on any input rail disables the input MCE of the other rail, similarly to the interlocking done at the output. The advantage is

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek
WIEN Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

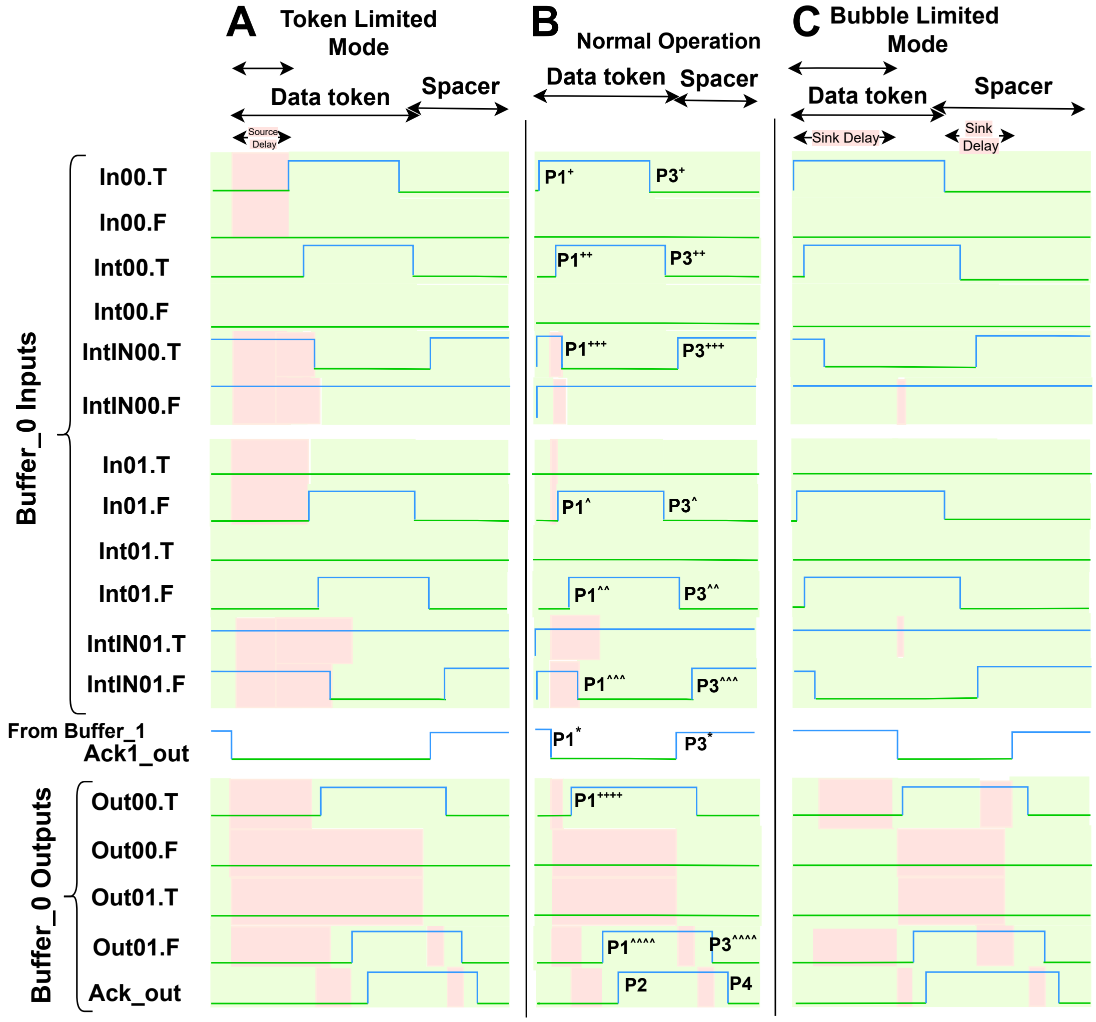


Figure 3.17: Sensitive Windows of InOutInterlock WCHB

that this interlocking is done without any delay and therefore supposed to more effectively filter illegal code words. This input filter passes the token to the NAND gate small pulse filter, which ensures that any pulse at the input shorter than the MCE propagation delay is blocked, unless the token is latched by the output MCE. The direct input of the NAND gate provides the possibility to flush a faulty transition latched by the input MCE such that short pulses on data rail have no effect other than a possible timing deviation, as they might temporarily block the correct transition to propagate. Output interlocking minimizes the probability of coding errors that may be generated due to a fault hitting a NAND gate output or a fault with a pulse width longer than the filtering threshold. The proposed modifications (InOutInterlock WCHB) are most effective in bubble-limited mode (high PLF), as only the first transition is passed to the output buffer and will then be latched when the acknowledge signal arrives.

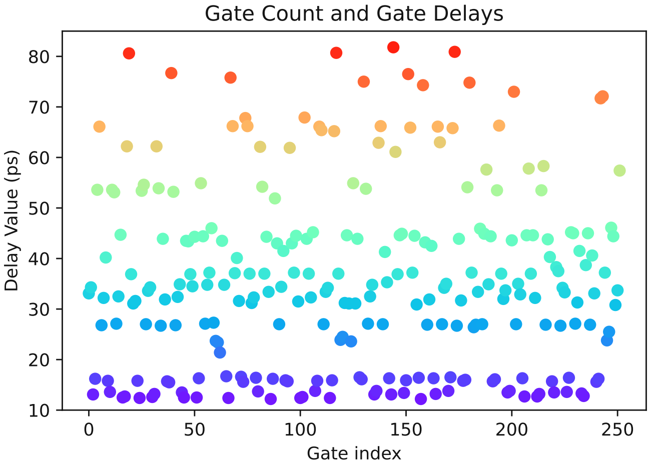


Figure 3.18: InOutInterlock WCHB Gate Count (252) and Gate Delays

Before the analysis of results, we first take a closer look at the sensitive windows of InOutInterlock WCHB. With the introduction of input interlocking extra nodes have been introduced which then contribute to the sensitivity of the circuit. As Figure 3.17 suggests we add all these nodes (*IntINXX.T and .F*) in our analysis figure. First, we compare the sensitivity of our buffer with Interlocking WCHB Figure 3.9 in each mode. For token limited mode we do not achieve anything, closer look reveals that we make our circuit even more susceptible to SETs because of the extra nodes. In balanced operation it again shows similar sensitivity, with the extra nodes problem. But moving to the bubble limited mode we analyze that the input side is clean as compared to its parent buffer template Interlocking WCHB. The reason is clearly that the buffer template is only designed with attained robustness in bubble limited mode in mind. We will discuss the main reasons in the respective section.

For the evaluation of the approach the target circuit remains the same, that is an 8-bit, 4-stage Empty Pipeline, while changing the buffer template to the InOutInterlock WCHB. To evaluate the resilience behavior of the InOutInterlock WCHB we first extract the gate delays of the target circuit as presented in Figure 3.18. The maximum delay a gate contains is 82ps whereas the one with the minimum propagation delay is 12ps.

We performed fault-injection simulations with 6 different injection pulses as presented in Figure 3.19. The first thing we noticed from these results is the masking of short injection pulses from 11ps to 35ps. From Figure 3.18, it is clear that the gate delays are distributed over the whole defined delay value space quite evenly, as compared to the Interlocking WCHB Figure 3.10, where the Figure 3.19 shows that the InOutInterlock WCHB only goes sensitive when we inject pulses equal to the maximum gate delay of the circuit. As already discussed, and with Figure 3.19 it is clear that we only improve resilience over the Interlocking WCHB in extreme bubble limited mode.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

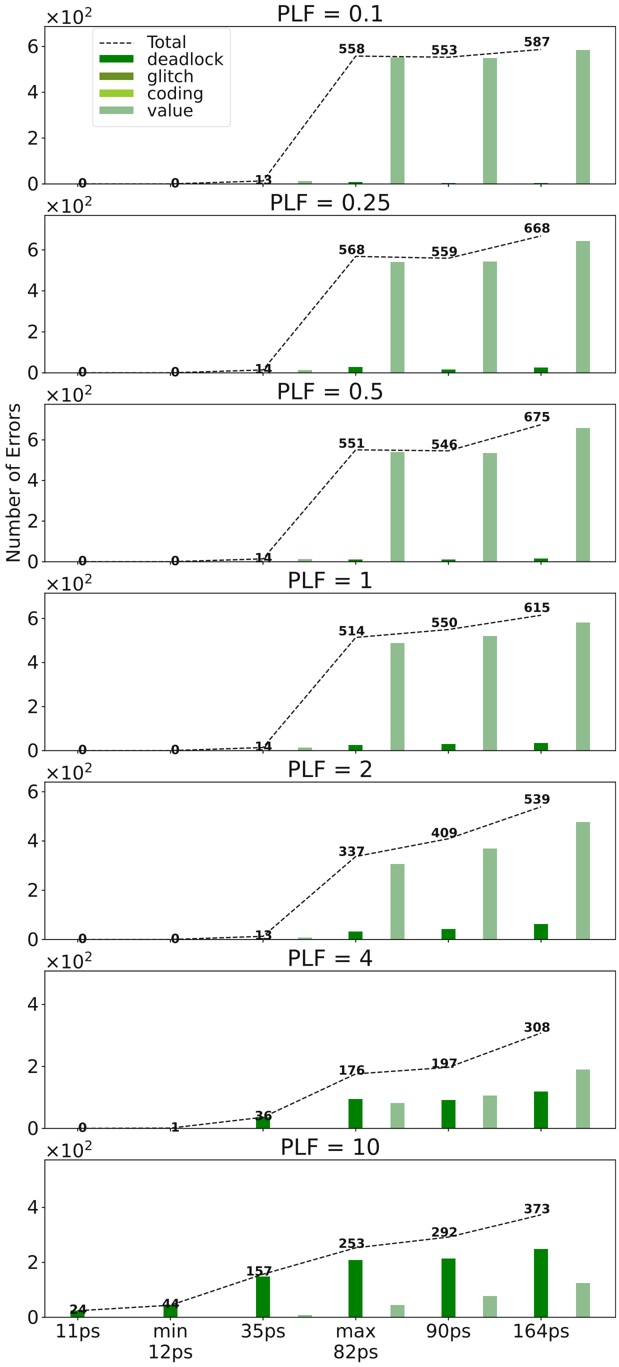


Figure 3.19: Suseptibility analyses of InOutInterlockWCHB under the influence of SETs with variable PLF and different injection pulse lengths

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

In the following discussions we will investigate what the main reasons for errors in each mode of operation are, if the input interlocking is beneficial, and what we can achieve with this enhancement.

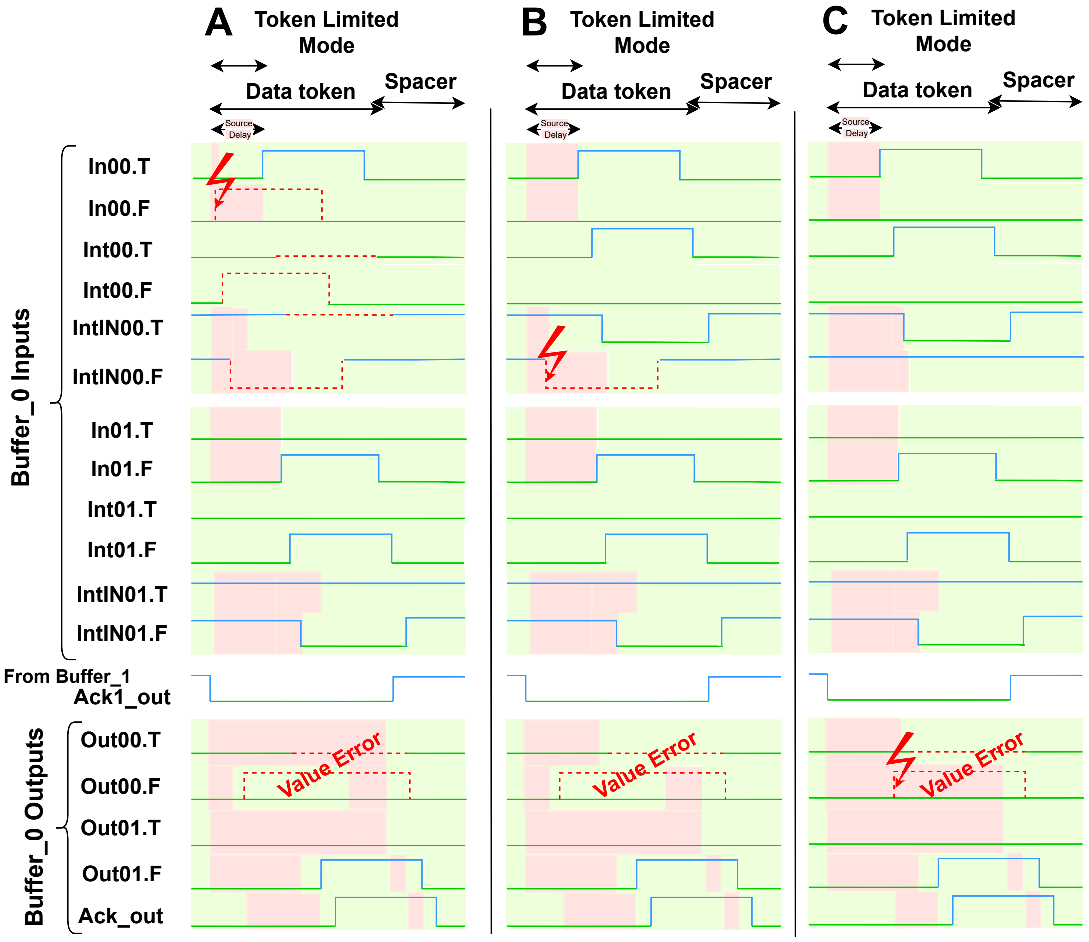


Figure 3.20: Sensitive Windows of InOutInterlock WCHB during Token Limited Mode

3.3.1 Main Causes of Errors in Token Limited Mode

From Figure 3.19, we conclude that most of the time the circuit shows a value error when it is not able to mitigate the SET effects. In the following we will discuss the main reasons behind the generation and propagation of the value errors.

1. Figure 3.20-A: fault appears during a sensitive window while we are waiting for the valid transition. Before the valid data token arrives the faulty transition gets latched by the respective MCE, because the *Ack*1_*out* is already there and the buffer is armed for the data token. This time the fault not only propagates to the

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

output, but it blocks the valid transition at the input level. Clearly, when *Int00.F* goes high, *Int00.T* is forced to remain low for this handshake cycle. This scenario is a clear evidence that the input interlocking utilized in this methodology is not appropriate for the token limited mode. It makes it even worse; the valid transition is not even able to propagate to the main buffer.

2. Figure 3.20-B: the internal signals are also sensitive, but as shown most of the time they are sensitive to a negative pulse, due to the inversion by the input gates. In the shown scenario a negative pulse appears during the sensitive window that is generated due to the source delay. It hits *IntIN00.F*, and as it matches with the *Ack*1_*out*, it gets latched by MCE-C1 while disabling the valid transition arriving at *IntIN00.T* after the source delay. A value error is generated and propagated to the sink in this case. The important thing to notice here is that this is the same susceptibility of the main MCE inputs as with to the Interlocking WCHB. On top of this main MCE input sensitivity we add other sensitive nodes associated with the input interlocking MCEs as discussed above, and as also visible from this part where *In00.T, and In00.F* are sensitive to SETs as well.

3. Figure 3.20-C: during token limited mode the buffer outputs are also sensitive to SETs, in the same way as with the Interlocking WCHB. A fault can easily flip the MCE state from the output side because the MCE is in storage mode. In the observed scenario the fault appearing at *Out00.F* during the sensitive window is not only retained by MCE-C1 but also blocks the valid transition waiting at the *IntIN00.T*. In this way a wrong data word is generated and passed to the next stage buffer as shown. This will finally result in the value error flag getting raised.

3.3.2 Main Causes of Errors during Balanced Operation

During balanced operation InOutInterlock WCHB shows a value error if the fault is not logically or temporally masked by the buffer. Again, it is more susceptible to SETs than the Interlocking WCHB. The one main difference is that we now also observe deadlocks, where we did not observe any of these with the Interlocking WCHB. In the following we discuss the main reasons behind the observed error types.

1. Figure 3.21-A: fault appears at *IntIn01.T* just before the valid transition arrives and propagates to the output, as *Ack*1_*out* is already low. In this case the fault disappears from *IntIn01.T* before the valid transition happens, but its effect remains there at *Out01.T*. It then blocks the valid transition from the output side by disabling MCE-C3. Clearly, the expected transition on *Out01.F* is not able to reach there. In this case the value error is propagated to the next buffer stage.

2. Figure 3.21-B: in the observed scenario the fault coincides with the valid transition, but the position of the fault is at the output of MCE-C1 that is supposed to remain zero during this handshake cycle, where the valid transition is at the input of C0

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

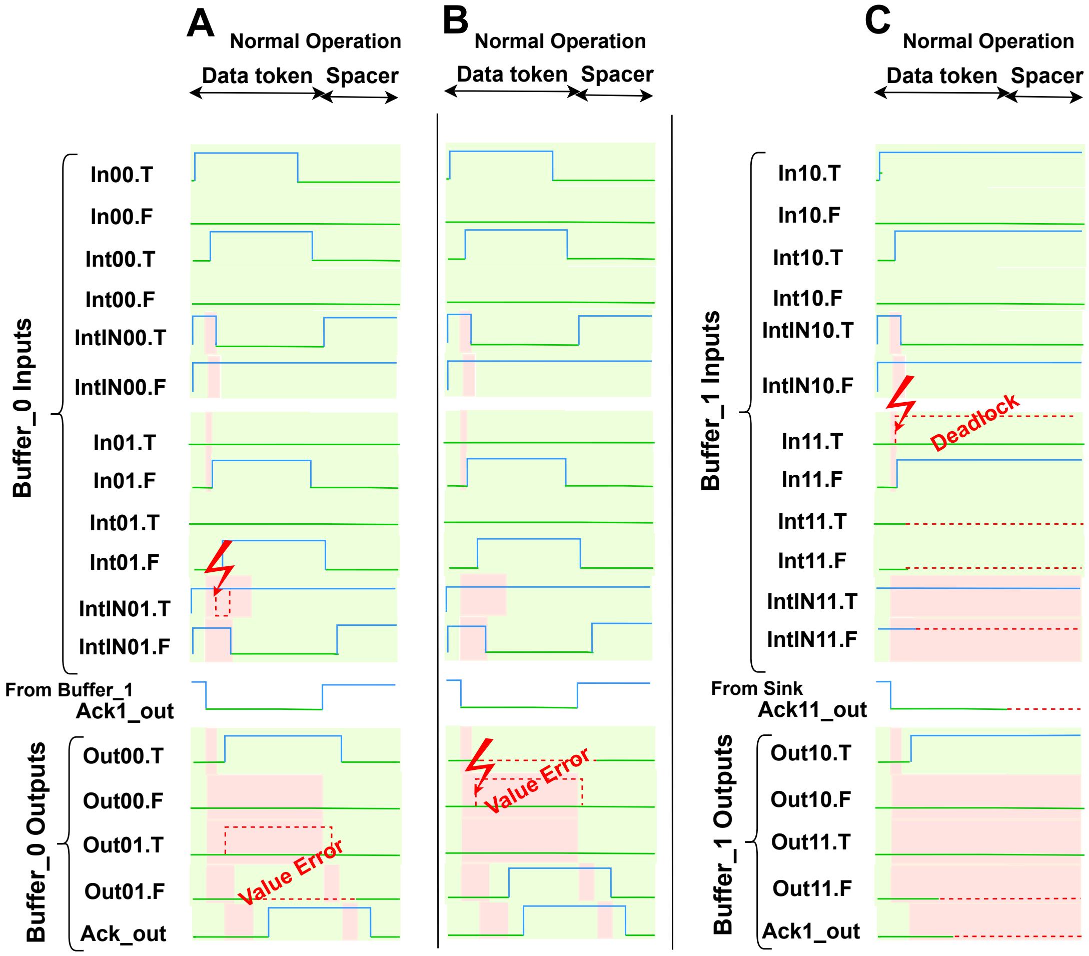


Figure 3.21: Sensitive Windows of InOutInterlock WCHB during balanced operation

as visible. Nevertheless, the valid transition will only arrive at $Out00.T$ after the propagation delay of the gate C0. But in this case the fault disabled MCE-C0 during this propagation delay window to fire. At the same time the faulty transition also flips the storage loop from the output side of C1 because it is in storage mode ($IntIN00.F = 1$ and $Ack1\_out = 0$). The faulty value is passed to the next stage instead of the valid data token, which then raises the value error flag.

3. InOutInterlock WCHB also shows a small percentage of deadlocks. So, in the last case we will discuss one of its observed root causes.

Figure 3.21-C: the fault appears at the input of $Buffer\_1$ $In11.T$ in very close proximity with a valid transition at $In11.F$. Now both input interlock MCE's are disabled for a high transition at the same time. In the specific case of our Null Function in the combinational part, the fault is latched by the previous stage's

MCE if the latter is in a combinational mode. In the illustrated case C2 therefore indeed latches the fault. Where for other logic functions like multiplier or adder this fault may be latched by the combinational circuit's MCE, but this depends on the specific signal source node. If the node is fed by any purely combinational gate this fault does not affects the integrity of the data. So even a buffer will provide good mitigation here. Moving back to the presented scenario, the fault persistis in C2. Consequently, both input interlocking MCE remain locked by these persisting HI inputs as shown and we are not able to progress the handshake further. The circuit is deadlocked for any further transitions.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

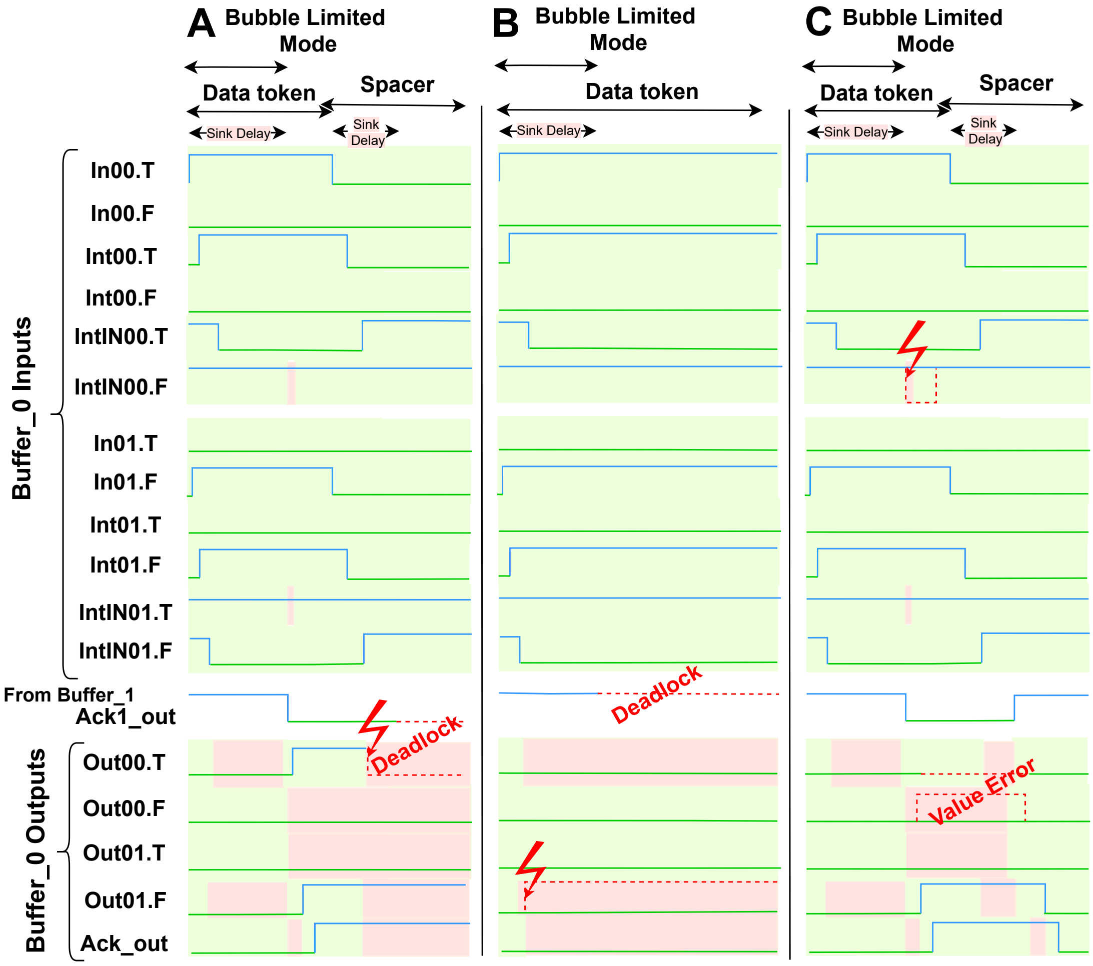


Figure 3.22: Sensitive Windows of InOutInterlock WCHB during Bubble Limited Mode

3.3.3 Main Causes of Errors during Bubble Limited Mode

In extreme bubble limited mode, the buffer template shows better resilience towards SETs. The number of errors dropped from 1006 (with the Interlocking buffer template) to 253 when we inject a pulse that is equal to the maximum propagation delay of a gate a circuit contains. This time the prominent error type is deadlock. During our discussion we also refer to Figure 3.16, if the signal is not listed in Figure 3.22 please check the Figure 3.16 for better understanding.

1. Figure 3.22-A: fault appears at the output of the MCE that is holding the data token for the next stage, but it appears during the waiting window highlighted with “Sink Delay”. Consequently, the respective MCE goes into storage mode because the data token is removed from the inputs *IntIN00.T*. In this scenario the fault forcefully removed the data from the respective rail before the next stage has acknowledged it on *Ack1_out*. As one of the rail data is pending and we are not able to advance the handshake phase without the expected transition that was deleted by the SET, as a result the circuit entered the deadlock situation as shown.

2. Figure 3.22-B: in the observed scenario, the spacer is latched by $Buffer\_0$, as all the outputs are at logic zero. However, the last data token is not retired from $Buffer\_1$ due to the sink delay; this is visible from $Ack1\_out$ that is still high. At the same time $Buffer\_0$ advances the handshake with the source and as a result the new data token arrives at the inputs of $Buffer\_0$; *In01.F* goes high which makes *IntIN01.F* low. The SET hits *Out01.F* at a moment when the MCE-C3 is already in storage mode, so it gets latched by C3 (Please note that the inverter bubble in front of the buffer MCE’s are symbolic, the real inversion is performed internally). This transition when arrived at *IntIn11.F* will again change MCE C8 into combinational mode, as $Ack11\_out$ is still low. After some arbitrary time when the sink responds and $Ack11\_out$ goes high to latch the spacer from $Buffer\_0$, all the other MCEs of $Buffer\_1$ go to low, except C8 which then forces $Ack1\_out$ to remain high, as visible from the red dotted line. In this way we are not able to advance the handshake between $Buffer\_0$ and $Buffer\_1$ and remain stuck in a deadlock situation.

3. Figure 3.22-C: the input of the buffer shows better resilience towards SETs in bubble limited mode, as most of the time it is green as compared to the other modes. But we still have some very small sensitive windows when $Ack1\_out$ goes low until the input transitions passed to the output of the MCE. This time the fault hits that sensitive window as shown. The propagation delay of the respective gate is less than the other rail MCE, so the transition arrives at the output before the other rail latched the data. This erroneous transition then blocks the valid one and generates the value error as shown.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

3.3.4 Final Comments on InOutInterlock WCHB

Figure 3.23 presents the comparison between the discussed buffer templates. For this comparison we set the same injection pulse for all, which is 60ps, ensuring that it is higher than the minimum threshold to observe some effects. When we move from the Interlocking WCHB to the InOutInterlock WCHB, it is clearly visible that we achieve higher resilience. But a closer look to the value errors tells us the strength of the buffer template that is in extreme bubble limited mode. The reasons we do not acheive much benefit in token limited mode are the input interlocking MCEs and the way it is deployed, which makes the circuit more suseptible. The fist thing we learn is that the input interlocking can be effectively utilized if it will be deployed in a way that it covers all the modes of the circuit, and we have to do input interlocking without using a MCE, because a storage element adds a suseptible node.

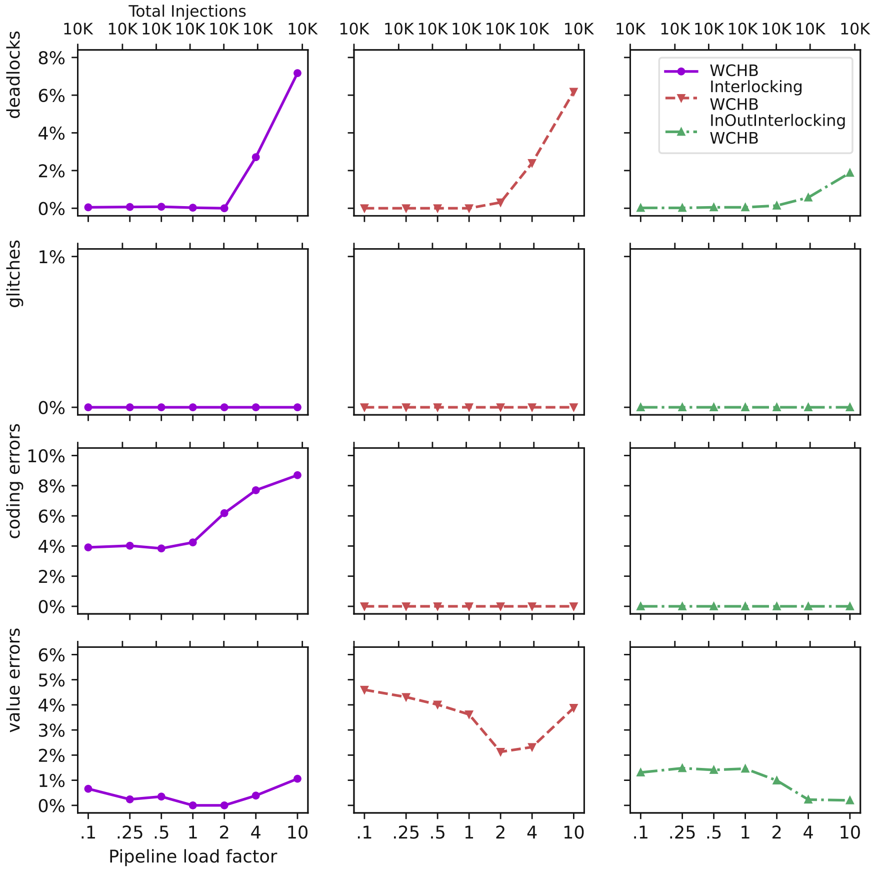


Figure 3.23: WCHB with InOutInterlocking WCHB (Injection pulse 60ps)

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

3.4 Input Output Interlocking with SR latch WCHB

The previous section showed that main problem with the InOutInterlock WCHB is the use of extra MCE for the input interlocking that makes the circuit more susceptible to the SET in some mode of operations. In [TS22b], we enhanced the InOutInterlock WCHB in a way that excludes the extra susceptible nodes.

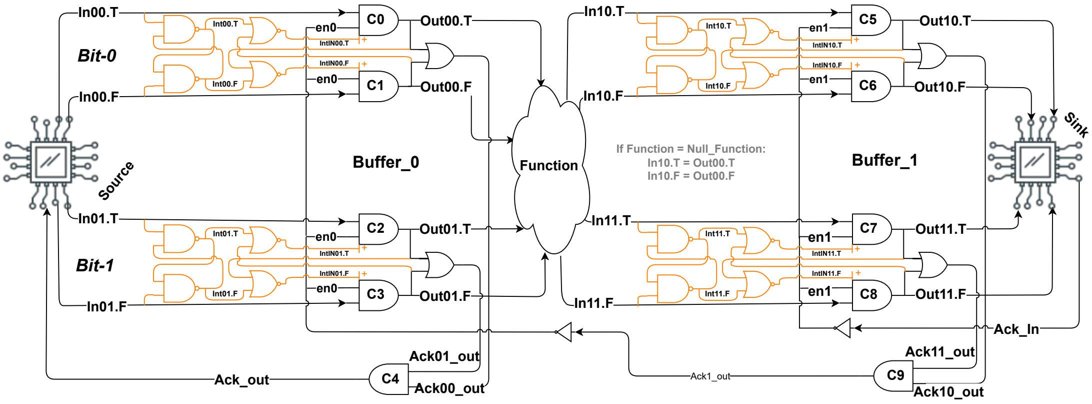


Figure 3.24: IOISRWCHB

We prefer using SR latches instead of MCEs for the input interlocking. This not only saves area, it also, most importantly, mitigates the problem of state flipping through faults at the output of the MCE used for input interlocking. This SR latch resembles the classical mutex implementation from [Sei80] – albeit without metastability filter – much closer than the interlocked MCEs did.

Importantly, the decision to arm the C-element+ is now combining input and output state using a NOR gate that, together with its companion from the other rail, forms another SR latch. We named this approach Input Output Interlocking with SR latch WCHB (IOISRWCHB). With the use of the purely combinational NAND and NOR gates for input and output interlocking the circuit may go back to its original state once the SET vanished out. The working principle is straightforward:

1. The first input blocks any further transition from the other rail by using the NAND gate-based SR latch.
2. The latched input is only able to arm the output MCE for a high transition if the other output MCE is storing logical zero.
3. If any of the rails' output MCE is storing high it simply forces the *IntIN* input to zero which then disables the respective MCE, as well as nullifies the input transition from *Int*.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

4. The *InXX.T* and *InXX.F* are directly connected with the normal inputs of the MCE, which gives us more information, when combined with the asymmetric input that is connected with the filter. This enhances the resilience towards SETs compared to the InOutInterlock WCHB where we only pass the filtered inputs to the output MCEs.

Here we first compare the sensitive windows of IOISRWCHB from Figure 3.25 with the InOutInterlock WCHB from Figure 3.17. As already pointed out in the discussion, the internal nodes of the locking mechanism are not sensitive anymore. This is now clearly visible from Figure 3.25 where *IntXX.T, IntXX.F* and *IntINXX.T, IntINXX.F* are throughout green but the primary inputs and outputs are still susceptible to SET. The susceptibility of these lines is the same as we observed in InOutInterlock WCHB.

For the evaluation we chose the 8-bit, 4-stage Empty Pipeline with the the IOISRWCHB buffer template. Figure 3.26 gives us insights about the gate delays of the target circuit and the gate count. The input filter MCEs of the InOutInterlock WCHB are replaced with the NAND gates so the region between 10 to 30ps is denser compared to the Figure 3.18. The maximum propagation delay a gate contain is 58ps where the minimum delay is 12ps.

By following the symmetry, we performed the simulations with 6 different injection pulses, 11ps to 116ps (that is with addition of 100% to the maximum). The first thing we observed from Figure 3.27, is in the results of 35ps: compared to its parent buffer template the circuit shows weak resilience. We investigate it and the reason is not surprising: the gate delay of the two-input NAND and NOR is sometimes less than 35ps, so the SET can pass the filter and is considered as valid token. Moving to the higher pulse lengths the susceptibility of the circuit increases but compared to its parent template it shows better resilience. It suffers from approximately half the number of error occurrences compared to the InOutInterlock WCHB. If we compare the 90ps from Figure 3.19 with 116ps of Figure 3.27 it is clearly visible that the IOISRWCHB is better in Token limited mode but in extreme bubble limited mode the resilience is approximately the same, as the IOISRWCHB is designed to eliminate the main problems raised by the dual MCE per rail of the buffer which makes the circuit more susceptible in the token limited mode. In the next sections we will take a closer look at the main causes of error in each mode of operation.

3.4.1 Main Causes of Errors in Token Limited Mode

As Figure 3.27 suggests, in Token limited mode the only observed effect is value error. With reference to Figure 3.24, we highlight these occurrences in Figure 3.28.

1. Figure 3.28-A: The MCEs are in storage mode until the new data token arrives. The sensitivity windows of MCEs from the output side is the same as InOutInterlock WCHB. This time the SET hits *Out00.F*. This transition forces *IntIN00.T* to

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

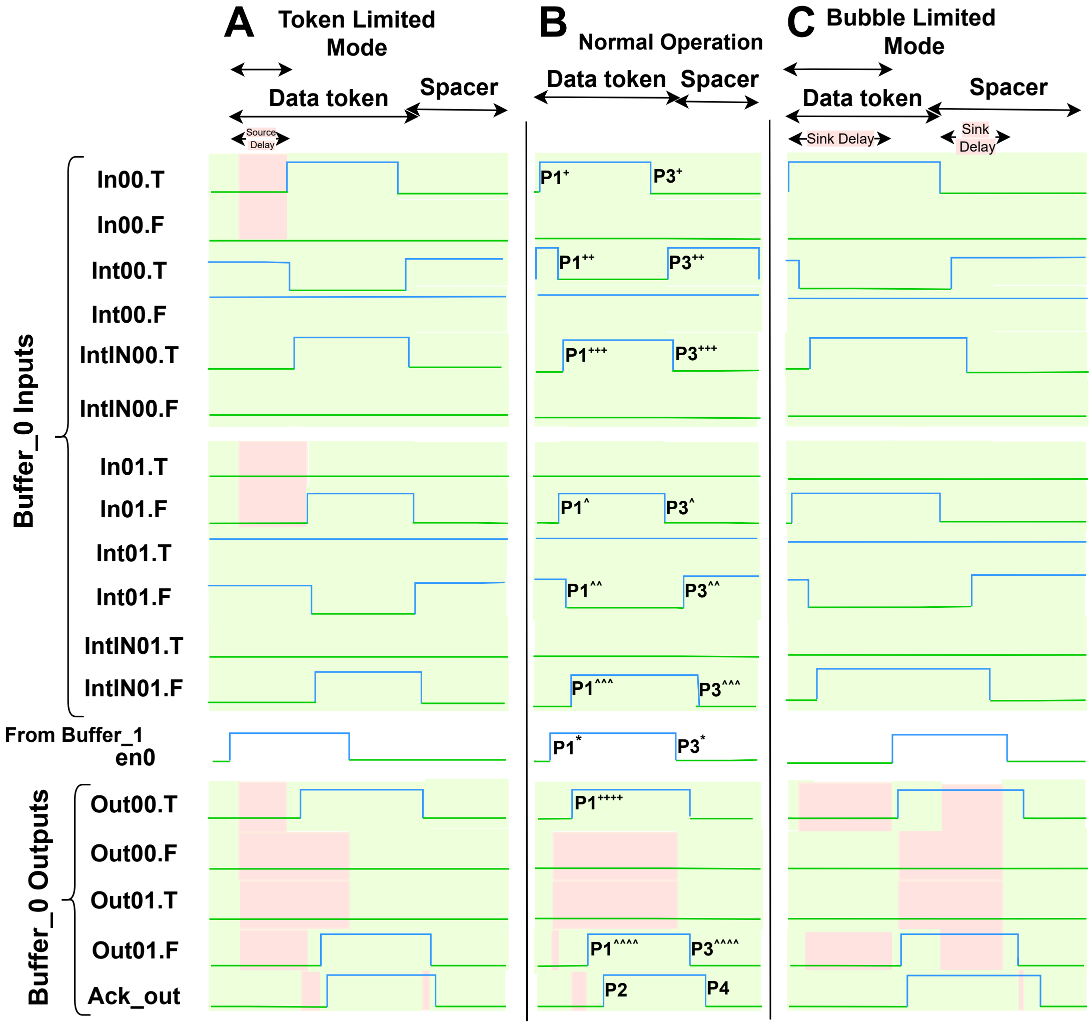


Figure 3.25: Sensitive Windows of IOISRWCHB

remain low after *Int00.T* goes low. As shown the correct transition is now waiting at the input of MCE-C0 and the SET is latched by MCE-C1 and considered as valid transition raising the value error flag.

2. Figure 3.28-B: for explanation purposes we change the target buffer; now *Buffer_1* is selected, and the fault hits *In10.F* very close to the valid transition. However, it is early enough to win the input interlocking, so, making its way to the output because *en1* is already high. The value error appears at the output while disabling the valid transition from propagating. After some arbitrary time, the fault disappears from *In10.F* allowing the valid transition at *In10.T* to propagate to *In10.T*. But this is worth-less because the next stage already acknowledged the token, as *en1* goes low.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

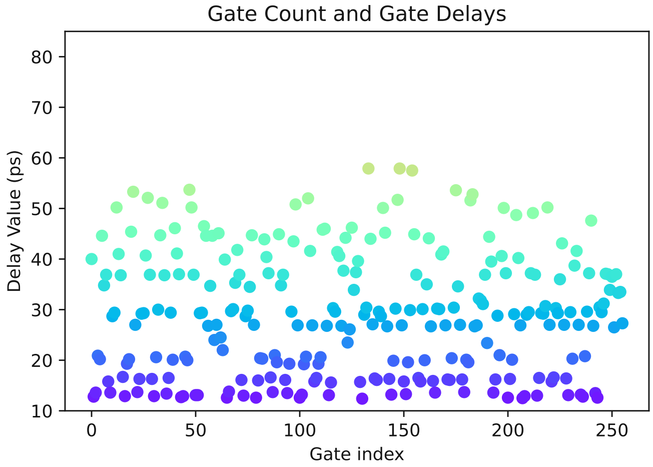


Figure 3.26: IOISRWCHB Gate Count (256) and Gate Delays

3. Figure 3.28-C: the fault appears at *In11.T* during the sensitive window as shown. As no valid transition is there, it makes its way to the output and gets latched by MCE-C7 that is supposed to be remain zero during this handshake phase. The fault length is so small and vanished out (from input ((In11.T)) before the valid transitions appears at the other rail (*In11.F*), but as C7 already latched, the faulty value remains there for this handshake cycle and raise value error.

With thorough investigation it becomes clear that most of the time errors are only caused by the positive SETs.

3.4.2 Main Causes of Errors during Balanced Operation

In a balanced mode the circuit generally shows better resilience than its parent template, compared with Figure 3.19 PLF=1. For instance, when injecting 90ps SETs the Figure 3.27 only shows 225 errors with the injection of 116ps injection pulses, while the parent circuit failed 550 times. The inputs of the buffer template are totally green as shown in Figure 3.25 for normal operation, which is the indirect argument why the number of error occurrences goes half compared to the InOutInterlock WCHB. In the following discussion we will figure out the main reasons behind that.

It should be noted here that for the Figure 3.29 we only discuss scenarios where the fault hits the output of an MCE. This is backed up by the simulation results we were examining: all error scenarios we observed could be backtracked to the output node of an MCE.

1. Figure 3.29-A: the fault hits *Out01.T* after the valid input arrived at *In01.F*. The valid transition advances to *Int01.F*, but the fault forces *IntIN01.F* to remain low,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

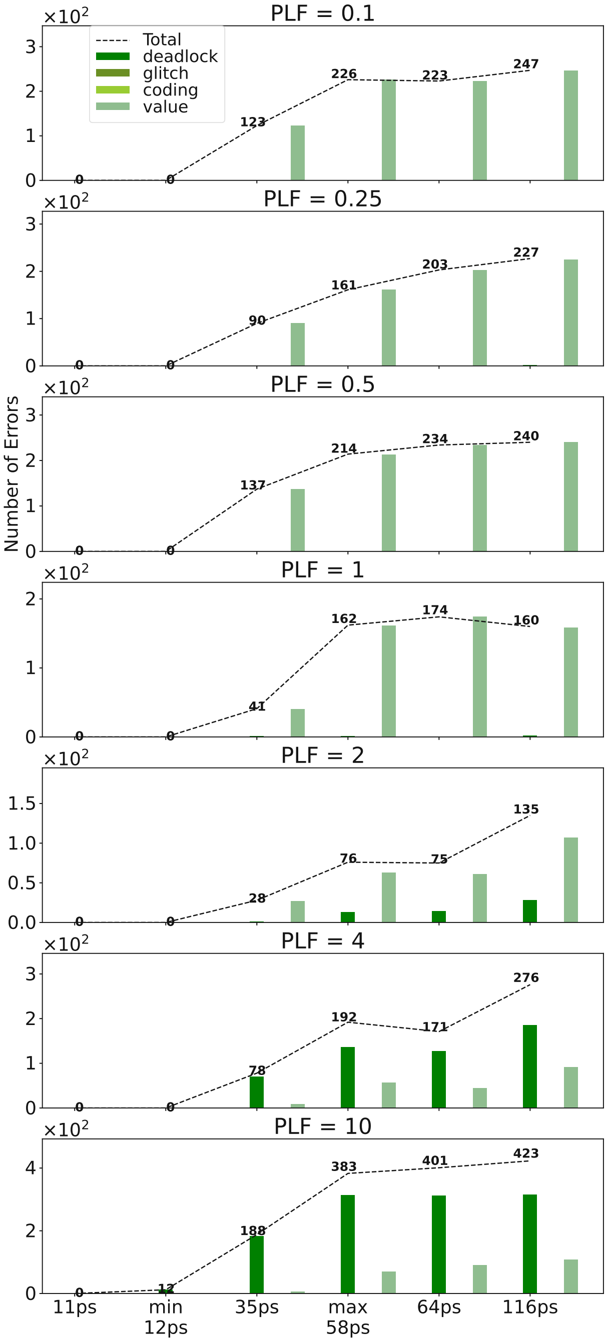


Figure 3.27: Suseptibility analyses of IOISRWCHB under the influence of SET with variable PLF and different injection pulse lengths

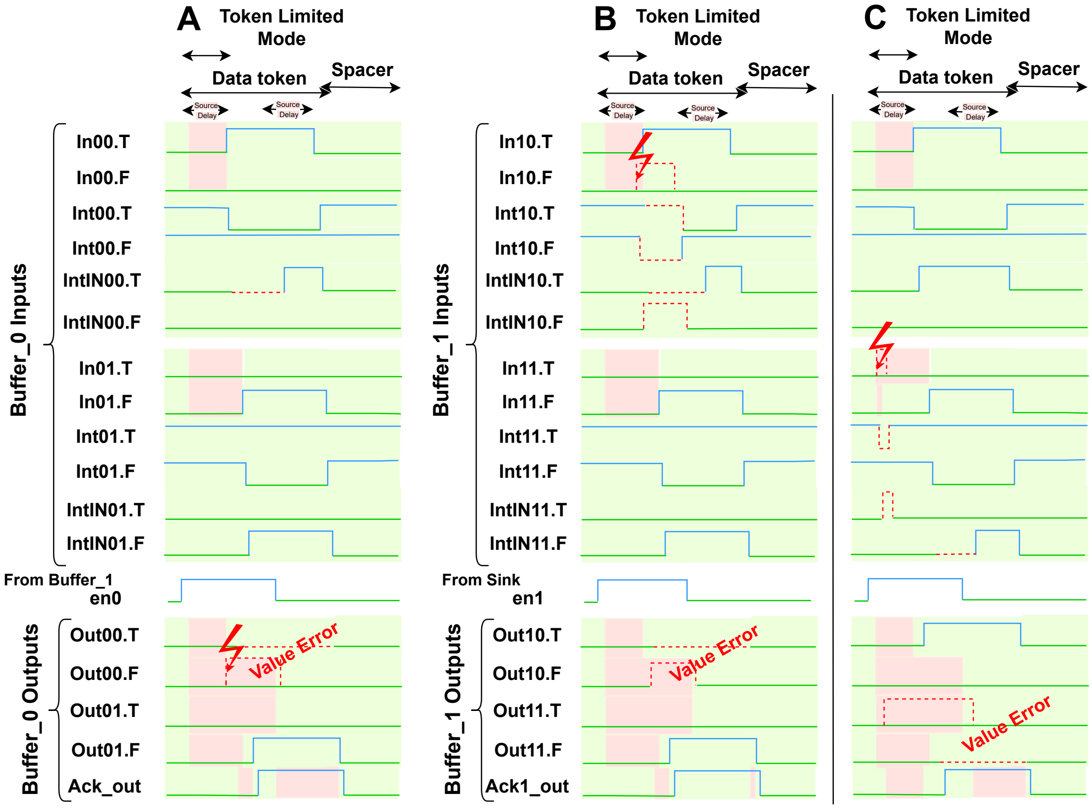


Figure 3.28: Sensitive Windows of IOISRWCHB during Token Limited Mode

and this SET also gets latched by MCE-C2 which is then considered as valid token as shown, causing a value error at the output.

2. Figure 3.29-B: the fault scenario is the same as the previous one but here a very important thing to be noticed is the time of fault appearance: It appeared in the green window while MCE-C1 is in combinational mode, which means *en0* is also low as shown. But the length of the fault is long enough to enter the red window and so it gets latched by C1 once the *en0* goes high. The remaining behavior is the same as in part A.

3. Figure 3.29-C: again the SET hits the output and raises a value error, but this time the valid input transition is not only passed by the first input filter but also by the output filter as shown. As the NOR gate is connected here with *Out01.T*, the logical high at *IntIN01.F* (valid token) is forced to low as shown, which then disables MCE-C3 from firing the valid transition. In this way we are not producing a coding error, but still the SET gets latched by C2 and propagates, producing a value error.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

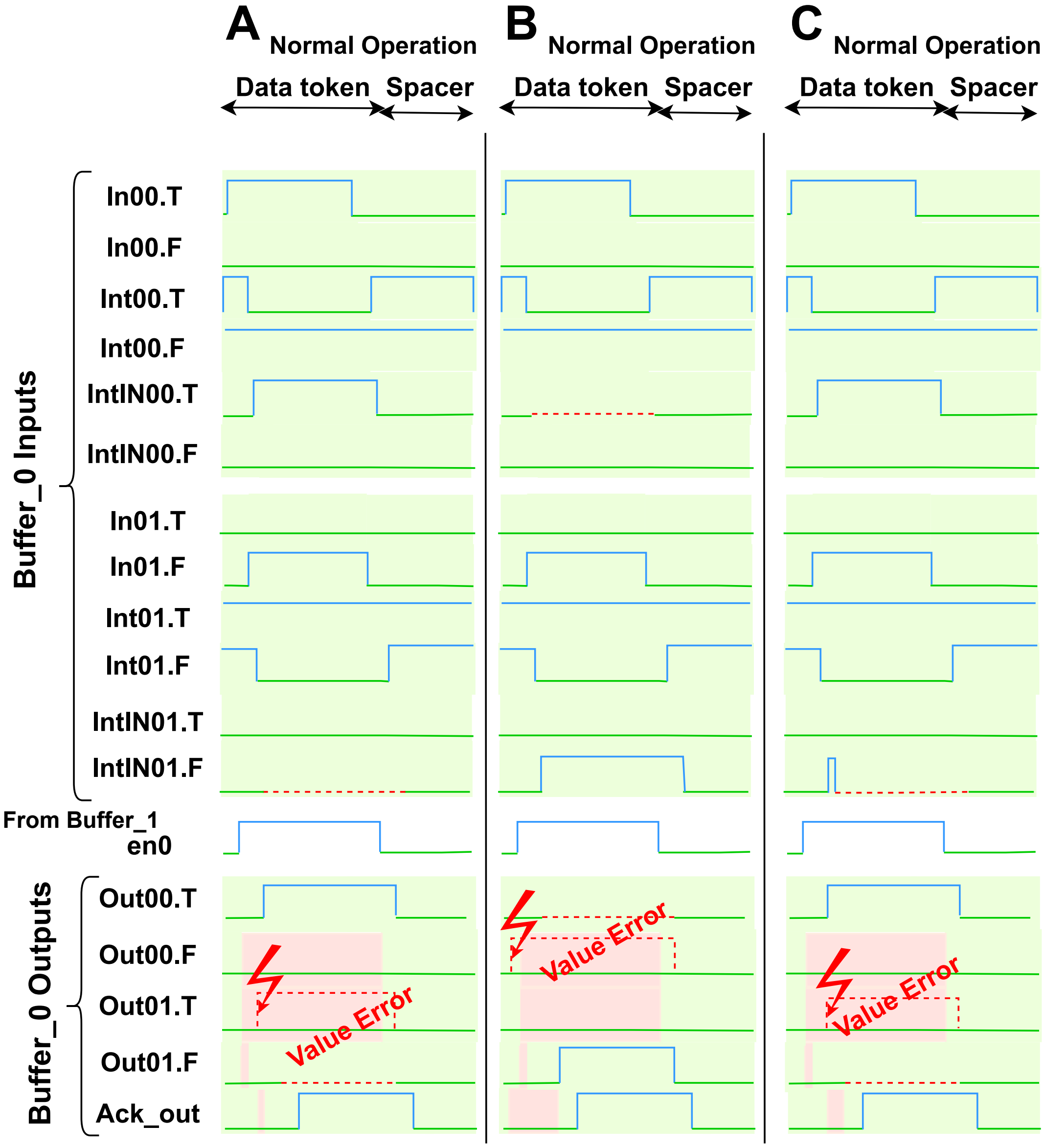


Figure 3.29: Sensitive Windows of IOISRWCHB during balanced operation

3.4.3 Main Causes of Errors during Bubble Limited Mode

Comparing Figure 3.27 in bubble limited mode with Figure 3.19 justifies the findings we made in context with Figure 3.25-C: with the modifications we applied we do not achieve any resilience gain in this mode. The circuit is even a bit more susceptible in this mode. Anyways, our main goal was to eliminate the extra MCE to achieve better resilience in Token limited mode, which we definitely achieved. In extreme bubble limited mode, the main contributor is deadlock, as Figure 3.27 suggests. With the help of Figure 3.30, we

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

will explain the main causes of deadlocks in the following discussion. We found that most of the time the main cause are output signals:

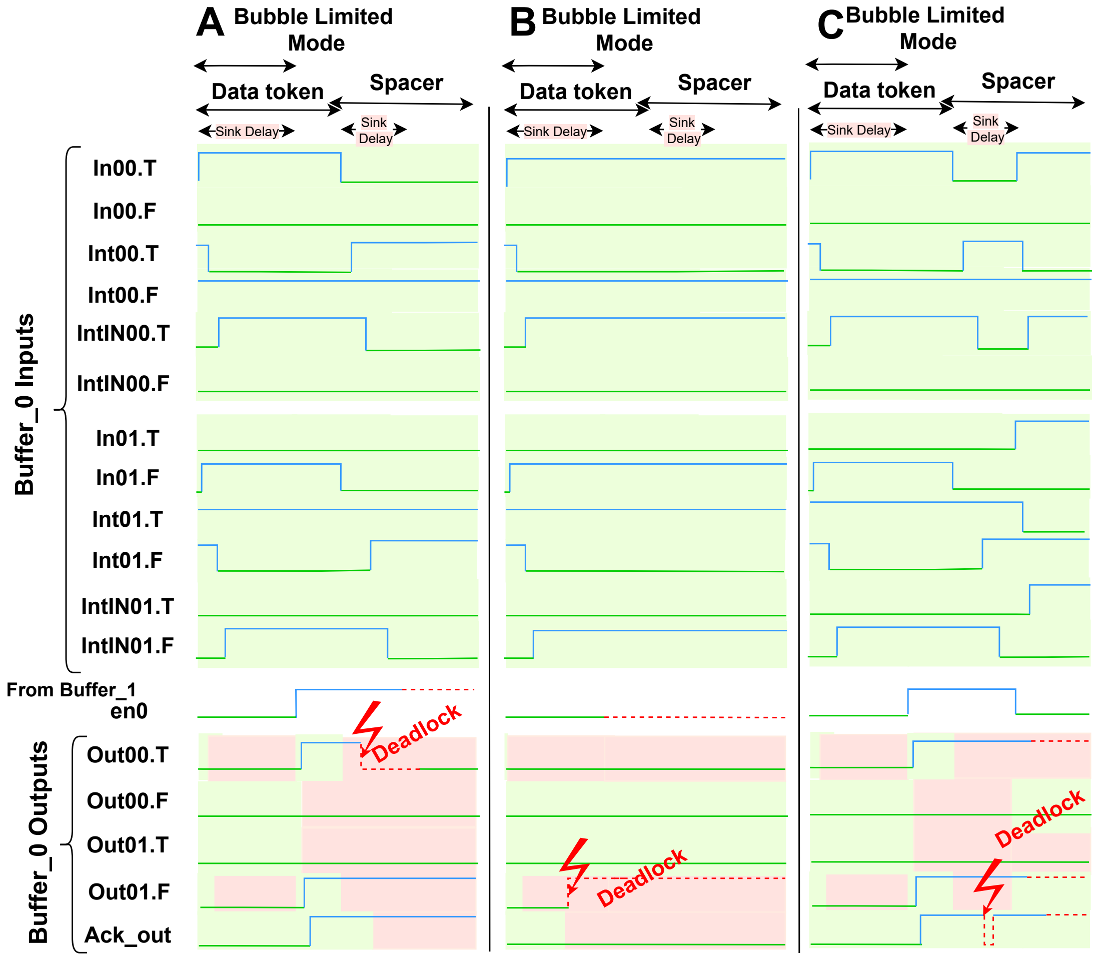


Figure 3.30: Sensitive Windows of IOISRWCHB during Bubble Limited Mode

1. Figure 3.30-A: fault occurs *Out00.T* at a point in time where the token has been latched by *Buffer_0* and the source has received the *Ack_out*, but the token has not yet been taken over by *Buffer_1*. *Buffer_0* holds the only copy of the token, and when that one is cleared by the SET, there is no recovery. As a result, the next stage remains in a waiting state for that removed data token and we are stuck in this handshake phase – which raises a deadlock situation.
2. Figure 3.30-B: *Buffer_0* after completing its last handshake with the source is now waiting for the next stage to respond to the spacer initiated, so that it can accept a new data token from the source. MCE-C0 and C3 are in a storage mode because

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

the input signals already changed, while the *en0* is low. As shown the fault appears at *Out01.F* inverting the stored value from logical zero to one. In this case we are not able to complete the last handshake phase with the next stage, as the spacer of Bit-1 is converted into an invalid data token. As shown the deadlock flag goes high.

3. Figure 3.30-C: this case is a bit different from those we presented so far. Here it is also important to recall that we mark the sensitivity window red where the MCE is in storage mode, while green represens the combinational mode. This is also the first time that we are going to discuss the acknowledgement signal. This signal does not contribute much but still accounts for some occurrence. So, as shown, the SET appears at *Ack_out* when MCE-C4 is in combinational mode. This means the fault will only be able to last for its original length. Here the important thing to notice is that *Ack_out* was high before the SET, telling the source that the data token was successfully received, and requesting the spacer. This small SET is long enough to be considered as valid acknowledgment of the spacer that is still waiting at the input of *Buffer_0*. This means the handshake with the source is completed, and so the source places a new data token at the inputs as shown. Now the spacer for the running handshake cycle with *Buffer_1* cannot be initiated, as that spacer is not able to propagate due to the erroneous acknowledgement. We experience a deadlock in this situation because the last latched token will never be retired from *Buffer_0*.

3.4.4 Final Comments on IOISRWCHB

To compare the IOISRWCHB with all the discussed buffer templates we choose the “maximum plus 100% of the maximum” injection pulse results from all of the susceptibility analysis figures. As shown in Figure 3.31 we only present deadlocks, coding and value errors, as we already discussed the cause of glitches and removed the last stage enable signal from the injection list. That is why we are not experiencing any glitches, so here it is not useful anymore to add the row with zero values. Moving back to the discussion on IOISRWCHB and Figure 3.31, in extreme token limited mode we achieve way better results with the same basic idea from InOutInterlock WCHB but with a different implementation. We conclude that we achieve higher resilience with the same gate count (Figure 3.26) and, relative to [TS22b], with approximately 50% reduction in transistor count. At the same time we attain faster operation because of the simpler input interlocking mechanism. The first main reason is that we are not using MCEs anymore for the input interlocking, which excludes two sensitive nodes from each bit. These enhancements allow us to fully utilize the input interlocking strategy, as visible from the results of extreme token limited mode. Secondly, the input filter is more effective in a way that if the length of the fault is shorter than the NAND plus NOR gate delays, the input is already changed back to the regular level when the fault appears at the positive asymmetric input of the MCE. In this situation the fault has no impact on the circuit. We can also add another argument to back the resilience of the circuit in token limited mode: in contrast to the InOutInterlock WCHB, with this configuration we are

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

not blindly blocking the other rail for further transition at the input, but it only blocks if it passes through the first NAND gate filter Figure 3.24.

For future direction we have also identified state flips of the MCE (while in state holding mode) through SETs on its output as a main contributor to error effects. So our deep analysis suggests that if we can somehow maintain the symmetry of input lines of the MCE during the waiting time of the pipeline (namely in token or bubble limited modes), faults at the output of the MCE lose their effectiveness. As a result, the remaining 1 to 3% faults are easily addressable.

With all these insights we can move on and will further enhance the resilience of the circuit by blocking the acknowledgment until all the bits arrive. That may add more resilience against SETs in Token limited mode. In the next section we continue our investigation to achieve our goal.

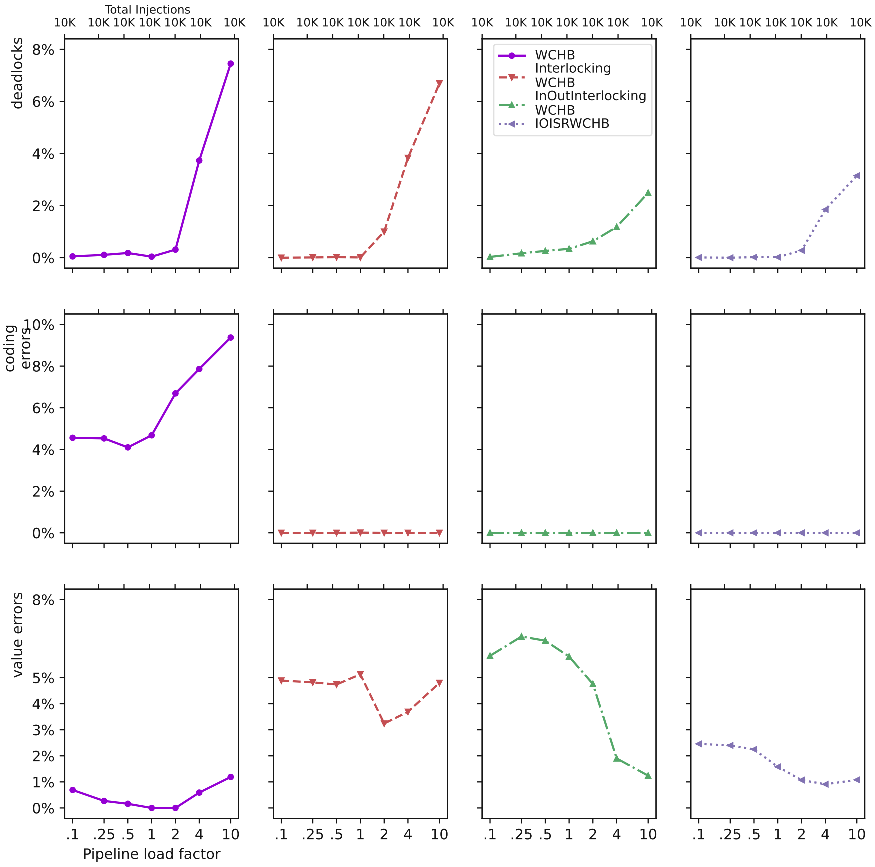


Figure 3.31: Comparison with IOISRWCHB (Injection pulse is equal to 2*(maximum gate delay))

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

3.5 Δ: Dual_Completion_Detection Input Output Interlocking WCHB

From the investigations we conclude that if we somehow maintain the symmetry of the MCE's input until both source and sink respond to the last respective token, we can shorten the sensitive windows of our circuit. In the last section we achieved a higher resilience in Token limited mode by carefully implementing the input interlocking. In [TS22a] we proposed a technique which utilizes the Dual completion detection from [BS09] (originally known as "normally closed pipeline latch") to shorten the armed windows of the MCEs. As the analysis presented in [TS22a] shows, using dual completion detection without any supportive SET hardened buffer technique only converts the coding errors into value errors. So in its original form it is not worthwhile to be deployed for better resilience.

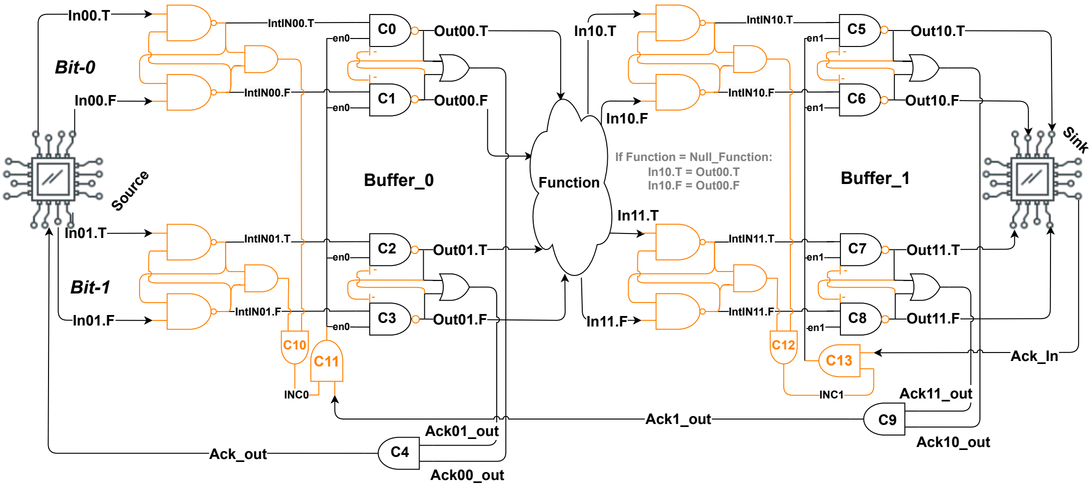


Figure 3.32: DualCD InOut WCHB Δ

We know that in Token limited mode data arrives late, and if we somehow block the acknowledgement signal to the buffer until all of the bits arrived, we can easily maintain the MCE's input symmetry for longer than normally. This keeps the MCE in combinational mode, during which, as we know, it is hard for the SET to corrupt its logical state either from the input or output side. Dual completion detection Input Output Interlocking, also known as Δ, is basically an enhanced version of the InOutInterlock WCHB with the dual completion detection being efficiently embedded into a buffer to acheive higher resilience against SETs, where the Δ approach is also inspired from the IOISRWCHB. Figure 3.32 presents an illustration of a 2-bit, 2-stage pipeline based on Δ.

The working principle of Δ is as follows:

1. A high transition on any input *InXX.T or InXX.F* is only allowed to pass the NAND gate filter if the other rail does not show any transition before its arrival.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

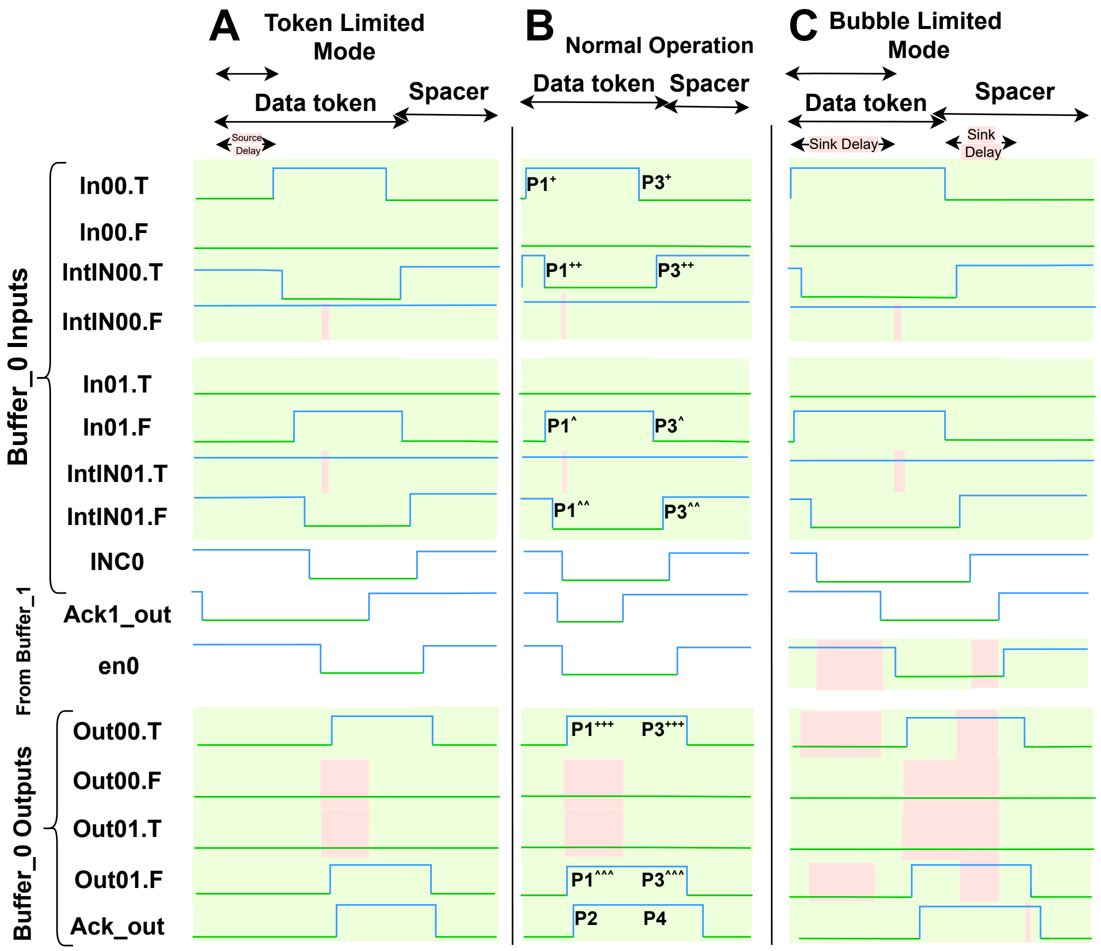


Figure 3.33: Sensitive Windows of Δ

This will then block any further transition by setting the other rail *IntINXX.T or IntINXX.F* high.

2. As the buffer works on inverted signals, the high transition on *InXX.T or .F* will be inverted by the NAND gate and we get a logical zero at *IntINXX.T or .F*.
3. This logical zero when matched with the low transition on *enX* will be stored by the respective MCE as logical high and the output will also go high.
4. However, the low transition on acknowledgement signal *AckX_out* is only passed to *enX* whenever *INCX* goes low, and this happens after the arrival of all bits.
5. In this way we somehow maintain the buffer MCEs in combinational mode for longer than usual in Token limited mode, as *enX* remains the same as in the previous handshake phase until source provides a new token.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

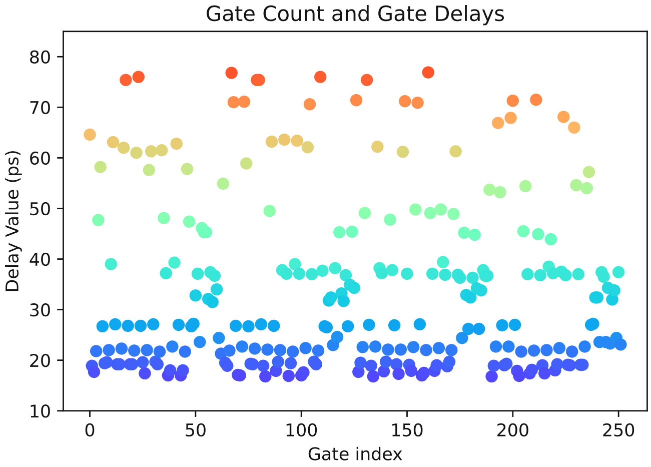


Figure 3.34: DualCD InOut WCHB Gate Count (252) and Gate Delays

6. The main buffer is the same as for the InOutInterlock WCHB, namely the MCEs with two normal and one negative inputs.

We first compare the sensitive windows of Δ from Figure 3.33 with IOISRWCHB Figure 3.25. It can be observed that in Token limited mode we are bit cleaner than the IOISRWCHB. Here a few differences must be checked before moving on, so we add the *Ack1_out* signal in our visual analysis as well as *INC0*. In Token limited mode input signals are green except for a very small sensitive window that is associated with the rail assumed to not participate in current handshake cycle. This is because when the *Ack1_out* signal is allowed to propagate to *en0* (high to low) it takes some time for the valid transition to propagate to the output of the buffer. Just recall that whenever *en0* goes low, a logical zero at *IntIN00.T* will show up at the output only after some propagation delay. Until it reaches the output *Out00.T* and blocks the opposite MCE, MCE-C1 enters into susceptible mode and any low transition is considered a valid transition.

Moving to the output signals, we achieved high resilience as shown. As *en0* remains high until the data arrived at the inputs (source delay) and the inputs *IntINXX.T and .F* are also high, our MCEs are in combinational mode and we can easily mitigate the SET effects from input and output sides. During Normal operation we also achieve a better resilience but in extreme Bubble limited mode we are not able to enhance our buffer, since we loose the resilience due to some signals which we will later discuss in detail. Here it is visible that we for the first time mark the *en0* signal's sensitivity in bubble limited mode.

As we are only injecting on nodes and considering a gate as atomic structure, we are comparing the area overhead on the basis of gate count. Figure 3.34 presents the gate count, which is the same as IOISRWCHB, whereas the gate delays are a bit different, as,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

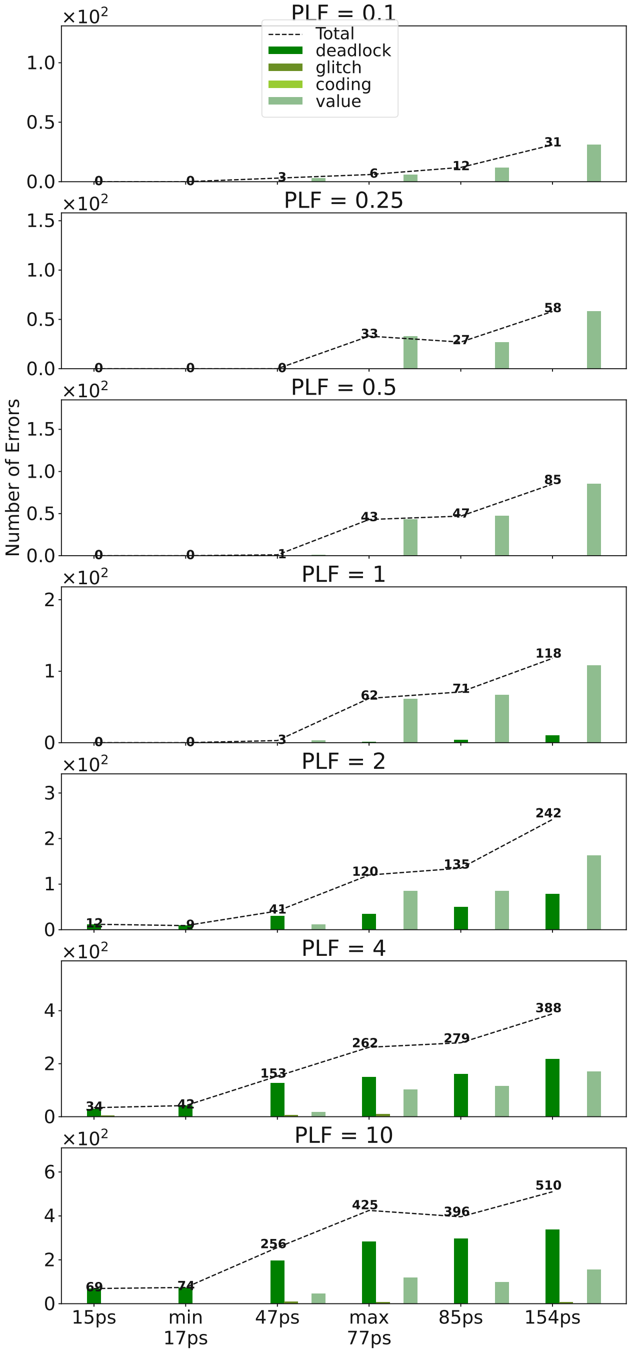


Figure 3.35: Suseptibility analyses of Δ under the influence of SET with variable PLF and different injection pulse lengths

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

in contrast to the IOISRWCHB, now a few gates cross the 60ps limit. The minimum propagation delay a gate contains is 17ps, with the maximum being 77ps. Again, we performed simulations with 6 different injection pulse lengths and varying PLFs as shown in Figure 3.35. Our target circuit is the same as before, namely the 8-bit, 4-stage Empty Pipeline.

From Figure 3.35 we start observing errors with 15ps in bubble limited mode, while in token limited mode fault effects remain rare, even with 154ps fault length(100% of the maximum). In balanced mode our approach shows better resilience compared to IOISRWCHB. In the next sections we will take a closer look at the main causes, which will help us to conclude what we achieve and where we still lack.

3.5.1 Main Causes of Errors in Token Limited Mode

Δ performs very well in extreme token limited mode as the results in Figure 3.35 suggest. The circuit fails to mitigate the fault only 31 times while injecting a pulse with a length of twice the maximum gate delays a circuit contains. Where we only observe value errors. The reason becomes already clear when observing in Figure 3.33-A that most of the time the signals are green shaded. With the help of Figure 3.32 we present the observed scenarios in Figure 3.36. We manually examine all scenarios when the circuit fails to mitigate the fault in extreme token limited mode ($\text{PLF} = 0.1$) while injecting 77ps pulse. We only observe scenarios, when fault hits the red shaded window of Figure 3.33-A. To present more scenarios we move to the 154ps injection pulse results, but the main reasons are approximately the same. In the following we will discuss these in detail:

1. Figure 3.36-A: fault appears at *Out00.F* at a time instance when *en0* goes low. As *IntIN01.F* is high, our MCE-C1 is in storage mode and the fault can flip the stored value if the fault length is longer than the propagation delay. This SET also disables MCE-C0 for its high transition, although *en0* and *IntIN00.T* are low: C0 is not allowed to fire as the asymmetric negative input is high. Since the SET length is higher than the propagation delay of gate it is stored by C0 and considered as valid data token – which will finally raise the value error flag.
2. Figure 3.36-B: MCE C2 which did not transition in the current handshake cycle goes into sensitive mode from the input side when the *en0* goes low and *IntIN01.T* is high. Unfortunately the transition that appeared at *IntIN01.F* has not arrived at *Out01.F* which means C2's asymmetric negative input is low allowing it to fire. If the fault appears during this sensitive red window as shown at *IntIN01.T* and long enough to cross the propagation delay of the gate, it appears at the output and gets latched propagating the value error. Here another thing should be noticed: The fault changes the *IntIN01.F* signal from 0 to 1 minimizing the chances of the valid value to propagate.
3. Figure 3.36-C: this scenario is the same as "Figure 3.36-A", the only difference is that the fault appears during the non-sensitive window but is long enough to

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

extend into the sensitive red-shaded window. Another important thing is that it disabled C3 slightly before *en0* goes low, which means the chances of the valid value to propagate are less than in the scenario presented in Figure 3.36-A. These results are found from injection pulse length 154ps, token limited mode.

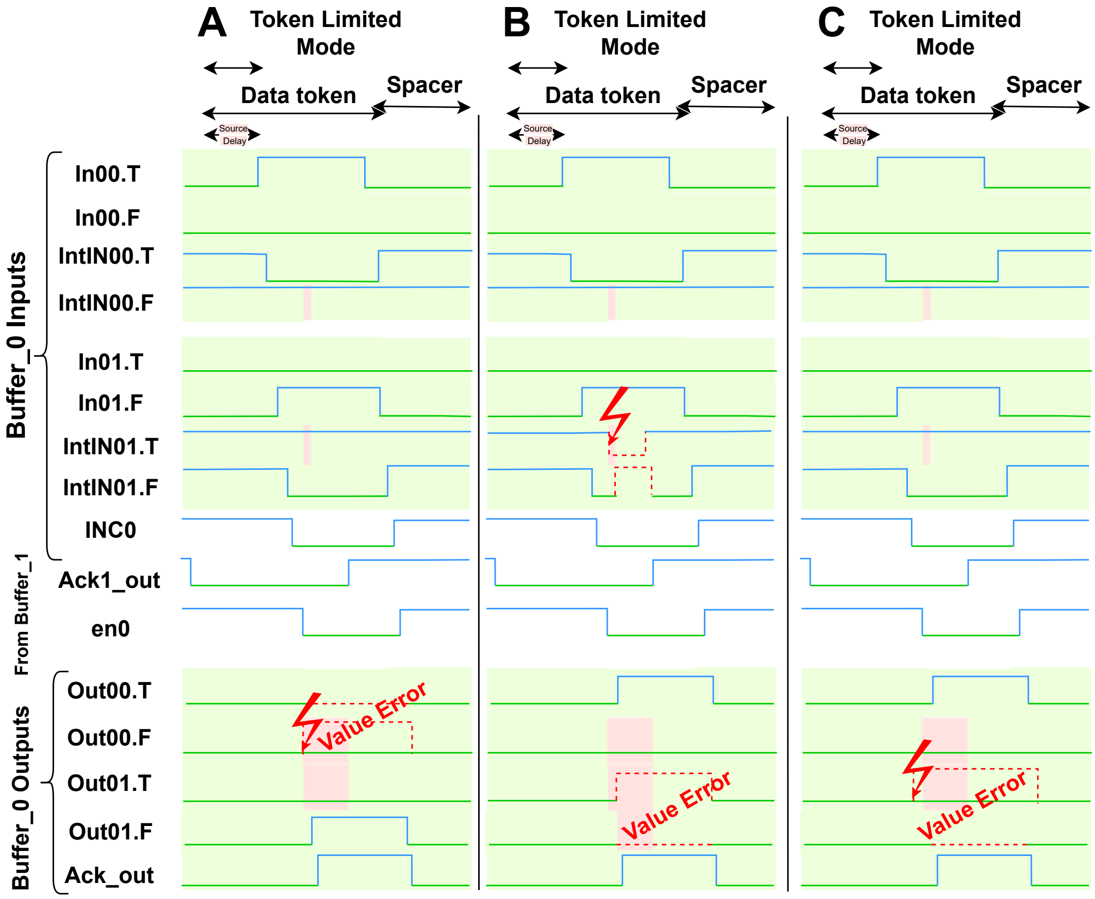


Figure 3.36: Sensitive Windows of Δ during Token Limited Mode

3.5.2 Main Causes of Errors during Balanced Operation

If we analyze the sensitive windows of the token limited mode in comparison with balanced operation from Figure 3.33 it is visible that their sensitive windows lie approximately on the same spot. However, when we go to the results in Figure 3.35 the circuit turns out to be more susceptible to SETs during balanced operation. The main reason behind the resilience in token limited mode is that the fault may vanish during the waiting time of the source. The very similar SET sensitivity of Δ in token limited and in balanced operation also made it hard to identify interesting new scenarios to show for balanced operation (PLF=1) here. We decided to analyze the results for fault length 154ps.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

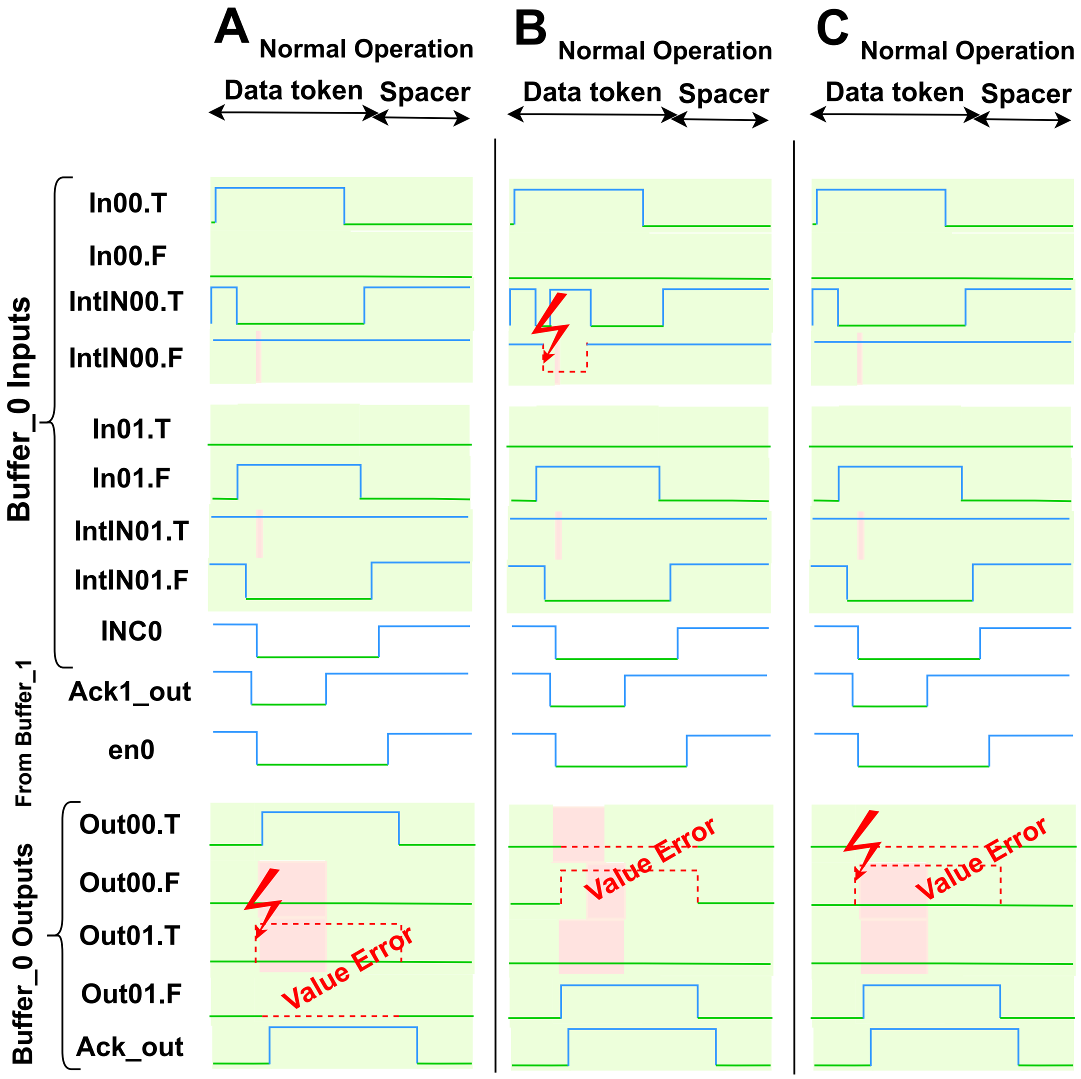


Figure 3.37: Sensitive Windows of Δ during balanced operation

1. Figure 3.37-A: a fault appears at *Out01.T* in close proximity of the falling edge on *en0*. As the respective MCE enters the storage mode there, the fault flips the stored logical value and nullifies the effect of the valid transition. As a result we get a value error.

2. Figure 3.37-B: a fault appears at *IntIN00.F*; this time it has negative polarity. It forces *IntIN00.T* that already performed the valid down-transition to go back to high. As fault covers the red shaded window of *IntIN00.F* and gets latched by C1, disabling the C0 for any further transition. After some time (length of the fault) *IntIN00.F* goes back to high because *In00.F* is low. This forces the NAND gate to

go high, and as a result *IntIN00.T* retains its correct value back as shown. But now it is worthless because C0 is blocked by the *Out00.F* high transition and we already propagated the value error.

3. Figure 3.37-C: as we were not able to find a different scenario than the ones presented in normal operation, we highlight a scenario that is, in essence, similar to Figure 3.37-A but the affected signal here is further downstream, namely *Out00.F*.

3.5.3 Main Causes of Errors during Bubble Limited Mode

Compared to the other two modes, the circuit is more susceptible to SETs in bubble limited mode: As shown in Figure 3.35 we observe more errors as compared to the IOISRWCHB Figure 3.27. For the maximum injection pulse length and with PLF=10, for instance, Δ shows 425 effects in comparison to 383 for IOISR. We investigated the main reasons and found a few new signals that were not so susceptible before our modification. These signals include the *enX* and *AckX_out*. A closer look to the Figure 3.35 extreme bubble limited mode tells us that there is higher number of value errors compared to the IOISRWCHB Figure 3.27 and lower number of deadlocks. And the main contributor behind this value error is *enX* and *AckX_out*. In all our previous discussions we did not consider these signals, because they are not a real contributor in those cases. Here, however, we have to carefully analyze their behavior – why are they sensitive? In Figure 3.38 we also showed the sensitivity of *en0* with red and green shades for the first time, but we were not able to explicitly identify the sensitivity of *Ack1_out*. The reason was that when looking back from *Buffer_0* we were not aware of the state of the input signals of MCE-C9. This information, however, is required to determine whether the MCE is in storage mode or combinational mode. In the following we will discuss the main reasons behind the generation and propagation of the faults.

1. Figure 3.38-A: the sensitivity of *en0* depends on *INC0* and *Ack1_out*. When these signals have different level, MCE-C11 is in storage mode and the shade must be red. This time the fault appears during the red shade, but at the same time our circuit is also sensitive in a way that this negative transition on *en0* allows the waiting token at the inputs of *Buffer_0* to pass to the output as shown. As a result, *Ack_out* goes high and the source removes the tokens from *In00.T* and *In01.F*. Consequently, *INC0* goes high again because all *IntIN.t and .F* go high, indicating a spacer. *Ack1_out* is still high, as the last passed spacer is not acknowledged yet by the next stage (*Buffer_1*). Now *INC0* and *Ack1_out* force *en0* to high. At this instance the effect of the SET has alrealy vanished. This transition at *en0* passes the new spacer at the output, and as the next stage is excepting the spacer, it responds by setting the acknowledgement signal low: as shown, *Ack_out* goes low after arbitrary time. But unfortunately, one token is lost due to the false transition at *en0* (SET hit). Now the token at the input of the *Buffer_0* is not the one that is supposed to pass, as *Out01.T* goes high in contrast to the expected one in which *Out01.F* was high. A value error is reported. In this scenario the seemingly

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

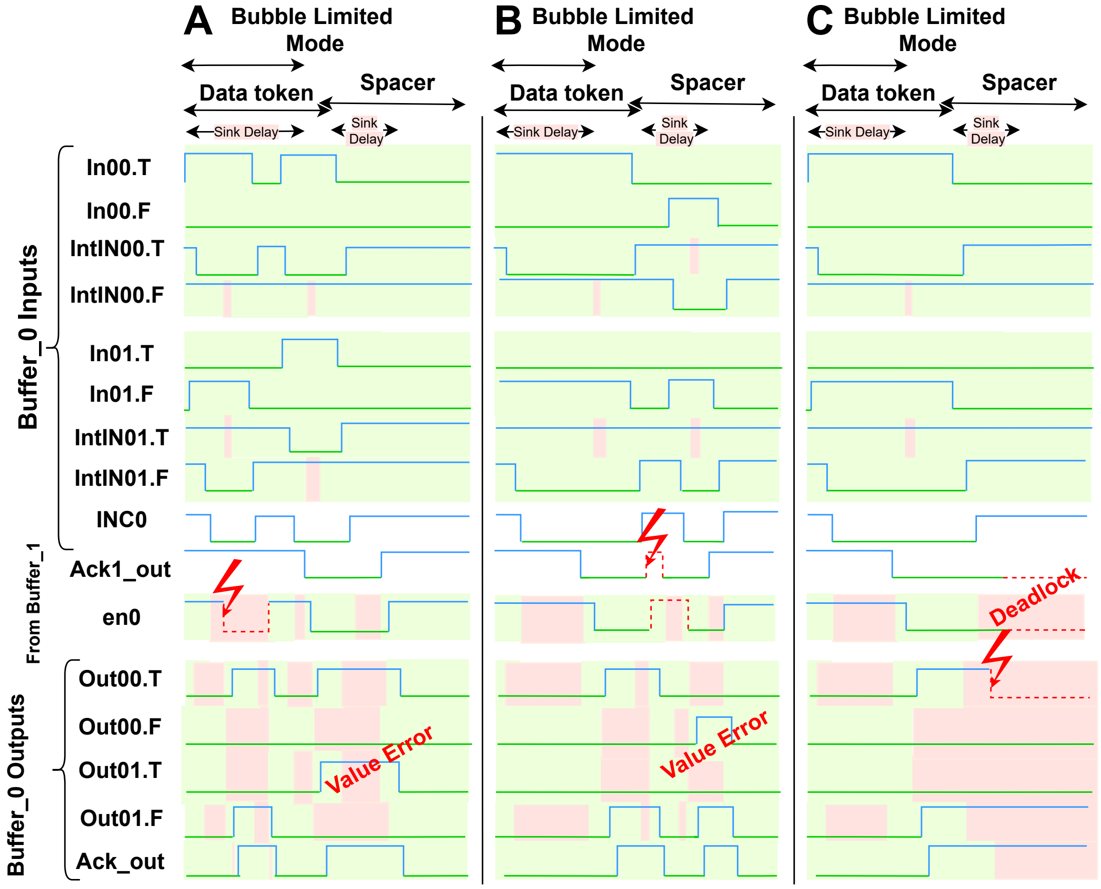


Figure 3.38: Sensitive Windows of Δ during Bubble Limited Mode

erroneous data token has indeed been sent by the source, but it is not the expected one, the presented token will be expected next in line by *Buffer_1*.

2. Figure 3.38-B: here *Ack1_out* goes low, which means the spacer is latched by the *Buffer_1* and a new token is requested. That token is placed and passed to the output of the *Buffer_0*. Next, as the *Ack_out* goes high, the source places a spacer to the inputs of *Buffer_0*. In consequence, all *IntINXX.T and .F* go high, which fires MCE-C10, and *INC0* goes high. Accidentally the fault appears at *Ack1_out* firing C11 when *en0* goes high. This is the indication that the next stage demands a spacer. The waiting spacer is now passed to the outputs which then also reset *Ack_out*, asking for a new token from the source. After some arbitrary time, the SET vanished from *Ack1_out*. At this point we are not aware whether MCE-C9 is in storage mode or combinational mode, but the fact that the effect will vanish confirms that C9 is in combinational mode. When all bits of the new token arrive at the input of *Buffer_0*, it will force *INC0* to zero, which then resets *en0*. This

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

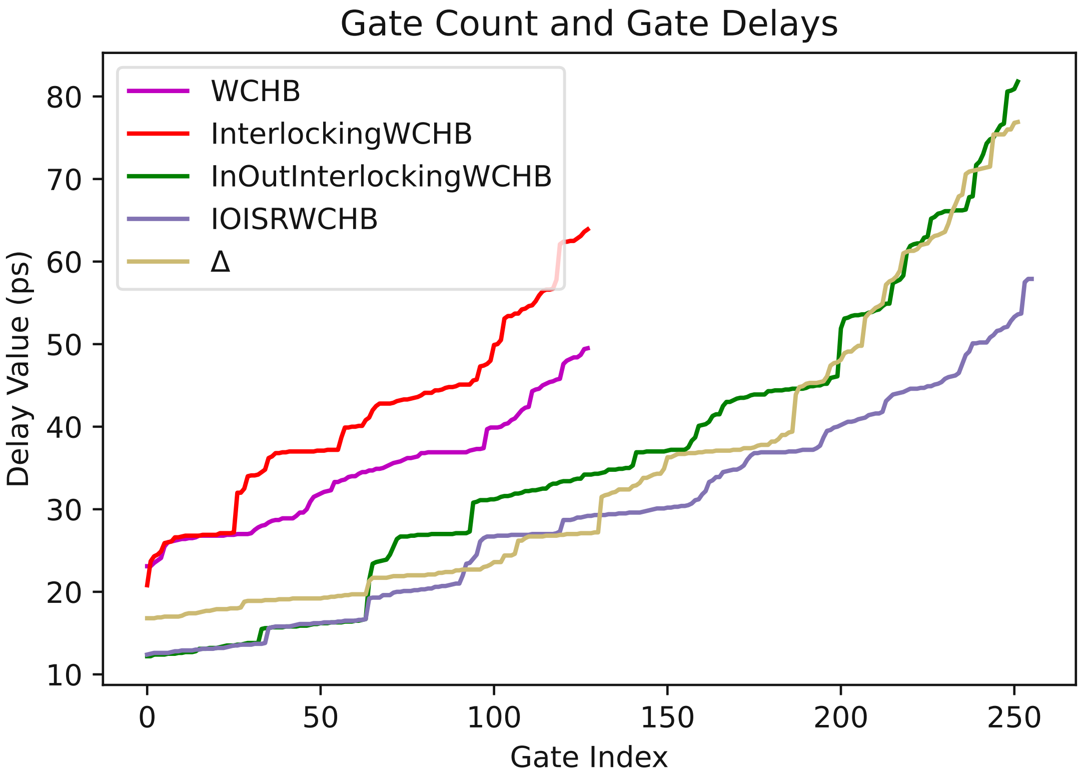


Figure 3.39: Gate Count and Gate Delays Comparison of all dicussed buffer templates

latter transition allows the new token to the output and is now considered as valid token by *Buffer_1* which is expecting a token. But as shown the presented token is different from the lost token due to the false statement of *Ack1_out*. The value error flag goes high.

3. Figure 3.38-C: a fault appears at the *Out00.T* when we are waiting for the acknowledgement from the next stage. As its polarity is negative it then forcefully removes the valid stored token from MCE-C0, which is in storage mode: its inputs *IntIN00.T* and *en0* are at high and lo, respectively. Our next stage remains in a waiting state for the unexpectedly removed token and we are not able to forward the handshake cycle, which then causes a deadlock.

3.5.4 Final Comments on Δ

With Figure 3.39, we can compare the area (gate count) of all discussed buffer templates, its clearly visible that the Δ comprise the same gate count as other resilient techniques, we achieve better resilience with same footprint. Where the Figure 3.40 presents the resilience comparison of all discussed buffer templates. It is visible that we achieve higher

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

resilience with Δ in token limited mode, while it is comparable with the alternatives in bubble limited mode. The reason that we are not able to show the difference in bubble limited mode is that our control signals – which includes *enable* and acknowledgement – become sensitive in that mode of operation. The dual completion detection adds an extra MCE to allow the acknowledgement signal to the buffer. That latter signal, however, then becomes susceptible in bubble limited mode as the acknowledgement is delayed while the source already toggles its input lines – as a result the MCE responsible for passing these signals remains in storage mode. But at the same time in token limited mode, as Figure 3.40 suggests, the error occurrence is near to zero, in contrast to IOISRWCHB where it is approximately 2.5% while maintaining the same gate count as the IOISRWCHB.

In [TS22a] we compared Δ with Dual completion detection versions of WCHB, Interlocking WCHB and InOutInterlock WCHB. Δ performs well compared to all these while saving 44% area (transistor count) by replacing the MCE used for input interlocking with an SR latch technique. Compared to its Dual Completion detection version of the InOutInterlock WCHB, Δ shows higher throughput, which is clearly due to the efficient and faster input interlocking.

The lesson learnt from the IOISRWCHB is implemented in Δ, which further minimizes the sensitive window in token limited and balanced operation.

We only considered the buffer templates in our analysis for this chapter which gives us good insights about the behavior of the buffer template alone. In the next chapter we continue our study by introducing combinational elements between these buffer stages, which is a step towards real-world circuits.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

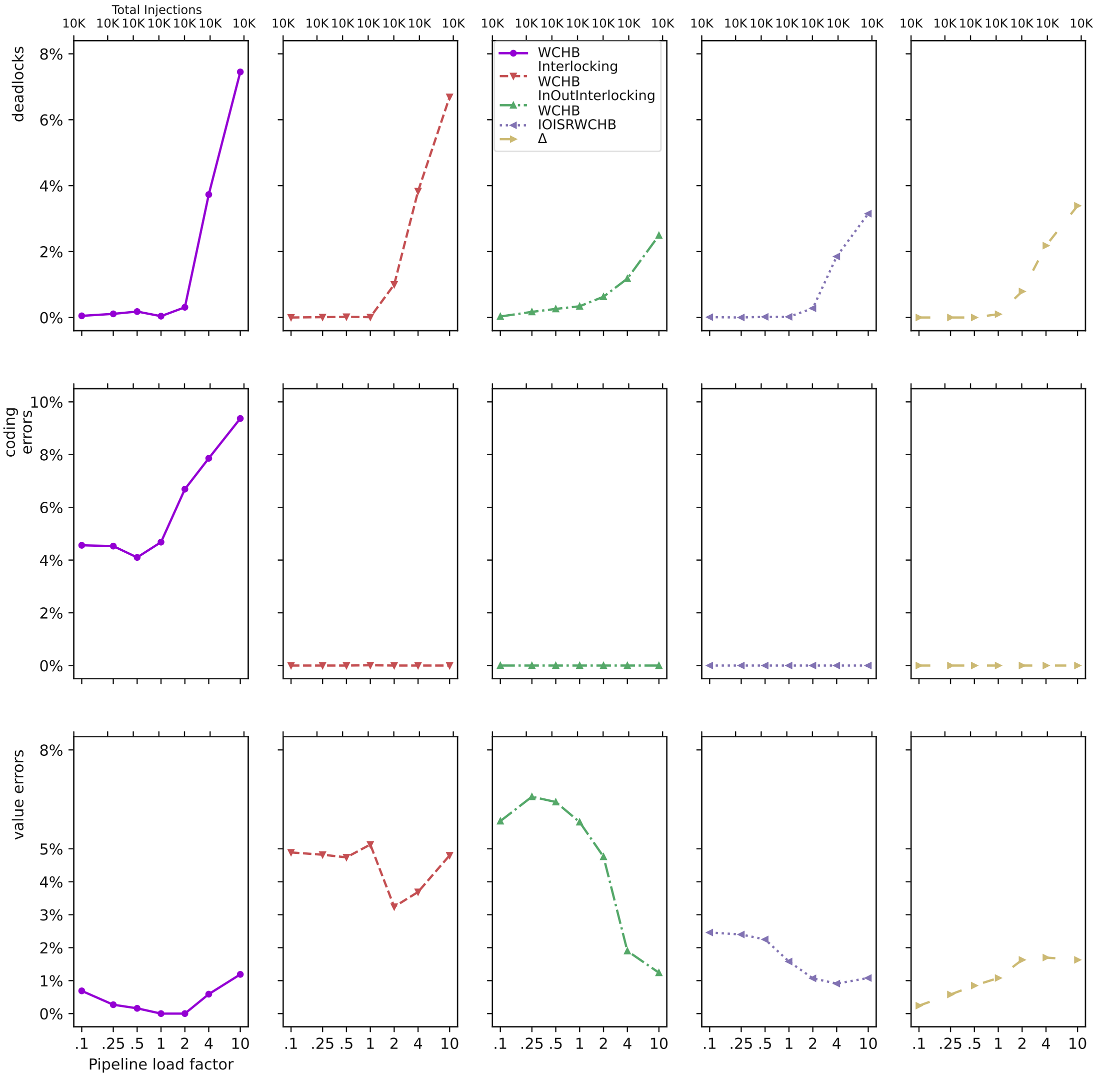


Figure 3.40: Δ comparison with all discussed buffer templates (Injection pulse is equal to 2*(maximum gate delay))

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

CHAPTER 4

The behavior of QDI Combinational Logic under the influence of SETs

So far, we analyzed the behavior of QDI buffer templates only, without any combinational logic included in the circuit. To extend our scope, we will specifically investigate the susceptibility of the combinational logic under the influence of SETs. As the QDI combinational logic is delay insensitive under the isochoric fork assumption, it contains MCEs. As we know, the MCE is the main contributor in converting SETs into SEUs, and it propagates SETs to the buffer template. In this section we will try to highlight the root cause behind both, the conversion and the propagation. We selected the 16-bit 7 stage pipelined multiplier circuit as our target circuit, the combinational logic style is DIMS [SS93] and [Spa20]. We start our investigation by segregating the error contributors from the circuit, which tells us how much the combinational logic is susceptible to SETs. Then we analyze the state-of-the-art techniques proposed to mitigate the effects of SETs in combinational logic. Lastly, we propose a respective novel technique. Our focus is to utilize the inherent behavior of QDI logic to make our circuit fault tolerant.

For the embedding into a pipeline we include the resilient buffer template named Δ from our previous findings [TS22a], but we also consider the baseline WCHB template for reference.

4.1 SET effects in DIMS Combinational Logic with a WCHB buffer template

[SS93] discussed the approach *Delay Insensitive Minterm Synthesis (DIMS)* to implement the combinational logic for delay insensitive circuits. As the name suggests, the circuit will

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

operate in a delay-insensitive manner; the minterms of the input variables are generated with MCEs to achieve the required behavior.

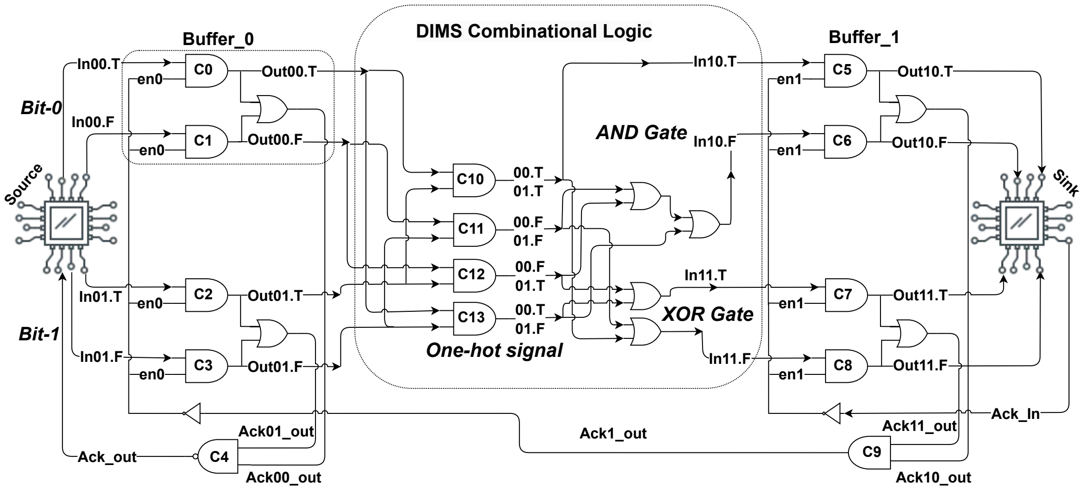


Figure 4.1: WCHB buffer with DIMS AND and XOR logic

Figure 4.1 is the same simple 2-bit, 2-stage circuit we used to explain the behavior of our buffer circuits, but here we add the combinational logic within the pipeline stages. The first stage of the DIMS logic transforms the binary input into a one-hot signal using MCEs as shown. Based on that one-hot signal we can implement any logic. In this case we implement AND and XOR gates. The AND gate's value will be stored as Bit-0 and the XOR's output as Bit-1 of *Buffer_1*. For the AND gate there is only one possibility that the output goes high, namely when both *.T* rails of the last buffer bits show a high transition, so we can connect the *00.T and 01.T* one-hot signal directly to *In10.T*. For all other combinations *In10.F* goes high and this will be checked by OR-ing all other one-hot signals. The XOR gate's *In11.T* will only go high if Bit-0 and Bit-1 are different so *00.F,01.T* and *00.T,01.F* are checked by an OR gate, while the *00.T,01.T* and *00.F,01.F* are checking whether the bits are same and then passed to the *In11.F* for the generation of the false-bit.

To get a first idea of the normal operation of that combinational logic and the effects when an SET strikes it, we explain both behaviors in the following discussion with the help of Figure 4.2. We use the Figure 4.1 for reference.

1. Figure 4.2-Fault Free: we already know the behavior of the WCHB from the last chapter, so here we focus on the propagation of the data token through the combinational logic. When the data token arrives at *Out00.T and Out01.F*, transitions labelled with "1" toggle *Ack_out* as an acknowledgment to the source, and one-hot signal *00.T, 01.F* to high $(1*)$. As a result, the AND gate must show a logical 0, as one of the input bits to the combinational logic is low (*Out01.F*). Consequently, *In10.F* shows a transition labelled with $1**$. The input bits are

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

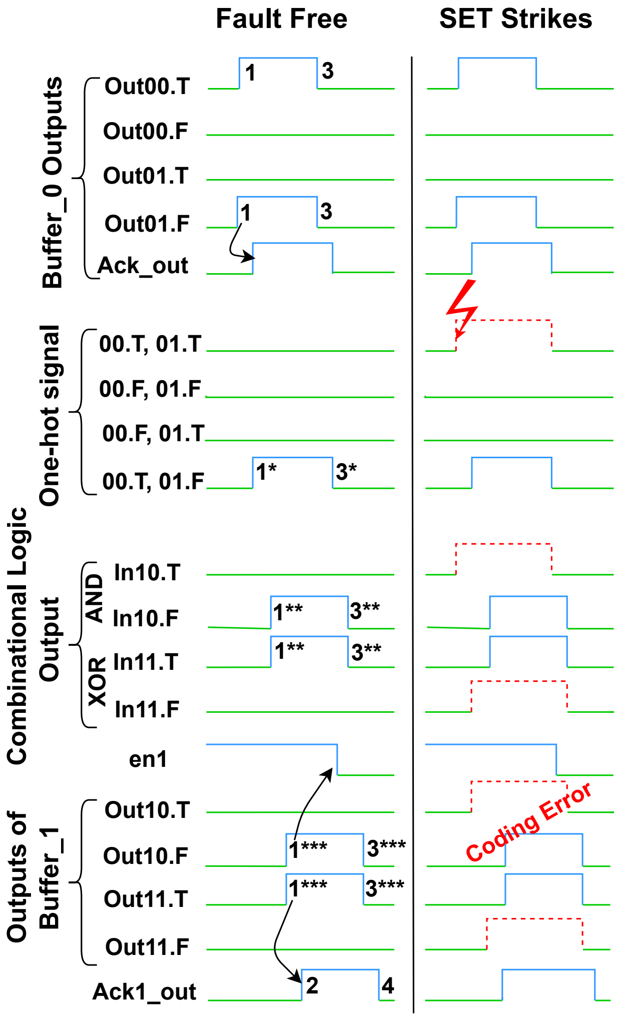


Figure 4.2: SET effects in combinational logic without any protection deployed

different, so the XOR generates a logical 1 output, as shown *In11.T* goes high (label $1**$). *en1* high means *Buffer_1* is ready to latch the new data token. These values are reflected at *Out10.F and Out11.T* marked with $1***$. These transitions toggle *Ack1_out* marked with 2, an acknowledgement signal to *Buffer_0*. When the data token is latched by the sink, *en1* goes low arming *Buffer_0* for the bubble.

2. Figure 4.2-SET Strikes: a fault appeared at the *00.T, 01.T* one-hot signal during a time window when the valid transition is expected on one of these lines. As MCE-C10 is in storage mode and the SET pulse is long enough, the value stored in C10 is flipped. At the same time the valid value also shows up at the *00.T, 01.F* one-hot signal as expected because of the input signals to the combinational logic. As the *In10.T* is the same signal as *00.T, 01.T*, this fault appears at the input

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

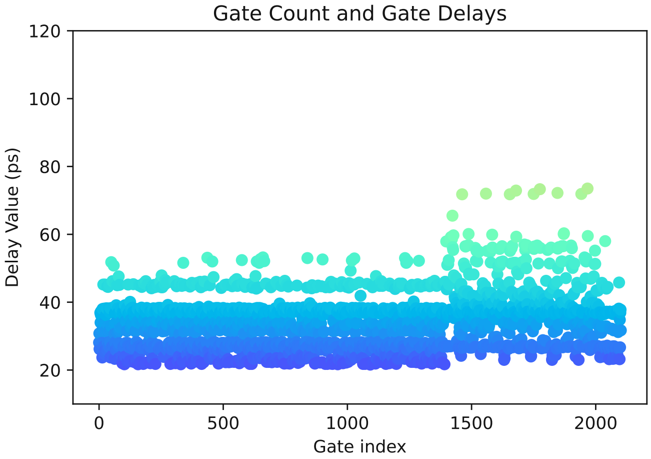


Figure 4.3: 16-bit 7 stage Pipelined multiplier with WCHB buffer template Gate Count (2102) and Gate Delays

> of *Buffer_1* without any propagation delay. There it also reaches *In11.F* before the valid transition propagtes to *In10.F and In11.T*. As *en1* is high, all transitions pass to the output without any further checks. This generates illegal code words on both bits as shown.

This example provided some insight how SETs threaten the data integrity and flip multiple rails that may propagate to the next stages and cause errors. Here we start investigating the fault-tolerance behavior for our target circuit, a 16-bit 7 stage pipelined multiplier Section 2.4 where the combinational logic is DIMS and the buffer template is a WCHB. Figure 4.3 presents the gate count and gate delays of the target circuit. The gate count is 2102 where the minimum propagation delay of a gate contained in the target circuit is 21ps, with a maximum of 73ps.

Figure 4.4 presents the susceptibility analysis of our target circuit under the influence of SETs. It presents the number of times the circuit failed to mask the effects of the SET with specific sub-counts for how much each part of the circuit is susceptible. On the right-hand side, we give the total number of injections for each presented scenario. The number of injections we perform is chosen dependent on the total number of nodes in the circuit and the time the circuit requires to process some specified number of tokens. In extreme token and bubble limited modes, due to extra waiting time, the injection count increases as shown; e.g. for PLF = 10 the injection count is 177K compared to 31K for PLF = 1.

We start our experimental analysis with injection pulse length 21ps, and by crossing the 73ps (maximum gate delay) we inject 1 and 2ns pulses that are safely long enough to test the circuit resilience towards transients. For all injection pulse lengths the behavior of the combinational logic is important. The results of Figure 4.4 show that the combinational

TU WIEN Bibliothek Your knowledge hub

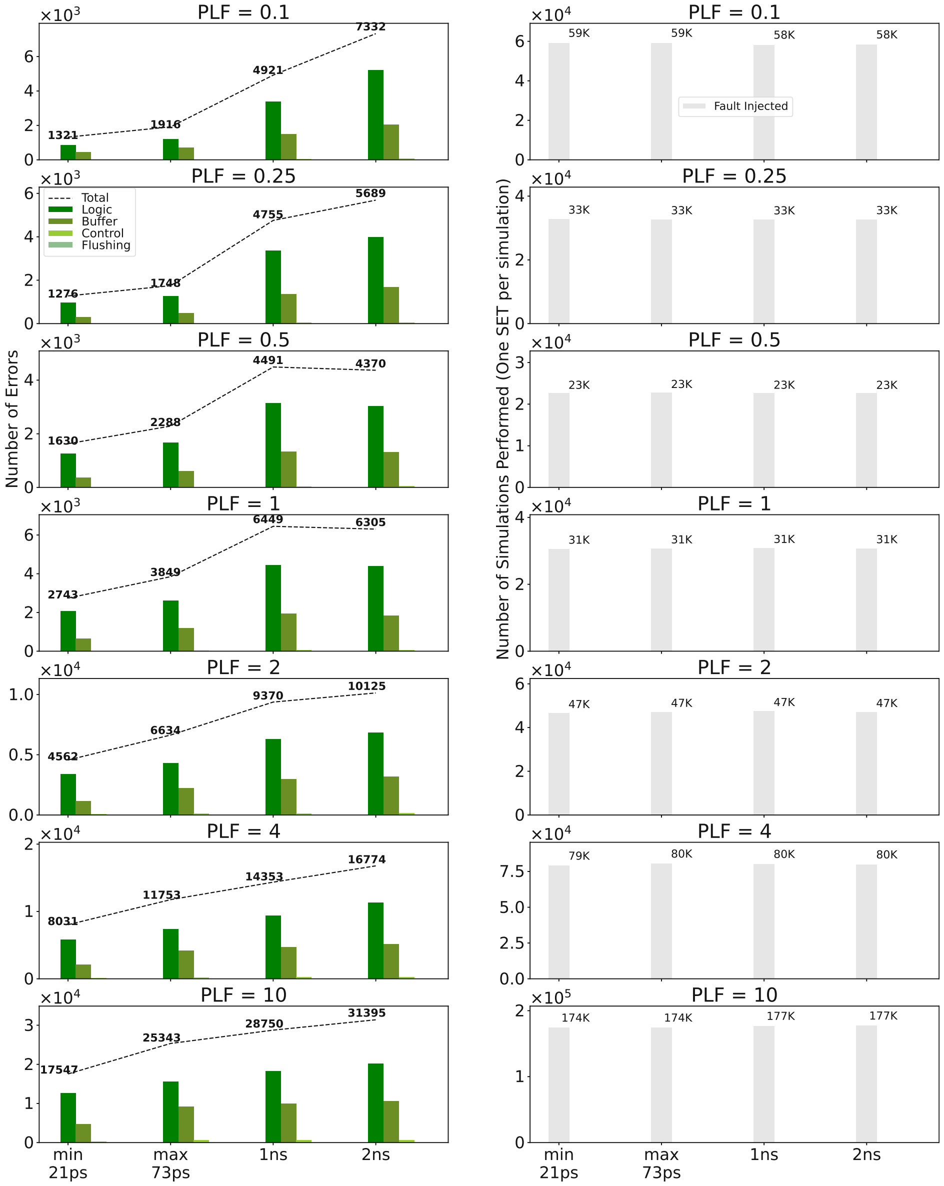


Figure 4.4: Suseptibility analyses of Pipelined multiplier with WCHB buffer template under the influence of SET with variable PLF and different injection pulse lengths

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

logic always contributes twice as much as the buffer template. In the last chapter our focus was the buffer template, and we proposed a technique to harden it to mitigate the effects of SET directly hitting its internal signals or its inputs and outputs. Most of the QDI hardening techniques for SETs (excluding modular redundant ones) focus on the buffer templates as these are the data latching elements in a pipeline circuit. However, as the QDI combinational part also contains MCEs, it is important to take this part into consideration as well. Our results now clearly indicate that the error rate of a basic QDI WCHB pipeline template receives a significant share from the combinational logic part. Considering that combinational logic is often the dominant part (area-wise) in a circuit, this is not surprising and may even become more pronounced for more complex logic. Fortunately, some inherent properties of QDI circuits help us minimize these effects. In the course of our investigations we found a couple of techniques utilizing these inherent QDI properties to mitigate SET effects in combinational logic. From the next section onward we will discuss a few, with most promising results.

4.2 The SE Tolerance Asynchronous Pipeline I approach for flushing SEUs from DIMS combinational logic with a WCHB buffer template

[KZYD10] presents a technique called Built in Soft Error Correction (BISEC) that flushes the whole combinational logic upon detection of an error. We will refer to this approach as SE Tolerance Asynchronous Pipeline I (SETAPI) in the following. With dual-rail 4-phase logic, only one rail can go high per handshake cycle, so a (single) erroneous rail can be easily detected by ANDing the rails of each bit. A logical one at the AND output means there is an illegal code word. In SETAPI, as presented in Figure 4.5, with specific extensions highlighted in orange color, first an SE detection circuit is placed at the output of the combinational logic which detects the illegal code word. Secondly, AND gates are inserted into every rail before feeding it to the combinational logic. These are used to flush (force to low) the whole combinational logic if the q goes low. The signal q only goes low if a fault is detected and the extra buffer (Buffer_E) also confirms the data latching by the *Req* signal. This simply arms the reset circuit, which then forces all AND gates to go low. The Delay block (difference of combinational and contamination delay of that stage) is used before the *Ack_E_Inv* to delay the extra buffer's completion signal. If SE detect did not flag any erroneous value before the arrival of *Ack_E_Inv*, it nullifies the SE Detect unit's output, as *Ack_E_Inv* = 1 forces the *Error* signal to "0". SETAPI conveys an error signal along with invalid data and moves the responsibility for its appropriate use for error handling to its receiver.

Using Figure 4.5 as reference we will study Figure 4.6 to understand the normal operation of the SETAPI approach, as well as what happens if a fault hits any rail:

1. Figure 4.6-Fault Free: as the logic is the same as that of Figure 4.1, the data propagation is essentially the same. The only difference is the extra buffer (Buffer_E).

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

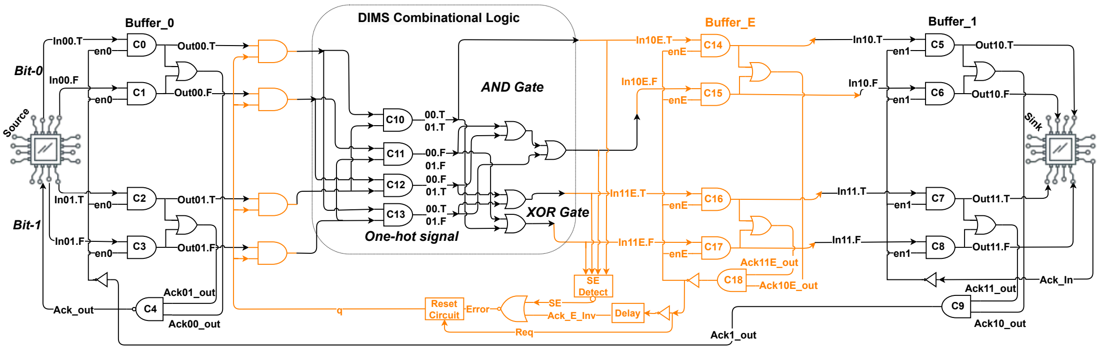


Figure 4.5: SETAPI approach with DIMS AND and XOR logic

The outputs of the combinational logic are first passed to the Buffer_E and then to the output buffer, where *enE*, an enable signal of Buffer_E, is generated by the completion detection of the same buffer. A transition on rails *In10.F and In11.T* will toggle *enE* to low, arming the stage for a spacer (highlighted with arrow from *1*** to *enE*).

2. Figure 4.6-SET Strikes: an SET appears at the *00.T, 01.T* rail. Since *In10E.T* is the same node, the fault also propagates to *In11E.F*. Together with these faulty transitions, the valid transitions also arrive at the output of the combinational logic, forcing all rails to high. This illegal codeword activates the SE detect, and when *Req* (check Figure 4.5) goes high, which is the completion detection signal of Buffer_E, *q* is reset which then flushes whole combinational logic part. Observe in the figure how all rails *In10E.T, In10E.F, In11E.T, In11E.F* are forcefully reset to zero. However, as faulty data has already been provided to the output buffer, it gets latched before the flushing of combinational logic, so a coding error is still generated and propagated to the sink.

With this discussion we conclude that the SETAPI approach is flushing the combinational logic but as the fault already propagated, this flushing is worthless. For the evaluation of the approach, we again first extract the gate delays and gate count presented in Figure 4.7. Due to the extra buffer per stage and the flushing logic the gate count goes from 2102(with no flushing circuitry and single buffer per stage Figure 4.3) to 3389. The maximum gate delays a circuit contain is 116ps and the minimum is 12ps.

We simulate the circuit with 7 different PLF's presented in Figure 4.8. These are the same as our last discussed target circuits. For PLF > 1 we (more or less) observe the intended resilience of the logic part against SETs. This is because in this bubble-limited mode the acknowledgment is delayed, and so the SET is blocked from propagating deeper into the pipeline, and the correction mechanism can flush and correct the current pipeline state. For understanding this point, consider the scenario presented in Figure 4.6-SET Strike part, but if *en1* is delayed the faulty transitions are not passed to the sink and meither is *Ack1_out* generated. So the valid value to the combinational logic input

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

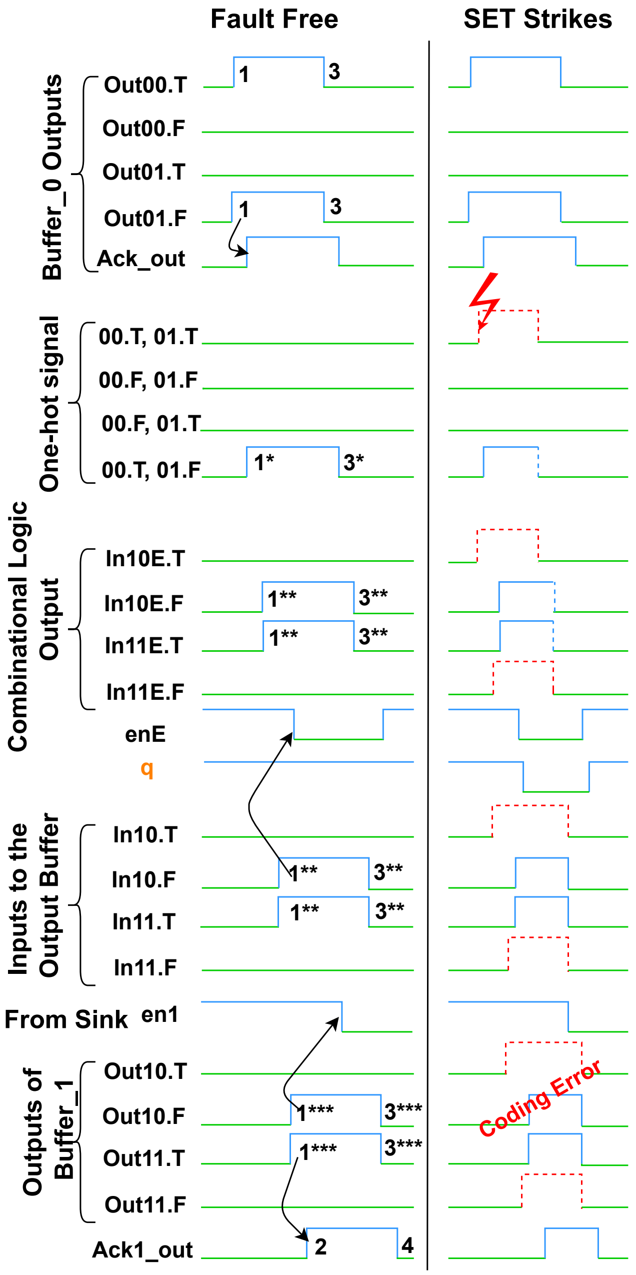


Figure 4.6: SET effects from combinational logic with SETAPI approach

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

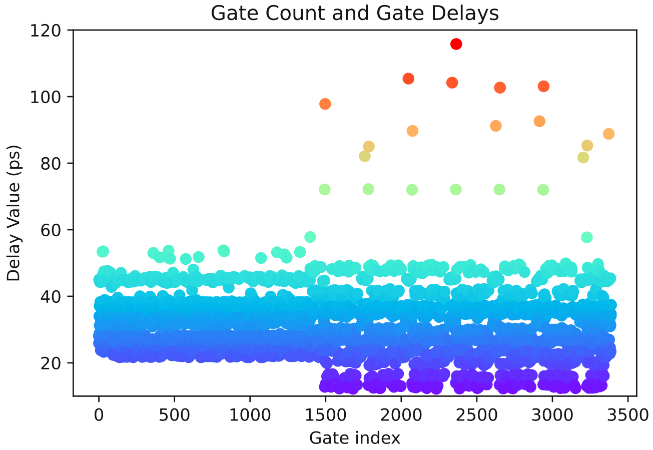


Figure 4.7: 16-bit 7 stage Pipelined multiplier with WCHB buffer template and SETAPI approach Gate Count (3389) and Gate Delays

(Buffer_0 output) remains there, while during this waiting time the fault can be flushed, and the valid values are calculated again. This is clearly visible from the results, where in bubble limited mode the contribution from combinational logic is low in comparison to the buffer and also when we compare this to the Figure 4.4. On the other hand, the buffer contribution is considerably higher compared to the baseline approach because SETAPI has one extra buffer per stage, which results in a higher overall injection count (reflecting the fact that its larger area is also more likely to be victim of an SET). Another important thing can be noticed from the result of extreme token limited mode PLF = 0.1 and 0.25 is that we also observe errors due to the flushing logic itself. If the whole combinational logic is accidentally flushed by an SET on the flushing logic, it may generate the spacer before the source generates it and as the sink or next stage is already armed, it may be accepted. This advances the handshake from the sink side without any valid response from the source side, which generates errors. With the contribution of the flushing logic in error generation the number of errors doubles, see Figure 4.4: for injecting 1ns pulses in PLF=0.1 the errors are 4921, while with the same selection in Figure 4.8 it is 10976. The flushing bar from the same selection suggests that the contribution is not negligible.

4.2.1 Final Comments on SETAPI

Figure 4.9 presents the comparison between the unprotected 16-bit 7 stage pipelined multiplier with the SETAPI approach realized with the WCHB buffer template. The first thing we observe from this figure is the number of injections presented on the top x-axis: due to the extra buffer and the flushing logic the number of injections is approximately doubled compared to the simple WCHB-based multiplier. We do not observe any glitches as we delete the last stage's enable signal from the injection list that is responsible for the glitches discussed in the last chapter. In the following discussion using Figure 4.9 we

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

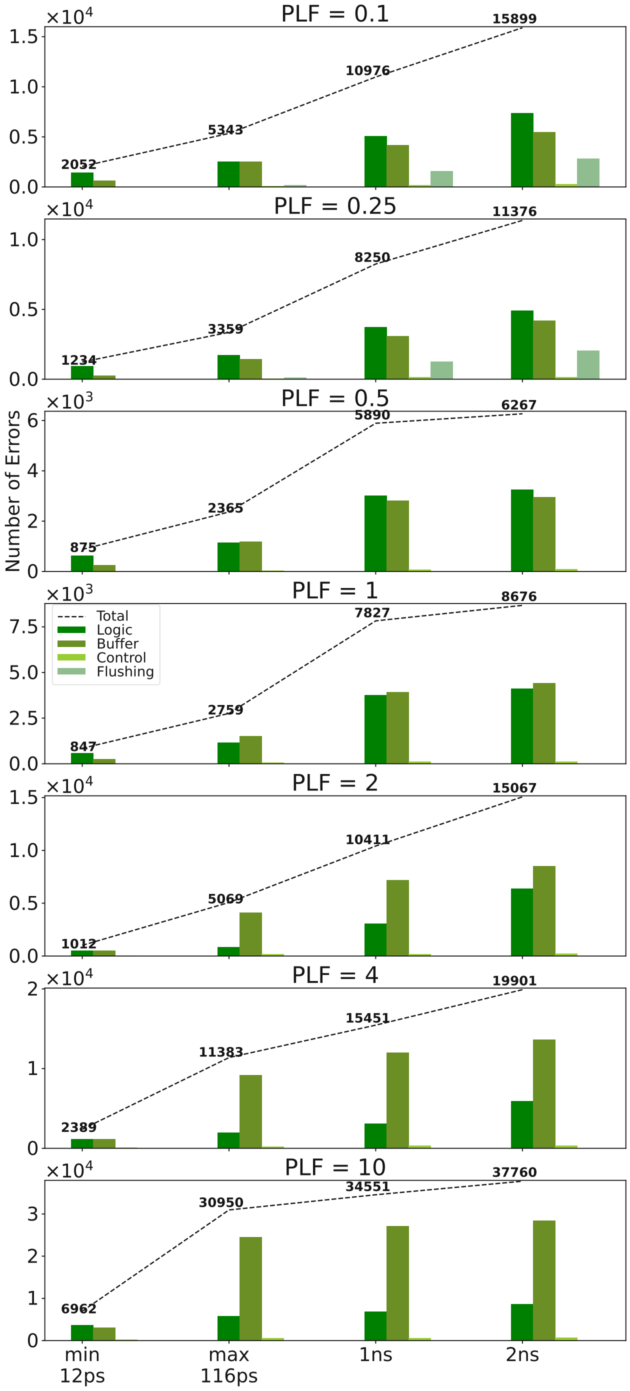


Figure 4.8: Suseptibility analyses of Pipelined multiplier with WCHB buffer template and SETAPI approach under the influence of SET with variable PLF and different injection pulse lengths

discuss briefly the main causes behind errors in all modes of operation.

1. In extreme token limited mode, as the buffer is already armed for the token or spacer, the transient is accepted as valid transition and causes a value or coding error as Figure 4.9-SETAPI concludes.

2. Compared to the WCHB for PLF = 0.1 we now also observed value errors where for PLF= 0.25 the circuit experiences more deadlock situations, and for PLF= 1 we observed fewer deadlocks and more coding errors.

3. For PLF = 0.1, due to an SET in between *Buffer_1 and Buffer_0*, “q” goes low, a spacer is generated (due to the flushing mechanism) and passed to *Buffer_1*. After being latched it generates Ack_1 = 0 (en0 = 1), but as the source is slow (the last data token is still at the inputs) we are able to recover the flushed data token as *Buffer_0* is still holding it. This generates the same data token for *Buffer_1* but it is now considered as new data token. *Buffer_1* toggles its Ack_1 = 1 (en0 = 0), which arms *Buffer_0* for a spacer. As we passed the same token two times the value error flag is raised, but the circuit does not enter into a deadlock.

4. For PLF = 0.25 the source responds faster than with PLF = 0.1, so the circuit may enter the deadlock situation easily due to the flushing mechanism. Consider in Figure 4.5 that due to an SET a spacer is produced (as the flushing mechanism does in case of SET) and passed to *Buffer_1* while holding a valid data token. Further assume that the source is near to respond with a spacer to the acknowledgement of that data token passed to *Buffer_1* by *Buffer_0*. The spacer due to the SET now latched by *Buffer_1* generates a bubble (Ack_1 = 0 makes en0 = 1) that is not expected as response before the actual generation of the spacer through the source. The latter then stops the valid spacer that arrives just now at the inputs of *Buffer_0*. We are not able to proceed further and get stuck in a deadlock. This is the main reason why we experience more deadlocks for PLF = 0.25.

5. Moving to the balanced mode (PLF = 1) we experience more coding errors than deadlocks. The reason is that the WCHB buffer template is not able to mitigate the SET effects. All transitions are passed to the output without any check and on the detection of the SET the flushing mechanism flushes the whole logic, generating a spacer. As the circuit is fast enough, the generated spacer is merged with the valid spacer by the source and we thus somehow manage to not enter into the deadlock situation.

6. For bubble limited mode again our circuit experiences deadlocks, but the prominent factor here is not the combinational logic, as Figure 4.8 suggests. Most errors are due to the buffers operating in bubble limited mode.

7. Consider the circuit Figure 4.5. *Buffer_1* passed a data token to the sink and as it is in bubble limited mode, we are waiting for en1 to go low. When *Buffer_1* passed

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

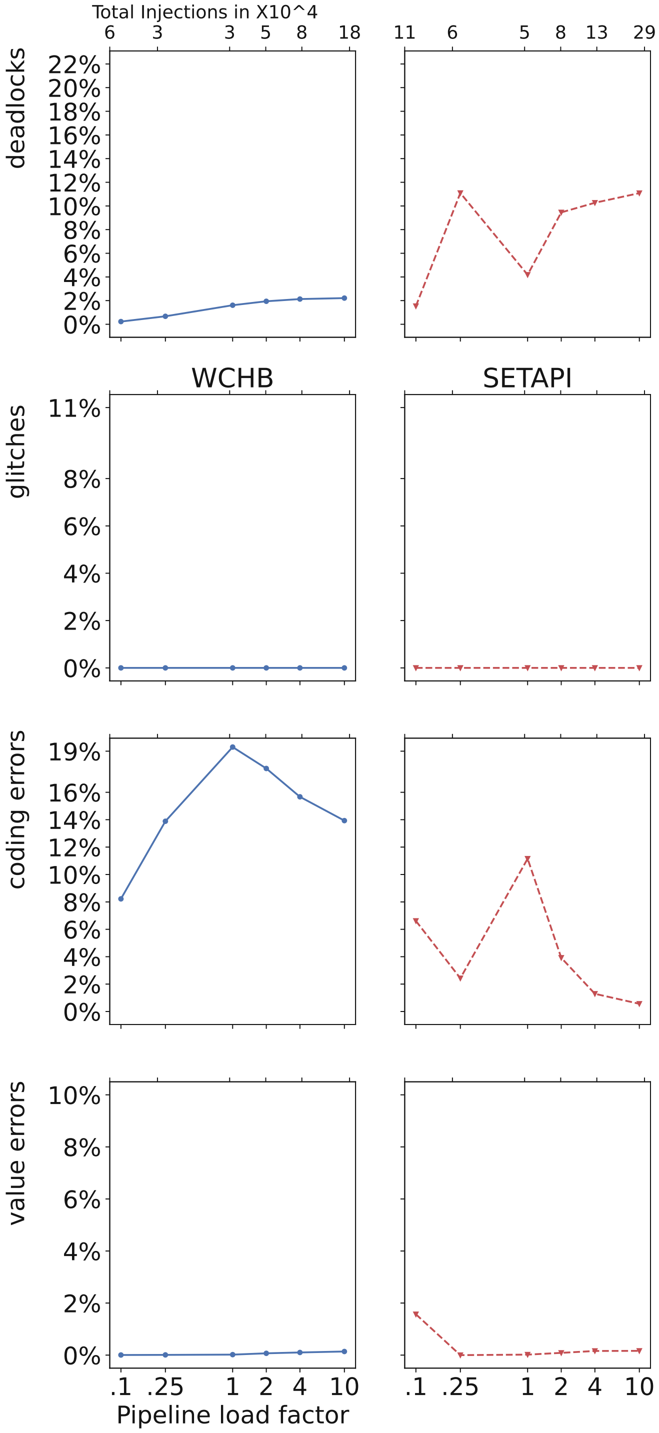


Figure 4.9: SETAPI comparison with WCHB based Pipelined Multiplier (Injection pulse is 1ns)

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

the data token to the sink, it also sends the acknowledgment signal to *Buffer_0* which then passes a spacer from the source to *Buffer_1*. All inputs to *Buffer_1* are now zero. In bubble limited mode *Buffer_E* already changes its enable signal state to “1”, putting MCE-C14 in storage mode. If a fault hits *In10.T* in this state, when en1 is high C5 also fires, regardless of the value of C6. After some arbitrary time the sink responds to the sent token, which makes *en1* low. So all rails except *Out10.T* go low, and C14 is holding a faulty transition that we are not able to flush. Our flushing is neither able to detect the SET as it is out of its scope. We end up in a deadlock situation.

The presented argument does not cover all scenarios but gives us an insight why there are more deadlocks in extreme bubble limited mode.

After this discussion we can conclude that SETAPI is not effective again SETs because of the extra buffer it introduces, and because of the flushing of the whole combinational logic that it performs without any knowledge of the environment, which causes a deadlock or advances the handshake cycle without any transition from the inputs – which sometimes violates the causal behavior of the QDI circuit. In the next section we discuss a successor approach of the SETAPI called SE Tolerance Asynchronous Pipeline II (SETAPII).

4.3 The SE Tolerance Asynchronous Pipeline II approach for mitigating SET effects in DIMS combinational logic with a WCHB buffer template

The flushing is meaningless if the data is already delivered to the next stage, and it advances the handshake cycle. This limitation of SETAPI (BISEC [KZYD10]) is already discussed by [LHHA12], and in a later paper the authors present a modified version named SETAPII [LHHA16]. They proposed some enhancements highlighted with orange color in Figure 4.10 . They basically modify the SE Detect unit to check the current state of the circuit as well and combine it to generate an SE signal that is now also capable of resetting *Buffer_E and Buffer_1* on the detection of the SET. So, if the faulty data is already latched by the *Buffer_E and Buffer_1* it will be flushed out on the detection of the SET. The handshake phase detection is derived from the output of *Buffer_0*. More specifically, rails *Out00.T, .F and Out01.T,.F* are passed to the *State Detection* and, after some delay (AND gate + combinational logic delay), fed to the SE detect unit. If the SE detects any illegal code word, it only generates the SE signal to flush the *Buffer_E and 1* if a data phase is detected. In their circuit diagram the authors of [LHHA12] connect SE to the reset of the extra buffer only, while text of the article suggests that SE is connected to both extra and regular buffer. In [LHHA16] they connect SE to the regular buffer reset signal only. Following the text of [LHHA12] we connect SE to both registers again in our evaluations.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

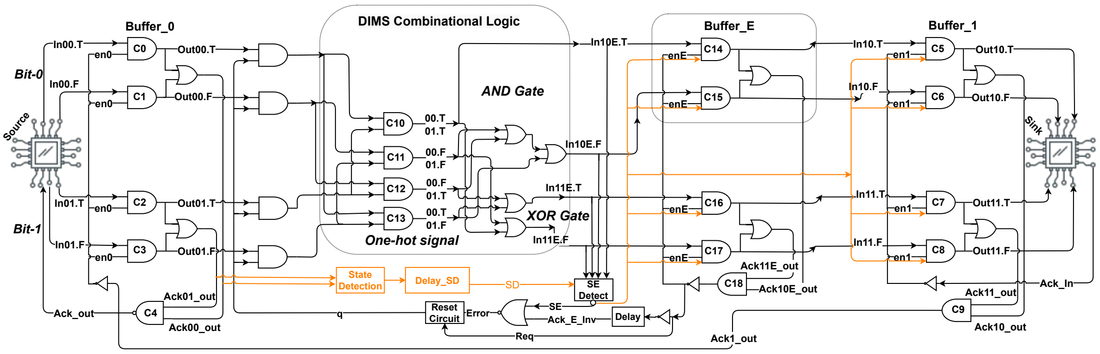


Figure 4.10: SETAPII approach with DIMS AND and XOR logic

Using Figure 4.11, we discuss the normal operation of the SETAPII and study what happens when an SET hits any of its combinational logic signals. The reference circuit is given in Figure 4.10.

1. Figure 4.11-Fault Free: normal data propagation is approximately the same as in SETAPI, the extra signal is *SD* (already discussed: it is a state detection signal). *SD* toggles aprroximately at the same time when the combinational logic generates its output; as shown, when *In10E.F and In11E.T* go high *SD* also goes to 1. We also add *SE* in our figure this time because it is important to tell when we must reset *Buffer_E and Buffer_1*.

2. Figure 4.11-SET Strikes: a fault appears at *00.T, 01.T* near to the valid transition at *00.T, 01.F*. As *In10E.T* and *00.T, 01.T* are the same, the fault is also visible on *In10E.T*, from where it is also propagated to *In11E.F*. The valid transitions are propagated to the inputs of *Buffer_0*, as shown on *In10.F, In11.T*, before the fault makes its way, while the valid transitions are also passed to the outputs *Out10.F and Out11.T*. Due to transitions on *Out10.F and Out11.T* the *Ack1_out* is removed and as a response a spacer is generated by *Buffer_0* as indicated by *Out00.T, Out01.F* showing transitions to zero. Due to the SET, SE goes low and resets both *Buffer_E and Buffer_1* before the sink responds via *en1*; data is removed from *Out10.F and Out11.T*. When *q* goes low on the detection of the SET it flushes the whole combinational logic. Unfortunately, we are not able to recover the correct state after flushing, as the subsequent spacer reset all inputs to the combinational logic. The resetting of *Buffer_E and Buffer_1* by SE also generates a bubble for *Buffer_0* which then comes with a new data token. The sink does not participate in the last handshake activity as this activity is falsely completed by the internal mechanism of SETAPII. The new data token from *Buffer_0* at (Out00.F, Out01.F) is now passed through the combinational logic and appeares at (Out10.F, Out11.F). It is considered as valid data token by the sink but as the last data token is lost it is not the expected data value, which raises the value error flag.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

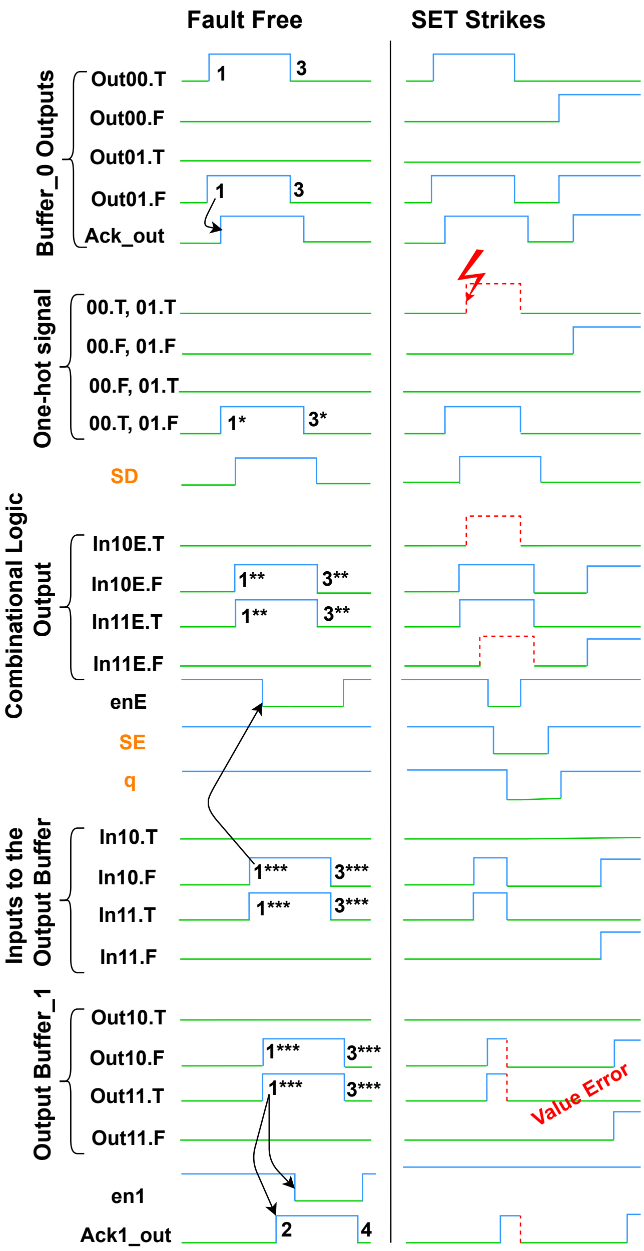


Figure 4.11: SET effects from combinational logic with SETAPII approach

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

A closer examination, however, reveals, that the proposed modification introduces a glitch problem because of the different delays (skew) of data rails. To better understand, we open the SE detect from [LHHA12, LHHA16] and present it in Figure 4.12. The AND gates connected at the output of the combinational logic are for the illegal code word detection, ultimately producing output *D*. Data completion is checked per bit using an OR gate from the output of the combinational logic as shown marked with *A and B*. The SD signal is combined with these to check whether we are in data phase or null phase. If *A and B* are low as well as SD, this means a spacer is provided by *Buffer_0*. In his case P0 is set high which then sets F (which combines P0 with D) to allow SE to go high.

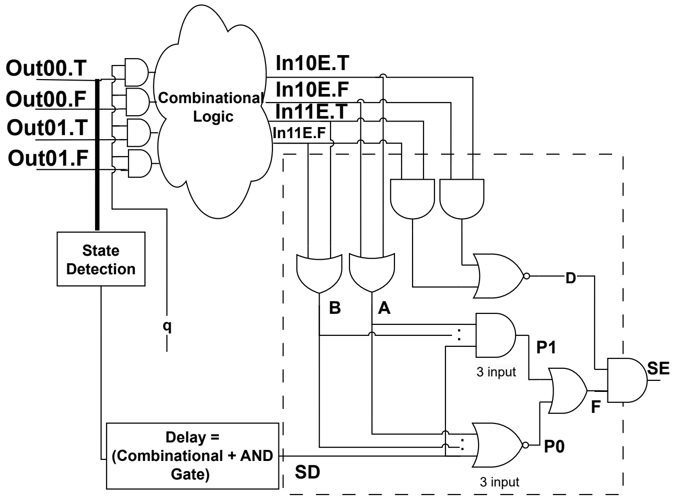


Figure 4.12: SETAPII approach Flush logic

In Figure 4.13-Internal Glitch we explain how these signal produce glitches at the SE signal which may cause an erroneous value even without any SET strike. For better understanding we refer to Figure 4.12 for the signals highlighted in orange color in Figure 4.13-Internal Glitch.

1. As each combinational logic part has different propagation delay, *In10E.F* may go high a bit earlier than *In11E.T*.
2. *A* goes high, which then forces P0 to low. P0 then resets F as P1 is already low (B and SD is still zero).
3. *F* forces SE to low, which then resets *Buffer_E and Buffer_1*. As shown in the figure, *In10.F* and *Out10.F* are reset without any transitions on the input signals *In10E.F*.
4. SE goes back to “1” when the second bit *In11E.T* is completed by the combinational logic and SD also arrives (after the added delay). As a consequence, P1 goes to high, which then makes F high and as a result SE goes high, too.

TU Bibliothek, WIEN Your knowledge hub

5. As we did not advance the handshake cycle, *Buffer_E and Buffer_1* again latch the provided values from the combinational logic and *Ack1_out* goes high.

6. This acknowledgement removes the data from the output of *Buffer_0*, namely *Out00.T* and *Out01.F*, which again produces a glitch at SE in the same way as we discussed for the positive transitions.

7. This glitch resets again both, *Buffer_E and Buffer_1*, where a closer look tells us that the data token is removed before *en1* shows any transition. This means the sink did not respond to the lost data token.

8. The forced flushing also generates a bubble for *Buffer_0* when *Ack1_out* goes low, as highlighted with the red dotted line.

9. *Ack1_out* generates a new data token (at *Buffer_0*'s outputs), which propagates to the output of *Buffer_1*. But luckily this time the propagation delays of bit 0 and 1 are approximately equal so the glitch on SE is small and not long enough to reset *Buffer_E and Buffer_1*. But as the values differ from the last data token (the lost one), a value error is generated, because the sink is expecteing a high on *Out10.F and Out11.T*.

Just like in Figure 4.11 here again we present a scenario without glitch effects. In the following discussion we explain, with the aid of Figure 4.13-SET Strike, how SET effects can be flushed by SETAPII without producing any error.

1. A fault appears at *00.T, 01.T* very near to the valid transition at *00.T, 01.F*. When the illegal code word is detected by the SE detect unit with SD being high, it will toggle its SE signal to low. At the point where SE goes low faulty transitions are only latched by *Buffer_E*. SE resets *Buffer_E* as well the "q" signal.

2. When the combinational logic is flushed on the resetting of "q" the traces of the SET will be also be flushed out.

3. As the *Buffer_0* is still holding the data token, it will be recovered and be passed to the sink. As shown, *Out10.F and Out11.T* transition to the value expected by the sink.

The SETAPII approach works in cases when (i) the glitch at SE is negligible as well as (ii) the resetting of *Buffer_E and Buffer_1* does not advance the handshake cycle. But the most important thing to be noted is that the circuit is not anymore QDI, due to the added delay constraint on the SD line. This SD line is also responsible for glitches at SE we discussed.

The state detection can be done with the last stage's completion detection circuit, so we don't have to really add extra state detection circuit for that. For resetting *Buffer_E and Buffer_1* the internal structure of the MCEs is modified without addition of any

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

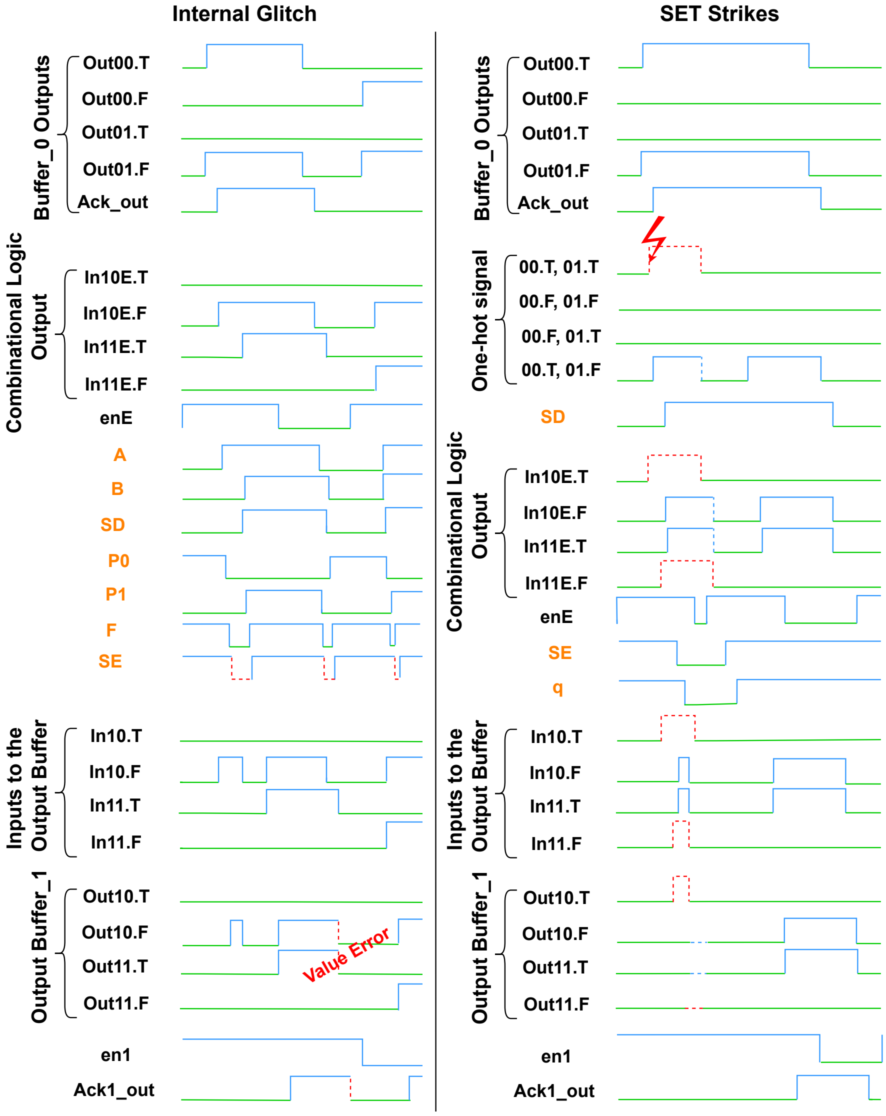


Figure 4.13: Behavior of SETAPII with and without SET strike

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

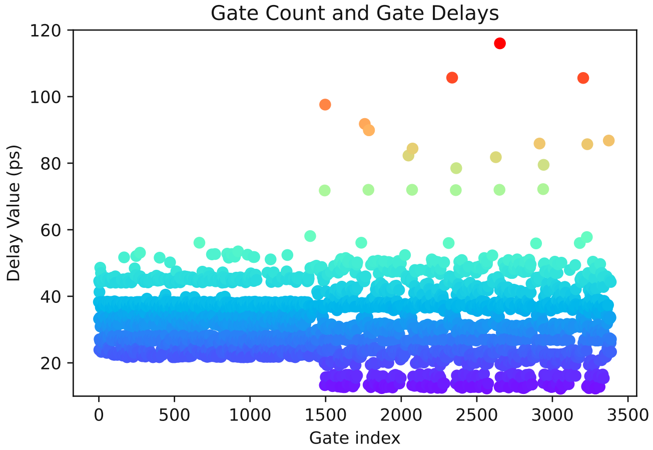


Figure 4.14: 16-bit 7 stage Pipelined multiplier with WCHB buffer template and SETAPII approach Gate Count (3389) and Gate Delays

extra gate. That is why the gate count remains the same, namely 3389 as counted in Figure 4.14. The minimum and maximum gate delays are 12ps and 116ps, respectively, the same as for SETAPI.

The SETAPII approach is evaluated in Figure 4.15 with the same 7 different PLFs as in the previous sections. A first look at Figure 4.15 in comparison to the Figure 4.8 tells us that the SETAPII is more susceptible to SETs. For PLF = 1 when injecting a 116ps pulse, e.g. , SETAPI Figure 4.8 fails 2759 times, while SETAPII Figure 4.8 fails 5019 times in mitigating the SET. The same situation is observed for all scenarios presented. Compared to SETAPI the logic contribution from SETAPII is higher for all PLFs. In extreme bubble limited mode when the *ACK* is delayed, we have more time to flush the combinational logic on the detection of the SET, where data is not allowed to pass through the output buffer that is not armed. So, as shown, in bubble limited mode the logic contribution decreases compared to the buffer part. Figure 4.15 tells another important fact about the flushing circuitry: for all PLFs the "Flushing" logic's contribution is non-negligible. So, SETs is itself contributing to make circuit susceptible to SETs. Compared to the token limited mode, in bubble limited mode the contribution of the "Flushing" part is not so prominent. This it is again due to delayed *ACK* signal. We have more time to recover before the output buffer goes active to latch the data.

4.3.1 Final Comments on SETAPII

Figure 4.16 presents a comparison of the SETAPII-based pipelined multiplier circuit with the simple WCHB-based one and the SETAPI approach. The first thing we notice is that the injection count is approximately the same as for SETAPI. This is due to the same structure and gate count. Compared to its predecessor, SETAPI, the circuit is more prone to deadlocks, glitch and value errors. The reason behind the glitches has already

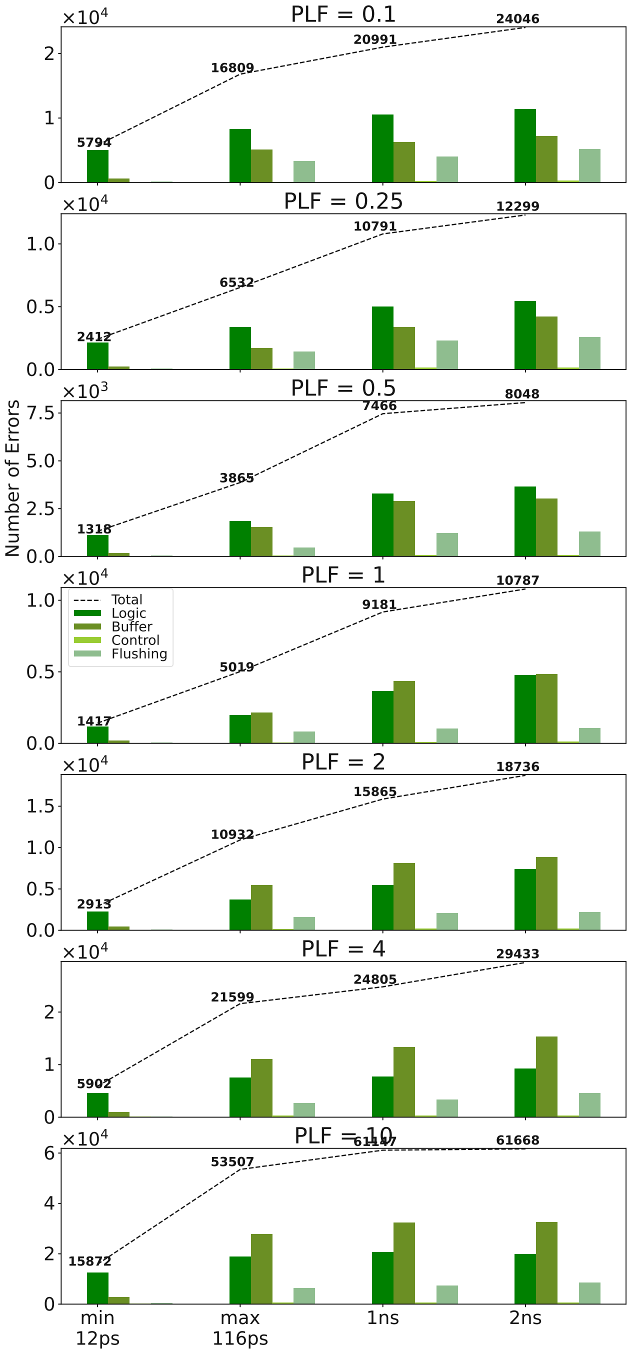


Figure 4.15: Suseptibility analyses of Pipelined multiplier with WCHB buffer template and SETAPII approach under the influence of SET with variable PLF and different injection pulse lengths

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

been explained: the SE signal can reset the output buffer without any extra check or awareness of the handshake phase, which may cause a glitch at the output. In addition, an SET within the combinational logic or in flushing circuitry can erroneously advance the circuit in its handshake phases without any signal from source or sink. This may cause deadlocks. In conclusion, due to these structural flaws, the SETAPII approach is not an effective choice for mitigating SETs within the combinational logic. Strictly speaking, it may not be considered a QDI approach either, because of the addition of delay assumption.

4.4 SETs effects in DIMS Combinational Logic with the Δ buffer template

Our proposed buffer template Δ showed good resilience with an empty pipeline configuration in the last chapter. Here we check its resilience with the pipelined multiplier circuit. In this technique the first transition locks the input and gives extra margin to finalize the decision whether it is a correct input or an SET. If the correct transition arrives before the completion of all bits and the SET has vanished, the correct transition takes the place of the faulty one, otherwise the first transition (correct or faulty) is considered valid. In addition, the output interlocks help to prevent erroneous behavior for different PLF. The buffer is equipped with the input interlocking, which cuts the probability of latching faulty transitions by the main buffer. This has poterntial to also mitigate some of the erroneous values propagated from the combinational logic. Figure 4.17 presents the 2-bit, 2 stage Δ buffer with a DIMS AND and XOR gate.

Using Figure 4.17 as a reference circuit, we will discuss the normal operation and the behavior of the circuit under the influence of SETs in the following. For the latter we only present a scenario where the circuit fails to mitigate the effect.

1. Figure 4.18-Fault Free: we follow the data propagation from the combinational logic to the *Buffer_1*. Data propagation from the *Buffer_0* to the output of the combinational logic is the same as in the simple WCHB based example. However, when data arrives at the *Buffer_1*'s internal inputs (after the input interlock stage), the first transition on *InIN10.F* or *InIN11.T* blocks the respective opposite rail for latching any transition, or, in other words, forcefully sets it to high. When all bits have arrived at the *IntIN* level, MCE-C12 changes its state to zero, and so *INC1* goes low. This transition, when combined with *Ack_In* changes *en1* to low. Now *Buffer_1*'s MCEs latch the transition, and *Out10.F* and *Out11.T* go high (marked with “1***”). This generates the *Ack1_out* “2” for *Buffer_0* and whenever the sink latches the provided data token, *Ack_In* goes low.
2. Figure 4.18-SET Strikes: a fault appears at *00.T, 01.T* close enough to the valid transition at *00.T, 01.F*. As we know, due to the realized logic function, *In10.T* also shows up as *In11.F*. As the faulty transition at *In10.T* occurs bit earlier than

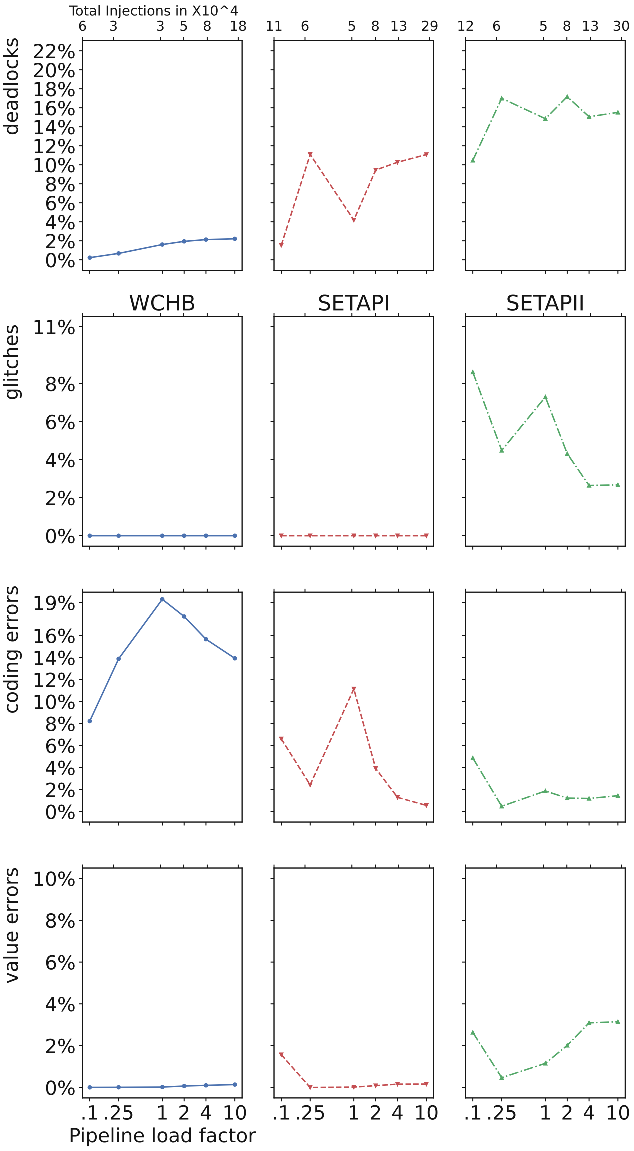


Figure 4.16: Comparison of SETAPII with all discussed approaches (Injection pulse is 1ns)

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

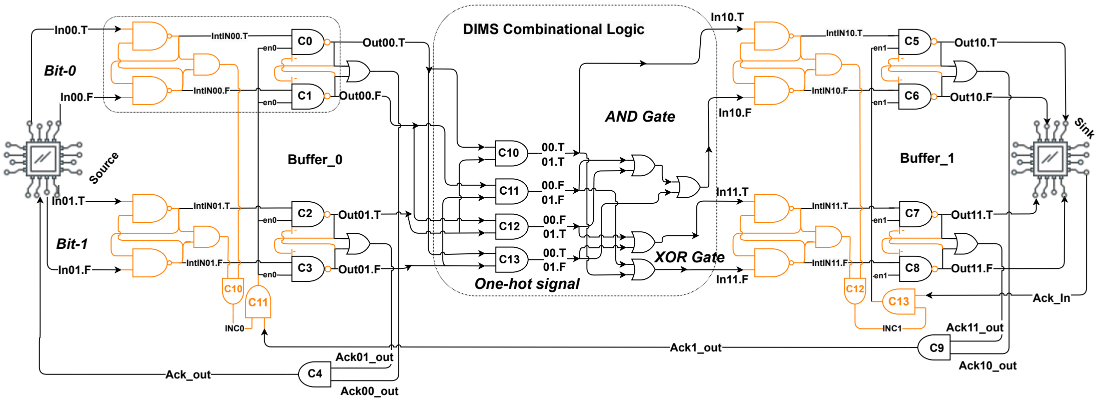


Figure 4.17: Δ with DIMS AND and XOR logic

the one on *In10.F*, it toggles the *IntIN10.T* to low and forces *InIN10.F* to remain high, which was actually expected to go low. Now the correct transition at *In11.T* as well as the faulty one at *In11.F* show up at the same time. For our setting we assume that the propagation delay of the NAND gate with the input *In11.T* is higher, so the faulty transition makes its way to *IntIN11.F*, locking out the correct one. As a result, a value error is detected at the output of *Buffer_1*, as shown.

From this discussion we conclude that the fault may propagate to the output if the SET appears a bit earlier than the correct transition and the buffer to which the transitions are fed is armed for the transitions. In contrast, if the fault arrives a bit late, we can safely mitigate its effect with the Δ approach.

Figure 4.18 tells us that the area overhead of the pipelined multiplier with the Δ buffer template compared to Figure 4.3 with the basic WCHB buffer is not so high. The difference is only 696 gates which stems from the input interlocking. The minimum gate delay is 17ps, with the maximum being 116ps.

For the evaluation we maintain our symmetry of performing simulations with 7 different PLFs and the 4 injection pulse values including minimum and maximum gate delay, as well as 1 and 2ns pulses. Figure 4.20 shows the results of fault-injection simulations for the 16-bit pipelined multiplier with the Δ buffer template. It can be observed that the susceptibility of the combinational logic is a bit higher than that of the buffer, only in extreme bubble limited mode it shows a more resilient behavior. This is again due to the delayed *ACK* signal: in those cases, the SET effect already vanished, and the valid transition is accepted by the input interlocking stage before the output buffer gets armed. From all discussed templates so far, except Δ, the WCHB is still the most resilient one.

In comparison with the basic WCHB Δ shows better resilience. If we specifically compare Δ with WCHB for pulse lengths equal to the minimum or maximum gate delay a circuit contains, the results from Figure 4.20 and Figure 4.4 show that Δ is way better. If we, e.g., select the result for PLF=1 when injecting a 73ps pulse, Figure 4.4 reports 3489,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

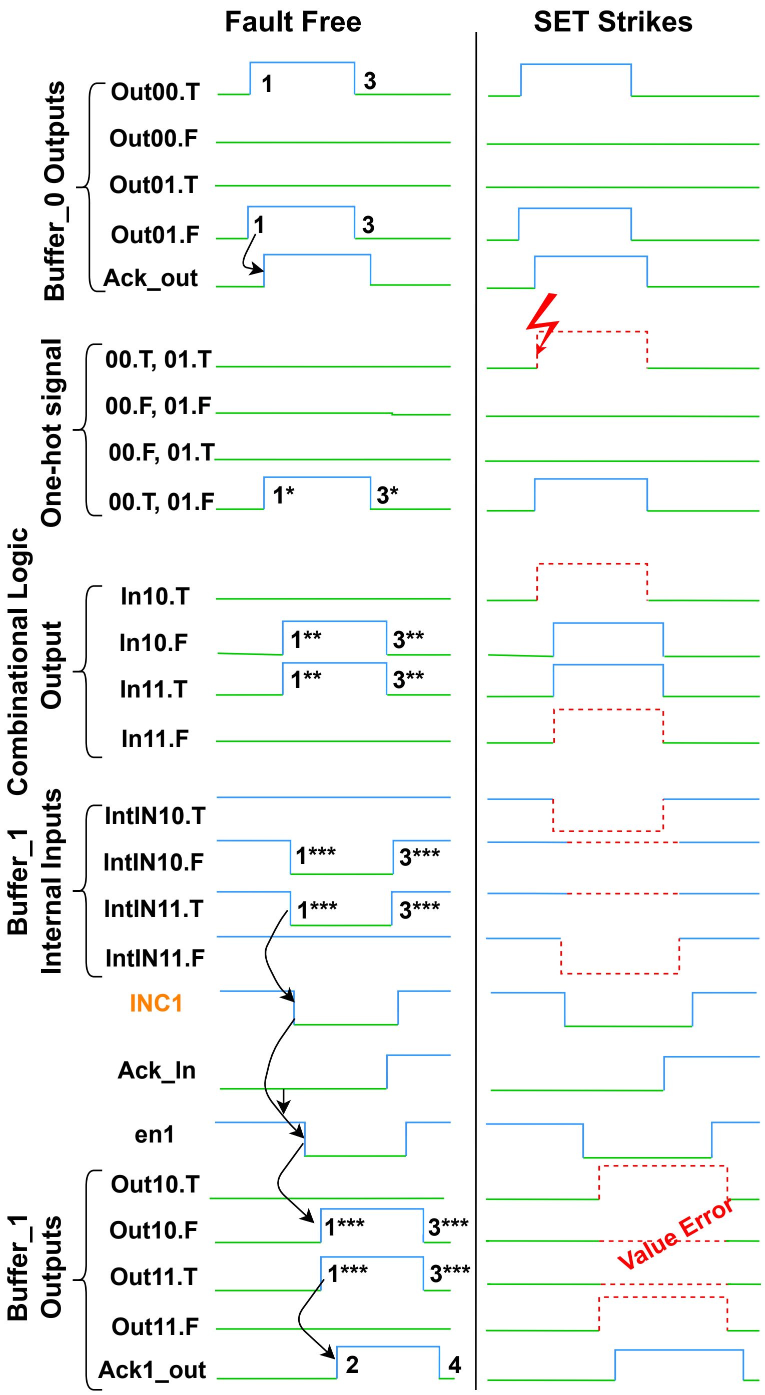


Figure 4.18: SET effects in combinational logic with theΔ buffer template

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

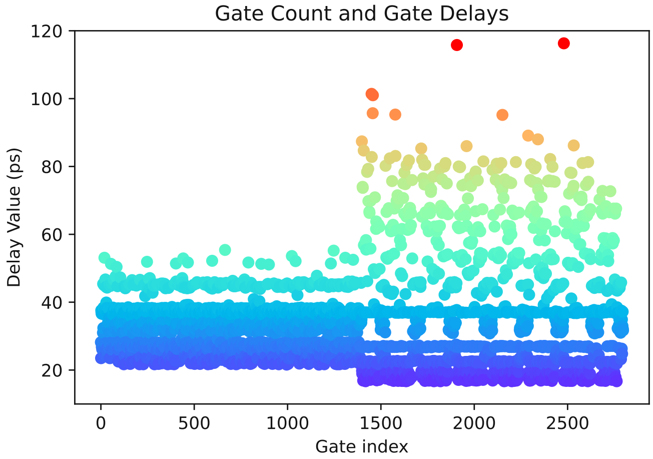


Figure 4.19: 16-bit 7 stage Pipelined multiplier with Δ buffer template: Gate Count (2798) and Gate Delays

while Figure 4.20 shows 844 for PLF=1 and 116ps – the difference is as large as 3005. These values show the resilience of Δ for injection pulses less or equal to maximum gate delay of the circuit.

4.4.1 Final Comments on Δ

Figure 4.21 presents the results for all discussed approaches based on Δ. Δ shows 6% value errors and approximately 0% for all other error types. These 6% occur only when the circuit is operating in balanced mode. Data tokens are received as fast as bubbles from the sink, so a faulty transition latched by the input interlock stage has a chance to get accepted as valid transition and cause a value error. In contrast, when the source is slow (token limited mode) the fault from the combinational logic or from the internal signals of the buffer is temporally masked because the buffer does not get armed before all bits arrived. As other bits may be delayed, the SET may, in the meantime, have vanished without effect. In bubble limited mode *ACK* is delayed, which gives an extra time margin during which the correct value can take the place of a potentially already decayed erroneous transient.

4.5 SETAPI approach for flushing SEUs from DIMS combinational logic with the Δ buffer template

In Figure 4.22, we present the 2-bit, 2 stage pipelined circuit with the DIMS AND and XOR logic. In the last section we concluded that the Δ buffer is a resilient buffer, so we decided to examine the SETAPI approach with this buffer template here. We call this configuration SETAPI_Δ, as the flushing approach is SETAPI and the “N” is for the new enhancement proposed with the buffer. To minimize the area overhead we proposed

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

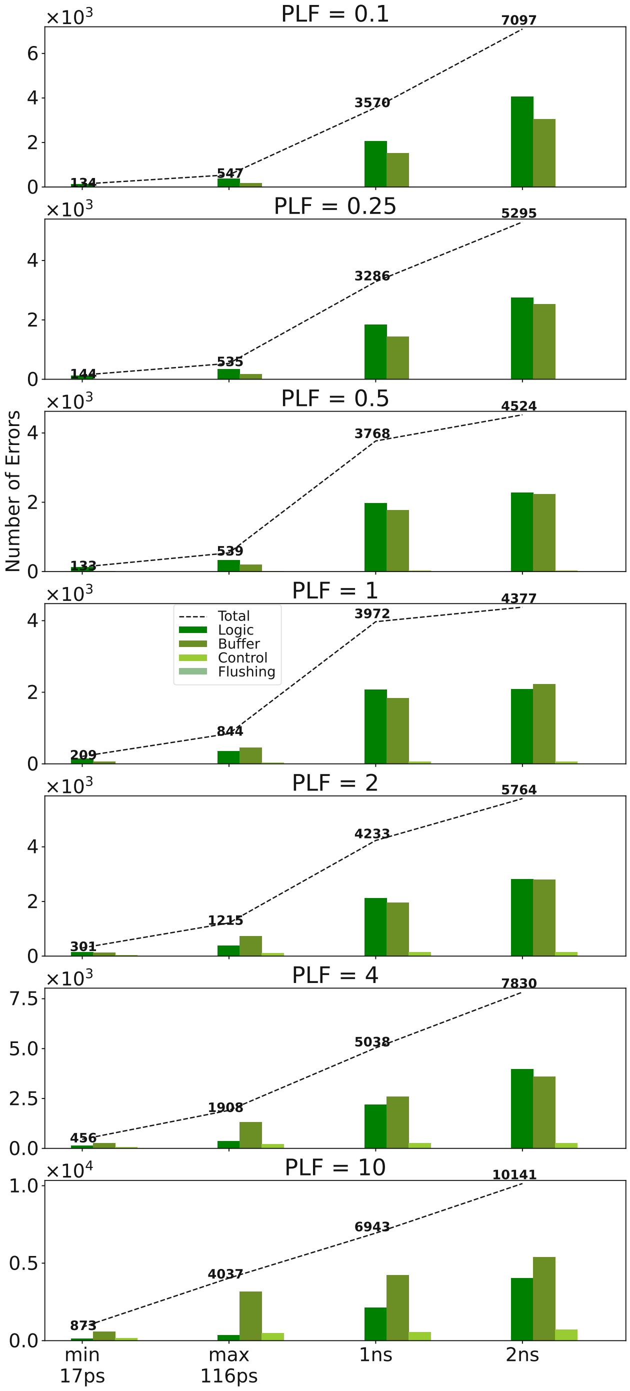


Figure 4.20: Suseptibility analyses of Pipelined multiplier with Δ buffer template under the influence of SET with variable PLF and different injection pulse lengths

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

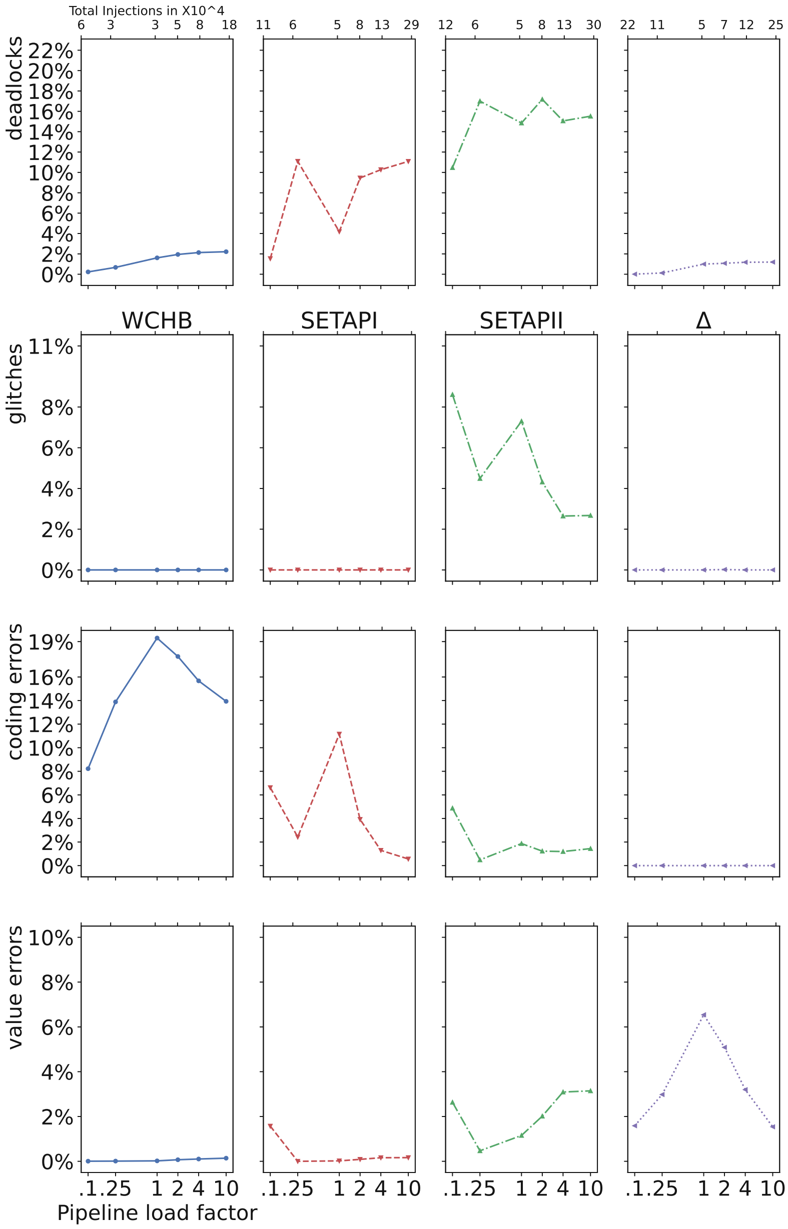


Figure 4.21: Comparison of all discussed approaches for Δ-based pipelined multiplier circuits (Injection pulse is 1ns)

some modification with buffer configuration. We utilize *Buffer_E*'s completion detection for *Buffer_1*'s input completion detection signal "INC1". *Buffer_E* is also a Δ-buffer but without input completion detection signal.

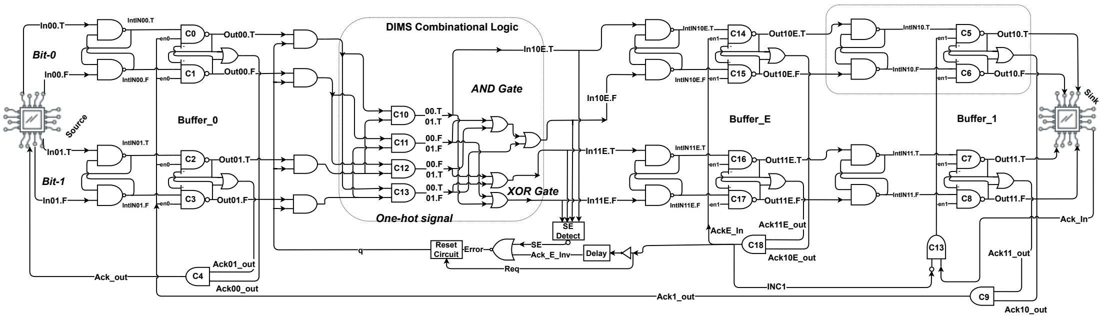


Figure 4.22: SETAPI approach with Δ buffer template with DIMS AND and XOR logic

In the following discussion we will use the waveforms in Figure 4.23, with reference to Figure 4.22, to discuss its normal operation and its behavior under SETs striking the internal signals of its combinational logic.

1. Figure 4.23-Fault Free: during the normal operation outputs of *Buffer_0* are passed to the combinational logic, which toggles a one-hot signal marked with "1*". This transition is passed to the output buffer *Out10E.F, Out11E.T* ("1**") after passing through the extra buffer. The completion detection of *Buffer_E* sets *enE* to high, arming for the spacer. The value of *enE* after inversion, when matched with *Ack_In* toggles *en1* to low, which then propagates the available data token to the output of the main buffer, as *Out10.F, Out11.T* (labeled with"1***") go high. This generates *Ack1_out* to *Buffer_0* and when these values are latched by the sink *Ack_In* finally goes high.

2. Figure 4.23-SET Strikes: an SET appears at *00.T, 01.T*. It is reflected at *In11E.F* and on the same signal *In10E.T*. The transition on *In10E.T* is a bit earlier than the valid transition at *In10E.F*, so the input interlocking only allows *In10E.T* to transition, which passes to *Out10E.T*. *In11E.T and In11E.F* show transitions at the same time, but here the valid transition at *In11E.T* blocks the faulty one on the opposite rail, because we assume smaller propagation delay of the respective NAND gate. When we procced further to the main buffer for *Out11E.T* a valid token is passed to the *Out11.T* without generating any error. But as the valid transition on *Out10E.F* is blocked by *Out10E.T*, now the faulty transition is passed to *Out10.T*, generating a value error. On the detection of the SET "q" goes low, which flushes the whole combinational logic. This is, however, worthless because the error has already propagated to the next stages, and now this flushing only generates a preliminary spacer which gets matched with the actual spacer generated by *Buffer_0*, without generating any deadlock situation.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

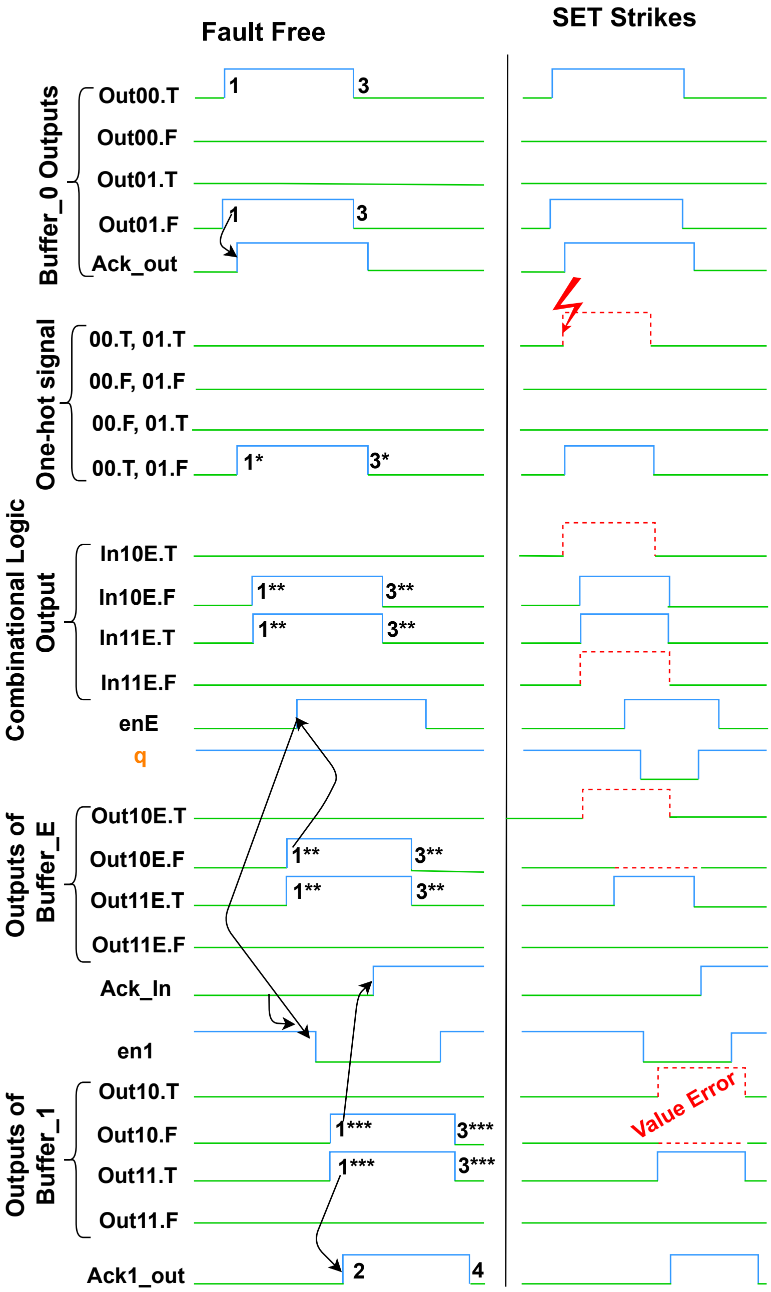


Figure 4.23: SET effects in combinational logic with the SETAPI approach combined with the Δ buffer template

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

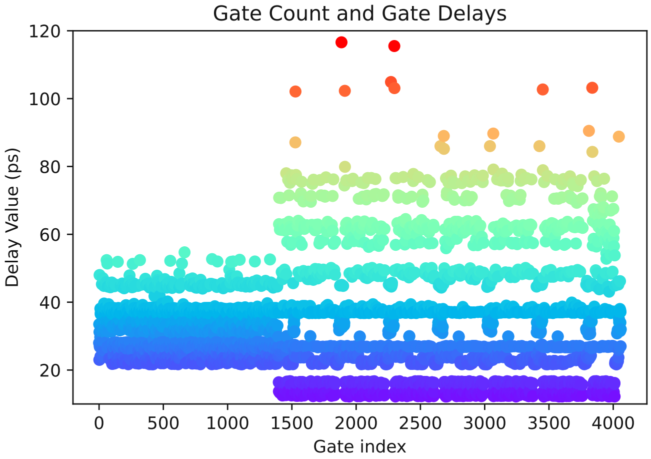


Figure 4.24: 16-bit 7 stage Pipelined multiplier with Δ buffer template and SETAPI approach: Gate Count (4060) and Gate Delays

We conclude that flushing is again worthless, but the Δ approach blocks the coding errors. These are, however, in some cases not the worst possible effect, as we are able to detect them more easily than value errors (in practice, without availability of reference results for comparison).

Figure 4.24 presents the gate count, that is 4060. Compared to the 3389 from SETAPI it is 671 higher, due to the input interlocking of the Δ buffer. The minimum and maximum gate delays are 12ps and 116ps, respectively.

The suseptibility analysis of SETAPI_Δ is presented in Figure 4.25 with 7 different PLF's and 4 injection pulses. When we compare it with SETAPI, we observe that the contribution from combinational logic part is approximately the same, but the buffer is more susceptible. This is simply because the number of nodes of the Δ buffer is higher than for the WCHB (in SETAPI), so the injection count is higher. For short injection pulses like 12ps it shows better resilience, as each transition must pass the input filters so it may get cancelled. In general, this configuration does not appear very attractive, but a closer look tells us that at least the contribution from the flushing circuit is lower than with SETPAI in token limited mode. Due to the Δ buffers' input interlocking, and since this input interlocking is done twice after signal passes from the combinational logic, the propagation delay doubles relative to the simple WCHB based approach. Now if the SET hits the flushing logic part and generates some erroneous value, it is likely that before these transitions appear at the input of the main output buffer and generate the unexpected *ACK* for the source, the source already responded with the last *ACK*, and this slightly early erroneous *ACK* is merged with the expected one. To illustrate this, just suppose we are holding the data token and just asked the source for the spacer. Now the SET on the flushing circuit accidently flushes the whole combinational logic. Due to the extra two layers of input interlocking NAND gates, however, this flushing takes more

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

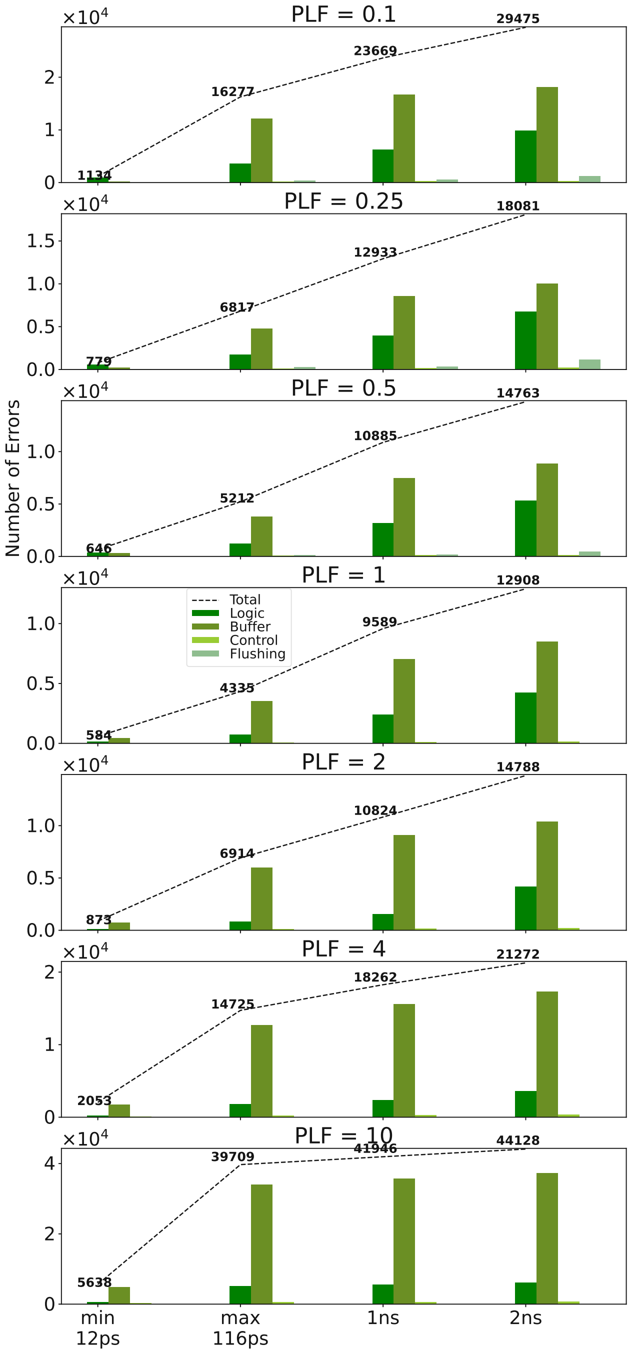


Figure 4.25: Suseptibility analyses of the Pipelined multiplier with Δ buffer template with SETAPI approach (SETAPI_Δ) under the influence of SETs with variable PLF and different injection pulse lengths

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

time than SETAPI needs to generate the bubble for the source. So meanwhile the source already responds to the last *ACK* signal and issues the spacer. So, this bubble may arrive a bit early but does not violate the handshake protocol. When comparing with Δ, SETAPI_Δ due to the extra buffer shows high deadlock rate . For example, for PLF = 1 when injecting 116ps pulses SETAPI_Δ fails 4338 times (Figure 4.25) compared to Δ alone which fails only 844 times (Figure 4.20).

4.5.1 Final Comments on SETAPI_Δ

Figure 4.26 shows that the injection count is approximately twice as high as for Δ and SETAPI. Thanks to the Δ buffer template there are no coding errors and no glitches. The value errors are approximately same in percentage as with Δ. But when it drops it is at the cost of more deadlocks, these are due to the flushing approach deployed which may violate the handshake protocol by generating the spacer as SET effect. So even with higher area overhead we still do not achieve the desired results compared to the Δ.

4.6 ζ approach to flush SEUs from DIMS combinational logic with a modified Δ buffer template

A key drawback of existing methods is that upon detection of an error they force a NULL wave, using, e.g., the AND gates in front of the combinational block in Figure 4.5 to clear the pipeline. This is counterproductive as it can turn the erroneous state into a data token that needs to be appropriately discarded in the receiver (if possible, at all), while at the same time causing a speed and energy penalty. A smarter approach would be to selectively reset only the one MCE that was erroneously set by the SET. This is possible by leveraging the fact that the SET will vanish, leaving the incorrectly set MCE in storage mode, while all correctly set MCEs remain in combinational mode (matching inputs) until the data token is acknowledged. So, if we manage to reset an MCE that is in hold mode only, we can elegantly flush out the error (A conventional reset applied to an MCE with both inputs at 1 will cause an undesired negative output pulse for the duration of the reset). To implement this, we need a special MCE implementation variant. Our proposed MCE is shown in Figure 4.27 *MCEn*. Its reset input n forms the (active low) reset that will force the MCE to output 0 only when it is in storage mode. In ζ [TSNH23] we replace all the MCEs within the combinational logic with our special "MCEn" presented in Figure 4.27 highlighted with orange color. Here the orange colored NAND gates in front of the combinational output detect the presence of SET and reset the *Flush* signal. This signal is then fed to the "MCEn". Here, we modified the Δ buffer to fit with our flushing technique presented in Fig. 4.27 with one extra signal highlighted with orange color, namely the asymmetric input signal of MCE-C13. This gate is responsible to pass *Ack_In* when input data is completed. The *Flush* signal only allows this gate to pass *Ack_In* when it is high.

The working principle of ζ then becomes as follows:

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

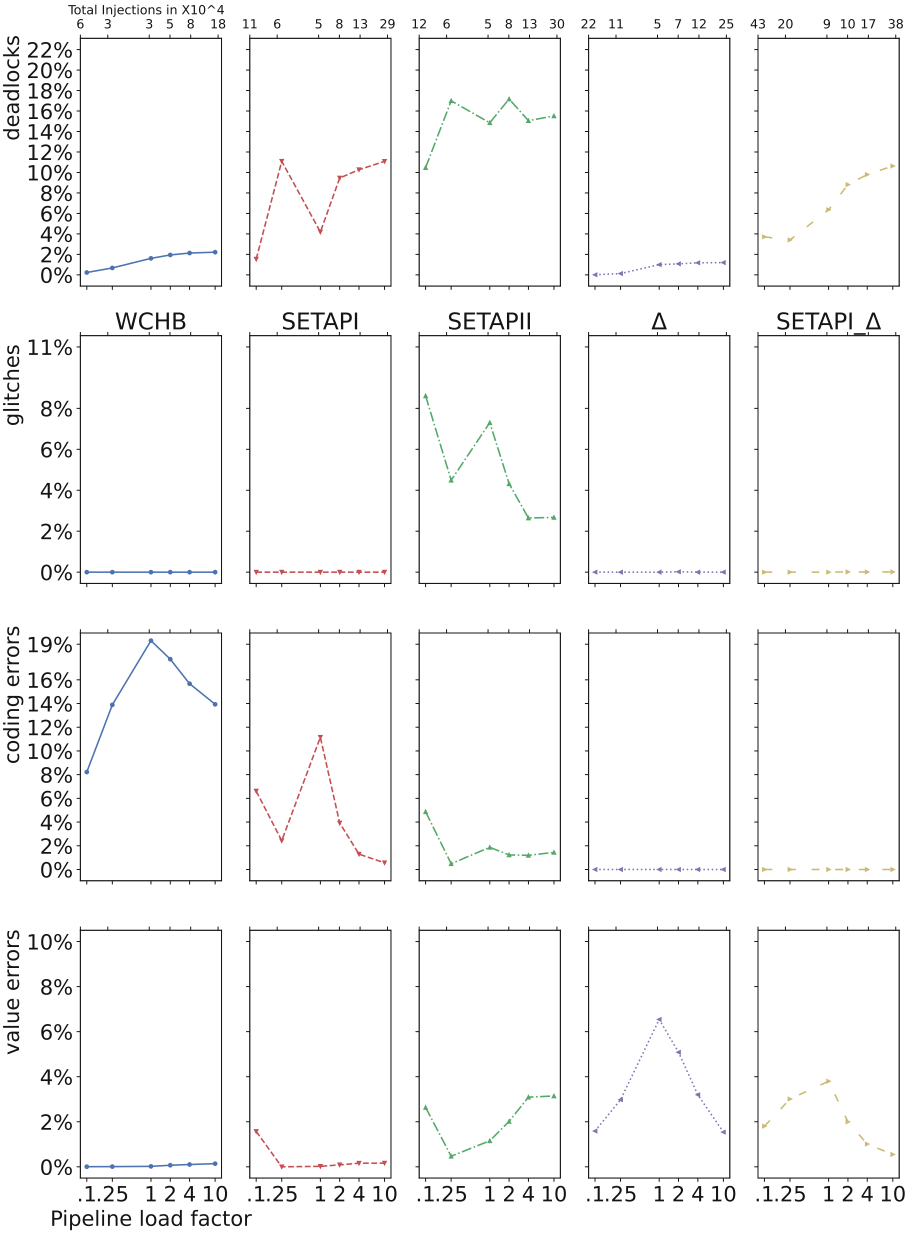


Figure 4.26: SETAPI_Δ (SETAPI with Δ buffer template) comparison with all discussed approaches (Injection pulse is 1ns)

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

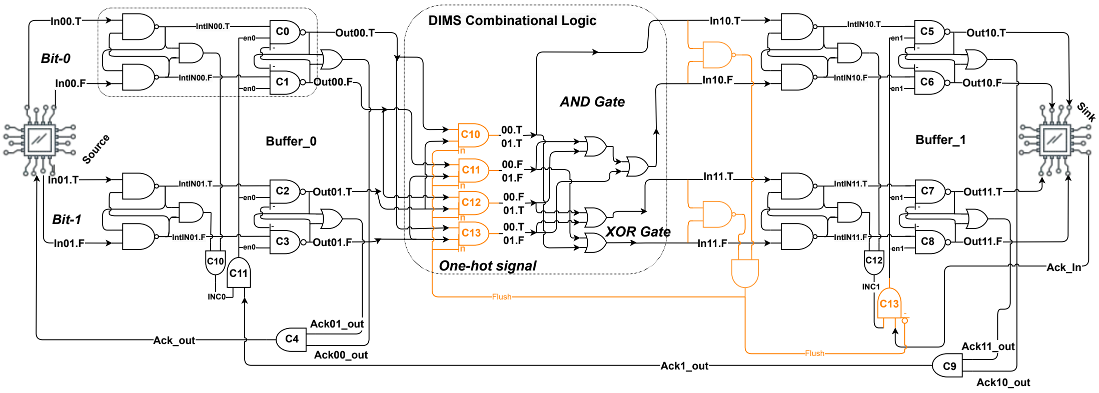


Figure 4.27: ζ approach with Δ' buffer template with DIMS AND and XOR logic

1. As Figure 4.27 suggests, upon detection of the illegal codeword (.T rail and .F rail of the combinational logic output go high) in the dual-rail 4-phase handshake protocol the *Flush* signal goes to 0 (i.e., it becomes active).
2. The main two functions of *Flush* are: halt the input acknowledgement to the receiving buffer *Buffer*_1 and perform our targeted reset on the MCEs in the combinational block, thus removing the SET.
3. Halting the acknowledgement to *Buffer*_1 only helps in mitigating faults that appear before the arrival of the acknowledgement (Ack_In) from the successor stage. In this case we can temporarily block the passage of the acknowledgement signal to the *en1* of *Buffer*_1, and in the meantime flush the erroneous value within the combinational logic.
4. During the data phase the predecessor stage *Buffer_0* provides a data value and within the combinational logic only those MCEs go high that have a 1 on both their standard inputs that remains stable until *Buffer_1*'s completion detector acknowledges the data.
5. All the MCEs in combinational mode (matching inputs) logically mask SETs that manifest as HI pulses on their input or output. An SET producing a LOW pulse may delay the switching of the MCE to 1, but that is acceptable with QDI logic.
6. The only critical scenarios are a HI SET on the output of an MCE in storage mode (asymmetric inputs) which can flip it to 1 if the length is larger than its storage loop delay. Or, alternatively, if one input of the MCE is high, and on the second one a HI SET appears, then depending on the gate delay the output will go to 1.
7. Our targeted reset allows flushing the errors exactly in those critical scenarios outlined above.
8. More specifically, whenever *Flush* goes low all MCEs with non-identical inputs are reset.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

9. If the fault is due to an SET at the MCE input, the *Flush* signal remains low until the SET effects vanished from the input and *Flush* resets the respective MCE.

With this elegant flushing mechanism, we can easily mitigate the effects of single event transients (SETs) from the combinational logic.

As mentioned in the above description, the flushing only takes effect as long as *Buffer_1* is not enabled. In an adverse scenario *Buffer_1* is already enabled when the SET occurs within the combinational block and latches the faulty input, irrespective of the flushing attempt. So, it is useful to avoid activating the buffer early.

During the data phase the acknowledgment is double-checked with the *Flush* signal. Due to the inverted nature of the main buffer the acknowledgment does not require inversion, so the data is only allowed to pass when *Flush* is 1. We call this approach with enhanced buffer "ζ_Δ'" or ζ approach with Δ' in [TSNH23].

Here we discuss the data propagation through the "ζ_Δ'" in normal operation and its behavior while experiencing an SET within its combinational logic node with the aid of Figure 4.28 where the reference circuit is given in Figure 4.27.

1. Figure 4.28-Fault Free: data propagation from the output of *Buffer_0* to the output of the combinational logic and then further on to the input interlocking where the *IntIN10.F and IntIN11.T* transitions are followed by the *INC1* (indication of data completion) is the same as for Δ Figure 4.18. The main difference is: *en1* only goes low if the *Flush* signal is high and *Ack_In* is low, arming *Buffer_1* for the data token. Then the generation of the data token at the output of *Buffer_1* (labeled with "1***") toggles *Ack1_out*, completing the data phase with *Buffer_0* which is also called the 2nd handshake phase. The spacer marked with "3" at *Buffer_0*'s output also follows the same sequence, ending at phase "4" and completing the handshake cycle.
2. Figure 4.28-SET Strikes: a fault hits *00.T, 01.T* (also with label *In10.T*) and propagates to *In11.F*. The valid transitions also appear approximately at the same time, but both faulty transitions get locked at *IntIN10.T and IntIN11.F*. The *Flush* signal goes low on the detection of the illegal code word, which then halts the propagation of the data to the outputs of *Buffer_1*, as well as resets MCE-C10 (Figure 4.27). After the transient effects have vanished, the reset clears rails *In10.T* and *In11.F*. In this case the transient had already disappeared, so resetting took place immediately and we recover the original transitions at *IntIN10.F, and IntIN11.T*. On the removal of the illegal code word the *Flush* signal goes back to high. We can now generate *en1* which then passes the data token to the output of *Buffer_1* as shown; the transitions on *Out10.F, Out11.T* are valid ones.

We conclude that ζ_Δ' flushes the targeted MCE and elegantly recovers the correct values without violating the handshake protocol. The most important thing to be noticed

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

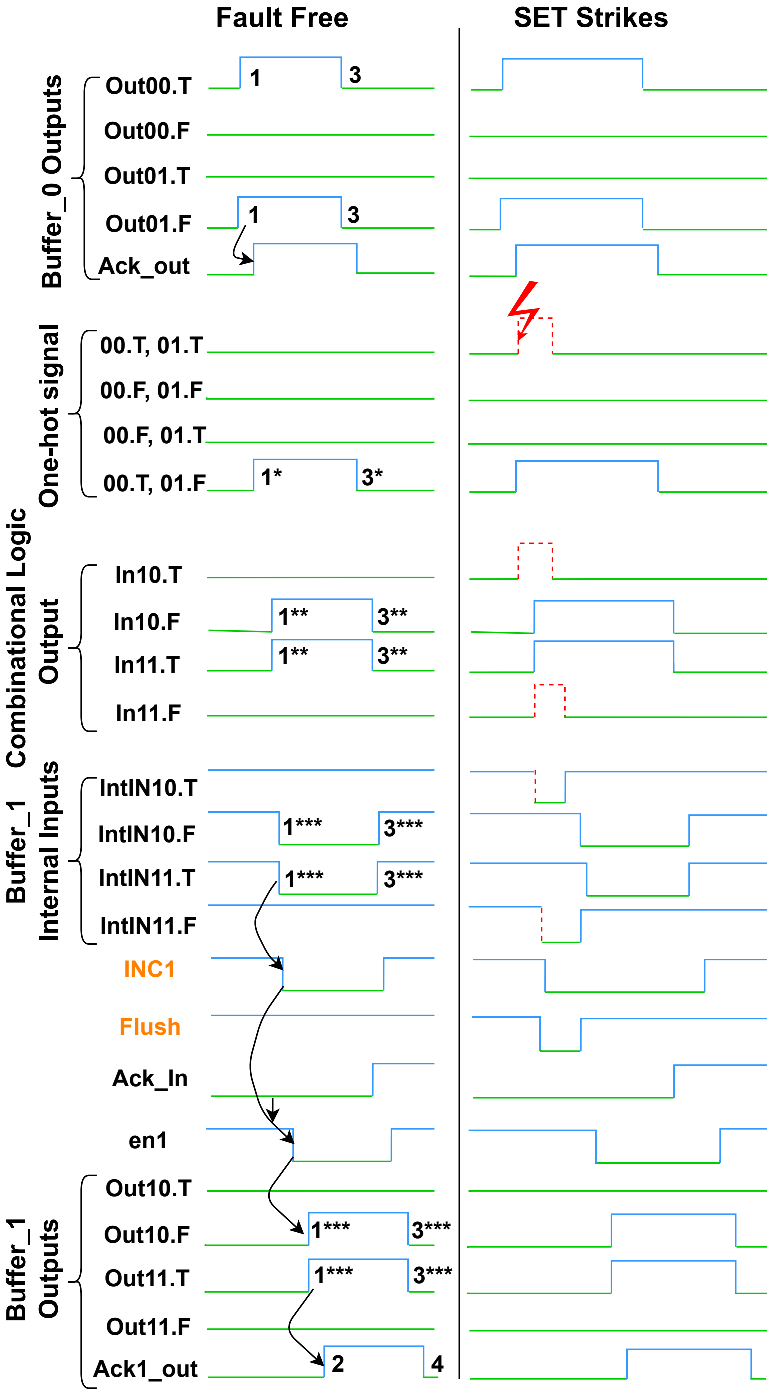


Figure 4.28: SET effects in combinational logic with the ζ approach combined Δ' buffer template

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

is that with this approach we are maintaining our valid tokens as shown in Figure 4.28: during our flushing process the *00.T, 01.F* rail remains high, providing the valid token to the buffer.

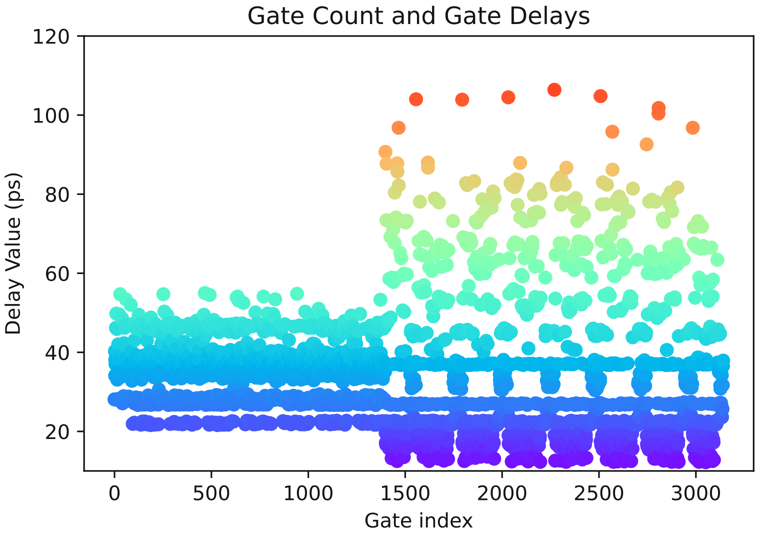


Figure 4.29: 16-bit 7 stage Pipelined multiplier using Δ' buffer template with ζ approach: Gate Count (3142) and Gate Delays

According to the circuit statistics presented in Figure 4.29, the circuit implementation usingζ_Δ' has a total gate count of 3142, which is only 344 gates higher than for Δ (2798). The maximum propagation delay of a gate a circuit contains is 116ps, with the minimum being 17ps.

With the same evaluation settings for ζ_Δ', Figure 4.30 suggests full resilience towards SETs appearing within the combinational logic: The count for logic-related errors is zero now, indicating that ζ_Δ' has perfectly accomplished its mission of mitigating these. In the previous sections we concluded that the Δ buffer template is one of the most resilient approaches. In spite of the fact that it is not designed to flush the combinational logic, it can still mitigate effects originating there when they reach the buffers inputs. So, here we only compare our approach ζ_Δ' with Δ (Figure 4.20).

For small pulses like 17ps the circuit shows full resilience with zero errors during token limited mode. This is due to the fact that the buffer is not enabled for the next transitions until the source responds with the new token or spacer. So while holding the data token all buffers are locked at input and output. Consequently, faults propagating from the input side are totally ignored as these inputs are locked already. If the pulse is small enough, it will disappear before the circuit changes its state. In contrast, the 116ps pulse is long enough to stay until the source starts responding, so the SET may flip any MCE that changes its state to storage mode. As we move from extreme token limited mode to

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

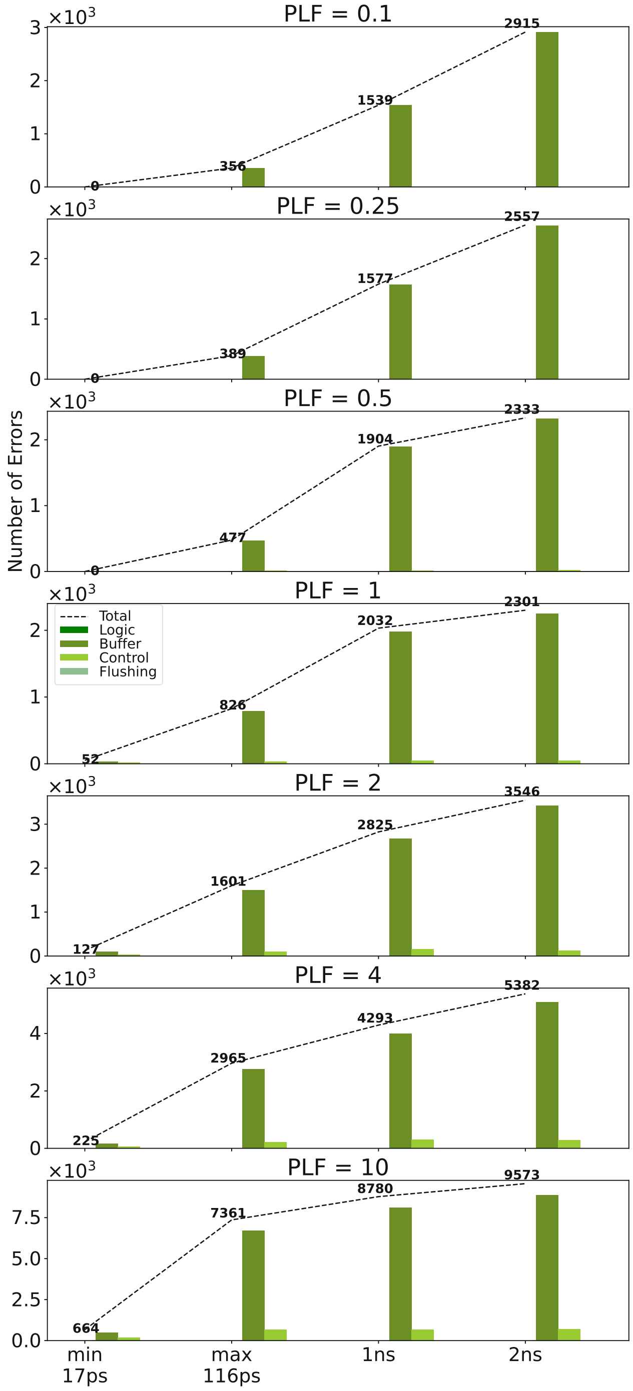


Figure 4.30: Suseptibility analysis of the Pipelined Multiplier using the Δ' buffer template with the ζ approach under the influence of SETs with variable PLF and different injection pulse lengths

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

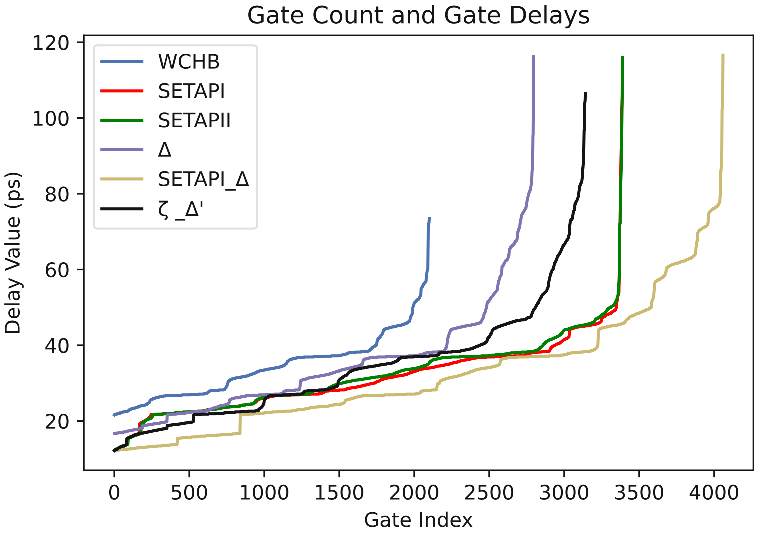


Figure 4.31: Gate Count and Gate Delays Comparison of all dicussed approaches

the balanced mode, the error counts go higher, because the source changes the circuit state faster and the probability increases that the SET is latched by any MCE.

During bubble limited mode the combinational logic already changed its state, but the main buffer is waiting for the *ACK* signal and therefore not armed yet for a new transition. But if the SET already passed the input interlock stage and is long enough, it has a non-zero probability to be latched by the main buffer on the reception of the *ACK*.

Comparing ζ_Δ' (Figure 4.30) with Δ (Figure 4.20) we notice that ζ_Δ' approximately halves the error occurrence. However, the buffer contribution is still there and even bit higher than with Δ. This is first due to the higher injection count, where the buffer's internal nodes are not covered by the flushing mechanism. Consider node *IntIN10.T* in Figure 4.27: if an SET hits there and blocks the valid transition from *In10.F*, this scenario is out of the reach of illegal code word detector, and on the arrival of *en1* the faulty transition gets latched by MCE-C5. Otherwise, if the SET hits the output of MCE-C0 and flips the state before the valid transition is latched by MCE-C1, the faulty transition will be considered as valid but raise the error flag when checked by the monitor at the end. While it is not possible to discuss all feasible scenarios here, this discussion hints why we are experiencing errors as the SET length increases.

4.6.1 Final Comments on ζ_Δ'

With Figure 4.31, we can compare the area (gate count) of all discussed approaches. It is evident that ζ_Δ' outperforms other combinational logic flushing techniques. Subsequently, Figure 4.32 presents the resilience comparison of ζ_Δ' with all discussed approaches. We observe no coding errors and glitches, just like with Δ and SETAPI_Δ.

In addition, we achieve approximately the same resilience for the deadlocks as Δ. With a higher injection count compared to Δ, ζ_Δ' shows a lower percentage of value errors. For example, for PLF $= 0.1$ the error rate of Δ is approximately 1.8% with the injection count being $22 * 10^4$, while ζ_Δ' shows approximately 0.5% of $26 * 10^4$. The spike of value errors in balanced mode also drops from 7% to 2%. This can be explained as follows: If the error is from the combinational logic while we are operating in balance mode, the detection of the illegal code word halts the ACK signal to the buffer during the time window when the signal propagates through the input interlock stage. Then we flush the erroneous transition and pass the valid transition. The remaining percentage, as we already discussed above, is from faults originating in the buffer template.

With all these results and discussion, we can conclude that the ζ_Δ' approach covers the whole combinational part and can easily mitigate the effects generated from that part. Although we do not further explore this avenue here, it is quite likely that further investigation may enhance the buffer template; maybe even to the point where we can finally experience a fully SET resilient QDI circuit. Throughout our analysis so far we only discussed a linear pipelined circuit. In the next chapter we will start investigating a nonlinear QDI pipelined configuration.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

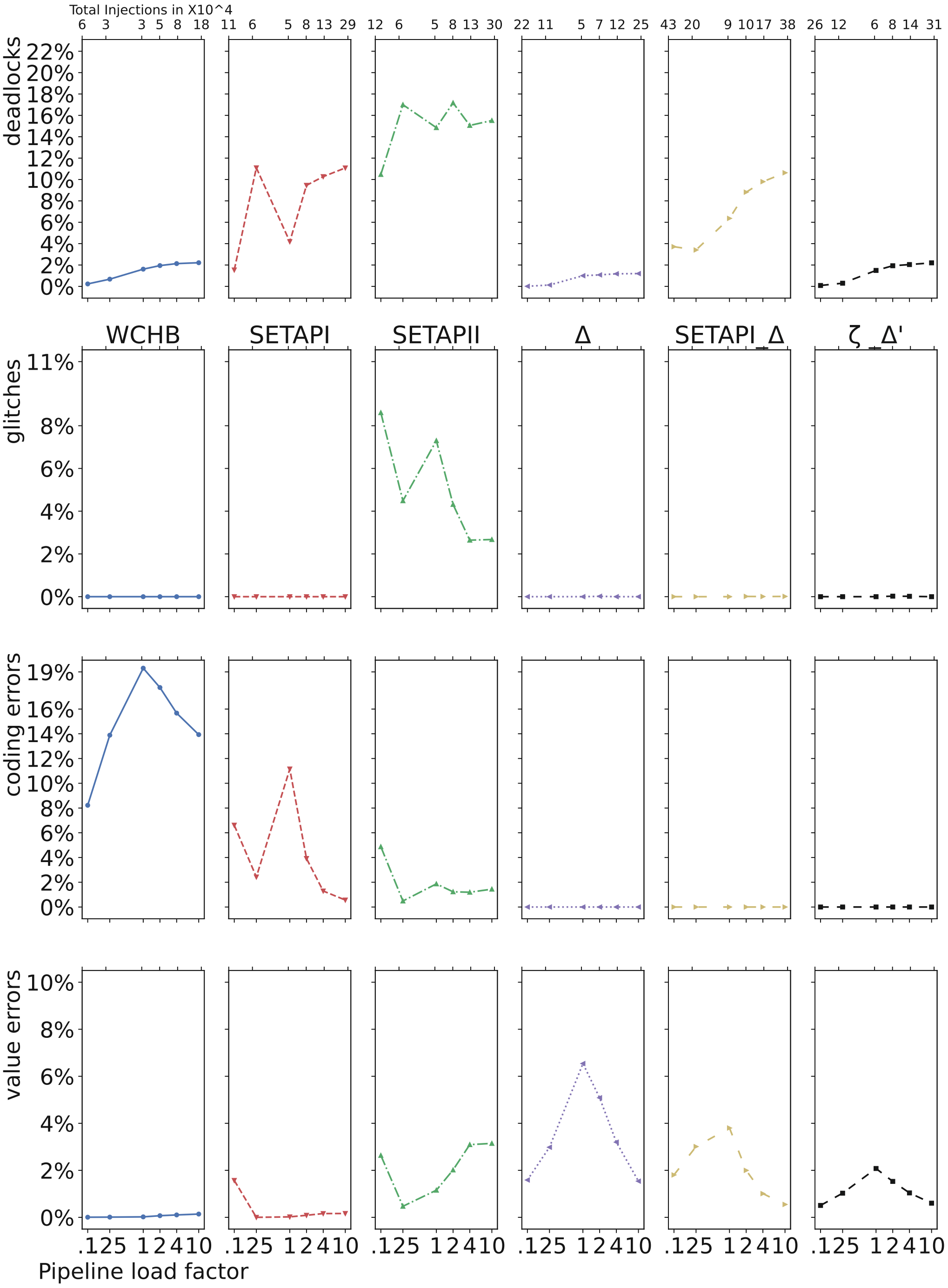


Figure 4.32: Comparison of the ζ approach with Δ' pipelined multiplier circuits with all discussed approaches (Injection pulse is 1ns)

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

CHAPTER 5

SET suseptibility of QDI Conditional Control Elements

In the last two chapters our focus was QDI buffer templates and the DIMS combinational logic, respectively, where we only considered linear pipelined circuits. Here we refer the nonlinear pipeline configuration introduced in Section 2.4 in which we have conditional control elements namely *Multiplexer* and *De-multiplexer*. To maintain the causal behavior of the QDI circuit these elements also contain MCEs, so in principle these elements also require special consideration to improve the overall fault tolerance of the circuit. Currently, to the best of our knowledge, literature does not address the topic of their explicit protection. We analyze the behavior of these elements under SETs and highlight the root cause behind the conversion and propagation. We select the 16-bit Iterative multiplier circuit as our target circuit, the combinational logic is DIMS. We first analyze the behavior with a WCHB buffer template as well as a simple *Multiplexer* and *De-multiplexer* with DIMS combinational logic. Then we propose enhanced *Multiplexer* and *De-multiplexer* implementations with our resilient buffer template Δ and the combinational logic with the ζ_Δ' flushing approach. As the behavior of the circuit is known in each phase of the handshake, we can utilize this information to harden the circuit efficiently.

5.1 SET effects on QDI *Multiplexer* and *De-multiplexer*

Figure 5.1 presents the gate level structure of the QDI *Multiplexer* with source and *Buffer_4* as inputs and *Buffer_3* as output. Our focus is the *Multiplexer* so we only present the specific part of the iterative multiplier here for the discussion. As Figure 5.1 suggests, each path of the *Multiplexer* is controlled by a dual-rail *SEL* line. As only one data path is activated, we merge the output lines of both paths with OR gates. The *Ack_out* from *Buffer_3* is only passed to the respective active source, as determined by *SEL.T and .F*.

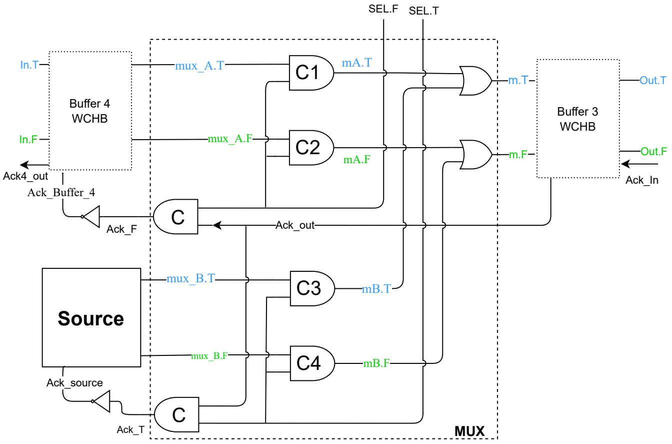


Figure 5.1: 1-bit Multiplexer with WCHB buffer template

In the following the propagation of the data through the *Multiplexer* is discussed, with Figure 5.1 as reference circuit and the waveform given in Figure 5.2.

1. Figure 5.2 Fault-Free: a data token from the source is placed at *mux_B.T* (labelled with “1”). As *SEL.T* is already high (“*1”) the data token is passed through *Multiplexer* path “B” and arrives at *mB.T* (“1*”). After propagating through the OR-gate it appeares at *Multiplexer* output *m.T* (“1**”). Once latched by *Buffer_3*, *Ack_out* is generated, which then, combined with the *SEL.T* (“*2”), toggles *Ack_source* to low (“2*”). The source now responds with a spacer (“3”), that will propagate in the same way as the data token (“4*”). In response to *Ack_source*, the source generates a new data token, but this token has to wait until the *SEL.T* goes high. This time *SEL.F* goes high, which allows *Buffer_4* to pass data through *Multiplexer* path “A”. There the data propagation is same as we discussed for path “B”.

2. Figure 5.2 SET Strike: this time the selected path is “A”, so the transition appeares at *mA.F* (“1*”) and propagates to *m.F*. Once it is latched by *Buffer_3*, *Ack_out* is again generated (“2”), and in reponse the spacer arrives from *Buffer_4* (“3”). Before that spacer reaches *Buffer_3*, an SET hits *mB.F* while MCE-C4 (Figure 5.1) is in storage mode (*mux_B.F* is high and *SEL.T* is low). Assuming the SET is long enough, it flips C4’s state. Although *mA.F* is low, *m.F* is now forcefully set to high by the faulty transition at *mB.F*. As we are not able to advance the handshake cycle from here, we enter a deadlock situation.

We can conclude that if an SET on the unactive path gets latched by the respective MCE, this will cause a deadlock.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

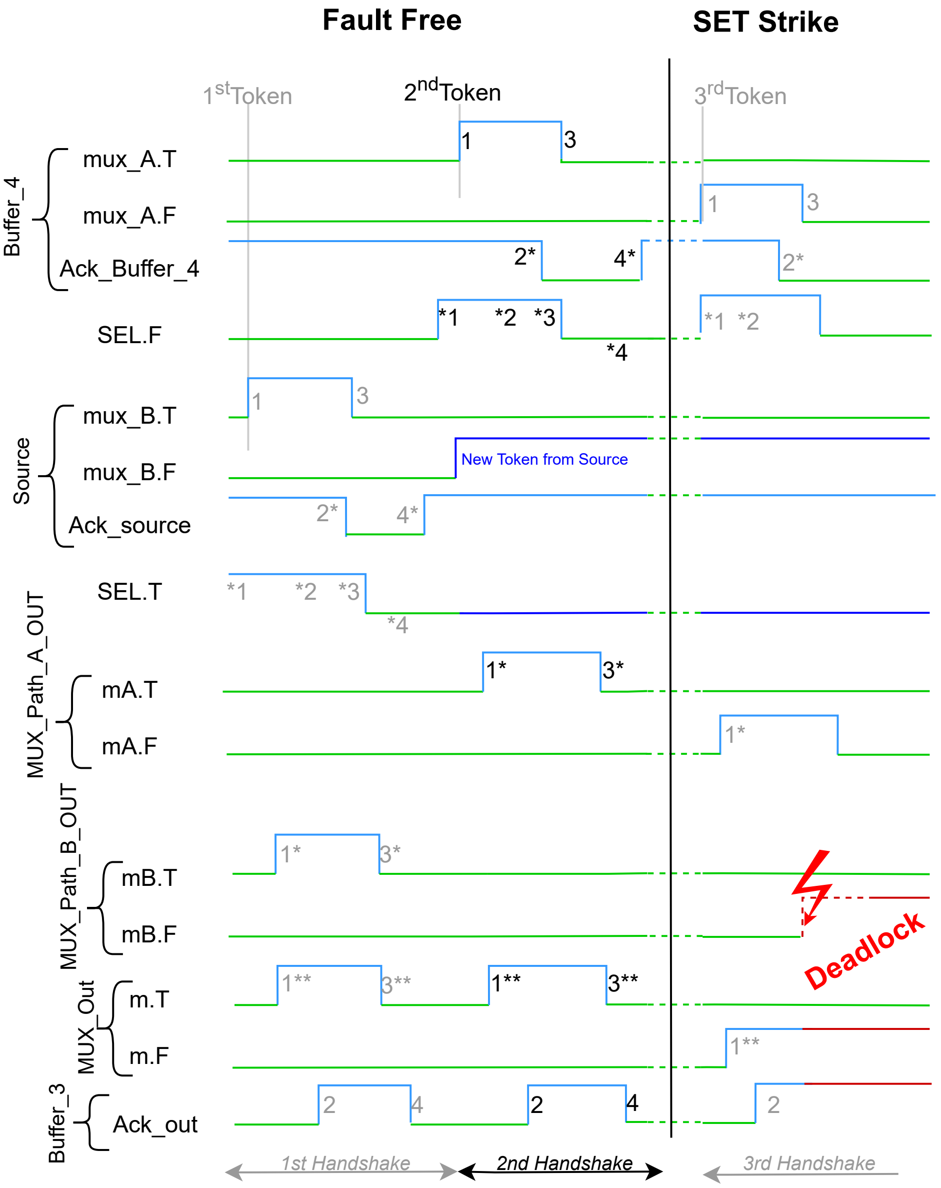


Figure 5.2: SET effects at the Multiplexer with WCHB buffer template

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

5.1.1 *De-multiplexer*

Figure 5.3 presents the 1-bit *De-multiplexer* circuit from the iterative multiplier configuration. The inputs *demux.T and demux.F* are passed to the *Sink* or *Buffer_5* depending on *SEL.T and SEL.F*. With this reference circuit we will now explain the normal data propagation and the behavior under the influence of SETs in Figure 5.3.

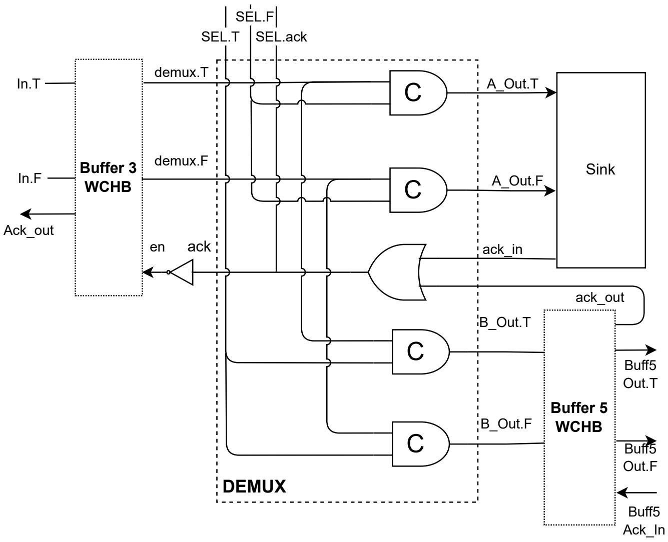


Figure 5.3: 1-bit DeMultiplexer with WCHB buffer template

1. Figure 5.3 Normal Operation: a data token labelled with “1” is passed to the *De-multiplexer* input *demux.T* (“1*”). This transition generates the acknowledgement for the predecessor stage toggling *Ack_out* to high (1**). As select line *SEL.T* is high (“*1”), the token appearing at *demux.T* is simply passed to *Buffer_5*’s input *B_Out.T* (1***). Once it is latched by *Buffer_5*, *ack_out* goes high (2). This transition propagates to *Buffer_3* generating the spacer there (3). The spacer propagation is the same and ends the first handshake cycle at (4**). A second token follows in the same way but this time it passed to the *Sink*.

2. Figure 5.3 SET strike at DEMUX Inputs: during the communication with *Buffer_5*, an SET appears at *De-multiplexer* input *demux.F*, which is also a primary output of *Buffer_3*. This erroneous transition gets latched and remains there even after the triggering SET vanished. *SEL.T* is already high, so both valid and faulty transition are propagated to the *De-multiplexer* output *B_Out.T* and *B_Out.F* respectively, generating the coding error. *Buffer_5* is a simple WCHB, so it accepts both transitions and propagates them to the next stages.

3. Figure 5.3 SET strike at DEMUX Output: the presented scenario is the same as above with respect to the data token propagation, but this time the SET appears

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

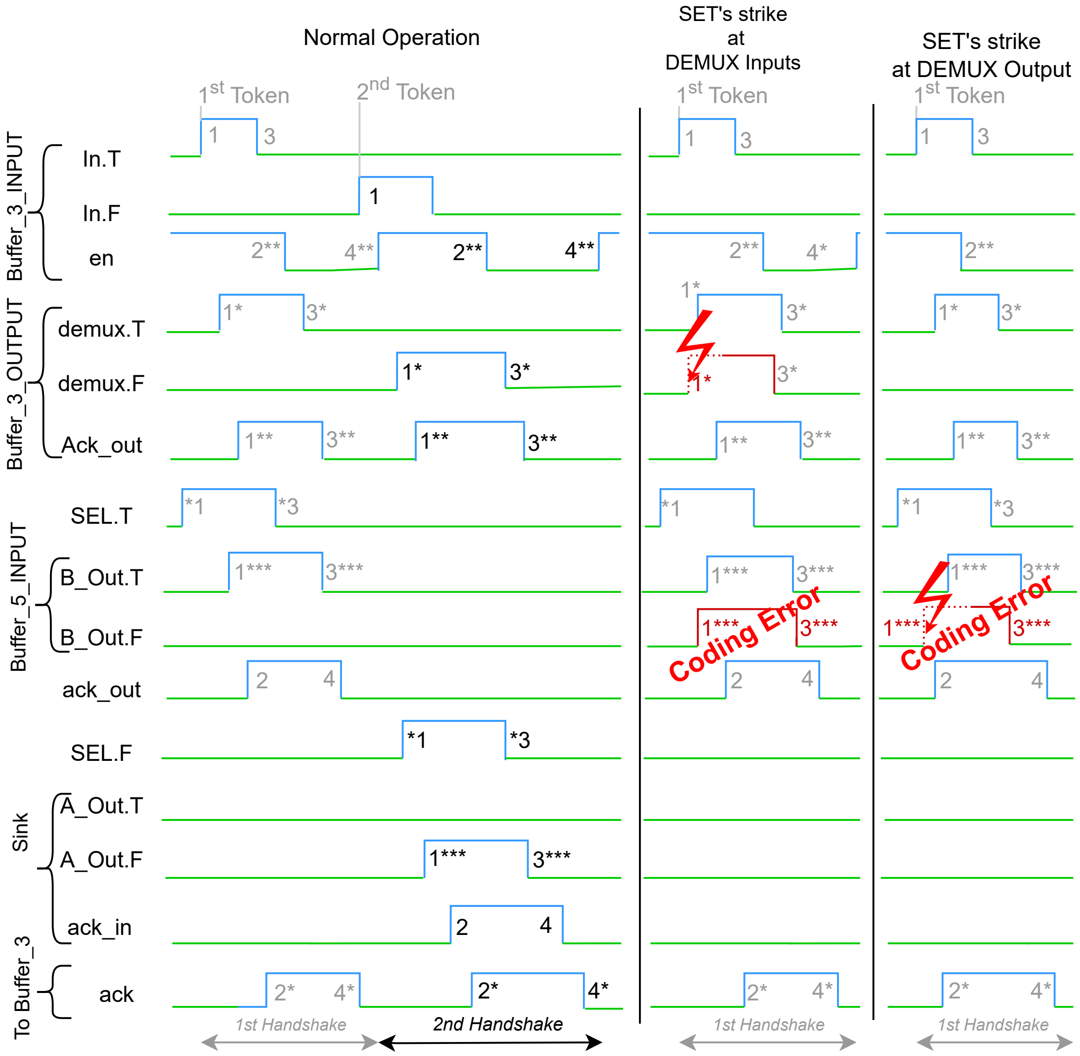


Figure 5.4: SET effects at the DeMultiplexer with WCHB buffer template

at the *De-multiplexer* output *B_Out.F*. The respective MCE is in storage mode, with input emphdemux.F at zero and *SEL.T* at high. The SET is long enough to flip the state of the MCE. Again a coding error is generated and propagated to the next stages.

To quantitatively evaluate the behavior of QDI *Multiplexer* and *De-multiplexer*, we select the 16-bit iterative multiplier circuit. Figure 5.5 presents the gate count and the propagation delay of the gates. The circuit contains 1187 gates with a minimum propagation delay of 13ps and maximum of 134ps.

The susceptibility analysis for this circuit is presented in Figure 5.6. As this is an iterative

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

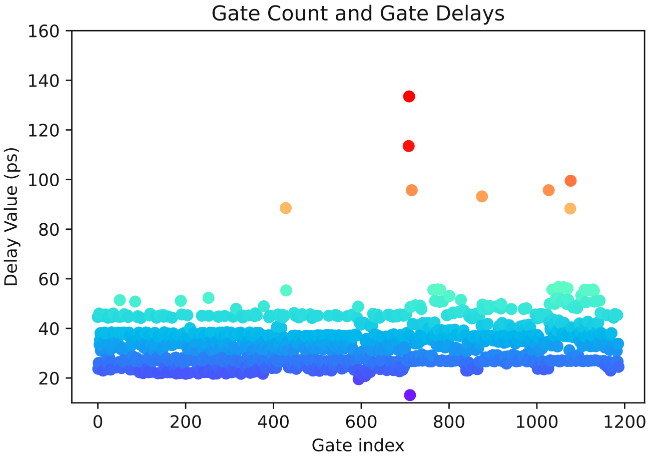


Figure 5.5: 16-bit Iterative multiplier with WCHB buffer template: Gate Count (1187) and Gate Delays

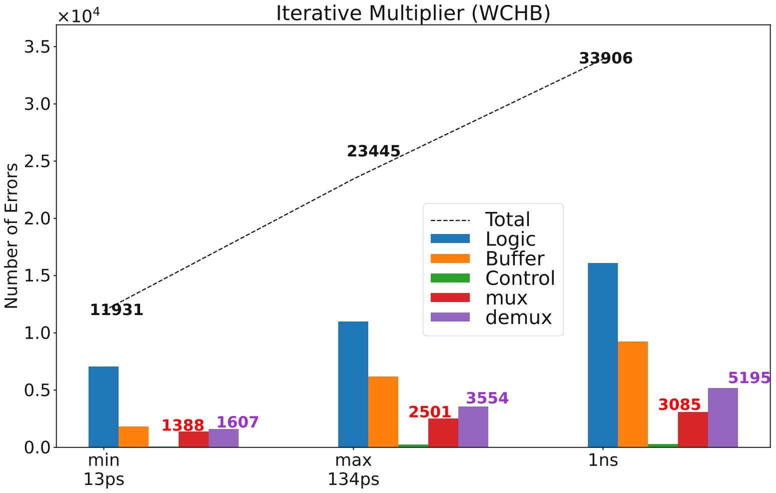


Figure 5.6: Suseptibility analysis of Iterative Multiplier with WCHB buffer template under the influence of SETs with different injection pulse lengths

multiplier, the time it takes to complete the internal iteration (number of rounds) totally depends on the input values. Our focus is to evaluate the resilience of the *Multiplexer* and *De-multiplexer* against SETs. In this iterative loop structure their speed of operation is not only controlled by the external enviornment, but rather the main control is from the delays along the internal loop.

So, adding extra delays for different PLF is not useful anymore, because both control elements' primary paths (connected with circuit input and output) remain inactive during internal iterations. We perform experiments with balanced mode operation, whenever

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

circuit generated something for the external environment source and sink react without any delay. The experiments are performed with 13, 134ps and 1ns injection pulses. The error contribution from each part of the circuit is presented with five bars, namely for combinational logic (logic), buffer, control (acknowledgement lines), *Multiplexer* (mux) and *De-multiplexer* (demux). For each injection pulse results the *Multiplexer* and *De-multiplexer* approximately contributes 25% of overall error rate.

5.2 Proposed SET resilience enhancements for QDI *Multiplexer* and *De-multiplexer*

The main problem with the *Multiplexer* we found is that if an SET affects one of the MCEs from the unselected path while in storage mode, it may deadlock the circuit for further communication. After thorough analysis of the circuit, we propose some enhancements in [TS23b] which makes the *Multiplexer* more SET resilient. The enhancements are presented with orange color in Figure 5.7. They resemble a special type of interlocking that not only, as usual, takes into account the status of the opposite rail, but also whether a valid transition passed through this path during pervious handshake cycle. As Figure 5.7 suggests, each *Multiplexer* MCE is replaced with a special-condition MCE with "sp" line. This line combines the status of the opposite rail with the acknowledgement line of the respective path. The special MCE is forcefully reset, if the select line *SEL.T or SEL.F* as well as the "sp" line is zero. This condition ensures that the path is neither selected, nor did it pass a valid token during previous handshake cycle. Consequently, anything stored by the MCE must be the effect of an SET. This time we replaced the WCHB buffer with our resilient buffer template Δ [TS22a] and we also deployed our proposed flushing technique ζ_Δ' [TSNH23] as shown in Figure 5.7.

With the aid of Figure 5.8 we will in the following discuss the regular behavior as well as the behavior under the influence of SETs.

1. Figure 5.8 Fault-Free: a new data token is received from the *Source* at *mux_B.T* (labelled with "1"). As *SEL.T* is high ("*1") it propagates to *mB.T* ("1*"). With this transition *sp4* goes high ("1**") and locks MCE-C4 for any further transitions. With this we ensure that no transition will pass from *mux_B.F* to *mB.F*. When the transition at *m.T* ("1***") is latched by *Buffer_3*, *Ack_out* (2) is generated and as *SEL.T* is high (*2) it generates *Ack_T* (2*). *Ack_F* and *sp1 and sp2* are low, which indicates that path A is not active in this handshake cycle. After the spacer of the first handshake cycle, the second data token propagation is same but this time path A is activated, so *Ack_T* keeps *sp3 and sp4* low and, combined with *SEL.T*, forcefully flushes any erroneous transition appearing at *mB.T and mB.F*.
2. Figure 5.8 SET-Strikes: *Buffer_4* produces a data token at *mux_A.F* ("1") which then propagates to *m.F* (1***). After *Ack_out* arrived (2), *Buffer_4* generates the spacer, but before it arrives at *m.F* an SET appears at *mB.F* which keeps

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

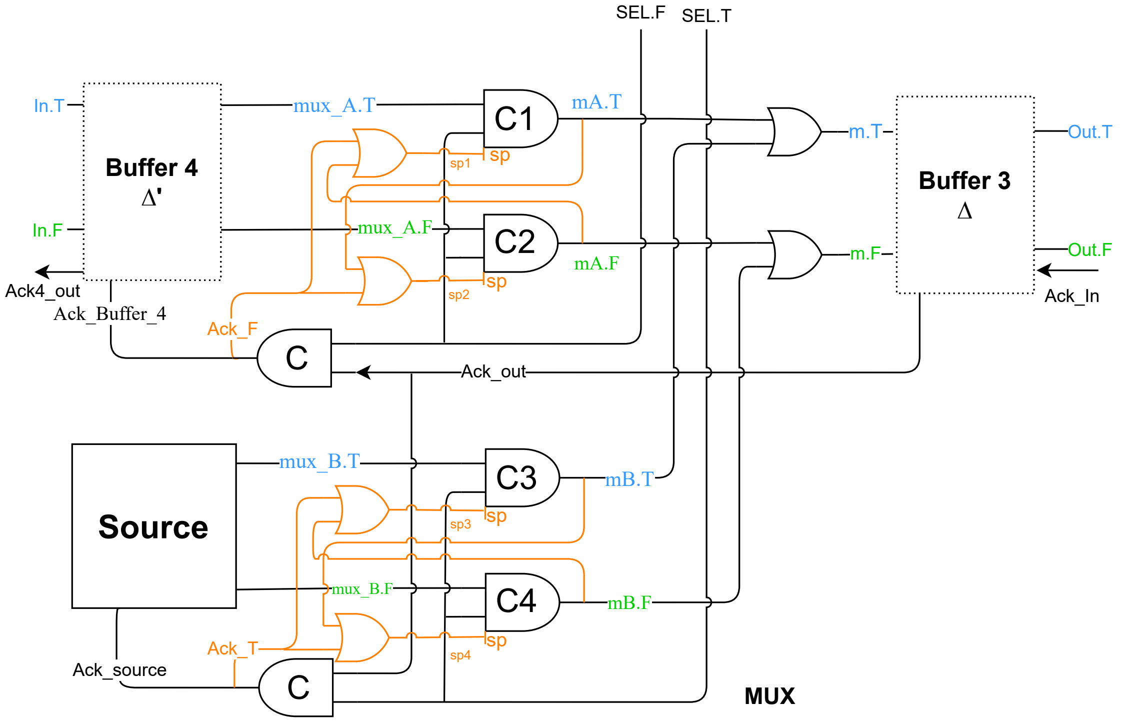


Figure 5.7: 1-bit Enhanced Multiplexer with Δ and Δ′ buffer template

the $m.F$ high after $mA.F$ (3*) went low. In contrast to Figure 5.2 this time after transient vanished *C4* flushes its effects from $mB.F$, and $m.F$ (3**) goes low. This completes the handshake cycle and we have successfully recovered our circuit from the deadlock situation.

When we move to the *De-multiplexer* we are facing different problems. As Figure 5.9 suggests, one output is directly connected with the *Sink*. So in case the faulty transition propagates to the output, if we forcefully flush it upon detection, this generates a glitch (monitors already count the transition). Note that we are not injecting at the outputs of the *De-multiplexer* path A, as these are primary outputs and directly connected with the monitor. Consequently, any fault detected at the output of path A must be a propagated one from the input side of the *De-multiplexer*. Now if we forcefully reset the transitioned MCE to enable the valid transition to proceed, there is no guarantee that the valid transition will make it in the second round either if the SET is already latched by the last buffer, in this case *Buffer_3*, in the second round it again propagated to the output generating the error. So, we decided to use the Interlocking methodology with the primary output port (path A in this case) of the *De-multiplexer* highlighted with the blue color. Whenever one MCE fires it locks the other rail for further transition. With this we can mitigate all those SET effects which arrive after the valid transitions got latched.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

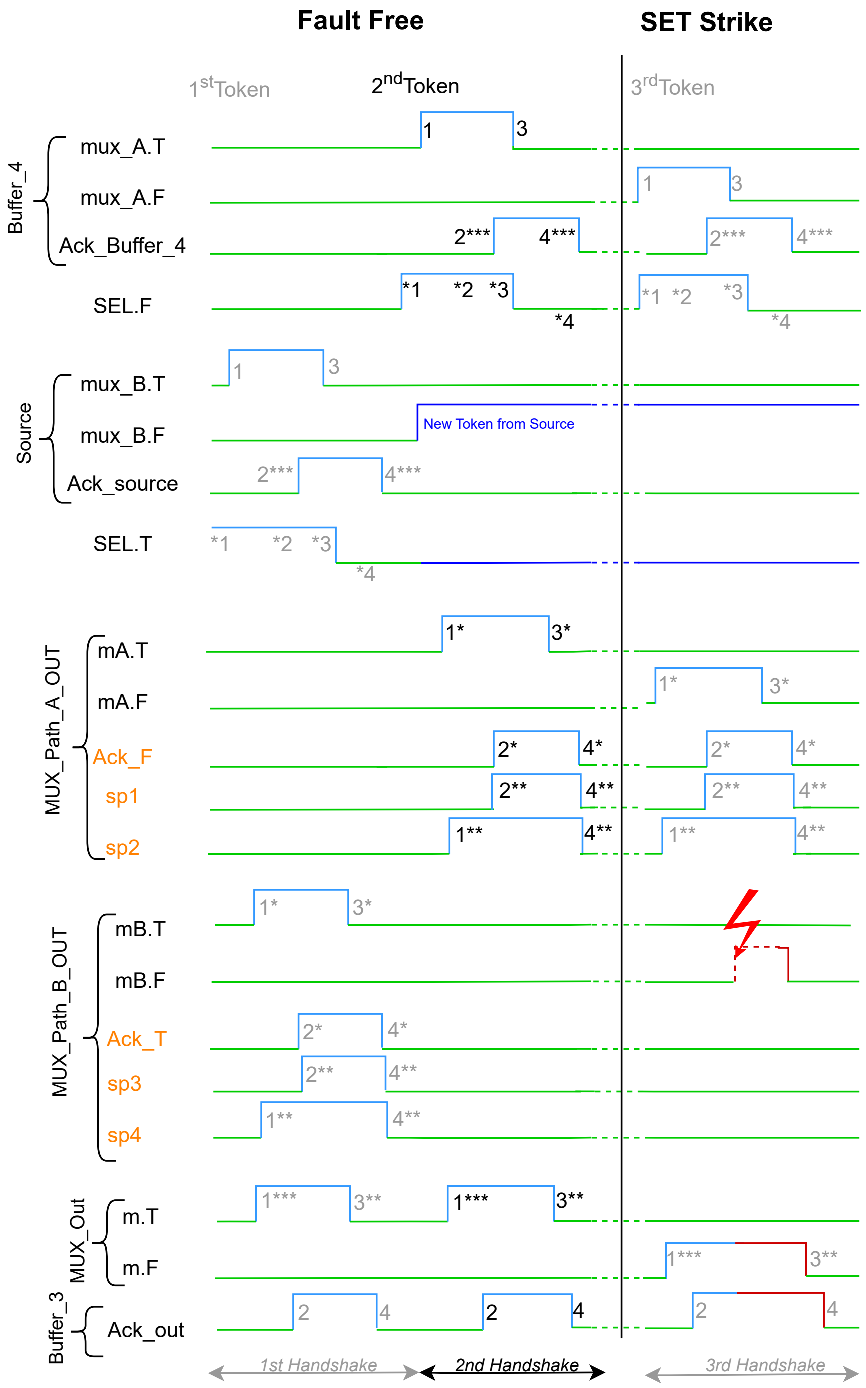


Figure 5.8: Mitigating SET effects in the Enhanced Multiplexer with Δ and Δ′buffer template

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

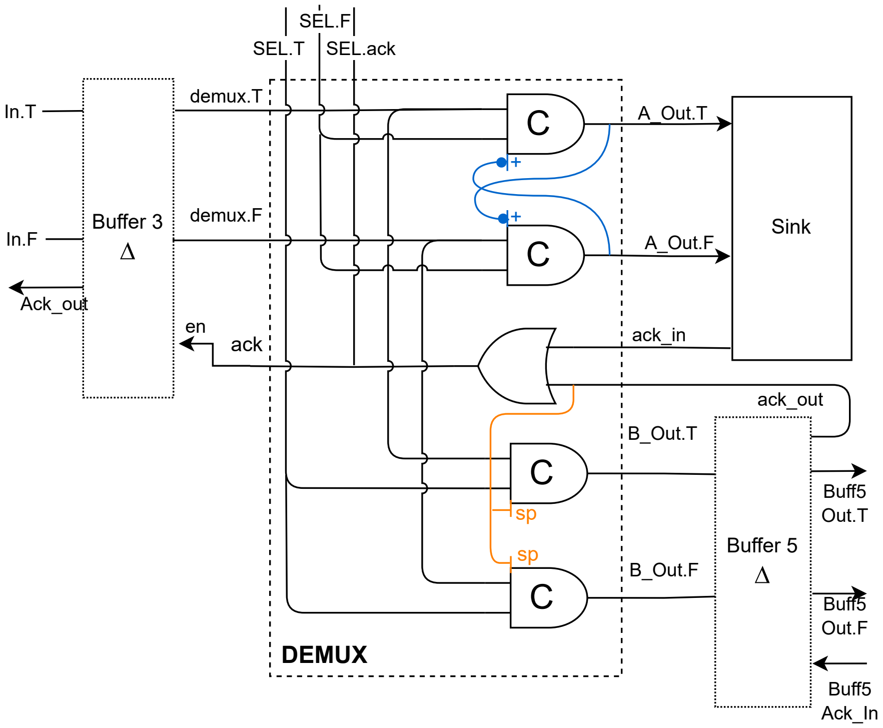


Figure 5.9: 1-bit Enhanced DeMultiplexer with Δ buffer template

The scenerios for path B are different. The latter is connected with the *Buffer_5* in our case. If an SET appears at the input of the *De-multiplexer* it may get latched by *Buffer_3*. If we utilize the same methodology as for the *Multiplexer* there, we are only able to mitigate some of the effects appearing at the *De-multiplexer*'s B output. More specifically, we can ensure the reset condition while *ack_out* as well as the *SEL.T* are zero (meaning this path is not selected). In this case SETs at *B_Out.T and B_Out.F* will forcefully be flushed out as soon as they ended (after their pulse length). Note that it is important to clear the MCEs the even if the path is currently not selected and the SET does not seem to do any harm. The reason is that when this path is again selected for communication the already latched SET will be considered as valid transition. So we flush it during the same handshake cycle.

In the following we will discuss a few scenarios shown in Figure 5.10 to illustrate how the proposed enhancements mitigate SET effects and where they fail.

1. Figure 5.10-SET's strike at DEMUX Inputs: when *SEL.F* (*1) is high the data token at the input of the *De-multiplexer demux.F* (1*) is propagated to the *Sink* via *A_Out.F* (1***). An SET appears at *demux.T* shortly after the valid transition. At this point the latter has already locked its opposite rail, which then blocks the SET from propagting to the *Sink*.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

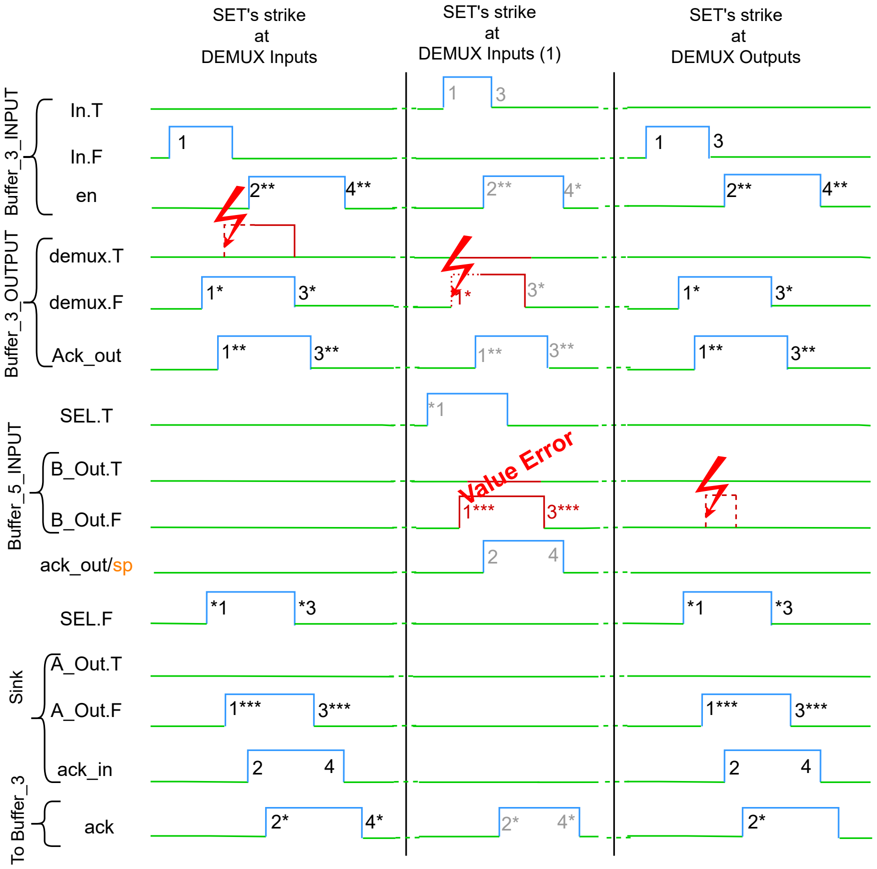


Figure 5.10: Mitigating SET effects with the Enhanced DeMultiplexer with Δ buffer template

2. Figure 5.10-SET's strike at DEMUX Inputs (1): This time the SET appears at *demux.F* slightly before the valid transition originating from *In.T* (1). As *Buffer_3* is of type Δ, the propagation path of the latter is locked by the output interlock mechanism, and we are not able to pass valid transition to *demux.T*. Instead, the faulty transition propagates to *Buffer_5* where it is considered as valid, generating the value error.
3. Figure 5.10-SET's strike at DEMUX Outputs: The scenario is same here as the first one, but this time the SET appears at *B_Out.F* (path B's output) when this path is not selected. As *sp* is zero (*ack_out*) as well as *SEL.T*, this triggers the reset condition of the respective MCE, flushing the value after the length of the SET.

From this discussion we can conclude that our proposed enhancement efficiently targets the affected rails and flushes the erroneous value without using any extra circuitry. For a

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

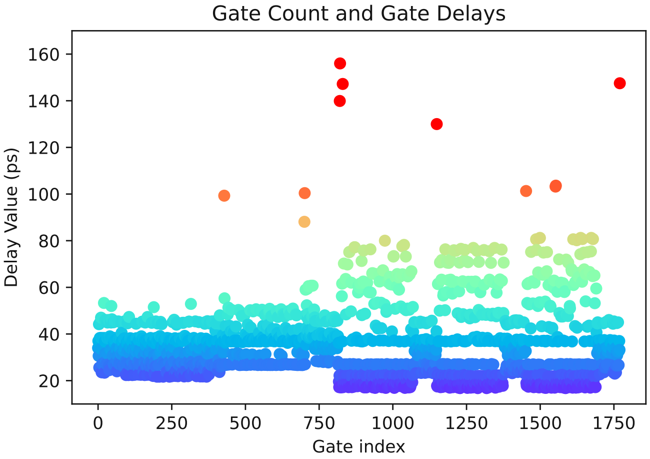


Figure 5.11: 16-bit Iterative multiplier with Δ buffer template and Enhanced Multiplexer and Demultiplexer: Gate Count (1771) and Gate Delays

quantitative assessment, this time we perform the simulations with two variants of the 16-bit Iterative multiplier with enhanced *Multiplexer* and *De-multiplexer*, first with the Δ buffer template and then with the ζ_Δ′ approach (to flush the combinational logic on the detection of the SET) with Δ. Here it is important to note that all buffer templates are type Δ, except those in front of the combinational logic; those are Δ′.

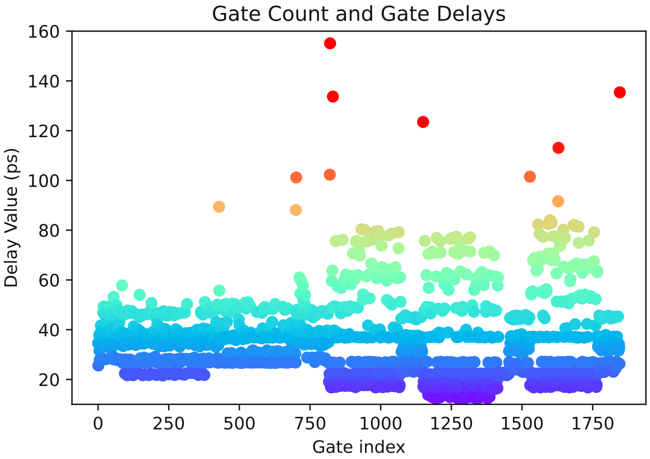


Figure 5.12: 16-bit Iterative multiplier with ζ_Δ′ approach and Enhanced Multiplexer and Demultiplexer: Gate Count (1847) and Gate Delays

Figure 5.11 presents the gate count and gate delays for the circuit with the Δ buffer template. The total gate count is 1771. Compared to the WCHB variant we only added 584 gates for the enhanced conditional control elements and resilient buffer template.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

The maximum and minimum gate delays are 155 and 12ps, respectively. We call this approach Δ_E.

With the ζ_Δ' approach the gate count is 1847 Figure 5.12. This is an increase of 76 gates, due to the combinational flushing approach. The maximum and minimum gate delays are 155 and 12ps respectively, the same as with the previous approach. This approach is termed $\zeta\_\Delta'\_E$. All these approaches are published in [TS23b].

To evaluate these proposed techniques, we again select three different injection pulses, namely minimum and maximum gate delay a circuit contains, as well as a 1ns pulse. The circuit operates in balanced mode (PLF=1). Figure 5.13 presents the results of the fault injection experiments we conducted with two variants of our target circuit.

First, we compare the results of the *Multiplexer* and *De-multiplexer* from Figure 5.6 with the enhanced versions Figure 5.13. Generally, the enhancements yield high resilience: For the injection pulse length of 155ps for Δ_E and $\zeta\_\Delta'\_E$ and 134ps for WCHB iterative multiplier (corresponding to the respective maximum delay), the error rate approximately drops from 25% to 15% (only considering the *Multiplexer* and *De-multiplexer* contributions). For the minimum pulse length we achieve even higher improvements: The error rate drops from 25% to 6% for Δ_E and 13% for $\zeta\_\Delta'\_E$.

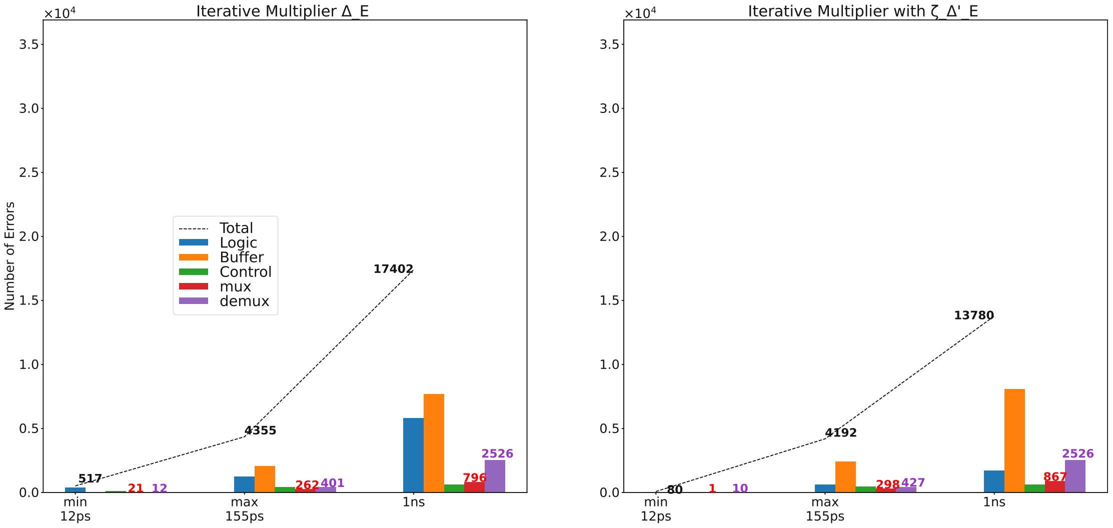


Figure 5.13: Suseptibility analysis of Iterative multiplier with Δ and ζ_Δ' approach and Enhanced Multiplexer and Demultiplexer under the influence of SETs with different injection pulse lengths

When we move to the 1ns pulse, for Δ_E the error rate is 19% and for $\zeta\_\Delta'\_E$ it is 24%, compared to 25% with the simple WCHB iterative multiplier. But a closer inspection of the results of Figure 5.6 and Figure 5.13 hints that the enhanced conditional control elements are showing much more resilience than the simple ones. For instance, while injecting the maximum pulse 134ps in Figure 5.6, the *Multiplexer* fails 2501 times where

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

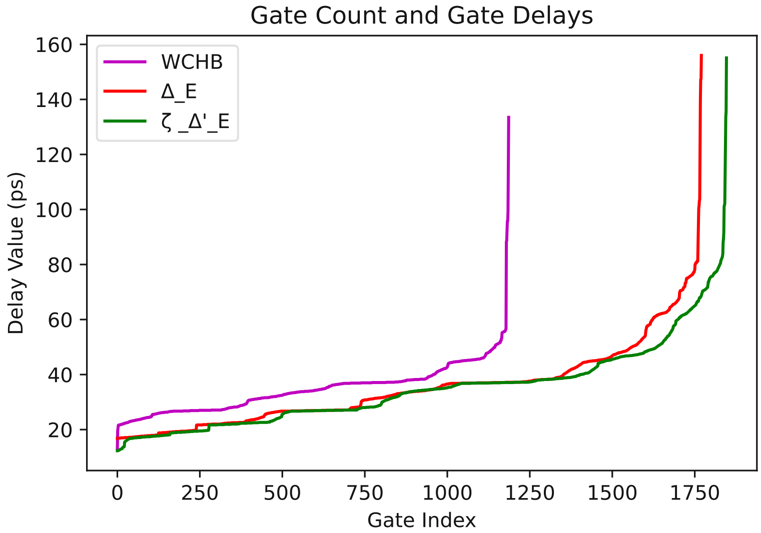


Figure 5.14: Gate Count and Gate Delays Comparison of all dicussed approaches

with enhanced versions it only fails 262 and 298 times, respectively, for 155ps pulses (see Figure 5.13). With the enhanced conditional control elements, we also utilize the resilient buffer template and combinational flushing logic in Figure 5.13 which enhances the overall resilience. If we compare the two variants presented in Figure 5.13, $\zeta\_\Delta'\_E$ shows smaller contribution from the combinational logic (blue bars). This is as expected, since this approach has been specifically designed to reduce the SET susceptibility of the combinational logic circuit.

5.3 Final Comments on Δ_E and $\zeta\_\Delta'\_E$

Figure 5.14 presents a comparison of the area (gate count) for all the discussed approaches in this chapter. Each column in Figure 5.15 presents the one-to-one resilience comparison between all the discussed target circuits. For comparison we add WCHB-based circuit, the Δ_E with our most resilient buffer template from the Chapter 3 and the $\zeta\_\Delta'\_E$ our best approach for flushing the combinational logic from Chapter 4.

On the top x-axis of Figure 5.15, we present the number of injections for each target circuit and on the bottom x-axis the injection pulse length, where we consider three different injection pulses minimum, maximum gate delay of the circuit and 1ns pulse. The number of injection is different for all target circuits: for WCHB it is 20×10^4 and for Δ_E its approximately 40×10^4. This is due to the fact that the number of gates and nodes is higher than the WCHB, and it requires more time to complete the same number of tokens. For $\zeta\_\Delta'\_E$ it is even a bit higher and touches 51×10^4. Each row presents the different types of errors a circuit produces when it fails to mitigate the SET effect.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

With the enhancement proposed we approximately halve the number of deadlocks moving from WCHB to Δ_E. When we further go to the $\zeta\_\Delta'\_E$ the behavior similar with respect to deadlocks, but with a higher number of injections. For the glitches we only experience negligible counts with the basic WCHB already. Thanks to the interlocking mechanism utilized to harden the *De-multiplexer*'s output to the *Sink* there is no way for the coding errors to propagate to the output. The first transition getting latched simply locks the buffer's opposite rail for further transition. This is evidenced by the results, where only the plain WCHB shows coding errors. While these errors may be converted into value errors, Δ_E and $\zeta\_\Delta'\_E$ still show as little as 4% and 2% value errors, respectively, compared to the 14% coding errors reported for WCHB when injecting 1ns pulse. But for the minimum injection pulse length the enhanced circuits exhibit excellent resilience against SETs, showing approximately 0% error rate.

With all these discussions we can conclude that our proposed enhancements prove efficient in making the conditional control elements more resilient against SETs.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

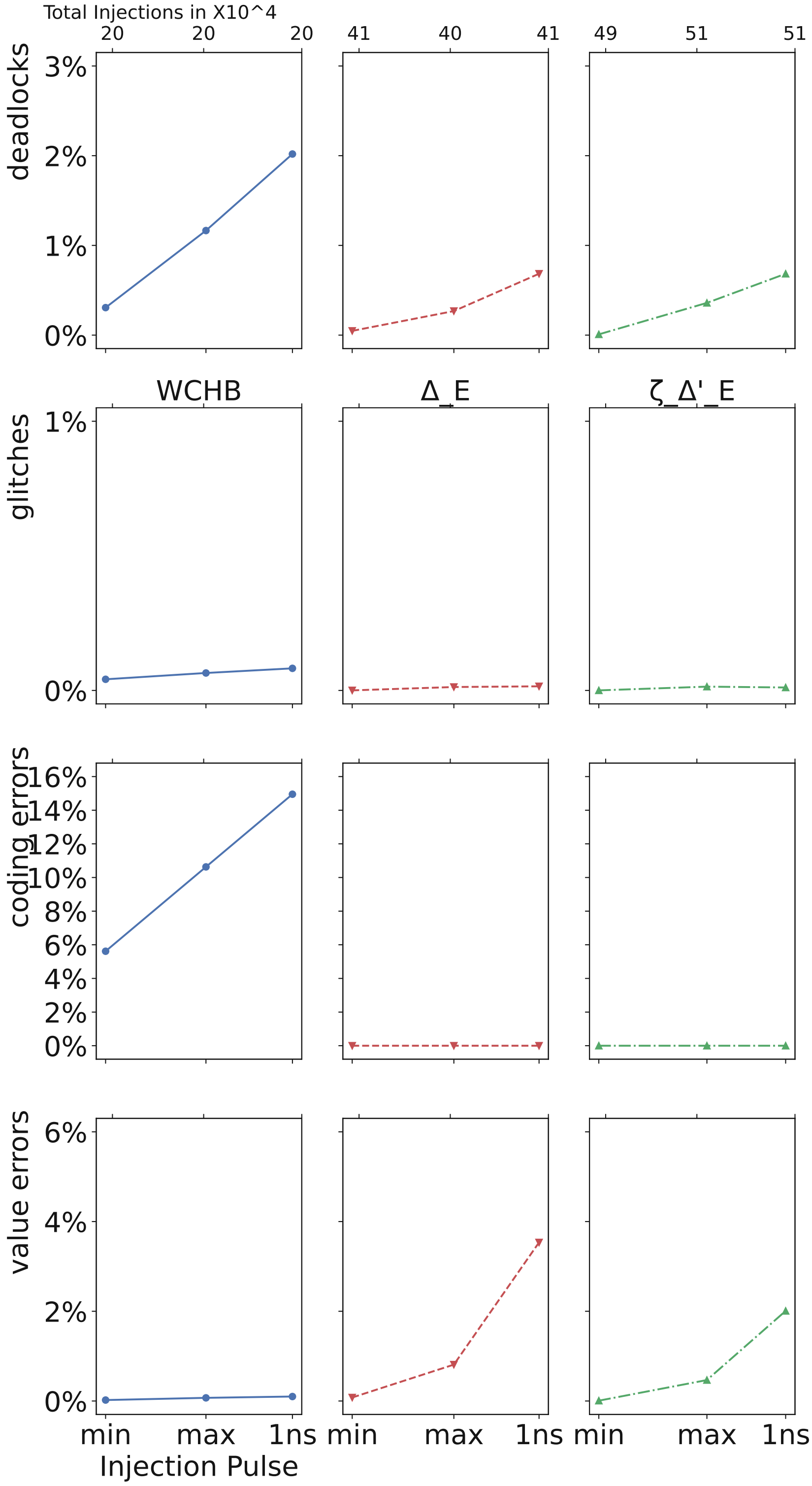


Figure 5.15: Comparing $\zeta\_\Delta'\_E$ with all discussed approaches

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

CHAPTER 6

Conclusion and Future Work

In this chapter we briefly summarize the contributions of this thesis. The main theme was to first investigate and then reduce the susceptibility of QDI circuits against SETs. Specifically we looked into WCHB buffer templates, DIMS combinational logic and QDI conditional control elements. Due to their DI nature, these circuits' operation is not effected by arbitrary response time of their environment. We have considered this variable response time by means of the PLF which was proven to have a heavy impact on the fault tolerance of the circuit, as it determines the windows that are susceptible to SETs.

6.1 Resilience of Buffer templates

According to our first research question we started out in Chapter 3 with the empty pipeline configuration for our experiments to get a better understanding of the buffer templates' behavior under SETs with variable PLFs. To cover all relevant corners, we chose different lengths for the fault pulses we injected, ranging from the minimum gate delay a circuit contains to twice the maximum gate delay. As an aid for our analysis, we made a classification of erroneous behaviors into different types and presented these separately in our results. Among others, this allowed us to observe how different error classes occur more frequently for different PLF. With respect to the PLF we distinguished three modes of operation, namely Token limited mode (PLF < 1), Balanced mode (PLF $= 0$) and Bubble limited mode (PLF > 1). For each of these we selected three error cases that we analyzed in detail. While these three cases did not cover the whole space, they helped us understand which nodes are sensitive when, how an SET gets latched and produces an error, and which factor decides the type of error being triggered.

To keep this detailed discussion feasible, we selected relatively simple and basic target circuits. For the buffer analysis we chose a 2-bit, 2-stage empty pipeline QDI circuit (with different buffer templates) for the discussions, while we performed our experimental assessment on an 8-bit, 4-stage empty pipeline QDI circuit with different buffer templates.

We used an existing tool [BHNS21] to perform the fault injection simulations at randomly selected nodes and points in time, where the number of injected faults was carefully aligned with the size (number of gates) and speed (time to complete a given number of computations) of the target. In this way we considered the fact that adding provisions for error mitigation to a circuit at the same time also makes it more complex and prone to fault occurrences.

After thorough analysis of the basic WCHB as well as its key constituent element, namely the MCE, we moved to the Interlocking WCHB [HNS20], an already existing fault tolerant derivative of the WCHB that is based on mutual interlocking of the two rails pertaining to the same data signal. While this approach is elegant in leveraging coding information for fault tolerance, virtually without any extra hardware, our analysis showed that rather than mitigating all coding errors, as expected, it tends to simply convert an appreciable share of them into value errors.

Based on insights gained from a more comprehensive analysis of different other fault-tolerant pipeline templates [BHN$^+$21], we proposed our first own buffer solution, namely the Input/Output-Interlocking WCHB buffer template [TBNS21], thus moving to the second research question. Its key mission is to establish the buffer as a firewall between two adjacent pipeline stages. To this end, we introduced an extra level of interlocking at the buffer input. As our thorough analyis unveiled, this approach was quite effective in bubble limited mode, while it showed lower performance in token limited mode. Its further limitations are its area overhead (close to 100%) as well as the use of extra MCEs, both of which make it more prone to SET impacts.

So naturally our next step was directed towards a simplification of the circuit while keeping its mitigation properties. This led to the proposal of the Input Output Interlocking with SR latch WCHB (IOISRWCHB) template [TS22b]. Its key ingredients are the replacement of the undesired MCE through an SR latch, which removes sensitive nodes from the circuit, as well as the addition of an input filter that removes glitches and better moderates the mutual exclusion. An experimental analysis of the circuit evidenced its substantially better resilience in token limited mode (as deisred), while keeping the good resilience of its predecessor for bubble limited operation.

All these fault tolerant approaches proposed up to that point somehow utilized the inherent fault-tolerance properties of QDI circuits without the full duplication of the circuit. They took benefit from dual-rail encoding which helps them to lock the buffer from the input and output side on the first rail transition. Investigations on IOISRWCHB directed us to a new direction for achieving higher resilience – there is more information available from the circuit that can be used to enhance the resilience. As the buffer template is more susceptible to SETs while its SETs are in storage mode, reducing their storage time directly impacts the resilience. To leverage this, we took inspiration from a respective existing approach (normally closed pipeline latch [BS09]) to enhance our buffer templates. More specifically, the input interlocking got more refined with a single SR latch and *Dual_Completion_Detection* from the outputs of the input filter. With these modifications our buffer, called Δ, remained in combinational mode most

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

of the time without any area overhead to its predecessors the new template showed a further increased resilience in token limited mode, while keeping similar resilience as IOISRWCHB in bubble limited mode. For extreme bubble limited mode, however, the results still indicated some residual weakness: approximately 4% of errors were reported for all circuits proposed so far. The main reason is that the token arrives earlier than the *Acknowledgment*, changing the buffer state from combinational mode to the storage mode and thus making it sensitive to SETs. What still would be needed is to block the newly arrived token from reaching the MCEs until the respective *Acknowledgment* arrived.

6.2 Combinational logic Flushing Techniques

In Chapter 4 we moved to the third research question and addressed the contribution of the combinational logic. As the latter also contains MCEs, which turned out to play a major role in SET susceptibility, our expectation was that protection will be needed. To investigate that in more detail, we again performed our fault-injection experiments, starting out with the basic WCHB buffer template, again in a 2-bit, 2-stage pipeline configuration. This time, according to the purpose, we added DIMS combinational logic, specifically an AND and an XOR gate in between these stages for the discussion part. For the experiments our target now was a 16-bit, 7-stage pipelined multiplier. The experiment settings were the same, but this time we also injected slightly longer pulses with 1ns and 2ns length to also cover potentially longer (multi-gate) propagation delays, and thus more pronounced electrical masking, through the combinational logic.

After having thus established a reference for the unprotected case, and having confirmed the significant contribution of combinational logic to the overall error rate, we took a closer look at respective mitigation approaches from the literature. For the BISEC approach [KZYD10] that essentially proposes completely flushing the combinational logic upon detection of an error, our experimental results showed no improvement in resilience. In fact, coding errors were traded for deadlocks. A modified version, SETAPII, published in [LHHA12, LHHA16], employs protocol phase detection in order to refine the resetting. However, in our experiments it performed even worse, while, as our analysis showed, also violating the handshake protocol by generating extra spacers upon pipeline flushing.

In a next attempt, we assessed the capabilities of our resilient buffer template Δ for mitigating fault effects originating in the combinational logic. After all, our resilient buffers with their rigid interlocking had been designed to prevent SET effects from propagating. Indeed, Δ showed reasonable resilience with negilgible overhead relative to the basic WCHB. We also checked the combination of WCHB with BISEC but did not yield any improvement, in spite of a doubled gate count. We continue our analysis of combinational logic with our Δ approach, as this approach can mitigate the SET effects from the combinational logic with its rigid input interlock mechanism. With negligible area overhead compared to WCHB based circuit, it shows high resilience. We also check the resilience of SETAPI approach by using the Δ buffer template with it, but with double gate count the results are not better than the simple Δ based circuit. Based

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

on the findings from our analyses we proposed a combinational logic flushing approach called ζ that selectively flushes those MCEs within the combinational logic only that potentially hold an erroneous transition. This approach was based on the insight that MCEs holding valid data are in combinational mode while those holding an erroneous value are in storage mode. This targeted flushing not only prevented the generation of erroneous spacers but also saved the time to propagate the correct token through the whole logic for recovery. On the illegal code word detection, we also disabled the MCE passing the *Acknowledgement* signal to the buffer which prevented the latching of the erroneous values. By finally combining this ζ approach with our Δ buffer, we could even largely mitigate the residual problem of an early arriving *Acknowledgement*. This approach that we called ζ_Δ', was able to cut the error rate (from the combinational logic) to 0% with approximately 12% area overhead (gate count) compared to Δ, by carefully utilizing the inherent fault-tolerance properties of QDI circuit.

6.3 Towards the resilient Conditional Control Elements

Conditional control elements of QDI circuits also contain MCEs, whichs make them susceptible to SETs. According to our fourth research question, we evaluated the resilience of these elements to SETs in Chapter 5. This time we selected an iteartive multiplier for evaluation purposes, so we had *Multiplexer* and *De-Multiplexer* in our target circuit. As the speed of the iterations was defined by the internal movement of the data token in this structure, environment delays played a minor role. Therefore we restricted our parameter space to PLF = 1. We performed experiments with three different injection pulses, namely minimum and maximum gate delay a circuit contains and a 1ns pulse. Even though our experimental results indicated a contribution of around 25% of overall error rate for the conditional control elements, we were not able to find respective mitigation approaches in the literature. Our analysis suggested that the unselected path is the root cause of most of the errors we observed. After closely examining the situation for a WCHB based design, we proposed some modification to the *Multiplexer*. In essence, our approach triggers a flushing only after considering whether the path in question is currently selected and whether it currently holds a token. With this modification we could cut the error contribution of the *Multiplexer* from 12% to 1% (in some cases).

For the *De-Multiplexer* its deployment turned out to be essential. If it is part of an internal circuit, we could utilize the same idea, but for the case that its output paths were primary outputs, we suggested the use of the Interlocking mechanism from [HNS20].

After integrating these approaches with our resilient buffer template Δ and the combinational logic flushing technique ζ_Δ' we obtained a respectable SET resilience for the complete circuit – and so finally achieved the overall goal of the thesis.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

6.4 Comparing SET Effects in Quasi Delay Insensitive and Synchronous Circuits

In this thesis we kept the focus on the resilience of QDI circuits against SETs. In our larger research context, however, we also evaluated this in comparison with synchronous logic [TS23a]. This work is not included in this thesis as full chapter form, but here we briefly discuss the results to make the picture complete. Our target circuit was a pipelined multiplier with different bit widths including 4*4, 8*8 and 16*16. We considered the WCHB buffer template with DIMS combinational logic (plus we also considered NCLX logic [KL02]) in comparison with a simple synchronous logic without any fault tolerant techniques applied. In this comparison we adopted a different method to select the injection pulses. More specifically, we aligned the injection pulse lengths with different percentages of the circuit's computation steps (clock/handshake cycle time). The analysis concludes that asynchronous QDI circuits show better resilience against SETs due to two main reasons: (1) if realized with a 4-phase handshake protocol they are resilient to negative fault pulses, and (2) the susceptibility of a QDI circuit is largely unchanged for increasing fault length because of the causality underlying the QDI principle. For instance, injecting a pulse with length of 32% of 157ps (clock period of synchronous circuit) shows 10% error rate where the QDI circuit with 32% of 1200ps shows approximately 18% error rate. However, when we inject 157ps injection pulses to the same synchronous circuit, the error rate touches 30%, where for QDI circuit with 1200ps pulses it remains 18%. This is due to the fact that the flip flops in synchronous circuits are edge triggered, and consequently the probability increases that the SET pulse gets latch if its longer than the clock period. In fact, their sensitive windows are always centered around the clock edges, regardless of fault origin and environment. In contrast to that, in asynchronous settings SET length does not matter but the injection time is crucial. If the SET appeared in the sensitive window (when the MCE is in storage mode) it will definitely alter the behavior, while if it occurs outside of the sensitive window it may widen the handshake cycle (extra delay) without corrupting the data token. Delays, however, can be easily managed by the delay-insensitive circuit design. Our analysis provided insights leading towards more resilient QDI circuits: if we only make a circuit or specific gates better resist ”1” faults, we are fully resilient towards the SETs because ”0” faults are already filtered out by its inherent behavior. This is also beneficial for area efficiency: as asynchronous circuits often require already twice the area (or more) and computation time compared to synchronous circuits, adding extra SET mitigation with full replication or other buffer redundant techniques tends to result in painful overheads. Being able to focus on specifc protection of circuit blocks against ”1” faults, as indicated by our analysis, can hence yield important savings.

6.5 Future Work

This thesis only focuses on the resilience of the WCHB buffer template and its fault-tolerance with DIMS combinational logic. It would be interesting to extend this for other

TU Bibliothek Your knowledge hub WIEN
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

buffer templates and logic designs. Combining all approaches together with ζ_Δ' for the linear configuration, the remaining errors are from buffer template. These could be further addresses by somehow managing to delay an incoming token during bubble limited mode such that it reaches the buffer only as soon as the enable signal arrives there. The findings of [TS23a] show that the WCHB mitigates negative fault pulses, so as a next step one could investigate how to modify the proposed buffer templates to make them also tolerate these. With all these extensions establishing full resilience against SETs seems within reach. With respect to the non-linear configuration the conditional control elements are still not fully protected against SETs, in next step it seems promising to investigate their behavior with different settings as they require different protection measures depending on their position in the circuit.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

List of Figures

2.1 Isochronic fork assumption: reprinted from [BOF10] 7
2.2 MCE and its derivatives normal behavior 8
2.3 MCE and its derivatives under SETs 10
2.4 Dual rail encoding (a) 4-phase, (b) 2-phase 11
2.5 Completion detection logics (a) 4-phase signaling with 1-bit message, (b) 4-phase signaling with m-bit message, (c) 2-phase signaling with m-bit message 12
2.6 DIMS AND Gate 12
2.7 NCL AND Gate 13
2.8 A: 2-bit, 2-stage QDI pipelined circuit realized with Weak-Conditioned Half Buffer (WCHB) buffer template, B: Waveform of Circuit in A 14
2.9 PLFall 16
2.10 PLFall 17
2.11 Fault Injection Simulation Environment 18
2.12 Empty Pipeline Circuit Configuration 20
2.13 16-bit 7 Stage Pipelined Multiplier 20
2.14 Iterative Multiplier 21

3.1 Gate Count (128) and Gate Delays of 8-bit, 4-stage Empty Pipeline Circuit with WCHB as Buffer Template 24
3.2 Susceptibility analyses of 8-bit, 4-stage Empty Pipeline Circuit with WCHB under the influence of SETs with variable PLF and different injection pulse lengths 25
3.3 Sensitive Windows of WCHB 26
3.4 Main Causes of Errors During Token Limited Mode 27
3.5 Main Causes of Errors during Normal Operation 28
3.6 Main Causes of Errors During Bubble Limited Mode 29
3.7 Error Rate of WCHB based 8-bit, 4-stage Empty Pipeline Circuit with and without last Stage Enable Signal in Injection List, with 60ps injection pulse 30
3.8 2-bit, 2-stage QDI pipelined circuit realized with Interlocking WCHB buffer template 32
3.9 Sensitive windows of Interlocking WCHB 32
3.10 InterlockingWCHB gate count (128) and gate delays 33
3.11 Suseptibility analyses of InterlockingWCHB under the influence of SET with variable PLF and different injection pulse lengths 34

3.12 Sensitive Windows of Interlocking WCHB during Token Limited Mode . . 36
3.13 Sensitive Windows of Interlocking WCHB during balanced operation . . . 37
3.14 Sensitive Windows of Interlocking WCHB during Bubble Limited Mode . 38
3.15 Suseptibility analyses of WCHB and Interlocking WCHB under the influence of SET (Injection pulse 60ps) 39
3.16 InOutInterlock WCHB 40
3.17 Sensitive Windows of InOutInterlock WCHB 41
3.18 InOutInterlock WCHB Gate Count (252) and Gate Delays 42
3.19 Suseptibility analyses of InOutInterlockWCHB under the influence of SETs with variable PLF and different injection pulse lengths 43
3.20 Sensitive Windows of InOutInterlock WCHB during Token Limited Mode 44
3.21 Sensitive Windows of InOutInterlock WCHB during balanced operation . 46
3.22 Sensitive Windows of InOutInterlock WCHB during Bubble Limited Mode 47
3.23 WCHB with InOutInterlocking WCHB (Injection pulse 60ps) 49
3.24 IOISRWCHB 50
3.25 Sensitive Windows of IOISRWCHB 52
3.26 IOISRWCHB Gate Count (256) and Gate Delays 53
3.27 Suseptibility analyses of IOISRWCHB under the influence of SET with variable PLF and different injection pulse lengths 54
3.28 Sensitive Windows of IOISRWCHB during Token Limited Mode 55
3.29 Sensitive Windows of IOISRWCHB during balanced operation 56
3.30 Sensitive Windows of IOISRWCHB during Bubble Limited Mode 57
3.31 Comparison with IOISRWCHB (Injection pulse is equal to 2*(maximum gate delay)) 59
3.32 DualCD InOut WCHB Δ 60
3.33 Sensitive Windows of Δ 61
3.34 DualCD InOut WCHB Gate Count (252) and Gate Delays 62
3.35 Suseptibility analyses of Δ under the influence of SET with variable PLF and different injection pulse lengths 63
3.36 Sensitive Windows of Δ during Token Limited Mode 65
3.37 Sensitive Windows of Δ during balanced operation 66
3.38 Sensitive Windows of Δ during Bubble Limited Mode 68
3.39 Gate Count and Gate Delays Comparison of all dicussed buffer templates 69
3.40 Δ comparison with all discussed buffer templates (Injection pulse is equal to 2*(maximum gate delay)) 71

4.1 WCHB buffer with DIMS AND and XOR logic 74
4.2 SET effects in combinational logic without any protection deployed 75
4.3 16-bit 7 stage Pipelined multiplier with WCHB buffer template Gate Count (2102) and Gate Delays 76
4.4 Suseptibility analyses of Pipelined multiplier with WCHB buffer template under the influence of SET with variable PLF and different injection pulse lengths 77

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

4.5 SETAPI approach with DIMS AND and XOR logic 79
4.6 SET effects from combinational logic with SETAPI approach 80
4.7 16-bit 7 stage Pipelined multiplier with WCHB buffer template and SETAPI approach Gate Count (3389) and Gate Delays 81
4.8 Suseptibility analyses of Pipelined multiplier with WCHB buffer template and SETAPI approach under the influence of SET with variable PLF and different injection pulse lengths 82
4.9 SETAPI comparison with WCHB based Pipelined Multiplier (Injection pulse is 1ns) 84
4.10 SETAPII approach with DIMS AND and XOR logic 86
4.11 SET effects from combinational logic with SETAPII approach 87
4.12 SETAPII approach Flush logic 88
4.13 Behavior of SETAPII with and without SET strike 90
4.14 16-bit 7 stage Pipelined multiplier with WCHB buffer template and SETAPII approach Gate Count (3389) and Gate Delays 91
4.15 Suseptibility analyses of Pipelined multiplier with WCHB buffer template and SETAPII approach under the influence of SET with variable PLF and different injection pulse lengths 92
4.16 Comparison of SETAPII with all discussed approaches (Injection pulse is 1ns) 94
4.17 Δ with DIMS AND and XOR logic 95
4.18 SET effects in combinational logic with theΔ buffer template 96
4.19 16-bit 7 stage Pipelined multiplier with Δ buffer template: Gate Count (2798) and Gate Delays 97
4.20 Suseptibility analyses of Pipelined multiplier with Δ buffer template under the influence of SET with variable PLF and different injection pulse lengths 98
4.21 Comparison of all discussed approaches for Δ-based pipelined multiplier circuits (Injection pulse is 1ns) 99
4.22 SETAPI approach with Δ buffer template with DIMS AND and XOR logic 100
4.23 SET effects in combinational logic with the SETAPI approach combined with the Δ buffer template 101
4.24 16-bit 7 stage Pipelined multiplier with Δ buffer template and SETAPI approach: Gate Count (4060) and Gate Delays 102
4.25 Suseptibility analyses of the Pipelined multiplier with Δ buffer template with SETAPI approach (SETAPI_Δ) under the influence of SETs with variable PLF and different injection pulse lengths 103
4.26 SETAPI_Δ (SETAPI with Δ buffer template) comparison with all discussed approaches (Injection pulse is 1ns) 105
4.27 ζ approach with Δ' buffer template with DIMS AND and XOR logic . . . 106
4.28 SET effects in combinational logic with the ζ approach combined Δ' buffer template 108
4.29 16-bit 7 stage Pipelined multiplier using Δ' buffer template with ζ approach: Gate Count (3142) and Gate Delays 109

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

4.30 Suseptibility analysis of the Pipelined Multiplier using the Δ' buffer template with the ζ approach under the influence of SETs with variable PLF and different injection pulse lengths 110
4.31 Gate Count and Gate Delays Comparison of all dicussed approaches 111
4.32 Comparison of the ζ approach with Δ' pipelined multiplier circuits with all discussed approaches (Injection pulse is 1ns) 113

5.1 1-bit Multiplexer with WCHB buffer template 116
5.2 SET effects at the Multiplexer with WCHB buffer template 117
5.3 1-bit DeMultiplexer with WCHB buffer template 118
5.4 SET effects at the DeMultiplexer with WCHB buffer template 119
5.5 16-bit Iterative multiplier with WCHB buffer template: Gate Count (1187) and Gate Delays 120
5.6 Suseptibility analysis of Iterative Multiplier with WCHB buffer template under the influence of SETs with different injection pulse lengths 120
5.7 1-bit Enhanced Multiplexer with Δ and Δ' buffer template 122
5.8 Mitigating SET effects in the Enhanced Multiplexer with Δ and Δ'buffer template 123
5.9 1-bit Enhanced DeMultiplexer with Δ buffer template 124
5.10 Mitigating SET effects with the Enhanced DeMultiplexer with Δ buffer template 125
5.11 16-bit Iterative multiplier with Δ buffer template and Enhanced Multiplexer and Demultiplexer: Gate Count (1771) and Gate Delays 126
5.12 16-bit Iterative multiplier with ζ_Δ' approach and Enhanced Multiplexer and Demultiplexer: Gate Count (1847) and Gate Delays 126
5.13 Suseptibility analysis of Iterative multiplier with Δ and ζ_Δ' approach and Enhanced Multiplexer and Demultiplexer under the influence of SETs with different injection pulse lengths 127
5.14 Gate Count and Gate Delays Comparison of all dicussed approaches . . . 128
5.15 Comparing $\zeta\_\Delta'\_E$ with all discussed approaches 130

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

List of Algorithms

2.1 MCE Model 9

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Bibliography

[BBN22] Davide Bertozzi, Kshitij Bhardwaj, and Steven M. Nowick. An asynchronous soft macro for ultra-low power communication in neuromorphic computing. In *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 178–181, 2022.

[Beh21] Patrick Behal. Quantitative comparison of the sensitivity of delay-insensitive design templates to transient faults. Master's thesis, TU Wien, 2021.

[BHN+21] Patrick Behal, Florian Huemer, Robert Najvirt, Andreas Steininger, and Zaheer Tabassam. Towards explaining the fault sensitivity of different qdi pipeline styles. In *2021 27th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 25–33, 2021.

[BHNS21] Patrick Behal, Florian Huemer, Robert Najvirt, and Andreas Steininger. An automated setup for large-scale simulation-based fault-injection experiments on asynchronous digital circuits. In *2021 24th Euromicro Conference on Digital System Design (DSD)*, pages 541–548, 2021.

[BOF10] Peter A Beerel, Recep O Ozdag, and Marcos Ferretti. *A designer's guide to asynchronous VLSI*. Cambridge University Press, 2010.

[BS09] W. J. Bainbridge and S. J. Salisbury. Glitch Sensitivity and Defense of Quasi Delay-Insensitive Network-on-Chip Links. In *15th IEEE Symposium on Asynchronous Circuits and Systems*, pages 35–44, May 2009.

[CRS21] Ney Laert Vilar Calazans, Taciano Ares Rodolfo, and Marcos LL Sartori. Robust and energy-efficient hardware: The case for asynchronous design. *Journal of Integrated Circuits and Systems*, 16(2):1–11, 2021.

[DIBM03] Monica Donno, Alessandro Ivaldi, Luca Benini, and Enrico Macii. Clock-tree power optimization based on rtl clock-gating. In *Proceedings of the 40th annual Design Automation Conference*, pages 622–627, 2003.

[DWO+21] Mike Davies, Andreas Wild, Garrick Orchard, Yulia Sandamirskaya, Gabriel A. Fonseca Guerra, Prasad Joshi, Philipp Plank, and Sumedh R. Risbud. Advancing neuromorphic computing with loihi: A survey of results and outlook. *Proceedings of the IEEE*, 109(5):911–934, 2021.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[FB96] KM Fant and SA Brandt. Null convention logictm: a complete and consistent logic for asynchronous digital circuit synthesis proceedings of international conference on application specific systems. *Architectures and Processors*, 1996.

[FCMG13] Véronique Ferlet-Cavrois, Lloyd W. Massengill, and Pascale Gouker. Single event transients in digital cmos—a review. *IEEE Transactions on Nuclear Science*, 60(3):1767–1790, 2013.

[Fri01] Eby G Friedman. Clock distribution networks in synchronous digital integrated circuits. *Proceedings of the IEEE*, 89(5):665–692, 2001.

[GYB07] K. T. Gardiner, A. Yakovlev, and A. Bystrov. A C-element Latch Scheme with Increased Transient Fault Tolerance for Asynchronous Circuits. In *13th IEEE International On-Line Testing Symposium (IOLTS 2007)*, pages 223–230, July 2007.

[HNS20] F. Huemer, R. Najvirt, and A. Steininger. Identification and confinement of fault sensitivity windows in qdi logic. In *2020 Austrochip Workshop on Microelectronics (Austrochip)*, pages 29–36, Oct 2020.

[Hue22] Florian Ferdinand Huemer. *Contributions to Efficiency and Robustness of Quasi Delay-Insensitive Circuits.* PhD thesis, TU Wien, 2022.

[JM05] Wonjin Jang and A. J. Martin. SEU-tolerant QDI circuits [quasi delay-insensitive asynchronous circuits]. In *11th IEEE International Symposium on Asynchronous Circuits and Systems*, pages 156–165, March 2005.

[JM07] Wonjin Jang and Alain J Martin. A soft-error-tolerant asynchronous microcontroller. In *13th NASA Symposium on VLSI Design.* Citeseer, 2007.

[KHS$^+$20] F. A. Kuentzer, M. Herrera, O. Schrape, P. A. Beerel, and M. Krstic. Radiation Hardened Click Controllers for Soft Error Resilient Asynchronous Architectures. In *26th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 78–85, May 2020.

[KK20a] Israel Koren and C Mani Krishna. *Fault-tolerant systems.* Morgan Kaufmann, 2020.

[KK20b] F. A. Kuentzer and M. Krstic. Soft Error Detection and Correction Architecture for Asynchronous Bundled Data Designs. *IEEE Transactions on Circuits and Systems I: Regular Papers*, pages 1–12, 2020.

[KL02] A. Kondratyev and K. Lwin. Design of Asynchronous Circuits by Synchronous CAD Tools. In *Proceedings 2002 Design Automation Conference (IEEE Cat. No.02CH37324)*, pages 411–414, June 2002.

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

[KMM15] S. Keller, A. J. Martin, and C. Moore. DD1: A QDI, Radiation-Hard-by-Design, Near-Threshold 18uW/MIPS Microcontroller in 40nm Bulk CMOS. In *21st IEEE International Symposium on Asynchronous Circuits and Systems*, pages 37–44, May 2015.

[Kob20] Daisuke Kobayashi. Scaling trends of digital single-event effects: A survey of seu and set parameters and comparison with transistor performance. *IEEE Transactions on Nuclear Science*, 68(2):124–148, 2020.

[KZYD10] W. Kuang, P. Zhao, J. S. Yuan, and R. F. DeMara. Design of Asynchronous Circuits for High Soft Error Tolerance in Deep Submicrometer CMOS Circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 18(3):410–422, March 2010.

[LHHA12] Faiq Khalid Lodhi, Osman Hasan, Syed Rafay Hasan, and Falah Awwad. Modified null convention logic pipeline to detect soft errors in both null and data phases. In *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 402–405. IEEE, 2012.

[LHHA16] Faiq Khalid Lodhi, Syed Rafay Hasan, Osman Hasan, and Falah Awwad. Analyzing vulnerability of asynchronous pipeline to soft errors: leveraging formal verification. *Journal of Electronic Testing*, 32(5):569–586, 2016.

[LHT$^+$21] Zhiyu Li, Yuhao Huang, Longfeng Tian, Ruimin Zhu, Shanlin Xiao, and Zhiyi Yu. A low-power asynchronous risc-v processor with propagated timing constraints method. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(9):3153–3157, 2021.

[LKM10] Dong-Jin Lee, Myung-Chul Kim, and Igor L Markov. Low-power clock trees for cpus. In *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 444–451. IEEE, 2010.

[LM04] C. LaFrieda and R. Manohar. Fault detection and isolation techniques for quasi delay-insensitive circuits. In *International Conference on Dependable Systems and Networks, 2004*, pages 41–50, June 2004.

[Mar86] Alain J Martin. Compiling communicating processes into delay-insensitive vlsi circuits. *Distributed computing*, 1(4):226–234, 1986.

[MR07] M. Marshall and G. Russell. A Low Power Information Redundant Concurrent Error Detecting Asynchronous Processor. In *10th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2007)*, pages 649–656, Aug 2007.

[MRL04] Y. Monnet, M. Renaudin, and R. Leveugle. Asynchronous circuits sensitivity to fault injection. In *10th IEEE International On-Line Testing Symposium*, pages 121–126, July 2004.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[MRL05a] Y. Monnet, M. Renaudin, and R. Leveugle. Hardening techniques against transient faults for asynchronous circuits. In *11th IEEE International On-Line Testing Symposium*, pages 129–134, July 2005.

[MRL05b] Yannick Monnet, Marc Renaudin, and Régis Leveugle. Asynchronous circuits transient faults sensitivity evaluation. In *Proceedings. 42nd Design Automation Conference, 2005.*, pages 863–868. IEEE, 2005.

[NKD+21] Spencer Nelson, Sang Yun Kim, Jia Di, Zhe Zhou, Zhihang Yuan, and Guangyu Sun. Reconfigurable asic implementation of asynchronous recurrent neural networks. In *2021 27th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 48–54, 2021.

[PM05] Song Peng and R. Manohar. Efficient failure detection in pipelined asynchronous circuits. In *20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)*, pages 484–493, 2005.

[Sak21] Ashiq A. Sakib. Soft error tolerant quasi-delay insensitive asynchronous circuits: Advancements and challenges. In *2021 34th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI)*, pages 1–6, 2021.

[Sei80] Charles L Seitz. Ideas about arbiters. *Lambda*, 1(1):10–14, 1980.

[SGY+09] Kenneth S Stevens, Daniel Gebhardt, Junbok You, Yang Xu, Vikas Vij, Shomit Das, and Krishnaji Desai. The future of formal methods and gals design. *Electronic Notes in Theoretical Computer Science*, 245:115–134, 2009.

[si214] si2.org. Silvaco open-cell 15nm library v0.1_2014_06 from si2, 2014.

[SM80] Charles L Seitz and C Mead. *System timing.* Reading, Massachusetts, 1980.

[Spa20] Jens Sparsø. *Introduction to Asynchronous Circuit Design.* DTU Compute, Technical University of Denmark, 2020. Paperback edition available here: `https://www.amazon.com/dp/B08BF2PFLN`.

[Sri22] Pallavi Srivastava. *Completion Detection in Asynchronous Circuits: Toward Solution of Clock-Related Design Challenges.* Springer Nature, 2022.

[SS93] Jens Sparsø and Jørgen Staunstrup. Delay-insensitive multi-ring structures. *Integration, the VLSI Journal*, 15(3):313 – 340, 1993. Special Issue on asynchronous systems.

[TBNS21] Zaheer Tabassam, Patrick Behal, Robert Najvirt, and Andreas Steininger. Input/output-interlocking for fault mitigation in qdi pipelines. In *2021 Austrochip Workshop on Microelectronics (Austrochip)*, pages 17–20, 2021.

TU Bibliothek WIEN Your knowledge hub
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

[TPES22] Pelopidas Tsoumanis, Georgios Ioannis Paliaroutis, Nestor Evmorfopoulos, and George Stamoulis. Analysis of the impact of electrical and timing masking on soft error rate estimation in vlsi circuits. *Technologies*, 10(1):23, 2022.

[TS22a] Zaheer Tabassam and Andreas Steininger. Set hardened derivatives of qdi buffer template. In *2022 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pages 1–6, 2022.

[TS22b] Zaheer Tabassam and Andreas Steininger. Towards resilient qdi pipeline implementations. In *2022 25th Euromicro Conference on Digital System Design (DSD)*, pages 657–664, 2022.

[TS23a] Zaheer Tabassam and Andreas Steininger. Set effects on quasi delay insensitive and synchronous circuits. In *2023 IEEE European Test Symposium (ETS)*, pages 1–6, 2023.

[TS23b] Zaheer Tabassam and Andreas Steininger. Towards resilient quasi delay insensitive conditional control elements. In *2023 26th Euromicro Conference on Digital System Design (DSD)*, 2023.

[TSNH23] Zaheer Tabassam, Andreas Steininger, Robert Najvirt, and Florian Huemer. ζ: A novel approach for mitigating single event transient effects in quasi delay insensitive logic. In *2023 28th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 48–57, 2023.

[VM02] T. Verdel and Y. Makris. Duplication-based concurrent error detection in asynchronous circuits: shortcomings and remedies. In *Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT)*, pages 345–353, 2002.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Curriculum Vitae
Zaheer Tabassam

Personal Information

Email	zaheer.tabassam1@gmail.com
Mobile	+43 665 65478628
City	Vienna, Austria
ORCID	0000-0002-9656-6927
LinkedIn	www.linkedin.com/in/zaheertabassam
Language	English

Education

2021-2024	PhD Student at the Institute of Computer Engineering, TU Wien, Austria Grade: Good PhD Thesis: “Utilizing and Extending the Inherent Fault Tolerance Properties of Asynchronous QDI Circuits”
2017-2019	Master’s degree Electrical Engineering, COMSATS University Islamabad, Wah Campus, Pakistan Grade: 3.83/4 Master Thesis: “A Speed Independent Microprocessor with Delay Insensitive Pipeline Stages”
2012-2016	Bachelor’s degree Electronics, University of Haripur, Pakistan Grade: 4/4 (passed with honors) Graduation Project: “Unmanned Aerial Vehicle for Surveillance (Flight Control)”

Professional Experience

2020-2024 (Dec - Jan)	Project assistant at the Institute of Computer Engineering, TU Wien, Austria Research: “Investigating the fault tolerance properties of asynchronous Quasi Delay Insensitive circuits” Skills: Circuit Designing • Python-based circuit description framework • RTL Design • Embedded System • Fault Analysis • Designed and implemented fault injection techniques for assessing circuit resilience • Automated the entire circuit generation, test bench creation, and fault injection processes • Pioneered new approaches to enhance fault tolerance, contributing to improved system reliability • Process Design Kit • HDL (Verilog and VHDL) • Python
2019-2020 (Aug - Nov)	Senior Design Officer, Aviation Design Institute, Pakistan Aeronautical Complex, Kamra Pakistan Responsibilities: “Indigenous development of an autopilot system, Indigenous development of RADAR, integrating different modules with autopilot and architecture design of Hardware in Loop Simulation setup using National Instruments hardware” Skills: System Architecture Design • Avionic Systems Development • Protocol Tailoring • Integration • Requirements Analysis • Flight Data Analysis • Collaboration • Engine Control Unit • FPGA and ASIC Design

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

2017-2019 (Sept - Sept)	Research Associate, Dept. of Electrical and Computer Engineering, COMSATS University Islamabad, Wah Campus, Pakistan Research: "Evaluating the performance and power effectiveness of asynchronous processors customized for many-core systems" Skills: Microprocessor Design • Logic Design • Testing and Debugging • Project Deployment • FPGA Tools • Linux Environment • Report Writing

Academic Achievements/Awards

2016	Distinction in BS (Gold Medal)
2021	Best Paper Award, IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)
2022	Best Regular Paper Candidate, IEEE International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS)
2022	Outstanding Paper Award, IEEE 25th Euromicro Conference on Digital System Design (DSD)

Highlighted Courses

Advanced Digital Design • Advanced Computer Architecture • Hardware/Software Codesign • Artificial Intelligence • Computer-Aided Design of Digital Systems • Biologically Inspired Computing • Circuit Analysis • Basic Electronics • Solid State Electronics • Semiconductor Devices • Industrial Electronics • Integrated Circuit Design • VLSI Design • Complex Variable and Transforms • Solid State Electronics • Amplifiers and Oscillators • Signal and System • Digital Signal Processing • Microprocessors and Interfacing • Microcontroller and Embedded System • Opto Electronics • Power Electronics • Control System • Data Communication • Analog and Digital Communication • Microwave Communication

Technical Proficiency

Languages	C/C++, Python, MATLAB, Shell scripting, SQL, MySQL, Git, Assembly, Verilog, VHDL, BALSA
Tools	ModelSim, QuestaSim, Xilinx Vivado Design Suite, Xilinx ISE Design Suite, Cadence Virtuoso Studio, LabVIEW, VeriStand, Arduino IDE, Jupyter Notebook, BALSA asynchronous design and synthesis tool
Boards	Cyclone V FPGA, Zynq UltraScale+ MPSoC ZCU102, National Instruments CompactRIO, Rasberry Pi, Arduino
FCS	Flight Control Systems: Embention Veronte Autopilots
GNSS	Global Navigation Satellite System: NovAtel PwrPak7

Journal Papers (peer-revieved)

1. Z. Tabassam, S. R. Naqvi, T. Akram, M. Alhussein, K. Aurangzeb, and S. A. Haider. Towards designing asynchronous microprocessors: From specification to tape-out. *IEEE Access*, 7:33978–34003, 2019. doi: 10.1109/ACCESS.2019.2903126.

Conference Papers (peer-revieved)

2. P. Behal, F. Huemer, R. Najvirt, A. Steininger, and Z. Tabassam. Towards Explaining the Fault Sensitivity of Different QDI Pipeline Styles. In *27th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 25–33, 2021. doi: 10.1109/ASYNC48570.2021.00012.

Conference Papers with Talk (peer-revieved)

3. Z. Tabassam, P. Behal, R. Najvirt, and A. Steininger. Input/output-interlocking for fault mitigation in qdi pipelines. In *2021 Austrochip Workshop on Microelectronics (Austrochip)*, pages 17–20, 2021. doi: 10.1109/Austrochip53290.2021.9576871.
4. Z. Tabassam and A. Steininger. Towards resilient qdi pipeline implementations. In *2022 25th Euromicro Conference on Digital System Design (DSD)*, pages 657–664, 2022. doi: 10.1109/DSD57027.2022.00093.
5. Z. Tabassam and A. Steininger. Set hardened derivatives of qdi buffer template. In *2022 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pages 1–6, 2022. doi: 10.1109/DFT56152.2022.9962344.
6. Z. Tabassam, S. R. Naqvi, and A. Steininger. Aμflips: An asynchronous microprocessor with flexibly-timed pipeline stages. In *2022 25th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, pages 32–37, 2022. doi: 10.1109/DDECS54261.2022.9770113.
7. Z. Tabassam and A. Steininger. Set effects on quasi delay insensitive and synchronous circuits. In *2023 IEEE European Test Symposium (ETS)*, pages 1–6, 2023. doi: 10.1109/ETS56758.2023.10173866.
8. Z. Tabassam, A. Steininger, R. Najvirt, and F. Huemer. ζ: A novel approach for mitigating single event transient effects in quasi delay insensitive logic. In *2023 28th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 48–57, 2023. doi: 10.1109/ASYNC58294.2023.10239589.
9. Z. Tabassam and A. Steininger. Towards resilient quasi delay insensitive conditional control elements. In *2023 26th Euromicro Conference on Digital System Design (DSD)*, 2023.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.