

# Low Power Digital Receivers for Multi-Gb/s Wireline/Optical Communication

by

A K M Delwar Hossain

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Integrated Circuits and Systems

Department of Electrical and Computer Engineering

University of Alberta

© A K M Delwar Hossain, 2017

# Abstract

As the gate length scaling continues to less than 10nm, digital performances of integrated circuits (IC) continue to improve at a faster rate than their analog performances. This naturally leads to a trend where traditional high-speed mixed signal transceivers have to be replaced by their digital counterparts. For a wireline receiver, this means replacing traditional analog mixed signal equalizers with digital equalization techniques while still fitting within challenging power budget. Similarly, traditional optical receivers use analog mixed signal techniques to solve data dependent DC offset, burst mode timing recovery, etc. This work introduces fully digital techniques to address these challenges to design compact, low power digitally-enhanced optical receivers.

The first receiver architecture of this dissertation describes the design technique of energy-efficient sequence detection and equalization without the use of any ADC and DSP. This scheme takes advantage of the inter-symbol-interference (ISI) in the channel to reconstruct the time domain bit sequence. It is the most power efficient digital receiver reported to date. It improves power efficiency i.e. power consumed per bit by 2.5X and power consumed per bit per dB channel loss by 2.65X of the state-of-art.

The second receiver takes the architecture of the first receiver and modifies it to introduce data trace-back. This is the first-time implementation of data trace-back in SerDes. The data trace-back improves noise immunity and the voltage margin of the system. The added

data trace-back with the decision feedback equalization (DFE) improved BER from  $10^{-10}$  (DFE only) to  $10^{-12}$ .

The third receiver architecture describes a low power 7-10 Gb/s burst mode DC-coupled receiver for photonic switch networks. The concurrent operation of DC and timing recovery implies low latency in burst mode receivers. DC is recovered using SAR (successive approximation register) logic within 6 cycles of 1/8th of data rate clock, which is 6.5X improvement from the current state-of-art. It consumes only 32.7 mW during runtime.

# Preface

The concept, architecture, and measurement results of the work in Chapter 2 has been published in *Symposium on VLSI Circuits*: ““A 35 mW 10 Gb/s ADC-DSP less direct digital sequence detector and equalizer in 65nm CMOS,” A. D. Hossain, Aurangozeb, M. Mohammad and M. Hossain, 2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits), Honolulu, HI, 2016, pp. 1-2.” I was responsible for the system design, data collection, and manuscripts composition. Aurangozeb assisted me during design and data collection. Dr. Masum Hossain was the supervisor of the project.

The paper containing overall concept and measurement results of optical receiver described in Chapter 4 has been submitted for the peer review journal of *IEEE Transactions on Circuits and Systems I: Regular Papers*. ““Burst Mode Optical Receiver with 10ns Lock Time Based on Concurrent DC Offset and Timing Recovery Technique,” A. D. Hossain, Aurangozeb and M. Hossain, IEEE Transactions on Circuits and Systems I.” I was responsible for the system design, data collection, and manuscripts composition. Aurangozeb assisted me during system design, manuscript editing, and data collection. Dr. Masum Hossain was the supervisor of the project.

# Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Masum Hossain, for giving me the golden opportunity of being part of his group when I knew little about circuits. Dr. Hossain has been a great inspiration for me in last three years. His way of teaching and supervising has helped me a lot in understanding very basics of circuit design. His friendly endeavor to me made working with him productive and fun. He will be a role model that I will look up to as I start my career. Special thanks to him for providing me with proper funds to carry out my studies.

I would also like to thank Prof. Duncan Elliott and Prof. Kambiz Moez for being on my supervisory committee. I would like to thank Prof. Pedram Mousavi and CMC microsystems for lending us the equipment for testing the chips.

It has been my privilege to work with team members of Mixed-signal Integrated systems in Nano-Technology (MINT) lab at University of Alberta. I would like to thank Amlan, Aurangozeb and Waleed for their friendship and technical insight. My sincere thanks to all friends and well-wishers, particularly Manir, Ishtiza and Shibli for making my life in Edmonton enjoyable.

Thanks to my better half, Raka, for all her support and encouragement. This three year in Edmonton would be impossible without her support and patience. Finally, I thank my parents and siblings for always believing in me.

# Contents

ABSTRACT.....	II
PREFACE.....	IV
ACKNOWLEDGEMENTS.....	V
CONTENTS.....	VI
LIST OF TABLES.....	IX
LIST OF FIGURES .....	X
CHAPTER 1. INTRODUCTION .....	1
1.1. MOTIVATION.....	3
1.2. ORGANISATION OF THE THESIS .....	3
CHAPTER 2. A 10 GB/S DIRECT DIGITAL SEQUENCE DETECTOR AND EQUALIZER WITHOUT ADC-DSP IN 65NM CMOS .....	5
2.1. EQUALIZATION: MOVING FROM ANALOG TO DIGITAL .....	6
2.2. CONCEPT OF SEQUENCE DECODING .....	11
2.2.1. Reference Placement: ADC vs. Sequence Detector .....	12
2.2.2. Digital Sequence Detection.....	14
2.2.3. Advantages & Challenges in Direct Sequence Decoding .....	17
2.2.4. Sequence DFE.....	19
2.3. LINK DESIGN.....	21
2.3.1. Comparator Reference for Sequence Generation .....	23
2.3.2. Sequence Generation and DFE Implementation.....	24
2.4. HARDWARE COST AND PERFORMANCE OF THE SYSTEM.....	31
2.5. ERROR TOLERANCE.....	33
2.5.1. Bottom Prediction Error Tolerance .....	34
2.5.2. Mid Prediction Error Tolerance .....	35
2.5.3. Top Prediction Error Tolerance.....	38

2.5.4. <i>In-Bank Comparator Error Tolerance</i> .....	39
2.6. SYSTEM DESIGN.....	40
2.6.1. <i>Passive Equalizer</i> .....	41
2.6.2. <i>Sample &amp; Hold (SH)</i> .....	44
2.6.3. <i>Comparator</i> .....	48
2.6.4. <i>SR Latch</i> .....	51
2.6.5. <i>Reference Muxing Timing Issue</i> .....	52
2.6.6. <i>Sequence DFE Critical Timing</i> .....	53
2.7. IMPLEMENTATION & EXPERIMENTAL RESULTS.....	54

CHAPTER 3. A 16 GB/S DIRECT DIGITAL SEQUENCE DETECTOR WITH DATA TRACE-BACK AND EQUALIZER IN 65NM CMOS..... 60

3.1. NOISE TOLERANCE LIMIT OF CURRENT RECEIVER.....	61
3.1.1. <i>Case I</i> .....	62
3.1.2. <i>Case II</i> .....	63
3.2. IMPROVING NOISE TOLERANCE OF CURRENT RECEIVER.....	64
3.2.1. <i>Setting Fixed Data Comparators</i> .....	65
3.2.2. <i>Case I</i> .....	66
3.2.3. <i>Case II</i> .....	69
3.2.4. <i>Trace-back Sequence Generation</i> .....	70
3.2.5. <i>Conditions of Data Trace-back</i> .....	71
3.2.6. <i>Improved Noise Margin</i> .....	72
3.3. SYSTEM DESIGN.....	74
3.3.1. <i>Reference Muxing Timing</i> .....	75
3.3.2. <i>DFE and Trace-back Feedback</i> .....	76
3.4. EXPERIMENTAL RESULTS.....	77

CHAPTER 4. BURST MODE OPTICAL RECEIVER WITH 10NS LOCK TIME BASED ON CONCURRENT DC OFFSET AND TIMING RECOVERY TECHNIQUE .  
..... 82

4.1. CONVENTIONAL OPTICAL RECEIVER.....	83
4.2. CHALLENGES IN BURST-MODE RECEIVER.....	85

4.3. PROPOSED BURST MODE RECEIVER .....	90
4.3.1. <i>Trans-impedance Amplifier (TIA)</i> .....	91
4.3.2. <i>DC Recovery</i> .....	99
4.3.3. <i>Clock Recovery</i> .....	101
4.3.4. <i>Timing Skew Correction</i> .....	104
4.4. IMPLEMENTATION AND MEASUREMENT RESULTS .....	106
4.5. COMPARISON WITH STATE-OF-ART .....	111
CHAPTER 5. CONCLUDING REMARKS.....	113
5.1. FUTURE WORKS .....	116
BIBLIOGRAPHY .....	117



# List of Tables

Table 2.1: Performance Summary of the 10 Gb/s Receiver. ....	59
Table 3.1: Reference placement for checking probable bank miss. ....	68
Table 3.2: Trace-back choice logic. ....	71
Table 3.3: Detecting Strong 1/0. ....	72
Table 3.4: Receiver Summary. ....	81
Table 4.1: Performance Comparison Summary of the TIA. ....	111
Table 4.2: Performance Comparison Summary of the Receiver. ....	112
Table 5.1: Performance summary of the implemented receivers. ....	113

# List of Figures

Figure 1.1: Data rate trend in digital receivers ([3]–[12]).	2
Figure 1.2: Power efficiency of high-speed digital receivers over the years ([4], [11]–[16]).	2
Figure 2.1: Conventional wireline receiver.	6
Figure 2.2: N-tap feedforward equalization technique.	7
Figure 2.3: N-tap decision feedback equalization technique.	7
Figure 2.4: 2-tap loop unrolled DFE.	8
Figure 2.5: Conventional ADC-based solution for the wireline receivers.	9
Figure 2.6: Digital receiver using FFE and maximum likelihood sequence detector.	10
Figure 2.7: ADC-based receiver block diagram and reference placement.	12
Figure 2.8: Sequence detector reference placement depending on tap coefficient.	14
Figure 2.9: Digital sequence detection technique at $t=t_0$ and $t=t_1$ .	15
Figure 2.10: Digital sequence detection technique at $t=t_2$ and $t=t_3$ .	15
Figure 2.11: Digital sequence detection technique (summarized).	16
Figure 2.12: ADC vs. Sequence Detection.	18
Figure 2.13: Sequence DFE choices for 3 taps.	20
Figure 2.14: Example showing sequence DFE technique.	21
Figure 2.15: Channel response and single bit response before and after the passive equalizer of example transmission link.	22

Figure 2.16: References for the link. ....	23
Figure 2.17: Placement of edge comparator for prediction .....	24
Figure 2.18: Use of the edge comparators for placing the data comparators. ....	26
Figure 2.19: Sequence DFE feedback for the link.....	27
Figure 2.20: Reduction of bank comparators using DFE. ....	27
Figure 2.21: (a) Placement of floating comparators. (b) Verification of edge prediction. (c) Example sequence generation and DFE implementation.....	29
Figure 2.22: Sequence sorting process of the overall system. ....	30
Figure 2.23: Comparison of the required number of comparators among loop unrolled DFE, ADC, and sequence DFE. ....	32
Figure 2.24: Comparison of noise margin between ADC-based DFE and sequence DFE. 33	
Figure 2.25: Error tolerance of edge comparator prediction to send the floating data comparators to bottom at time $t=t_1$ . ....	35
Figure 2.26: Error tolerance of edge comparator prediction to send the floating data comparators to the middle at time $t=t_2$ . ....	36
Figure 2.27: Error tolerance of edge comparator prediction to send the floating data comparators to bottom at time $t=t_3$ . ....	37
Figure 2.28: Error tolerance of edge comparator prediction to send the floating data comparators to the top at time $t=t_3$ . ....	38
Figure 2.29: In-bank comparator error tolerance of the system using DFE. ....	39
Figure 2.30: Maximum likelihood sequence detector with passive equalization and timing recovery.....	41
Figure 2.31: Schematic of passive equalizer with its AC response for different settings. ....	42

Figure 2.32: Passive equalizer response for example link. (a) AC analysis, (b) input to the equalizer and (c) output of the equalizer.....	43
Figure 2.33: Layout of the implemented passive equalizer. ....	43
Figure 2.34: Concept of sample and hold. ....	44
Figure 2.35: Implemented sample and hold circuitry (a) and its simulated differential operation (b). ....	46
Figure 2.36: Layout of sample and hold circuit. ....	48
Figure 2.37: Schematic of the implemented comparator. ....	49
Figure 2.38: Operational stages of the comparator. ....	49
Figure 2.39: Simulation results showing sampling, regeneration, decision, and reset stages of the comparator. ....	50
Figure 2.40: Layout of the implemented comparator. ....	50
Figure 2.41: SR Latch implementation and truth table.....	51
Figure 2.42: Layout of the implemented SR latch. ....	52
Figure 2.43: Edge and data SH and comparator clocking for CH0. ....	53
Figure 2.44: Reference settling behavior. ....	53
Figure 2.45: Sequence DFE and loop unrolling for $B_{+1}$ feedback (red box). ....	54
Figure 2.46: Implemented prototype in 65nm CMOS. ....	55
Figure 2.47: Complete receiver with its power consumption. ....	56
Figure 2.48: Pin diagram and test setup of the prototype. ....	56
Figure 2.49: Measured half rate 4-bit sequence DAC output. ....	57

Figure 2.50: Measured 10 Gb/s input eye (a) and 16 levels output eye of sequence detector (b). .....	57
Figure 2.51: Measured 2.5 GHz recovered clock eye (a) and histogram (b). .....	57
Figure 2.52: BER bathtub (a) and the recovered 2.5 Gb/s PRBS (Pseudo-random bit sequence) checked data eye. ....	58
Figure 3.1: Case I – error in sequence detection. $B_0$ and $B_{-1}$ detected incorrect. ....	63
Figure 3.2: Case II – error in sequence detection. $B_{-1}$ detected incorrect. ....	64
Figure 3.3: Introduction of two fixed data comparator reference instead of fixed edge comparators. ....	65
Figure 3.4: Fixing error with little margin using fixed data comparators. ....	66
Figure 3.5: Introduction of check comparator to find out whether it may miss a bank or not. ....	67
Figure 3.6: Overall reference placement of the architecture with trace-back. ....	69
Figure 3.7: Resolving the issue of case II. ....	70
Figure 3.8: Comparison of noise margins of 4 tap sequence DFE with and without data trace-back with ADC-based DFE. ....	73
Figure 3.9: System architecture of quad-rate receiver with data trace-back. ....	74
Figure 3.10: Reference Muxing for quadrature channel running at 4 GHz. ....	75
Figure 3.11: Feedback for DFE and Trace-back. ....	76
Figure 3.12: Implemented prototype in 65nm. ....	78
Figure 3.13: Pin diagram and test setup of the prototype. ....	79
Figure 3.14: Channel Response at 8 GHz. ....	79
Figure 3.15: Measured 16 Gb/s input eye after passive equalizer (a) and 16-level output eye of the sequence decoder (b). ....	80

Figure 3.16: Measured 4 GHz clock eye (a) and PRBS checked recovered 4 Gb/s data eye (b). .....	80
Figure 3.17: BER bathtub comparing DFE and Trace-back. ....	80
Figure 3.18: 2D color-map of BER showing voltage margin in Y-axis and timing margin in X-axis of sequence DFE (a) and data trace-back (b). ....	81
Figure 4.1: Conventional vs. proposed implementation of the optical receiver. ....	84
Figure 4.2: Conventional implementation of burst mode receiver with DC offset calibration. ....	86
Figure 4.3: Settling time vs. bandwidth of DC recovery loop. ....	87
Figure 4.4: Conventional timing recovery loop. ....	88
Figure 4.5: Effect of DC on duty cycle of the data. ....	89
Figure 4.6: Block Diagram of the receiver with power breakdown. ....	91
Figure 4.7: Detailed schematic of TIA and its DC recovery loop. ....	92
Figure 4.8: TIA transfer function and its simulated and measured gain. ....	93
Figure 4.9: Measured 10 Gb/s output eye of the front end. ....	93
Figure 4.10: Simulated amplifier stage output eye with (left) and without (right) C1 & C2 for input currents of 100 $\mu$ A, 250 $\mu$ A and 600 $\mu$ A (from top to bottom). ....	94
Figure 4.11: Simulated and calculated input referred noise of TIA. ....	99
Figure 4.12: Proposed DC recovery technique. ....	100
Figure 4.13: Measured scope shot of DC recovery operation within 4.8 ns. ....	101
Figure 4.14: DC recovery without LPF and use of rising edge pulses allowing concurrent operation of DC and timing recovery. ....	102
Figure 4.15: Ring oscillator with pulse filtering and its timing diagram. ....	103

Figure 4.16: Timing Skew compensation. ....	104
Figure 4.17: Slope Detection. ....	105
Figure 4.18: Slope detection logic. ....	106
Figure 4.19: Implemented die photo in 0.13 $\mu\text{m}$ . ....	107
Figure 4.20: Pin diagram and test setup of the prototype. ....	107
Figure 4.21: Measured scope shot of DC and timing recovery with preamble and data pattern. ....	108
Figure 4.22: Recovered clock eye.....	108
Figure 4.23: Recovered clock histogram. ....	109
Figure 4.24: Phase noise plot of the recovered clock. ....	109
Figure 4.25: On-chip PRBS checked recovered 2.5 Gb/s channel data eye. ....	110

# Chapter 1.

## Introduction

Wired data communication systems can be classified into two major categories based on their medium of transmission: wireline and optical. Wireline systems use copper medium for short distance data transmission. For chip-to-chip backplane data transmission, copper is a cost-effective option. On the other hand, optical systems use fibre optic cable to carry a large amount of data over a long distance. Fibre optic cables can provide significantly higher data bandwidth than its counterpart. As a result, optical receivers are finding their place in data centers. Wireline and optical receivers implemented in the analog domain have shown excellent performance and power efficiency at high speed ([1], [2]). However, with shrinking technology nodes and increasing data rates, the performance of these receivers is not improving as much as their digital counterparts due to supply scaling, increased device leakage, increased noise and process variations. Therefore, the recent trend shows that digital implementations of these receivers are getting higher interest from the circuit design industry than their analog counterparts. Figure 1.1 shows the increasing data rates of state-of-art digital receivers over the years implying we can achieve good high-speed performance in digital receivers as well. However, digital implementations require analog-to-digital converters (ADC) followed by digital signal processing (DSP)



units which make them exceed existing transceiver power budget. Figure 1.2 illustrates power efficiency (consumed power per bit) of high-speed digital receivers over the years.

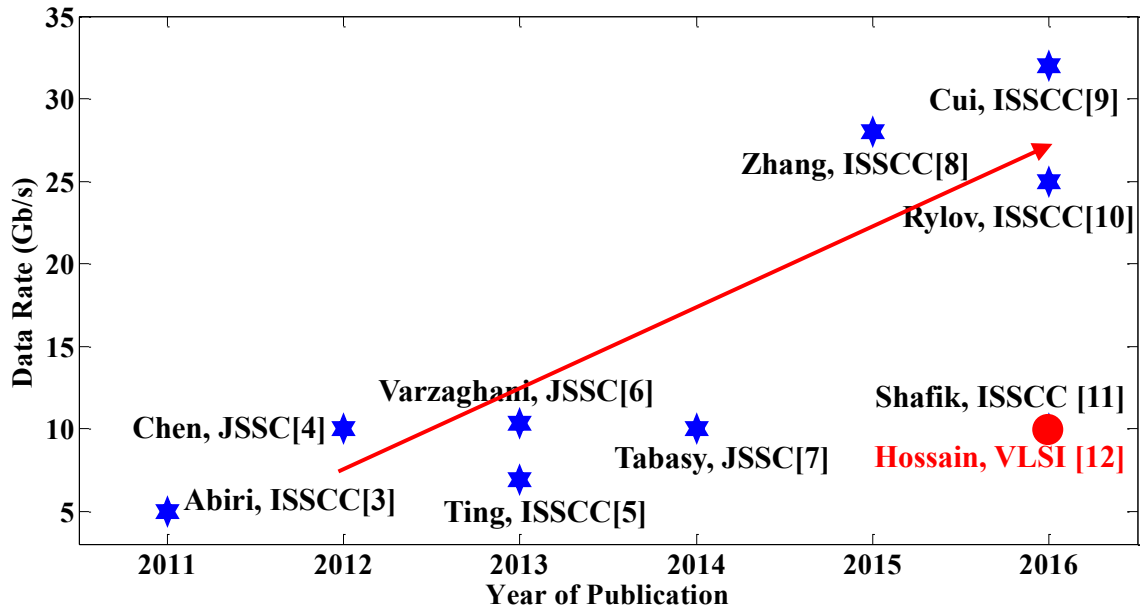


Figure 1.1: Data rate trend in digital receivers ([3]–[12]).

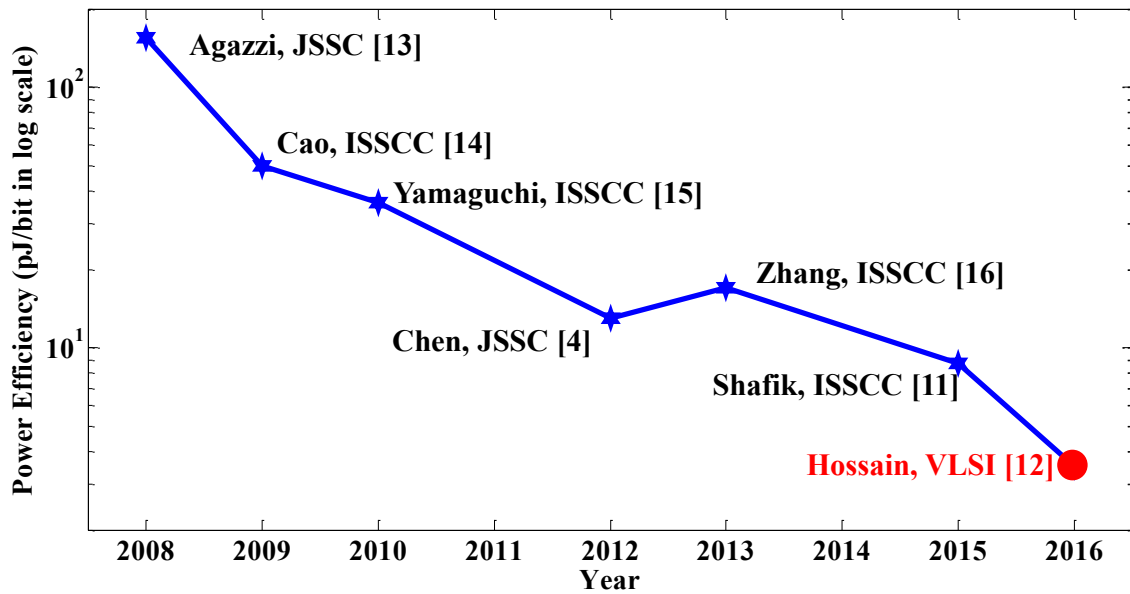


Figure 1.2: Power efficiency of high-speed digital receivers over the years ([4], [11]–[16]).

## **1.1. Motivation**

The main motivation of this thesis is to make digital receivers power efficient in both wireline and optical cases. For wireline receivers, as the high-speed data passes through the copper medium, it suffers from dielectric loss and parasitic crosstalk. The channel appears as a low pass filter to the high-frequency component of the signal and attenuates it by adding random and deterministic noise. So, at the presence of high-speed data and channel loss, un-equalized transceivers cannot provide adequate performance. Therefore, different equalization schemes have become a necessary part of receiver design. One of the motivations of this thesis is to search for a cost-effective low power equalization scheme for wireline digital receivers.

Burst mode optical receivers are used in rapidly reconfigurable photonic switch networks. This reconfiguration operation requires both burst mode DC and timing recovery in optical receivers. This thesis also concentrates on a low latency low power digital burst mode optical receiver.

Although pulse amplitude modulation (PAM) provides us with lower bandwidth requirement, for the sake of simplicity and higher signal to noise ratio (SNR), non-return-to-zero (NRZ) binary data will be used in this thesis.

## **1.2. Organisation of the Thesis**

A total of three data receivers are designed for this thesis. Two of them are focused on wireline applications and the other one is an optical receiver.

Chapter 2 describes an energy efficient 10 Gb/s sequence detector and equalizer without the use of any ADC and DSP implemented in 65nm CMOS technology. The chapter contains the concept of sequence detection and equalization using a highly digital architecture. Implemented analog front end, digital logics and timing issues of the system are also described in detail.

Chapter 3 takes the receiver design from Chapter 2 to have improved noise immunity. The receiver design is modified to have 1-bit data trace-back that provides better noise performance and voltage margin than before. Front-end and digital logics are made to handle a higher data rate of 16 Gb/s for this receiver.

Chapter 4 describes a 7-10 Gb/s burst mode optical receiver implemented in 0.13  $\mu\text{m}$ . DC recovery operation without low-pass filter enables us to have timing recovery operation going on at the same time resulting in a low latency receiver. The chapter covers analog front-end with its noise performance, successive approximation register (SAR) logic based DC recovery, injection locking based timing recovery and timing skew correction using slope detection of the signal.

Chapter 5 summarizes the performance of all three receivers and proposes future work that can be done.

# **Chapter 2.**

## **A 10 Gb/s Direct Digital Sequence**

## **Detector and Equalizer without ADC-**

## **DSP in 65nm CMOS**

This chapter describes a low power 10 Gb/s sequence detector and equalizer without the use of ADC and DSP in TSMC 65nm. The chapter begins with a brief description of the conventional receivers in Section 2.1. This section compares state-of-art mixed signal and ADC-based solutions. Section 2.2 discusses the technique of digital sequence detection and compares this technique with ADC-based solutions. The digital sequence decoding naturally leads us to sequence DFE technique. Section 2.3 takes an example transmission link and discusses sequence generation and DFE implementation techniques used in this receiver. Section 2.4 discusses hardware cost and noise margin of the system and compares it with ADC-based solutions. Section 2.5 discusses error tolerance of the receiver with examples. Section 2.6 provides details of the receiver design starting from the top-level block diagram and then descending to the overall design. This section includes description of sensitive analog front-end blocks and critical timing margins of the system. Section 2.7 shows experimental results and comparison with the state-of-art ADC-based receivers.

## 2.1. Equalization: Moving from Analog to Digital

In high-speed wireline transceivers, the frequency dependent channel loss is the main source of inter-symbol interference (ISI). In simple words, ISI is the residue of the current symbol that affects the following symbols (pre-cursor) as well as the previous symbols (post-cursor). For high loss channels, conventional receiver designs ([1], [17]–[20]) usually feature analog continuous time linear equalization (CTLE), implemented using active or passive elements, in the front end as shown in Figure 2.1. In addition, feed-forward equalization (FFE) (Figure 2.2) and decision feedback equalization (DFE) (Figure 2.3) techniques are used for further ISI cancellation and bit detection. There are feedback timing concerns for the first tap cancellation in DFE for high-speed data since this feedback has to be done within 1 unit interval (UI). Although prior work [19] has demonstrated that

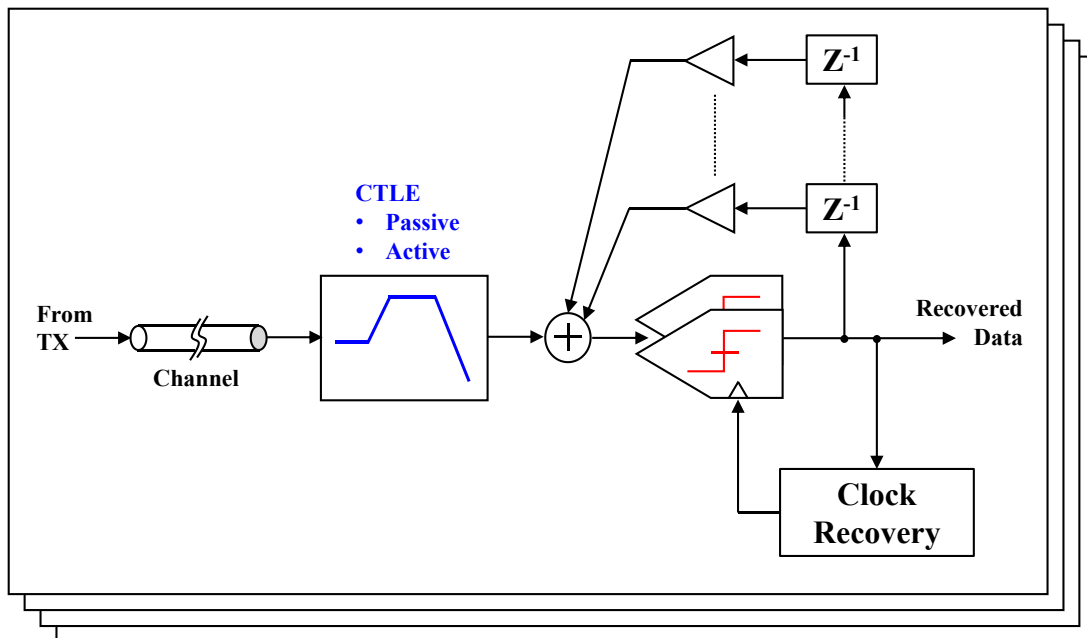


Figure 2.1: Conventional wireline receiver.

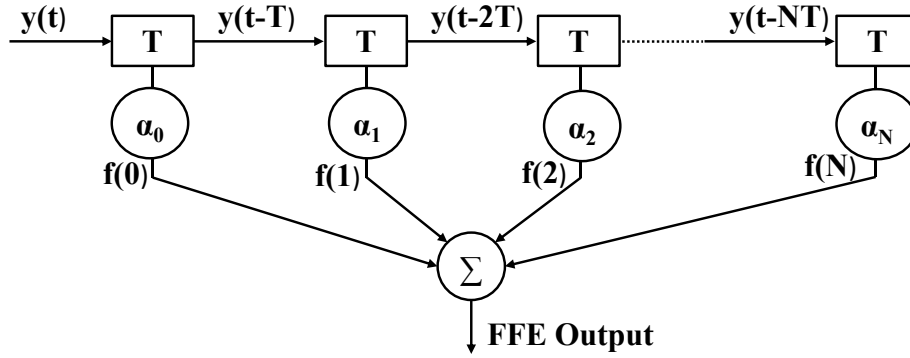


Figure 2.2: N-tap feedforward equalization technique.

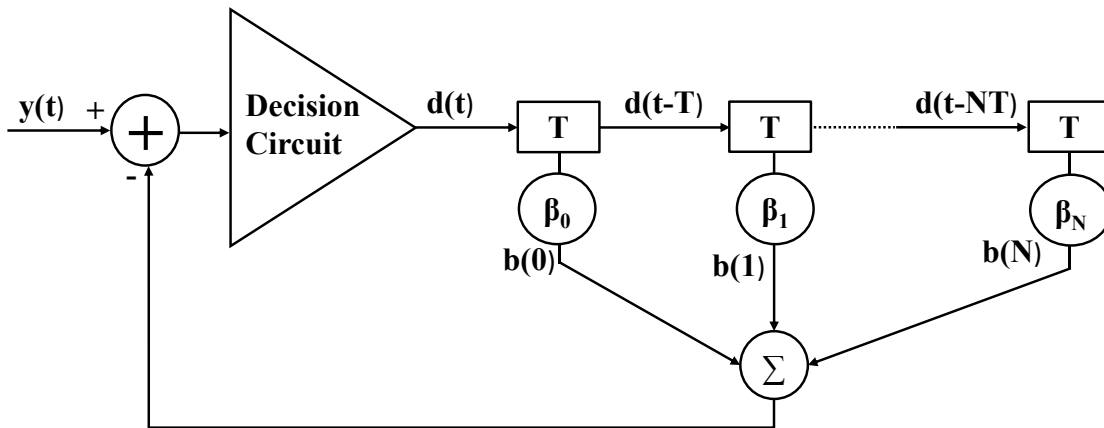


Figure 2.3: N-tap decision feedback equalization technique.

direct cancellation of the first tap is possible, but as the data rates go up, it becomes hard to comply with the DFE timing margin for the first tap. To resolve this timing issue, a commonly used approach is loop unrolled DFE ([21], [22]). In this approach, signals are pre-calculated and then one of them is selected using previously decoded bits. This approach can be extended to other taps if there are feedback timing concerns with them. Figure 2.4 shows 2-tap loop unrolled DFE architecture. The loop unrolled approach increases front-end hardware and power by two times as the number of taps with timing concerns increase.

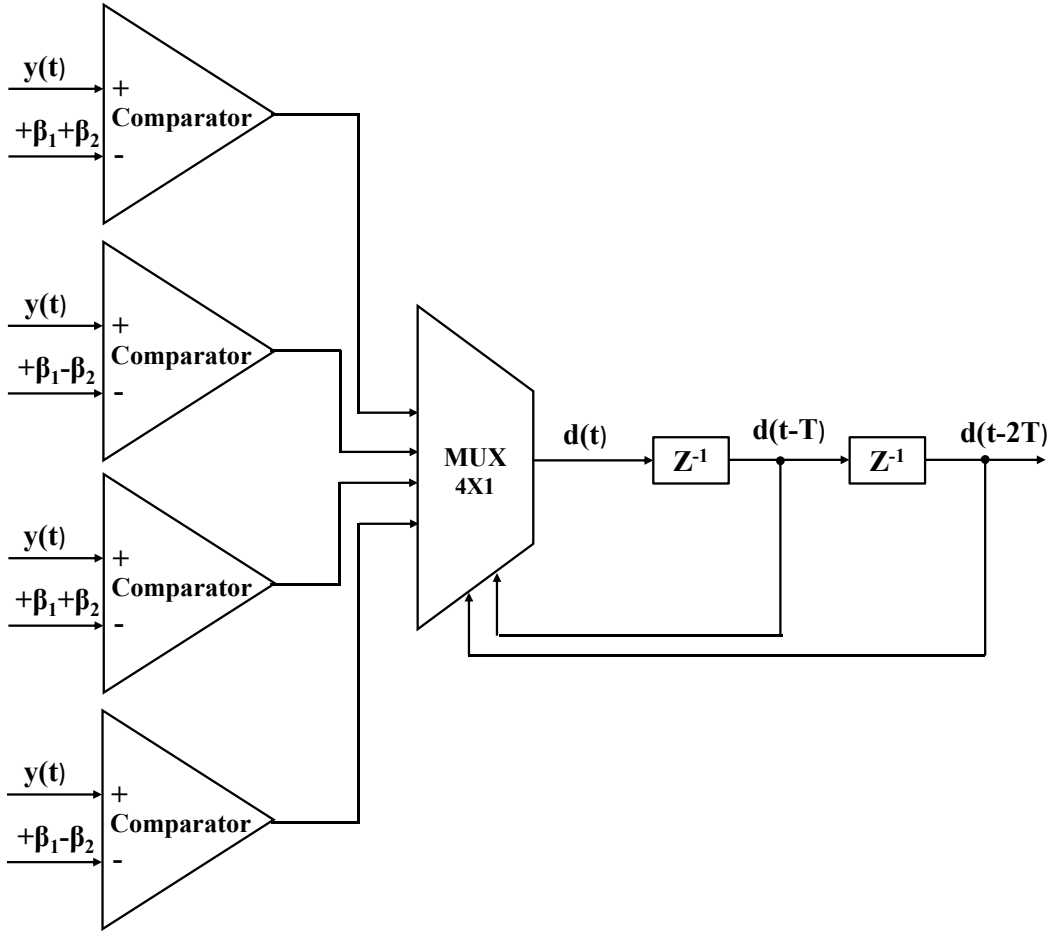


Figure 2.4: 2-tap loop unrolled DFE.

In general, analog mixed-signal solution can equalize with excellent energy efficiency (around  $\sim 3\text{pJ/bit}$  ([1], [20])). However, their performance can be limited by the analog performance of the technology such as gain-bandwidth product and comparator resolution. First, there is SNR degradation which results from the CTLE, that generally inverts the channel, which amplifies noise, including crosstalk. Second, the linearity requirement is hard to achieve - scaled supply reduces maximum achievable linear swing. Third, process variation makes it very difficult to achieve reliable control over zero and pole frequencies to achieve the desirable frequency response. All these factors limit the performance of the symbol-by-symbol detection technique. In such SNR-limited cases, sequence decoders

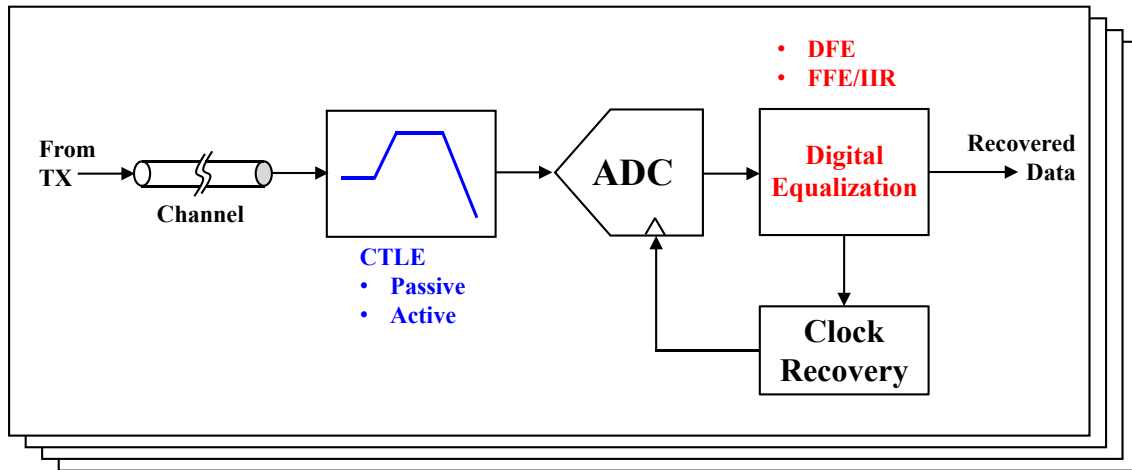


Figure 2.5: Conventional ADC-based solution for the wireline receivers.

outperform symbol-by-symbol detectors. However, existing maximum likelihood sequence detectors (MLSD) require an analog-to-digital (ADC) converter in the front-end ([13], [16]). There are previously-reported ADC-based solutions where equalization (FFE/DFE) is moved to the digital domain (Figure 2.6) ([4], [14], [16], [23]).

Harwood et. al. [23] implemented a receiver with 2-tap FFE and 5-tap DFE in digital domain using baud rate sampling ADC for 12.5 Gb/s operation. The architecture used two 6.25 Gb/s (half rate) 4.5-bit flash ADC and DSP to perform the numerical FFE and DFE to compensate 24dB channel loss with a power efficiency of 26.4 pJ/bit. Zhang et. al. [16] combined both conventional approaches by designing a dual path receiver; CTLE and DFE for high SNR input and ADC-based solution for low SNR data. The time-interleaved 6-bit ADC alone takes 195 mW for 10.3125 GS/s operation. Moreover, additional DSP consumes 500 mW [14]. In recent works ([4], [11]), it is demonstrated that ADC-based solutions can achieve a power efficiency of  $\sim 10$  pJ/bit. Chen et. al. [4] demonstrated a power efficiency of 13 pJ/bit using a four-way time interleaved 4-bit flash ADC. A three-



stage continuous time high pass filter and a 2-tap FFE were implemented in the analog front-end and a 5-tap DFE was implemented in digital domain to equalize 29 dB channel loss. Shafik et. al.[11] demonstrated a power efficiency of 8.7 pJ/bit using a 32-way time interleaved 6-bit SAR ADC. A 4-tap FFE and 3-tap DFE were implemented in digital domain for compensating channel loss up to 25.3 dB.

In [13], Agazzi et. al. demonstrated the first maximum likelihood sequence detector (MLSD) for multimode fibers (Figure 2.6). The reported sequence decoder outperforms symbol-by-symbol detectors by at least 2 electrical dB for 10.3125 Gb/s operation. The analog front-end of the digital receiver incorporates an 8-way time-interleaved 10-stage pipelined ADC with self-calibration where each path has two slices. At any given time in one path, one slice is in normal operation mode and the other one is in calibration or power down mode. For on-chip DSP, the outputs of the ADCs are further demultiplexed by a factor of 2. The digital back-end of the receiver has a nonlinear MIMO (multiple-input,

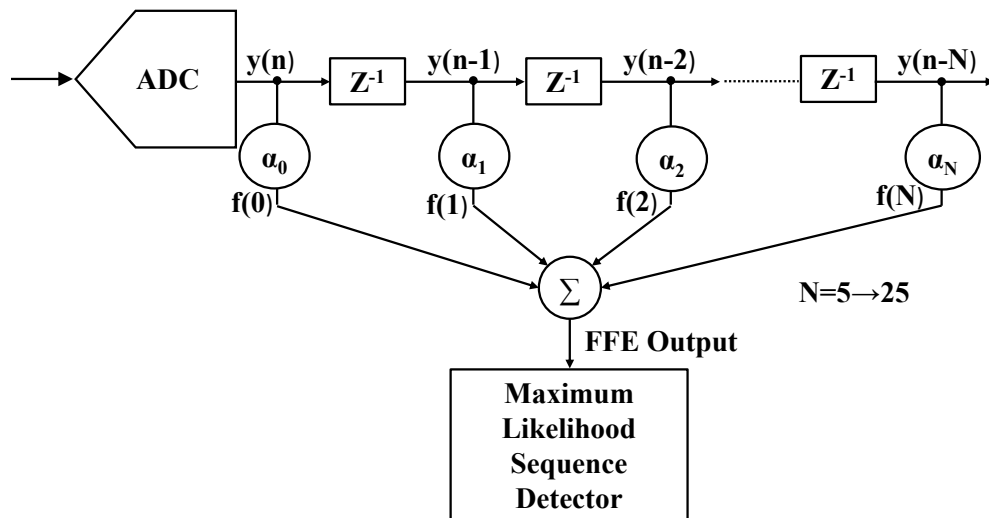


Figure 2.6: Digital receiver using FFE and maximum likelihood sequence detector.

multiple-output) channel estimator, a 5-25 tap FFE and an 8-state sliding block Viterbi decoder (SBVD).

In all these cases, while ADC alone approaches 10 pJ/bit power consumption, including the DSP [16] significantly exceeds the receiver power budget for SerDes solution space. Therefore, a solution without ADC and DSP is an attractive low-power alternative to existing approaches.

## **2.2. Concept of Sequence Decoding**

Any high-speed data going through a lossy medium suffers from ISI, crosstalk, and random noise. In most cases, ISI is the most dominant factor contributing to channel loss. Due to ISI, we can say the channel has memory. This memory can extend up to hundreds of previous symbols and a few following symbols. For simplicity, we can consider a channel where ISI is limited to within three taps - one pre-cursor ( $h-1$ ), main ( $h_0$ ) and one post-cursor ( $h+1$ ). We can assume a transmitter from which bits are transmitted, and a receiver receives those bits with ISI after passing through the lossy channel. The received signal is essentially the result of convolution between the transmitted bit symbols with the impulse response of the channel. Therefore, baud spaced sampled values will be the combination of these three taps. An example of the transmitted bit sequence and the received signal after channel loss are illustrated in Figure 2.7 which will be used throughout this section to discuss the concept of sequence detection.

### 2.2.1. Reference Placement: ADC vs. Sequence Detector

For ADC-based solutions, if we consider a flash ADC, the received signal is fed to a comparator bank (Figure 2.7). The total signal space is divided into  $2^N$  levels for N-bit ADC to cover the whole dynamic range. As we are considering 3-taps of ISI for sequence detector, we can also consider a 3-bit ADC where the signal space will be divided into  $2^3=8$  levels. The comparator bank output is ideally thermometric in nature. A thermometric to

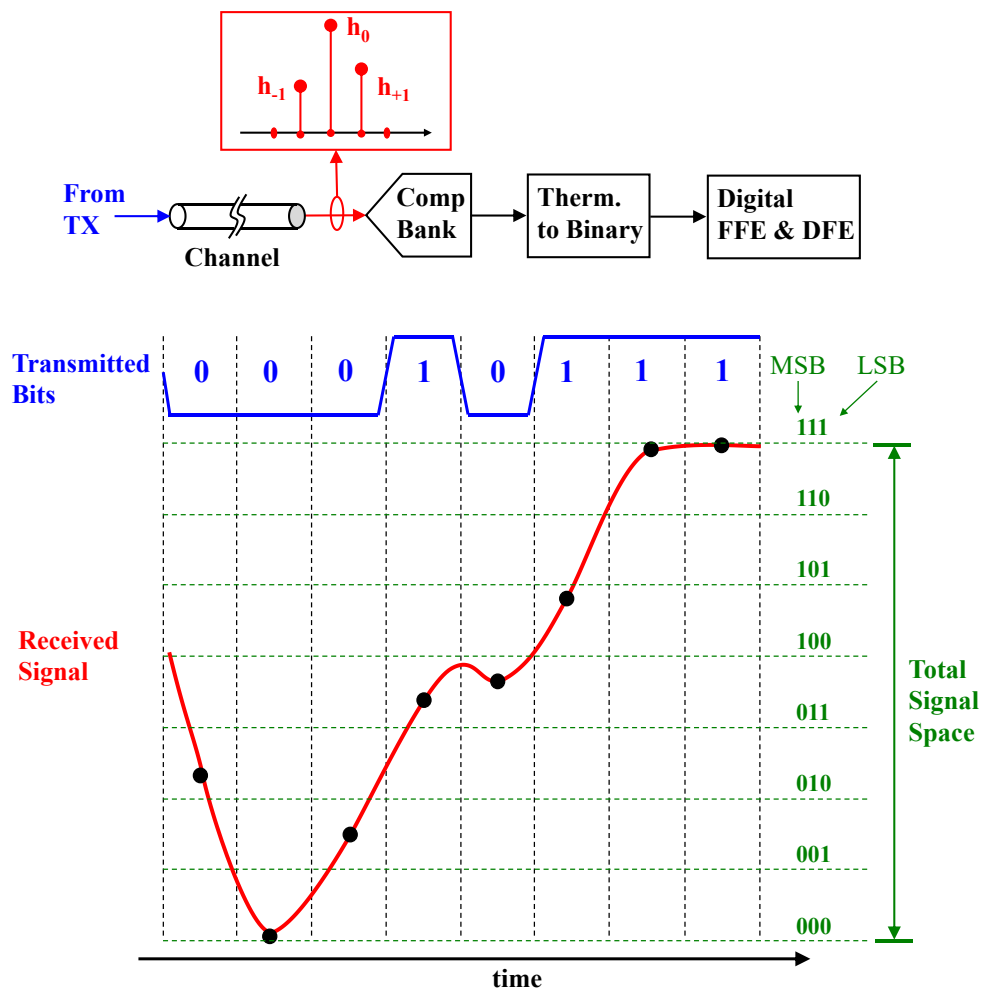


Figure 2.7: ADC-based receiver block diagram and reference placement.

binary converter gives us the ADC output, which is then processed in the digital domain for DFE and FFE operation. The digital implementation of the FFE is realized by subtracting the ADC output bits of nearby cursors from the ADC output of the main cursor. In digital subtractors, the main cursor output of the ADC will go as is and the nearby-cursor taps will be subtracted using the ratio of their tap coefficients. For the DFE implementation in the digital domain, the previous bit decisions are fed back which is then subtracted from the ADC output to get the current bit decision.

For sequence detection, the received signal at any point in time can be seen as a combination of pre-cursor and post-cursor components of the neighboring bit stream and main cursor component of the current bit. Therefore, from the received signal sample, we can reconstruct the corresponding time sequence of previous, current and next bit. In this case (Figure 2.8), as there are only three taps, the bit sequence is  $B_{+1}B_0B_{-1}$ . Here,  $B_{+1}$  is the previous bit,  $B_0$  is the current bit, and  $B_{-1}$  is the next bit. The received signal has to go through a comparator bank. To set the references for these comparators, the simplest approach is to directly calculate the distance from the received sampled value to each sequence constellation. It can be done by comparing the sampled value to a set of references based on different combinations of main ( $h_0$ ), pre-cursor ( $h_{-1}$ ) and post-cursor ( $h_{+1}$ ) taps. In general, we will have a time sequence of length  $N$  if there are  $N$  number of un-equalized taps in the single bit response. In the signal space, these  $N$  taps can combine in  $2^N$  number of ways, creating  $2^N$  signal levels corresponding to  $2^N$  unique sequences.

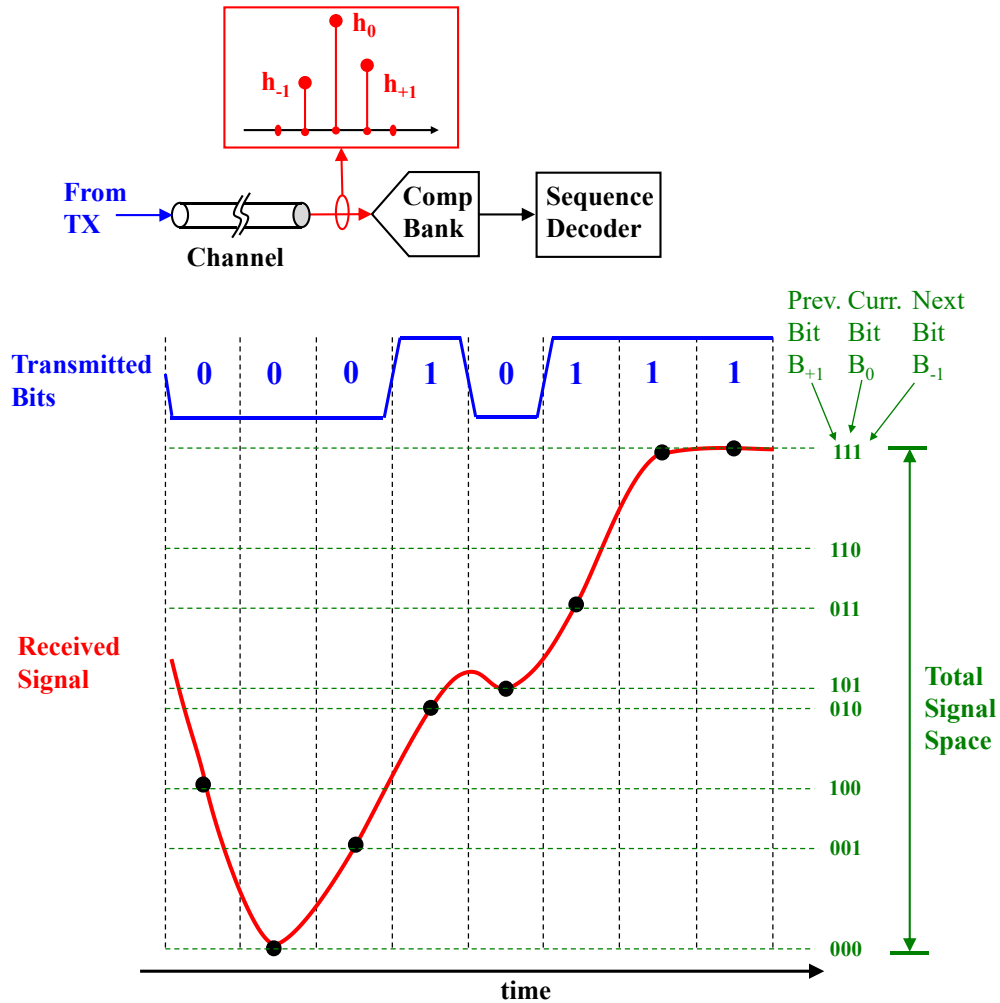


Figure 2.8: Sequence detector reference placement depending on tap coefficient.

### 2.2.2. Digital Sequence Detection

Figure 2.11-Figure 2.11 presents the sequence detection technique. For digital sequence detection, we are placing the comparator references as a combination of tap values representing time sequence as illustrated in Figure 2.8. At time  $t=t_0$ , the signal is at the bottom of signal space (Figure 2.9 (a)). At this time, from transmitted bit sequence, the previous bit is 0, the current bit is 0, and the next bit is also 0. So, at this point, the 3-bit sequence decoder should give

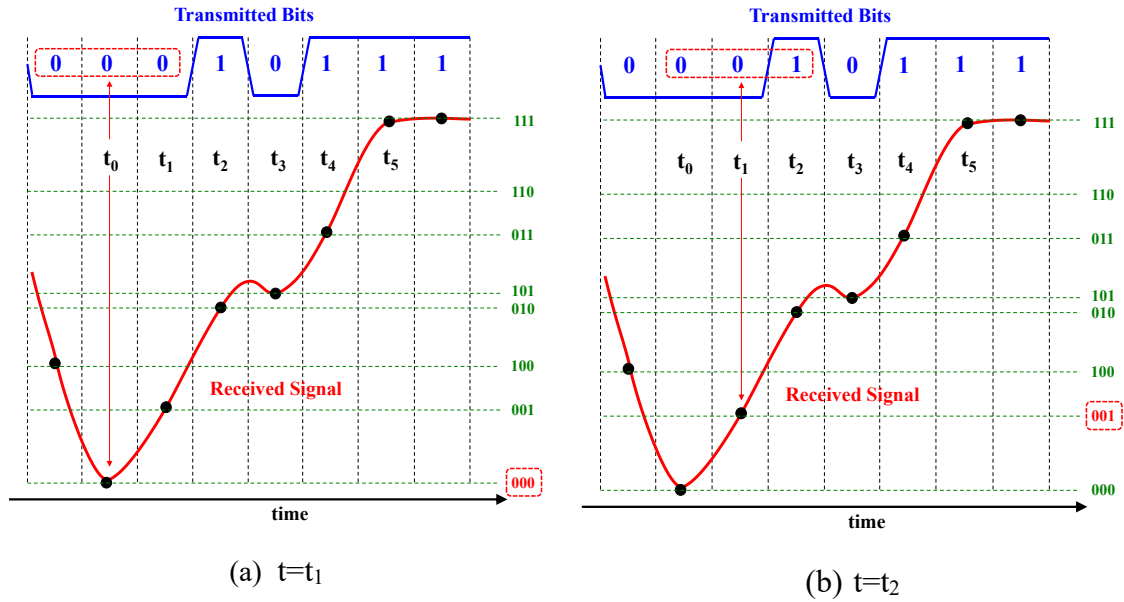


Figure 2.9: Digital sequence detection technique at  $t=t_0$  and  $t=t_1$ .

000. The current bit for  $t=t_0$  becomes the previous bit at  $t=t_1$  (Figure 2.9 (b)). Similarly, the next bit for  $t=t_0$  becomes the current bit at  $t=t_1$ . Now at  $t=t_1$ , the next bit is 1. Due to this bit, the received signal starts to increase. The decoded sequence at  $t=t_1$  should be 001.

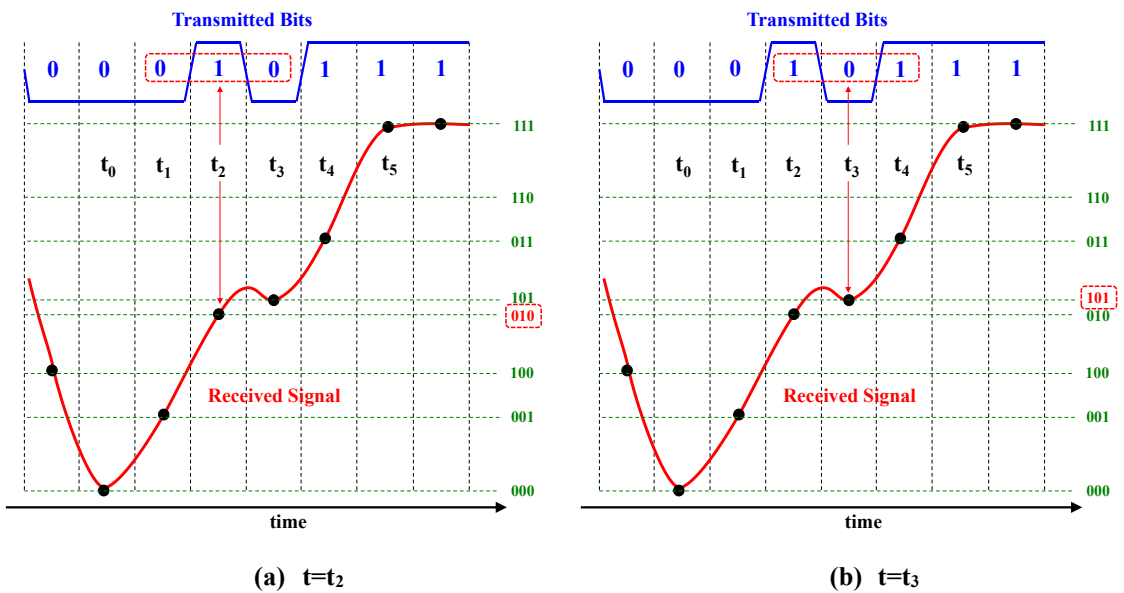


Figure 2.10: Digital sequence detection technique at  $t=t_2$  and  $t=t_3$ .

	$t_0$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$
	0					
	0	0				
Decoded Sequence, $B_{+1}$ $B_0$ $B_{-1}$	0	0	0			
		1	1	1		
			0	0	0	
				1	1	1
					1	1
						1

Figure 2.11: Digital sequence detection technique (summarized).

Similar things happen at  $t=t_2$  (Figure 2.10 (a)) and  $t=t_3$  (Figure 2.10 (b)). At  $t=t_2$  and  $t=t_3$ , the signal is in the middle of signal space. Although at  $t=t_3$  the main bit of the sequence is 0, it cannot go down due to its post-cursor and pre-cursor taps. Figure 2.11 shows decoded time sequence from  $t=t_0$  to  $t=t_5$ . As we are considering a sequence length of 3 bits corresponding to three taps of ISI from the channel, each bit is decoded three times: first as next bit, then as current bit and after that as previous bit. In general, if there are  $N$  taps present in the signal, each bit should be decoded  $N$  times. This  $N$  time decoding of same bit results in improved signal to noise ratio (SNR) and can be used later for verification purposes. The post-cursor bits can be used for DFE operation, whereas the pre-cursor bits can be used for further error correction as will be discussed later. If the received signal is pushed up or down due to noise, the DFE can compensate for those errors by the sequence decoder.

### **2.2.3. Advantages & Challenges in Direct Sequence Decoding**

The basic concepts of ADC and sequence detection have been discussed in prior sections which give us the background to compare between these two techniques and find out the challenges in implementing a sequence decoder.

#### **Advantages of sequence decoding over ADC-based links:**

- Sequence decoder uses ISI in a constructive way. References are set using ISI tap coefficient for each dominant tap in single bit response. ADCs work independently of ISI; however, the DSP and the FFE/DFE do care about ISI. In the FFE/DFE, ISI components are subtracted. Due to this subtraction, we lose signal power as well as add quantization noise.
- If there are  $N$  dominant taps present in single bit response of sequence decoder, each bit will be decoded  $N$ -times. The example discussed so far has 3 dominant taps, thus each bit is detected 3 times which improves SNR. In ADC-based links, additional quantization noise degrades SNR.
- Sequence decoder will detect the time sequence directly. The main symbol comes as a part of the sequence. There is no need for additional DSP.
- In sequence decoder, we are decoding and predicting previous and upcoming bits respectively. These bits can be used later for further error correction. However, for ADC-based links, an additional power hungry DSP is required for the DFE/FFE implementation.



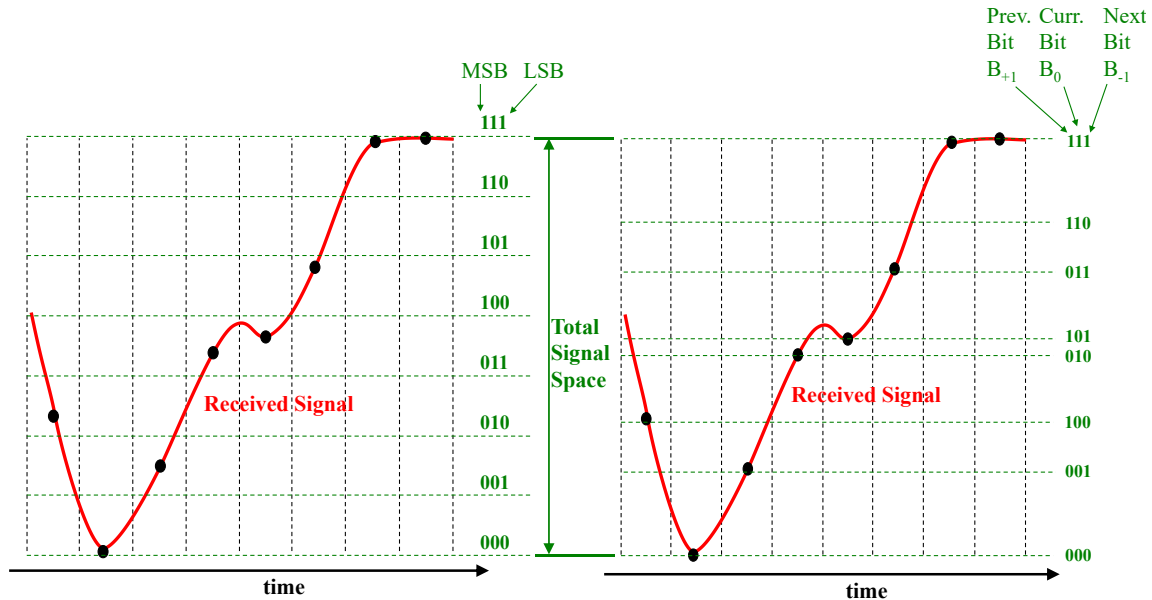


Figure 2.12: ADC vs. Sequence Detection.

### Challenges of sequence decoding over ADC-based links:

- In ADC-based links, the comparator references are binary and monotonically increasing. However, in sequence decoder, references are non-binary and non-monotonic (Figure 2.12).
- In ADC-based links, the required comparator resolution for N-bit ADC is: total signal space/ $2^N$ . The required comparator resolution is not well defined for sequence decoding as the references are non-binary and non-monotonic. It can be less than total signal space/ $2^N$  for N-bit sequence detection.
- In ADC-based links, the thermometric comparator output can be easily converted to binary values. However, converting comparator output to possible sequences, so far, seems difficult.

## 2.2.4. Sequence DFE

The channel we assumed for understanding this concept has three dominant taps. Out of these taps, only one is post-cursor ( $h_{+1}$ ) which corresponds to  $B_{+1}$  in the bit sequence. As shown in sequence detection, this  $B_{+1}$  at any time is detected as  $B_0$  just 1-unit interval (UI) before. We can use this  $B_0$  of previous time (1 UI before) as a feedback for current sequence detection. This post-cursor bit ( $B_{+1}$ ) feedback eases the pressure on the sequence detector to give us the correct sequence directly. From sequence decoder, we can take a set of two possible sequences differentiable using  $B_{+1}$ , and out of those two we can choose one depending on the value of  $B_{+1}$  (Figure 2.13). This post-cursor feedback is same as DFE implementation. As we are implementing it for sequence detector, we can call it sequence DFE. In general, if there are  $N$  post-cursor taps present in the single bit response of the received signal at sequence detector input, we can generate  $2^N$  sequences. Out of these  $2^N$  sequences, only one will be chosen using  $N$  post-cursor bits.

For this example channel, to give choices to DFE, let us have eight comparators in the comparator bank. Each of them is getting references for different possible combinations of  $h_{+1}$ ,  $h_0$ , and  $h_{-1}$ . Although sequences are not increasing monotonically, comparators  $C_{0-7}$ , here are set in monotonic increasing order. The truth table in Figure 2.13 shows the choices for DFE implementation as each comparator output changes. We will discuss how easily we can generate these choices in a later section. In the truth table, if all the comparators give 0 or 1, there is only one choice to DFE. For all 0 case, 000 goes as a choice, which is already paired with 001 when only one comparator is 1. We can get rid of this single option

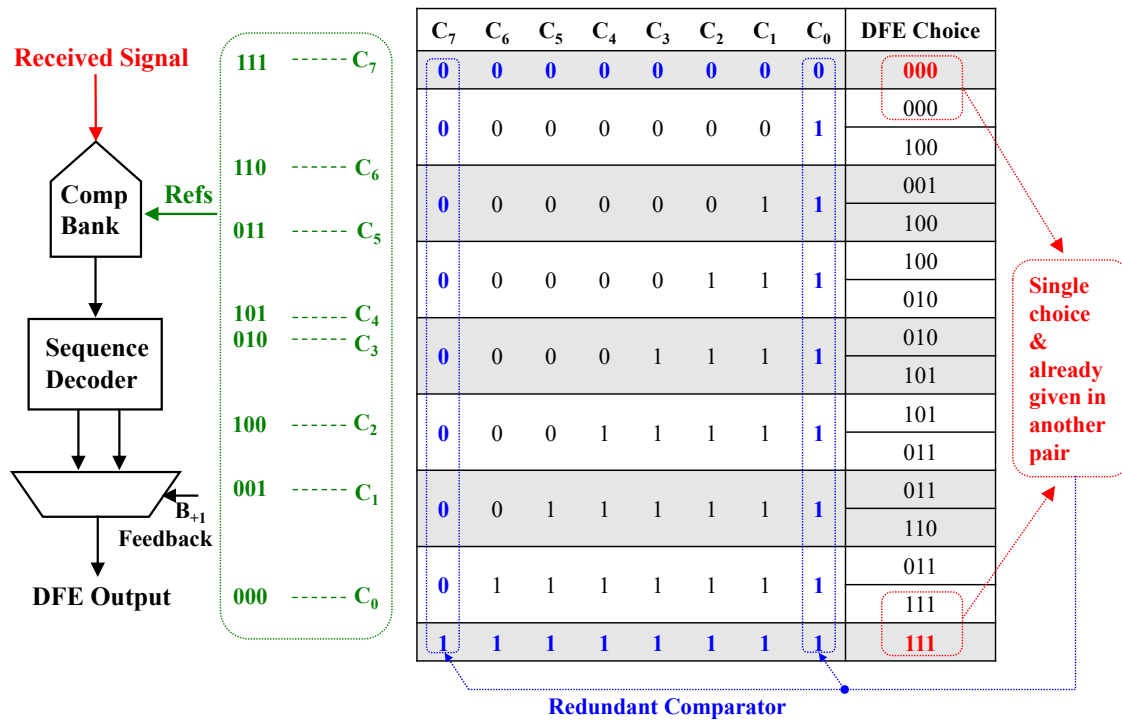


Figure 2.13: Sequence DFE choices for 3 taps.

as this choice is already available. The same thing goes for all 1 case. So, this DFE feedback gives us the option to remove the top and bottom most comparators and still have all the possible sequences in the DFE choice list.

Figure 2.14 shows an example of the DFE implementation. At time  $t=t_2$ , four of the bottom comparators are 1 and the rest of them are 0. According to the truth table in Figure 2.13, DFE choices would be 010 and 101 going to a 2:1 MUX. As the previously decoded bit is 0, out of these two choices the one with  $B_{+1}=0$  will go to the output. So, the output here is 010. Sequence generation for DFE, DFE implementation and timing margin of DFE will be discussed in detail in later sections.

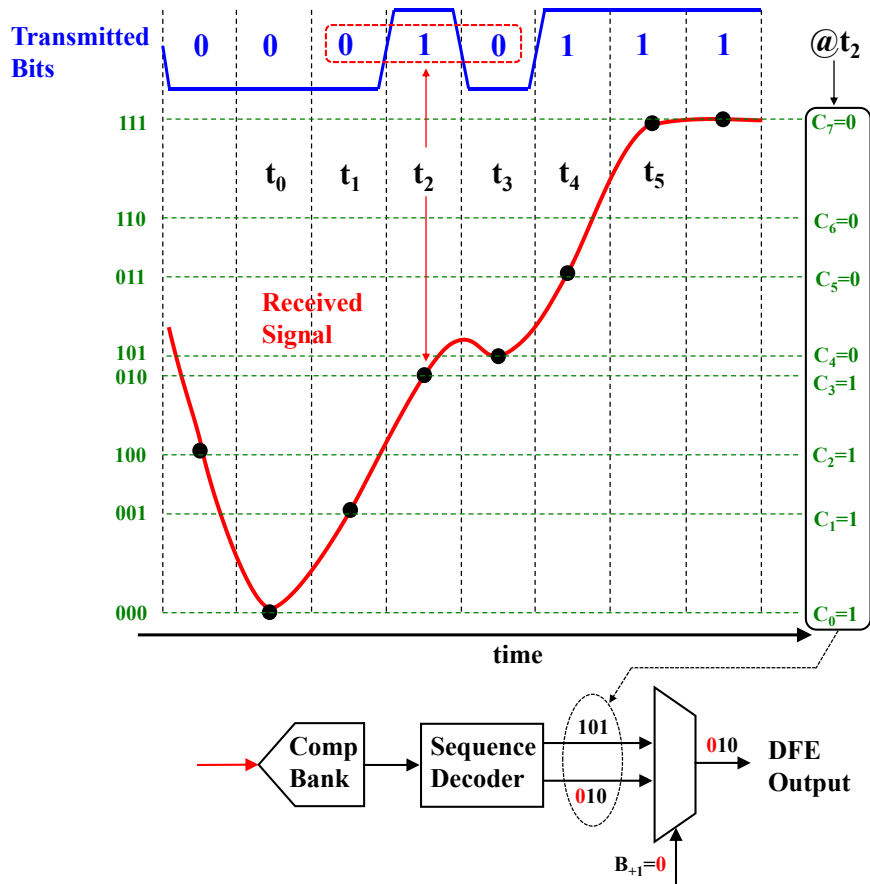


Figure 2.14: Example showing sequence DFE technique.

## 2.3. Link Design

The receiver to be designed considers a voltage mode transmitter where the transmitter differential swing varies from 600mVpp to 1Vpp. Therefore, signal attenuation is needed to scale the received signal to match the dynamic range at the comparator bank input. Since the high-frequency signal is already attenuated by the channel (top of Figure 2.15), only the low-frequency signal is attenuated which translates to passive equalization.

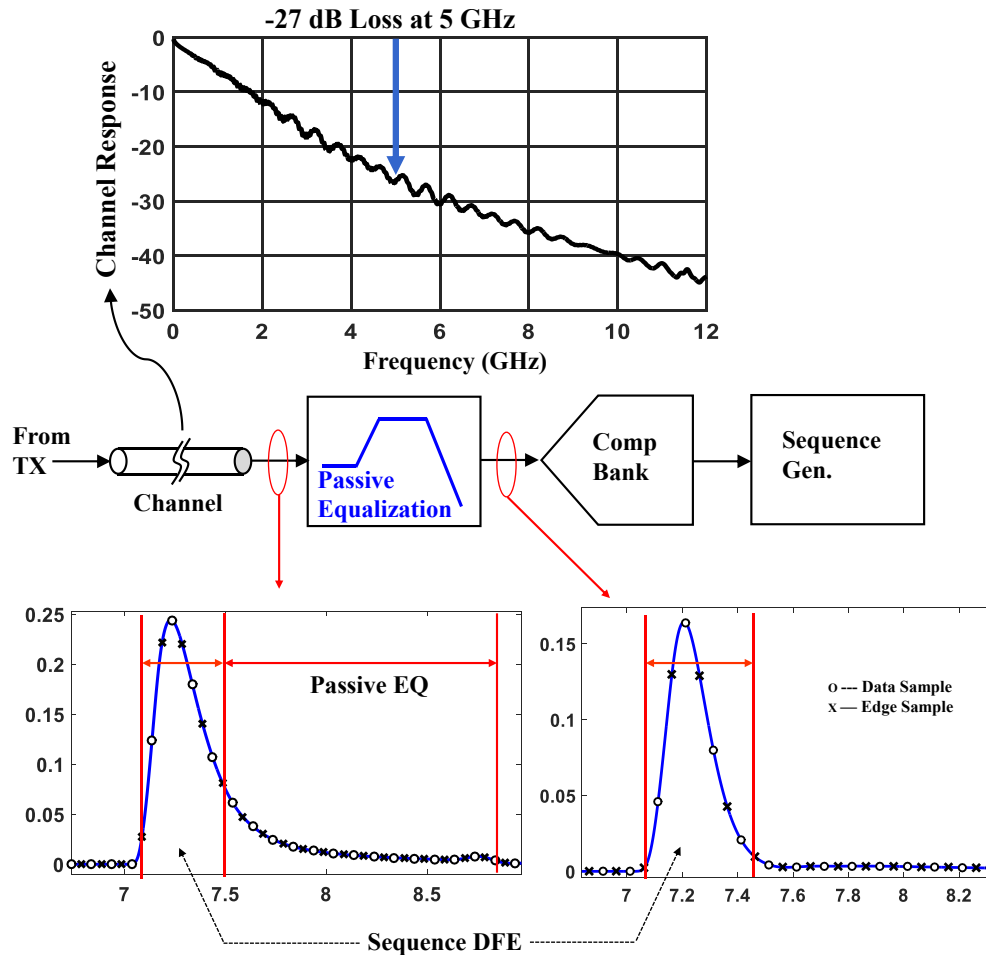


Figure 2.15: Channel response and single bit response before and after the passive equalizer of example transmission link.

For a channel giving 27 dB loss, only 5 to 7 dB boost (or DC attenuation) is sufficient to contain ISI components within four taps – one pre-cursor, main and two post-cursor taps (Figure 2.15). This partially equalized signal is then fed to a 4-bit sequence decoder. The same scheme works for other channels as well if we can have a programmable boost from the passive equalizer.

### 2.3.1. Comparator Reference for Sequence Generation

The channel we assumed in section 2.2 had only 3 taps before going to comparator bank or sequence decoder. The practical channel after passive equalization has 4 dominant taps in its single bit response. For reference generation, instead of having the taps in terms of their time sequence we can have them organized using the descending order of their weights. Main cursor tap ( $h_0$ ) will always have the highest weight among them. In most cases, we will find:  $h_0 > h_{+1} > h_{-1} > h_{+2}$ . So, we will have MSB bit representing  $B_0$ , MSB-1 bit representing  $B_{+1}$ , MSB-2 giving  $B_{-1}$  and LSB representing  $B_{+2}$  (Figure 2.16).

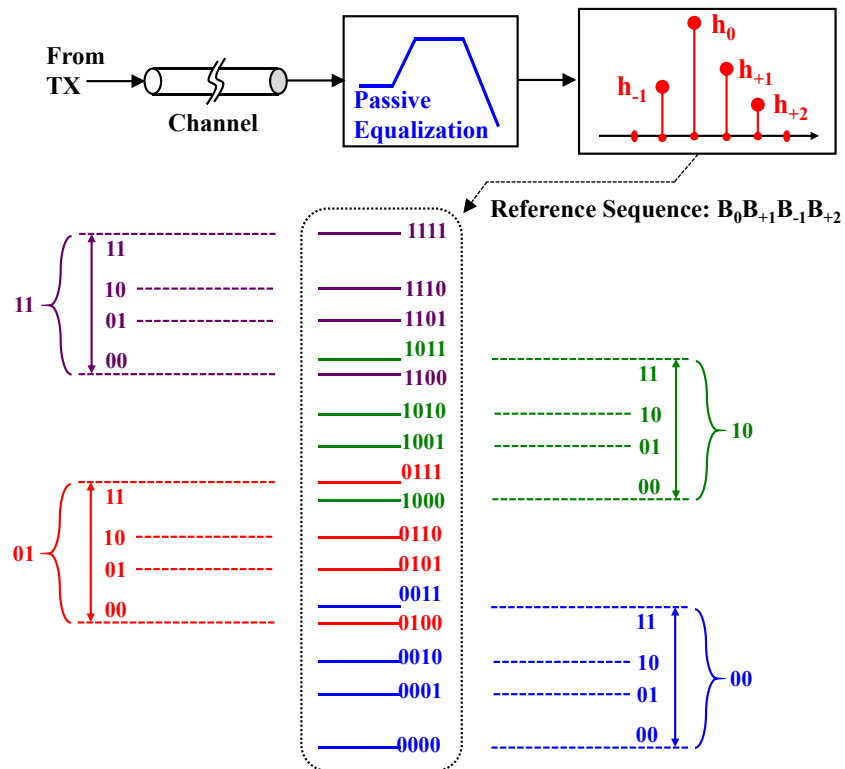


Figure 2.16: References for the link.

The taps don't have binary relation with each other which gives rise to overlaps between reference levels. For example, here 0011 goes above 0100 levels. There are a total of 3 overlaps for this link. If the channel loss increases further, the number of overlap will increase. However, if we divide the references into different banks of  $B_0$  and  $B_{+1}$ , within each bank there will be no overlap as usually  $h_{-1} > h_{+2}$ . The banks will have overlap between them, but not within them.

### 2.3.2. Sequence Generation and DFE Implementation

For an N-bit sequence detector, the input signal needs to be compared to  $2^N$  reference levels that result in a power and area penalty similar to a flash ADC. So, for a 4-bit sequence, 16 reference levels will be there. Note that due to ISI, sample to sample signal variation is limited – therefore, it is not necessary to cover the entire signal space. Rather based on previous sample position, covering only 50% of the signal space is sufficient.

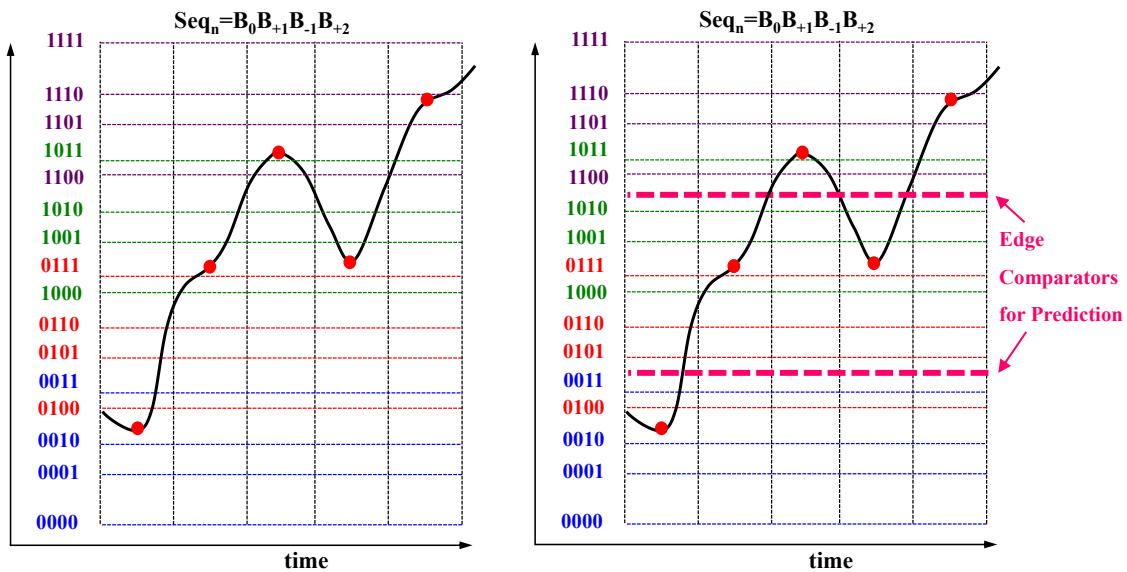


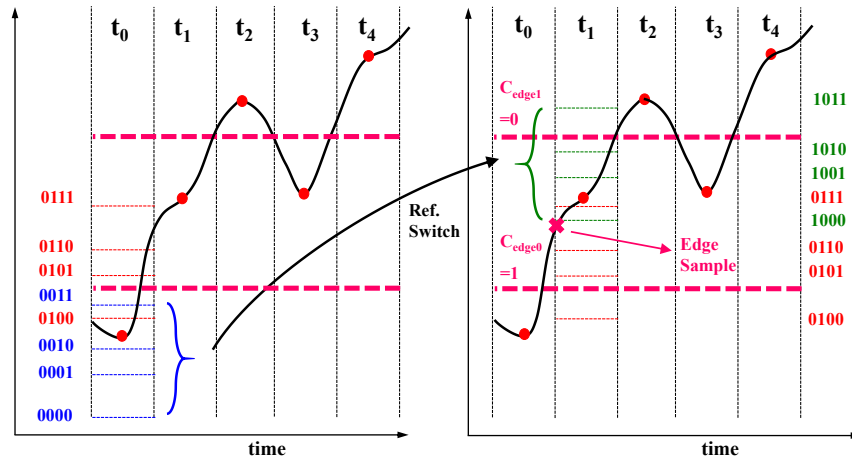
Figure 2.17: Placement of edge comparator for prediction

As we are covering 50% of signal space, rather than having  $2^4=16$  comparators, we can have half of them. In other words, rather than having 4 banks, we can have only 2 of them at a time. This selection of two banks depends on previous sample value. In clock and data recovery circuits, the sample before a data sample is an edge sample. We can place two comparators at the edge sample and use the output of these comparators to recycle the data comparators each time (Figure 2.17).

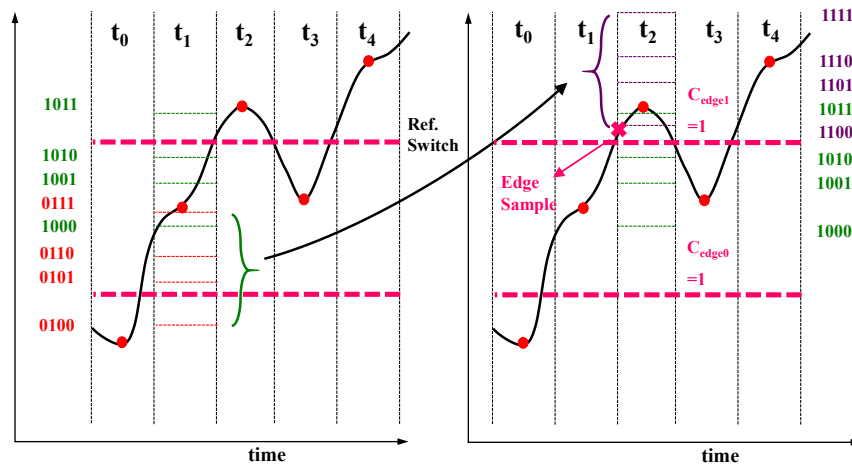
Figure 2.18 illustrates how two edge comparators ( $C_{edge1}$  &  $C_{edge0}$ ) predict data position and place the references for the data comparators. In Figure 2.18(a) at time  $t=t_0$ , the data was at the bottom of signal space. The edge sample in between  $t=t_0$  and  $t=t_1$ , results in  $C_{edge1}=0$  and  $C_{edge0}=1$ . This edge information implies that the next sample may be in the middle of signal space. So, rather than having bottom bank (00) references, we can give 10 bank references at  $t=t_2$ . However, 01 bank references remain in their position.

In Figure 2.18(b) at time  $t=t_1$ , the data was in the middle of signal space. The edge sample in between  $t=t_1$  and  $t=t_2$ , results in  $C_{edge1}=1$  and  $C_{edge0}=1$ . This edge data implies that the next sample may be on the top part of signal space. So, rather than having one of the middle banks (01) references, we can give 11 bank references at  $t=t_3$ . However, 10 bank references remain in their position. So, we can see these two edge comparators directly control references going to the floating comparators.  $C_{edge1}$  controls reference switching between 11 and 01 banks and  $C_{edge0}$  does the same for 10 and 00 banks. Figure 2.18(c) illustrates reference placement for all the cases. The number of comparators now reduces from 16 to 10 out of which 2 are edge comparator and 8 are floating data comparator.

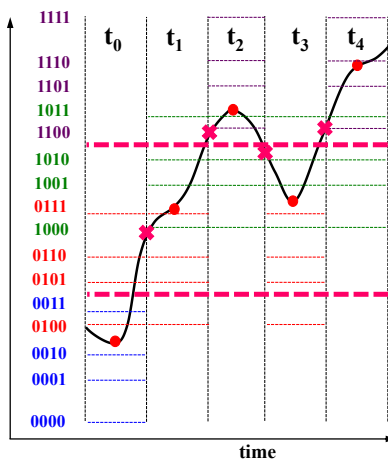




(a)



(b)



(c)

Figure 2.18: Use of the edge comparators for placing the data comparators.

In sequence detection, we have two post-cursor bits. Sequence DFE allows the sequence decoder to give four possible sequences differentiated by  $B_{+1}$  and  $B_{+2}$ . At first,  $B_{+2}$  feedback will come in action as in Figure 2.19. Within the banks, sequences are differentiated by  $B_{+2}$ . So, when we are using a bank reference, we can take two possible sequences from there with different  $B_{+2}$  values and sort out the correct value of  $B_{+2}$  using DFE feedback. Figure 2.20 takes 00 bank as an example. Here even if we remove the comparators  $C_0$  and  $C_3$  of the bank, we will get all possible sequences.

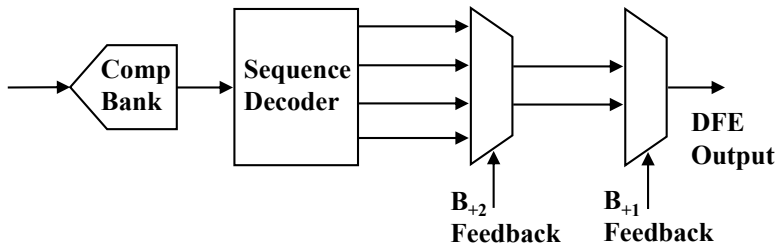


Figure 2.19: Sequence DFE feedback for the link.

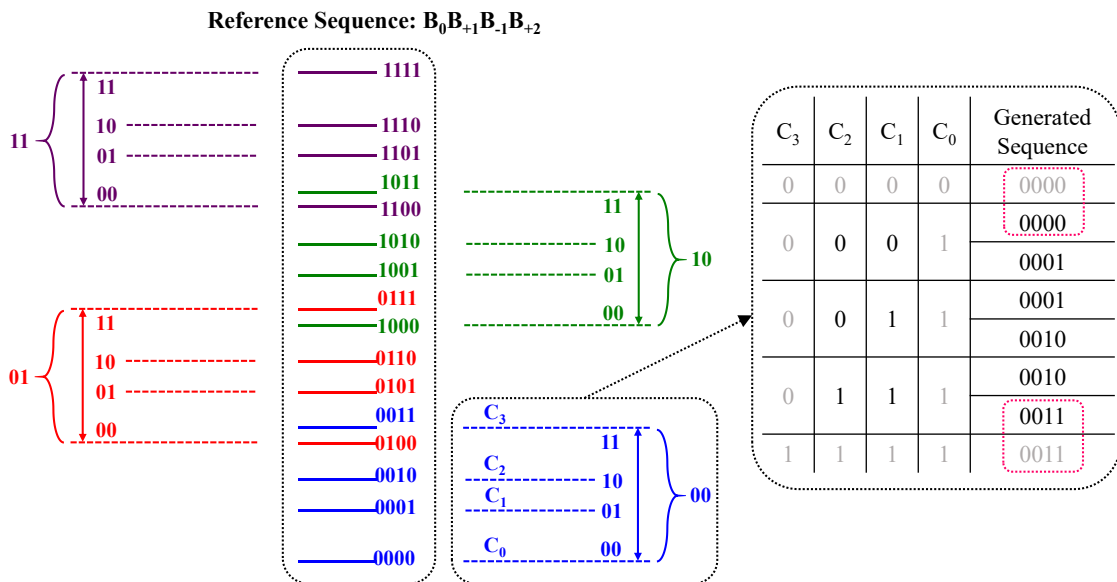
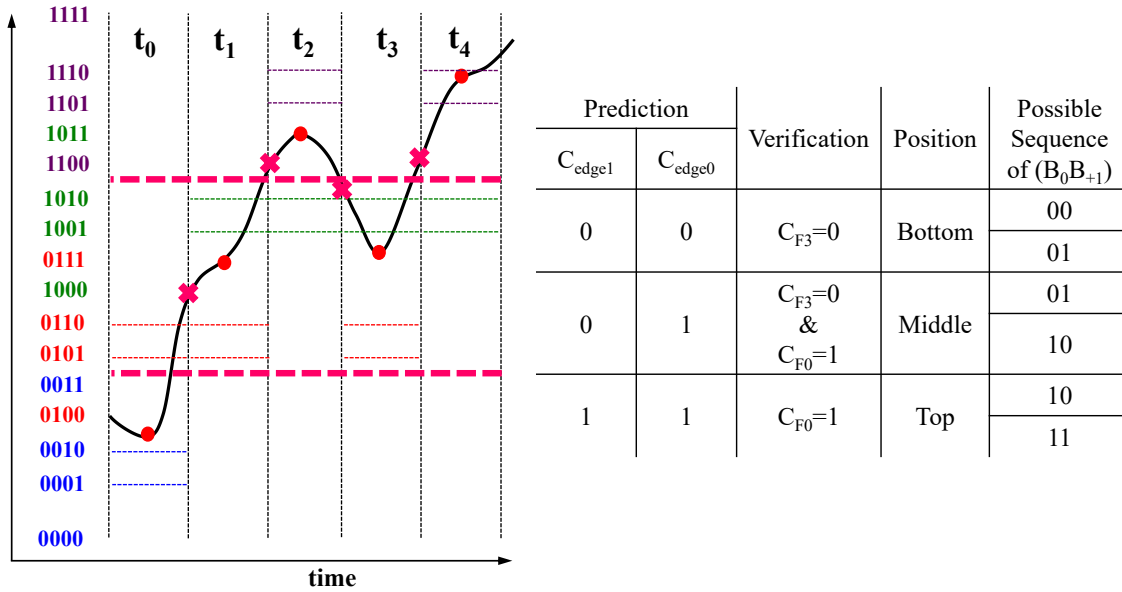


Figure 2.20: Reduction of bank comparators using DFE.

So, the number of comparator per bank is now 2. In total, we need 6 comparators for sequence detection. Two edge comparators are for prediction and four floating data comparators ( $C_{F0-3}$ ) are for in-bank comparison.

In-bank comparison gives us possible combinations of  $B_{-1}B_{+2}$ .  $B_0$  and  $B_{+1}$  differentiate the banks. But out of four banks, two are selected using the edge comparators. We can't use the edge prediction directly to generate possible combinations of  $B_0B_{+1}$ . A verification of that prediction is needed using the data comparators. Although the data comparators are floating, the data comparator having the top floating reference will be called  $C_{F3}$  and the bottom one will be called  $C_{F0}$ . Figure 2.21(a) shows the positions of all six comparators at different times. For edge prediction verification in Figure 2.21(b), when the edge comparators send the floating comparators to bottom two banks, we check the top floating comparator. If  $C_{F3}=0$ , it implies that the signal is actually at the bottom. For the middle case, both the top ( $C_{F3}$ ) and the bottom ( $C_{F0}$ ) floating comparators are checked.  $C_{F3}=0$ , in this case, implies we have not missed the top bank and  $C_{F0}=1$  implies we have not missed the bottom bank. The possible combinations of  $B_0B_{+1}$  is also given in Figure 2.21(b).

Figure 2.21(c) shows two examples of how sequence generation and DFE works. At time  $t=t_2$ , the edge comparators predict and the data comparators verify that the signal is actually on top. The top position implies 11 and 10 are the possible sequences of  $B_0B_{+1}$ . For the 11 bank, the floating comparators,  $C_{F3}$  and  $C_{F2}$ , are 0 at  $t=t_2$  giving possible sequences of 00 and 01 for  $B_{-1}B_{+2}$ . So from the 11 bank the two possible sequences are: 1101 and 1100.



Prediction		Verification	Position	Possible Sequence of (B <sub>0</sub> B <sub>+1</sub> )
C <sub>edge1</sub>	C <sub>edge0</sub>			
0	0	C <sub>F3</sub> =0	Bottom	00 01
0	1	C <sub>F3</sub> =0 & C <sub>F0</sub> =1	Middle	01 10
1	1	C <sub>F0</sub> =1	Top	10 11

(a)						(b)				
Time	Position	C <sub>F3</sub>	C <sub>F2</sub>	C <sub>F1</sub>	C <sub>F0</sub>	Gen. Seq.	B <sub>+2</sub> Feedback	After B <sub>+2</sub> Feedback	B <sub>+1</sub> Feedback	Output
t <sub>2</sub>	Top	0	0			1 1 0 1	0	1 1 0 0	1	1100
						1 1 0 0		1 0 1 0		
						1 0 1 1		1 0 1 0		
						1 0 1 0				
t <sub>3</sub>	Mid	0	0		1	1 0 0 1	1	1 0 0 1	1	0111
						1 0 0 0		0 1 1 1		
						0 1 1 1		0 1 1 1		
						0 1 1 0				

Figure 2.21: (a) Placement of floating comparators. (b) Verification of edge prediction.

(c) Example sequence generation and DFE implementation.

C<sub>F1</sub> and C<sub>F0</sub> do the in-bank comparison for the 10 bank. At t=t<sub>2</sub>, both of them are 1 which implies 11 and 10 are the two possible combinations of B<sub>-1</sub>B<sub>+2</sub> in the 10 bank. So from the 10 bank, the two possible combinations are 1011 and 1010. The generated four combinations go to the DFE multiplexer. One sequence from each bank is selected using B<sub>+2</sub> feedback. At t=t<sub>2</sub>, B<sub>+2</sub> feedback is 0. As B<sub>+2</sub> is LSB here, feedback is compared with the LSB and the combinations with B<sub>+2</sub>=0 is passed to next MUX. Now, there are two

sequences that can be differentiated using  $B_{+1}$ . At  $t=t_2$ ,  $B_{+1}$  feedback is 1. In the combinations coming to MUX,  $B_{+1}$  bit is the MSB-1 bit. After comparison of  $B_{+1}$  feedback, 1100 sequence is passed as DFE output. A similar example is shown for  $t=t_3$ .

Figure 2.22 shows the overall sorting process of the system. As there are 4 dominant taps left in the single bit response after partial equalization by the passive equalizer, we have total 16 sequences- 4 banks. The edge comparator prediction and the floating data comparator verification give us 8 possible sequences from 2 banks. The in-bank floating comparators get rid of half of them giving us a total of four sequences each from 2 banks. These four sequences enter the DFE MUX.  $B_{+2}$  feedback chooses one sequence from each bank.  $B_{+1}$  feedback chooses a final sequence that comes out as the DFE output.

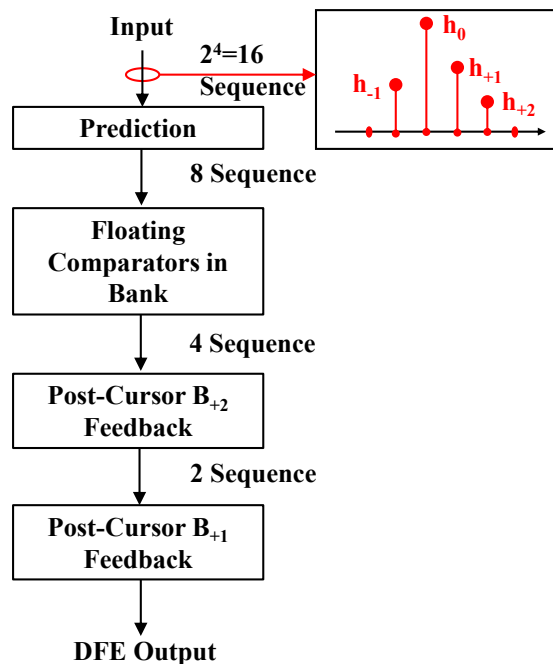


Figure 2.22: Sequence sorting process of the overall system.

## 2.4. Hardware Cost and Performance of the System

For comparison among loop unrolled DFE, ADC-based DFE, and sequence DFE, we can consider a general channel, which has  $N$  dominant taps remaining to be equalized. Out of these taps, let us consider  $L$  taps are precursors and  $M$  taps are post-cursors. There is always one main cursor. Therefore, the number of comparators required for the loop unrolled DFE architecture can be written as

$$C_{\text{Loop UnrolledDFE}} = 2^M \text{ or } 2^{N-L-1}. \quad 2.1$$

For  $N$  tap channel, if we consider  $P$ -bit ADC, the number of comparators required can be written as

$$C_{\text{ADC}} = 2^P - 1. \quad 2.2$$

In the sequence DFE architecture discussed so far, there are edge comparators for prediction and data comparators for sequence generation and verification of prediction. The number of comparators required can be written as

$$C_{\text{Sequence DFE}} = 2^{M-1} + \frac{2^M 2^L}{\text{PredictionFactor}} \quad 2.3$$

where, the prediction factor is the measure of prediction capability of the edge comparators. The first term in Eq. 2.3 gives the number of comparators required for prediction and the second term gives the number of comparators required for in-bank comparison.

Figure 2.23 shows the comparison among the three cases. For the sequence DFE, the prediction factor for the example 4 tap channel is considered to be 2, as we are covering half of the total signal space. This figure shows the required number of comparators for the sequence DFE for three values of the prediction factor (2, 3, and 4). In all cases, the number

of pre-cursors is considered 1. Figure 2.23 shows the number of comparators required for the sequence DFE and the loop unrolled DFE is comparable.

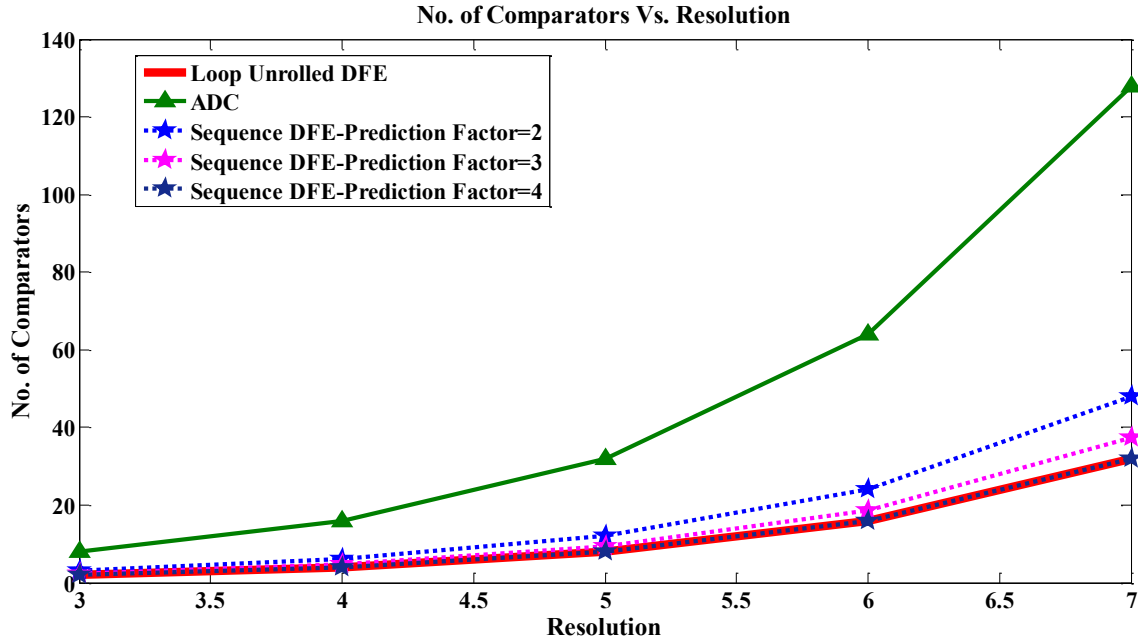


Figure 2.23: Comparison of the required number of comparators among loop unrolled DFE, ADC, and sequence DFE.

The noise margin of the sequence DFE can be defined as half the distance between two banks having similar  $B_{+1}$  values. For an  $N$  tap sequence DFE having  $L$  taps of precursors ( $h_{-i}$ ), main cursor ( $h_0$ ), and  $M$  taps of post-cursors ( $h_{+i}$ ), the noise margin can be written as

$$NM_{\text{Sequence DFE}} = \frac{h_0 - \sum_{i=1}^L h_{-i} - h_{+M}}{2}. \quad 2.4$$

We get the noise margin of a conventional DFE when the un-equalized taps (in this case precursors ( $h_{-i}$ )) interfere destructively with the main cursor ( $h_0$ ). In the case of an ADC-based DFE, the noise margin will further reduce due to the quantization noise ( $Q_N$ ) from the ADC. The quantization noise reduces as the resolution of the ADC increases.

Figure 2.24 shows the comparison of noise margins between ADC-based DFE and sequence DFE. In this figure, practical channels are considered that have 4-7 dominant taps. The tap values are:  $h_0=0.26$  mV,  $h_{+1}=0.16$  mV,  $h_{-1}=0.12$  mV,  $h_{+2}=0.08$  mV,  $h_{+3}=0.04$  mV,  $h_{+4}=0.02$  mV,  $h_{+5}=0.01$  mV. For all cases, only one precursor is considered. For N tap channel, N bit sequence DFE and N bit ADC-based DFE are considered. As the number of bits of the ADC increases the noise margin also increases. For sequence DFE, the noise margin increases as the number of post-cursor taps increases.

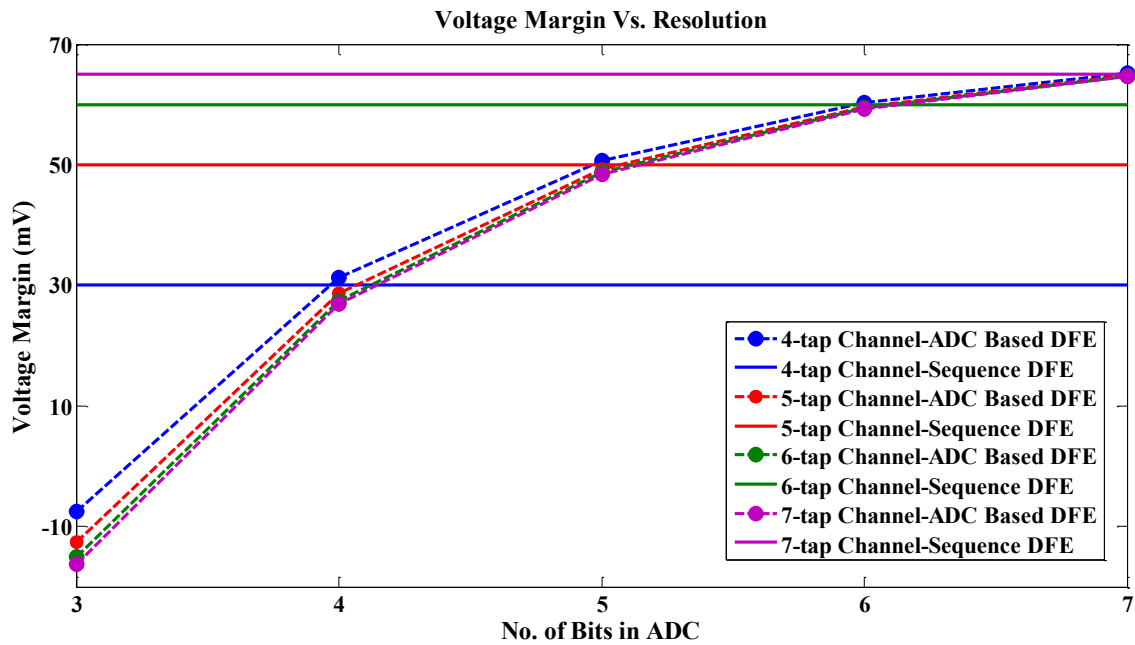


Figure 2.24: Comparison of noise margin between ADC-based DFE and sequence DFE.

## 2.5. Error Tolerance

Section 2.3.2 discusses sequence generation and DFE implementation. There are prediction error tolerance logic implementations that give the system margin to work with errors.



Figure 2.21(b) discusses the verification of prediction done by the edge comparators. Sections 2.5.1-2.5.3 discuss the prediction error correction techniques of the edge comparators. Section 2.5.4 discusses the in-bank comparator error tolerance due to DFE feedback.

### **2.5.1. Bottom Prediction Error Tolerance**

Figure 2.25 shows one of the examples of edge comparator prediction error. With correct prediction, the floating data comparators should be in the middle at time  $t=t_1$ . Due to noise or sampling error, the edge comparators predicted wrong and the floating data comparators stayed at the bottom. At bottom position, possible  $B_0B_{+1}$  combinations are 00 and 01. DFE feedback at time  $t=t_1$  should be 0 as the previous bit is 0. This feedback will make final output for  $B_0$  to be 0 after DFE feedback. So, due to the error in prediction, the main cursor will be detected wrong. However, there are prediction error tolerance logic in the system prevents it from happening. As the edge comparators send the data comparators to the bottom, all the floating data comparators output go to 1. At the bottom, if all the data comparators give 1, this implies an error in prediction of position of the floating data comparators. So, we should overwrite the position of the data comparators to the middle. For the middle position,  $C_{F3}$  and  $C_{F2}$  become  $C_{F1}$  and  $C_{F0}$  respectively. Their outputs are overwritten in  $C_{F1}$  and  $C_{F0}$ . In this error case, if the comparators were placed in the middle, the most probable outputs of the top two comparators are 0. So, 0 is overwritten there. This error check gives the correct choice for DFE operation.

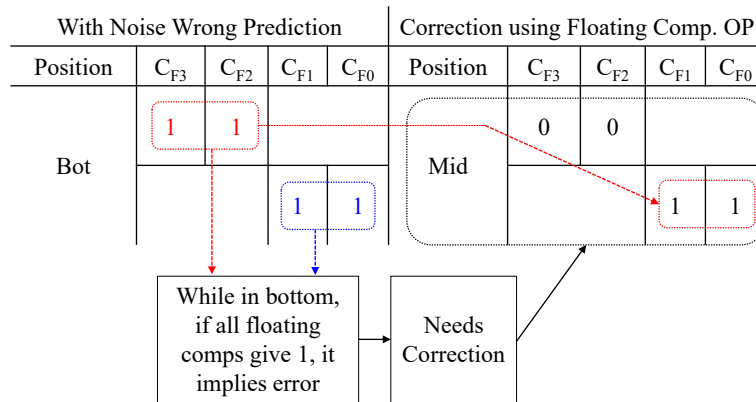
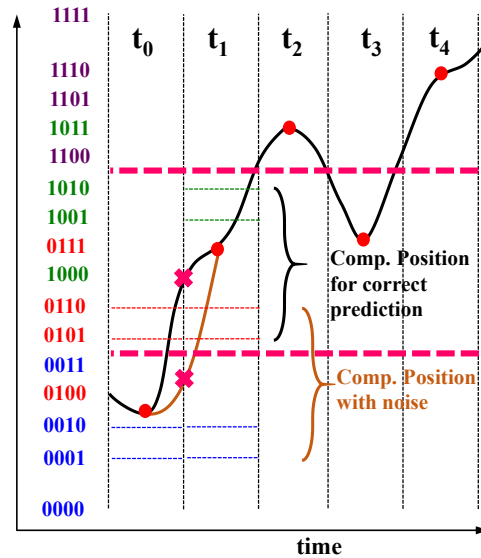


Figure 2.25: Error tolerance of edge comparator prediction to send the floating data comparators to bottom at time  $t=t_1$ .

### 2.5.2. Mid Prediction Error Tolerance

The prediction error tolerance of the mid position has two cases. In case I, the correct position for the floating data comparators are on top, but due to noise, they ended up in the middle (Figure 2.26). Any time an error happens in predicting the position of the floating data comparators, the system will make an error in B<sub>0</sub>. Case I for the mid position prediction error tolerance is similar to the bottom position prediction error tolerance. While in the

middle, if all the floating data comparators give 1, it suggests an error has occurred. As all of them are 1, the signal may be on top instead. So, after verification, mid position is overwritten to the top here. For the top position,  $C_{F3}$  and  $C_{F2}$  become  $C_{F1}$  and  $C_{F0}$  respectively. Their output is overwritten in  $C_{F1}$  and  $C_{F0}$ .  $C_{F3}$  and  $C_{F2}$  will be overwritten with 0 as it is the most probable outcome.

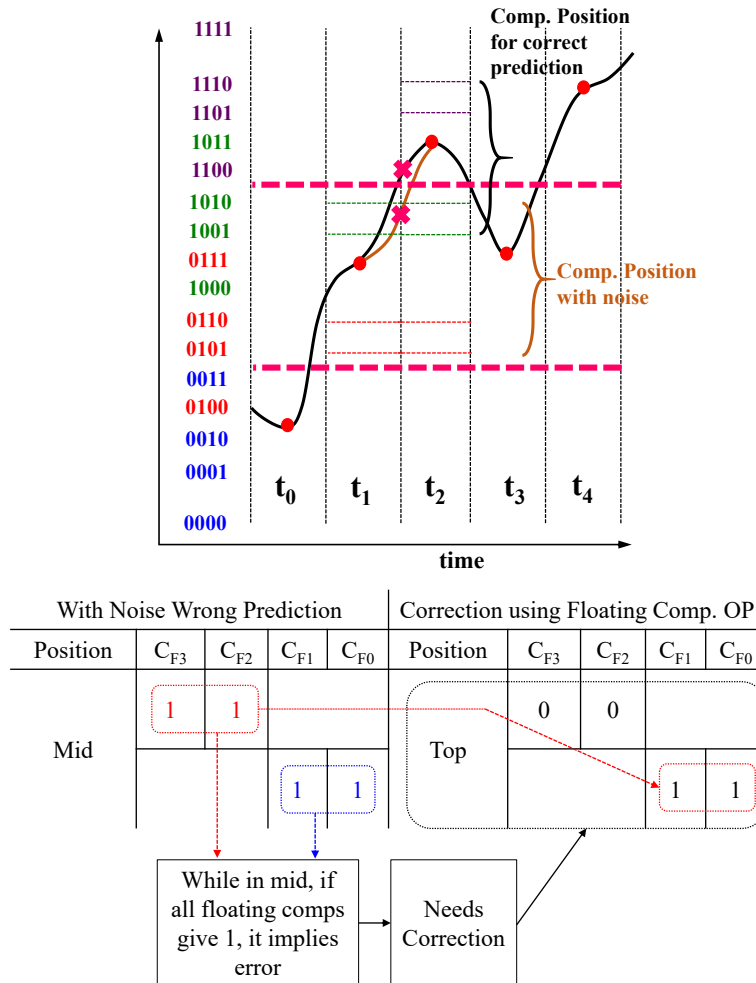


Figure 2.26: Error tolerance of edge comparator prediction to send the floating data comparators to the middle at time  $t=t_2$ .

Case II of the mid prediction error tolerance is illustrated in Figure 2.27. At time  $t=t_2$ , the correct floating comparator position is at the bottom but due to an error, it ended up in the

middle. While in the mid position, if all the floating data comparators give 0, it implies an error has occurred. As all of them are 0, the signal may be at the bottom instead. So, now mid position is overwritten to the bottom. For the bottom position,  $C_{F1}$  and  $C_{F0}$  become  $C_{F3}$  and  $C_{F2}$  respectively. Their output is overwritten in  $C_{F3}$  and  $C_{F2}$ .  $C_{F0}$  and  $C_{F1}$  will be overwritten with 1 as it is the most probable outcome.

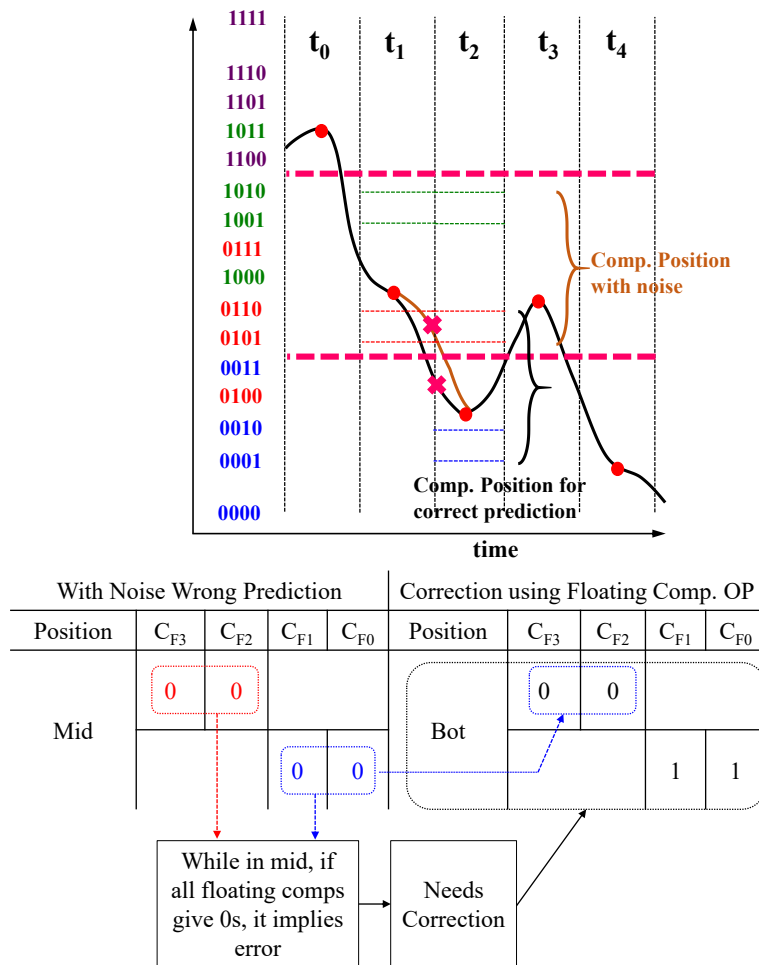


Figure 2.27: Error tolerance of edge comparator prediction to send the floating data comparators to bottom at time  $t=t_3$ .

### 2.5.3. Top Prediction Error Tolerance

Top prediction error tolerance is similar to case II of the mid prediction error tolerance. From the top position, the most probable miss will be the mid position. While on top, if all the data comparators are 0, prediction verification overwrites position to the mid. Outputs of  $C_{F1}$  and  $C_{F0}$  go to  $C_{F3}$  and  $C_{F2}$  respectively.  $C_{F1}$  and  $C_{F0}$  are overwritten with 1.

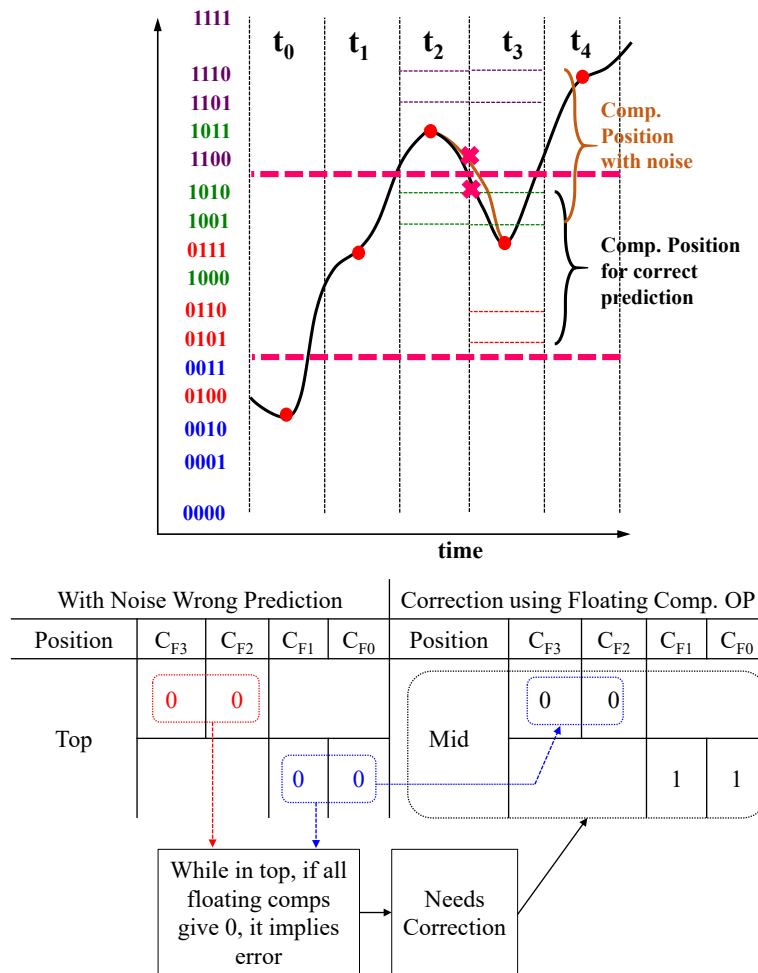
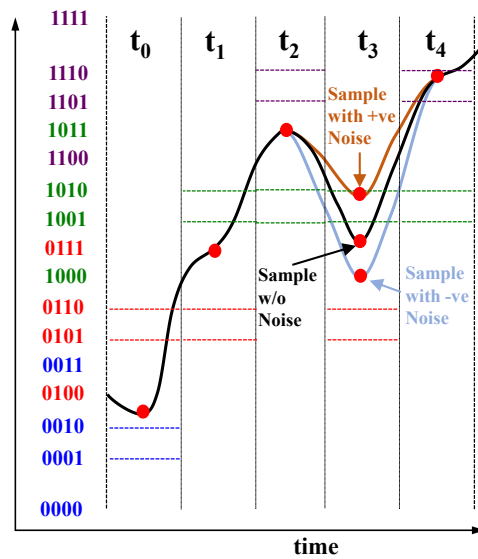


Figure 2.28: Error tolerance of edge comparator prediction to send the floating data comparators to the top at time  $t=t_3$ .

## 2.5.4. In-Bank Comparator Error Tolerance

In prior sections, we discussed how we could fix the issues of comparator and sampling error that result in incorrect positioning of the floating data comparators. We showed that if we make an error in predicting the position of the floating data comparators, we can fix that using the verification logics. Now, if there is an error in data comparators, the DFE is there to fix it.



Noise	Position	$C_{F3}$	$C_{F2}$	$C_{F1}$	$C_{F0}$	Gen. Seq.	$B_{+2}$ Feedback	After $B_{+2}$ Feedback	$B_{+1}$ Feedback	Output
w/o	Mid	0	0			1 0 0 1	1	1 0 0 1	1	0111
						1 0 0 0		1 0 0 1		
			1	1	0 1 1 1	0 1 1 1				
					0 1 1 0	0 1 1 1				
w +ve noise	Mid	0	1			1 0 1 0	1	1 0 0 1	1	0111
						1 0 0 1		1 0 0 1		
			1	1	0 1 1 1	0 1 1 1				
					0 1 1 0	0 1 1 1				
w -ve noise	Mid	0	0			1 0 0 1	1	1 0 0 1	1	0111
						1 0 0 0		1 0 0 1		
			1	1	0 1 1 1	0 1 1 1				
					0 1 1 0	0 1 1 1				

Figure 2.29: In-bank comparator error tolerance of the system using DFE.

Figure 2.29 shows three cases of sampling at time  $t=t_3$ . In the first case, when there is no noise, the correct output is 0111. Now, with some noise, if the signal is pushed up, one of the floating comparators,  $C_{F2}$ , gives 1 which was giving 0 in the no noise case. So, from 10 bank DFE choices are now 1010 and 1001. However, from 01 bank DFE choices do not change, as comparators related to this bank are still giving the correct result. Now, the DFE comes in action and detects the correct sequence. If noise pushes signal down, as long as it is above  $C_{F1}$  reference level, no error will occur.

## 2.6. System Design

The overall block diagram of the quadrature implementation of the system is illustrated in Figure 2.30. The architecture requires three edge comparators and four data comparators in each time-interleaved path. Three edge comparators serve dual purposes: (a) provide the timing error information with higher resolution and (b) place the data comparators in the vicinity of the next sample. In addition, the edge samples with the decoded sequence allow us to filter edges with ISI. Two edge comparators and four data comparators are used to generate sequences with redundancy for sequence DFE. The decoded main bits from previous two time-interleaved paths come in as  $B_{+1}$  and  $B_{+2}$  feedbacks and help in DFE operation.

In subsequent sections, we will talk about the analog front-end and the critical timing issues of the system. Section 2.6.1 - 2.6.4 discuss passive equalizer, sample and hold (SH), comparator, and SR latch respectively. Section 2.6.5 - 2.6.6 discuss the critical timing issues of the system.

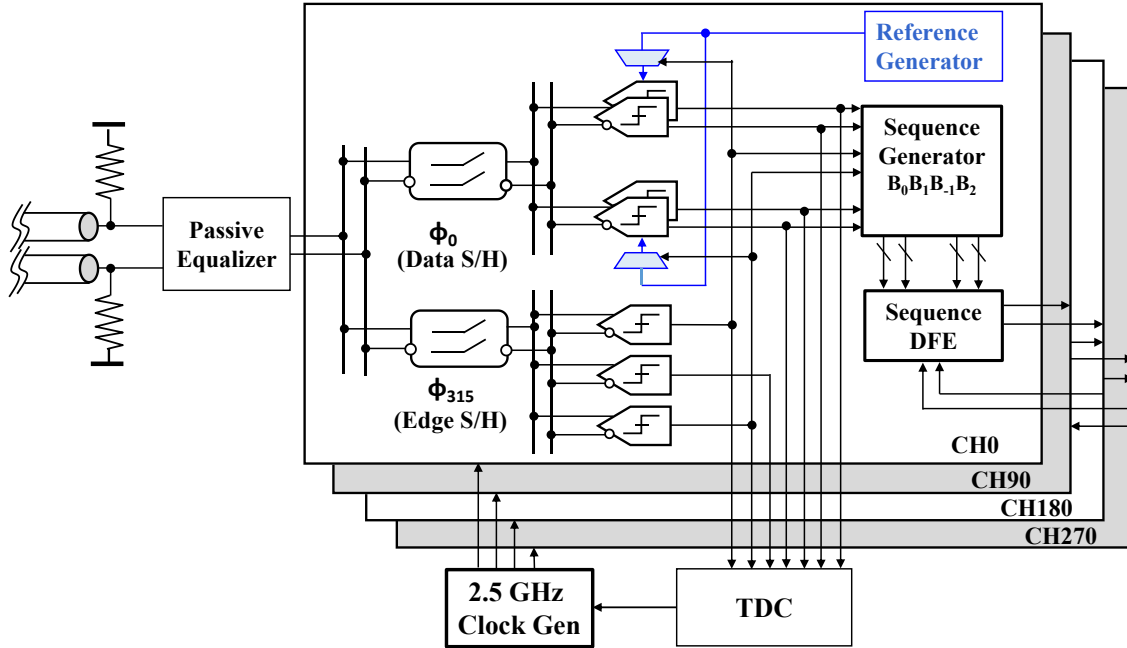


Figure 2.30: Maximum likelihood sequence detector with passive equalization and timing recovery.

### 2.6.1. Passive Equalizer

The passive equalizer used in this system is a C-R high pass filter that attenuates low-frequency portion of the incoming signal. Figure 2.31 shows the schematic and AC response of the equalizer. The transfer function of the equalizer can be written as:

$$\text{Transfer Function, } \frac{V_{in}}{V_{out}} = \frac{R_4 + sCR_{EQ}R_4}{(R_{EQ} + R_4) + sCR_{EQ}R_4}. \quad 2.5$$

$$\text{DC Gain} = \frac{R_4}{R_{EQ} + R_4}.$$

High Frequency Gain = 1.

$$\text{Zero of the Equalizer, } \omega_z = \frac{1}{CR_{EQ}}.$$



$$\text{Pole of the Equalizer, } \omega_p = \frac{1}{C(R_{EQ} \parallel R_4)}$$

The choice of values of resistors and capacitor is critical. Small values of R imply degradation in input matching and the equalizer needs a bigger capacitor to have zero at proper frequency. Larger values of R give higher input time constant. The thermal noise generated from these resistors is small enough that it does not effect the sequence DFE operation. The tunable  $R_{EQ}$  of the equalizer makes it possible to have a DC attenuation from 3 to 8 dB. It also moves the pole and the zero of the system. The tunable boost also makes it possible to work with channels of different loss. As long as the passive equalizer gives us a single bit response with four dominant taps remaining to be equalized, the system can equalize the rest. If channel taps remain unequalised, their impact will come as deterministic noise. Figure 2.32(a) shows the AC analysis of the equalizer for the example link with 27 dB loss. Here, the passive equalizer equalizes 6 dB of channel loss and the maximum likelihood sequence detector will equalize the remaining 21 dB of loss. The transient eyes of input and output of the passive equalizer are shown in Figure 2.32(b) and Figure 2.32(c). As the equalizer output only has 4 dominant taps, 16 levels with overlap appear at the equalizer output. Figure 2.33 shows the layout of the passive equalizer.

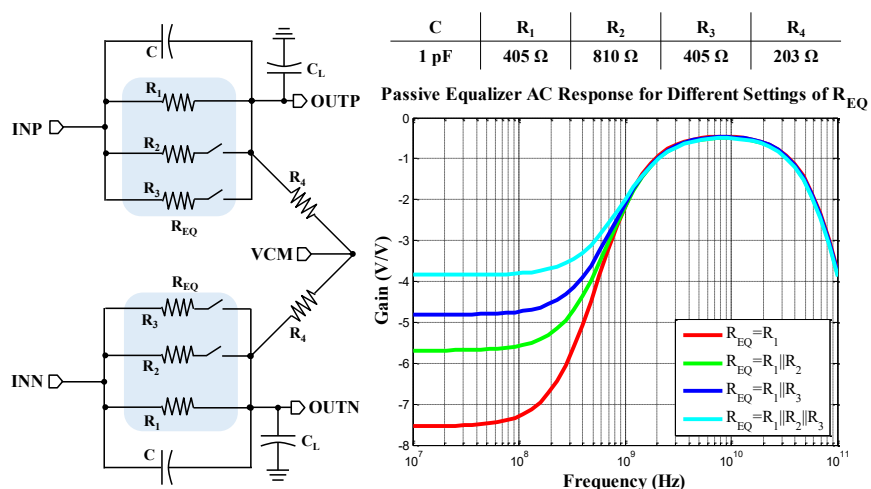


Figure 2.31: Schematic of passive equalizer with its AC response for different settings.

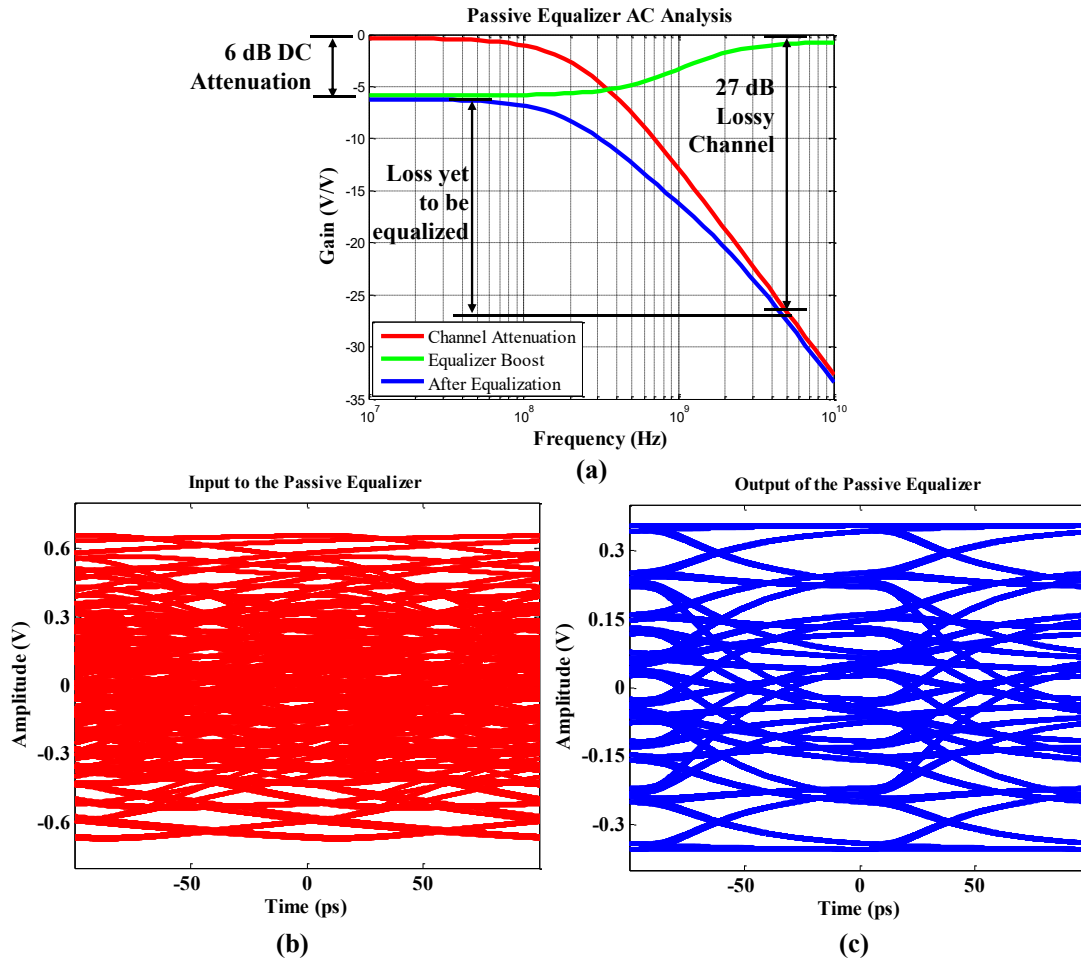


Figure 2.32: Passive equalizer response for example link. (a) AC analysis, (b) input to the equalizer and (c) output of the equalizer.

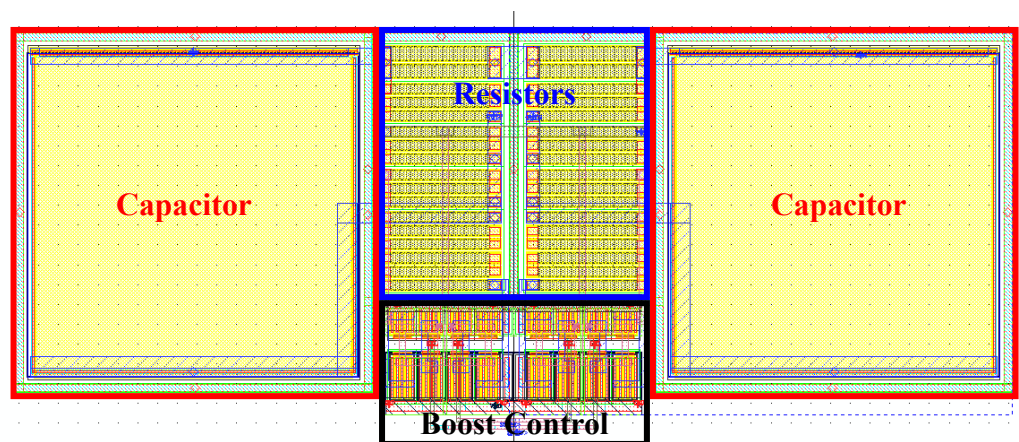


Figure 2.33: Layout of the implemented passive equalizer.

## 2.6.2. Sample & Hold (SH)

Figure 2.34 shows the concept of a conventional sample and hold circuit. The input data,  $V_{in}$ , comes in and hits a switch. A Clock signal controls the switch. Whenever the Clock is high, the switch gives a direct path from the input to the output. The RC time constant of resistance from switch and  $C_{hold}$  has to be small so that the SH has enough bandwidth to follow the high-speed input and change accordingly. When Clock goes low,  $C_{hold}$  will hold the last value it gets from the input.

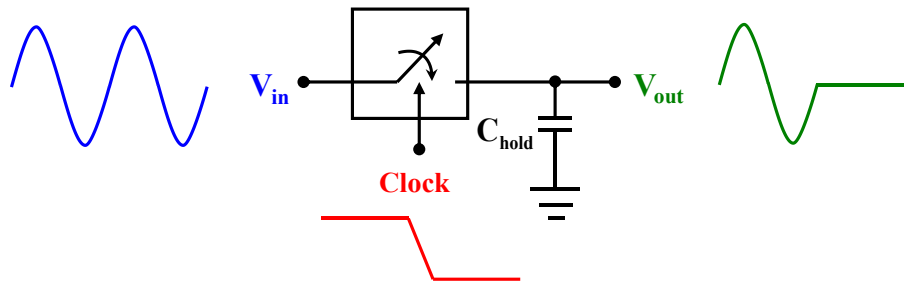


Figure 2.34: Concept of sample and hold.

High performance and high-speed data receivers require good linear performance from their SH circuits. High-performance SH circuits are usually implemented using Switched Capacitor (SC) circuits. The input sampling switch limits the linearity of these SH circuits. Non-linearity associated with the sampling switch is mainly attributed to non-linear on-resistance ( $R_{ON}$ ) and associated parasitic capacitance of the transistors. These transistor switches produce harmonic distortion when sampling high-frequency signals. As a result, SNDR (Signal to Noise and Distortion Ratio), SFDR (Spurious Free Dynamic Range), and THD (Total Harmonic Distortion) of the incoming signal deteriorate.

The on-resistance ( $R_{ON}$ ) of a transistor switch is given by [24]

$$R_{ON}(t) = \frac{1}{\mu C_{ox} \frac{W}{L} (V_{gs}(t) - V_{TH}(t))} \quad 2.6$$

where,  $\mu$  = Mobility of charge carrier of transistor  
 $C_{ox}$  = Oxide Capacitance per unit area  
 $V_{GS}(t)$  = Gate to source voltage  
 $V_{TH}(t)$  = Threshold voltage of the transistor.

$V_{GS}(t)$  and  $V_{TH}(t)$  depend on the incoming data signal,  $V_{in}(t)$ .

$$V_{gs}(t) = V_g(t) - V_{in}(t) \quad 2.7$$

$$V_{TH}(t) = V_{TH0} + \gamma(\sqrt{2|\phi_F| + V_{SB}(t)} - \sqrt{2|\phi_F|}) \quad 2.8$$

where,  $V_{TH0}$  = Threshold voltage when  $V_{SB} = 0$   
 $V_{SB}(t)$  = Source to body voltage  
 $\gamma, \phi_F$  = Device parameters.

$$V_{SB}(t) = V_{in}(t) - V_B \quad 2.9$$

where,  $V_B$  = Body Voltage.

The  $R_{ON}$  along with the  $C_{hold}$  define the tracking bandwidth of the SH circuit as given by Eq. 2.10. The dependence of  $R_{ON}$  on the time varying input signal means that  $f_{-3dB}$  will be different for different values of the input signal. Therefore, the SH will not track all values of the input signal equally.

$$f_{-3dB} = \frac{1}{2\pi} \frac{1}{R_{ON} C_{hold}} = \frac{1}{2\pi} \frac{\mu_n C_{ox} \frac{W}{L} (V_{gs}(t) - V_{TH}(t))}{C_{hold}} \quad 2.10$$

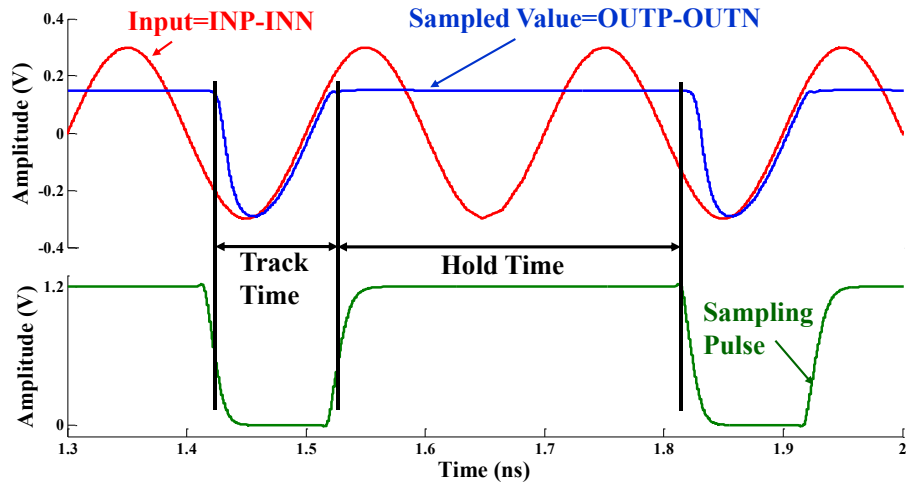
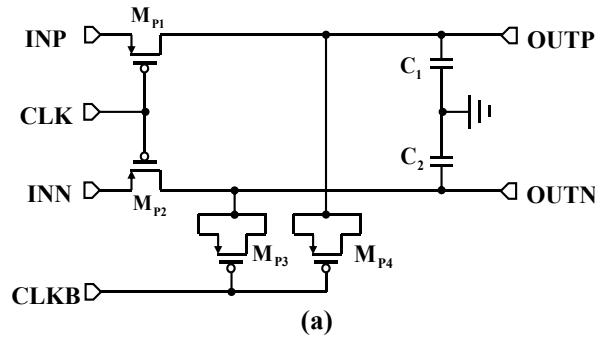


Figure 2.35: Implemented sample and hold circuitry (a) and its simulated differential operation (b).

For a fixed  $C_{hold}$ , the  $R_{ON}$  has to be small enough to achieve high bandwidth in the SH. So, in our design, as illustrated in Figure 2.35 (a),  $M_{P1}$  and  $M_{P2}$  have high W/L ratio, which allows low  $R_{ON}$  and high bandwidth. Differential operation gets rid of clock feedthrough that comes when sampling pulses come in or when the comparator in the next stage of the SH is clocked. The expected signal swing at the input of SH is about 600 mV with a common mode of around 800 mV. Using transmission gates here as switches will not help. Bootstrapping makes  $R_{ON}$  independent of  $V_{in}(t)$  but adds a lot of complexity.

The sampling time is also a function of  $V_{in}(t)$ . When the voltage of the capacitor node is lower than the input voltage, the input node acts as the source of the sampling PMOS. The relation among  $V_{in}(t)$ ,  $V_{gs}(t)$ , and  $V_{TH}(t)$  is shown in Eq. 2.7 - 2.9. In the implemented SH, for low values of  $V_{in}(t)$ , the PMOS switch turns off too fast while for high values of  $V_{in}(t)$  it turns off too late causing distortion.

Charge injection is another source of non-linearity in the SH. In the case of PMOS switches, when CLK signal goes high, the charge in the channel is distributed between the drain and the source of the MOSFET. The amount of charge escaping from the channel is a complex function of impedance defined by the amount of charge to the ground and the transition time of the controlling clock. This charge injection gives us gain error and DC offset in the SH. The transistors,  $M_{P3}$  and  $M_{P4}$ , are there to absorb the injected charge from the channel, which is a crude way to get rid of this problem without adding complexity to the SH. The transistors  $M_{P3}$  and  $M_{P4}$  are sized half of  $M_{P1}$  and  $M_{P2}$ , assuming that half of the injected charge will flow in each way. Figure 2.35(b) shows the differential operation of the implemented SH. The SH in this design is triggered with 25% duty cycle pulses. As a quadrate system, the SH tracks the incoming data for 1 UI and holds the data for 3 UI. Figure 2.36 shows the layout of the implemented sample and hold circuit. The figure specifies the positions of sampling PMOS, compensating PMOS, and capacitor in the sample and hold.

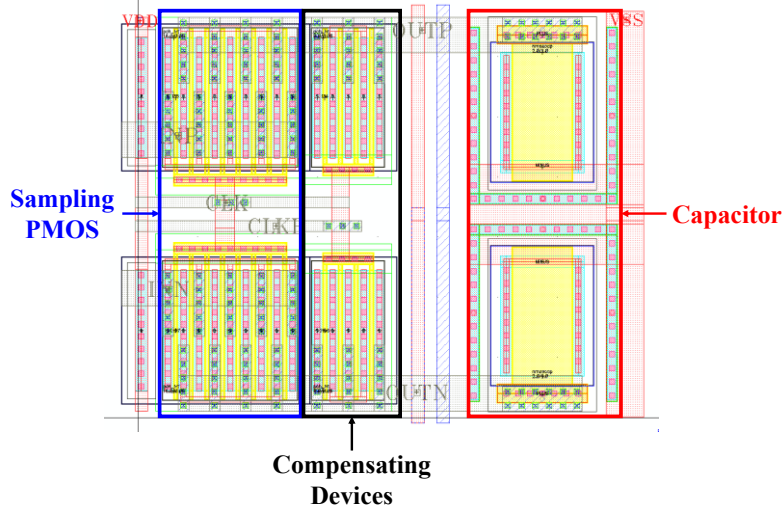


Figure 2.36: Layout of sample and hold circuit.

### 2.6.3. Comparator

Figure 2.37 shows the implemented 4-input comparator. First, for simplicity, we can remove  $M_{N5}$ ,  $M_{N6}$ , and  $M_{N7}$  and consider this as a 2-input comparator. The strong-arm latch based comparator gets the sampled value from the SH circuit. The theoretical depiction in Figure 2.38 shows four stages of operation of the comparator. At first, the sampling phase starts when CLK signal goes from low to high at  $t=t_0$ . The transistors  $M_{N3}$  and  $M_{N4}$  discharge  $V_N$  and  $V_P$  node respectively.  $M_{N1}$  and  $M_{N2}$  discharge  $OUTN$  and  $OUTP$  respectively. The sampling phase ends when PMOS transistors  $M_{P5}$  and  $M_{P6}$  turn on. The regeneration phase starts at  $t=t_1$ . At this stage, as PMOS transistors are ON, we get cross-coupled inverters formed by  $M_{N1}$  and  $M_{P5}$  pair and  $M_{N2}$  and  $M_{P6}$  pair. The positive feedback from the cross-coupled inverters amplifies the signal. When the comparator outputs touch the rail at  $t=t_2$ , it enters the decision phase. The decision from this stage is latched using an SR latch afterwards. When the CLK signal goes low at  $t=t_3$ , the comparator outputs resets, i.e. both outputs go to VDD. The speed of the comparator depends on its sampling and

regeneration stage. Figure 2.39 and Figure 2.40 show the simulation results and the layout of the implemented comparator.

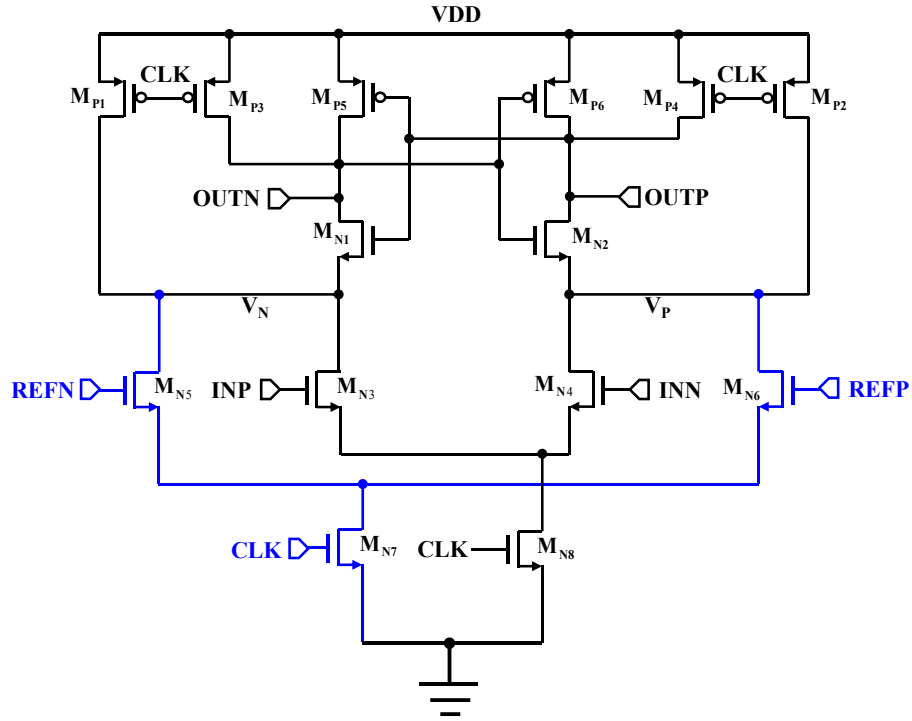


Figure 2.37: Schematic of the implemented comparator.

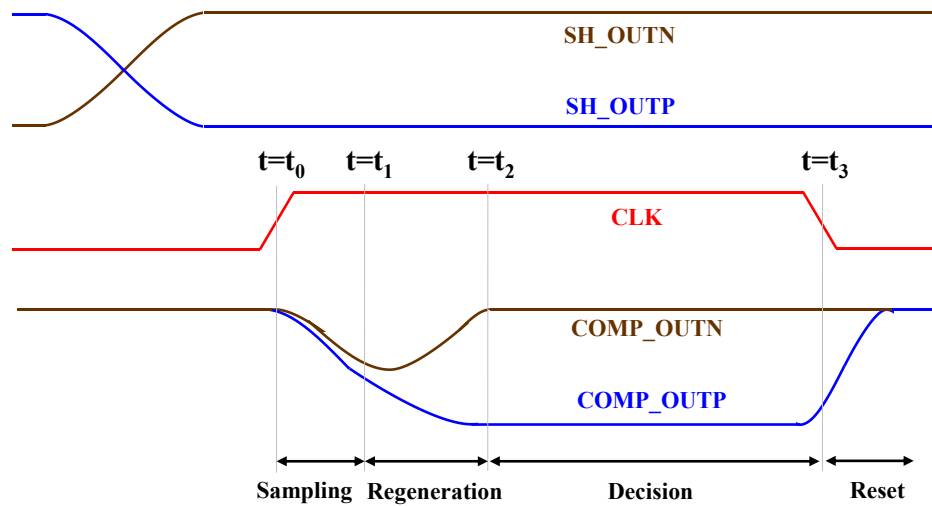


Figure 2.38: Operational stages of the comparator.



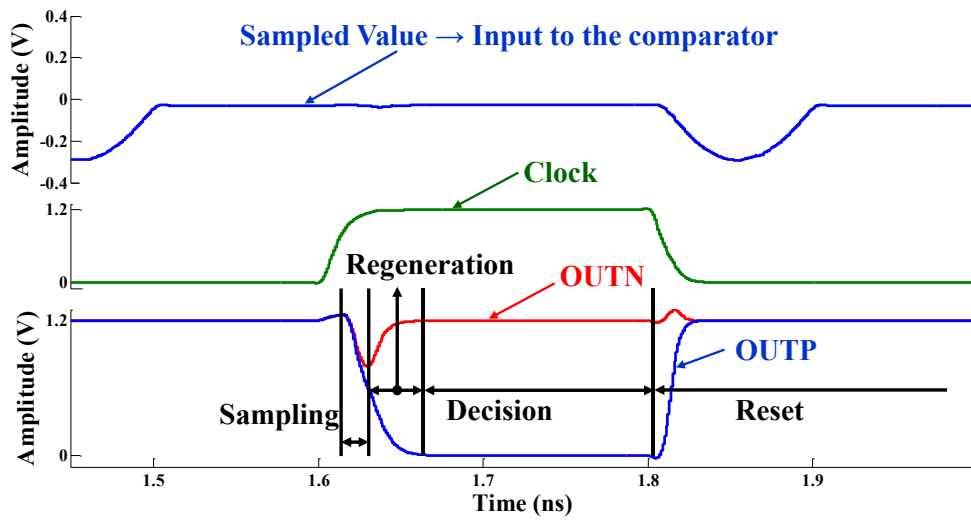


Figure 2.39: Simulation results showing sampling, regeneration, decision, and reset stages of the comparator.

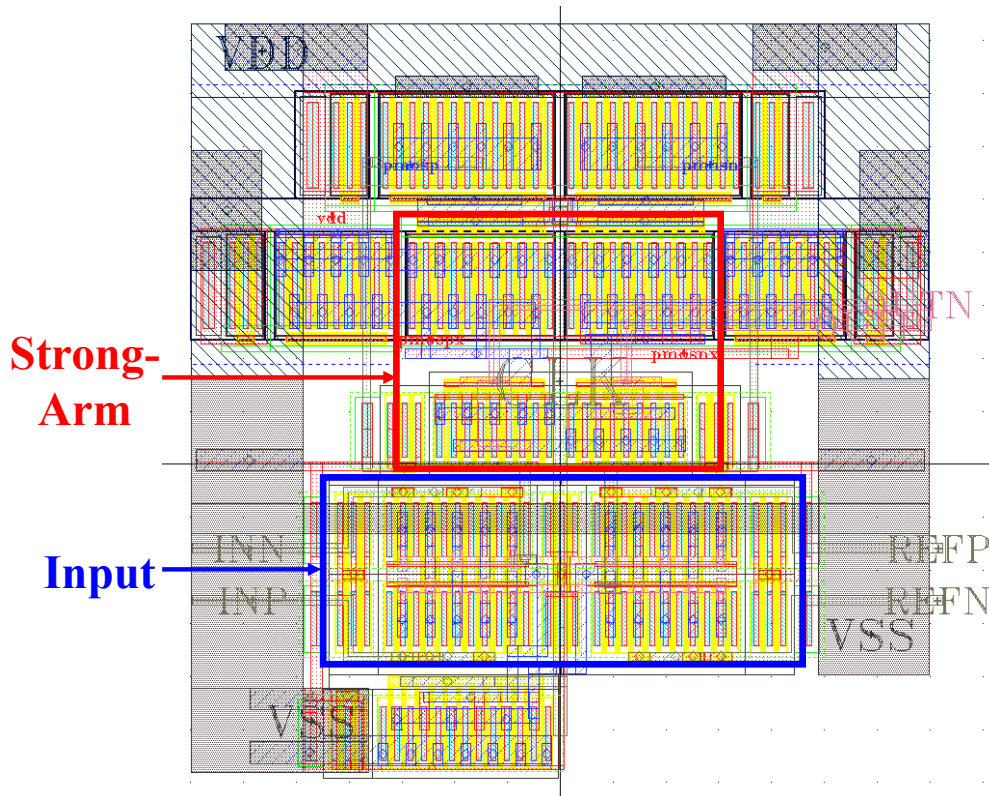


Figure 2.40: Layout of the implemented comparator.

## 2.6.4. SR Latch

The implemented comparator resets to VDD, as CLK goes low. Therefore, the implemented set-reset (SR) latch has to hold the previous value when both of its inputs are high. Figure 2.41 shows the NAND implementation of SR latch. In this implementation, both  $S=0$  and  $R=0$  are invalid inputs. In our case, the comparator outputs (OUTP and OUTN) will never go to VSS at the same time. When both signals are high, the SR latch retains its previous value. When  $R=0$ , it forces  $Q(t+1)=1$ . As,  $S=1$  and  $Q(t+1)=1$ ,  $QB(t+1)$  goes to 0. When  $S=0$ , it forces  $QB(t+1)=1$ . As,  $R=1$  and  $QB(t+1)=1$ ,  $Q(t+1)$  goes to '0'. In this way, cascading a strong-arm based comparator with an SR latch, we get a strong-arm flip-flop. The advantages of this strong-arm flip-flop are zero static power consumption and full rail to rail swing.

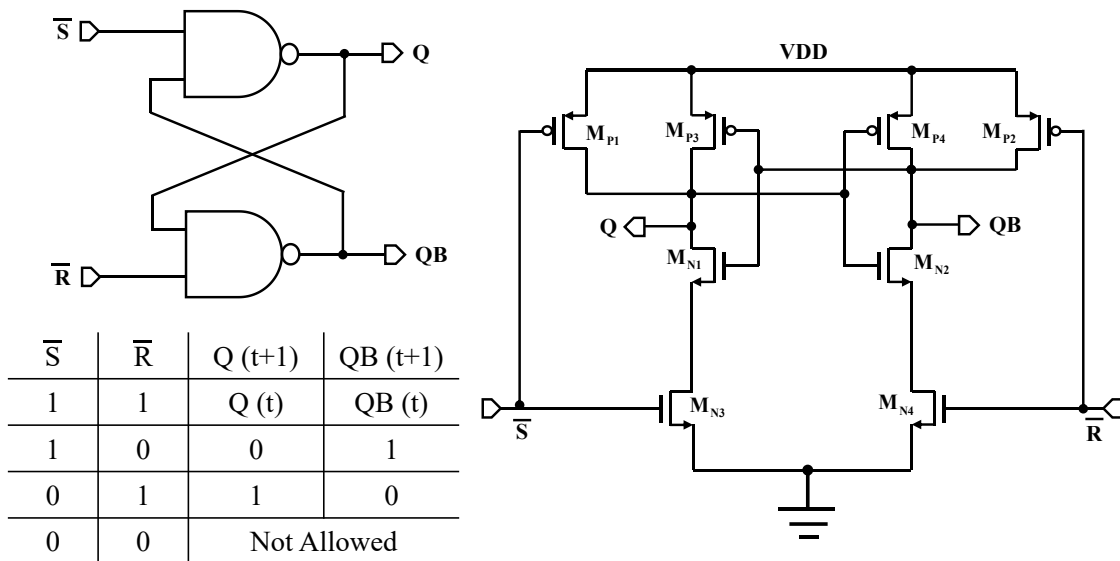


Figure 2.41: SR Latch implementation and truth table.

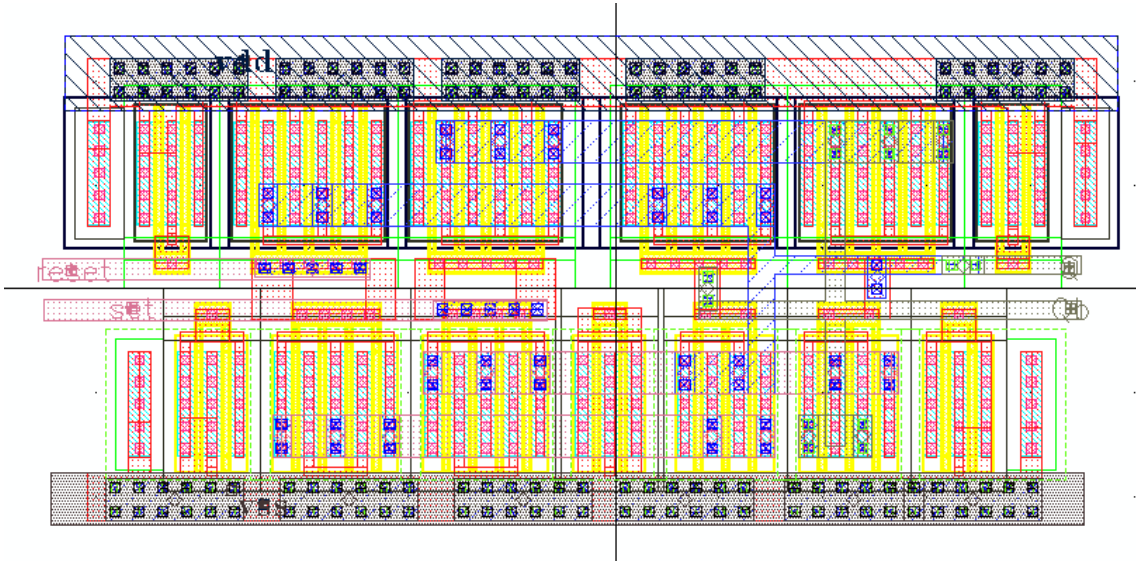


Figure 2.42: Layout of the implemented SR latch.

### 2.6.5. Reference Muxing Timing Issue

The receiver architecture utilizes the output of the edge comparators to place the references of the data comparators. The references of the data comparators have to settle before they are clocked. The timing diagram of the SHs and comparators for the edge and data are shown in Figure 2.43. For CH0, the edge SH is clocked by  $P_{315}$ , i.e. the pulse rising in-line with  $\Phi_{315}$ . The edge comparators are clocked 0.5 UI later using  $\Phi_0$ , and at the same time, the data signal is sampled using  $P_0$ . As the generated pulses have 3 UIs of hold time, the data comparators have to be clocked within these 3 UIs. In this receiver, we have clocked the data comparators using  $\Phi_{180}$ , which is 2 UI later. The simulation results of Figure 2.44 shows that reference settles within 10% of its final value which is within 150 ps. Therefore, in this case, it gives 50 ps timing margin.

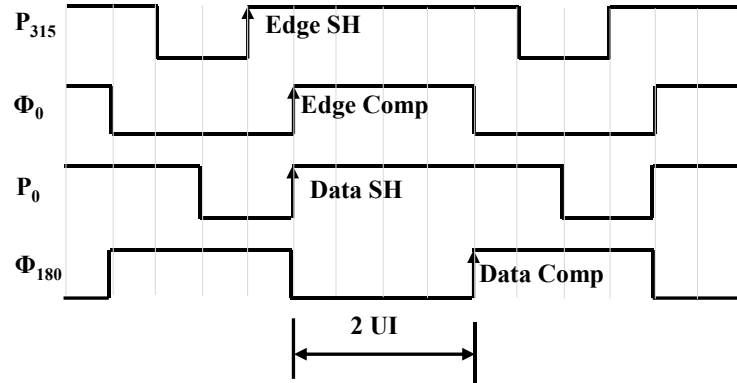


Figure 2.43: Edge and data SH and comparator clocking for CH0.

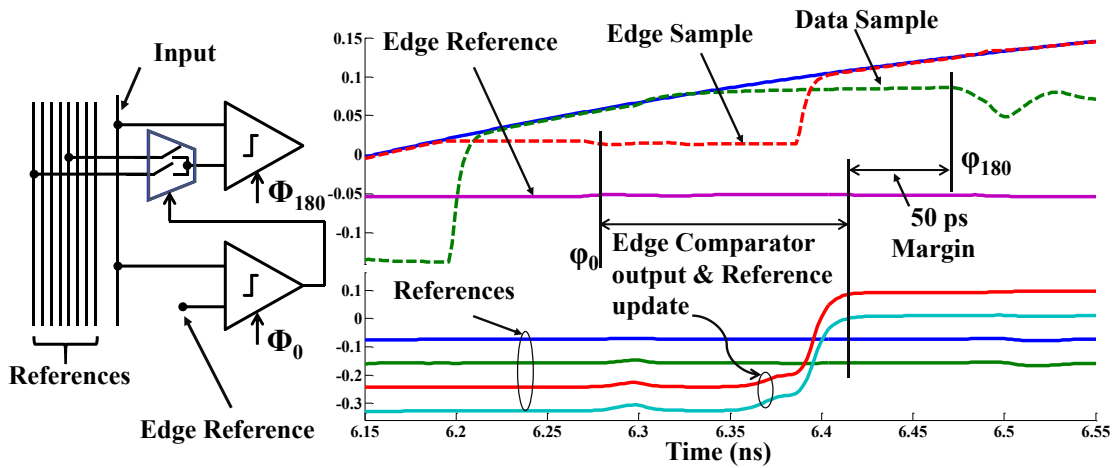


Figure 2.44: Reference settling behavior.

### 2.6.6. Sequence DFE Critical Timing

The sequence generator gives the sequence DFE four 4-bit sequences which are differentiable using post-cursors,  $B_{+1}$  and  $B_{+2}$ . In this quadrature system, DFE feedback is shown in Figure 2.45. In this 2-tap DFE,  $B_{+2}$  feedback comparison is done using the usual XOR operation. However, like all DFE architecture, 1<sup>st</sup>-tap DFE feedback timing is critical here. To facilitate this timing, for  $B_{+1}$  feedback, sequences are pre-calculated for XOR comparison which save one gate delay. Whenever  $B_{+1}$  feedback arrives, the data has to pass through just one MUX (Figure 2.45). The logic implemented is given by the equations below:

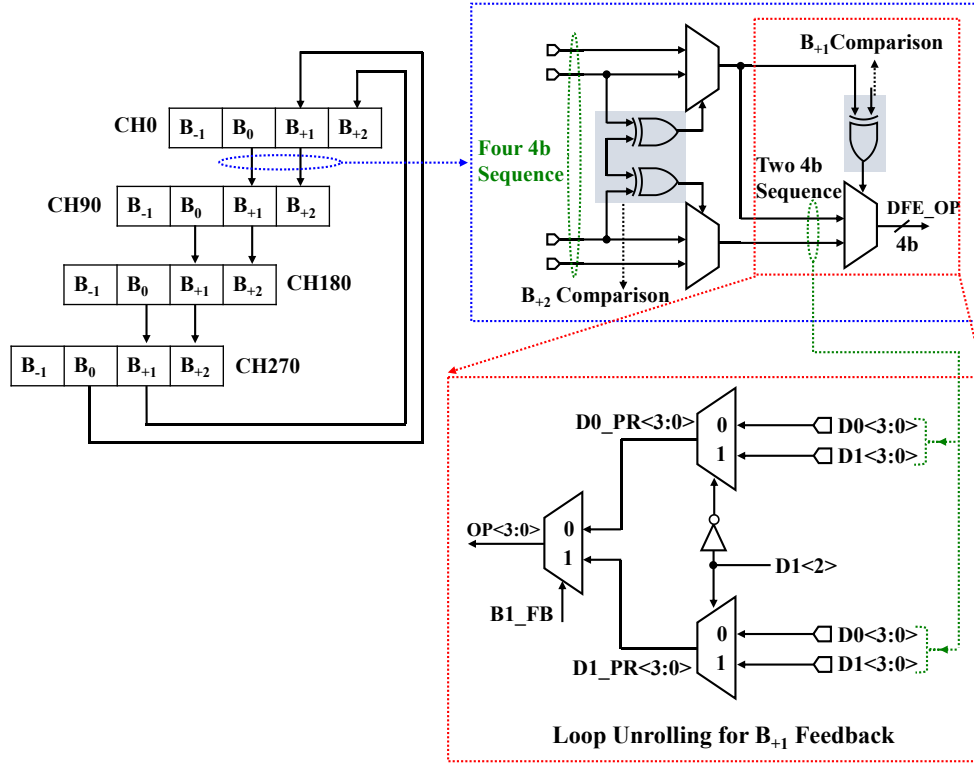


Figure 2.45: Sequence DFE and loop unrolling for  $B_{+1}$  feedback (red box).

$$D0\_PR<3:0> = D0<3:0> \bullet D1<2> + D1\_PR<3:0> \bullet \overline{D1<2>} \quad 2.11$$

$$D1\_PR<3:0> = D0<3:0> \bullet \overline{D1<2>} + D1\_PR<3:0> \bullet D1<2> \quad 2.12$$

$$\begin{aligned} OP<3:0> &= D0\_PR<3:0> \bullet \overline{B1\_FB} + D1\_PR<3:0> \bullet B1\_FB \\ &= \overline{B1\_FB} [D0<3:0> \bullet D1<2> + D1\_PR<3:0> \bullet \overline{D1<2>}] + \\ &\quad B1\_FB [D0<3:0> \bullet \overline{D1<2>} + D1\_PR<3:0> \bullet D1<2>] \\ &= D0<3:0> [\overline{B1\_FB} \bullet D1<2> + B1\_FB \bullet \overline{D1<2>}] + \\ &\quad D1<3:0> [\overline{B1\_FB} \bullet \overline{D1<2>} + B1\_FB \bullet D1<2>] \\ &= D0<3:0> [B1\_FB \oplus D1<2>] + D1<3:0> [\overline{B1\_FB \oplus D1<2>}] \quad 2.13 \end{aligned}$$

## 2.7. Implementation & Experimental Results

The implemented quadrature receiver shown in Figure 2.46 occupies only 0.23 mm<sup>2</sup> with each time-interleaved path taking 240 μm X 240 μm. The digital sequence decoder consumes only 35 mW of power out of which 14 mW is taken in digital operation (Figure

2.47). The additional clock recovery circuit consumes around 9 mW. The test setup with all the required instruments is shown in Figure 2.48. Different amount of channel loss is realized by having different lengths of coaxial cable. Each time with different channel loss, the single bit response of the channel is observed and the dominant tap values are measured. These tap values are used to set the reference values for the comparators. Tunable current sources of the reference generator of the chip allow us to do the tuning of references for different channels. The received signal and the decoded half-rate time sequence DAC output are put on top of one another in Figure 2.49. Each time the decoded sequence DAC output relates close to the received signal after the passive equalizer. The 10 Gb/s input eye and the 16 level sequence detector output eye from one of the four channels are shown in Figure 2.50. Figure 2.51 shows the measured recovered clock of the system with only 11.56 ps peak to peak jitter. Without any transmit equalization, 4-bit sequence decoder operates error-free over a 27 dB loss channel with 90 mV voltage margin and 25 ps timing margin (Figure 2.52).

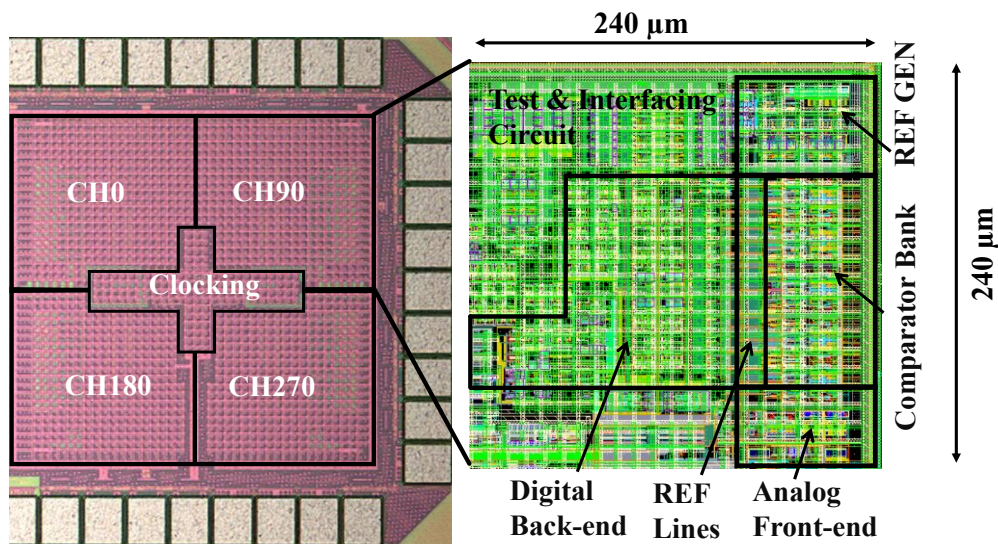


Figure 2.46: Implemented prototype in 65nm CMOS.

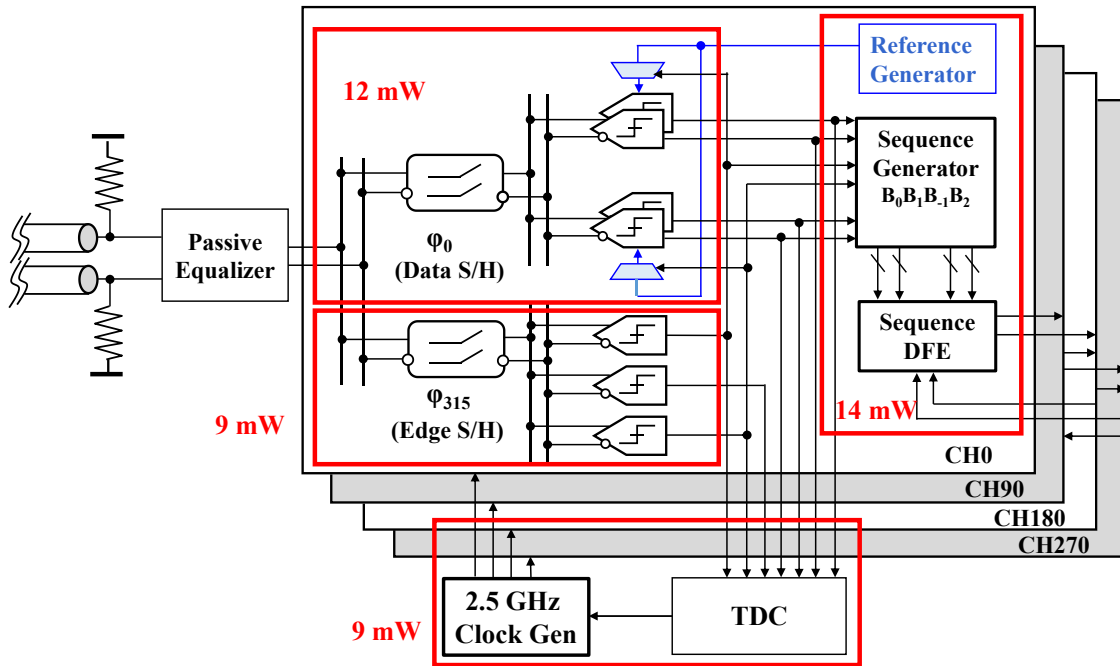


Figure 2.47: Complete receiver with its power consumption.

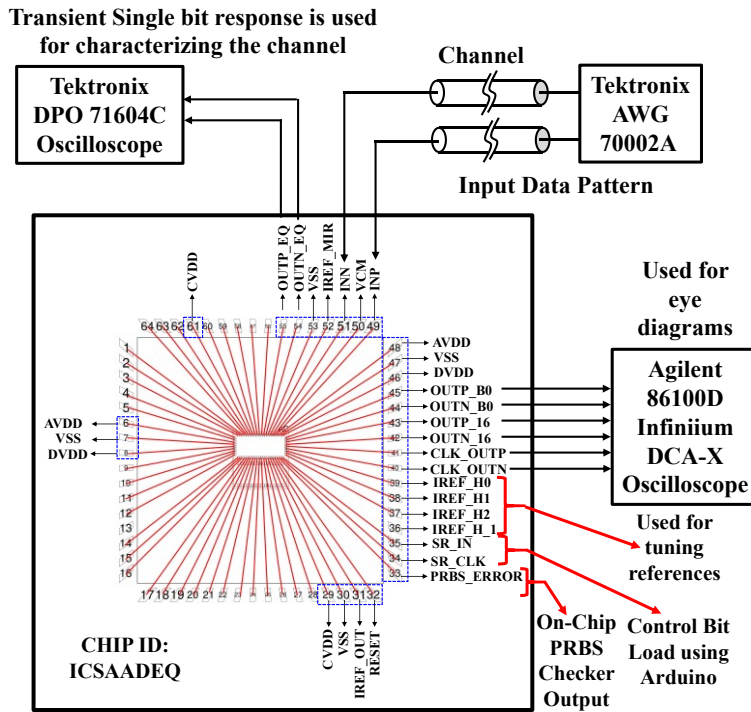


Figure 2.48: Pin diagram and test setup of the prototype.

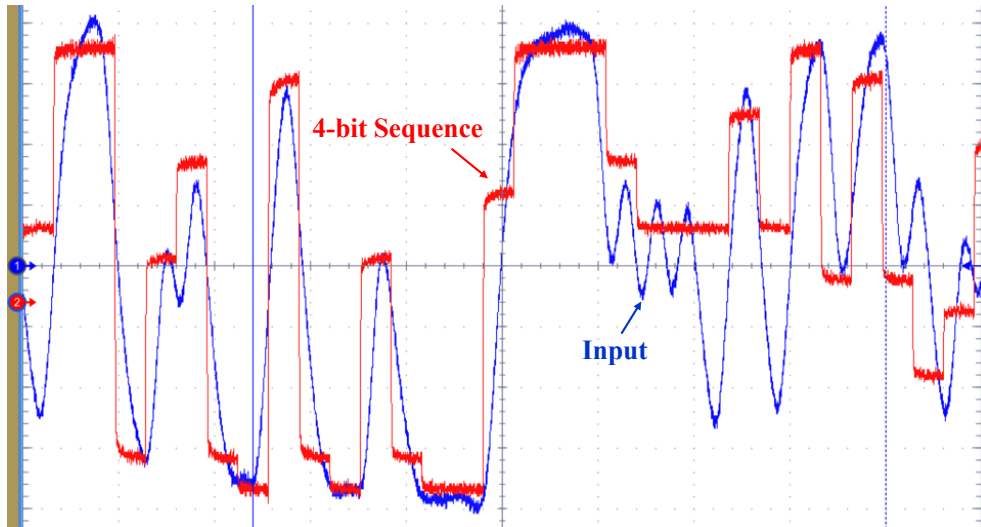
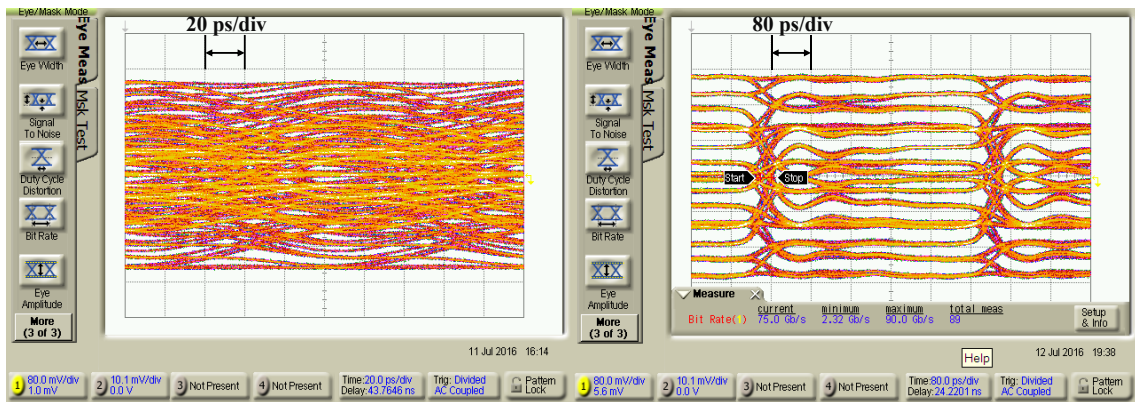


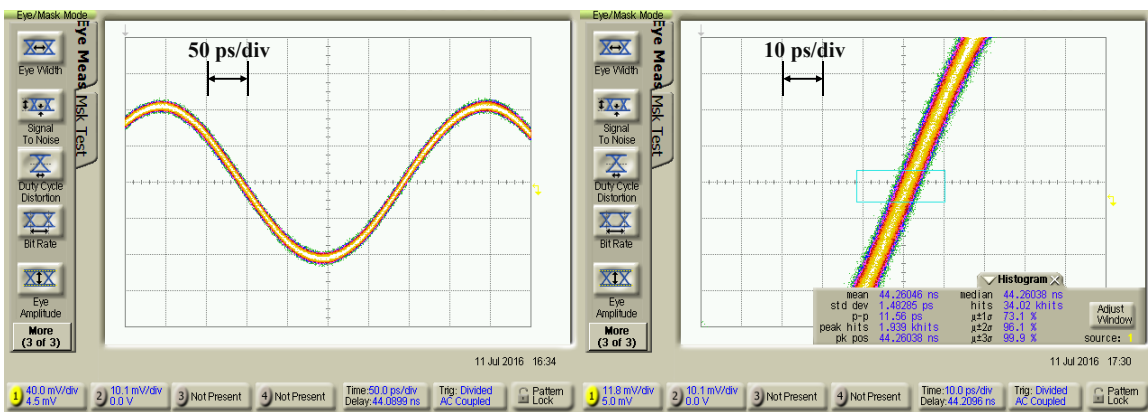
Figure 2.49: Measured half rate 4-bit sequence DAC output.



(a) 10 Gb/s Input Eye

(b) 16-level output eye of the sequence decoder

Figure 2.50: Measured 10 Gb/s input eye (a) and 16 levels output eye of sequence detector (b).

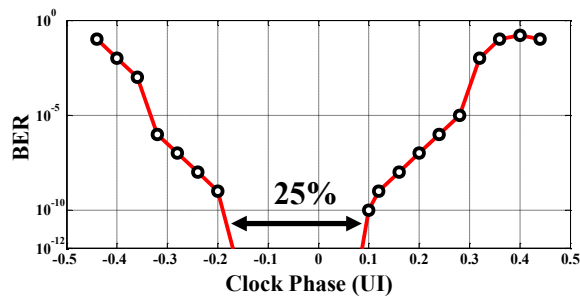


(a) Recovered Clock Eye

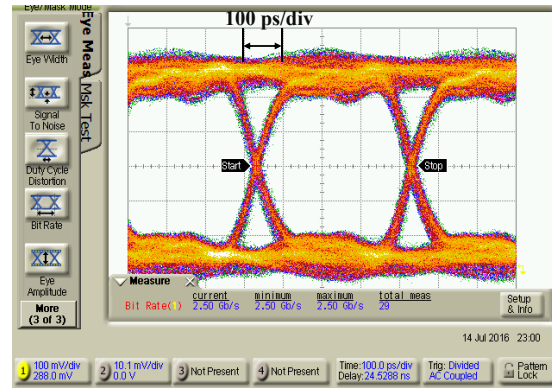
(b) Recovered Clock Histogram

Figure 2.51: Measured 2.5 GHz recovered clock eye (a) and histogram (b).





(a) BER bathtub



(b) Recovered Data Eye

Figure 2.52: BER bathtub (a) and the recovered 2.5 Gb/s PRBS (Pseudo-random bit sequence) checked data eye coming out of the chip using 1-bit DAC (50Ω buffer).

The comparison between this work and the existing state-of-art ADC-based solutions is listed in Table 2.1. Chen *et al.* [12] and Zhang *et al.* [16] worked with 4-way time-interleaved flash ADC architecture with baud-rate timing recovery in place. Although Zhang *et al.* can compensate channel loss up to 34 dB, the additional DSP required for that consumes 500 mW of power. Shafik *et al.*[11] demonstrated a power efficiency of 8.7 pJ/bit using a 32-way time-interleaved SAR ADC without any clock recovery system. The architecture can compensate up to 25.3 dB channel loss, which gives it a figure of merit (FoM) of 0.344 pJ/bit/dB. This work improves the state-of-art by compensating 27dB channel loss at 10 Gb/s consuming only 35 mW. Therefore, the design achieves a power efficiency of 3.5 pJ/bit and FoM of only 0.13 pJ/bit/dB.

Table 2.1: Performance Summary of the 10 Gb/s Receiver.

	Chen JSSC'12[4]	Zhang ISSCC'13[16]	Shafik ISSCC'15[11]	This work
Equalizer Architecture	4x Variable Ref. Flash ADC	4x Rectified Flash	32xSAR	4x Sequence DFE
Data Rate	10 Gb/s	10.3125 Gb/s	10 Gb/s	10 Gb/s
Technology	65nm	40nm	65nm	65nm
Supply Voltage (V)	1.1	N/A	1	1.2
Compensated Channel loss	29/23 dB @10 Gb/s	34 dB @10.3125 GS/s	25.3 dB @10 GS/s	27 dB @10 GS/s
BER	$<10^{-12}$	$<10^{-12}$	$<10^{-10}$	$<10^{-12}$
Timing Recovery	Baud-rate	Baud-rate	None	Data-Edge Sampled
Power Consumption (mW)	Rx- 130	ADC-195 DSP- 500	ADC - 79 Dig. EQ - 8	Sampler - 12 Digital - 14 Clocking - 9
Area (mm <sup>2</sup> )	0.26	0.82	0.81	0.23
Efficiency (pJ/bit)	13/10.6	67.4	8.7	3.5
FoM (pJ/bit/dB)	0.45/0.46	1.98	0.344	0.13

## **Chapter 3.**

### **A 16 Gb/s Direct Digital Sequence**

### **Detector with Data Trace-back and**

### **Equalizer in 65nm CMOS**

The previous chapter introduced the concept of sequence decoding where we use the ISI components in a constructive way to recreate the time sequence of bits transmitted from the other side of the channel. This naturally leads to a fundamentally different way of processing the pre-cursor. In the traditional approach, the pre-cursor is reduced at the cost of SNR whereas here the pre-cursor ISI is used to predict the next bit. This opens up the potential for better equalization strategy that results in higher voltage margin. Due to the fixed supply, SNR will degrade especially when higher order modulations are introduced. The system may benefit from techniques as discussed in this chapter by achieving lower BER. The main concept of the pre-cursor ISI utilization is to correct and mitigate the DFE error propagation. Since the next bit at time  $T_{N-1}$  is part of the sequence that is selected through the sequence DFE, one UI later at time  $T_N$  we can compare the current bit with the next bit from the previous cycle to detect error. In addition to error detection, we can also correct the error. The error correction mechanism involves three steps. First, we identify bit decisions that are not corrupted by the DFE – we name those decisions as high

confidence decisions. We only use such high confidence decisions for error correction. Second, we make use of the pre-cursor to generate the next bit. Rather than generating a single sequence, we generate two potential sequences differentiable based on the next bit. Third, when the bit is detected with high confidence, we use that to select the sequence with the next bit that matches the high confidence correct bit.

The chapter begins with performance limiting factors of the receiver architecture described in Chapter 2. Section 3.1 discusses two cases where noise limits the performance of the current architecture. Section 3.2 discusses what we can do to overcome these limitations to give better noise immunity to the system. This section arrives at the technique of 1-bit data trace-back. Sequence generation and feedback for data trace-back are also discussed here. Section 3.3 describes the modified block diagram of the receiver. The receiver in this chapter also works at a higher speed than the previous one. Section 3.4 shows the experimental results and summary of the modified receiver performance with data trace-back.

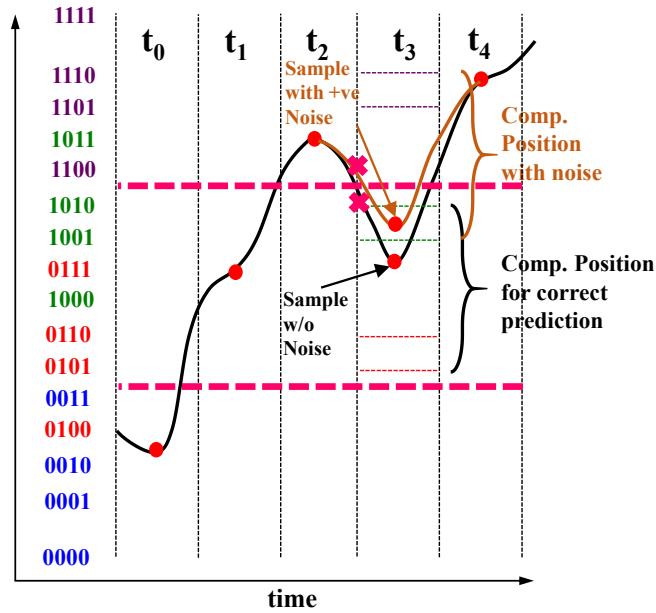
### **3.1. Noise Tolerance Limit of Current Receiver**

The receiver architecture discussed in Chapter 2 is capable of compensating 27 dB channel loss with a voltage margin of around 10% of total signal space at BER of  $10^{-12}$ . Section 2.5 of Chapter 2 discussed how we can recover from errors introduced by noise and prediction by the edge comparators. If there is an error in prediction by the edge comparators, we can verify that prediction using the floating data comparators. Errors caused by the floating data comparators can be fixed using the DFE feedback.

In this section, we will discuss two cases where noise limits the performance of the current receiver architecture. In case I, the edge comparators predict incorrectly and the verification using the data comparators cannot fix that error. In case II, the edge comparators predict correctly, but the data comparators see a large noise for in-bank comparison.

### 3.1.1. Case I

Figure 3.1 shows a case of simultaneous error demonstrated by both the edge and the data comparators due to noise at time  $t=t_3$ . In the correct operational case, the prediction by the edge comparators will send the data comparators to the middle position with references from the 10 and 01 banks. In this position, two of the floating comparators,  $C_{F1,0}$ , give 1 as output and the other two,  $C_{F3,2}$ , give 0. So, the generated sequences from the 10 bank are 1000 and 1001; the generated sequences from the 01 bank are 0110 and 0111.  $B_{+2}$  feedback will come at first and as it is 1, 1001 sequence from the 10 bank and 0111 from the 01 bank will be forwarded for  $B_{+1}$  feedback sorting. At  $t=t_3$ ,  $B_{+1}$  feedback is 1. The correct DFE output should be 0111 sequence. For the error case, the edge sample of the incoming data was pushed up by noise and this noise sent the data comparators to the top. The position verification system does not work here as the bottom data comparators see data that is higher than its reference level. Here,  $C_{F0}=1$  and the rest are 0. For the top position, sequences from the 11 bank are 1100 and 1101 and from the 10 bank are 1001 and 1010. If we assume the DFE feedbacks are still there and working the way they should work, the final output will be 1101. In this situation, both  $B_0$  and  $B_{-1}$ , that are not in feedback, are incorrect.

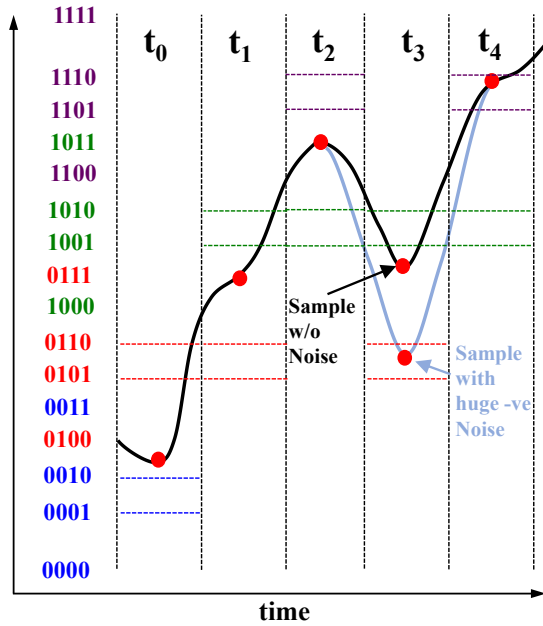


Noise	Position	$C_{F3}$	$C_{F2}$	$C_{F1}$	$C_{F0}$	Gen. Seq.	$B_{+2}$ Feedback	After $B_{+2}$ Feedback	$B_{+1}$ Feedback	Output
w/o	Mid	0	0			1 0 0 1	1	1 0 0 1	1	0111
						1 0 0 0		0 1 1 1		
			1	1	0 1 1 0	0 1 1 1				
					0 1 1 0	0 1 1 1				
w +ve noise	Top	0	0			1 1 0 1	1	1 1 0 1	1	1101
						1 1 0 0		1 0 1 1		
			0	1	1 0 1 0	1 0 0 1				
					1 0 0 1	1 0 0 1				

Figure 3.1: Case I – error in sequence detection.  $B_0$  and  $B_{-1}$  detected incorrect.

### 3.1.2. Case II

Figure 3.2 shows an error by the in-bank floating comparators at time  $t=t_3$ . The correct case is already discussed in Section 3.1.1. In this case, the prediction by the edge comparators is correct. At time  $t=t_3$ , the possible banks are 10 and 01. Due to large noise, the incoming signal is pushed down and  $C_{F1}$  becomes 0; however, in the ideal case, it should be 1. So, from the 01 bank possible sequences are 0101 and 0110. After DFE feedback, 0101 is selected as the output which implies an error in  $B_{-1}$  detection.



Noise	Position	C <sub>F3</sub>	C <sub>F2</sub>	C <sub>F1</sub>	C <sub>F0</sub>	Gen. Seq.	B <sub>+2</sub> Feedback	After B <sub>+2</sub> Feedback	B <sub>+1</sub> Feedback	Output
w -ve noise	Mid	0	0	0	1	1 0 0 1	1	1 0 0 1	1	0 1 0 1
						1 0 0 0		0 1 0 1		
						0 1 1 0		0 1 0 1		
						0 1 0 1				

Figure 3.2: Case II – error in sequence detection. B<sub>-1</sub> detected incorrect.

### 3.2. Improving Noise Tolerance of Current Receiver

The current architecture as discussed in Chapter 2 uses the DFE feedback and detects the main cursor, B<sub>0</sub>. The precursor, B<sub>-1</sub>, comes as a by-product. The by-product can be used further to verify the detection of B<sub>0</sub>. The current architecture works with the prediction from the edge comparators that is later verified using the floating data comparators. Instead of the prediction, we can use sub-ranging ADC approach, where the floating data comparators will be placed based on the data sample rather than the edge sample.

### 3.2.1. Setting Fixed Data Comparators

Two fixed data comparators can be used instead of using the edge comparators. These two data comparators will directly drive the reference multiplexers of the floating comparators. The reference levels of these two comparators are well defined (Figure 3.3). The two fixed comparators have to differentiate between banks.  $C_{FIX1}$  is there to differentiate between the 11 and 01 banks. So, it can be placed in between the top of the 01 bank and the bottom of the 11 bank, i.e. in between levels 0111 and 1100. If  $C_{FIX1}=0$ , the reference levels of the 01 bank will be passed to the floating comparators. If it is 1, the reference levels of the 11 bank will be passed to the floating comparators. The same is true for the 10 and 00 banks.  $C_{FIX0}$  is in between 0011 and 1000 levels.  $C_{FIX0}=0$  implies references of the 00 bank will be passed to the floating comparators and  $C_{FIX0}=1$  implies references of the 10 bank will be passed to the floating comparators.

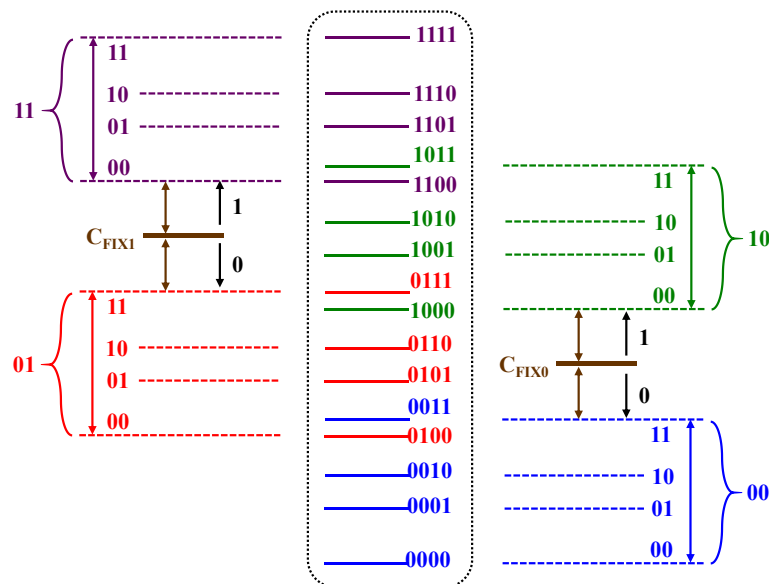


Figure 3.3: Introduction of two fixed data comparator reference instead of fixed edge comparators.



### 3.2.2. Case I

The fixed data comparators fix the error of case I with little margin (Figure 3.4(a)). If there is higher noise, it will push the signal up a little bit more again producing error (Figure 3.4(b)). The additional noise sends the floating comparators to the top again and the verification logic cannot recover: the error prevails. To fix this issue, we can use the decoded  $B_{-1}$  bit of the sequence. Figure 3.5 shows how the concept of using  $B_{-1}$  bit works. At time  $t=t_3$ , both  $B_{+2}$  and  $B_{+1}$  are 1. To illustrate the concept, we can start with the feedback first.  $B_{+2}$  and  $B_{+1}$  DFE feedback leave 4 possible combinations out of 16 possible sequences; two combinations are from the 11 bank and two are from the 01 bank. As the fixed data comparators place the floating comparators to the top position, it is highly unlikely to have 0101 as a possible sequence. 0101 is the bottom-most possible sequence while the fixed data comparators placed the floating comparators on the top. So, we can

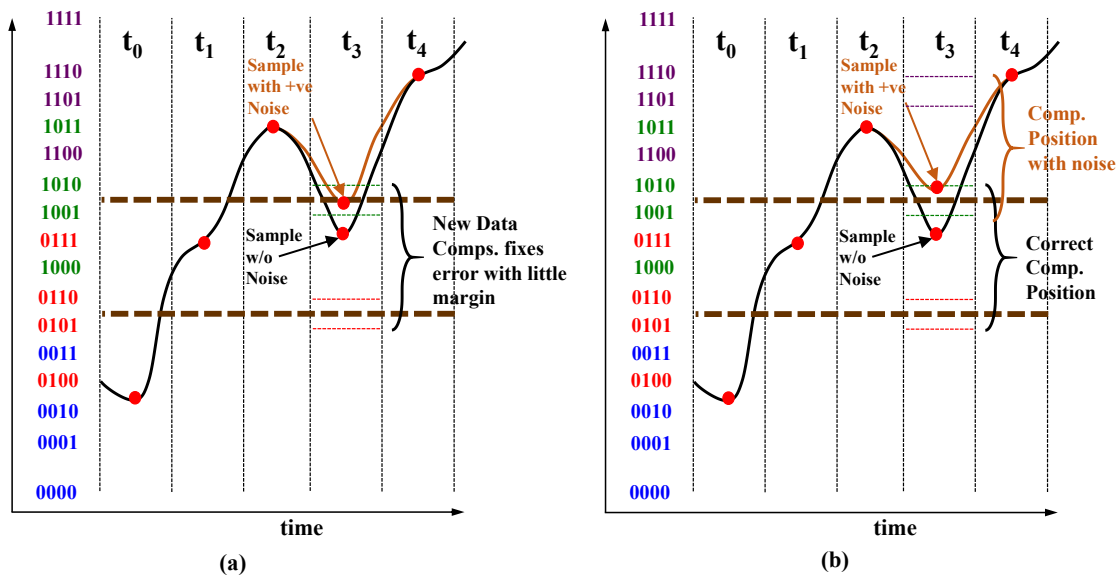


Figure 3.4: Fixing error with little margin using fixed data comparators.

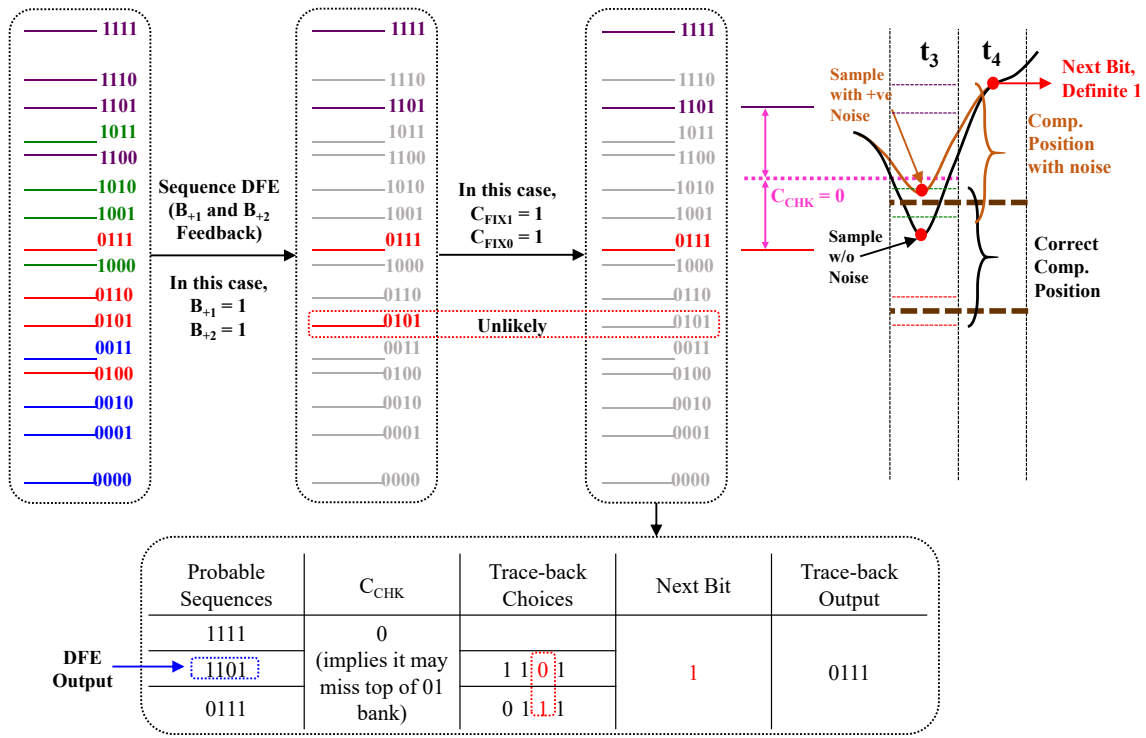


Figure 3.5: Introduction of check comparator to find out whether it may miss a bank or not.

neglect 0101 sequence. When the fixed data comparators are placing the floating comparators for the 11 and 10 banks, the most probable sequence we can miss is the top of the 01 bank i.e. 0111 sequence. To see if there is any chance of missing the top of the 01 bank, we can use a check comparator. In this case, the placement of the check comparator is in between the top of the probable missing bank and the in-bank bottom sequence having the same values of  $B_{+2}$  and  $B_{+1}$ . The check comparator at  $t=t_3$  gives 0. Out of the probable 3 sequences, this implies that the top most can be neglected. As the next bit is a definite 1, it can come as feedback and correct the bit decision. Here, the tracing back options are 1101 and 0111 sequence. LSB+1 bit is  $B_{-1}$  bit. After comparing  $B_{-1}$  bit with LSB+1 bit, the output will be 0111, which is the correct output. If we go back to what we had at the

beginning of case I, DFE output was already 1101. We added the check comparator and it said the data might be from the missing bank. As the next bit was a definite 1, we checked back to see if we decoded the correct sequence (0111) and found out we had the wrong sequence (1101). We corrected the error using  $B_{-1}$  bit.

When the floating data comparators are on the top position (11 and 10 bank), the check comparator checks the probability of a missing sequence from the nearest bank (01 bank). Table 3.1 shows all the possible cases of missing the probable banks and the corresponding comparator placement. When the signal is supposed to be in the middle (10 and 01 bank), there are two cases: (i) missing the bottom of the 11 bank (1100 sequence) and (ii) missing the top of the 00 bank (0011 sequence). 1100 has  $B_{+1}=1$  and  $B_{+2}=0$ . The closest in-bank sequence having the same  $B_{+1}$  and  $B_{+2}$  is 0110. So, the check comparator reference will be set in between 1100 and 0110. The case of missing the top of 00 bank is similar to the case of missing the top of 01 bank. So, for the mid position we need two comparators to check whether there is any probable missing bank. However, for the top and the bottom position, we need only one comparator.

Table 3.1: Reference placement for checking probable bank miss.

Position	Available Banks	Most Probable Bank Missed	Most Probable Sequence outside Banks	$B_{+1}$ of the Most Probable Sequence	$B_{+2}$ of the Most Probable Sequence	Closest Ref. in bank having same $B_{+1}$ & $B_{+2}$	Check Comp Placement
Top	11 10	01	0111	1	1	1101	1101 0111
Mid	10 01	11	1100	1	0	0110	1100 0110
		00	0011	0	1	1001	1001 0011
Bottom	01 00	10	1000	0	0	0010	1000 0010

The references for the check comparator can also be multiplexed using the fixed data comparators. However, when the signal is on the top or on the bottom, one of the check comparators will not be clocked and that one will receive common mode voltage ( $V_{CM}$ ) as reference. Figure 3.6 illustrates placement of all the comparator references; for the check comparators, it also describes why it is being used.

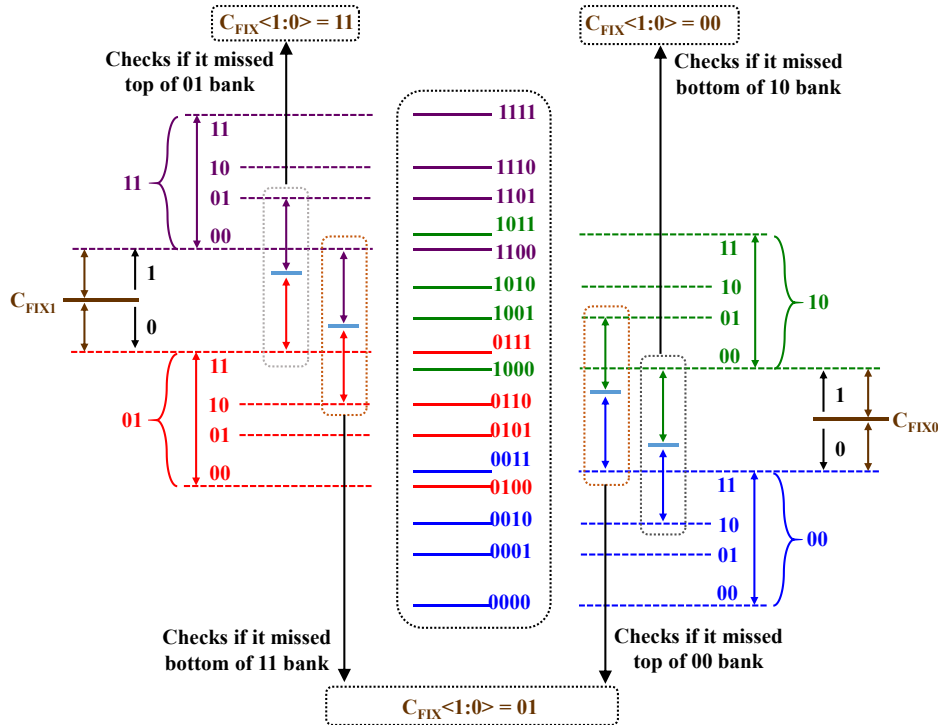
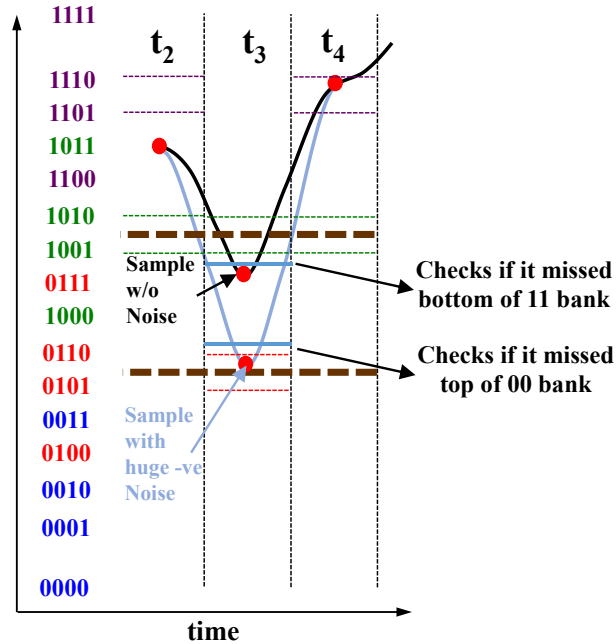


Figure 3.6: Overall reference placement of the architecture with trace-back.

### 3.2.3. Case II

The check comparators also resolve the issue of Section 3.1.2. At  $t=t_3$ , the check comparators check whether the system has missed the bottom of the 11 bank and the top of the 00 bank. As the DFE output was 0101 in case II, the check comparator that checks whether it missed the 11 bank becomes the one to be considered for trace-back. As it says 0, there is no chance of data being on top. Another probable sequence is the 0111 in-bank

option. As the next bit is a definite 1, it comes for trace-back and fixes the sequence as shown in Figure 3.7.



$C_{CHK}$	Trace-back Choices	Next Bit	Trace-back Output
0 (implies it is not missing bottom of 11 bank)	0 1 0 1	1	0111
	0 1 1 1		

Figure 3.7: Resolving the issue of case II.

### 3.2.4. Trace-back Sequence Generation

There are two types of sequence generated for trace-back in this system: (i) outside bank trace-back sequence (case I) and (ii) within bank trace-back sequence (case II). Table 3.2 lists the logic showing which trace-back option will be considered for the different outputs of the check comparators.

Table 3.2: Trace-back choice logic.

Bank	Check Comp Output	Trace Back Choice
11	0	Outside
	1	Within
10	0	Outside
	1	Within
01	0	Within
	1	Outside
00	0	Within
	1	Outside

Now, for outside bank trace-back, there may be an error in  $B_0$  and  $B_{-1}$ . So, the sequence choices given for the trace-back are:

- DFE selected one ( $B_0B_1B_{-1}B_2$ )
- The other option having  $B_0 B_{-1}$  flipped.

Within bank trace-back implies there may be an error in  $B_{-1}$  only. So, the sequence choices given for the trace-back are:

- DFE selected one ( $B_0B_1B_{-1}B_2$ )
- The other option having  $B_{-1}$  flipped.

### 3.2.5. Conditions of Data Trace-back

For the data trace-back discussed so far, we have always mentioned the next bit to be definite 1/0. When both the fixed data comparators detect the received signal to be on the top or on the bottom position and the floating top and bottom check comparators verify the sample location, the data is considered to be a strong 1/0. The logic for detecting strong

1/0 is listed in Table 3.3. If the next bit is not strong 1/0, the incorrect DFE feedback will propagate. The strong 1/0 logic ensures the next bit is not dependent on the DFE feedback for its main cursor decision. If the next bit is a strong 1/0, it can be used for error detection by the DFE.

Table 3.3: Detecting Strong 1/0.

Fixed Data Comparators		Floating Top/Bottom Check Comparators		Bit
C <sub>FIX1</sub>	C <sub>FIX0</sub>	C <sub>CHK_TOP</sub>	C <sub>CHK_BOT</sub>	
1	1	1	x	Strong 1
		0		Not strong 1
0	0	x	1	Not strong 0
			0	Strong 0

We cannot do trace-back when top-bottom-mid verification has already resolved the fixed data comparators errors.

### 3.2.6. Improved Noise Margin

The data trace-back adds two additional comparators to increase the noise margin of the system. The required number of comparators increases by 2 only. We can modify Eq. 2.3 to get the required number of comparators:

$$C_{\text{Sequence DFE and T race-back}} = 2^{M-1} + \frac{2^M 2^L}{\text{PredictionFactor}} + 2. \quad 3.1$$

The noise margin of the modified architecture of the sequence DFE can be defined as half the distance between two banks having similar  $B_{+1}$  value as discussed in Section 2.4. For a 4 tap sequence DFE having one precursor ( $h_{-1}$ ), main cursor ( $h_0$ ) and two post-cursors ( $h_{+1}$  &  $h_{+2}$ ), the noise margin can be written as

$$NM_{\text{Sequence DFE}} = \frac{h_0 - h_{-1} - h_{+2}}{2}. \quad 3.2$$

The noise margin of the data trace-back increases as we consider similar values of  $B_{+2}$  as well. However, the additional noise margin will be there if and only if the next bit is a strong 1/0.

Figure 3.8 shows the comparison of noise margins for these three cases. As the number of bits of the ADC increases, the noise margin also increases. However, the noise margin achieved by the 4 tap sequence DFE with data trace-back is always higher than the ADC-based DFE. In this figure,  $h_0=0.26$  mV,  $h_{+1}=0.16$  mV,  $h_{-1}=0.12$  mV and  $h_{+2}=0.08$  mV.

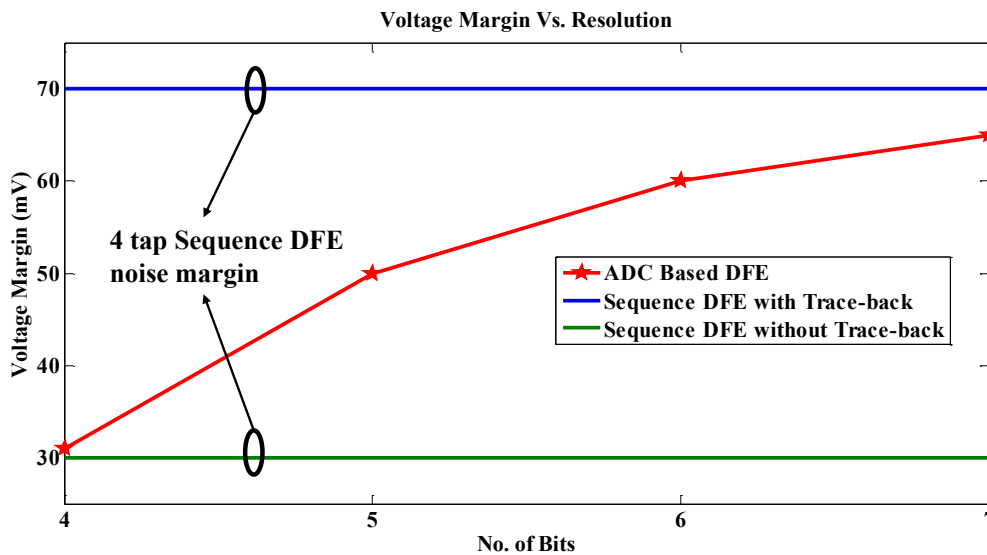


Figure 3.8: Comparison of noise margins of 4 tap sequence DFE with and without data trace-back with ADC-based DFE.



### 3.3. System Design

The overall implementation of the quadrate system is shown in Figure 3.9. The receiver uses similar passive equalizer at the front end as used in Chapter 2. The system requires two fixed data comparators, four floating data comparators and two check comparators – overall eight comparators. The output of the fixed comparators places the references of the floating and check comparators. For the check comparators, when in the top/bottom, only one of them is clocked. If the comparator is not clocked, it gets common mode voltage as reference input. The system uses an external clock as it does not have any built-in timing recovery in place. The receiver works with 16 Gb/s input data with each time-interleaved path running at 4 GHz.

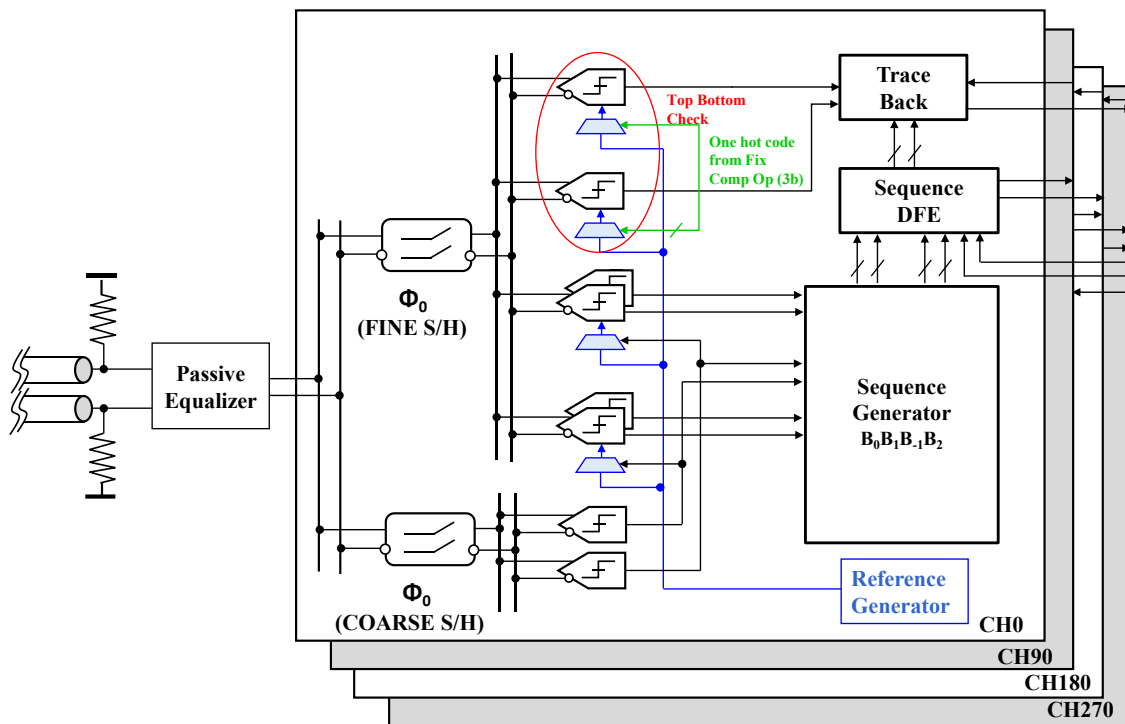


Figure 3.9: System architecture of quad-rate receiver with data trace-back.

### 3.3.1. Reference Muxing Timing

As the system is running at a higher speed than the receiver described in Chapter 2, the comparators are resized to get similar timing margin for reference muxing. As we are not using the prediction by the edge comparators, coarse (SH for fixed data comparators) and fine (SH for floating comparators) SHs sample the data at  $P_0$ , i.e. the pulse rising at  $\Phi_0$ . The coarse comparators are clocked instantly using a delayed version of  $\Phi_0$ . Now, we have time up to the hold time of  $P_0$ , which is less than 3 UI. The fine and top/bottom check comparators are clocked 2 UI later using  $\Phi_{180}$ . The outputs of the fixed comparators take around 100 ps to update the references of the floating comparators (Figure 3.10). It still gives 25 ps timing margin to clock the floating comparators to achieve 16 Gb/s operation. The check comparators are clocked using the same scheme.

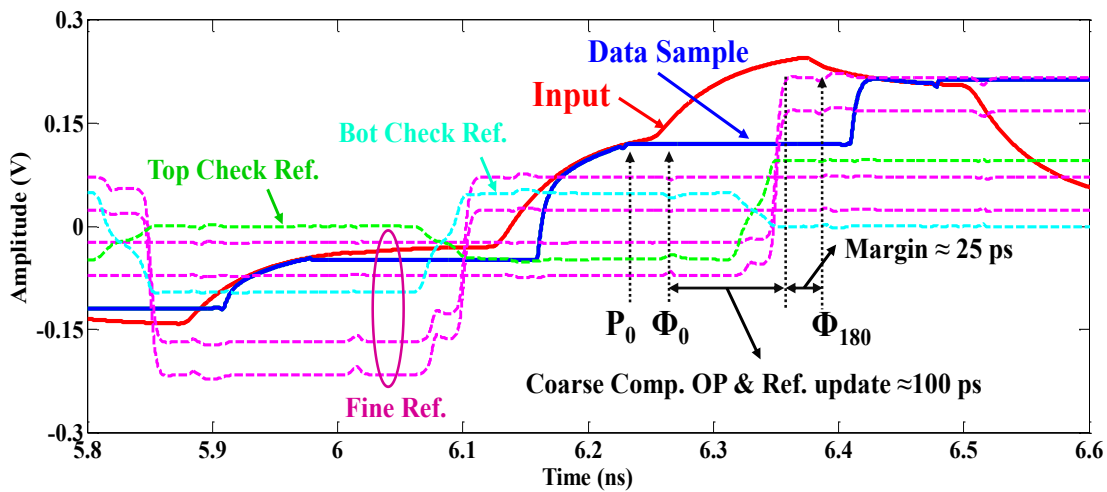


Figure 3.10: Reference Muxing for quadrate channel running at 4 GHz.

### 3.3.2. DFE and Trace-back Feedback

The DFE feedback works as described in Section 2.6.6. Loop unrolled architecture is carried over from the previous architecture described in Figure 2.33. The new addition here is a sequence generator after the DFE output. The sequence generator generates a choice for trace-back as described in Section 3.2.4. The DFE output and generated sequence have different values of  $B_{-1}$ . Now,  $B_{-1}$  feedback comes only if the next channel has strong 1/0 and the current channel does not have strong 1/0. After this comparison, we get trace-back output.

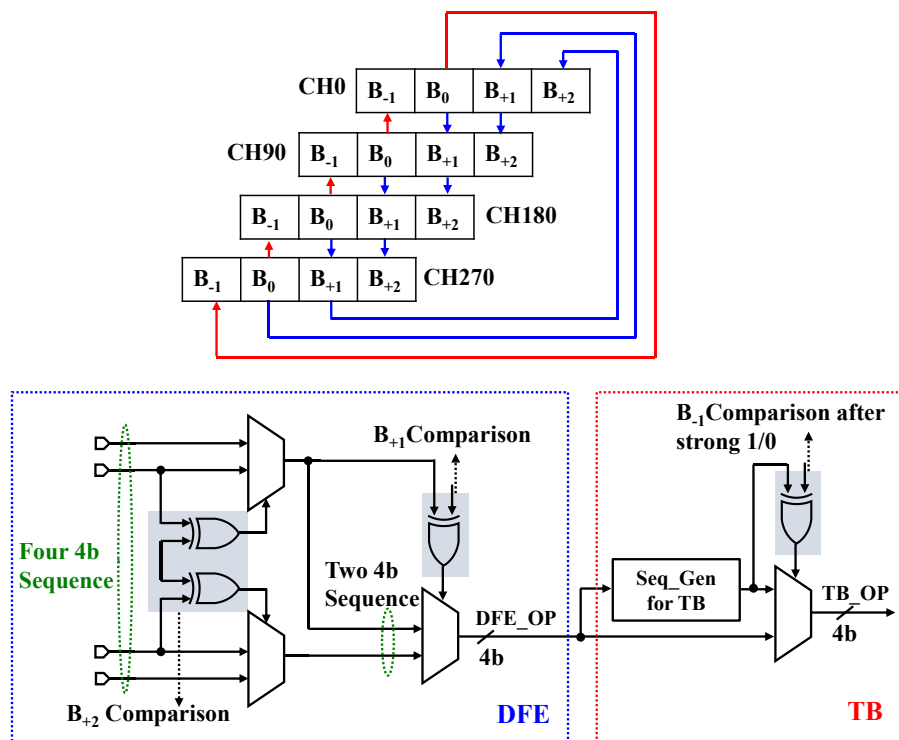


Figure 3.11: Feedback for DFE and Trace-back.

### 3.4. Experimental Results

The prototype of the architecture implemented in TSMC 65nm is shown in Figure 3.12. The design takes an area of  $300\ \mu\text{m} \times 560\ \mu\text{m}$  with each channel taking  $300\ \mu\text{m} \times 140\ \mu\text{m}$  area. This prototype works with 16 Gb/s input data. The pin diagram and test setup of the chip with all the required instruments are shown in Figure 3.13. Different amount of channel loss is realized by having different lengths of coaxial cable as discussed in Chapter 2. Each time with different channel loss, the single bit response of the channel is observed and the dominant tap values are measured. These tap values are used to set the reference values for the comparators. Tunable current sources of the reference generator of the chip allow us to do the tuning of references for different channels. The outputs of the sequence DFE with and without data trace-back is taken separately. Figure 3.14 shows the channel response for which the receiver was tested. Figure 3.15 shows the input after passive equalization and the channel output side-by-side. The 16 Gb/s input after passive equalizer has 4 dominant taps. The generated clock and PRBS checked recovered data eye is given in Figure 3.16. The BER curves in Figure 3.17 (BER bathtub curve) and Figure 3.18 (2D color-map of BER showing voltage margin in Y-axis and timing margin in X-axis) show improvement in BER after the data trace-back. For 16 Gb/s operation, the receiver achieves a BER of  $10^{-10}$  when DFE works alone. However, with added trace-back BER of the receiver improves to  $10^{-12}$ . The BER plots of Figure 3.18 show improvement in voltage margin of the modified architecture. The design consumes 71.8 mW power with DFE and data trace-back in action. However, in low loss case, the data trace-back can be turned off to save power. Without the trace-back, it consumes only 55 mW power. Table 3.4 summarizes overall receiver performance. At higher loss cases ( $\sim 35$  dB), both the trace-

back and DFE is applied; whereas in lower loss cases, the system can achieve desirable performance using DFE only while consuming less power. For 16 Gb/s operation, when both the trace-back and DFE are active, the receiver consumes 71.8 mW at an efficiency of 4.4875 pJ/bit. The figure of merit (FoM) during 35 dB operation turns out to be 0.1282 pJ/bit/dB, which is better than the FoM of 0.1375 pJ/bit/dB during DFE-only operation as it compensates less loss.

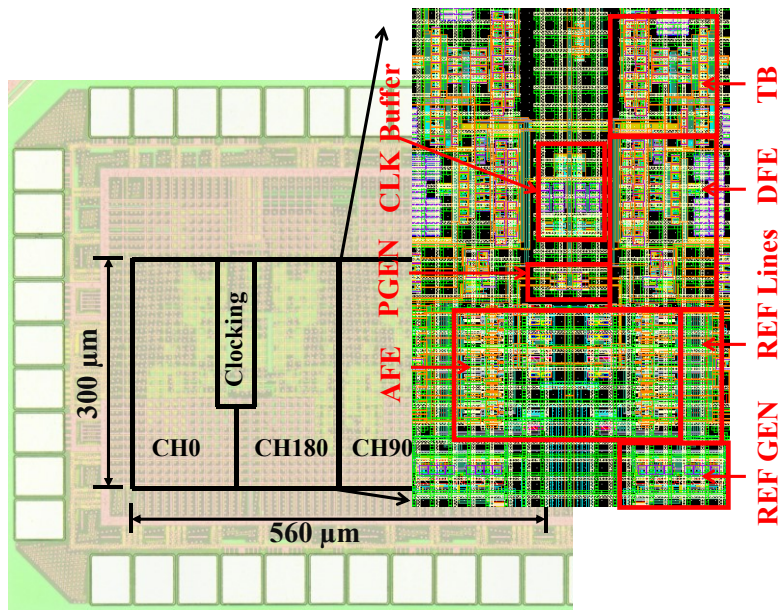


Figure 3.12: Implemented prototype in 65nm.

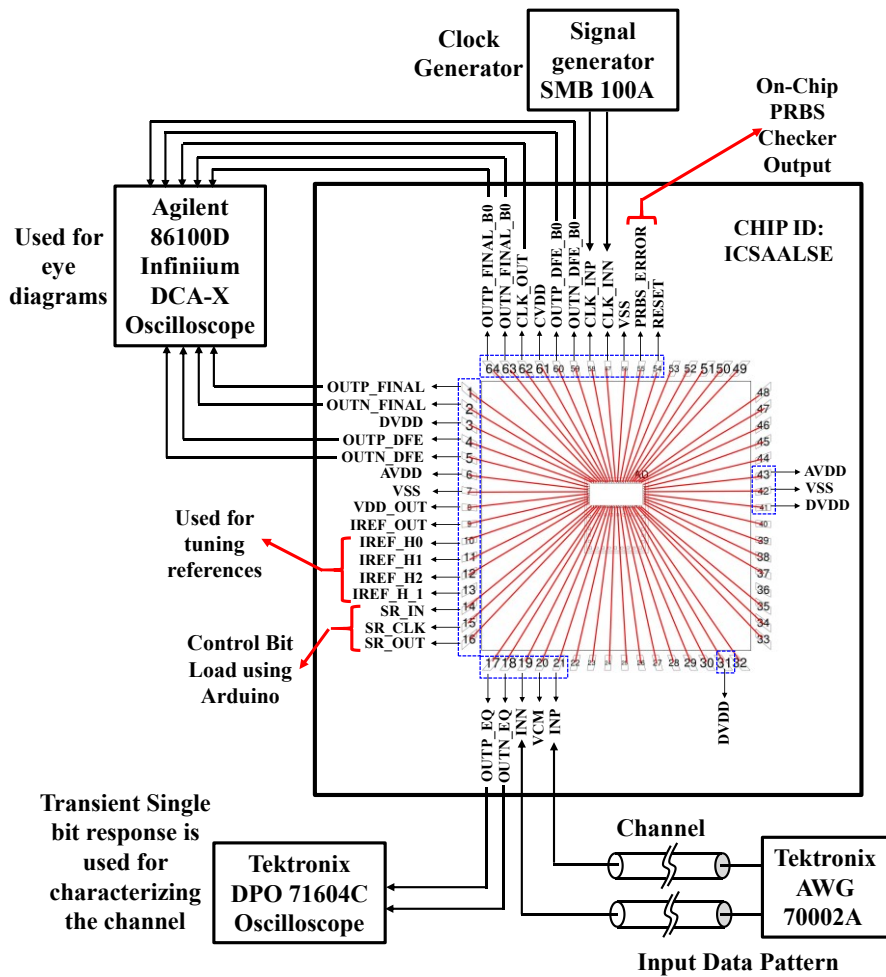


Figure 3.13: Pin diagram and test setup of the prototype.

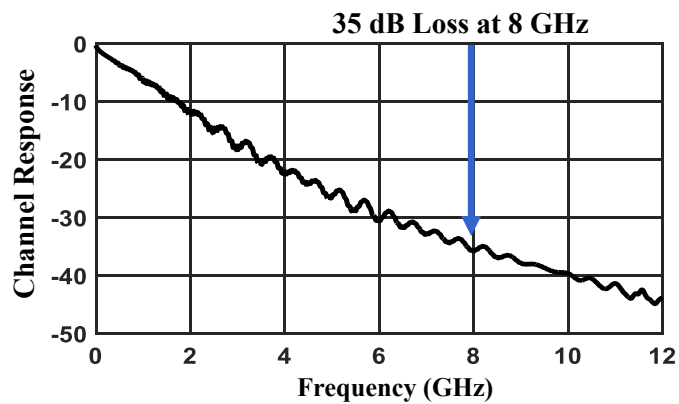
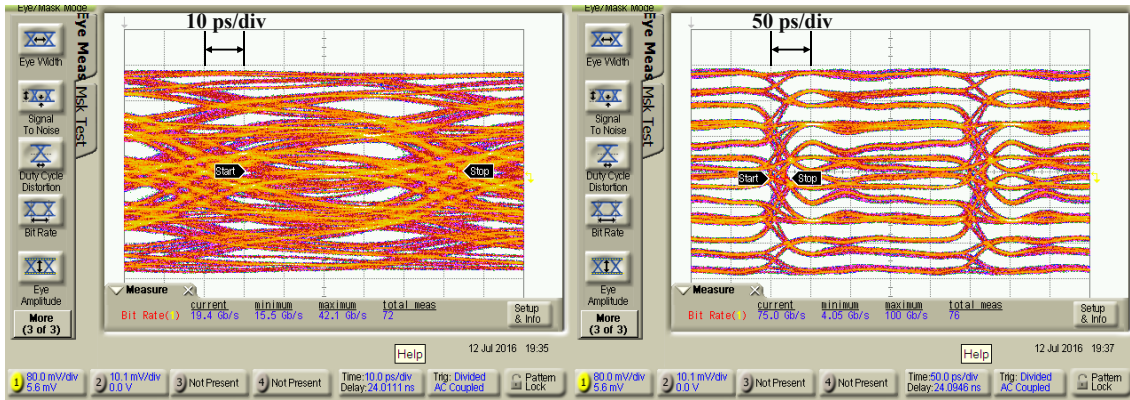


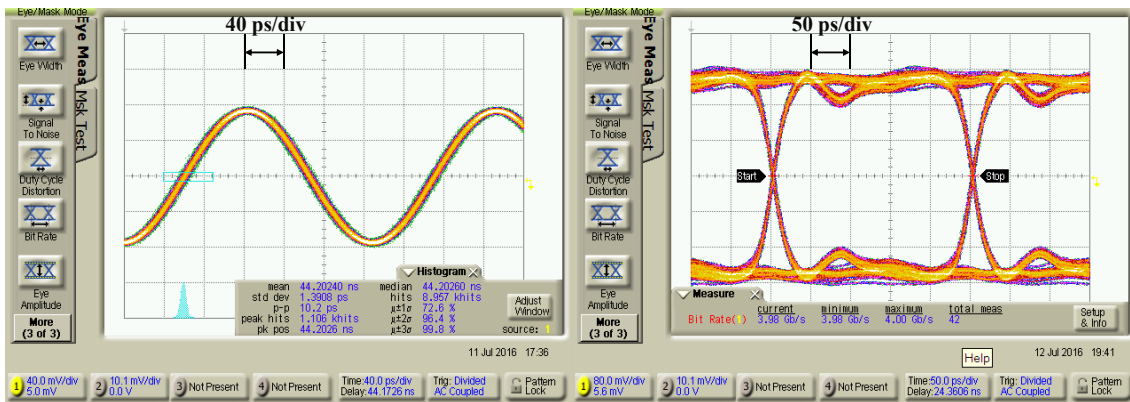
Figure 3.14: Channel Response at 8 GHz.



(a) 16 Gb/s input eye after passive equalizer

(b) 16-level output eye of the sequence decoder

Figure 3.15: Measured 16 Gb/s input eye after passive equalizer (a) and 16-level output eye of the sequence decoder (b).



(a) 4 GHz Clock Eye

(b) Recovered 4 Gb/s Data Eye

Figure 3.16: Measured 4 GHz clock eye (a) and PRBS checked recovered 4 Gb/s data eye (b).

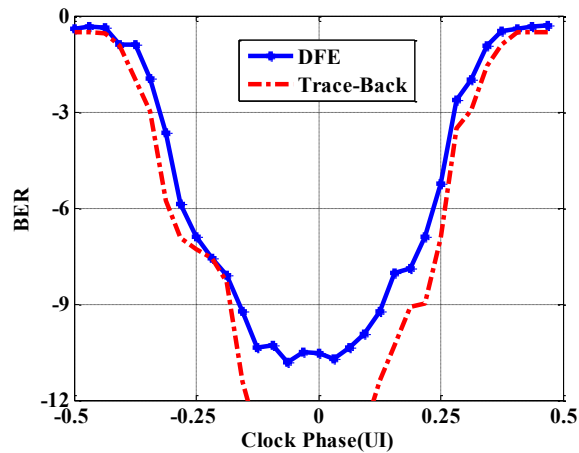


Figure 3.17: BER bathtub comparing DFE and Trace-back.

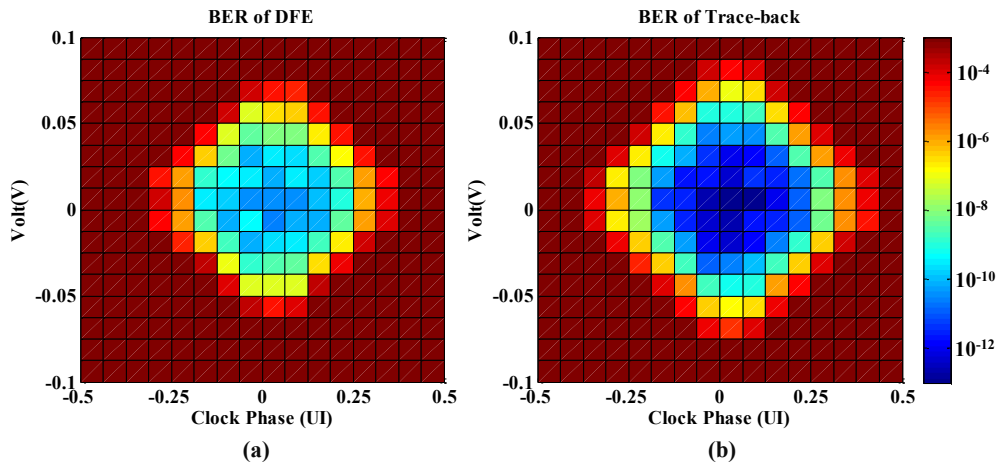


Figure 3.18: 2D color-map of BER showing voltage margin in Y-axis and timing margin in X-axis of sequence DFE (a) and data trace-back (b).

Table 3.4: Receiver Summary.

	Shafik ISSCC'15 [11]	Hossain VLSI'16 [12]	This Work
Technology	TSMC 65 nm	TSMC 65 nm	TSMC 65 nm
Supply Voltage (V)	1	1.2	1.2
Data Rate (Gb/s)	10	10	16 Gb/s
Active Die Area (mm <sup>2</sup> )	0.81	0.23	0.17
Timing Margin (UI)	0.2	0.25	0.25
BER	10 <sup>-10</sup>	10 <sup>-12</sup>	10 <sup>-12</sup> (DFE and TB) 10 <sup>-10</sup> (DFE only)
Loss Compensation (dB)	25.3	27	35
Power Consumption (mW)	87	35	71.8 (DFE and TB) 55 (DFE only)
Power Efficiency (pJ/bit)	8.7	3.5	4.4875 (DFE and TB) 3.4375 (DFE only)
FoM (pJ/bit/dB)	0.343	0.13	0.1282 (DFE and TB) 0.1375 (DFE only)



# **Chapter 4.**

## **Burst Mode Optical Receiver with 10ns Lock Time Based on Concurrent DC Offset and Timing Recovery Technique**

This chapter describes a low power low latency 7-10 Gb/s burst mode DC-coupled receiver for photonic switch networks. The chapter begins with a discussion of the conventional optical receivers in section 4.1. Section 4.2 covers the state-of-art DC and timing recovery methods applied in burst-mode optical receivers and their performance limiting factors. Section 4.3 discusses overall architecture of the proposed receiver. Analog front-end comprising of a trans-impedance amplifier (TIA) followed by three stage of amplifiers, DC recovery loop, timing recovery loop and timing skew correction loop are also discussed in details in section 4.3. Section 4.4 covers the implementation of the system in IBM 0.13 $\mu$ m and the measured experimental results. Section 4.5 compares the proposed burst mode optical receiver with the current state-of-art optical receivers. The implemented trans-impedance amplifier (TIA) is also compared with the state-of-art TIAs.

## 4.1. Conventional Optical Receiver

The continuous growth of cloud computing, “big data” and social media applications is placing enormous bandwidth demand on data communication networks. The interconnection between servers, which has traditionally relied on electrical cables, is becoming a major bottleneck in today’s data networks, as we approach the limits of copper wires in terms of speed, loss and power consumption. To meet this demand, optical interconnects are already finding their place in data centers to connect racks which are only a few meters apart. Optics can provide large transmission bandwidth, which reduces latency and fast processing speed. Extending these benefits, cross-point network switches can also be optical to provide significant increase in cross-sectional bandwidth. In addition, rapidly reconfigurable optical switching network can significantly improve latency and throughput [25].

There are electrical transceivers to interface these optical switches. However, unlike existing SerDes, these optical transceivers need to accommodate different optical power levels when the cross point is reconfigured. Interestingly, existing passive optical network (PON) provides such functionalities. Unfortunately, to fit in the data center solution space, existing PON solutions need to improve significantly: First, in existing PON receivers TIA and LA are located on one IC and clock and data recovery (CDR) is located on a different IC ([26], Figure 4.1). It is desirable to integrate them into a single complete receiver and then to integrate multiple receivers in a single IC. Second, the power efficiency of burst mode PON transceivers is in the order of 50 to 100 pJ/bit. This efficiency has to improve to better than 10 pJ/bit considering the cooling cost of data centers.

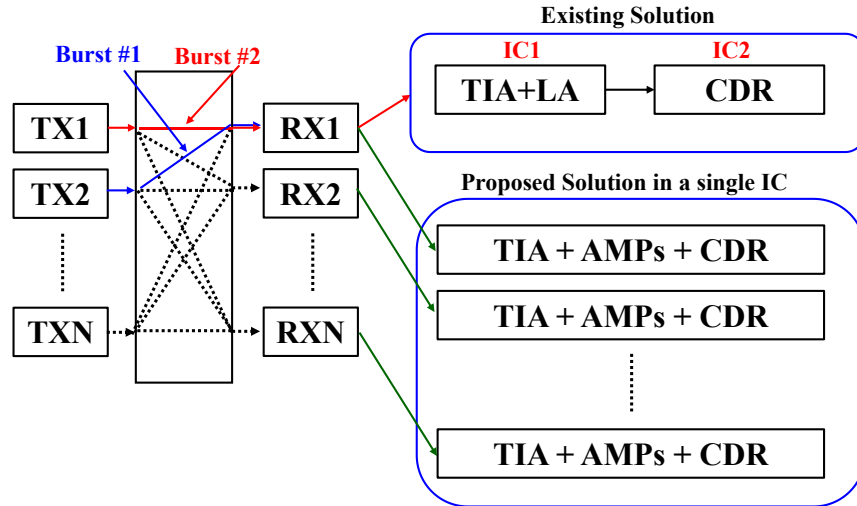


Figure 4.1: Conventional vs. proposed implementation of the optical receiver.

Currently, most of these links are VCSEL based, and their performance is excellent but lacks in integration density and bandwidth. To meet next generation's bandwidth demand, VCSEL has to improve its bandwidth and at the same time, we have to look for alternative solutions such as silicon-photonic modulators that have higher integration density. Optical interconnects based on silicon integrated photonic waveguides and devices aim to replace metal wires with optical dielectric waveguides on the same CMOS chip. They have been considered as a promising technology that can meet the projected bandwidth demand, while offering the advantage of being compatible with CMOS electronics.

Energy efficiency of the link has become as critical as its bandwidth. From an application point of view, there is growing demand for energy efficiency improvement over the entire traffic of data including idle mode, not just at the peak of data rate. Therefore, transceivers have to adapt to this bursty nature of data traffic and thereby achieve both the highest bandwidth and excellent efficiency. When used in such power-down mode, it is required

that the transceiver would be able to wake-up, receive and transmit data without significant preamble to keep high burst efficiency. Transceivers require nanosecond scale burst to take advantage of the dynamically reconfigurable high-speed optical switches based on Silicon Photonics. After each switching event, the receiver has to support the data stream that may come from a different transmitter having different modulator with different extinction ratio and different switching loss. In addition, the receiver has to support un-encoded data stream where the data pattern desired to be received can have long strings of consecutive identical data (CID) bits, i.e. long strings of 1's and 0's. In an AC coupled link, a long string of 1's and 0's may cause baseline wandering issue. The length of the non-transition period of the incoming optical signal has to be small compared to the time constant determined by the capacitance used for coupling between optical and electrical interface to avoid wander issue. Encoding can avoid DC balancing problem, but it adds latency and degrades the overall efficiency of the transmission. DC coupled interfaces can easily overcome these issues. However, the receiver has to compensate for the DC offset created due to switching event during its fast lock time in this burst mode format.

## **4.2. Challenges in Burst-Mode Receiver**

In conventional implementations ([26], Figure 4.1), TIA and CDR are designed in two different ICs interfaced with high speed interconnect that mandates sufficient signal swing to meet the signal integrity requirement. Usually, in a 50-ohm environment, meeting the swing specification leads to significant power consumption in the limiting amplifier (LA).

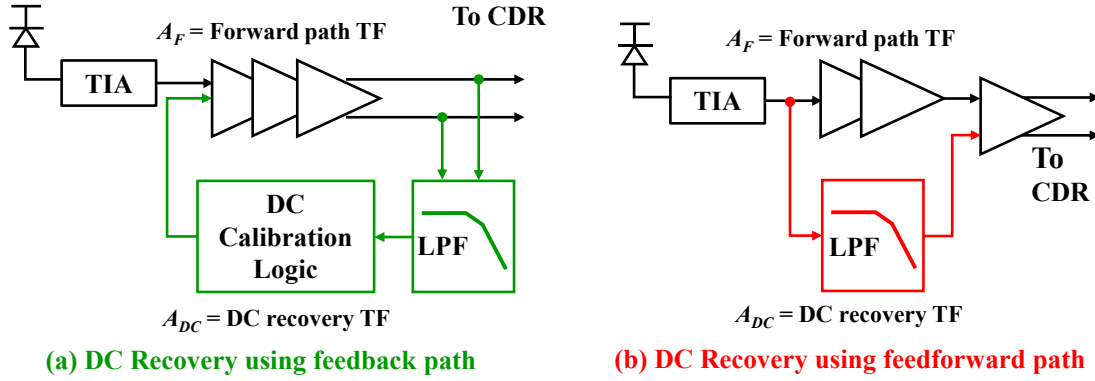


Figure 4.2: Conventional implementation of burst mode receiver with DC offset calibration.

In addition, multiple chip implementations have cost penalty. Therefore, in state-of-art DC-coupled burst mode optical receiver [2], TIA and CDR are integrated on the same chip. However, lock time still remains an issue that can be broken into two components – DC offset correction and timing recovery. DC offset correction loop can be based on either feedback ([2], Figure 4.2(a)) or feedforward ([27], Figure 4.2(b)) concept – in both cases, a low pass filter senses the DC offset and subtracts that at the input of the receiver. Feedback can be implemented in analog domain using V-to-I [28] or in digital using DAC [2]. However, in both cases, the pole frequency of the feedback path ( $\omega_p$ ) sets the tradeoff between lock time and bandwidth. For feedback system, if  $A_F$  with pole frequency  $\omega_f$  and  $A_{DC}$  are the transfer functions of the forward path and DC recovery path respectively, then the system transfer function ( $A_{FB}$ ) can be written as shown in Eq. 4.1. The step response of the feedback system gives us the lock time and the high pass pole frequency  $f_c$  defines the usable bandwidth (Eq. 4.2) [29].

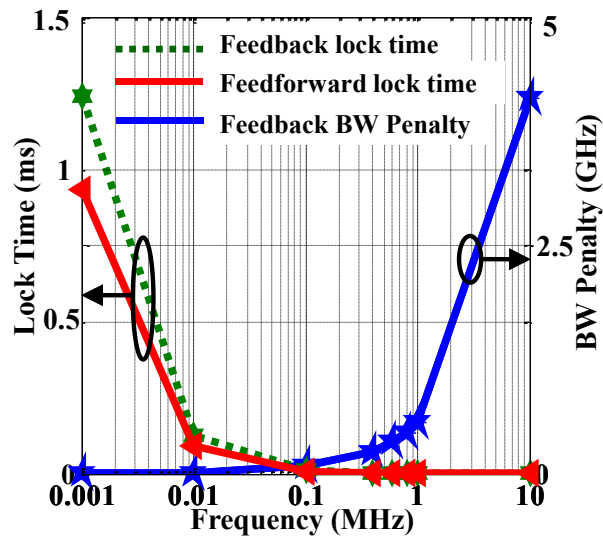
$$A_{FB} = \frac{A_F}{1 + A_F A_{DC}} \quad 4.1$$

$$\text{where, } A_F = \frac{A}{1 + s/\omega_f}, \quad A_{DC} = \frac{1}{1 + s/\omega_p}.$$

$$f_c = \frac{A/2 + 1}{2\pi R_F C_F} \quad 4.2$$

where,  $R_F C_F = \tau =$  Time constant of the LPF.

In general, if the pole ( $\omega_p$ ) in the feedback path is at a relatively higher frequency, feedback loop settles faster, but that also reduces the usable effective bandwidth (Figure 4.3). This problem can be avoided by shifting the feedback pole at a lower frequency, but that also increases the lock time. A good compromise can be to place the pole at 1/10th of the preamble frequency and with additional digital post-processing; this way, it is possible to reduce lock time to less than 50 cycles of the digital clock [2].



(c) Lock Time and BW Penalty vs. Frequency

Figure 4.3: Settling time vs. bandwidth of DC recovery loop.

For feedforward DC recovery concept, the system transfer function is

$$A_{FF} = \frac{1}{1+s/\omega_f} - \frac{1}{1+s/\omega_p}. \quad 4.3$$

Compared to feedback solution, DC settles faster in this concept and bandwidth penalty is set by  $\omega_p$ . As a result, the fundamental trade-off between usable bandwidth and settling time is still significant. In both feedback and feedforward cases of DC recovery, if the recovered DC information during preamble period is not stored using memory, the system faces baseline wandering during CID pattern.

Similarly, feedback based timing recovery loops (Figure 4.4(a)) also suffer from longer lock time issues in burst mode applications. Fortunately, open loop approaches such as injection lock based timing recovery loops have reduced the lock time to few cycles ([30], Figure 4.4(b)). However, open loop timing recovery loops suffer from several limitations; first, VCO drifts away in the absence of injection pulses which leads to poor tolerance to CID and suboptimal lock position. Second, unlike traditional SerDes, where half-rate or quarter rate solutions are often used to reduce power, injection locking is usually done at full rate – therefore, receiver power is usually high. Although injection locking can achieve

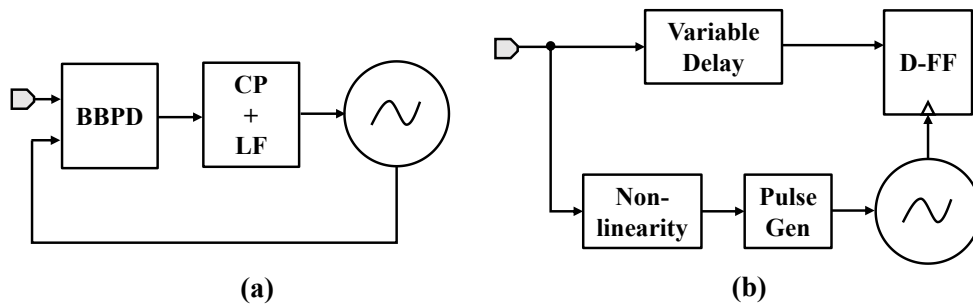


Figure 4.4: Conventional timing recovery loop.

fast lock time, receiver initialization time is still gated by DC offset correction as shown in Figure 4.5. At the beginning, when DC offset is still uncorrected, outputs of the limiting amplifier also have non 50% duty cycle. That means generated injection pulses are spaced such that the generated frequency is at an offset from the correct one. Therefore, when injected with these pulses, VCO does not lock at the correct phase or frequency. Only after the DC offset is corrected, generated pulses are properly spaced to lock the VCO. Although existing works have shown very fast burst mode clock recovery [31], without addressing the DC offset corrections, receiver lock time does not improve.

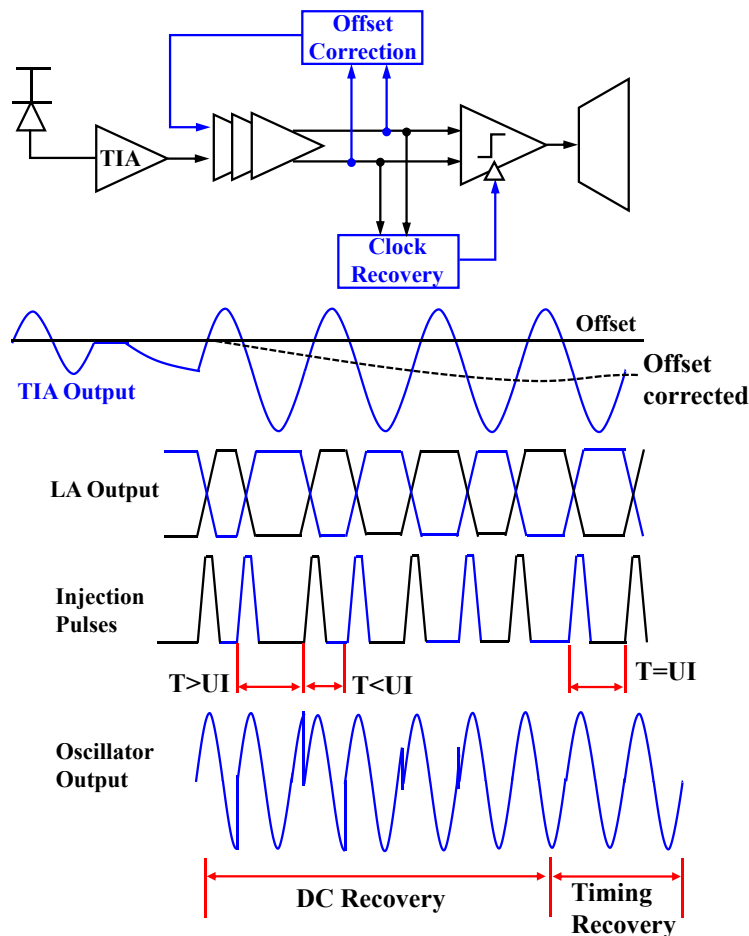


Figure 4.5: Effect of DC on duty cycle of the data.



This work introduces a burst mode quad-rate 7-10 Gb/s receiver architecture for dynamically reconfigurable photonic switch network. An entire receiver including TIA, three stage amplifiers, and CDR are implemented on the same IC; in fact, the inductor-less receiver fits within  $465\ \mu\text{m} \times 265\ \mu\text{m}$  area. Use of limiting amplifier leads to a power and area consuming design; therefore, in this design, we eliminate the limiting amplifier to reduce power. Unlike traditional DC offset correction that relies on sensing DC at the output, we propose a novel DC offset correction that can directly measure DC offset from alternating data. Since there is no LPF needed, this DC offset correction loop can settle significantly faster compared to the conventional approach. To achieve fast wake up, DC offset correction and timing recovery loops run concurrently, and that reduces the lock time to less than 10ns. To achieve a robust solution, timing skew adaptation algorithm is used to get optimum timing margin for the receiver.

### **4.3. Proposed Burst Mode Receiver**

The overall block diagram of the implemented receiver is shown in Figure 4.6. The system consists of an analog front-end (TIA followed by three stage differential amplifiers), a DC recovery loop, an injection lock based quad-rate CDR and a timing skew adaptation loop.

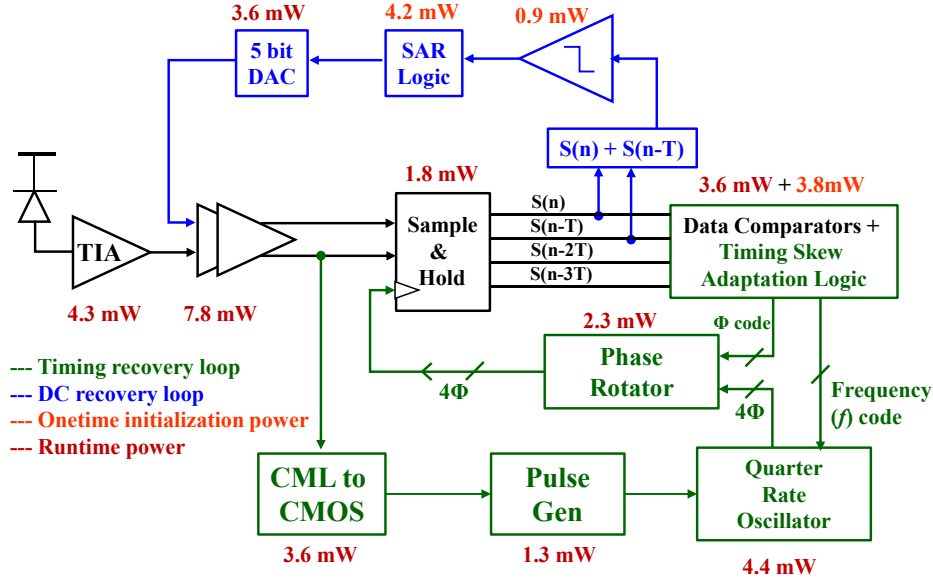


Figure 4.6: Block Diagram of the receiver with power breakdown.

### 4.3.1. Trans-impedance Amplifier (TIA)

TIA receives the current from the photodiode and the single-ended output voltage goes to one of the inputs of differential amplifier chain whereas the other input is recovered DC. A CMOS common gate amplifier is used as TIA followed by a common source amplifier with source degeneration (Figure 4.7). So, trans-impedance gain of the common gate stage is

$$\text{Trans - impedance Gain, } A_{F1} = \left( \frac{\frac{1}{g_{m_2}} + \frac{1}{sC_1}}{\frac{1}{g_{m_1}} + \frac{1}{g_{m_2}} + \frac{1}{sC_1}} \right) R_1. \quad 4.4$$

Voltage gain of the common source stage with source degeneration impedance is

$$\text{Voltage Gain, } A_{F2} = \frac{-g_{m_3} R_3}{1 + g_{m_3} (R_4 \parallel \frac{1}{sC_2})}. \quad 4.5$$

So, overall gain of the forward path is

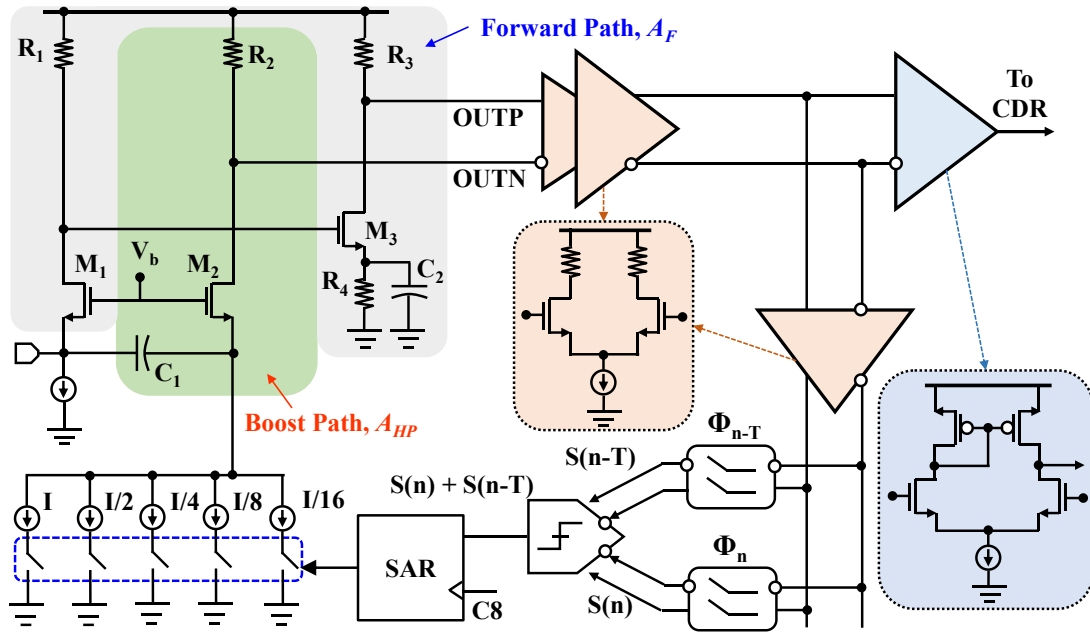


Figure 4.7: Detailed schematic of TIA and its DC recovery loop.

$$A_F = - \left( \frac{\frac{1}{g_{m_2}} + \frac{1}{sC_1}}{\frac{1}{g_{m_1}} + \frac{1}{g_{m_2}} + \frac{1}{sC_1}} \right) R_1 \frac{g_{m_3} R_3}{1 + g_{m_3} (R_4 \parallel \frac{1}{sC_2})}. \quad 4.6$$

The forward path through M2 serves two purposes: first, the SAR DAC creates the DC output (OUTN) that corrects the data dependent DC offset. Second, the signal path through AC coupling capacitor C1 goes through the high pass transfer function,

$$A_{HP} = \left( \frac{\frac{1}{g_{m_1}}}{\frac{1}{g_{m_1}} + \frac{1}{g_{m_2}} + \frac{1}{sC_1}} \right) R_2. \quad 4.7$$

At the input of the differential amplifier, these two paths appear such that the DC components of the signals cancel each other but the high-frequency parts of the signals

combine constructively. By boosting the high-frequency signals, the front-end bandwidth is extended to 11 GHz without using any inductor.

Figure 4.8 shows simulated gain of the analog front end with and without high pass path.

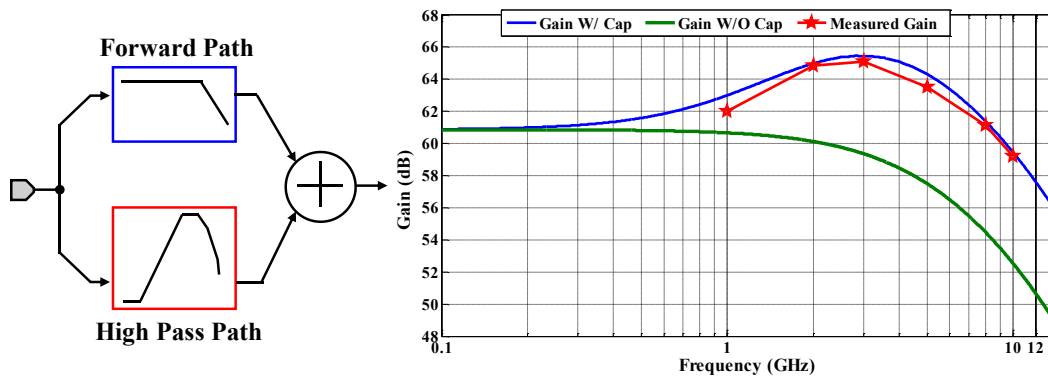


Figure 4.8: TIA transfer function and its simulated and measured gain.

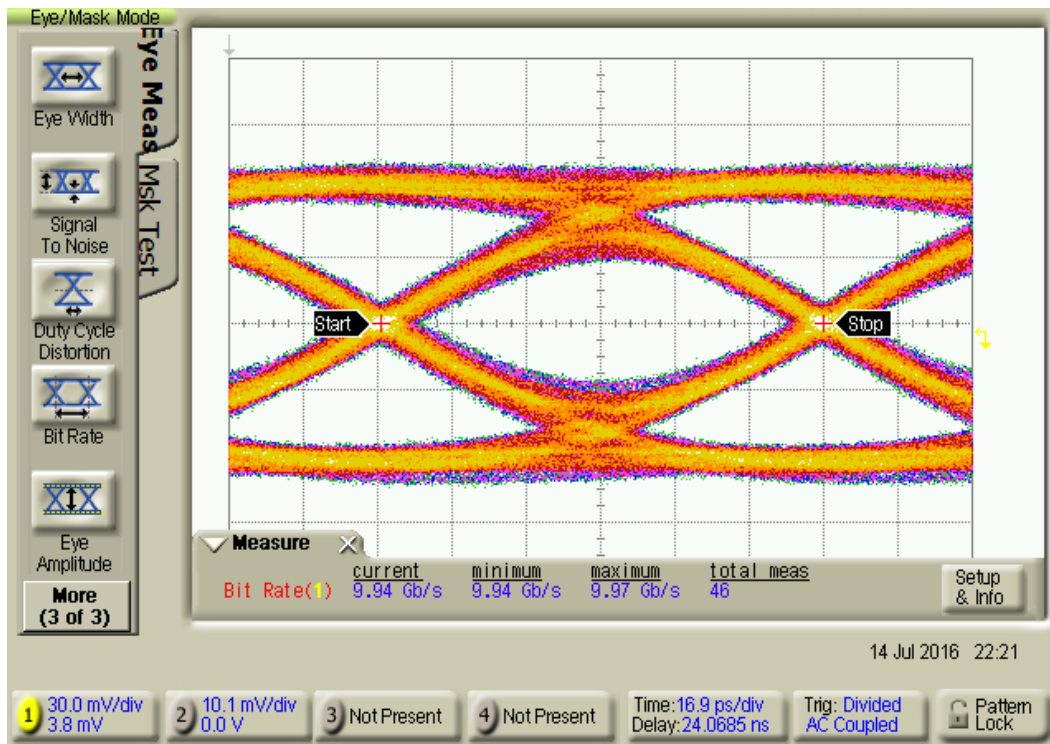


Figure 4.9: Measured 10 Gb/s output eye of the front end.

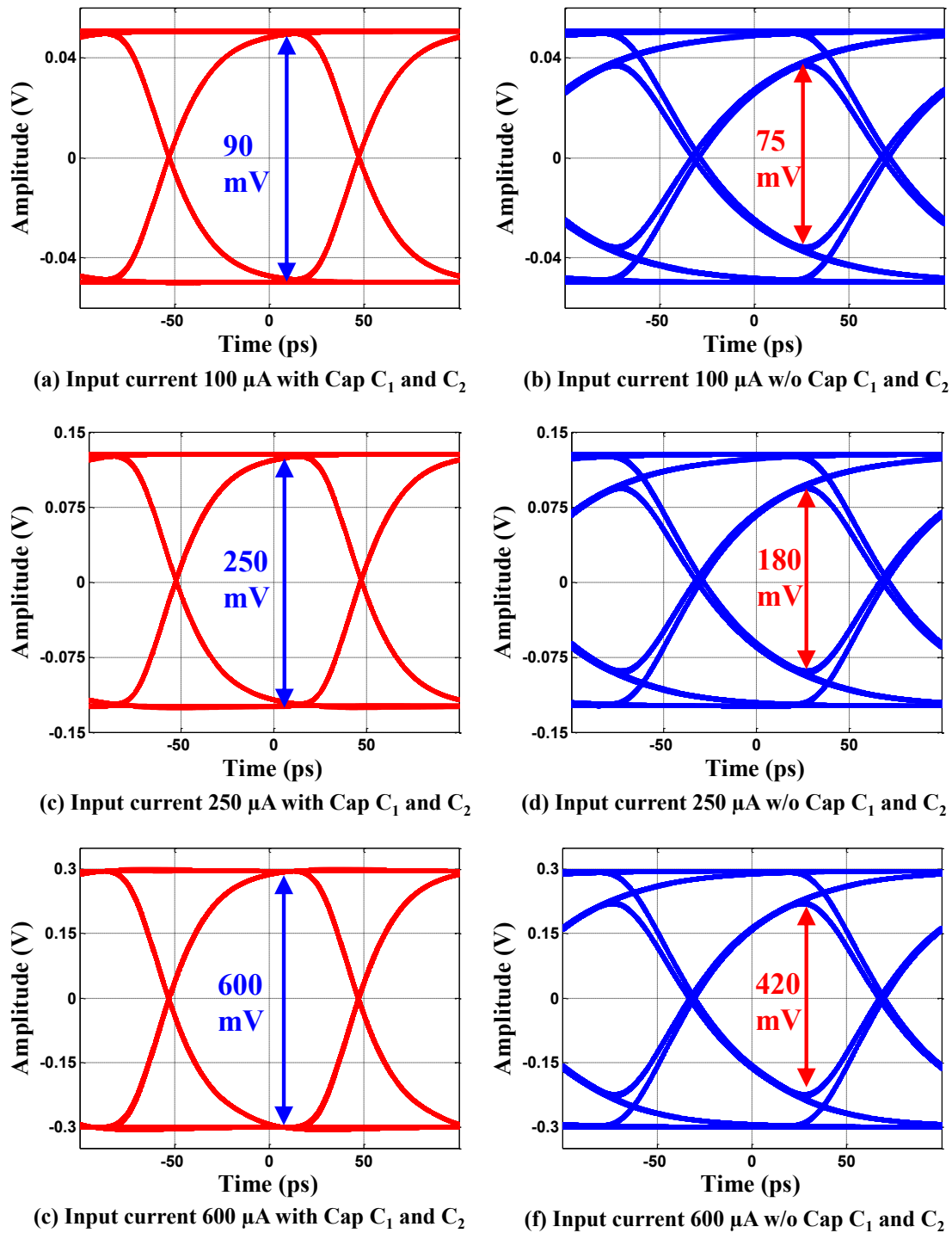


Figure 4.10: Simulated amplifier stage output eye with (left) and without (right)  $C_1$  &  $C_2$  for input currents of  $100 \mu\text{A}$ ,  $250 \mu\text{A}$  and  $600 \mu\text{A}$  (from top to bottom).

Benefit of the high pass frequency response is visible in the simulated eye diagram shown in Figure 4.10 for different input currents. Measured TIA output eye is consistent with the simulated results and indicates the extended bandwidth benefit of the front-end.

TIA has to sense small ( $\sim 60 \mu\text{A}$ ) current from the photodiode and convert it to voltage. If the noise component from this TIA front end is high, it will propagate through the whole front end which may result in incorrect operation of data and clock recovery. One of the design considerations of this TIA is to have the noise component as low as possible while getting the most amount of gain and boost out of it. The trade-off is that if we try to increase the boost in the DC recovery path, it produces higher noise. So, a nominal point was chosen where we can get a high boost and low noise. The noise sources in this TIA are all the transistors and resistors handling 7~10 Gb/s high-speed data. Noise from these sources will appear at both OUTP and OUTN of the TIA as shown in Figure 4.7. However, these two paths have different gains. The two path gains are derived in Eq. 4.6 and 4.7. For noise calculation, first, we will show all the noise accumulated at OUTP and OUTN. Then, this noise at OUTP and OUTN will be referred to input by dividing them using their respective path gain.

Noise current from any transistor is given by [24]

$$\overline{I_{n,M}^2} = 4KT\gamma g_m. \quad 4.8$$

Noise generated by  $M_{N1}$  sees two impedances in parallel; one from  $M_{N1}$  and the other from  $C_1$  and  $M_{N2}$ . The part of noise current that passes through  $M_{N1}$  appears at OUTP with gains

from the TIA common gate stage and the common source with source degeneration stage ( $A_{CS}$ ).

$$\text{Noise at OUP due to } M_{N1}, \overline{V_{n,OUTP,M_{N1}}^2} = 4KT\mathcal{G}_{m_1} \left( \frac{\frac{1}{g_{m_2}} + \frac{1}{sC_1}}{\frac{1}{g_{m_1}} + \frac{1}{g_{m_2}} + \frac{1}{sC_1}} \right)^2 R_1^2 A_{CS}^2 \cdot \quad 4.9$$

The part of noise current passing through DC path will appear at OUTN node.

$$\text{Noise at OUTN due to } M_{N1}, \overline{V_{n,OUTN,M_{N1}}^2} = 4KT\mathcal{G}_{m_1} \left( \frac{\frac{1}{g_{m_1}}}{\frac{1}{g_{m_1}} + \frac{1}{g_{m_2}} + \frac{1}{sC_1}} \right)^2 R_2^2 \cdot \quad 4.10$$

Noise generated by  $M_{N2}$  sees two impedances in parallel; one from  $M_{N2}$  and the other from  $C_1$  and  $M_{N1}$ . The noise appearing at OUP will pass through  $C_1$  and  $M_{N1}$ . For OUTN, this noise will come through  $M_{N2}$  and will see the gain of its path.

$$\text{Noise at OUP due to } M_{N2}, \overline{V_{n,OUTP,M_{N2}}^2} = 4KT\mathcal{G}_{m_2} \left( \frac{\frac{1}{g_{m_2}}}{\frac{1}{g_{m_1}} + \frac{1}{g_{m_2}} + \frac{1}{sC_1}} \right)^2 R_1^2 A_{CS}^2 \cdot \quad 4.11$$

$$\text{Noise at OUTN due to } M_{N2}, \overline{V_{n,OUTN,M_{N2}}^2} = 4KT\mathcal{G}_{m_2} \left( \frac{\frac{1}{g_{m_1}} + \frac{1}{sC_1}}{\frac{1}{g_{m_1}} + \frac{1}{g_{m_2}} + \frac{1}{sC_1}} \right)^2 R_2^2 \cdot \quad 4.12$$

The noise due to  $M_{N3}$  appears at OUP only. It sees the gain from the common source with degeneration stage.

$$\text{Noise at OUP due to } M_{N3}, \overline{V_{n,OUTP,M_{N3}}^2} = 4KT\mathcal{G}_{m_3} R_3^2 A_{CS}^2 \cdot \quad 4.13$$

The noise voltage and current from any resistor, R, are given by [24]

$$\overline{V_{n,R}^2} = 4KTR \quad 4.14$$

$$\overline{I_{n,R}^2} = \frac{4KT}{R}. \quad 4.15$$

Now, the noise from  $R_1$  of TIA appears both at OUTP and OUTN. The noise voltage generated from  $R_1$  sees the gain from the common source with degeneration stage and shows up at OUTP.

$$\text{Noise at OUTP due to } R_1, \overline{V_{n,OUTP,R_1}^2} = 4KTR_1 A_{CS}^2. \quad 4.16$$

For noise from  $R_1$  to appear at OUTN, it has to go through the impedances introduced by  $M_{N1}$ ,  $C_1$ , and  $M_{N2}$ . The noise current from  $R_1$  will generate a voltage across  $M_{N2}$  which will see a gain from  $M_{N2}$  and appear at OUTN node.

$$\text{Noise at OUTN due to } R_1, \overline{V_{n,OUTN,R_1}^2} = \frac{4KT}{R_1} \left( \frac{\frac{1}{g_{m_1}} + \frac{1}{sC_1}}{\frac{1}{g_{m_1}} + \frac{1}{g_{m_2}} + \frac{1}{sC_1}} \right)^2 g_{m_2}^2 R_2^2. \quad 4.17$$

Noise from  $R_2$  only appears across OUTN.

$$\text{Noise at OUTN due to } R_2, \overline{V_{n,OUTN,R_2}^2} = 4KTR_2. \quad 4.18$$

Noise from  $R_3$  and  $R_4$  only appear in OUTP node.

$$\text{Noise at OUTP due to } R_3, \overline{V_{n,OUTP,R_3}^2} = 4KTR_3. \quad 4.19$$

$$\text{Noise at OUTP due to } R_4, \overline{V_{n,OUTP,R_4}^2} = \frac{4KT}{R_4} \left( \frac{R_4 \parallel \frac{1}{sC_2}}{R_4 \parallel \frac{1}{sC_2} + \frac{1}{g_{m_3}}} \right)^2 R_3^2. \quad 4.20$$



All the noise from different sources appearing at OUTP and OUTN will be summed up to get total noise at respective nodes.

$$\begin{aligned}
\text{Noise at OUTP, } \overline{V_{n,OUTP}^2} &= 4KT\gamma g_{m_1} \left( \frac{\frac{1}{g_{m_2}} + \frac{1}{sC_1}}{\frac{1}{g_{m_1}} + \frac{1}{g_{m_2}} + \frac{1}{sC_1}} \right)^2 R_1^2 A_{CS}^2 + 4KTR_1 A_{CS}^2 + 4KTR_3 \\
&+ 4KT\gamma g_{m_2} \left( \frac{\frac{1}{g_{m_2}}}{\frac{1}{g_{m_1}} + \frac{1}{g_{m_2}} + \frac{1}{sC_1}} \right)^2 R_1^2 A_{CS}^2 + \frac{4KT}{R_4} \left( \frac{R_4 \parallel \frac{1}{sC_2}}{R_4 \parallel \frac{1}{sC_2} + \frac{1}{g_{m_3}}} \right)^2 R_3^2 + 4KT\gamma g_{m_3} R_3^2 A_{CS}^2.
\end{aligned} \tag{4.21}$$

$$\begin{aligned}
\text{Noise at OUTN, } \overline{V_{n,OUTN}^2} &= 4KT\gamma g_{m_1} \left( \frac{\frac{1}{g_{m_1}}}{\frac{1}{g_{m_1}} + \frac{1}{g_{m_2}} + \frac{1}{sC_1}} \right)^2 R_2^2 + 4KTR_2 \\
&+ 4KT\gamma g_{m_2} \left( \frac{\frac{1}{g_{m_1}} + \frac{1}{sC_1}}{\frac{1}{g_{m_1}} + \frac{1}{g_{m_2}} + \frac{1}{sC_1}} \right)^2 R_2^2 + \frac{4KT}{R_1} \left( \frac{\frac{1}{g_{m_1}} + \frac{1}{sC_1}}{\frac{1}{g_{m_1}} + \frac{1}{g_{m_2}} + \frac{1}{sC_1}} \right)^2 g_{m_2}^2 R_2^2.
\end{aligned} \tag{4.22}$$

To get the input referred noise we need to divide noise at OUTP and OUTN with their path gains, which we get from Eq. 4.6 and 4.7.

$$\text{Input Referred Noise, } \overline{I_{n,IN}^2} = \frac{\overline{V_{n,OUTP}^2}}{A_F^2} + \frac{\overline{V_{n,OUTN}^2}}{A_{HP}^2}. \tag{4.23}$$

Noise estimated by the above equation correlates well with the simulated noise over the frequency band as shown in Figure 4.11. Consuming only 4.3 mW, the TIA achieves a sensitivity of -13.98dBm assuming photodetector responsivity of 0.5A/W at BER of  $10^{-12}$ .

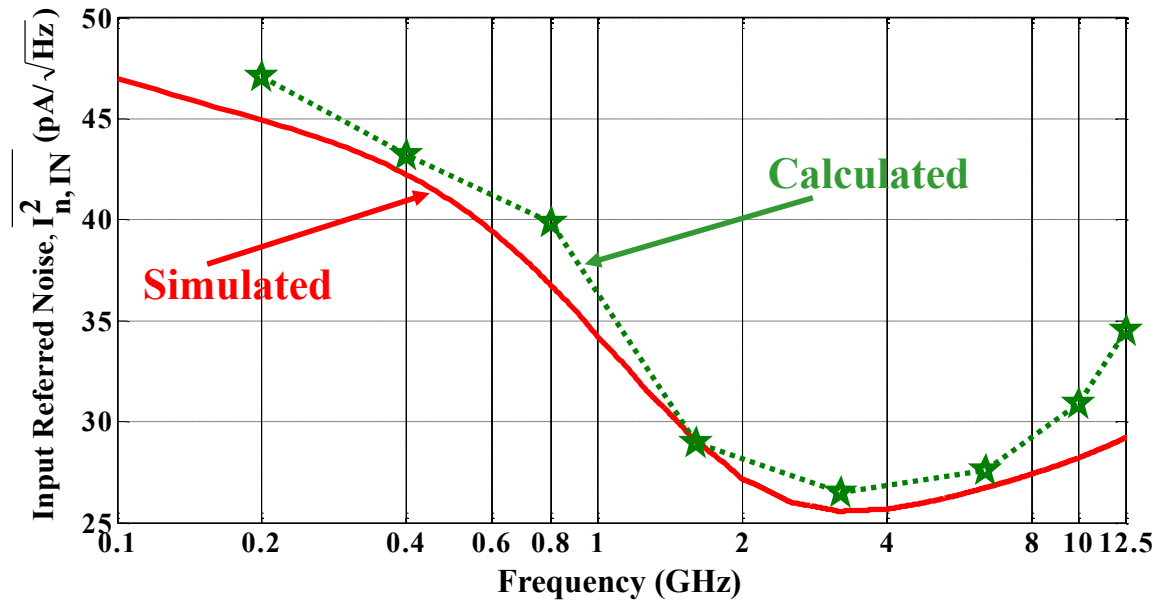


Figure 4.11: Simulated and calculated input referred noise of TIA.

### 4.3.2. DC Recovery

DC recovery loop works during the preamble period (101010.... pattern). The differential voltage from amplifier chain is sampled using the sample and hold (SH) circuit. Two consecutive samples,  $S(n)$  and  $S(n-T)$  from two nearby time-interleaved paths, are compared to get  $S(n)+S(n-T)$ , which is then fed to SAR logic block (Figure 4.6). The SAR block is clocked using  $1/8^{\text{th}}$  of data rate clock (C8 clock) of the receiver to have the DC well settled before next decision cycle. The overall operation takes six cycles to complete. The 1st cycle is used to reset the SAR logic. The other five cycles are used to update the DAC. Figure 4.12 illustrates the implemented DC recovery technique. In Figure 4.12, AMP\_OUTP and AMP\_OUTN are the outputs of the amplifier stage. At the beginning of each burst, in worst case scenario the DC offset may even fully saturate the amplifier

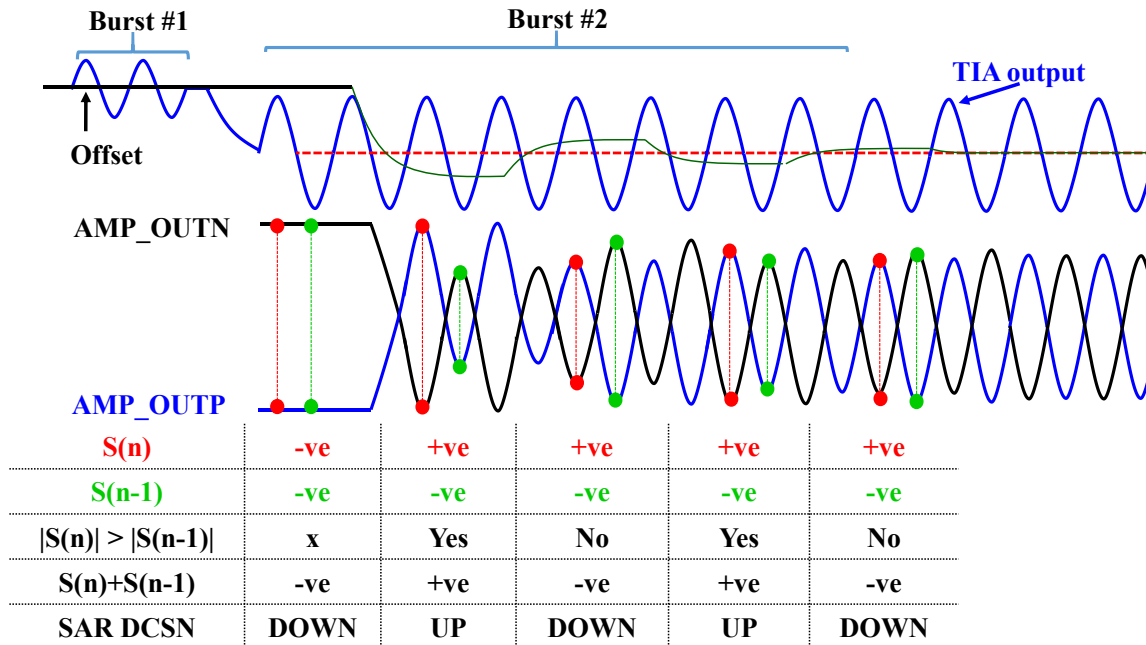


Figure 4.12: Proposed DC recovery technique.

output. In such scenario, both sampled values appear to be  $-ve$ . Therefore, DC offset continues to reduce until output starts to toggle. During the preamble period, the signal goes through a transition in each cycle. Therefore, the sampler that is sampling '1' will have  $+ve$  value and the other one that is sampling '0' should have  $-ve$  value. Since we are not using any limiting amplifier, their amplitude will not saturate. When these two samples are added, their summation indicates the DC offset in the input signal. To design a binary search algorithm, we only consider the polarity of  $S(n)+S(n-T)$ . Each time SAR logic updates the DAC depending on this output. As the algorithm progresses, their amplitudes will vary until the SAR DAC output DC voltage matches the incoming data dependent offset. As the DC recovery information is stored digitally during the preamble period, the system does not face any baseline wandering issue during CID pattern. Figure 4.13 shows

a measured scope shot of DC recovery operation taking only 4.8 ns for incoming 10 Gb/s data.

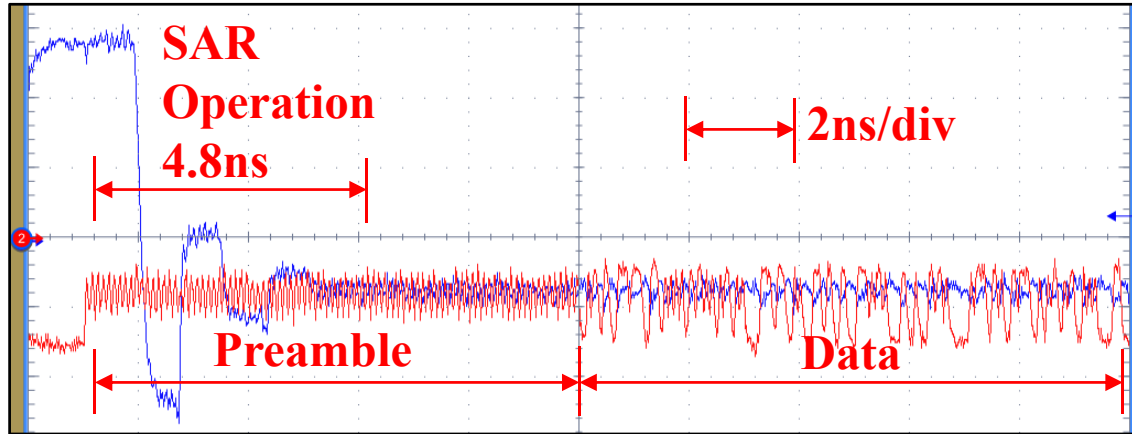


Figure 4.13: Measured scope shot of DC recovery operation within 4.8 ns.

### 4.3.3. Clock Recovery

In the proposed receiver architecture, DC and timing recovery works concurrently. During DC recovery, as there is no LPF, DC settles faster as the SAR updates the DAC at every C8 clock. As the DC is moving through different SAR logic, if it settles within the signal range of the TIA output, injection pulses begin to appear (Figure 4.14). Non 50% duty cycle amplifier output during DC recovery gives pulses that are not 1-unit interval (UI) apart. For the injection-locked oscillator (ILO) to lock at the proper frequency, the separation between pulses has to be an integer multiple of UI. Although the distance between rising and falling edge pulses depend on DC offset, the distance from one rising edge to next rising edge is always 2UI. Therefore, rather than using both rising/falling edge, only rising edge pulses are used to facilitate concurrent operation. When it comes to the regular data pattern, this distance will always be  $N \cdot UI$  where  $N \geq 2$ .

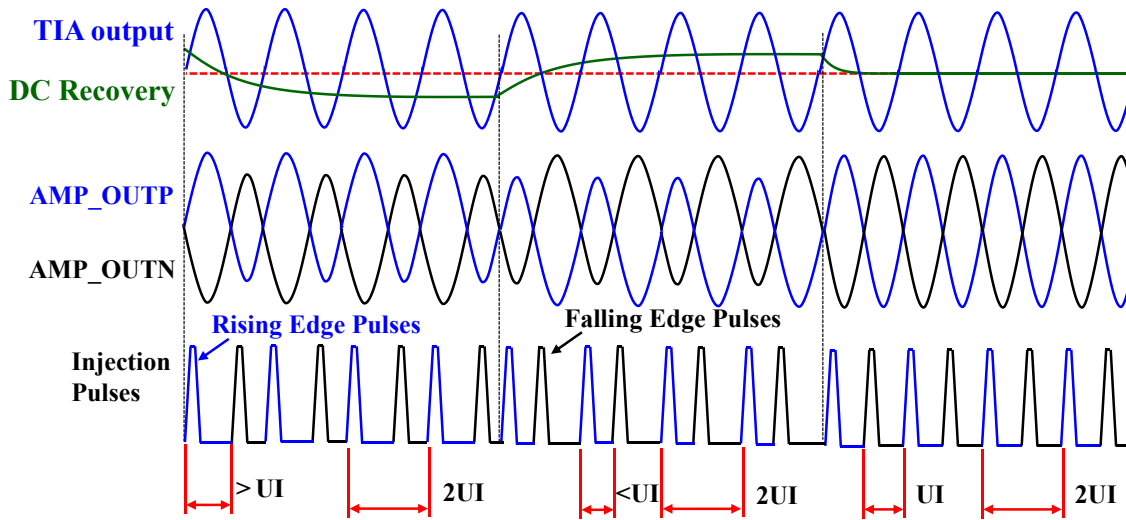


Figure 4.14: DC recovery without LPF and use of rising edge pulses allowing concurrent operation of DC and timing recovery.

A ring oscillator is chosen for this work as it provides a wide tuning range with a compact area requirement. The quadrate operation of the oscillator helps to achieve higher data rate with better power efficiency in comparison to half rate or full rate clocking. The oscillator has to operate in 1.75-2.5 GHz range as the data rate is between 7-10 Gb/s. The ring oscillator has four stages as shown in Figure 4.15. Stage I, which gives  $\Phi_0$  and  $\Phi_{180}$ , is chosen for data pulse injection. The injected pulse corrects the zero crossings of  $\Phi_0$  and  $\Phi_{180}$ . As injection is happening in only one stage of the quadrate oscillator, only the rising edge pulses can do the correction which are separated by  $4UI$ . The  $4UI$  separated pulses also have to arrive at the time of zero crossing of  $\Phi_0$ . In this work, a pulse window is created by having an AND between  $\Phi_{225}$  and  $\Phi_{315}$ . However, any harmonic component of the pulses generated from data can still go through and provide the phase correction if it falls

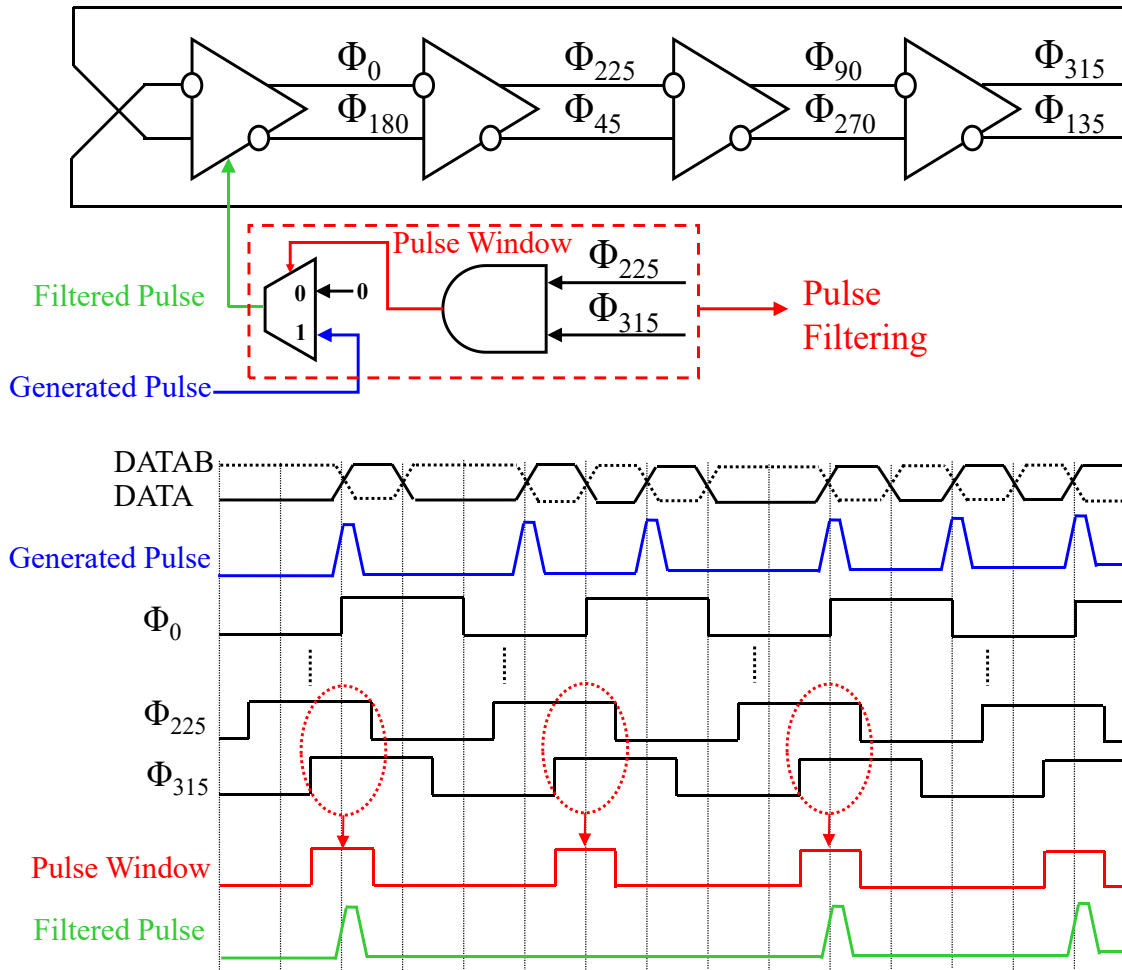


Figure 4.15: Ring oscillator with pulse filtering and its timing diagram.

within the pulse window. Figure 4.15 illustrates the whole pulse filtering function of the oscillator. With this pulse filtering in place during the preamble period each window gets a pulse transferred to the oscillator which in turn makes it lock instantaneously. During DC recovery operation as the DC goes up and down the lock point of the oscillator may shift, but it always remains locked during that time enabling concurrent operation.

### 4.3.4. Timing Skew Correction

There are delays associated with the CML-to-CMOS and the injection pulse generator that go on to correct the phase of the oscillator. The corrected data phases of the injection locked oscillator go through buffers and pulse generators to do the sampling of the incoming data. In this way, the timing margin for data samplers becomes a function of path delay (Figure 4.16). Ideally, this path delay should be 0.5 UI. However, in different corners, this delay varies which reduces the timing and voltage margin of the data recovery path. To get rid of this issue, a slope detection technique is applied. The slope detection logic only works during the no data transition period as the ILO already takes care of transition edges. If the data is sampled at the middle of the eye (Figure 4.17(a)), the slope between two consecutive samples should ideally be zero. This is true for both 11 and 00 data sequences. However,

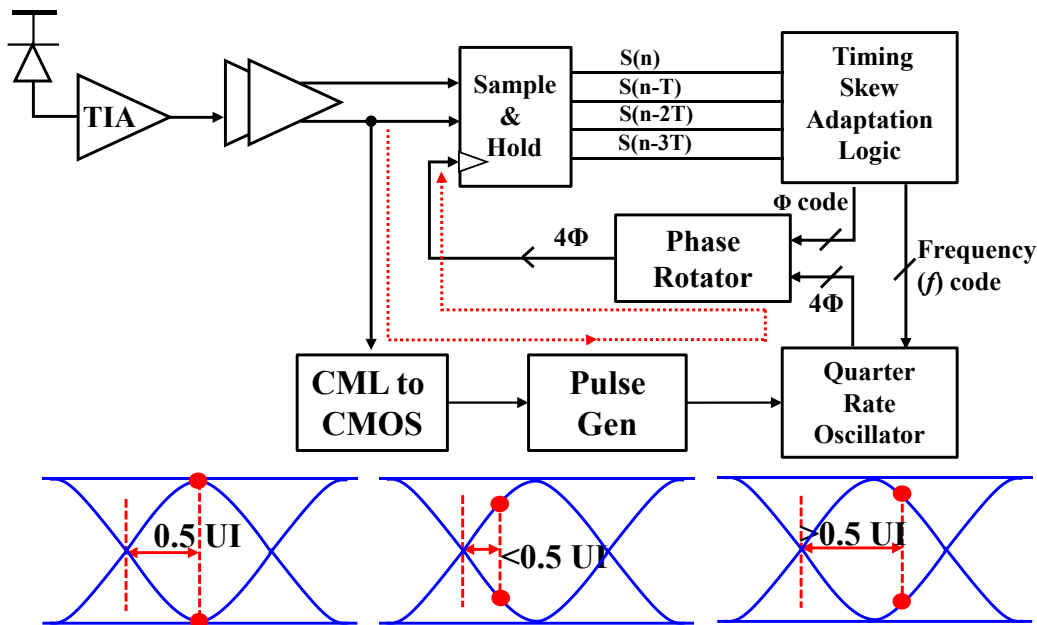


Figure 4.16: Timing Skew compensation.

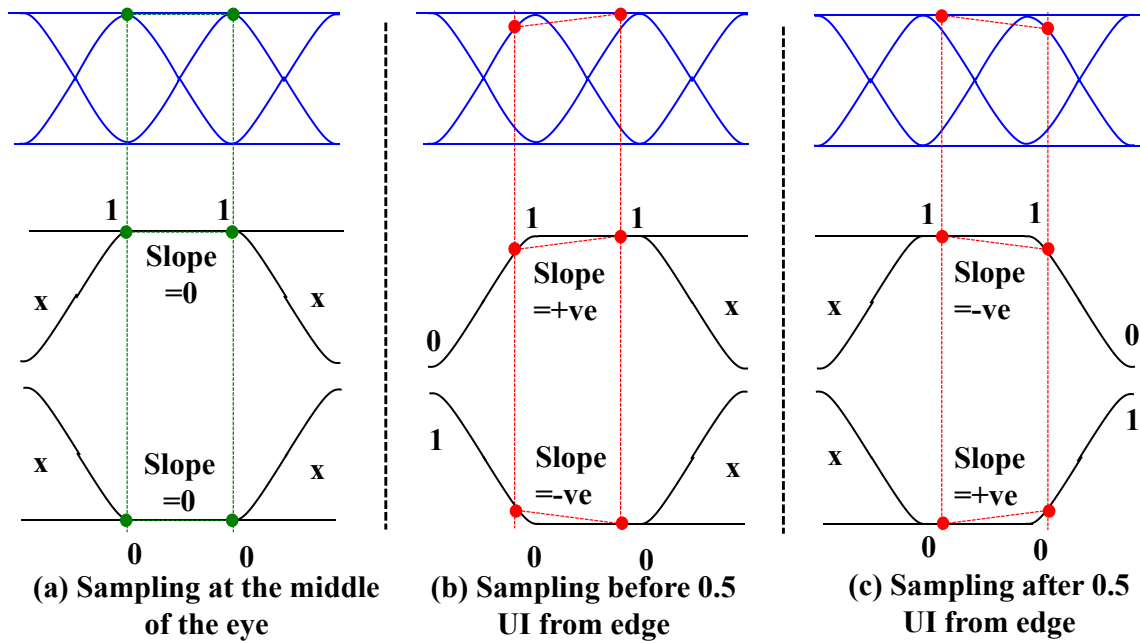


Figure 4.17: Slope Detection.

if the non-transition data is sampled before/after 0.5 UI from the edge, there will be a +ve/-ve slope. As illustrated in Figure 4.17(b), when the data sequence is 011x, if the slope between two samples of consecutive 1s is +ve, the sampling phase should be moved to the right to have the phase in the middle of the eye. However, for 100x data sequence it has to move right if the slope between two samples of consecutive 0s is -ve. Figure 4.17(c) shows the situations where the data sampling phase has to move left. The data sequences to consider here are x110 and x001. This adaptation logic runs after the preamble period during regular data pattern and updates the phase rotator to have correct sampling phase with the highest timing margin. To implement this slope detection, we can reuse S/H, comparator and SAR algorithm used for DC recovery. The comparison between two consecutive samples,  $S(n)$  and  $S(n-T)$  during the non-transition data period will give the slope of the incoming signal. The phase rotator has 5-bit control allowing 32 phase steps



between two data phases. The slope detection algorithm output goes to a majority voter and C16 (1/16th of data rate) is used to take data out of the majority voter. After majority voting, the phase rotator control bits are updated using the successive approximation algorithm. This logic runs once during each burst. There is a frequency ( $f$ ) code that controls the frequency of the oscillator. This frequency code ensures that the oscillator is at the correct frequency. As the frequency is locked, the oscillator can't drift away during the CID data pattern.

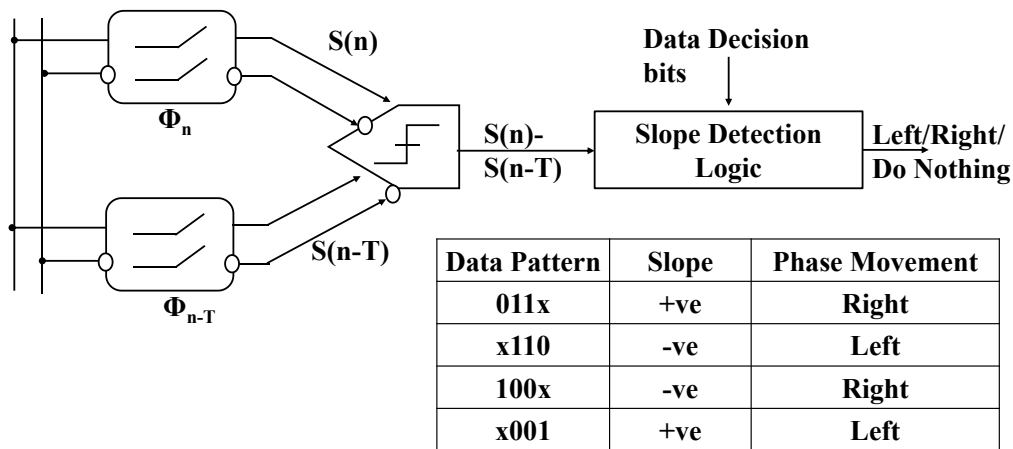


Figure 4.18: Slope detection logic.

## 4.4. Implementation and Measurement Results

The implemented die photo is shown in Figure 4.19. The quadrature receiver takes only 465  $\mu\text{m}$  x 265  $\mu\text{m}$  area in 0.13  $\mu\text{m}$  technology. Figure 4.20 shows the test setup with all the necessary instruments. The receiver recovers DC offset in only 4.8 ns compared to current state-of-art architecture taking 12.5 ns. The burst mode clock recovery takes 1 ns more (i.e. 5.8 ns) during the preamble period (Figure 4.21). The receiver works at 10 Gb/s consuming only 41.6 mW out of which 5.1 mW goes for the initial DC recovery, and 3.8 mW goes for

the timing skew correction. The power efficiency of the receiver during runtime is only 3.27 pJ/bit. During low data rates, the consumed power goes down, but power efficiency is reduced. The recovered clock has around 10 ps measured peak to peak jitter (Figure 4.23). Figure 4.24 shows the phase noise plot for 7 Gb/s operation giving only 2.3 ps rms jitter when jitter is integrated from 1 KHz to 1 GHz.

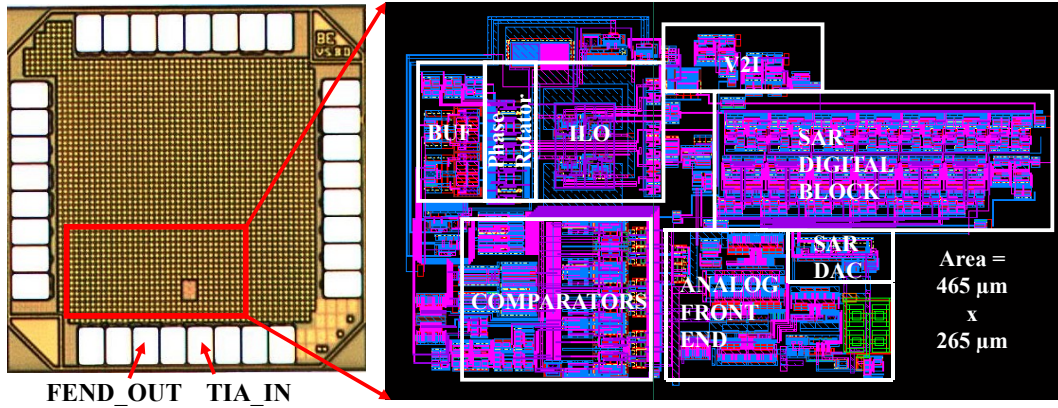


Figure 4.19: Implemented die photo in 0.13  $\mu\text{m}$ .

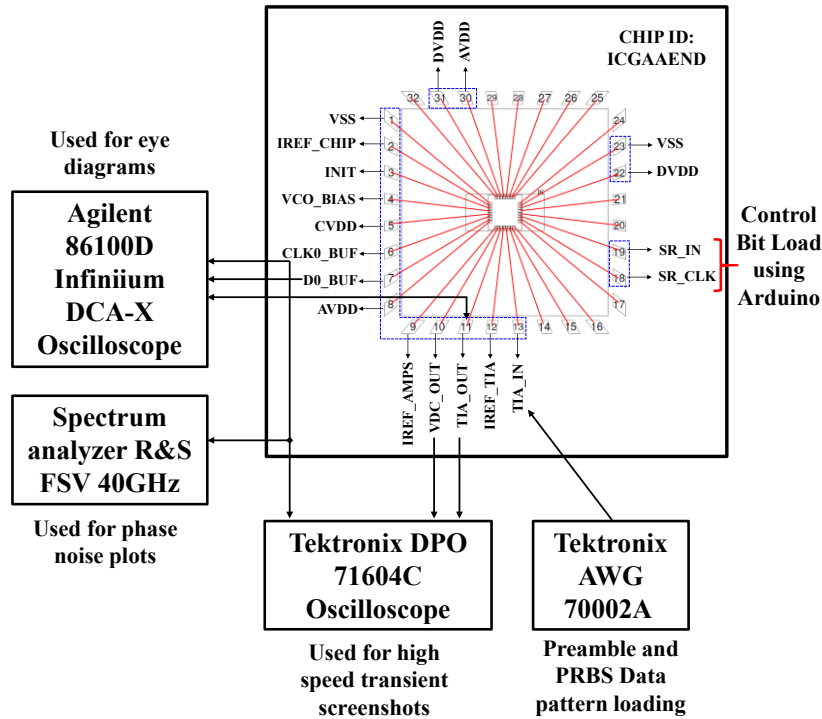


Figure 4.20: Pin diagram and test setup of the prototype.

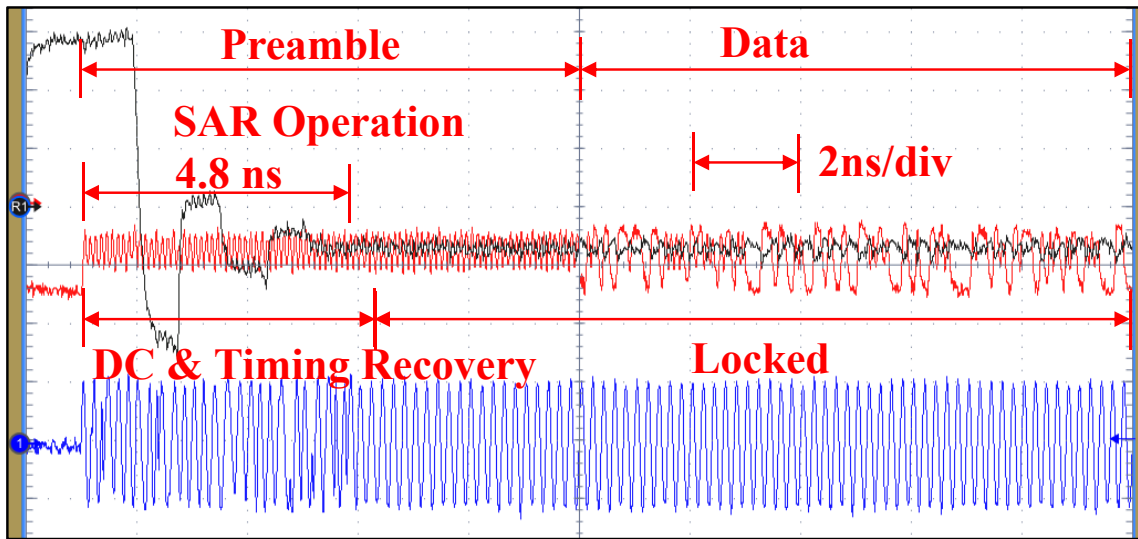


Figure 4.21: Measured scope shot of DC and timing recovery with preamble and data pattern.

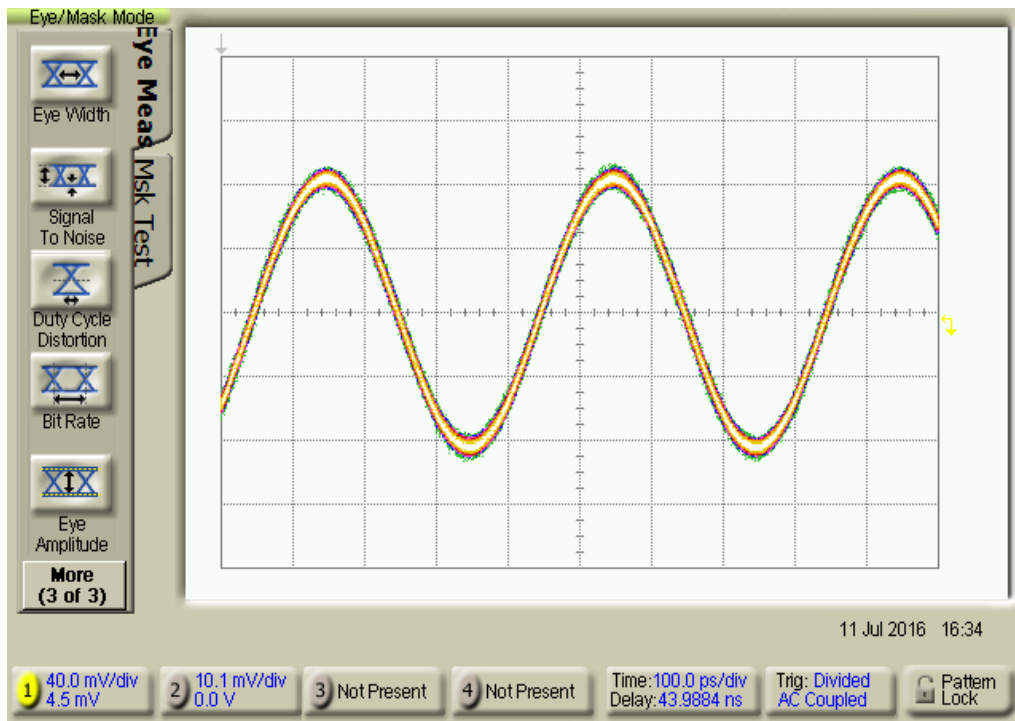


Figure 4.22: Recovered clock eye.

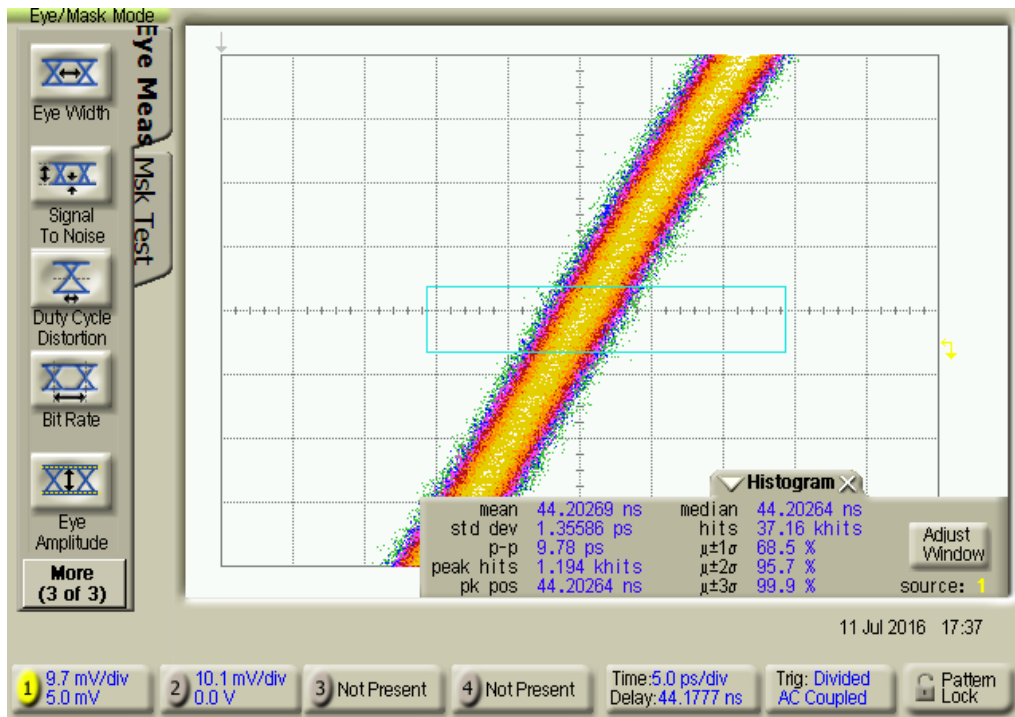


Figure 4.23: Recovered clock histogram.

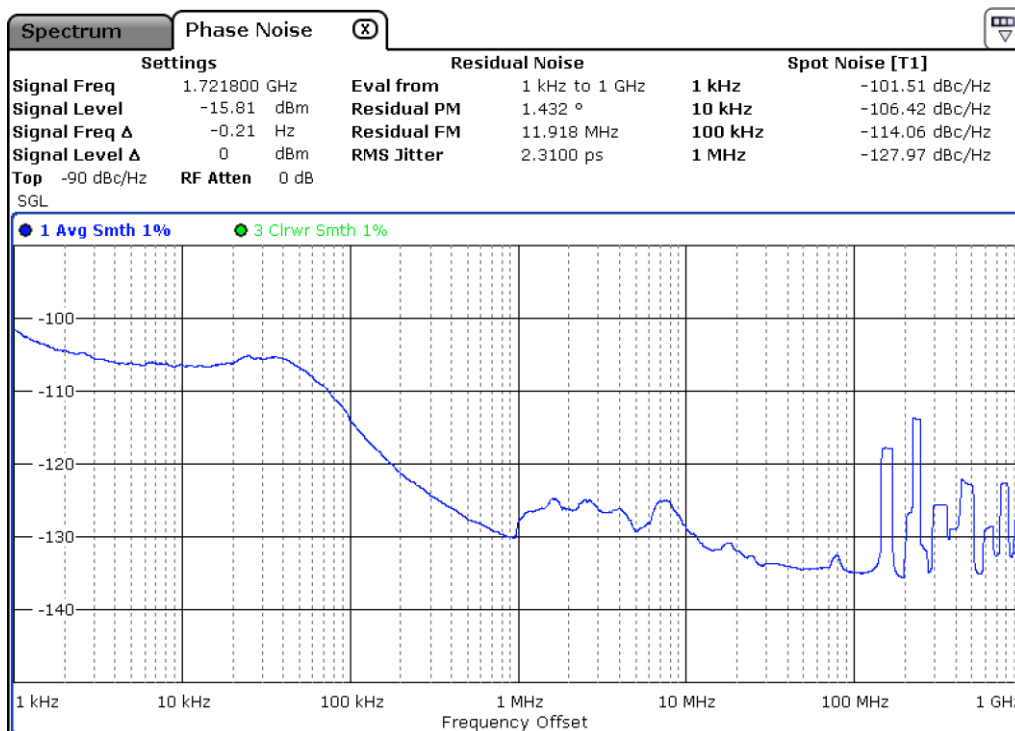


Figure 4.24: Phase noise plot of the recovered clock.

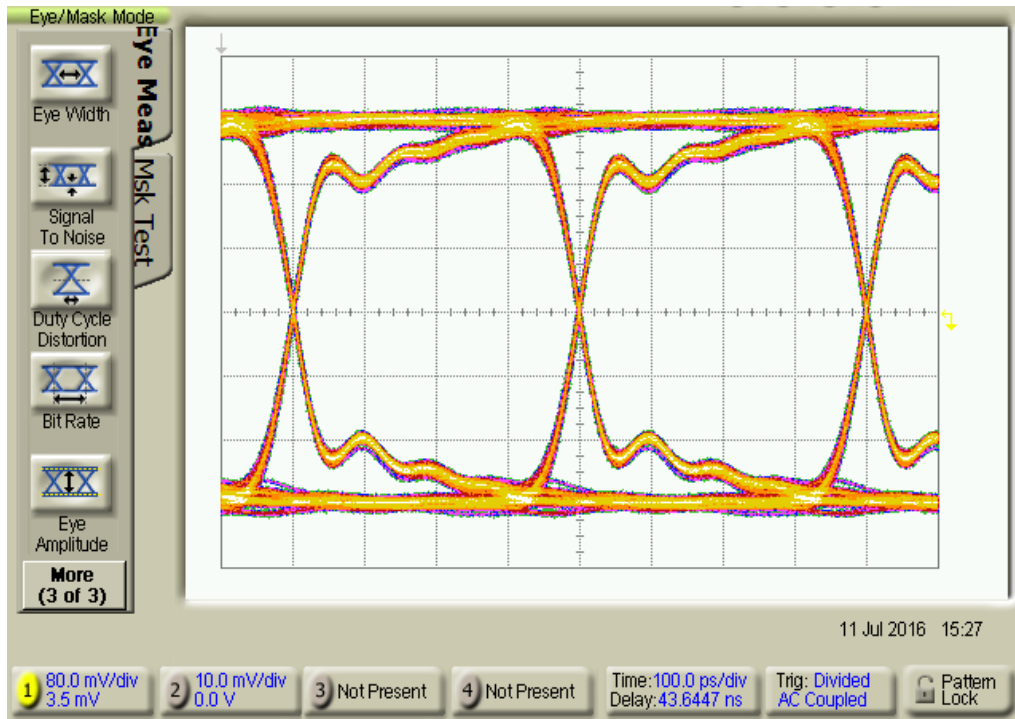


Figure 4.25: On-chip PRBS checked recovered 2.5 Gb/s channel data eye.

## 4.5. Comparison with State-of-Art

The proposed optical receiver is compared to existing works in Table 4.1 and Table 4.2. The existing 10 Gb/s solutions require inductive peaking to meet the gain-bandwidth product requirement. In the proposed TIA, we have achieved comparable performance without any inductive peaking. The entire receiver takes only  $465 \mu\text{m} \times 265 \mu\text{m}$  area in  $0.13 \mu\text{m}$  technology with comparable performance when compared to state-of-art works. While most of the works concentrate on either timing or DC recovery, here we have addressed both challenges with excellent power efficiency. Prior work [2] shows that, for dynamically reconfigurable optical networks, a receiver lock time of 100 ns is required. Receiver lock time of 50 ns is highly desirable, which contributes to overall network performance. In this work, we have a lock time of 5.8 ns only, which improves overall state-of-art by 6.5X.

Table 4.1: Performance Comparison Summary of the TIA.

	RFIC '09 [32]	ASSCC '07 [33]	JSSC '15 [2]	This Work
<b>Technology</b>	0.13 $\mu\text{m}$ CMOS	0.18 $\mu\text{m}$ CMOS	32 nm CMOS	0.13 $\mu\text{m}$ CMOS
<b>Gain (dB <math>\Omega</math>)</b>	57	59	46.4	60.9
<b>Bandwidth (GHz)</b>	10	10	18.4	11.7
<b>Sensitivity (dBm)</b>	N/A	N/A	-14.4	-13.98
<b>Power (mW)</b>	1.8	18	2.7	4.3
<b>FoM (GHz <math>\Omega/\text{mW}</math>)</b>	3933	495	1431	3018
<b>Inductive Peaking</b>	Yes	Yes	No	No
<b>No. of Stages</b>	1	2	1	2

Table 4.2: Performance Comparison Summary of the Receiver.

	ISSCC '12 [28]	ISSCC '11 [12]	JSSC '15 [2]	This Work
<b>Data Rate (Gb/s)</b>	10	1-6	25	7 – 10
<b>Technology</b>	0.13 $\mu\text{m}$ SiGe BiCMOS	65 nm CMOS	32 nm CMOS	0.13 $\mu\text{m}$ CMOS
<b>Area (mm<sup>2</sup>)</b>	1.6202 (TIA+LA)	0.0175	0.06	0.123225
<b>DC Recovery + Timing Recovery Operation</b>	DC only	Timing Only	Cascaded	Concurrent
<b>DC Recovery Technique</b>	Feedback type AOC circuit with switchable loop BW	-	LPF with Calibration logic	Successive Approximation Algorithm
<b>DC Recovery Cycle/Time (Data_rate/8 Clock Cycle)</b>	75 ns	-	39 Cycle/ 12.5 ns	6 Cycle/ 4.8 ns
<b>Clock Recovery Technique</b>	-	Phase Interpolator	Phase Interpolator & Bang Bang PD	Quarter Rate ILO
<b>Clock Recovery Time (ns)</b>	-	<0.16-1	18.5	<1
<b>Total Lock Time (ns)</b>	-	-	31	5.8
<b>Power Consumption (mW)</b>	630 (TIA+LA)	22 (CDR only)	109	TIA+Amps $\rightarrow$ 12.1 Timing Recovery $\rightarrow$ 11.6 Data Recovery $\rightarrow$ 5.4 DC Recovery $\rightarrow$ 8.7 Skew Adaptation $\rightarrow$ 3.8
<b>Power Efficiency (pJ/bit)</b>	63 (TIA+LA)	3.67 (CDR only)	4.4	3.27 (Runtime power-TIA+Amps+CDR)

# Chapter 5.

## Concluding Remarks

The main concentration of this thesis has been to develop the concept and architecture of low power energy efficient digital receiver. This dissertation has featured a total of three receivers, out of which two are for wireline application, and the other one is an optical receiver. The performance summary of the implemented receivers is listed in Table 5.1.

Table 5.1: Performance summary of the implemented receivers.

	<b>Receiver1</b>	<b>Receiver2</b>	<b>Receiver3</b>
<b>Application</b>	Wireline	Wireline	Optical
<b>Technology</b>	TSMC 65nm	TSMC 65nm	IBM 0.13 $\mu$ m
<b>Supply Voltage (V)</b>	1.2	1.2	1.2
<b>Area (mm<sup>2</sup>)</b>	0.23	0.168	0.123225
<b>Area per channel</b>	240 $\mu$ m X 240 $\mu$ m	300 $\mu$ m X 140 $\mu$ m	--
<b>Data Rate (Gb/s)</b>	10	16	7-10
<b>Architecture</b>	4X Sequence DFE	4X Sequence DFE and Trace-back	4X Time-interleaved
<b>Clock Recovery</b>	Edge and Data Sampled	--	Injection locked
<b>DC Recovery</b>	N/A	N/A	SAR Algorithm



<b>Consumed Power (mW)</b>	35@27 dB	71.8 @ 35 dB 55 @ 25 dB	32.7 (runtime)
<b>Power Efficiency (pJ/bit)</b>	3.5@ 27 dB	4.4875 @ 35 dB 3.4375@25 dB	3.27
<b>FoM (pJ/bit/dB)</b>	0.13	0.1282@ 35 dB 0.1375@ 25 dB	--
<b>Improvement from current state-of-art</b>	2.5X in Power Efficiency [11]	1.94X in Power Efficiency [11]	1.35X in Power Efficiency [2]
	2.65X in FoM [11]	2.68X in FoM [11]	

We have proposed a low power maximum likelihood sequence equalization technique without ADC-DSP for wireline application. The implemented prototype in TSMC 65nm receiver has worked at 10 Gb/s achieving a BER of  $10^{-12}$ . The receiver has improved the state-of-art by consuming only 3.5 pJ/bit while compensating 27 dB channel loss, which gives it a figure of merit (FoM) of 0.13 pJ/bit/dB. In comparison with the current state-of-art ([11]), FoM has improved by 2.65X as shown in Table 5.1. The concept, architecture and measurement results of this work has been published in *Symposium on VLSI Circuits*:

“A 35 mW 10 Gb/s ADC-DSP less direct digital sequence detector and equalizer in 65nm CMOS,” A. D. Hossain, Aurangozeb, M. Mohammad and M. Hossain, 2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits), Honolulu, HI, 2016, pp. 1-2.”

The equalization technique applied for the 10 Gb/s receiver in Chapter 2 has been improved in Chapter 3 to have a better noise immunity. The receiver prototype with 1-bit data trace-back has worked at 16 Gb/s demonstrating BER of  $10^{-10}$  for DFE operation only and BER of  $10^{-12}$  when DFE and trace-back worked together. For the data traceback, two additional

comparators and additional digital logics have been added in the architecture which results in higher power consumption of about 71.8 mW. Therefore, the power efficiency turns out to be 4.4875 pJ/bit; higher than previous architecture but with improved voltage margin. However, as it could compensate higher loss of up to 35 dB, FoM has been 0.1282 pJ/bit/dB which was similar to the first receiver. Traditionally error correction capabilities have been introduced through encoders and decoders such as FEC (forward error correction). These encoders and decoders not only add overhead, but also add latency to the system that in turn negatively impacts compute performance. Traditional transceivers by themselves did not have error corrections capability. To the best of our knowledge, this has been for the first time that DFE error correction capability has been introduced within SerDes transceivers and this has opened up the potential for very low overhead and low latency error correction.

The optical receiver described in Chapter 4 has concentrated both on power and latency for burst mode application. The receiver implemented in 0.13  $\mu\text{m}$  technology has demonstrated a lock time of less than 10 ns while consuming only 32.7 mW during runtime. The recovered clock for the receiver has had around 10 ps measured peak to peak jitter. The paper containing overall concept and measurement results has been submitted for the peer review journal of *IEEE Transactions on Circuits and Systems I: Regular Papers*.

“Burst Mode Optical Receiver with 10ns Lock Time Based on Concurrent DC Offset and Timing Recovery Technique,” A. D. Hossain, Aurangozeb and M. Hossain, *IEEE Transactions on Circuits and Systems I*.”

## 5.1. Future Works

The equalization technique developed in this dissertation can compensate channel loss up to 35 dB while employing highly digital architecture. Digital designs are inherently scalable and portable to deep sub-micron CMOS processes. This technique can be implemented in smaller technology nodes such as 28nm FD-SOI or 14nm FinFET to achieve higher data rate and better power efficiency. With the increased use of pulse-amplitude modulation (PAM) in receiver design, this concept of equalization can be implemented for PAM-4/PAM-8 data streams where SNR is lower than NRZ streams.

The optical receiver architecture provides comparable performance to state-of-art, although it is implemented in 0.13  $\mu\text{m}$ . This portable architecture is attractive for higher data rates in scaled technology.

# Bibliography

- [1] T. Toifl *et al.*, “A 2.6 mW/Gbps 12.5 Gbps RX With 8-Tap Switched-Capacitor DFE in 32 nm CMOS,” *IEEE J. Solid-State Circuits*, vol. 47, no. 4, pp. 897–910, Apr. 2012.
- [2] A. Rylyakov *et al.*, “A 25 Gb/s Burst-Mode Receiver for Low Latency Photonic Switch Networks,” *IEEE J. Solid-State Circuits*, vol. 50, no. 12, pp. 3120–3132, Dec. 2015.
- [3] B. Abiri, A. Sheikholeslami, H. Tamura, and M. Kibune, “A 5Gb/s adaptive DFE for 2x blind ADC-based CDR in 65nm CMOS,” in *2011 IEEE International Solid-State Circuits Conference*, 2011, pp. 436–438.
- [4] E. H. Chen, R. Yousry, and C. K. K. Yang, “Power Optimized ADC-Based Serial Link Receiver,” *IEEE J. Solid-State Circuits*, vol. 47, no. 4, pp. 938–951, Apr. 2012.
- [5] C. Ting, J. Liang, A. Sheikholeslami, M. Kibune, and H. Tamura, “A blind baud-rate ADC-based CDR,” in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2013, pp. 122–123.
- [6] A. Varzaghani *et al.*, “A 10.3-GS/s, 6-Bit Flash ADC for 10G Ethernet Applications,” *IEEE J. Solid-State Circuits*, vol. 48, no. 12, pp. 3038–3048, Dec. 2013.
- [7] E. Z. Tabasy, A. Shafik, K. Lee, S. Hoyos, and S. Palermo, “A 6 bit 10 GS/s TI-SAR ADC With Low-Overhead Embedded FFE/DFE Equalization for Wireline Receiver Applications,” *IEEE J. Solid-State Circuits*, vol. 49, no. 11, pp. 2560–2574, Nov. 2014.

- [8] B. Zhang *et al.*, “3.1 A 28Gb/s multi-standard serial-link transceiver for backplane applications in 28nm CMOS,” in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, 2015, pp. 1–3.
- [9] D. Cui *et al.*, “3.2 A 320mW 32Gb/s 8b ADC-based PAM-4 analog front-end with programmable gain control and analog peaking in 28nm CMOS,” in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, 2016, pp. 58–59.
- [10] S. Rylov *et al.*, “3.1 A 25Gb/s ADC-based serial line receiver in 32nm CMOS SOI,” in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, 2016, pp. 56–57.
- [11] A. Shafik, E. Z. Tabasy, S. Cai, K. Lee, S. Hoyos, and S. Palermo, “3.6 A 10Gb/s hybrid ADC-based receiver with embedded 3-tap analog FFE and dynamically-enabled digital equalization in 65nm CMOS,” in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, 2015, pp. 1–3.
- [12] A. D. Hossain, Aurangozeb, M. Mohammad, and M. Hossain, “A 35 mW 10 Gb/s ADC-DSP less direct digital sequence detector and equalizer in 65nm CMOS,” in *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, 2016, pp. 1–2.
- [13] O. E. Agazzi *et al.*, “A 90 nm CMOS DSP MLSD Transceiver With Integrated AFE for Electronic Dispersion Compensation of Multimode Optical Fibers at 10 Gb/s,” *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2939–2957, Dec. 2008.
- [14] J. Cao *et al.*, “21.7 A 500mW digitally calibrated AFE in 65nm CMOS for 10Gb/s Serial links over backplane and multimode fiber,” in *2009 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*, 2009, p. 370–371,371a.
- [15] H. Yamaguchi *et al.*, “A 5Gb/s transceiver with an ADC-based feedforward CDR and CMA adaptive equalizer in 65nm CMOS,” in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, 2010, pp. 168–169.

- [16] B. Zhang *et al.*, “A 195mW / 55mW dual-path receiver AFE for multistandard 8.5-to-11.5 Gb/s serial links in 40nm CMOS,” in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2013, pp. 34–35.
- [17] T. Beukema *et al.*, “A 6.4-Gb/s CMOS SerDes core with feed-forward and decision-feedback equalization,” *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2633–2645, Dec. 2005.
- [18] V. Balan *et al.*, “A 4.8-6.4-Gb/s serial link for backplane applications using decision feedback equalization,” *IEEE J. Solid-State Circuits*, vol. 40, no. 9, pp. 1957–1967, Sep. 2005.
- [19] R. Payne *et al.*, “A 6.25Gb/s binary adaptive DFE with first post-cursor tap cancellation for serial backplane communications,” in *ISSCC. 2005 IEEE International Digest of Technical Papers. Solid-State Circuits Conference, 2005.*, 2005, p. 68–585 Vol. 1.
- [20] N. Kocaman *et al.*, “A 3.8 mW/Gbps Quad-Channel 8.5-13 Gbps Serial Link With a 5 Tap DFE and a 4 Tap Transmit FFE in 28 nm CMOS,” *IEEE J. Solid-State Circuits*, vol. 51, no. 4, pp. 881–892, Apr. 2016.
- [21] S. Kasturia and J. H. Winters, “Techniques for high-speed implementation of nonlinear cancellation,” *IEEE J. Sel. Areas Commun.*, vol. 9, no. 5, pp. 711–717, Jun. 1991.
- [22] V. Stojanovic *et al.*, “Adaptive equalization and data recovery in a dual-mode (PAM2/4) serial link transceiver,” in *2004 Symposium on VLSI Circuits, 2004. Digest of Technical Papers*, 2004, pp. 348–351.
- [23] M. Harwood *et al.*, “A 12.5Gb/s SerDes in 65nm CMOS Using a Baud-Rate ADC with Digital Receiver Equalization and Clock Recovery,” in *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, 2007, pp. 436–591.

- [24] B. Razavi, *Design of Analog CMOS Integrated Circuits*, 1 edition. Boston, MA: McGraw-Hill Education, 2000.
- [25] R. G. Beausoleil, M. McLaren, and N. P. Jouppi, "Photonic Architectures for High-Performance Data Centers," *IEEE J. Sel. Top. Quantum Electron.*, vol. 19, no. 2, pp. 3700109–3700109, Mar. 2013.
- [26] K. Hara, S. Kimura, H. Nakamura, N. Yoshimoto, and H. Hadama, "New AC-Coupled Burst-Mode Optical Receiver Using Transient-Phenomena Cancellation Techniques for 10 Gbit/s-Class High-Speed TDM-PON Systems," *J. Light. Technol.*, vol. 28, no. 19, pp. 2775–2782, Oct. 2010.
- [27] T. D. Ridder *et al.*, "10 Gbit/s burst-mode post-amplifier with automatic reset," *Electron. Lett.*, vol. 44, no. 23, pp. 1371–1373, Nov. 2008.
- [28] X. Yin *et al.*, "A 10Gb/s burst-mode TIA with on-chip reset/lock CM signaling detection and limiting amplifier with a 75ns settling time," in *2012 IEEE International Solid-State Circuits Conference*, 2012, pp. 416–418.
- [29] S. Galal and B. Razavi, "10-Gb/s limiting amplifier and laser/modulator driver in 0.18- $\mu\text{m}$  CMOS technology," *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp. 2138–2146, Dec. 2003.
- [30] M. Hossain and A. C. Carusone, "5 #x2013;10 Gb/s 70 mW Burst Mode AC Coupled Receiver in 90-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 45, no. 3, pp. 524–537, Mar. 2010.
- [31] J. Luo, J. Parra-Cetina, P. Landais, H. J. S. Dorren, and N. Calabretta, "Performance Assessment of 40 Gb/s Burst Optical Clock Recovery Based on Quantum Dash Laser," *IEEE Photonics Technol. Lett.*, vol. 25, no. 22, pp. 2221–2224, Nov. 2013.

- [32] F. Aflatouni and H. Hashemi, "A 1.8mW Wideband 57dB $\Omega$  transimpedance amplifier in 0.13  $\mu\text{m}$  CMOS," in *2009 IEEE Radio Frequency Integrated Circuits Symposium*, 2009, pp. 57–60.
- [33] C.-Y. Wang, C.-S. Wang, and C.-K. Wang, "An 18-mW two-stage CMOS transimpedance amplifier for 10 Gb/s optical application," in *Solid-State Circuits Conference, 2007. ASSCC '07. IEEE Asian*, 2007, pp. 412–415.