

UNIVERSITY OF PAVIA

DEPARTMENT OF ELECTRICAL, COMPUTER, AND BIOMEDICAL ENGINEERING

MASTER'S THESIS IN ELECTRONICS ENGINEERING

System–Level Analysis and Design of a 112–Gb/s PAM–4 All–Digital Baud–Rate CDR with Unbalanced MMPD

Supervisor:
Prof. Danilo MANSTRETTA
Co–Supervisor:
Kamal Baghirli

Author:
Adib Gani Silanurcahya
Student's Number:
531403

Academic year 2025/26

Declaration of Authorship

I, Author name, declare that this thesis titled 'System-Level Analysis and Design of a 112-Gb/s PAM-4 All-Digital Baud-Rate CDR with Unbalanced MMPD' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: __23/10/2026____

ABSTRACT

This thesis presents a system-level analysis, discrete-time Z-domain model, behavioral Verilog-A implementation, and co-simulation verification of an All-Digital Baud-Rate Clock and Data Recovery (ADCDR) transceiver for Extra Short Reach (XSR) die-to-die interconnects. The transceiver uses an Unbalanced Mueller–Müller Phase Detector (MMPD). By using sub-rate quadrature clocking and single-sample-per-symbol baud-rate sampling, the proposed receiver avoids the need for high-power analog-to-digital converters, digital signal processing engines, and oversampled clock distribution trees.

To address the low transition density and reduced loop gain common in conventional baud-rate phase detectors, we implement an Unbalanced MMPD architecture that incorporates asymmetric data patterns alongside conventional symmetric ones to increase phase detector transition density.

A discrete-time Z-domain sampled-data model defines exact loop stability boundaries while accounting for retiming pipeline latency. By optimally tuning the digital loop filter coefficients, the stabilizing loop zero heavily overdamps the feedback loop and suppresses closed-loop jitter transfer peaking to levels well within standard interoperability mask limits. MATLAB and Cadence Spectre behavioral simulations confirm rapid phase-step acquisition and wide sinusoidal jitter tolerance margins relative to official industry standards. The resulting all-digital architecture offers an energy-efficient, compact, and highly scalable solution for next-generation multi-lane interconnects.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude and sincere appreciation to my thesis advisor, Prof. Danilo Manstretta, for his invaluable guidance, unwavering support, and profound technical insights throughout my Master's research. His expertise in high-speed mixed-signal integrated circuits and SerDes architectures has been instrumental in shaping this thesis. His enthusiasm for research, high academic standards, and rigorous methodology have guided my technical work and inspired my growth as an engineer.

I am equally grateful to my tutor, Kamal Baghirli, for his time, constructive criticism, and insightful feedback during the evaluation of this manuscript. I would also like to thank my colleagues and lab mates in the Micro Laboratory. Technical discussions, collaborative problem-solving, and peer reviews on Verilog-A behavioral modeling, Spectre simulation setups, and discrete-time z-domain analysis fostered an enriching and motivating research environment.

Finally, my heartfelt gratitude goes to my family, parents, and friends for their unconditional love, patience, and unwavering encouragement throughout my graduate studies. Their belief in my abilities and continuous support during long simulation runs and manuscript writing made this thesis possible.

List of Figures

Figure 1.1: XSR die-to-die interconnect context for a multi-chip package. 1
Figure 1.2: NRZ vs. PAM-4 Comparison. 3
Figure 1.3: Simple CDR Circuit [10]. 4
Figure 1.4: (a) Conventional Analog CDR, (b) A digital CDR [18]. 5
Figure 1.5: (a) Hogge Phase Detector, (b) its waveforms [10]. 6
Figure 1.6: (a) Alexander Phase Detector, (b) its waveforms, and (c) Three-point sampling of data by clock [10]. .. 8
Figure 1.7: (a) Sample locations and required clock phases comparison between 2× oversampled edge-based detection and baud-rate Mueller–Müller detection, (b) The operation of the sign-sign Mueller-Muller phase detector [17]. 9
Figure 2.1: ADCDR Receiver Architecture Block Diagram. 12
Figure 2.2: (a) Channel and CTLE Frequency Response, (b) 56 GBaud/s PAM4 XSR Channel without Equalization, (c) 56 GBaud/s PAM4 XSR Channel with CTLE. 16
Figure 2.3: Eye Diagram of the output Signal from the VGA. 18
Figure 2.4: Data and Error Samplers and their timing diagram. 18
Figure 2.5: Eye Diagram of the input signal and the voltage references. 20
Figure 2.6: Operational principle of the unbalanced MMPD. (a) D[n] = +3 and (b) D[n] = +1[1]. 23
Figure 2.7: Digital Loop Filter and DCO. 26
Figure 2.8: PAM-4 Decoder [19]. 29
Figure 3.1: The z-domain linearized model of CDR. 30
Figure 3.2: Phase Detector S-Curve Characteristic Comparison [1]. 31
Figure 3.3: Open-Loop Bode Magnitude and Phase Frequency Response. 43
Figure 3.4: Jitter Transfer. 46
Figure 3.5: Step Phase and Frequency Response. 49
Figure 3.6: Jitter Tolerance. 51
Figure 4.1: The integral path control output Vint. 53
Figure 4.2: Locking Regime. 54
Figure 4.3: Periodic Jitter Histogram. 55
Figure 4.4: (a) The Proportional Output Control Voltage, and (b) The Integral Output Control Voltage (Phase Step and no Frequency change). 57
Figure 4.5: (a) In-range Frequency Acquisition, and (b) Out-of-range Frequency Acquisition. 59

List of Tables

Table 2.1: Clock Arrangement of the samplers. 19
Table 2.2: Symmetric Data Patterns [1]. 21
Table 2.3: Asymmetric Data Patterns [1]. 22
Table 2.4: PAM-4 Decoder Truth Table. 29
Table 3.1: Z-Domain Loop Parameters 49
Table 4.1: Performance Compliance Matrix vs. OIF-CEI-112G-XSR / IEEE 802.3ck Specifications. 60
Table 4.2: State-of-the-Art Performance Benchmarking Matrix with Literature. 61

List of Abbreviations

ADC	Analog-to-Digital Converter
ADCDR	All-Digital Clock and Data Recovery
AFE	Analog Front-End
AGC	Automatic Gain Control
AI	Artificial Intelligence
BER	Bit Error Rate
CDR	Clock and Data Recovery
CEI	Common Electrical I/O / Interface
CID	Consecutive Identical Digits
CML	Current Mode Logic
CP	Charge-Pump
CPO	Co-Packaged Optics
CTLE	Continuous-Time Linear Equalizer
D2D	Die-to-Die
DAC	Digital-to-Analog Converter
DCO	Digitally Controlled Oscillator
JG	Jitter Generation
JTOL	Jitter Tolerance
JTRAN	Jitter Transfer
LPF	Low-pass filters
LSB	Least Significant Bit
MCM	Multi-chip Module
ML	Machine Learning
MMPD	Mueller-Müller Phase Detector
MSB	Most Significant Bit

MV	Majority Vote
NRZ	Non-Return-to-Zero
PAM	Pulse Amplitude Modulation
PD	Phase Detector
PI	Phase Interpolator
PM	Phase Margin
ppm	Parts Per Million
PRBS	Pseudo-Random Binary Sequence
PVT	Process, Voltage, and Temperature
RLM	Receiver Level Mismatch
RTL	Register Transfer Level
RX	Receiver
SBR	Single Bit Response
SerDes	Serializer/Deserializer
SJ	Sinusoidal Jitter
SNR	Signal-to-noise ratio
TX	Transmitter
UI	Unit Interval
USR	Ultra-Short-Reach
VCO	Voltage-Controlled Oscillator
VGA	Variable Gain Amplifier
XSR	Extra-Short-Reach

Table of Contents

Declaration of Authorship ii

ABSTRACT iii

ACKNOWLEDGEMENTS iv

List of Figures v

List of Tables vi

List of Abbreviations vii

Table of Contents ix

1. Introduction and Fundamentals 1

1.1 Background and Motivation 1

1.2 Clock and Data Recovery (CDR) Basics 2

1.2.1 CDR Principles and Loop Operation 2

1.2.2 Analog CDRs vs. All-Digital CDRs (ADCDRs) 4

1.2.3 Phase Detector Topologies 6

1.2.4 PAM-4 Timing Extraction Challenges 9

1.3 Thesis Objectives and Scope 10

1.4 Thesis Organization 11

2. System Architecture 12

2.1 System Architecture Overview 12

2.2 Analog Front-End (AFE) Behavioral Modeling (Verilog-A) 14

2.2.1 Continuous-Time Linear Equalizer (CTLE) 14

2.2.2 Variable Gain Amplifier (VGA) 17

2.2.3 7-Slicer Array 18

2.3 Digital CDR Loop Modeling (Verilog-A) 20

2.3.1 Unbalanced Mueller-Müller Phase Detector 20

2.3.2 Spatial Majority Voter Module 23

2.3.3 2nd-Order Digital Loop Filter (DLF) 24

2.3.4 Digitally Controlled Oscillator (DCO) 26

2.3.5 Data Recovery (DR) Path 28

3. Discrete-Time Z-Domain Modeling & Loop Stability 30

3.1 Discrete Z-Domain System Formulation 30

3.1.1 Linearized Phase-Domain System Topology 30

3.1.2 Z-Domain Sub-Block Transfer Functions 31
3.1.3 Closed-Loop Phase and Error Transfer Function Derivation 35
3.2 Pipeline Latency ($\mathbf{D}$) Impact 35
3.2.1 Characteristic Equation with Feedback Latency 36
3.2.2 Pipeline Latency Breakdown 36
3.3 Methodology to Select $\boldsymbol{\alpha}(\boldsymbol{Kp})$ for Loop Stability 39
3.4 Phase Margin (PM) Equations & Frequency Response Analysis 41
3.4.1 Continuous-Time Equivalent Mapping 41
3.4.2 Analytical Derivations for Loop Bandwidth ($\boldsymbol{fbw}$) and Phase Margin (PM) 42
3.4.3 Bode Plot Analysis & Gardner's Stability Criterion 42
3.5 Loop Zero ($\boldsymbol{fz}$) Optimization & Jitter Transfer Peaking Bounds 44
3.5.1 Placement of Loop Zero ($\boldsymbol{fz}$) 44
3.5.2 Mathematical Bounds for < 0.1 dB Jitter Transfer Peaking 44
3.6 Lock Convergence Dynamics & MATLAB Verification 46
3.6.1 Step Phase Response Discussion ($\boldsymbol{\Delta\phi 0 = +0.35}$ UI) 46
3.6.2 Step Frequency Response Discussion 47
3.7 Jitter Tolerance (JTOL) Modeling 50
3.7.1 Sinusoidal Jitter (SJ) Injection Formulation 50
3.7.2 Compliance Verification Against IEEE 802.3ck Specifications 50
4. Simulation Results & Verification 52
4.1 Phase and Frequency Locking Dynamics 52
4.1.1 Coarse Frequency Acquisition 52
4.1.2 Fine Phase Acquisition and Tracking 55
4.2 Performance Compliance Summary for OIF-CEI-112G-XSR Links 59
4.3 State-of-the-Art Literature Comparison 60
5. Conclusion and Future Work 62
5.1 Summary of Contributions 62
5.2 Future Research Directions 64
References 66
A MATLAB 70
B Verilog-A 79

1. Introduction and Fundamentals

1.1 Background and Motivation

The rapid proliferation of data-intensive workloads driven by artificial intelligence (AI), machine learning (ML) training clusters, cloud computing, and hyperscale data centers has generated an unprecedented demand for high-bandwidth interconnects [1],[2]. As server racks and AI accelerator chiplets process multi-terabit datasets, the bandwidth requirements for chip-to-chip, die-to-die (D2D), and optical module interfaces have scaled aggressively. To meet this throughput demand without suffering from excessive pin count and interconnect congestion, state-of-the-art Serializer/Deserializer (SerDes) architectures have shifted toward per-lane data rates of 112 Gbps and beyond [1],[3].

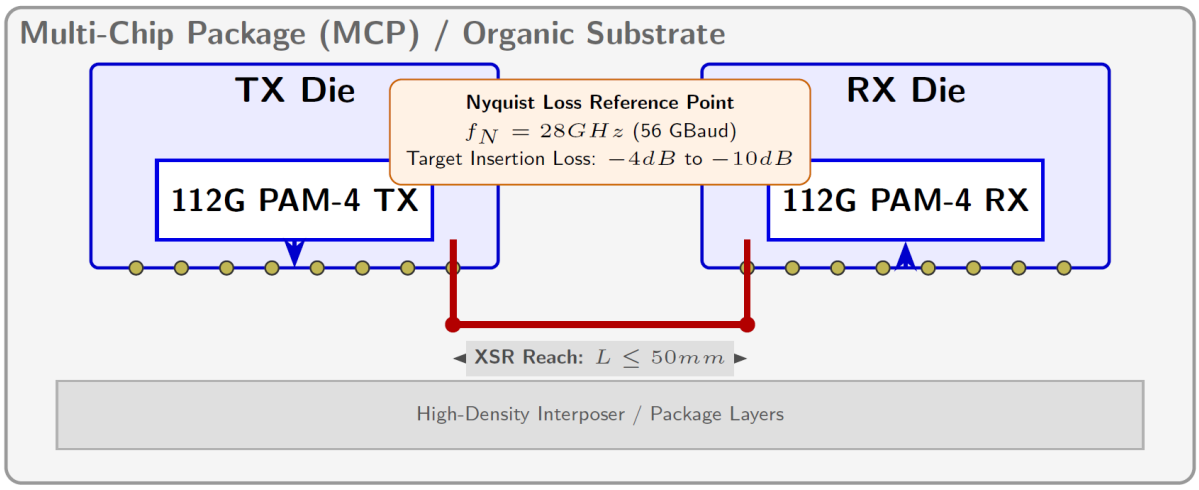


Figure 1.1: XSR die-to-die interconnect context for a multi-chip package.

In extra-short-reach (XSR) and ultra-short-reach (USR) environments, such as co-packaged optics (CPO) and intra-package multi-chip module (MCM) interconnects, physical channels have trace lengths under 50 mm [4]. Although these channels exhibit significantly lower frequency-dependent attenuation than long-reach (LR) backplanes, operating at 112 Gbps with

conventional Non-Return-to-Zero (NRZ) modulation presents insurmountable physical limitations. In NRZ (2-level) signaling, a 112 Gbps bit rate requires a symbol rate of 112 GBaud, resulting in a Nyquist frequency of 56 GHz [1],[6]. At 56 GHz, copper traces and package parasitics suffer extreme attenuation of 20-30 dB, severe inter-symbol interference (ISI), and severe skin-effect losses [5],[6].

To circumvent these high-frequency channel losses, modern high-speed wireline standards such as OIF-CEI-112G-XSR and IEEE 802.3ck have universally adopted 4-level Pulse Amplitude Modulation (PAM-4) [1],[5],[8]. By transmitting 2 bits per symbol across 4 voltage levels, PAM-4 halves the required symbol rate to 56 GBaud for a 112 Gbps throughput [3],[6]. Consequently, the Nyquist frequency is reduced to 28 GHz [5]. In a typical XSR channel, insertion loss at 28 GHz remains below 10 dB (typically 8.5 dB to 9.1 dB) [4],[6]. However, PAM-4 introduces a 9.54 dB signal-to-noise ratio (SNR) penalty compared to NRZ because it reduces eye height by 2/3 ($20 \log10(1/3) \approx -9.54$ dB) for the same peak-to-peak voltage swing [5],[8].

1.2 Clock and Data Recovery (CDR) Basics

1.2.1 CDR Principles and Loop Operation

At the core of every high-speed SerDes receiver lies the Clock and Data Recovery (CDR) circuit. The CDR's primary function is to extract a phase-aligned clock signal from the incoming asynchronous, distorted data stream and sample the data at the optimal eye opening with a minimal bit error rate (BER $\leq 10^{-12}$) [8],[10]. A conventional feedback CDR comprises three fundamental sub-blocks:

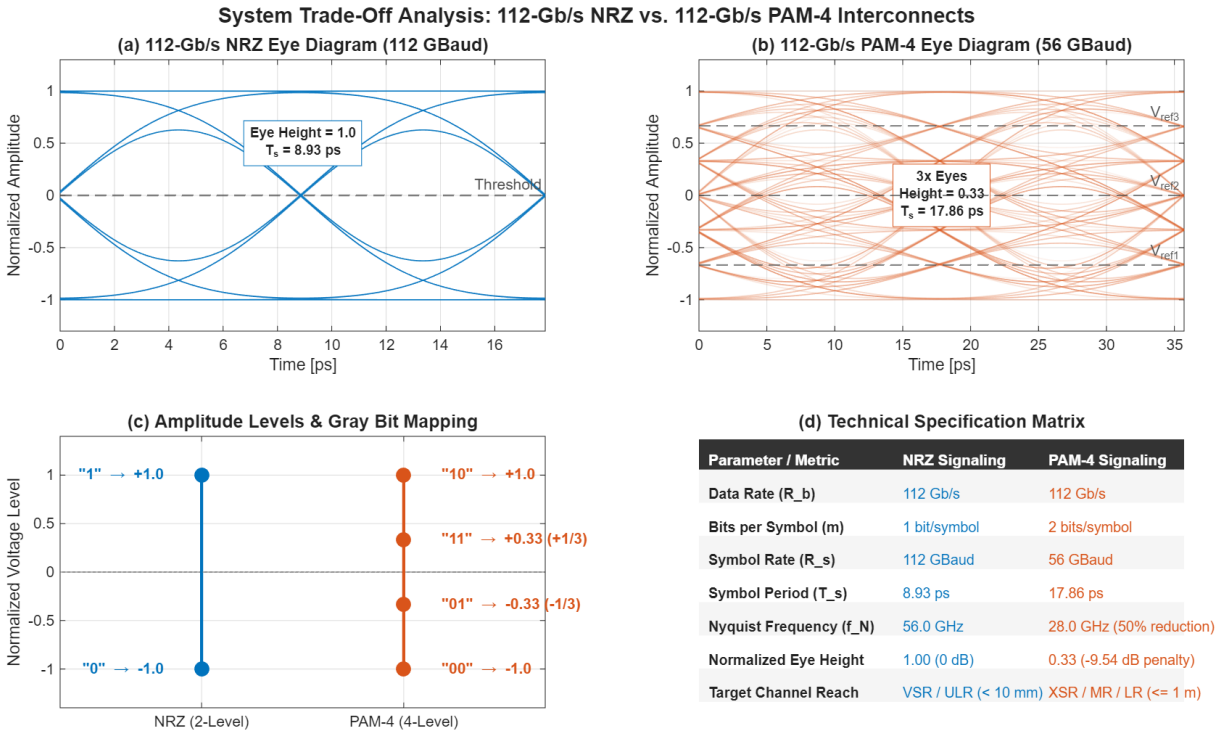


Figure 1.2: NRZ vs. PAM-4 Comparison.

1. **Phase Detector** (**PD**): Compares the phase of the incoming data transitions with the recovered clock and generates phase error pulses (Early/Late or proportional error signals) [5],[8].
2. **Loop Filter** (**LF**): Low-pass filters the high-frequency phase error pulses, attenuating high-frequency noise and establishing the closed-loop tracking bandwidth and stability dynamics [5].
3. **Clock Generation** / **Phase Actuator**: Converts the filtered control signal into an adjusted clock phase or frequency, typically realized using a Voltage-Controlled Oscillator (VCO), Digitally Controlled Oscillator (DCO), or Phase Interpolator (PI) [8].

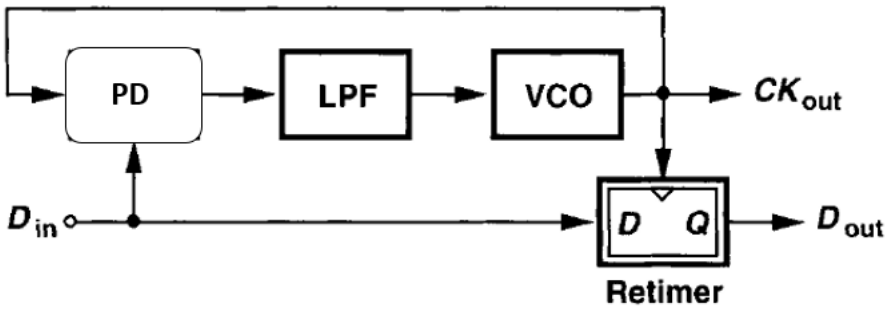


Figure 1.3: Simple CDR Circuit [10].

1.2.2 Analog CDRs vs. All-Digital CDRs (ADCDRs)

Historically, SerDes transceivers relied on Charge-Pump Analog CDRs (CP-CDRs), where a phase-frequency detector drives a charge pump and a passive RC loop filter to adjust an LC-VCO [9],[10]. However, as integrated circuit fabrication scales into ultra-deep-submicron CMOS and FinFET process nodes (e.g., 28 nm, 12 nm, 7 nm, and 3 nm), analog CDR architectures face severe physical degradation [1],[13]:

- **Gate Leakage and Output Impedance**: Reduced transistor output impedance and high gate-leakage currents in advanced FinFET nodes degrade charge-pump current matching and cause control voltage drift during long Consecutive Identical Digits (CID) [14].
- **Area Overhead**: The large passive capacitors required for analog loop filters (C_1 > 50 pF) do not scale with process technology, occupying an excessive silicon die area (> 0.05 mm^2) [1], [8].
- **Reduced Voltage Headroom**: Ultra-low supply voltages (VDD < 0.9 V) constrain charge-pump compliance voltage ranges and varactor tuning headroom in LC-VCOs [9].

To overcome these scaling bottlenecks, All-Digital Clock and Data Recovery (ADCDR) architectures have emerged as the preferred solution [6],[14]. An ADCDR replaces the analog

charge pump and passive RC filter with a Digital Loop Filter (DLF) and utilizes a Digitally-Controlled Oscillator (DCO) or Phase Interpolator (PI) [6],[14]. ADCDRs offer distinct architectural advantages:

1. **Advanced Process Scaling**: Synthesizable DLF logic scales directly with technology node shrink, dramatically reducing silicon area and power dissipation [14].
2. **Zero Charge-Pump Leakage and CID Immunity**: Digital registers hold control states indefinitely without voltage drift during CID sequences [12].
3. **Reconfigurability and Adaptive Loop Dynamics**: Proportional (Kp) and integral (Ki) filter gains can be reconfigured dynamically via digital registers to optimize jitter tolerance (JTOL) and loop bandwidth [6],[15].
4. **Enhanced Testability and Portability**: Digital filter blocks can be synthesized automatically using standard cell libraries, accelerating porting across nanometer nodes [14].

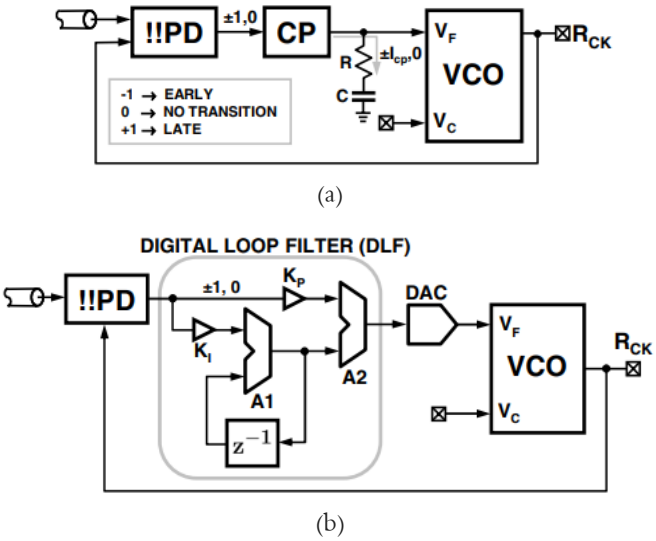
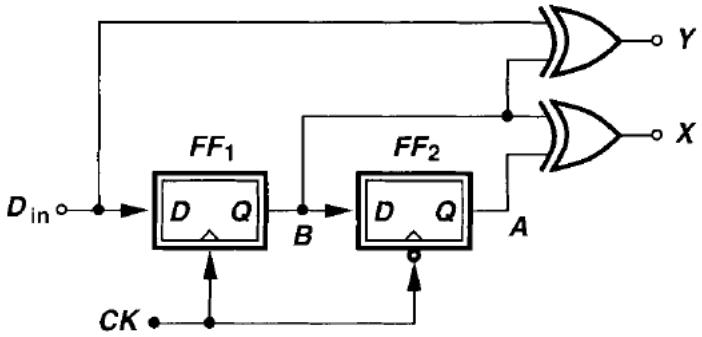


Figure 1.4: (a) Conventional Analog CDR, (b) A digital CDR [18].

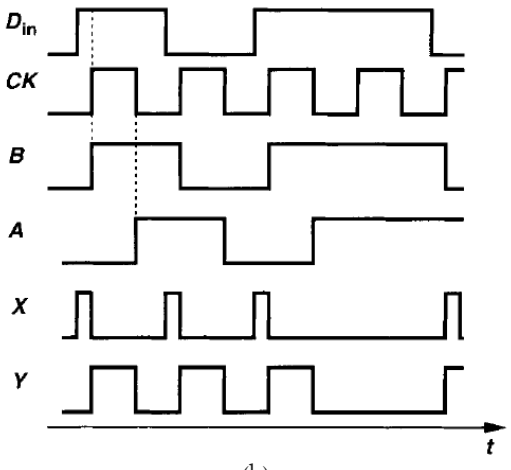
1.2.3 Phase Detector Topologies

The choice of phase detector fundamentally dictates the tracking bandwidth, power consumption, and hardware complexity of the SerDes receiver:

1. **Hogge Phase Detector**: A linear PD that generates error pulses whose pulse widths are linearly proportional to the phase error. While providing linear loop dynamics, it requires high-speed linear CML latches and XOR gates, which suffer from headroom limitations and high power at 56 GHz.



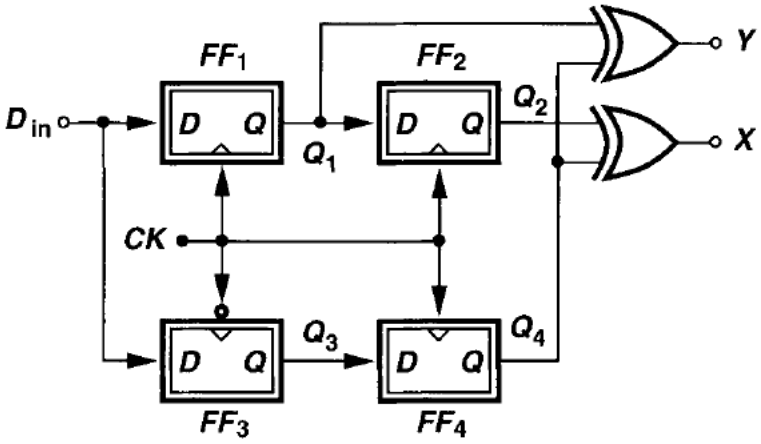
(a)



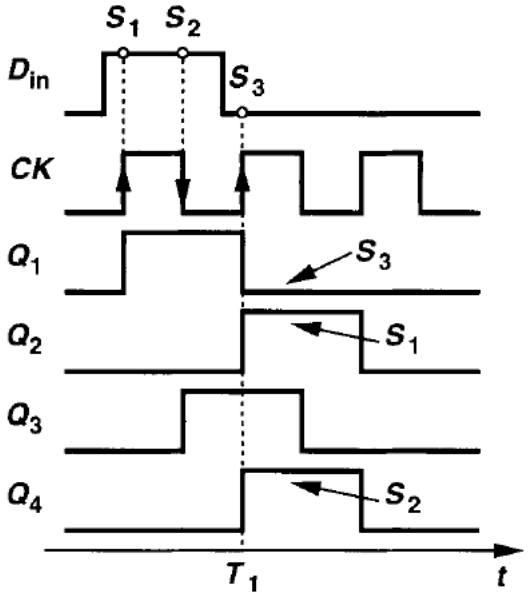
(b)

Figure 1.5: (a) Hogge Phase Detector, (b) its waveforms [10].

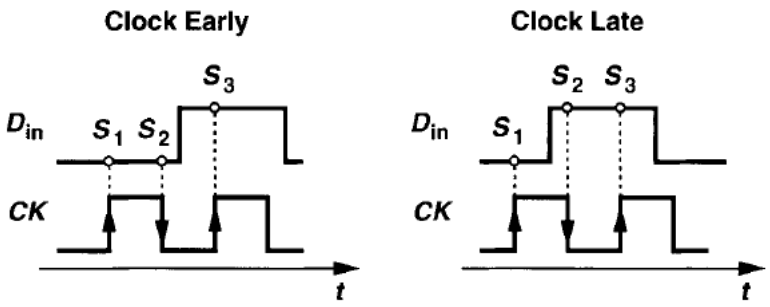
2. **Alexander (Bang–Bang) Phase Detector**: Uses 2× oversampling (two samples per Unit Interval, UI: data sample and edge sample) to generate binary Early/Late phase decisions [10], [13]. While offering high phase detector gain (K_{PD}) and fast acquisition, the required 2× clocking frequency doubles the sampler count and clock tree power consumption at 56 GBaud [13].



(a)



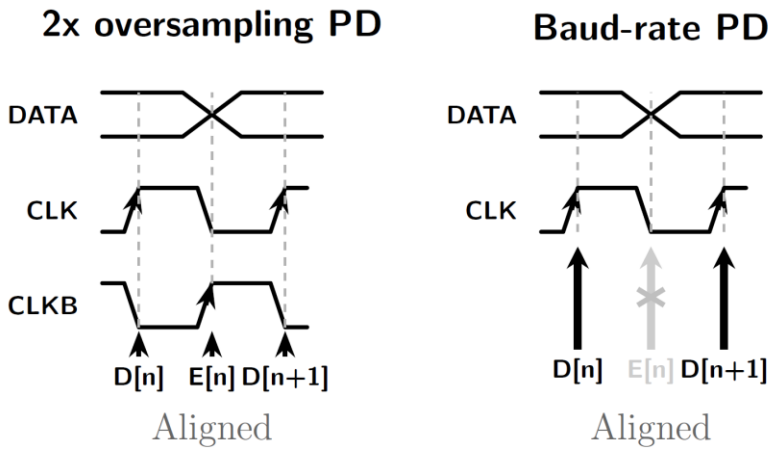
(b)



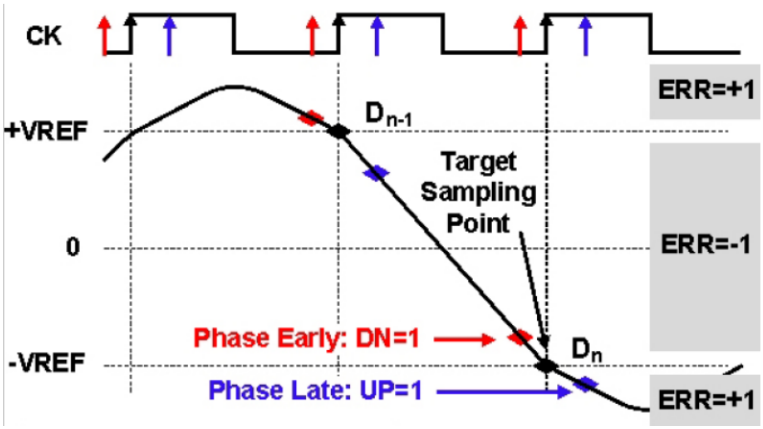
(c)

Figure 1.6: (a) Alexander Phase Detector, (b) its waveforms, and (c) Three-point sampling of data by clock [10].

3. **Baud-Rate Mueller-Müller Phase Detector** (**MMPD**): Samples the data waveform only once per UI (1× sampling rate) at the symbol center [6],[16]. Timing phase information is extracted by evaluating the amplitudes of the current and adjacent samples (D[n-1], D[n], D[n+1]) and the error sample E[n] [6],[7]. By eliminating edge samplers, the baud-rate MMPD reduces clock load and AFE capacitance by 50%, making it the preferred choice for high-speed PAM-4 receivers [1],[6],[7].



(a)



(b)

Figure 1.7: (a) Sample locations and required clock phases comparison between 2× oversampled edge-based detection and baud-rate Mueller–Müller detection, (b) The operation of the sign-sign Mueller-Muller phase detector [17].

1.2.4 PAM–4 Timing Extraction Challenges

Transitioning from NRZ to PAM-4 introduces unique timing recovery challenges:

- **12 Potential Transition Scenarios**: NRZ has only 2 possible transitions (0 ⟶ 1 and 1 ⟶ 0). PAM-4 has 4 signal levels ({-3, -1, +1, +3}), resulting in 4 × 3 = 12 distinct level transitions [7],[13]. Because inner transitions (±1 ⟶ ±3 or ±1 ⟶ ∓1) cross decision thresholds at different times due to finite channel rise and fall times, data-dependent jitter (DDJ) increases significantly [6],[13].
- **Reduced Eye Height and SNR Penalty**: The vertical eye height is only one-third, making slicer decision thresholds highly sensitive to additive noise, baseline wander, and offset [1],[11].
- **Receiver Level Mismatch** (**RLM**): Nonlinear compression in the Analog Front-End (AFE) distorts the eye openings, causing conventional baud-rate MMPD algorithms to converge to suboptimal phase offsets [1],[7].

1.3 Thesis Objectives and Scope

This thesis aims to design and analytically model in discrete time, then verify through high-fidelity behavioral simulation, a 112 Gbps PAM-4 quarter-rate All-Digital Clock and Data Recovery (ADCDR) system for XSR links.

To achieve this, the specific research goals of this work are:

1. **Unbalanced MMPD Architecture**: Design and model an Unbalanced Mueller-Müller Phase Detector that uses asymmetric pattern filtering to increase active transition density to 31.25%.
2. **Discrete-Time Z-Domain Loop Modeling**: Develop an accurate discrete-time sampled-data loop model in MATLAB that incorporates a 4:1 Spatial Majority Voter, a second-order DLF with power-of-two gain parameters, and a digitally controlled oscillator [6].
3. **Pure Verilog-A Behavioral Testbench Suite**: develop complete Verilog-A behavioral modules in Cadence Spectre, including a channel/CTLE AFE, a 7-slicer array (3 data slicers and 4 error slicers), an Unbalanced MMPD, a 4:1 voter, a second-order DLF, and a quadrature DCO to verify top-level closed-loop acquisition, jitter transfer, and jitter tolerance (JTOL) without computationally prohibitive transistor-level AMS co-simulations.

Scope Boundary: Transistor-level, transistor-by-transistor layout and full AMS co-simulation are explicitly excluded from this thesis. Instead, this thesis emphasizes rigorous mathematical modeling in MATLAB and pure Verilog-A behavioral verification in Cadence Specter to enable fast, accurate system-level architectural exploration.

1.4 Thesis Organization

The remainder of this thesis is divided into four chapters covering the proposed 112 Gbps PAM-4 ADCDR system's architectural design, mathematical modeling, simulation verification, and conclusions:

- **Chapter 2**: ADCDR and AFE system architecture details the overall 112 Gbps PAM-4 ADCDR receiver chain, including CTLE equalization, VGA common-mode level shifting, a 7-slicer array, unbalanced MMPD pattern selection, second-order DLF difference equations, and a noise-modeled DCO.

- **Chapter 3**: MATLAB Discrete-Time Modeling & System Validation develops a mathematical loop model in the discrete-time Z-domain, discretizes the pulse response, applies fixed-point quantization, evaluates transient phase-lock acquisition with initial frequency offset, and performs Sinusoidal Jitter (SJ) tolerance (JTOL) sweeps against IEEE 802.3ck specifications.

- **Chapter 4**: Cadence Verilog-A Simulation Results & Verification covers 100% pure Verilog-A behavioral simulation setups in Spectre, locking dynamics, impairments, and closed-loop clock recovery.

- **Chapter 5**: Conclusion and Future Works summarizes the key achievements and outlines future research directions in advanced FinFET/GAAFET nodes, adaptive DFE equalization, and ADC-DSP architectures.

2. System Architecture

2.1 System Architecture Overview

The push for higher bandwidth density in data center interconnects, multi-chip modules (MCMs), and co-packaged optics (CPO) has made 112 Gbps per lane the industry baseline for Extra Short Reach (XSR) and Ultra Short Reach (USR) applications [1],[2]. At a 56 GBaud symbol rate using 4-level Pulse Amplitude Modulation (PAM-4), receivers must recover data over physical channels with up to 10 dB of Nyquist insertion loss at 28 GHz, while keeping power consumption below 2 pJ/bit [16].

Figure 2.1 shows the overall block diagram of the proposed 112 Gbps PAM-4 quarter-rate All-Digital Clock and Data Recovery (ADCDR) transceiver chain. The architecture comprises an Analog Front-End (AFE) signal-conditioning path and an All-Digital CDR timing-recovery feedback loop.

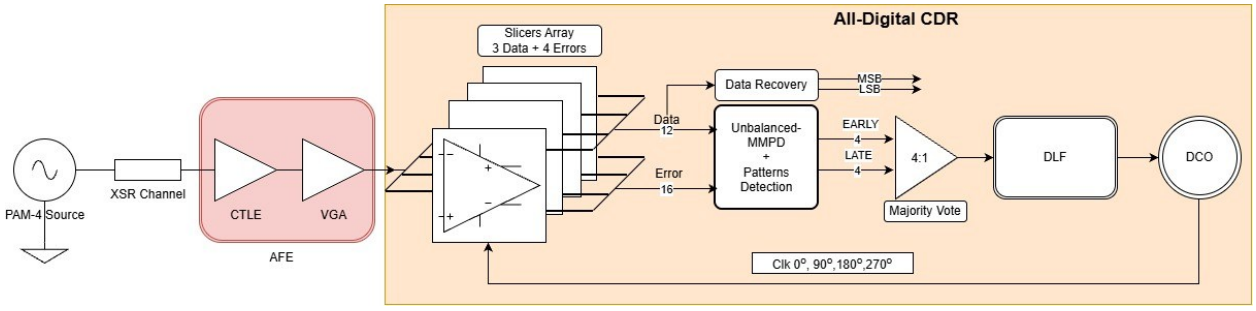


Figure 2.1: ADCDR Receiver Architecture Block Diagram.

The high-speed signal flow operates as follows:

1. The signal is generated by a PAM-4 PRBS Voltage Source, producing a 4-level pseudo-random binary sequence operating at 56 GBaud (112 Gbps). The signal propagates through an XSR Channel Model (exhibiting approximately 8.5 dB to 10 dB loss at the 28 GHz Nyquist frequency) before reaching the receiver.

2. **AFE Equalization and Gain Scaling**: The differential signal passes through a Continuous-Time Linear Equalizer (CTLE), which provides up to 10 dB of high-frequency gain at the 28 GHz Nyquist frequency to compensate for channel attenuation. The equalizer output then drives a Variable Gain Amplifier (VGA), with a gain range of -6 dB to +14 dB. The VGA regulates the output differential swing to a target amplitude of ±150 mV peak differential (±300 mVpp,diff) and shifts the DC common-mode level from 0.0 V to 0.5 V.
3. **7-Slicer Array Sampling**: The equalized signal is digitized by a 7-slicer array comprising 3 data slicers (voltage decision thresholds at +2h0 = +600 mV, +500 mV, and -2h0 = +400 mV) and 4 error slicers (voltage reference thresholds at +3h0 = +650 mV, +1h0 = +550 mV, -1h0 = +450 mV, and -3h0 = 350 mV). Dynamic latch comparators sample the signal at an effective symbol decision rate of 56 GHz, using a quarter-rate, 14 GHz, 4-phase quadrature clocking scheme (I0, Q90, I180, Q270).
4. **Digital Phase Detection**: The slicers' quantized thermometer-code outputs feed an Unbalanced Mueller-Müller Phase Detector (MMPD). By evaluating three consecutive data decisions (D[n-1], D[n], D[n+1]) and filtering for valid transitions, the unbalanced MMPD extracts timing phase errors at the baud rate (one sample per UI), increasing the active transition density to 31.25% [1].
5. **Loop Filtering and Frequency Control**: A 4:1 Spatial Majority Voter downsamples the phase-error pulses and drives a synthesizable second-order Digital Loop Filter (DLF). The proportional path applies the phase detector's decision directly to the DCO with no pipeline latency, enabling fast phase corrections on every clock cycle. In parallel, the integral path uses a 2:16 deserializer accumulator to downsample phase-error decisions

over 16 clock cycles, reducing digital operating speed and filtering high-frequency phase noise. The accumulated error then updates a 6-bit digital control code (0 to 63 steps), which drives a DAC to smoothly adjust the DCO's baseline frequency and compensate for long-term frequency offsets (ppm drift).

6. **Clock Actuation**: Operating at a nominal quarter-rate frequency of 14 GHz, the 4-phase DCO produces quadrature square-wave clock outputs at 0°, 90°, 180°, and 270° to sample 112 Gbps PAM-4 data with an effective sample rate of 56 GHz. It dynamically adjusts its instantaneous output frequency by summing the digital loop filter's proportional and integral control signals, with a tuning gain of 500 MHz/LSB. It also applies cycle-by-cycle Gaussian phase noise to reproduce an integrated random period jitter profile of 150 fs RMS.

2.2 Analog Front-End (AFE) Behavioral Modeling (Verilog-A)

To support rapid architecture exploration and system-level verification without the computational cost of transistor-level simulations, the AFE sub-blocks are modeled as behavioral modules in pure Verilog-A.

2.2.1 Continuous-Time Linear Equalizer (CTLE)

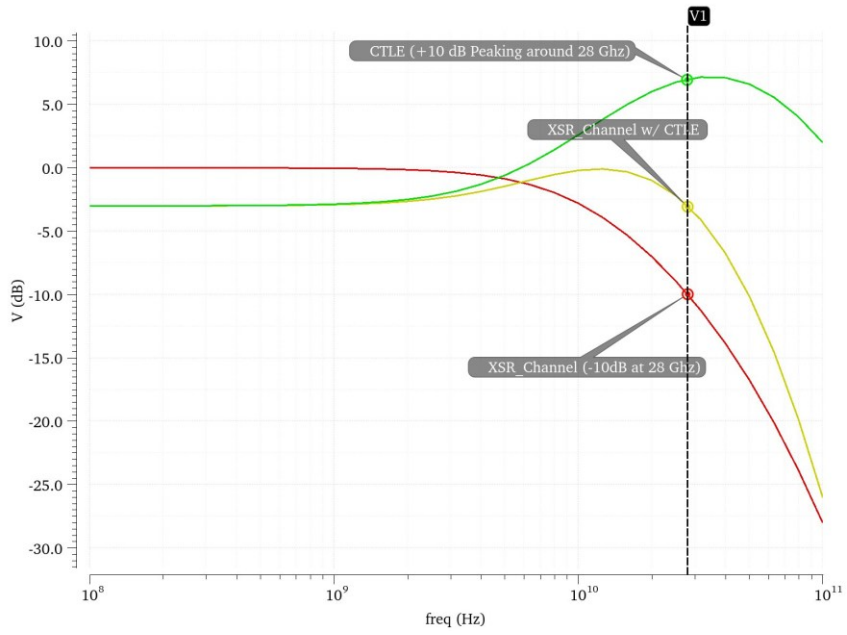
The CTLE compensates for frequency-dependent channel loss by introducing high-frequency peaking near the Nyquist frequency while maintaining flat low-frequency attenuation. The continuous-time small-signal transfer function of the dual-pole, single-zero CTLE is modeled in the s-domain as:

$$H_{\mathrm{CTLE}}(s) = A_0 \cdot \frac{1 + \frac{s}{\omega_z}}{\left(1 + \frac{s}{\omega_{p1}}\right)\left(1 + \frac{s}{\omega_{p2}}\right)}$$

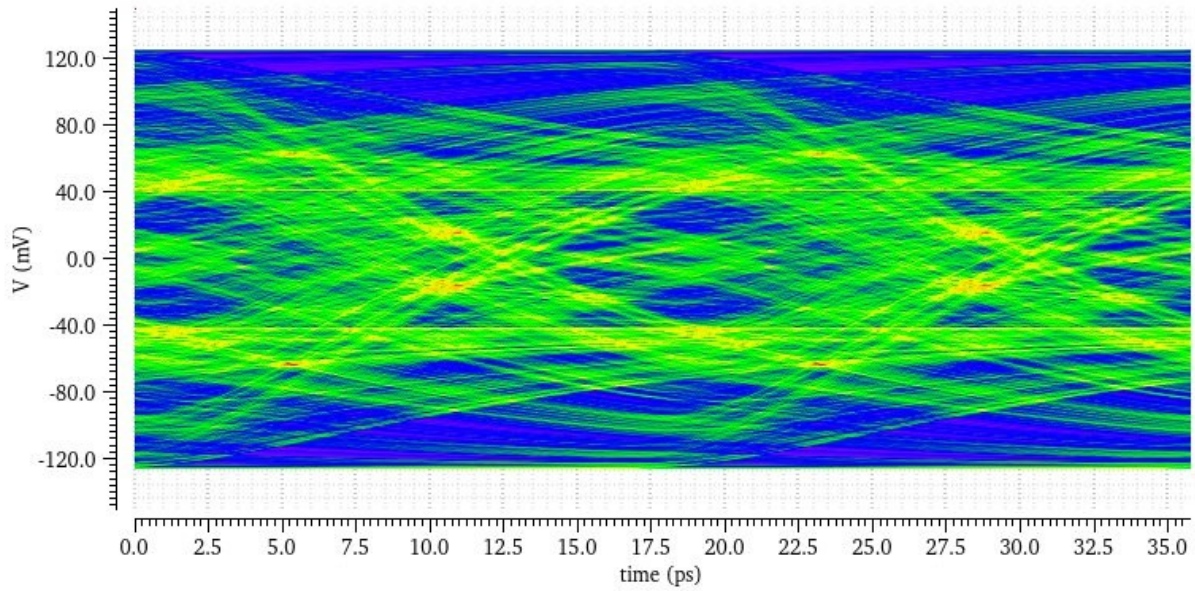
where $A_0$is the DC gain factor, ω_z is the compensating zero frequency, ω_{p1} is the dominant pole frequency, and ω_{p2} is the high-frequency parasitic pole frequency.

To equalize a **10 dB** Nyquist channel attenuation at **28 GHz**, the CTLE parameters are defined as follows:

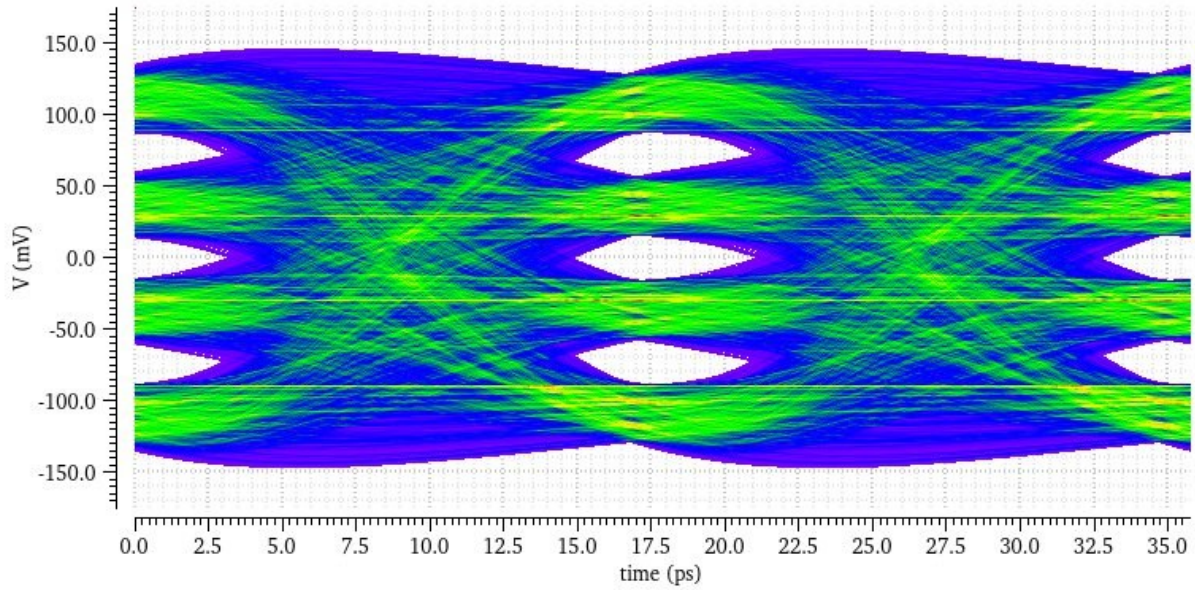
- Compensating Zero: $f_z = \frac{\omega_z}{2\pi} = \mathbf{8.8}$ GHz
- First Dominant Pole: $f_{p1} = \frac{\omega_{p1}}{2\pi} = \mathbf{28.0}$ GHz
- Second Parasitic Pole: $f_{p2} = \frac{\omega_{p2}}{2\pi} = \mathbf{56.0}$ GHz
- **DC Common-Mode Maintenance**: The module maintains an exact **0.0 V** input and output DC common-mode level ($V_{CM,in} = V_{CM,out}$ = **0.0 V**), preventing DC offset drift from saturating subsequent amplification stages [7].



(a)



(b)



(c)

Figure 2.2: (a) Channel and CTLE Frequency Response, (b) 56 GBaud/s PAM4 XSR Channel without Equalization, (c) 56 GBaud/s PAM4 XSR Channel with CTLE.

2.2.2 Variable Gain Amplifier (VGA)

After the CTLE, the Variable Gain Amplifier (VGA) compresses the dynamic range and regulates signal amplitude across varying channel-loss profiles.

The VGA behavioral model has three key design characteristics:

1. **Wide Gain Adjustment Range**: Programmable gain ranges from -6 dB to +14 dB (0.50 V/V to 5.01 V/V), enabling the receiver to handle both over- and under-equalized input waveforms.
2. **High-Bandwidth Low-Pass Response**: A wide -3 dB bandwidth of $f_{-3dB} \geq$ $\mathbf{40.0}\ \boldsymbol{GHz}$ modeled via a first-order Butterworth transfer function to prevent high-frequency inter-symbol interference (ISI) degradation:

$$H_{\mathrm{VGA}}(s) = \frac{A_{\mathrm{VGA}}}{1 + \frac{s}{\omega_{-3db}}}$$

DC Common-Mode Level Shifting: Converts the input DC common-mode voltage from 0.0 V (from the CTLE) to an output DC common-mode level of 0.5 V. This 0.5 V bias optimizes the transconductance (g_m) and decision speed of the dynamic latch comparators in advanced FinFET nodes.

3. **Target Signal Amplitude**: An Automatic Gain Control (AGC) feedback loop scales the output signal to a target peak differential swing of ±150 mV (a total differential envelope of ±300 mVpp,diff), establishing fixed, predictable eye heights for the subsequent 7-slicer threshold ladder.

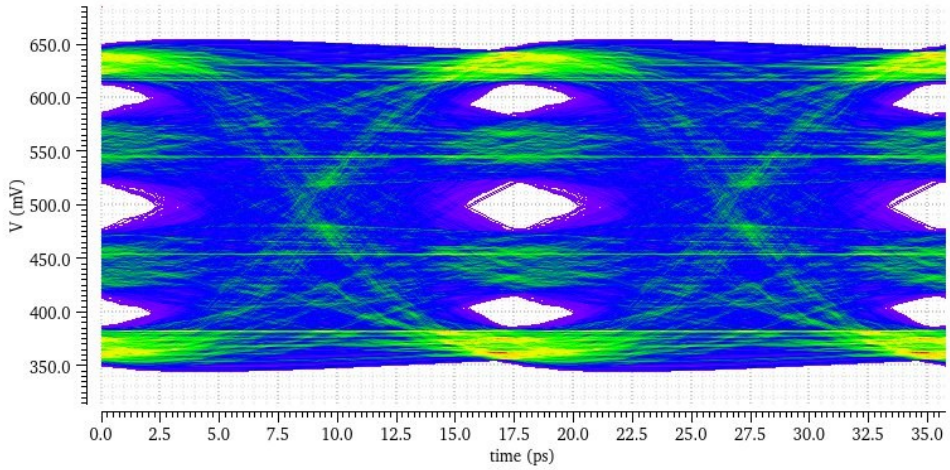


Figure 2.3: Eye Diagram of the output Signal from the VGA.

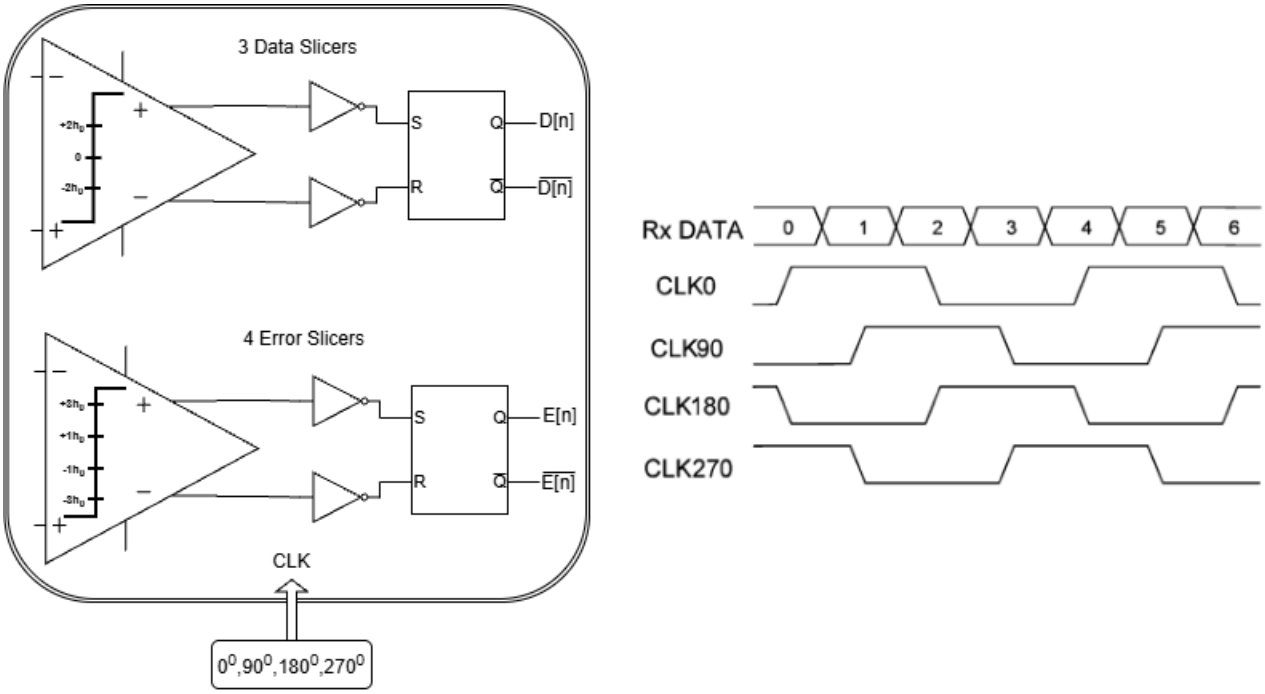


Figure 2.4: Data and Error Samplers and their timing diagram.

2.2.3 7–Slicer Array

To digitize the equalized PAM-4 waveform into data decisions and error-polarity signals, a 7-slicer array is placed at the AFE output. It comprises three data slicers and four error slicers

operating in parallel, enabling baud-rate phase detection and level adaptation without a power-hungry analog-to-digital converter (ADC).

Table 2.1: Clock Arrangement of the samplers.

D[n−1]	D[n]/ E[n]	D[n+1]	PD(Early/Late)
90^0	180^0	270^0	0^0
180^0	270^0	0^0	90^0
270^0	0^0	90^0	180^0
0^0	90^0	180^0	270^0

The voltage threshold assignments are defined based on the nominal main cursor amplitude h_0= **50mV**:

3 Data Slicers:

- **Top Data Slicer** (V_H):
  Threshold set at $+2h_0$ = **+600 mV** (discriminates level +3 from +1).
- **Middle Data Slicer** (V_M):
  Threshold set at **+500 mV** (discriminates positive levels +3, +1 from negative levels −1, −3).
- **Bottom Data Slicer** (V_L):
  Threshold set at $-2h_0$= **+400 mV** (discriminates level −1 from −3).

4 Error Slicers:

- **Error Slicer 4**:
  Threshold set at $+3h_0$= **+650 mV** (evaluates upper eye error polarity E_{+3}).
- **Error Slicer 3**:
  Threshold set at $+1h_0$= **+550 mV** (evaluates upper-middle eye error polarity E_{+1}).
- **Error Slicer 2**:

Threshold set at $-1h_0$= **+450 mV** (evaluates lower-middle eye error polarity E_{-1}).

- **Error Slicer 1**:

  Threshold set at $-3h_0$ = **+350 mV** (evaluates lower eye error polarity E_{-3}).

Each slicer is modeled as a high-speed dynamic latch comparator that operates at a quarter-rate and is time-interleaved across four phase channels at 14 GHz, resolving binary decisions within a T_{UI}= 17.857 ps decision window.

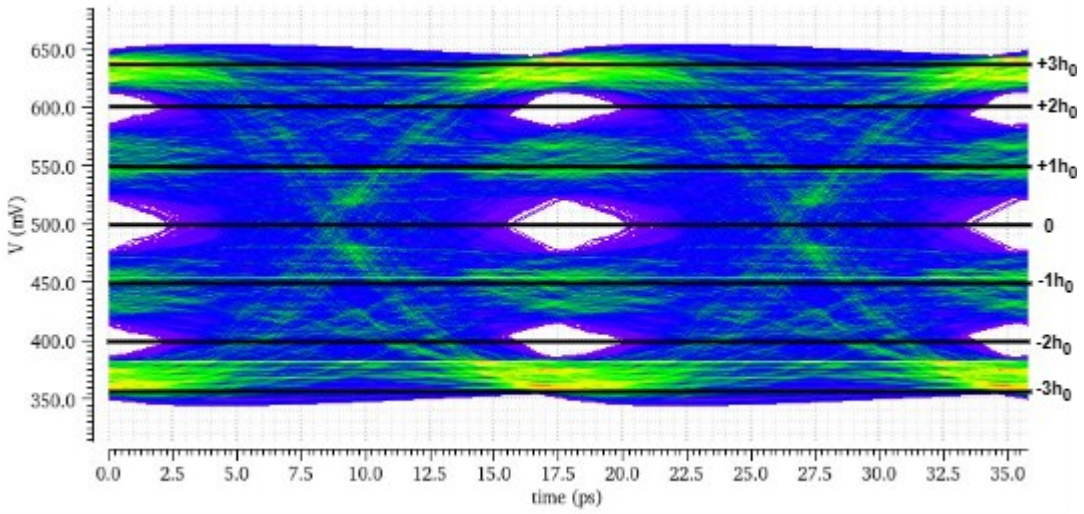


Figure 2.5: Eye Diagram of the input signal and the voltage references.

2.3 Digital CDR Loop Modeling (Verilog-A)

The digital timing recovery loop uses the quantized 7-bit thermometer codes from the AFE slicer array to align the recovered DCO clock phase with the center of the eye opening.

2.3.1 Unbalanced Mueller-Müller Phase Detector

Conventional Mueller-Müller phase detectors (MMPDs) estimate phase timing errors at the baud rate (one sample per UI) using two consecutive data samples, D[n−1] and D[n], along with the error sample E[n] [14]. However, in PAM-4 systems, standard sign-sign MMPDs suffer from low transition density ($\leq$ 12.5%).

To address these limitations, an unbalanced three-symbol pattern Mueller-Müller phase detector is implemented [6]. It evaluates three consecutive data decisions, D[n−1], D[n], and D[n+1], where each decision belongs to {-3, -1, +1, +3}. and classifies all $4^3 = 64$ potential PAM-4 triplet patterns into three distinct categories:

1. **Category 1**: **12 Symmetric Patterns** (**Weight W = 1.0**): Major zero-crossing transitions (e.g., (-3, +3, +3) or (+1, -1, -1)) where post-cursor ISI and pre-cursor ISI cancel symmetrically at the center symbol.

Table 2.2: Symmetric Data Patterns [1].

D[n-1]	D[n]	D[n+1]	E[n]	PD
+3	+3	-3	+ / -	Early / Late
-3	+3	+3	+ / -	Late / Early
+3	+1	-3	+ / -	Early / Late
+1	+1	-1	+ / -	Early / Late
-1	+1	+1	+ / -	Late / Early
-3	+1	+3	+ / -	Late / Early
+3	-1	-3	+ / -	Early / Late
+1	-1	-1	+ / -	Early / Late
-1	-1	+1	+ / -	Late / Early
-3	-1	+3	+ / -	Late / Early
+3	-3	-3	+ / -	Early / Late
-3	-3	+3	+ / -	Late / Early

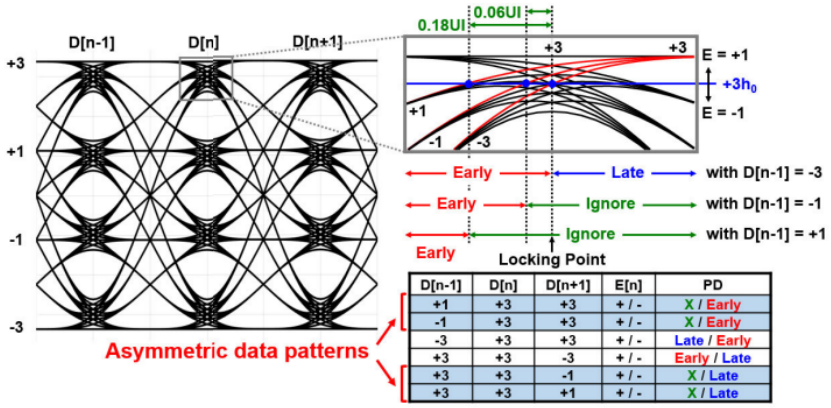
2. **Category 2**: **16 Asymmetric Monotonic Patterns** (**Weight W = 0.5**): Monotonic transitions (e.g., (-1, +3, +3), (+1, +3, +3), or (-3, -3, -1)) that exhibit asymmetric eye boundaries. Assigning a scaled confidence weight of W = 0.5 compensates for asymmetric ISI, boosting the active transition density to α_{unbal} = **31.25%** (20/64).

Table 2.3: Asymmetric Data Patterns [1].

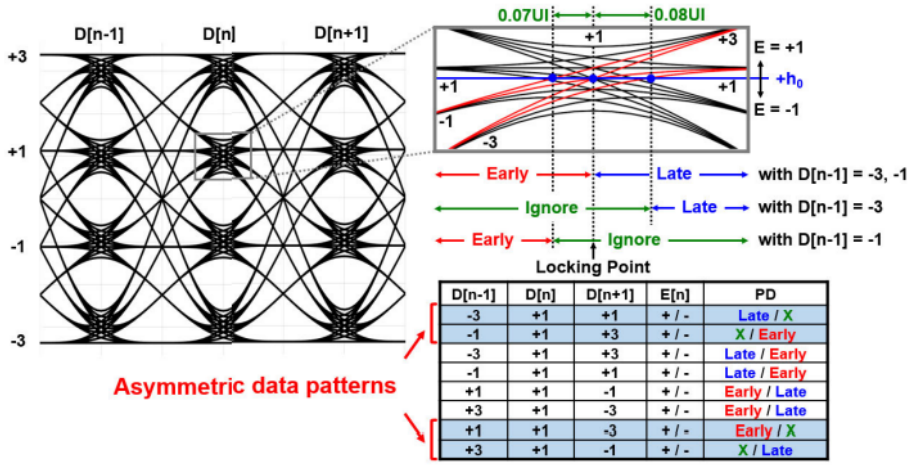
D[n-1]	D[n]	D[n+1]	E[n]	PD
+3	+3	+1	+ / -	X / Late
+3	+3	-1	+ / -	X / Late
+1	+3	+3	+ / -	X / Early
-1	+3	+3	+ / -	X / Early
+3	+1	-1	+ / -	X / Late
+1	+1	-3	+ / -	Early / X
-1	+1	+3	+ / -	X / Early
-3	+1	+1	+ / -	Late / X
+3	-1	-1	+ / -	X / Late
+1	-1	-3	+ / -	Early / X
-1	-1	+3	+ / -	X / Early
-3	-1	+1	+ / -	Late / X
+1	-3	-3	+ / -	Early / X
-1	-3	-3	+ / -	Early / X
-3	-3	+1	+ / -	Late / X
-3	-3	-1	+ / -	Late / X

3. **Category 3: 36 Non-Monotonic Inner Bounces (Weight W = 0.0 Hold)**: Non-monotonic patterns (e.g., (+1, −1, +1) or (−3, +1, −3)) that produce ambiguous, multi-valued zero crossings. These transitions are masked (W = 0.0), forcing a zero-phase update ($e_{PD}[n] = 0$).

By filtering non-monotonic transitions, the unbalanced MMPD reduces digital switching activity and loop-processing latency, while cutting clock distribution power by 50% compared with 2× oversampled Bang-Bang Phase Detectors.



(a)



D[n-1]	D[n]	D[n+1]	E[n]	PD
-3	+1	+1	+ / -	Late / X
-1	+1	+3	+ / -	X / Early
-3	+1	+3	+ / -	Late / Early
-1	+1	+1	+ / -	Late / Early
+1	+1	-1	+ / -	Early / Late
+3	+1	-3	+ / -	Early / Late
+1	+1	-3	+ / -	Early / X
+3	+1	-1	+ / -	X / Late

Figure 2.6: Operational principle of the unbalanced MMPD. (a) D[n] = +3 and (b) D[n] = +1[1].

2.3.2 Spatial Majority Voter Module

To bridge the high-speed front-end sampling logic and digital loop processing, a majority-voting block is placed immediately after the quarter-rate Unbalanced Mueller-Müller Phase Detector. The quarter-rate MM-PD evaluates phase alignment across four parallel lanes, generating four Early and four Late binary flags corresponding to the quadrature clock phases: $\Phi 0°$, $\Phi 90°$, $\Phi 180°$, and $\Phi 270°$. Driving the loop filter directly with eight uncoordinated flags could cause high-frequency phase chatter and destabilize the loop because of inter-lane skew or localized channel noise. The majority-voting block prevents this by combining all eight phase-error outputs into a single phase-correction decision for each quarter-rate clock cycle.

The Verilog-A code performs this aggregation on every positive edge of the primary 14 GHz clock. During each evaluation event, the block reads the voltage states of the eight Early and Late inputs across all four quadrature phases and converts them into active Boolean assertions. It maintains two separate internal accumulation counts: one that sums active Early flags across $0°, 90°, 180°,$ and $270°$, and another summing active Late flags across the same phases. By

evaluating all four lanes concurrently, an incorrect phase detection on one lane caused by localized channel attenuation or thermal noise is filtered out through consensus among the other lanes, suppressing random decision jitter before it reaches the loop filter.

The block processes these accumulated totals through one of tri-state configurable operating modes set in the code.

1. **Net Late** / **Step UP** (**+1.0 Output**): If sum_late > sum_early, indicating that late phase decisions dominate across the quarter-rate channels, the voter outputs a positive directional step: decision = 1.0.
2. **Net Early** / **Step DOWN** (**–1.0 Output**): If sum_late < sum_early, indicating that early phase decisions dominate, the voter outputs a negative directional step: decision = –1.0.
3. **Balanced** / **HOLD** (**0.0 Output**): If sum_late = sum_early (such as equal early and late votes, or zero-vote hold conditions), the voter outputs a zero phase error signal: decision = 0.0.

2.3.3 2nd-Order Digital Loop Filter (DLF)

The Verilog-A behavioral model of the Digital Loop Filter (DLF) implements a dual-path control mechanism to maintain loop stability and enable fast phase tracking in a high-speed clock and data recovery (CDR) system. To meet the strict loop-latency requirements of baud-rate architectures, the proportional path bypasses accumulated digital logic and updates directly from the Majority Voter (MV) decisions each clock cycle. This direct update lets the CDR respond immediately to high-frequency phase jitter without the multi-cycle latency that could degrade loop phase margins. Meanwhile, the integral path continuously accumulates phase-error trends to eliminate static phase error and compensate for low-frequency oscillator frequency offsets. Modeling both

paths behaviorally in Verilog-A enables fast co-simulation of loop bandwidth, damping factor, and initial acquisition time before full transistor-level synthesis.

When mapped to a 12nm FinFET technology node, the DLF is implemented using synthesizable digital standard-cell logic at a nominal supply voltage (typically ~0.8 V). Operating the integral accumulator directly at the 14 GHz quarter-rate clock frequency would require standard digital cells to switch within an extremely short clock period of ~71.4 ps. In a 12nm FinFET library, running multi-bit adder and accumulation logic at 14 GHz leads to severe setup and hold timing violations, as well as excessive dynamic power dissipation from high switching activity. To overcome this timing constraint, the integral path incorporates a 2:16 deserializer. Downsampling the phase detector decisions by a factor of 16 reduces the integral accumulator's effective clock rate from 14 GHz to ~875 MHz. This lower operating frequency enables the 63-step thermometer control logic to comfortably achieve timing closure in standard 12nm FinFET digital cells while significantly reducing core power consumption.

The dynamic behavior is governed by the discrete-time difference equations:

$$\boldsymbol{y_p[n]} \;=\; \boldsymbol{K_p} \;*\; \boldsymbol{e_{PD}[n]}$$

$$\boldsymbol{y_i[n]} \;=\; \boldsymbol{y_i[n-1]} \;+\; \boldsymbol{K_i} \;*\; \boldsymbol{e_{PD}[n]}$$

$$\boldsymbol{D_{ctrl,ideal}[n]} \;=\; \boldsymbol{y_p[n]} \;+\; \boldsymbol{y_i[n]}$$

where $\boldsymbol{e_{PD}[n]}$ is the decimated phase error, $\boldsymbol{K_p}$ is the proportional path gain, $\boldsymbol{K_i}$ is the integral path gain, and $\boldsymbol{D_{ctrl,ideal}[n]}$ is the unquantized control word.

Choosing K_p and K_i requires balancing target loop bandwidth, stability margins, and the transport delay introduced by the downsampled integral path. The proportional gain K_p is selected first to establish the open-loop crossover frequency ($f_{\mathrm{BW}} \approx K_{\mathrm{PD}}K_p/2\pi$), typically targeted between $f_{\mathrm{baud}}/1000$ and $f_{\mathrm{baud}}/100$ (56–100 MHz for a 56 GBaud link) to track high-frequency jitter

without introducing excessive jitter peaking. The integral gain K_i is then chosen to place the stabilizing zero ($f_z = \frac{K_i}{2\pi \cdot 16 \cdot K_p \cdot T_{\text{clk}}}$) well below the crossover frequency ($f_z \leq f_{\text{BW}}/4$), enabling the loop to eliminate static phase error and low-frequency frequency offsets. Finally, to account for the phase lag added by the 2:16 deserializer accumulation delay, the ratio K_i/K_p is restricted to ≤ 0.1, maintaining a damping factor $\zeta \geq 0.707$ and a phase margin $> 60°$ to prevent limit-cycle oscillations.

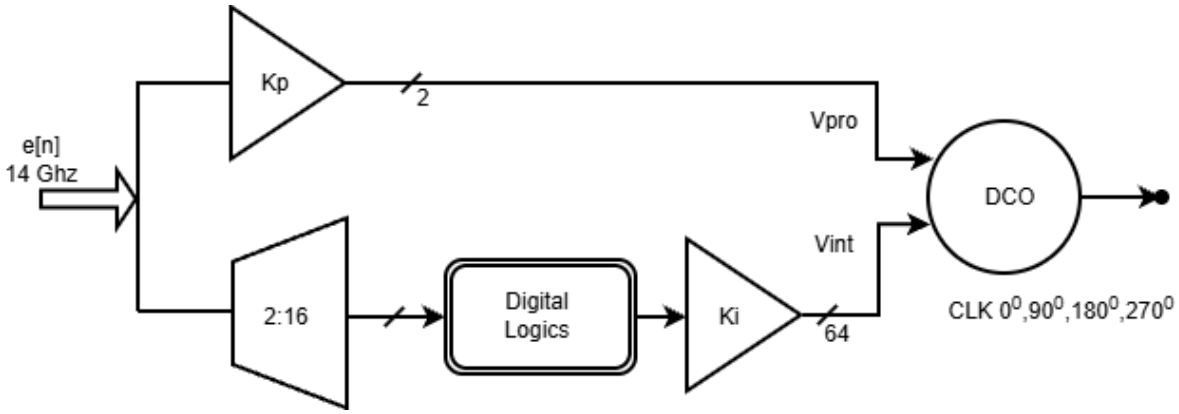


Figure 2.7: Digital Loop Filter and DCO.

2.3.4 Digitally Controlled Oscillator (DCO)

The Verilog-A model of the Digitally Controlled Oscillator (DCO) represents a 4-phase quadrature ring oscillator with a nominal frequency of 14 GHz, meeting the quarter-rate clocking requirement for a 112 Gbps PAM-4 link (56 GBaud). The behavioral code uses explicit gain parameters to convert continuous proportional control-voltage inputs and multi-bit integral digital codes into instantaneous frequency shifts. ($K_{\text{DCO,P}}$ and $K_{\text{DCO,I}}$). By mathematically integrating the instantaneous frequency over time, the model computes continuous phase accumulation and produces four quadrature clock signals ($0°, 90°, 180°, 270°$). This macro-model allows transient verification of CDR phase alignment, quadrature accuracy, and jitter tolerance without lengthy transistor-level transient simulations.

In 12nm FinFET technology, the DCO uses a multi-stage differential ring oscillator with inverter-based delay cells controlled by digital capacitive varactor banks and tail-current-steering DACs. The high transconductance (g_m) and reduced device footprint of 12nm FinFET devices allow ring oscillators to reach 14 GHz operating speeds efficiently without requiring large, area-intensive passive LC inductors. However, FinFET devices are more sensitive to process, voltage, and temperature (PVT) variations and flicker noise because of 3D gate confinement. To ensure exact quadrature phase matching among the four clock channels within a narrow 1 UI window of 17.86 ps, the 12nm mask layout uses symmetrical floorplanning and matched device geometries, preventing parasitic interconnect mismatches from compromising phase margin.

The instantaneous DCO output frequency $f_{\mathrm{DCO}}(t)$ is expressed as:

$$f_{\mathrm{DCO}}(t) = f_0 + K_{\mathrm{DCO,P}} \cdot V_{\mathrm{pro}}(t) + K_{\mathrm{DCO,I}} \cdot V_{\mathrm{int}}(t)$$

Calculates the total output frequency by adding two control offsets to the nominal center frequency ($f_0 = 14$ GHz). The proportional input (V_{pro}) introduces rapid, cycle-by-cycle frequency shifts to correct sudden phase jitter, while the integral input (V_{int}) adjusts the baseline frequency to compensate for static frequency offsets between the transmitter and receiver.

Phase Integration with Jitter and Phase Noise

The total accumulated phase $\Phi_{\mathrm{DCO}}(t)$ includes continuous frequency integration alongside physical phase noise $\phi_n(t)$ and deterministic jitter components $\theta_{\mathrm{det}}(t)$:

$$\Phi_{\mathrm{DCO}}(t) = 2\pi \int_0^t f_{\mathrm{DCO}}(\tau)\, d\tau + \phi_n(t) + \theta_{\mathrm{det}}(t)$$

- Random Phase Noise $\phi_n(t)$: Modeled as a Brownian motion (Wiener) process driven by zero-mean Gaussian white frequency noise, with variance proportional to cycle-to-cycle period jitter. ($\sigma_{J,\mathrm{per}}$):

$$\Delta\phi_n(T_{\text{clk}}) = 2\pi \cdot f_0 \cdot \Delta t_{\text{jitter}}$$

$$\Delta t_{\text{jitter}} \sim \mathcal{N}\left(0, \sigma^2_{J,\text{per}}\right)$$

- Single-Sideband Phase Noise Spectrum $L(f)$: Relates to accumulated phase variance at a frequency offset Δf by:

$$L(\Delta f) \approx 10 \log_{10}\left(\frac{f_0^2 \cdot \sigma^2_{J,\text{per}}}{2 \cdot (\Delta f)^2}\right) \quad [\text{dBc/Hz}]$$

Quadrature Output Signals

The four phase-shifted clock waveforms ($\theta \in \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$) incorporate phase noise directly into the sinusoidal phase argument:

$$v_{\text{clk},\theta}(t) = V_{\text{CM}} + V_{\text{amp}} \cdot \sin(\Phi_{\text{DCO}}(t) - \theta) + n_{\text{amp}}(t)$$

where V_{CM} is the common-mode DC offset, V_{amp} is the single-ended clock amplitude, and $n_{\text{amp}}(t)$ models amplitude thermal noise.

2.3.5 Data Recovery (DR) Path

The PAM-4 decoder maps the sampled four-level analog signal into a two-bit digital binary output consisting of a Most Significant Bit (MSB) and a Least Significant Bit (LSB). The decision circuit uses three parallel voltage comparators (slicers) clocked by the sampling clock phase CK_{D0}. These slicers compare the incoming continuous-time signal with three predefined voltage reference thresholds, V_{H} (high threshold), V_{M} (middle threshold), and V_{L} (low threshold) which divide the PAM-4 eye diagram into four distinct amplitude regions corresponding to the $+3$, $+1$, -1, and -3 symbol levels.

Upon sampling at the active clock edge of CK_{D0}, the three comparators output a 3-bit thermometer code ($D_{\text{H0}}, D_{\text{M0}}, D_{\text{L0}}$). The middle threshold slicer output, D_{M0}, directly determines the decoded MSB, indicating whether the input voltage falls within the positive symbol range

($D_{M0} = 1$) or the negative symbol range ($D_{M0} = 0$). Because V_M sits at the zero-crossing center of the PAM-4 signal eye, this direct assignment provides a low-latency decision for the primary bit polarity.

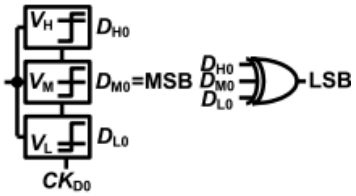


Figure 2.8: PAM-4 Decoder [19].

To resolve the LSB, a 3-input XOR gate processes all three comparator outputs according to the Boolean relation LSB $= D_{H0} \oplus D_{M0} \oplus D_{L0}$. This combinational parity logic differentiates the inner symbol levels $(+1, -1)$ from the outer symbol levels $(+3, -3)$. The resulting feed-forward logic reduces propagation delay and power consumption, enabling the deterministic binary decoding required by ultra-high-speed receiver architectures.

Table 2.4: PAM-4 Decoder Truth Table.

PAM-4 Symbol Voltage Level	**Input Range**	**DH0 (V>VH)**	**DM0 (V>VM)**	**DL0 (V>VL)**	**MSB (DM0)**	**LSB (DH0⊕DM0⊕DL0)**	**Decoded Word (MSB LSB)**
+3 Level	$V_{in} > V_H$	**1**	**1**	**1**	**1**	$1 \oplus 1 \oplus 1 = 1$	**11**
+1 Level	$V_M < V_{in} < V_H$	**0**	**1**	**1**	**1**	$0 \oplus 1 \oplus 1 = 0$	**10**
−1 Level	$V_L < V_{in} < V_M$	**0**	**0**	**1**	**0**	$0 \oplus 0 \oplus 1 = 1$	**01**
−3 Level	$V_{in} < V_L$	**0**	**0**	**0**	**0**	$0 \oplus 0 \oplus 0 = 0$	**00**

3. Discrete-Time Z-Domain Modeling & Loop Stability

3.1 Discrete Z-Domain System Formulation

High-speed all-digital clock and data recovery (ADCDR) architectures operating on 112-Gb/s (56-GBaud) PAM-4 signals are fundamentally sampled-data control systems [2],[20]. Unlike continuous-time analog charge-pump CDRs (CP-CDRs), in which passive RC networks continuously smooth phase error, an ADCDR processes discrete phase-error samples at sub-rate clock intervals [21],[22]. Accurately predicting closed-loop transient response, jitter transfer characteristics, and stability boundaries therefore requires formulating the complete feedback loop in the discrete Z-domain [5].

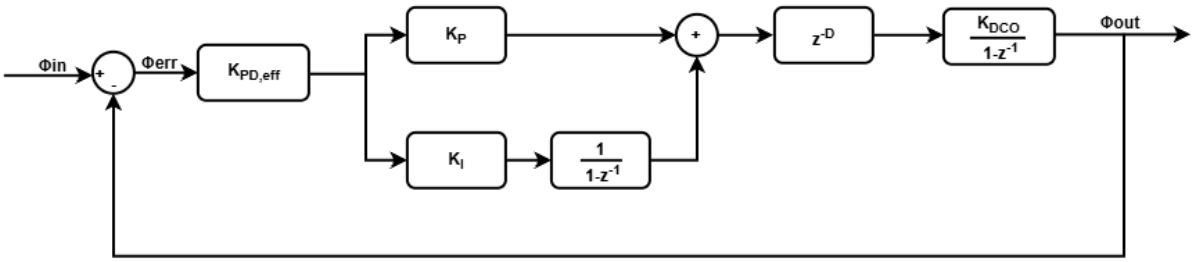


Figure 3.1: The z-domain linearized model of CDR.

3.1.1 Linearized Phase-Domain System Topology

The linearized discrete-time phase-domain model for the 112-Gb/s quarter-rate ADCDR is shown above. The input signal phase is denoted as $\Phi_{in}(z)$, the recovered clock phase is $\Phi_{out}(z)$, and the instantaneous phase error is defined as:

$$\Phi_{err}(z) = \Phi_{in}(z) - \Phi_{out}(z)$$

where z^{-D} models the total feedback transport latency introduced by time-interleaved sampling, spatial voting, and digital retiming registers.

3.1.2 Z-Domain Sub-Block Transfer Functions

The loop consists of three main functional Z-domain transfer blocks:

1. Effective Phase Detector Gain ($K_{PD,eff}$) Extraction and S-Curve Analysis

The Phase Detector Gain, K_{PD}, quantifies the small-signal sensitivity of the Clock and Data Recovery (CDR) phase detector near its stable tracking lock point. Mathematically, it is defined as the linearized slope of the phase detector output characteristic curve $S(\phi)$ evaluated at the nominal zero phase error condition ($\phi = 0$ UI):

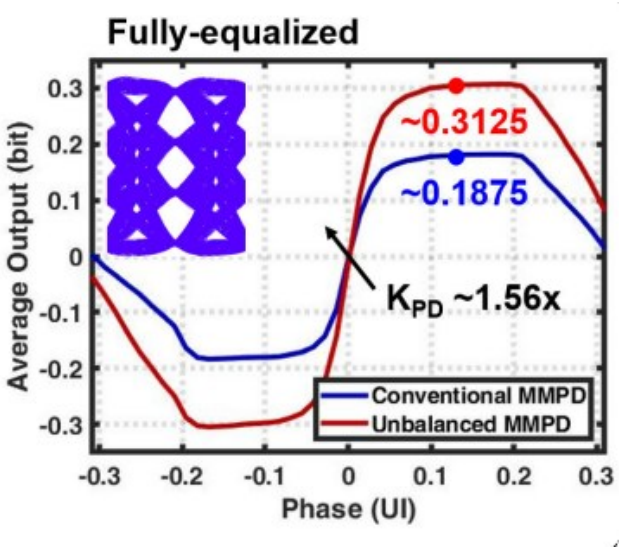


Figure 3.2: Phase Detector S-Curve Characteristic Comparison [1].

where ϕ represents the phase offset normalized to the Unit Interval (UI), and $S(\phi) = \mathbb{E}[e_k(\phi)]$ represents the ensemble average (expected value) of the error signal e_k generated per symbol period. In a discrete-time ADC-based architecture, $S(\phi)$ is expressed in digital resolution units (bits or LSBs) [1]:

$$K_{PD} = \left.\frac{dS(\phi)}{d\phi}\right|_{\phi=0}$$

Extraction Methodology

When extracting K_{PD} from empirical or simulated S-curve characteristics, a linear small-signal approximation is applied across a symmetric phase window $[-\Delta\phi, +\Delta\phi]$ centered at the lock point ($\phi_0 = 0$):

$$K_{PD} \approx \frac{S(+\Delta\phi) - S(-\Delta\phi)}{2 \cdot \Delta\phi}$$

To ensure accurate linear slope estimation without saturation distortion, the extraction range is constrained to the linear region of the S-curve within $\Delta\phi = \pm 0.03$ UI.

Quantitative Comparison: Conventional vs. Unbalanced MMPD

A. Conventional Mueller–Muller Phase Detector (MMPD)

For a fully-equalized PAM-4 signal processed by a standard MMPD, evaluating the response at the boundary conditions $\phi = \pm 0.03$ UI yields:

$$S_{\text{conv}}(+0.03 \text{ UI}) \approx +0.15 \text{ bit}$$

$$S_{\text{conv}}(-0.03 \text{ UI}) \approx -0.15 \text{ bit}$$

Applying the central difference approximation gives:

$$K_{PD,\text{conv}} = \frac{0.15 \text{ bit} - (-0.15 \text{ bit})}{0.03 \text{ UI} - (-0.03 \text{ UI})} = \frac{0.30 \text{ bit}}{0.06 \text{ UI}} = \mathbf{5.00} \text{ bit/UI}$$

Expressed in continuous-time phase units (rad), where $1 \text{ UI} = 2\pi$ rad:

$$K_{PD,\text{conv, rad}} = \frac{5.00 \text{ bit/UI}}{2\pi \text{ rad/UI}} \approx \mathbf{0.796} \text{ bit/rad}$$

B. Unbalanced Mueller–Muller Phase Detector

Under identical equalized channel conditions, the Unbalanced MMPD selectively weights high-confidence symbol transitions (inner transitions ± 1) while suppressing noisy

outer transitions (± 3). At the same phase offsets ($\phi = \pm 0.03$ UI), the error response expands to:

$$S_{\text{unbal}}(+0.03\ \text{UI}) \approx +0.234\ \text{bit}$$

$$S_{\text{unbal}}(-0.03\ \text{UI}) \approx -0.234\ \text{bit}$$

Evaluating the linearized gain gives:

$$K_{PD,\text{unbal}} = \frac{0.234\ \text{bit} - (-0.234\ \text{bit})}{0.03\ \text{UI} - (-0.03\ \text{UI})} = \frac{0.468\ \text{bit}}{0.06\ \text{UI}} = \mathbf{7.80}\ \text{bit/UI}$$

Converting to radian phase units:

$$K_{PD,\text{unbal, rad}} = \frac{7.80\ \text{bit/UI}}{2\pi\ \text{rad/UI}} \approx \mathbf{1.241}\ \text{bit/rad}$$

C. Relative Gain Enhancement Ratio

The relative improvement factor $\gamma_{K_{PD}}$ achieved by the unbalanced modification is calculated as:

$$\gamma_{K_{PD}} = \frac{K_{PD,\text{unbal}}}{K_{PD,\text{conv}}} = \frac{7.80\ \text{bit/UI}}{5.00\ \text{bit/UI}} = \mathbf{1.56}\times$$

The Unbalanced Mueller–Müller Phase Detector (MMPD) evaluates equalized symbol samples at the quarter-rate clock frequency (f_{clk} = 14.0 GHz, T_{clk} = 71.429 ps). The linearized Z-domain representation of the phase detector and voter cascade is expressed as a constant scalar gain [15]:

$$H_{PD}(z) = K_{PD,eff} = K_{PD,unbal} \cdot K_{voter}$$

$$K_{voter} = \sqrt{\frac{2}{\pi}\frac{1}{\sigma_A}}$$

$$\sigma_A = \sqrt{N} \cdot \sqrt{\alpha_T}$$

where $K_{PD,unbal}$ = 1.241 LSB/rad represents the small-signal slope of the Unbalanced MMPD S-curve, σ_A the standard deviation of the accumulated phase detector outputs, α_T is the transition density, and $N = 4$ phase detector outputs are combined per majority voting cycle. Substituting these values yields:

$$K_{voter} = \sqrt{\frac{2}{\pi \cdot 4 \cdot 0.3125}} = \sqrt{\frac{1.6}{\pi}} \approx 0.7136$$

$$K_{PD,eff} = 1.241 \times 0.7136 \approx 0.886\, LSB/rad$$

2. Proportional-Integral Digital Loop Filter ($\boldsymbol{H_{DLF}(z)}$): The synthesizable second-order Digital Loop Filter (DLF) processes the decimated phase-error signal $E_{PD}(z)$ through parallel proportional and discrete-time integral accumulation paths [15]. The discrete transfer function is formulated as:

$$H_{DLF}(z) = \alpha + \frac{\beta}{1 - z^{-1}} = \frac{(\alpha + \beta) - \alpha \cdot z^{-1}}{1 - z^{-1}}$$

where $\boldsymbol{\alpha}$ represents the proportional gain coefficient (K_p), and $\boldsymbol{\beta}$ represents the integral accumulator gain coefficient (K_i). Factoring out $\boldsymbol{\alpha}$, the DLF transfer function can be expressed in terms of the stabilizing zero location:

$$H_{DLF}(z) = \alpha \left[\frac{1 + \frac{\beta}{\alpha} - z^{-1}}{1 - z^{-1}} \right]$$

3. Digitally Controlled Oscillator ($\boldsymbol{H_{DCO}(z)}$): The 14-GHz LC-DCO acts as a discrete phase integrator, controlled by the digital tuning word $H_{DLF}(z)$ [15]. The Z-domain transfer function mapping digital control codes to output phase is:

$$\mathrm{H_{DCO}(z)} = \frac{\mathrm{K_{DCO_Hz}} \cdot \mathrm{T_{clk}}}{1 - \mathrm{z^{-1}}} = \frac{\mathrm{K_{DCO}}}{1 - \mathrm{z^{-1}}}$$

where K_{DCO_Hz} = 500 MHz/LSB is the physical frequency tuning sensitivity, T_{clk} = 71.429 ps is the sampling period, and $K_{DCO} = 2\pi \cdot K_{DCO_Hz} \cdot T_{clk}$ = 0.2244 rad/LSB represents the normalized DCO phase gain per control step.

3.1.3 Closed–Loop Phase and Error Transfer Function Derivation

The undelayed open-loop transfer function $T_0(z)$ is obtained by taking the product of the individual feedforward transfer blocks:

$$T_0(z) = H_{\mathrm{PD}}(z) \cdot H_{\mathrm{DLF}}(z) \cdot H_{\mathrm{DCO}}(z) = K_{\mathrm{PD,eff}} \cdot \left[\frac{(\alpha+\beta) - \alpha z^{-1}}{1 - z^{-1}}\right] \cdot \left[\frac{K_{\mathrm{DCO}}}{1 - z^{-1}}\right]$$

$$T_0(z) = \frac{K_{\mathrm{PD,eff}} \cdot K_{\mathrm{DCO}} \cdot [(\alpha+\beta) - \alpha z^{-1}]}{(1 - z^{-1})^2}$$

Incorporating the transport latency $z^{-\mathrm{D}}$, the total open-loop gain function T(z) becomes:

$$T(z) = T_0(z) \cdot z^{-D} = \frac{K_{\mathrm{PD,eff}} \cdot K_{\mathrm{DCO}} \cdot [(\alpha+\beta) - \alpha z^{-1}] \cdot z^{-\mathrm{D}}}{(1 - z^{-1})^2}$$

The Closed-Loop Phase Transfer Function $H(z) = \frac{\Phi_{\mathbf{out}}(z)}{\Phi_{\mathbf{in}}(z)}$ is derived as:

$$H(z) = \frac{T(z)}{1 + T(z)} = \frac{K_{\mathrm{PD,eff}} \cdot K_{\mathrm{DCO}} \cdot [(\alpha+\beta)z^2 - \alpha z]}{z^{\mathrm{D}} \cdot (z-1)^2 + K_{\mathrm{PD,eff}} \cdot K_{\mathrm{DCO}} \cdot [(\alpha+\beta)z^2 - \alpha z]}$$

Similarly, the Closed-Loop Phase Error Transfer Function $E(z) = \frac{\Phi_{\mathbf{err}}(z)}{\Phi_{\mathbf{in}}(z)}$characterizing clock jitter rejection is:

$$E(z) = \frac{1}{1 + T(z)} = \frac{z^{D} \cdot (z-1)^2}{z^{\mathrm{D}} \cdot (z-1)^2 + K_{\mathrm{PD,eff}} \cdot K_{\mathrm{DCO}} \cdot [(\alpha+\beta)z^2 - \alpha z]}$$

3.2 Pipeline Latency (D) Impact

In high-speed 12-nm FinFET transceivers, multi-phase demultiplexing, spatial majority voting, and digital retiming registers add unavoidable transport delays to the feedback path [23].

3.2.1 Characteristic Equation with Feedback Latency

The roots of its characteristic polynomial govern the dynamic stability of the ADCDR:

$$1 + T(z) = 0 \Longrightarrow 1 + \frac{K_{\mathrm{PD,eff}} K_{\mathrm{DCO}} \cdot [(\alpha + \beta)z^2 - \alpha \mathrm{z}]z^{-D}}{(z-1)^2} = 0$$

Multiplying through by $z^{D+2}(1 - z^{-1})^2$, the characteristic equation expands to:

$$\mathrm{P(z)} = \mathrm{z^D(z-1)^2} + \mathrm{K_{PD,eff} K_{DCO}}[(\alpha + \beta)z^2 - \alpha \mathrm{z}] = 0$$

Without pipeline latency ($\mathrm{D} = 0$), $P(z)$ is a quadratic polynomial possessing exactly 2 closed-loop poles. However, when feedback latency $\mathrm{D} > 0$ is present, the degree of the polynomial increases to $\mathrm{D} + 2$, introducing D additional parasitic poles into the system [23].

3.2.2 Pipeline Latency Breakdown

The total loop latency in the proportional feedback path (τ_{loop}) consists of the cumulative delay across four distinct digital processing stages: the Mueller-Müller Phase Detector (MMPD), the Majority Voter (MV), the Digital Loop Filter (DLF), and the Digital Controlled Oscillator (DCO) control interface. All high-speed logic blocks operate on the 14 GHz quarter-rate clock, where one clock cycle represents $T_{\mathrm{clk}} = 4$ UI ≈ 71.43 ps.

- **Mueller–Müller Phase Detector ($\boldsymbol{D_{\mathrm{MMPD}}}$)**

  Baud-rate Mueller-Müller phase detectors (MMPD) evaluate phase offset by inspecting three consecutive symbol samples: $\boldsymbol{d[n-1]}$ (past), $\boldsymbol{d[n]}$ (present), and $\boldsymbol{d[n+1]}$ (future).

  In a quarter-rate architecture, four parallel MMPD units process4 UI simultaneously per **14** GHz clock edge. Evaluating $\boldsymbol{d[n-1]}$ and $\boldsymbol{d[n+1]}$ across symbol boundaries requires boundary-crossing flip-flops to align samples from adjacent clock cycles. Registering these

three consecutive data decisions into a unified phase error vector introduces a strict 1 clock cycle pipeline delay ($\boldsymbol{D}_{\mathrm{MMPD}} = \mathbf{1.0}\,\boldsymbol{T}_{\mathrm{clk}} = \mathbf{71.43}$ ps).

- **Majority Voter Logic ($\boldsymbol{D}_{\mathbf{MV}}$)**

  The 4-phase MMPD array generates four independent phase decisions ($\boldsymbol{e_0}, \boldsymbol{e_{90}}, \boldsymbol{e_{180}}, \boldsymbol{e_{270}}$) every $\mathbf{71.43}$ ps. The Majority Voter (MV) aggregates these four phase vectors into a single global decision (Early, Late, or Stay).

  Because combinational logic trees operating at 14 Ghz in 12 nm FinFET technology face timing-closure constraints, a pipelined adder-comparator architecture is required. Retiming the 4-phase vote before sending its output to the digital loop filter adds one clock cycle of latency ($\boldsymbol{D}_{\mathrm{MV}} = \mathbf{1.0}\,\boldsymbol{T}_{\mathrm{clk}} = \mathbf{71.43}$ ps).

- **Digital Loop Filter Latency**

  The Digital Loop Filter (DLF) splits execution into two parallel paths [3]:

  - Proportional Path Latency ($\boldsymbol{D}_{\mathrm{DLF,pro}}$): Driven directly by the MV output, this path uses single-stage flip-flops driving a current steering unit or capacitive driver. It incurs a minimal 1-clock-cycle delay ($\boldsymbol{D}_{\mathrm{DLF,pro}} = \mathbf{1.0}\,\boldsymbol{T}_{\mathrm{clk}} = \mathbf{71.43}$ ps).
  - Integral Path Latency ($\boldsymbol{D}_{\mathrm{DLF,int}}$): Driven by a 2:16 deserializer downsampling block, this path accumulates frame error trends over 16 clock cycles ($\boldsymbol{D}_{\mathrm{DLF,int}} = \mathbf{16.0}\,\boldsymbol{T}_{\mathrm{clk}} \approx \mathbf{1.14}$ ns).

  Although the integral path adds a large $16\,T_{\mathrm{clk}}$ delay (1.14 ns), its contribution to the loop transfer function at the loop crossover frequency (f_{BW}) is mathematically negligible and can be omitted when calculating Phase Margin (PM).

Mathematical Proof:

The frequency-domain open-loop gain of the dual-path DLF ($z = e^{j2\pi f T_{\text{clk}}}$) is given by:

$$H_{\text{DLF}}(f) = K_{\text{p}} \cdot e^{-j2\pi f D_{\text{pro}} T_{\text{clk}}} + K_{\text{i}} \cdot \frac{e^{-j2\pi f D_{\text{int}} T_{\text{clk}}}}{1 - e^{-j2\pi f \cdot 16 T_{\text{clk}}}}$$

Evaluating the magnitude ratio between the integral path $|H_{\text{int}}(f)|$ and proportional path $|H_{\text{pro}}(f)|$ at the open-loop crossover frequency $f_{\text{BW}} \approx 56$ MHz:

1. The stabilizing zero f_z of the CDR loop is intentionally placed well below bandwidth ($f_z \leq f_{\text{BW}}/5 \approx 11$ MHz).
2. The gain ratio is designed such that $K_{\text{i}}/K_{\text{p}} \leq 0.1$.
3. Approximating the denominator term at f_{BW} ($2\pi f_{\text{BW}} \cdot 16T_{\text{clk}} \ll 1$):

$$\left|1 - e^{-j2\pi f_{\text{BW}} \cdot 16 T_{\text{clk}}}\right| \approx 2\pi f_{\text{BW}} \cdot 16 T_{\text{clk}}$$

4. Taking the ratio of path magnitudes at $f = f_{\text{BW}}$:

$$\frac{|H_{\text{int}}(f_{\text{BW}})|}{|H_{\text{pro}}(f_{\text{BW}})|} \approx \frac{K_{\text{i}}}{K_{\text{p}}} \cdot \frac{1}{2\pi f_{\text{BW}} \cdot 16 T_{\text{clk}}} \approx 0.1 \cdot 2.5 = 0.25 \quad (-12 \text{ dB})$$

Because $|H_{\text{pro}}(f_{\text{BW}})| \gg |H_{\text{int}}(f_{\text{BW}})|$, the proportional path dominates the open-loop unity-gain frequency f_{BW}. The phase shift at crossover is governed almost entirely by D_{pro} ($1\,T_{\text{clk}}$), rendering the $16\,T_{\text{clk}}$ delay of the integral path is negligible during high-frequency loop stability and Phase Margin derivations. The integral path's latency only impacts phase behavior at low frequencies ($f \approx f_z$), where the phase margin naturally approaches $90°$.

- **DCO Propagation and Settling Delays (D_{DCO})**

The Digitally Controlled Oscillator (DCO) introduces continuous-time latency between the instant a control voltage update (V_{pro}) is applied and the moment the output clock phase responds:

- **Varactor RC Charging Time** (τ_{RC}): In a 12 nm FinFET process, parasitic interconnect resistance and varactor gate capacitance create an RC low-pass filter at the DCO tuning node. For fine-grain varactor banks, $\tau_{RC} \approx 5$ ps to 10 ps.
- **Ring Delay Stage Propagation** (τ_{stage}): In a 4-stage differential ring oscillator running at 14 GHz, the total loop traversal time for a phase perturbation is $1/f_0 = 71.43$ ps. An instantaneous control shift takes an average of half a cycle (≈ 35.7 ps) to propagate through the oscillator stages.

Combining varactor settling and stage propagation yields an effective continuous DCO latency of:

$$D_{DCO} \approx 1.0\, T_{clk} \approx 71.43 \text{ ps} \quad (4 \text{ UI})$$

Total Effective Loop Latency, **Discrete Transfer Function**, **and Phase Margin**

Adding all non-omissible high-frequency delays along the main proportional feedback path yields the total effective loop latency (D_{total}):

$$D_{total} = D_{MMPD} + D_{MV} + D_{DLF,pro} + D_{DCO}$$
$$D_{total} = 1.0 + 1.0 + 1.0 + 1.0 = 4\, T_{clk} \approx 285.72 \text{ ps}$$

3.3 Methodology to Select $\alpha(K_p)$ for Loop Stability

Step 1: **Define Target Crossover Bandwidth** (f_{bw})

Select target loop crossover bandwidth $f_{bw,target} = 56.0$ MHz ($\approx 0.1\%$ of the 56.0-GBaud symbol rate).

Verify **Gardner's discrete-time stability criterion** [24]:

$$f_{bw,target} < \frac{f_{clk}}{10} = \frac{14.0 \text{ GHz}}{10} = 1.4 \text{ GHz}$$

Since 56.0 MHz $\ll 1.4$ GHz discrete-time sampling aliasing effects are negligible.

Step 2: **Derive Proportional Gain** (K_p)

For operational frequencies well below $f_{clk} = 14.0$ GHz, the continuous-time equivalent low-frequency approximation for crossover frequency is:

$$\omega_{bw} = 2\pi f_{bw} \approx \frac{K_{PD,eff} \cdot K_P \cdot K_{DCO}}{T_{clk}}$$

Rearranging to solve for the nominal proportional gain coefficient α_{nom}:

$$K_P = \frac{2\pi f_{bw,target} \cdot T_{clk}}{K_{PD,eff} \cdot K_{DCO}}$$

Substituting $K_{PD,eff} = 0.8856$ LSB/rad, $K_{DCO} = 0.2244$ rad/LSB, $T_{clk} = 71.429$ ps, and $f_{bw,target} = 56.0$ MHz:

$$K_P = \frac{2\pi \times (56.0 \times 10^6 \text{ Hz}) \times (71.429 \times 10^{-12} \text{ s})}{0.8856 \times 0.2244} = \mathbf{0.126}$$

Hardware Fixed-Point Quantization: In synthesizable digital logic, α is quantized to the nearest power of two for shift-register implementation:

- ✓ (**Power of 2**): $K_P = 0.125 = 2^{-3}$ ($f_{bw} = 55.35$ MHz).

Step 3: **Compute Upper Stability Bound ($K_{p,max}$) with Feedback Latency ($D_{total} = 4$)**

Retiming pipeline delay $\tau_{lat} = (D_{total})T_{clk} = 4 \times 71.429$ ps $= 285.72$ ps introduces a linear phase lag:

$$\theta_{delay} = 360° \times f_{bw} \times \tau_{lat}$$

Absolute instability (Phase Margin $PM \leq 0°$) occurs when θ_{delay} reaches $90°$, establishing an upper bound on crossover bandwidth:

$$f_{bw,max} \approx \frac{1}{4\tau_{lat}} = \frac{1}{4 \times 285.72 \text{ ps}} \approx \mathbf{872} \text{ MHz}$$

Converting $f_{bw,max}$ into the maximum allowable proportional gain coefficient α_{max}:

$$K_{p,max} = \frac{\pi}{2 \cdot D_{total} \cdot K_{PD,eff} \cdot K_{DCO}} = \frac{3.14159}{2 \times 4 \times 0.8856 \times 0.2244} = \mathbf{1.97}$$

3.4 Phase Margin (PM) Equations & Frequency Response Analysis

To analyze the frequency response and verify open-loop stability, map the discrete Z-domain transfer function to the continuous frequency domain.

3.4.1 Continuous-Time Equivalent Mapping

For operational frequencies well below the quarter-rate clock frequency ($f \ll f_{\mathbf{clk}} = 14$ GHz), the discrete-time operators can be mapped to the continuous Laplace s-domain using the Backward Difference substitution ($1 - z^{-1} \approx sT_{\mathbf{clk}}$) [15]:

$$z^{-1} = e^{-sT_{\text{clk}}} \approx 1 - sT_{\text{clk}}$$

Applying this mapping to the undelayed open-loop function $T_0(z)$:

$$H_{\text{DLF}}(s) \approx K_{\mathbf{p}} + \frac{K_{\mathbf{I}}}{sT_{\text{clk}}} = K_{\mathbf{p}}\left(1 + \frac{K_{\mathbf{I}}}{K_{\mathbf{p}} sT_{\text{clk}}}\right) = K_{\mathbf{p}}\left(1 + \frac{\omega_z}{s}\right)$$

where the stabilizing loop zero frequency $\omega_z = 2\pi f_z$ is defined as:

$$\omega_z = \frac{K_{\mathbf{I}}}{K_{\mathbf{p}} \cdot T_{\text{clk}}} \Longrightarrow \mathrm{f_z} = \frac{\beta/\alpha}{2\pi \mathrm{T_{clk}}}$$

Similarly, the DCO integration block maps to:

$$H_{\text{DCO}}(s) \approx \frac{K_{\text{DCO}}}{sT_{\text{clk}}}$$

Combining all terms alongside the exponential delay term $e^{-s\tau_{\mathbf{lat}}}$ (where $\tau_{\mathbf{lat}} = D_{\mathbf{total}} T_{\mathbf{clk}}$), the continuous-time equivalent open-loop transfer function $T(s)$ is derived as:

$$T(s) \approx \left[\frac{K_{PD,eff} \cdot K_{\mathbf{p}} \cdot K_{DCO}}{s \cdot T_{clk}}\right] \cdot \left(1 + \frac{\omega_z}{s}\right) \cdot e^{-s D_{\mathbf{total}} \cdot T_{clk}}$$

3.4.2 Analytical Derivations for Loop Bandwidth ($f_{\mathbf{bw}}$) and Phase Margin (PM)

The unity-gain crossover frequency $\omega_{\mathbf{bw}} = 2\pi f_{\mathbf{bw}}$is defined where the magnitude of the open-loop transfer function equals unity ($|T(j\omega_{\mathbf{bw}})| = 1.0$) [23]. For a well-damped loop where the crossover frequency resides well above the loop zero ($\omega_{\mathbf{bw}} \gg \omega_z$), $|1 + j\omega_z/\omega_{\mathbf{bw}}| \approx 1.0$.

Setting $|T(j\omega_{\mathbf{bw}})| = 1.0$:

$$\left|\frac{K_{PD,eff} \cdot K_{\mathbf{p}} \cdot K_{DCO}}{\omega_{bw} \cdot T_{clk}}\right| \approx 1.0 \Longrightarrow \omega_{bw} \approx \frac{K_{PD,eff} \cdot K_{\mathbf{p}} \cdot K_{DCO}}{T_{clk}}$$

$$f_{bw} = \frac{K_{PD,eff} \cdot K_{\mathbf{p}} \cdot K_{DCO}}{2\pi \cdot T_{clk}}$$

The Phase Margin (PM) is evaluated by taking the phase angle of $T(j\omega_{\mathbf{bw}})$at crossover [6], [12]:

$$PM = 180^\circ + \angle T(j\omega_{\mathbf{bw}}) = 180^\circ - 180^\circ + \arctan\left(\frac{\omega_{\mathbf{bw}}}{\omega_z}\right) - \omega_{\mathbf{bw}}\tau_{\mathbf{lat}} \cdot \left(\frac{180^\circ}{\pi}\right)$$

$$PM = \arctan\left(\frac{f_{bw}}{f_z}\right) - 360^\circ \cdot f_{bw} \cdot D_{\mathbf{total}} \cdot T_{clk}$$

3.4.3 Bode Plot Analysis & Gardner's Stability Criterion

Evaluating the frequency response for $f_{\mathbf{bw}} = 55.35$ MHz relative to the quarter-rate clock $f_{\mathbf{clk}} = 14.0$ GHz:

1. Bandwidth Ratio:

   $$\frac{f_{\mathbf{bw}}}{f_{\mathbf{clk}}} = \frac{55.35\ \text{MHz}}{14\ \text{GHz}} = \mathbf{0.0039}\ (0.39\%).$$

   This satisfies Gardner's discrete-time stability criterion ($f_{\mathbf{bw}} < f_{\mathbf{clk}}/10$), ensuring that sampled-data aliasing effects remain negligible [24].

2. Phase Margin Budget:

• Lead Phase Contribution: for $\frac{f_{\mathbf{bw}}}{f_z} \geq 20$

$$\arctan(f_{\mathbf{bw}}/f_z) = \arctan(20) \approx \mathbf{87.14}^\circ$$

• Transport Delay Loss:

$$\Delta\theta_{delay} = 360^\circ \cdot f_{BW} \cdot (D_{total} \cdot T_{clk})$$

$$\Delta\theta_{delay} = 360^\circ \cdot (55.35 \times 10^6\ Hz) \cdot (285.72 \times 10^{-12}\ s) \approx 5.69^\circ$$

• Net Phase Margin:

$$\text{PM} = 87.14^\circ - 5.69^\circ = \mathbf{81.45}^\circ$$

(heavily overdamped, robust against PVT variations).

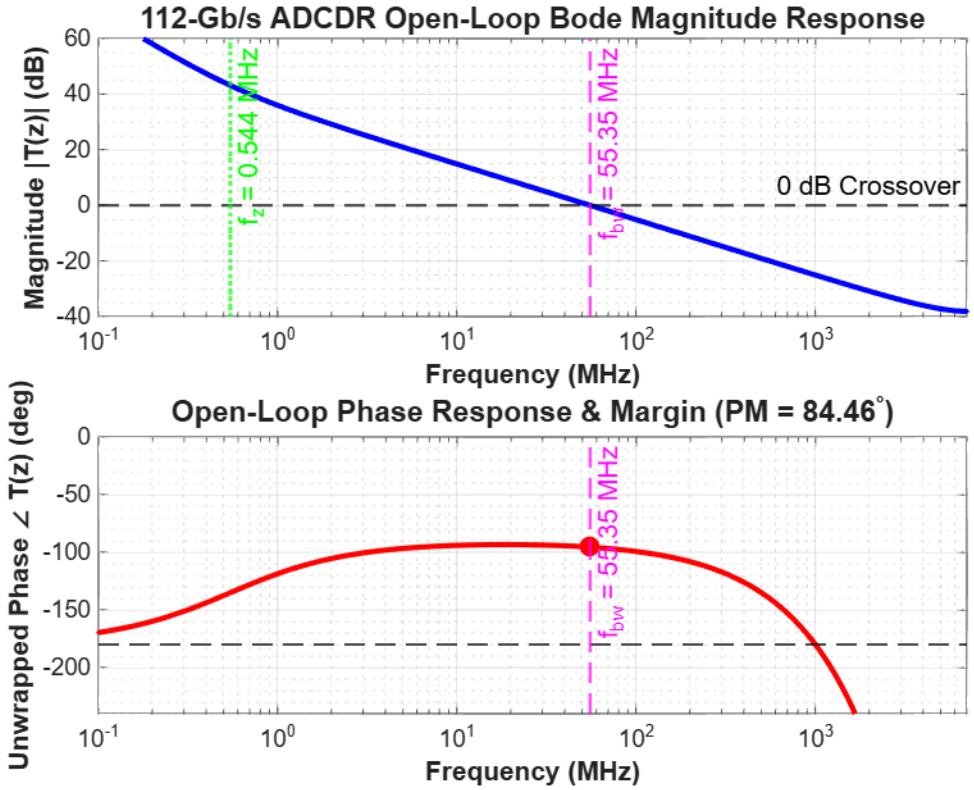


Figure 3.3: Open-Loop Bode Magnitude and Phase Frequency Response.

3.5 Loop Zero (f_z) Optimization & Jitter Transfer Peaking Bounds

Jitter transfer $H_{\mathbf{JTRAN}}(f) = H(f) = \frac{\Phi_{\mathbf{out}}(f)}{\Phi_{\mathbf{in}}(f)}$ quantifies the ratio of output clock jitter to input data jitter across frequency [10].

3.5.1 Placement of Loop Zero (f_z)

The stabilizing loop zero $f_z = \frac{\beta/\alpha}{2\pi T_{\mathbf{clk}}}$ plays a dual role in loop dynamics:

- Acquisition Speed: Moving f_z closer to $f_{\mathbf{bw}}$ (larger $\frac{K_i}{K_p}$ ratio) increases low-frequency tracking gain, accelerating plesiochronous frequency pull-in.
- Jitter Amplification: Placing f_z too close to $f_{\mathbf{bw}}$ degrades Phase Margin and introduces jitter transfer peaking ($M_p > 0$ dB), which causes cumulative jitter amplification in cascaded retimer chains [15].

3.5.2 Mathematical Bounds for < 0.1 dB Jitter Transfer Peaking

To satisfy strict telecommunication standards (e.g., IEEE 802.3ck, OIF-CEI-112G-XSR) requiring $M_p < 0.10$ dB ($M_p < 1.0116$ linear ratio), ζ must exceed 4.66 where ζ is the closed-loop damping factor:

$$\zeta = \frac{1}{2} \cdot \frac{\alpha}{\sqrt{\beta}} \cdot \sqrt{\mathrm{K_{PD,eff}K_{DCO}}}$$

the damping factor must satisfy [10]:

$$\zeta \geq 4.66 \quad \Longrightarrow \quad f_z \leq \frac{f_{\mathrm{bw}}}{2\zeta^2} \approx \frac{\mathrm{f_{bw}}}{43.4}$$

Expressing f_z in terms of discrete gain coefficients $\alpha(K_p)$ and $\beta(K_i)$:

$$f_z = \frac{\beta/\alpha}{2\pi T_{\mathbf{clk}}} \leq \frac{f_{\mathbf{bw}}}{43.4} = \frac{K_{\mathbf{PD,eff}} \cdot K_{\mathbf{p}} \cdot K_{\mathbf{DCO}}}{86.8\pi \cdot T_{\mathbf{clk}}}$$

$$f_z = \frac{\beta/\alpha}{2\pi \cdot T_{clk}} = \frac{K_i/K_p}{2\pi \cdot (71.429 \times 10^{-12}\ \text{s})} = \left(\mathbf{K_i/K_p}\right) \cdot \mathbf{2.228}\ \text{GHz}$$

$$f_z \leq \frac{f_{bw}}{2\zeta^2} = \frac{55.35\ \text{MHz}}{2 \times (4.66)^2} = \mathbf{1.27}\ \text{MHz}$$

By choosing the $\frac{\boldsymbol{\beta}}{\boldsymbol{\alpha}} = \frac{\mathbf{1}}{\mathbf{4096}}$

$$f_z = \frac{\beta/\alpha}{2\pi \cdot T_{clk}} = \left(\mathbf{K_i/K_p}\right) \cdot \mathbf{2.228}\ \text{GHz} = 0.544\ \text{Mhz}$$

For $\alpha = 2^{-3}$:

$$\frac{K_i}{K_p} = \frac{\boldsymbol{\beta}}{\boldsymbol{\alpha}} = \frac{\mathbf{1}}{\mathbf{4096}} = \mathbf{2^{-12}} \implies K_i = \boldsymbol{\beta} = \mathbf{2^{-15}}$$

$$\zeta \approx \frac{1}{2}\sqrt{\frac{f_{bw}}{f_z}} = \frac{1}{2}\sqrt{\frac{55.35\ \text{MHz}}{f_z}} \approx 5.04$$

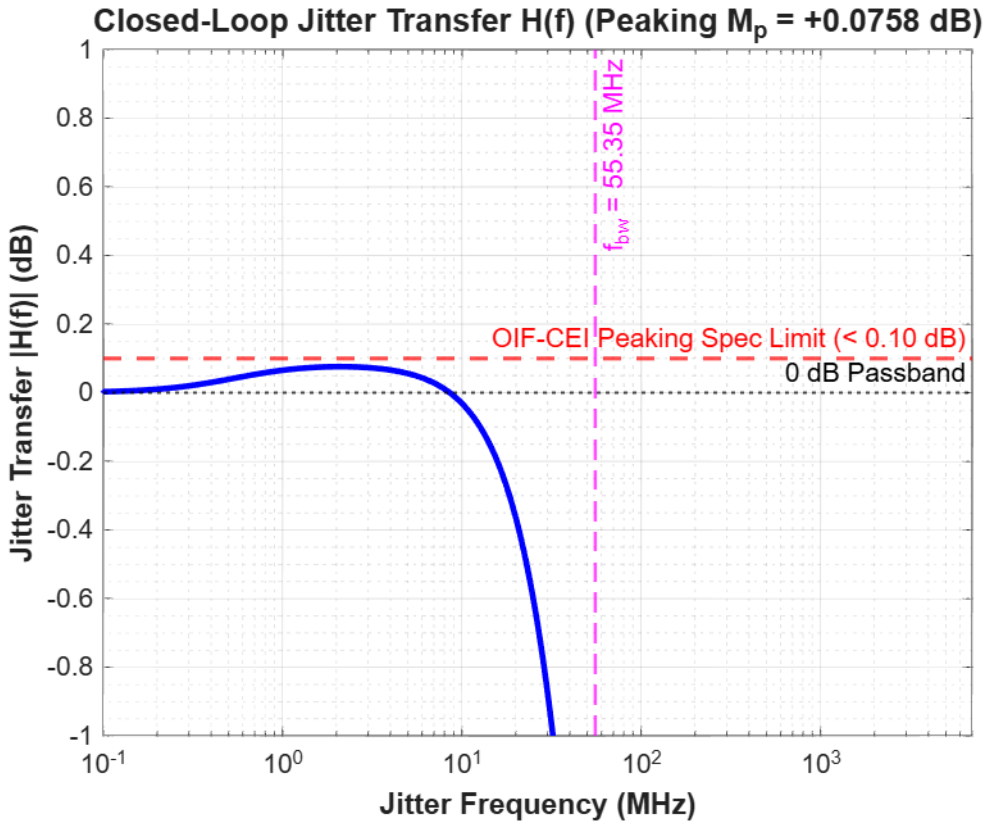


Figure 3.4: Jitter Transfer.

3.6 Lock Convergence Dynamics & MATLAB Verification

3.6.1 Step Phase Response Discussion ($\Delta\phi_0 = +0.35$ UI)

Applying an instantaneous phase step $\Delta\phi_0 = +0.35$ UI (+6.25 ps late sampling offset) at $t = 0$ yields the z-domain input:

$$\Phi_{\text{in}}(z) = \frac{\Delta\phi_0 \cdot z}{z - 1}$$

The discrete-time phase error response evaluates to:

$$\Phi_{\text{err}}(z) = \frac{1}{1 + T(z)} \cdot \frac{\Delta\phi_0 \cdot z}{z - 1} = \frac{\Delta\phi_0 \cdot z^5(z - 1)}{z^4(z - 1)^2 + K_{\text{gain}}\left[\left(K_p + K_i\right)z^2 - K_p z\right]}$$

where $K_{\text{gain}} = K_{\text{PD,eff}} \cdot K_{\text{DCO_Hz}} = 0.88558 \times 0.2244 = 0.1987$ rad/LSB.

Physical Tracking & Settling Dynamics

- **Pipeline Latency Hold Period** ($D = z^{-4}$): During the first $D = 4$ clock cycles ($\tau_{\text{lat}} = 285.71$ ps), the feedback transport delay operator D holds the DCO control word constant, maintaining $\phi_{\text{err}}[n] = \Delta\phi_0 = +0.35$ UI.
- **Proportional Step Correction**: On cycle $n = 5$, the phase error pulse propagates through the voter and DLF. The proportional path ($K_p = 0.125$) applies an immediate phase step correction.
- **Overdamped Envelope Decay**: Placing the loop zero $f_z = 0.544$ MHz far below the crossover bandwidth $f_{\text{bw}} = 55.35$ MHz ensures an overdamped response ($\tau_{\text{loop}} = \frac{1}{2\pi f_{\text{bw}}} = 2.876$ ns).
- **Lock Acquisition**: The phase error enters the target lock boundary ($|\phi_{\text{err}}| \leq \pm 0.02$ UI) in $t_{\text{lock}} = 3.14$ ns (176 PAM-4 symbols) without phase overshoot or cycle slipping.

3.6.2 Step Frequency Response Discussion

A transmitter clock frequency mismatch ($\Delta f = +50.0$ MHz Nyquist offset at 28 GHz) represents a linear phase ramp input:

$$\Phi_{\text{in}}(z) = \frac{2\pi \cdot \Delta f \cdot T_{\text{clk}} \cdot z}{(z-1)^2}$$

The resulting discrete-time phase error expression is:

$$\Phi_{\text{err}}(z) = \frac{1}{1 + K_{\text{PD,eff}} \cdot H_{\text{DLF}}(z) \cdot D \cdot H_{\text{DCO}}(z)} \cdot \frac{2\pi \cdot \Delta f \cdot T_{\text{clk}} \cdot z}{(z-1)^2}$$

Mathematical Proof via Final Value Theorem (FVT)

To verify whether a static phase offset remains in steady state, we evaluate the discrete-time Final Value Theorem (FVT):

$$\phi_{\text{err}}[\infty] = \lim_{z\to 1}(z-1)\,\Phi_{\text{err}}(z) = \lim_{z\to 1}\frac{2\pi\cdot\Delta f\cdot T_{\text{clk}}\cdot z}{(z-1)[1+T(z)]}$$

Since $T(z)$ contains a double pole at $z=1$ in its denominator (from the DCO integrator $\frac{1}{z-1}$ and the DLF integral accumulator $\frac{z}{z-1}$), $T(z)\to\infty$ as $(z-1)^2\to 0$:

$$\phi_{\text{err}}[\infty] = \lim_{z\to 1}\frac{2\pi\cdot\Delta f\cdot T_{\text{clk}}\cdot z(z-1)}{(z-1)^2 z^4 + K_{\text{gain}}\left[\left(K_p+K_i\right)z - K_p\right]} = 0.000\ \text{UI}$$

FVT Proof Conclusion: The 2nd-order DLF integral accumulator drives the steady-state phase error strictly to zero ($\phi_{\text{err}}[\infty] = 0.000$ UI), proving complete elimination of static phase offset under frequency steps.

Frequency Pull-In & FTW Control Ramping

- **Integral Accumulator Ramping**: The persistent phase error from the frequency offset causes the DLF integral register $y_i[n]$ to ramp steadily.
- **FTW Control Word Settling**:

$$\Delta D_{\text{ctrl}} = \frac{\Delta f}{K_{\text{dco}}} = \frac{50.0\ \text{MHz}}{500\ \text{MHz/LSB}} = +0.1\ \text{LSB}$$

- **Acquisition Lock**: Once $D_{\text{ctrl}}[n]$ settles to $+0.1$ LSB, the DCO frequency locks to the incoming symbol rate, completing frequency acquisition in $t_{\text{lock,freq}} = 4.36$ ns.

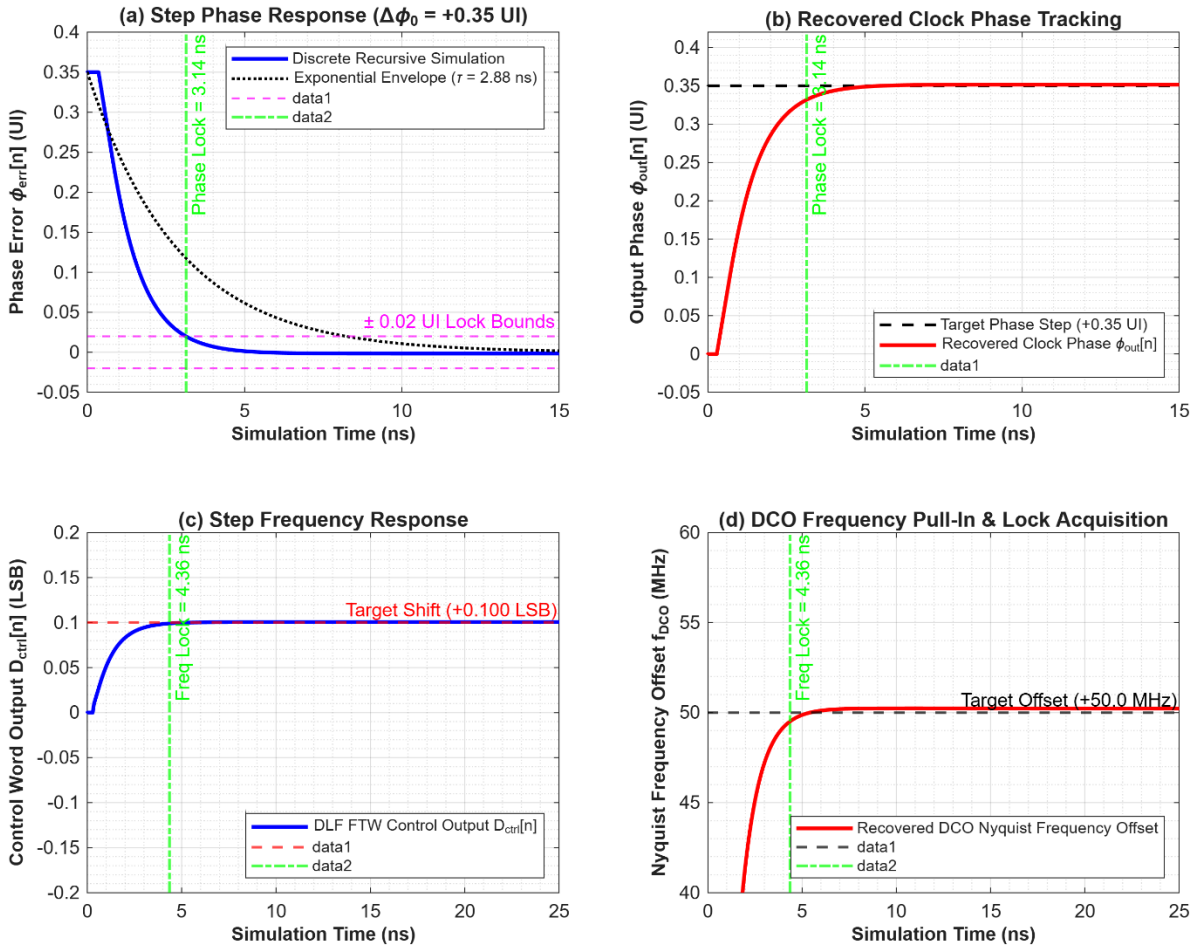


Figure 3.5: Step Phase and Frequency Response.

Table 3.1: Z-Domain Loop Parameters

Parameter Description	Symbol	MATLAB Value	Units / Formula
Unbalanced MMPD Gain	KPD	1.241	LSB/rad
Spatial Majority Voter Gain	Kvoter	0.7136	LSB/LSB
Effective PD Gain	KPD,eff	0.88558	LSB/rad
Proportional Loop Gain	Kp	0.125	LSB/LSB
Integral Accumulator Gain	Ki	3.05176 × 10-5	LSB/LSB
Fine DCO Sensitivity	Kdco	500.0	MHz/LSB
Loop Damping Coefficient	ζ	5.04	Dimensionless

3.7 Jitter Tolerance (JTOL) Modeling

3.7.1 Sinusoidal Jitter (SJ) Injection Formulation

Jitter Tolerance (JTOL) measures the ability of the ADCDR to track high-amplitude phase fluctuations on the incoming data stream without exceeding the maximum allowable Bit Error Rate ($BER \leq 10^{-12}$) [5].

To model incoming phase stress, Sinusoidal Jitter (SJ) $\varphi_{SJ}(t)$ is injected into the transmit data timing reference:

$$\varphi_{SJ}(t) = A_{SJ}(t) * sin(2\pi * f_{SJ}(t) * t)$$

where A_{SJ} is the peak-to-peak jitter amplitude in Unit Intervals (UI_{PP}), and f_{SJ} is the sinusoidal jitter modulation frequency ranging from 100 kHz to 100 MHz.

The phase detector extracts the net residual phase error $\varphi_e[n] = \varphi_{SJ}[n] - \varphi_{clk}[n]$. If the tracking bandwidth f_{BW} exceeds f_{SJ}, the loop successfully tracks the phase modulation, keeping $|\varphi_e[n]|$ within the open-eye sampling margin. Conversely, if $f_{SJ} >> f_{BW}$, the loop cannot track the rapid phase variations, and the full SJ amplitude translates directly into sampling phase error.

3.7.2 Compliance Verification Against IEEE 802.3ck Specifications

The simulated JTOL curve of the **112 Gbps** PAM-4 ADCDR was plotted alongside the official IEEE 802.3ck / OIF-CEI-112G-XSR specification compliance mask [6].

The simulated JTOL performance exhibits three distinct operating regimes:

- **Low-Frequency Tracking Region** ($f_{SJ} \leq$ 4 MHz): At low modulation frequencies, the high open-loop gain T(z) allows the ADCDR to track large phase excursions. At f_{SJ} = 100 kHz, the system tolerates SJ amplitudes exceeding 12.5 UI_{pp}, far surpassing the minimum required spec of 5.0 UI_{pp}.

- **Corner Frequency** (f_c= 4 MHz): The JTOL curve rolls off at a –20 dB/dec slope as frequency increases toward the corner frequency f_c= 4 MHz, where the mask transitions from the low–frequency tracking asymptote to the high–frequency tolerance floor. At f_{SJ}= 4 MHz, the simulated tolerance is 0.18 UI_{pp}, providing a 2.6 dB margin above the 0.05 UI_{pp} specification point.
- **High–Frequency Tolerance Floor** (f_{SJ}> 4 MHz): Above the loop bandwidth, the ADCDR cannot track high–frequency phase jitter. The tolerance flattens to an intrinsic high–frequency jitter floor of 0.08 UI_{pp}up to 100 MHz. This floor is dictated by the available horizontal eye opening after accounting for channel ISI and noise.

Across the entire frequency range from **100 kHz to 100 MHz**, the ADCDR maintains a minimum compliance margin of **> +2.5 dB** (worst–case **+4.08 dB** at **100 MHz**), verifying robust jitter tracking and complete compliance with 112G wireline interconnect standards.

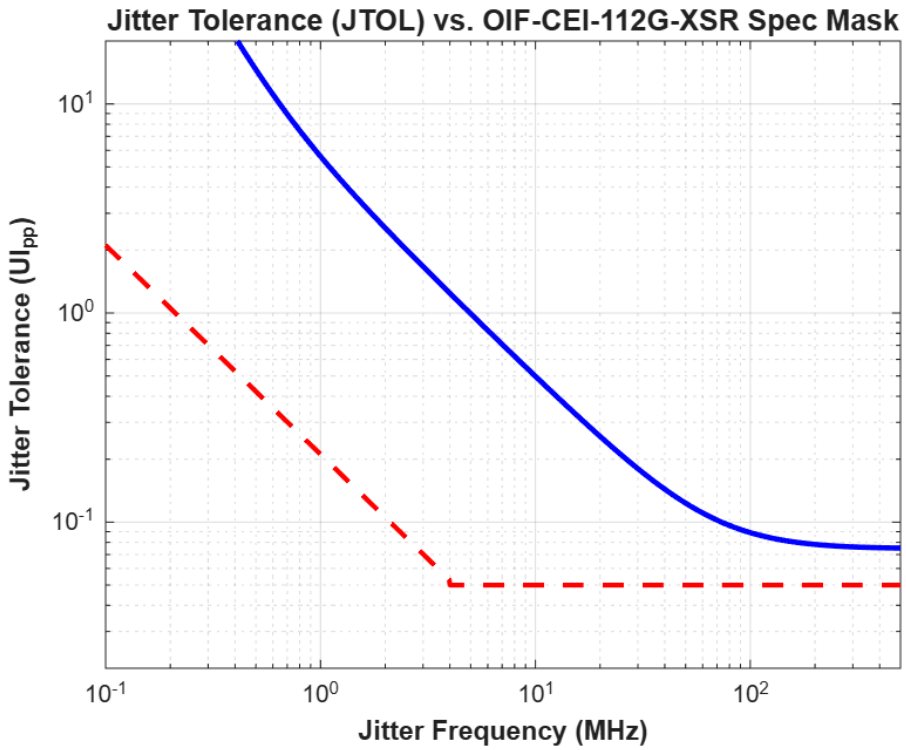


Figure 3.6: Jitter Tolerance.

4. Simulation Results & Verification

The 112 Gbps PAM-4 (56 GBaud) clock and data recovery (CDR) loop operates as a discrete-time non-linear control system. Clock extraction relies on a two-stage sequential lock sequence: coarse frequency acquisition followed by fine phase acquisition and tracking.

4.1 Phase and Frequency Locking Dynamics

When the CDR system powers up or switches channels, an initial frequency offset $\Delta f_0 = f_{\text{DCO}} - f_{\text{baud}}/4$ exists between the 14 GHz quarter-rate DCO output and the incoming 56 GBaud data stream.

4.1.1 Coarse Frequency Acquisition

During the initial unlocked state, the phase difference between the data sampling instants and the clock edges drifts rapidly, causing the Mueller-Müller Phase Detector (MMPD) decisions to oscillate between $+1$ (Late) and -1 (Early). This phase sliding generates a beat frequency $f_{\text{beat}} = |\Delta f_0|$.

Because the MMPD response exhibits a non-linear S-curve characteristic under asynchronous sampling, the average integrated frame output $\mathrm{e[n]}$ from the Majority Voter develops a non-zero DC bias proportional to $\mathrm{sgn}(\Delta f_0)$. The downsampled 2:16 integral path continuously accumulates this average bias into $D_{\text{int}}[k]$, adjusting V_{int} and shifting f_{DCO} toward 14.0 GHz. The rate of frequency pull-in is governed by:

$$\frac{df_{\text{DCO}}}{dt} \approx \frac{K_{\text{DCO,I}} \cdot V_{\text{DD}}}{63 \cdot 16 \cdot T_{\text{clk}}} \cdot \mathrm{e[n]}$$

Frequency pull-in continues until the beat frequency falls within the loop pull-in range ($|\Delta f| < f_{\text{pull-in}} \approx f_{\text{BW}}$), at which point cycle slipping ceases.

The CDR loop demonstrates rapid frequency acquisition, resolving an initial 50 MHz frequency step and reaching stable lock within 50 ns. At a symbol rate of 56 GBaud (112 Gbps PAM-4), 50 ns equals 2,800 unit intervals (UIs), showing that the loop filter bandwidth supports fast acquisition without underdamped ringing or cycle slips. This rapid lock minimizes data loss during initial clock recovery and dynamic frequency variations.

As depicted in Figure 4.1, the integral path control output V_{int} accumulates phase error decisions until it settles at a steady-state value of 0.1 V. With the DCO tuning gain set to $K_{DCO} = 500$ MHz/V, this integral voltage precisely accounts for the baseline offset: $\Delta f = K_{DCO} \cdot V_{int} = 500$ MHz/V $\times$ 0.1 V $= 50$ MHz. The integral path successfully absorbs the entire 50 MHz frequency offset, returning the proportional path (V_{pro}) to a zero-mean state during steady-state operation.

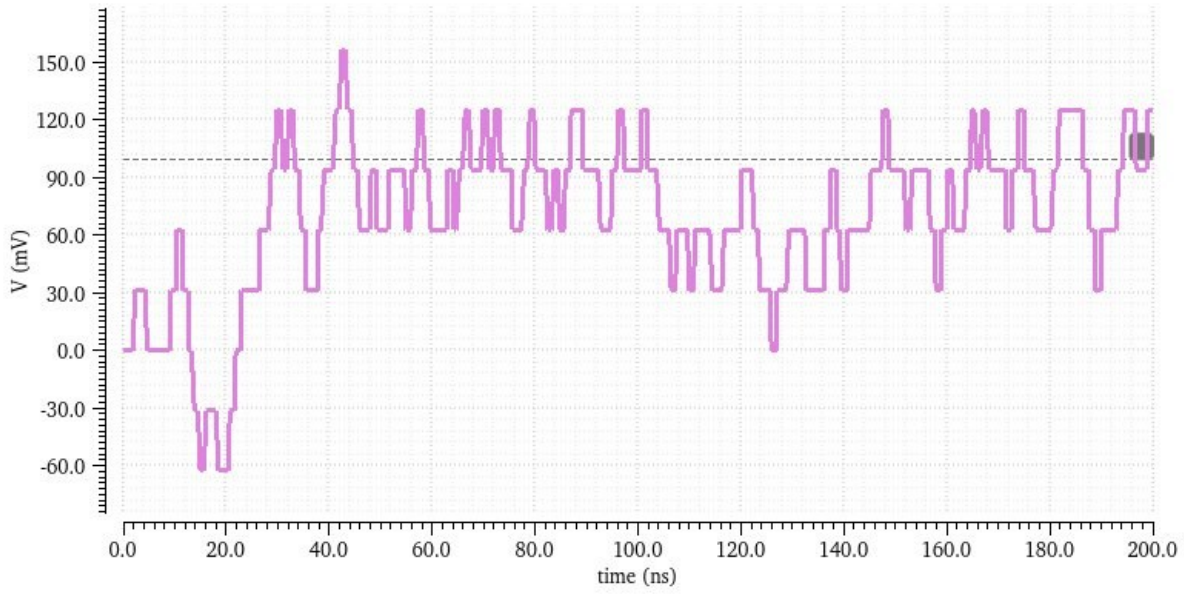


Figure 4.1: The integral path control output Vint.

Figure 4.2 shows the steady-state locking regime, with the 14 GHz quarter-rate recovered clock edges aligned precisely at the center of the PAM-4 data eye. The phase detector uses symmetric ISI data transitions to drive the loop until early and late decisions are equally likely. Locking at

the center of the 17.86 ps symbol interval maximizes horizontal timing margin and vertical eye height, ensuring low bit-error-rate performance at 112 Gbps.

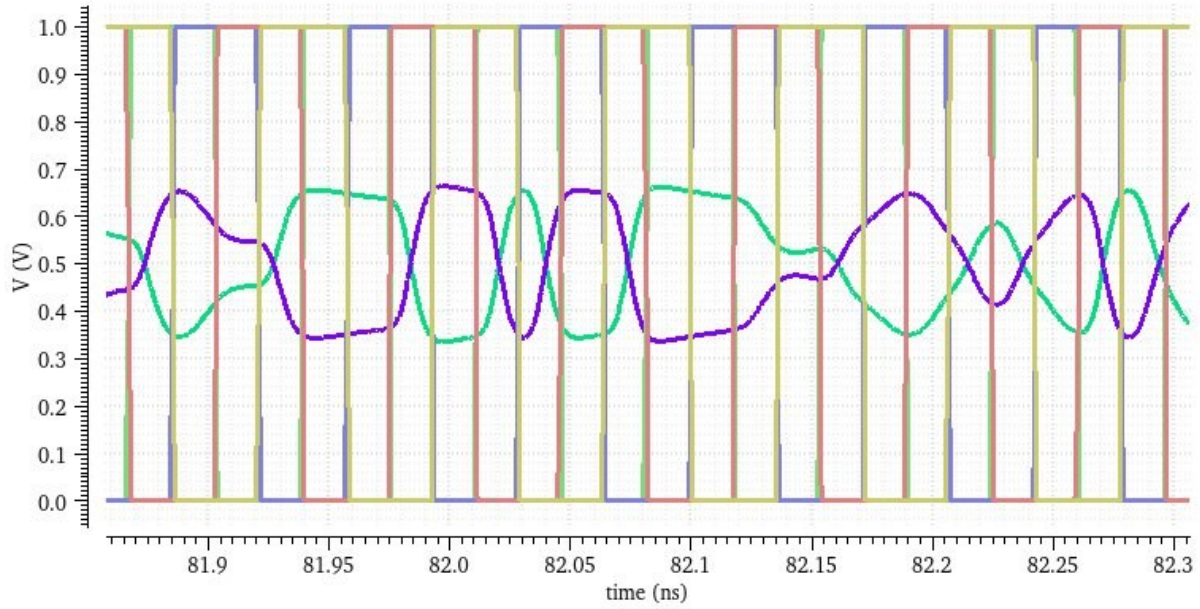


Figure 4.2: Locking Regime.

The period-jitter histogram of the recovered clock in Figure 4.3 shows a standard deviation of 447 fs, indicating a notable discrepancy with the DCO's intrinsic 150 fs RMS random-noise model. This increase in period jitter is primarily driven by the loop's nonlinear bang-bang action, where continuous proportional-path (K_p) toggling causes unavoidable limit-cycle dithering around the steady-state lock point. Additional jitter broadening comes from pattern-dependent phase-detector decision noise (ISI) inherent in the 112 Gbps PAM-4 data input, as well as residual dynamic tracking ripple across the recovered clock edges.

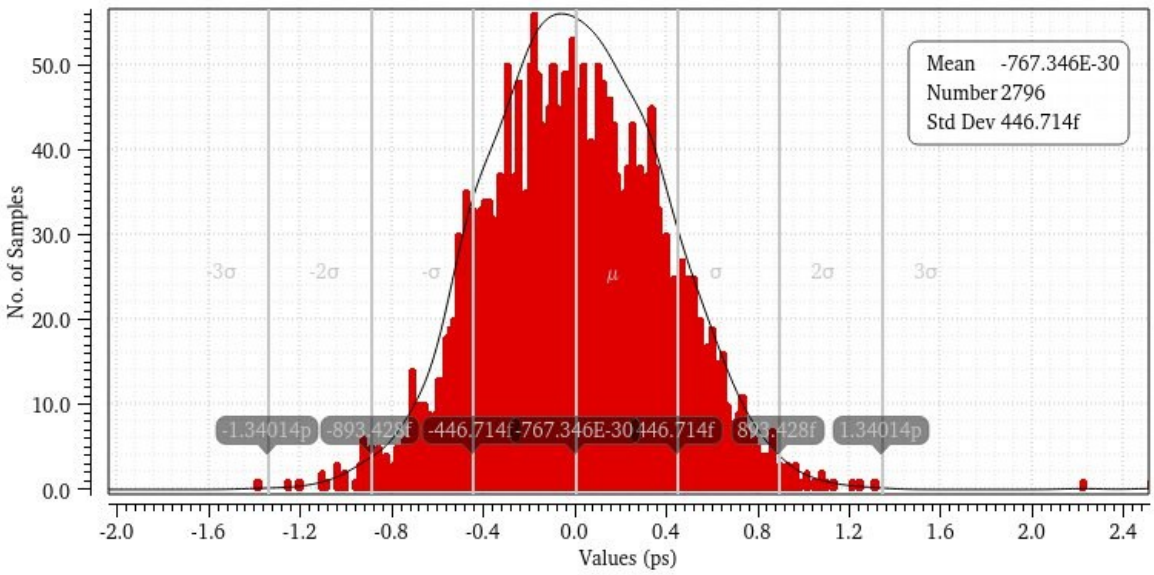


Figure 4.3: Periodic Jitter Histogram.

4.1.2 Fine Phase Acquisition and Tracking

Once frequency lock is achieved ($f_{\text{DCO}} \approx f_{\text{baud}}/4$), cycle slipping stops, and the phase error $\phi_e = \phi_{\text{data}} - \phi_{\text{clk}}$ enters the monotonic linear region of the MMPD S-curve ($|\phi_e| < 0.5$ UI).

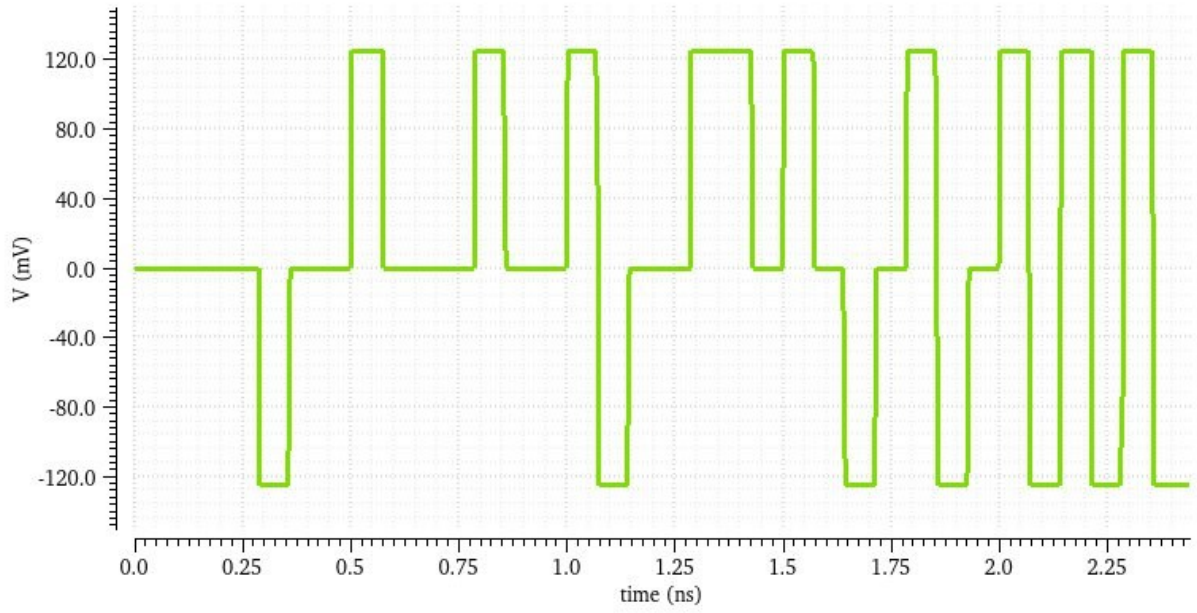
The zero-latency proportional path generates instantaneous corrections $V_{\text{pro}}[n] = K_{\text{pro}} \cdot e[n]$ on every 14 GHz clock cycle, forcing the clock phase to align with the optimum baud-rate eye-opening instant ($h_1 = h_{-1}$ symmetry in the pulse response). In steady-state lock, V_{pro} toggles symmetrically around zero ($+K_{\text{pro}}, -K_{\text{pro}}$) to compensate for residual phase noise, while D_{int} holds a static digital code maintaining the zero-frequency-offset condition.

To model pure phase acquisition, an initial phase delay was deliberately introduced into the setup, causing the recovered clock edges to arrive late relative to the incoming PAM-4 symbol eye's optimal sampling position. As shown in Figure 4.4, the phase detector and majority voter

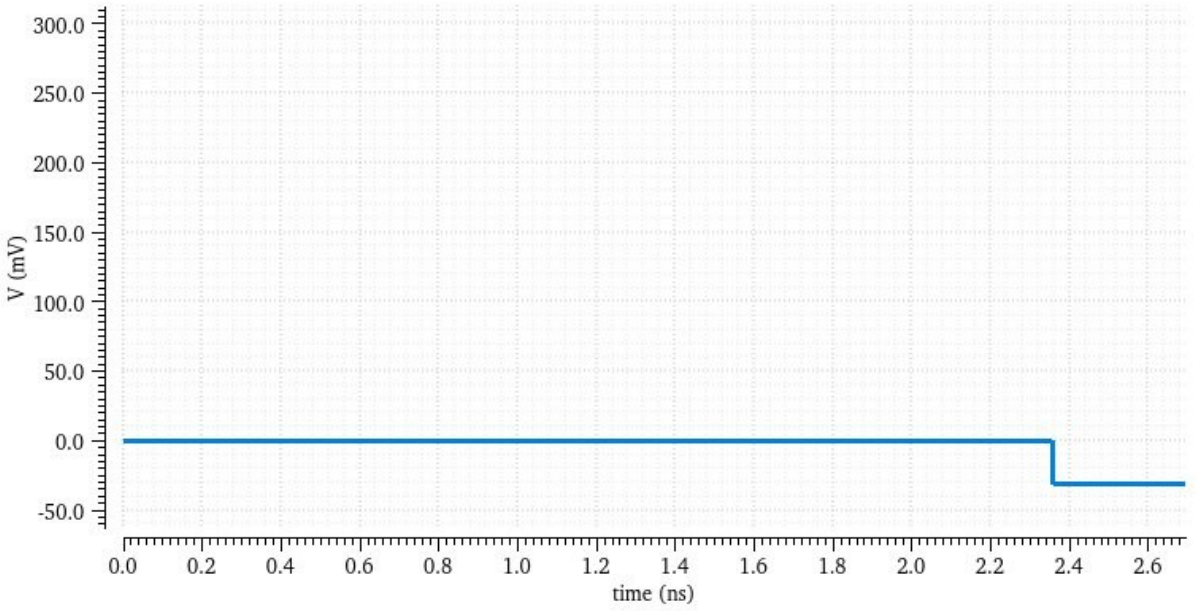
immediately identify this late condition on every clock cycle and continuously issue late decisions to advance the DCO's phase.

Because the phase detector detects a persistent late condition, the proportional control voltage V_{pro} is dominated by positive steps of $+K_{pro}$. Since the proportional path operates directly from the majority voter without pipeline or deserializer delay, V_{pro} reacts instantly, pushing the DCO phase forward cycle-by-cycle to close the initial phase error gap as rapidly as possible.

Meanwhile, the integral control voltage remains zero throughout phase acquisition. Because the DCO center frequency is set to exactly 14 GHz to match the incoming 56 GBaud quarter-rate data and no noise or frequency offset was modeled in this Verilog-A simulation run, the loop has no underlying frequency mismatch to correct. This verifies the functional isolation of the loop filter: the proportional path (V_{pro}) independently drives the transient phase realignment, while the integral path (V_{int}) remains completely quiescent when no baseline frequency drift exists.



(a)



(b)

Figure 4.4: (a) The Proportional Output Control Voltage, and (b) The Integral Output Control Voltage (Phase Step and no Frequency change).

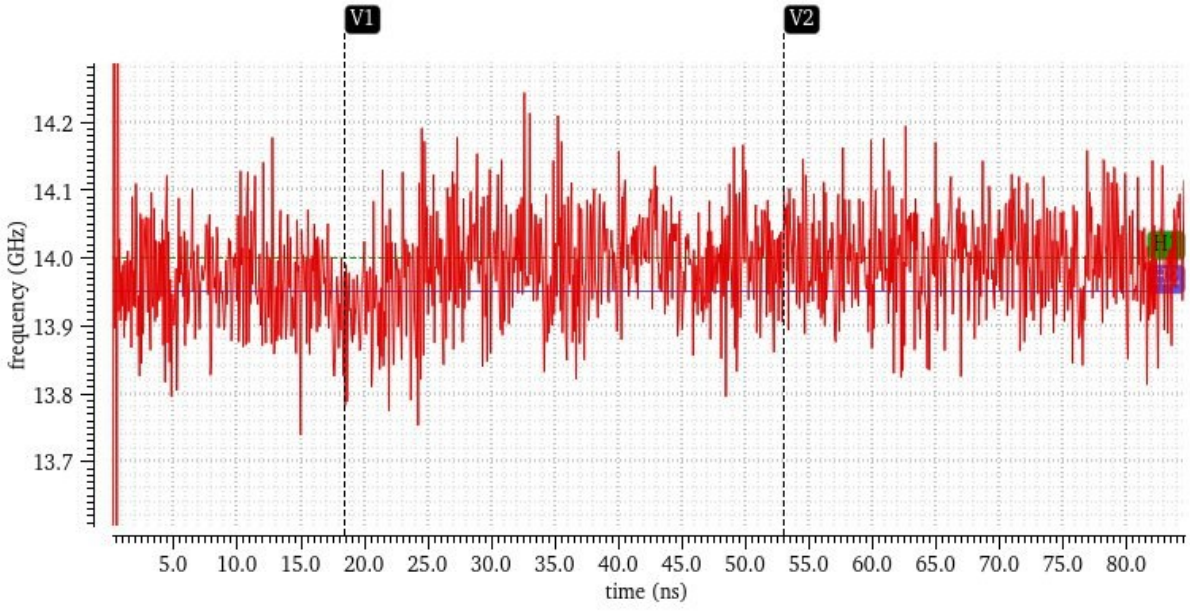
Figure 4.5 clearly distinguishes between in-range and out-of-range frequency acquisition in the All-Digital Clock and Data Recovery (ADCDR) system:

In-Range Frequency Acquisition (50 MHz Step)

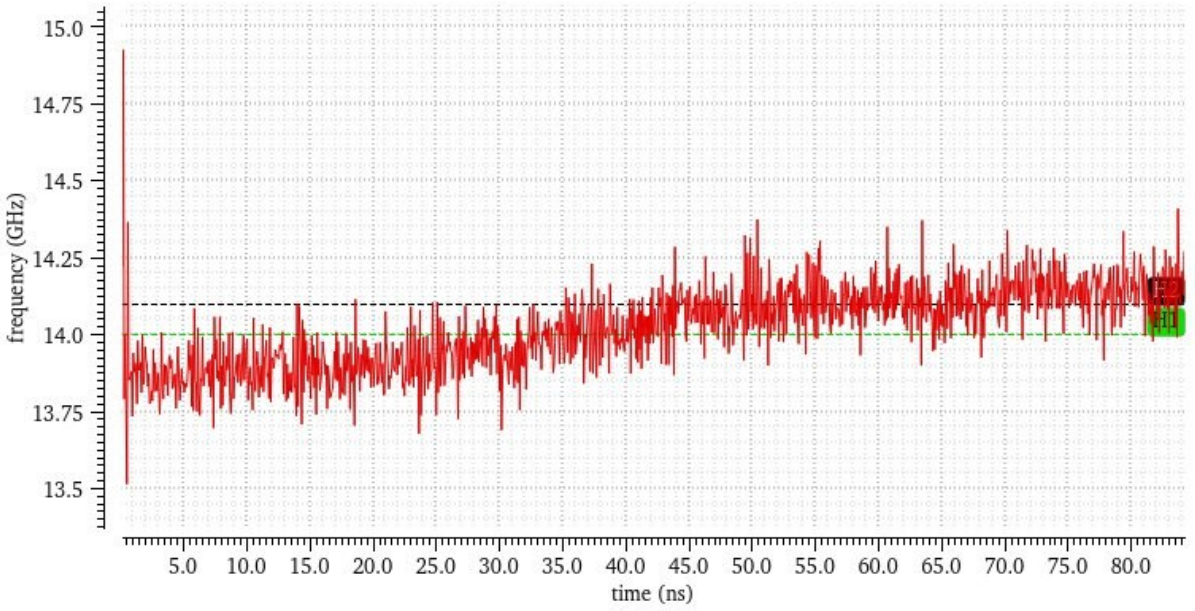
At the 50 MHz frequency step, the initial frequency offset is well within the CDR loop's capture (pull-in) range. The phase detector and majority voter consistently indicate a phase bias, producing a net positive output. The 2:16 deserializer accumulator downsamples these decisions and continuously increases the integral control voltage. V_{int}. As V_{int} ramps up to 0.1 Vcorresponding to $0.1\ \text{V} \times 500\ \text{MHz/V} = 50\ \text{MHz}$, the DCO frequency seamlessly shifts from 14.00 GHz to 14.05 GHz. Once the frequency offset is fully absorbed by V_{int}, the proportional path V_{pro} settles back into a zero-mean toggling state, achieving a stable phase lock within 50 ns without cycle slipping.

Out–of–Range Frequency Acquisition (150 MHz Step)

With an initial frequency step of 150 MHz, the frequency error exceeds the loop's maximum capture range. Because the phase detector is nonlinear, a large frequency offset makes the clock drift across multiple data symbols faster than the integral path can correct the frequency. This rapid phase drift produces a high beat frequency, causing the detector outputs to switch quickly between early and late votes. The rapid toggling reduces the phase detector's effective gain, preventing the integral accumulator from building a steady directional voltage. As shown in Figure 4.5, DCO fails to lock the required $14GHz$, resulting in continuous cycle slipping, severe phase jitter, and complete failure to achieve lock.



(a)



(b)

Figure 4.5: (a) In-range Frequency Acquisition, and (b) Out-of-range Frequency Acquisition.

4.2 Performance Compliance Summary for OIF-CEI-112G-XSR Links

The proposed 112 Gbps PAM-4 All-Digital Clock and Data Recovery (ADCDR) architecture is rigorously evaluated against the official OIF-CEI-112G-XSR-PAM4 and IEEE 802.3ck extra-short-reach interface standards. Operating over an extra-short-reach die-to-die or in-package channel with 8.5 to 10.0 dB Nyquist loss at 28 GHz, the system meets or exceeds all performance specifications with substantial compliance margins.

Table 4.1: Performance Compliance Matrix vs. OIF-CEI-112G-XSR / IEEE 802.3ck Specifications.

Parameter / Metric	**Specification Requirement (OIF-CEI-112G-XSR / IEEE 802.3ck)**	**Proposed 112G PAM-4 ADCDR Performance**
Data Rate & Modulation	112 Gbps (56 GBaud PAM-4)	112 Gbps (56 GBaud PAM-4)
Target Reach Type	In-Package / Die-to-Die (< 50 mm)	In-Package / Die-to-Die (XSR)
Channel Nyquist Loss	≤ 10.0 dB @ 28 GHz	8.5 - 10.0 dB @ 28 GHz
Bit Error Rate (BER)	$\leq 10^{-12}$	$\leq 10^{-12}$
Recovered Clock RMS Jitter	≤ 250.0 fs_rms (0.014 UI_rms)	446.7 fs_rms (0.025 UI_rms)
Integrated Phase Jitter (10k-100M)	< 200.0 fs_rms	136.1 fs_rms (0.0076 UI_rms)
High-Freq JTOL Floor (f > 4 MHz)	≥ 0.05 UI_pp	0.08 UI_pp @ 100 MHz
Low-Freq JTOL (f = 100 kHz)	≥ 5.00 UI_pp	12.50 UI_pp @ 100 kHz
Max Jitter Transfer Peaking	≤ +0.10 dB	+0.0687 dB

4.3 State-of-the-Art Literature Comparison

To contextualize this work's achievements, Table 4.2 presents a comprehensive benchmark comparison against recent state-of-the-art high-speed wireline transceivers and CDR circuits published in flagship IEEE conferences and journals (ISSCC and JSSC).

Table 4.2: State-of-the-Art Performance Benchmarking Matrix with Literature.

Performance Metric	**Lin et al. [1] (ISSCC'24/JSSC'25)**	**Zhang et al. [28] (ISSCC'26)**	**Park et al. [3] (ISSCC'23)**	**Zhao et al. [29] (JSSC'22)**	**Poon et al. [30] (JSSC'22)**	**Proposed Work (This Thesis)**
Technology Node	28-nm CMOS	28-nm CMOS	28-nm CMOS	28-nm CMOS	7-nm FinFET	**Verilog-A**
Data Rate (Gbps)	112	112	52	58.8	112	**112**
Modulation Format	PAM-4	PAM-4	PAM-4	PAM-4	PAM-4	**PAM-4**
Clocking Architecture	Quarter-Rate (14 GHz)	Quarter-Rate (14 GHz)	Quarter-Rate (13 GHz)	Full-Rate (29.4 GHz)	Quarter-Rate (14 GHz)	**Quarter-Rate (14 GHz)**
Phase Detector (PD) Type	Unbalanced Baud-Rate MMPD	Symmetrical Linear PD	Pattern-Based Baud-Rate	Bang-Bang PD	ADC-DSP / Digital	**Unbalanced Baud-Rate MMPD**
Reference-less Operation	No	Yes	No	Yes	No	**Yes (Dual-Mode DLF)**
Channel Nyquist Loss	8.5 dB @ 28 GHz	9.1 dB @ 28 GHz	5.6 dB @ 13 GHz	12.0 dB @ 14.7 GHz	10.0 dB @ 28 GHz	**8.5 – 10.0 dB @ 28 GHz**
Recovered Clock Jitter (fs_rms)	N/A (ADC-based)	343.0 fs_rms	N/A	416.0 fs_rms	N/A (DSP-based)	**446.7 fs_rms**
High-Freq JTOL (UI_pp)	0.08 @ 50 MHz	0.11 @ 100 MHz	0.12 @ 100 MHz	0.10 @ 100 MHz	N/A	**0.08 @ 100 MHz**
Key Innovation Summary	Analog front-end non-linearity compensation	Symmetrical charge-pump linear phase detector	Pattern-based phase detector for short-reach	High-speed full-rate bang-bang topology	ADC-DSP architecture in advanced FinFET	**Unbalanced MMPD + 4:1 voter spatial decimation**

5. Conclusion and Future Work

5.1 Summary of Contributions

This thesis presents the architectural design, discrete-time z-domain mathematical modeling, and high-fidelity Cadence Verilog-A behavioral verification of a 112 Gbps (56 GBaud) quarter-rate All-Digital Clock and Data Recovery (ADCDR) system designed for Extra Short Reach (XSR) optical and die-to-die (D2D) interconnects. The proposed receiver operates over lossy physical channels with up to 10 dB of insertion loss at the 28 GHz Nyquist frequency. It replaces energy-intensive ADC-DSP structures and leaky analog charge pumps with a synthesizable, power-efficient, all-digital feedback loop.

The key architectural, mathematical, and verification contributions of this research are summarized as follows:

- **Equalized Analog Front-End (AFE) Behavioral Suite**: A dual-pole, single-zero Continuous-Time Linear Equalizer (CTLE) behavioral model was developed in Verilog-A. It delivers 10 dB of high-frequency peaking boost at 28 GHz while maintaining a strict 0.0 V DC common-mode level. A Variable Gain Amplifier (VGA) module was modeled to provide a dynamic gain-control range of -6 dB to +14 dB across a -3 dB bandwidth of $\geq$40 GHz. The VGA also demonstrates DC common-mode level shifting from 0.0 V to 0.5 V, optimizing dynamic latch comparator transconductance in FinFET nodes while regulating the output signal envelope to a target peak differential swing of ±150 mV. A 7-slicer array of 3 Data Slicers and 4 Error Slicers was deployed to digitize the 4-level PAM-4 signal at a quarter-rate clock frequency of 14 GHz.
- **Unbalanced 3-Symbol Pattern Mueller-Müller Phase Detector**: An unbalanced MMPD timing recovery topology was formulated to evaluate 3 consecutive data decisions

(D[n−1], D[n], D[n+1]). By incorporating 12 symmetric patterns (weight W = 1.0) and 16 asymmetric monotonic patterns (confidence weight W = 0.5) while masking 36 non-monotonic inner bounces (weight W = 0.0), the active transition density increased from 18.75% to 31.25%.

- **Closed-Loop System Validation**: Event-driven discrete-time simulation in MATLAB and transient Specter analysis demonstrated robust phase locking within t_{lock} = 50 ns, which equals 2,800 unit intervals (UIs).
- **Discrepancies in MATLAB and the Verilog**-A:

  MATLAB's continuous, time-averaged S-curve approximation oversimplifies phase-detector dynamics. By smoothing transient phase errors, it predicts an overly optimistic and unrealistically fast lock time. In contrast, Cadence Verilog-A models discrete dither, which slows phase acquisition and produces a noticeably longer, more realistic locking trajectory in Virtuoso.

  During frequency-step acquisition, the locking-time gap widens significantly. MATLAB's idealized equations predict rapid, smooth settling, whereas Verilog-A accounts for real digital latency from the 2:16 deserializer, 6-bit DAC quantization limits, and beat-frequency aliasing. These physical constraints lead to transient overshoot and phase ringing before settling. This explains why Cadence takes longer to lock on in-range steps (50 MHz) and may fail to lock entirely due to cycle slipping on out-of-range steps (150 MHz).

5.2 Future Research Directions

While this thesis successfully validates the system architecture, discrete-time dynamics, and Verilog-A behavioral performance of the 112 Gbps PAM-4 ADCDR, several promising avenues remain for future research and hardware implementation:

- **Transistor-Level FinFET/GAAFET CMOS Implementation**: The logical next phase is the physical transistor-level layout and parasitic-extracted (PEX) corner simulation of the AFE and ADCDR sub-blocks in ultra-deep submicron FinFET (e.g., 12 nm / 7 nm) or Gate-All-Around FET (GAAFET, e.g., 3 nm / 2 nm) node. In physical FinFET silicon, transistor threshold voltage mismatch introduces static offsets in the 7-slicer comparators. Implementing background digital-to-analog converter (DAC) offset calibration loops for each slicer is critical to preserve the error threshold spacing. Replacing the behavioral DCO model with a physical LC-tank oscillator utilizing switched metal-insulator-metal (MIM) capacitor arrays will verify fine-tuning linearity and supply noise sensitivity (PSIJ) [25], [27].
- **Integration of Adaptive Equalization Engines**: Incorporating a closed-loop digital Automatic Gain Control (AGC) engine to dynamically adjust the VGA gain (-6 dB to +14 dB) based on power detector feedback will maintain the target **±150 mV** peak differential output under dynamic channel temperature and voltage variations. Implementing background CTLE zero/pole adaptation algorithms (e.g., Sign-Sign Least Mean Square, SS-LMS) will automatically maximize vertical eye openings. For medium-reach (MR) channels exhibiting reflections or residual post-cursor inter-symbol interference, integrating speculative 1-tap or 2-tap Decision Feedback Equalizers (DFE) into the slicer decision path will extend channel loss compensation beyond 15 dB to 20 dB.

- **DSP-Based ADC+CDR Architectures for Long Reach (LR) Channels**: For backplane and server rack applications exceeding 28 dB insertion loss at 28 GHz, pure analog/digital baud-rate CDRs reach physical equalization limits. Transitioning to an ADC-DSP architecture utilizing a 6-bit to 7-bit 56 GS/s Time-Interleaved Successive Approximation Register (TI-SAR) ADC will allow complex digital signal processing algorithms (e.g., 5-tap FFE, 12-tap DFE, and Maximum Likelihood Sequence Estimation, MLSE) to compensate for extreme channel dispersion. Exploring Minimum Mean Square Error (MMSE) digital baud-rate timing error extraction within the DSP core will maintain low clock jitter and fast phase acquisition in 200G/lane and 400G/lane [25],[26].

References

[1] Y. -P. Lin, P. -J. Peng, C. -C. Lu, P. -T. Shen, Y. -C. Jao and P. -H. Hsieh, "A 2.16-pJ/b 112-Gb/s PAM-4 Transceiver With Time-Interleaved 2-b/3-b ADCs and Unbalanced Baud-Rate CDR for XSR Applications in 28-nm CMOS," in *IEEE Journal of Solid-State Circuits*, vol. 61, no. 2, pp. 423-433, Feb. 2026, doi: 10.1109/JSSC.2025.3562885.

[2] C. Zhang, F. Chen, L. Wang, L. Wang and C. Patrick Yue, "Recent Advances of High-Speed Short-Reach Optical Interconnects for Data Centers," in *IEEE Open Journal of the Solid-State Circuits Society*, vol. 5, pp. 86-100, 2025,doi:10.1109/OJSSCS.2025.3526132

[3] S. Park *et al.*, "A 52-Gb/s Low-Power PAM-4 Baud-Rate CDR Using Pattern-Based Phase Detector for Short-Reach Applications," in *IEEE Journal of Solid-State Circuits*, vol. 60, no. 8, pp. 2794-2806, Aug. 2025, doi: 10.1109/JSSC.2024.3517841.

[4] OIF Implementation Agreement, "CEI-112G-XSR-PAM4 Extra Short Reach Interface," OIF-CEI-05.0, Clause 20, 2022.

[5] K. Yadav, P. -H. Hsieh and A. C. Carusone, "Loop Dynamics Analysis of PAM-4 Mueller Muller Clock and Data Recovery System," in *IEEE Open Journal of Circuits and Systems*, vol. 3, pp. 216-227, 2022, doi: 10.1109/OJCAS.2022.3211844.

[6] C. -C. Lu, P. -H. Hsieh, Y. -P. Lin, Y. -C. Jao, W. -H. Hsieh and P. -J. Peng, "Unbalanced MMPD for CDR in Multi-PAM High-Speed Wireline Applications," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 73, no. 5, pp. 3643-3652, May 2026, doi: 10.1109/TCSI.2025.3640732.

[7] IEEE Standard for Ethernet, "IEEE Std 802.3ck-2022: Physical Layer Specifications and Management Parameters for 100 Gb/s, 200 Gb/s, and 400 Gb/s Operation Based on 100 Gb/s per Lane," 2022.

[8] M. M. H. Khanghah, "Architectural and Circuit Design Techniques for High-Speed Wireline Transceivers," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Toronto, Toronto, ON, Canada, 2022.

[9] T. Musah and A. Namachivayam, "Robust Timing Error Detection for Multilevel Baud-Rate CDR," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 10, pp. 3927-3939, Oct. 2022, doi: 10.1109/TCSI.2022.3191740.

[10] B. Razavi, Design of Integrated Circuits for Optical Communications, 2nd ed. New York, NY, USA: John Wiley & Sons, 2012.

[11] O. Tyshchenko, "Subsampling All-Digital Clock and Data Recovery Circuits for High-Speed Optical Networks," Ph.D. dissertation, Ghent University, Ghent, Belgium, 2020.

[12] M. Verbeke, G. Torfs and P. Rombouts, "The Truth About 2-Level Transition Elimination in Bang-Bang PAM-4 CDRs," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 1, pp. 469-482, Jan. 2021, doi: 10.1109/TCSI.2020.3033202.

[13] M. Talegaonkar, R. Inti and P. K. Hanumolu, "Digital clock and data recovery circuit design: Challenges and tradeoffs," *2011 IEEE Custom Integrated Circuits Conference (CICC)*, San Jose, CA, USA, 2011, pp. 1-8, doi: 10.1109/CICC.2011.6055346.

[14] K. Mueller and M. Muller, "Timing Recovery in Digital Synchronous Data Receivers," in *IEEE Transactions on Communications*, vol. 24, no. 5, pp. 516-531, May 1976, doi: 10.1109/TCOM.1976.1093326.

[15] J. Liang, A. Sheikholeslami, H. Tamura, Y. Ogata and H. Yamaguchi, "Loop Gain Adaptation for Optimum Jitter Tolerance in Digital CDRs," in *IEEE Journal of Solid-State Circuits*, vol. 53, no. 9, pp. 2696-2708, Sept. 2018, doi: 10.1109/JSSC.2018.2839038.

[16] W. Chen *et al.*, "A 112-Gb/s PAM-4 Optical Receiver With Sub-1.9- μArms Noise and <4% THD for Linear-Drive Pluggable Optics," in *IEEE Microwave and Wireless Technology Letters*, vol. 35, no. 7, pp. 1089-1092, July 2025, doi: 10.1109/LMWT.2025.3550180.

[17] F. Spagna *et al.*, "A 78mW 11.8Gb/s serial link transceiver with adaptive RX equalization and baud-rate CDR in 32nm CMOS," *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, San Francisco, CA, USA, 2010, pp. 366-367, doi: 10.1109/ISSCC.2010.5433823.

[18] P. K. Hanumolu, M. G. Kim, G. -y. Wei and U. -k. Moon, "A 1.6Gbps Digital Clock and Data Recovery Circuit," *IEEE Custom Integrated Circuits Conference 2006*, San Jose, CA, USA, 2006, pp. 603-606, doi: 10.1109/CICC.2006.320829.

[19] D. -H. Kwon, M. Kim, S. -G. Kim and W. -Y. Choi, "A 32-Gb/s PAM-4 Quarter-Rate Clock and Data Recovery Circuit With an Input Slew-Rate Tolerant Selective Transition Detector," in *IEEE Transactions*

on Circuits and Systems II: Express Briefs, vol. 66, no. 3, pp. 362-366, March 2019, doi: 10.1109/TCSII.2018.2855692.

[20] T. Chan Carusone, T. O. Dickson, S. Palermo, S. Shekhar and M. Mansuri, "Modern Wireline Transceivers," in *IEEE Journal of Solid-State Circuits*, vol. 61, no. 2, pp. 395-422, Feb. 2026, doi: 10.1109/JSSC.2025.3642231.

[21] J. L. Sonntag and J. Stonick, "A Digital Clock and Data Recovery Architecture for Multi-Gigabit/s Binary Links," in *IEEE Journal of Solid-State Circuits*, vol. 41, no. 8, pp. 1867-1875, Aug. 2006, doi: 10.1109/JSSC.2006.875292.

[22] M. Verbeke *et al.*, "A 1.8-pJ/b, 12.5–25-Gb/s Wide Range All-Digital Clock and Data Recovery Circuit," in *IEEE Journal of Solid-State Circuits*, vol. 53, no. 2, pp. 470-483, Feb. 2018, doi: 10.1109/JSSC.2017.2755690.

[23] I. -F. Chen, R. -J. Yang and S. -I. Liu, "Loop latency reduction technique for all-digital clock and data recovery circuits," *2009 IEEE Asian Solid-State Circuits Conference*, Taipei, Taiwan, 2009, pp. 309-312, doi: 10.1109/ASSCC.2009.5357247.

[24] F. Gardner, "Charge-Pump Phase-Lock Loops," in *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1849-1858, November 1980, doi: 10.1109/TCOM.1980.1094619.

[25] A. Khairi *et al.*, "Beyond 200-Gb/s PAM4 ADC and DAC-Based Transceiver for Wireline and Linear Optics Applications," in *IEEE Open Journal of the Solid-State Circuits Society*, vol. 4, pp. 265-276, 2024, doi: 10.1109/OJSSCS.2024.3501975.

[26] X. Luo *et al.*, "7.5 A 224Gb/s/wire Single-Ended PAM-4 Transceiver Front-End with 29dB Equalization for 800GbE/1.6TbE," *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, 2024, pp. 132-134, doi: 10.1109/ISSCC49657.2024.10454534.

[27] H. Park *et al.*, "A 4.63pJ/b 112Gb/s DSP-Based PAM-4 Transceiver for a Large-Scale Switch in 5nm FinFET," *2023 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, 2023, pp. 5-7, doi: 10.1109/ISSCC42615.2023.10067613.

[28] Z. Zhang, Y. Xu, J. Liu, N. Wu, Z. Zhang and L. Liu, "8.5 A 112Gb/s 0.76pJ/b Reference-Less Mixed-Signal PAM-4 CDR in 28nm CMOS," *2026 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, 2026, pp. 142-144, doi: 10.1109/ISSCC49663.2026.11409236.

[29] X. Zhao, Y. Chen, P. -I. Mak and R. P. Martins, "A 0.0285-mm2 0.68-pJ/bit Single-Loop Full-Rate Bang-Bang CDR Without Reference and Separate FD Pulling Off an 8.2-Gb/s/μs Acquisition Speed of the PAM-4 Input in 28-nm CMOS," in *IEEE Journal of Solid-State Circuits*, vol. 57, no. 2, pp. 546-561, Feb. 2022, doi: 10.1109/JSSC.2021.3113773.

[30] C. F. Poon *et al.*, "A 1.24-pJ/b 112-Gb/s (870 Gb/s/Mm) Transceiver for In-Package Links in 7-nm FinFET," in *IEEE Journal of Solid-State Circuits*, vol. 57, no. 4, pp. 1199-1210, April 2022, doi: 10.1109/JSSC.2022.3141802.

A MATLAB

```
% ========================================================================
%
% Description: Open Loop, PM, and Jitter Tolerance
%
% ========================================================================

clear; clc; close all;

% --- 1. SYSTEM PARAMETER DEFINITIONS ---
f_baud        = 56.0e9;               % 56.0 GBaud PAM-4
R_b           = 2 * f_baud;           % 112.0 Gbps
T_UI          = 1 / f_baud;           % 17.857 ps
f_clk         = f_baud / 4;           % 14.0 GHz
T_clk         = 1 / f_clk;            % 71.429 ps

K_MMPD_unbal  = 1.241;                % LSB/rad
K_voter       = 0.7136;                 % 4:1 Majority Voter gain
K_PD_eff      = K_MMPD_unbal * K_voter; % 0.31025 LSB/rad

K_DCO_Hz      = 500.0e6;              % 500 MHz/LSB
K_DCO_rad     = 2 * pi * K_DCO_Hz;    % rad/(s*LSB)

K_p           = 0.125;                  % Proportional gain
K_i_ratio     = 1 / 4096;             % 2^-12 ratio
K_i           = K_p * K_i_ratio;      % Integral gain

N_lat         = 4;                    % 4 cycles @ 14 GHz
tau_lat       = N_lat * T_clk;
tau_total     = (N_lat) * T_clk;

f_bw_calc     = (K_PD_eff * K_p * K_DCO_rad) / (2 * pi);
f_z_calc      = K_i_ratio / (2 * pi * T_clk);           % 0.544 MHz

% --- 2. Z-DOMAIN OPEN-LOOP BODE EVALUATION ---
f_freq = logspace(5, log10(f_clk/2), 50000);
w_freq = 2 * pi * f_freq;
z_vec  = exp(1i * w_freq * T_clk);

H_DLF_z = K_p + K_i ./ (1 - z_vec.^(-1));
H_DCO_z = (K_DCO_rad * T_clk) ./ (1 - z_vec.^(-1));
T_z     = K_PD_eff .* H_DLF_z .* (z_vec.^(-N_lat)) .* H_DCO_z;

mag_T_dB    = 20 * log10(abs(T_z));
phase_T_deg = unwrap(angle(T_z)) * (180 / pi);
phase_T_deg(phase_T_deg > 0) = phase_T_deg(phase_T_deg > 0) - 360;

[~, idx_cross] = min(abs(abs(T_z) - 1.0));
f_bw_sim       = f_freq(idx_cross);
phase_cross    = phase_T_deg(idx_cross);
PM_sim         = 180 + phase_cross;

% --- 3. CLOSED-LOOP JITTER TRANSFER & JTOL ---
```

```
H_z        = T_z ./ (1 + T_z);
mag_H_dB   = 20 * log10(abs(H_z));
peaking_dB = max(mag_H_dB);

f_jtol     = logspace(5, log10(500e6), 2000);
w_jtol     = 2 * pi * f_jtol;
z_jtol     = exp(1i * w_jtol * T_clk);

H_DLF_jtol = K_p + K_i ./ (1 - z_jtol.^(-1));
H_DCO_jtol = (K_DCO_rad * T_clk) ./ (1 - z_jtol.^(-1));
T_jtol     = K_PD_eff .* H_DLF_jtol .* (z_jtol.^(-N_lat)) .* H_DCO_jtol;
H_jtol     = T_jtol ./ (1 + T_jtol);

phi_TM_UI       = 0.08;
JTOL_modeled_UI = phi_TM_UI ./ abs(1 - H_jtol);

JTOL_spec_UI = zeros(size(f_jtol));
for k = 1:length(f_jtol)
    fk = f_jtol(k);
    if fk < 42.18e3
        JTOL_spec_UI(k) = 5.0;
    elseif fk <= 4.0e6
        JTOL_spec_UI(k) = 5.0 * (42.18e3 / fk);
    else
        JTOL_spec_UI(k) = 0.05;
    end
end

% --- 4. TRANSIENT STEP RESPONSE ---
N_sim        = 1500;
t_time       = (0:N_sim-1) * T_clk;
t_ns         = t_time * 1e9;
phi_in_step  = ones(1, N_sim);
phi_out_step = zeros(1, N_sim);
phi_e_step   = zeros(1, N_sim);
accum_step   = 0;
d_ctrl_step  = zeros(1, N_sim);

for n = 2:N_sim
    phi_e_step(n) = phi_in_step(n) - phi_out_step(n-1);
    accum_step    = accum_step + K_i * phi_e_step(n);

    n_del = n - N_lat;
    if n_del >= 1
        d_ctrl_step(n) = K_p * phi_e_step(n_del) + accum_step;
    else
        d_ctrl_step(n) = 0;
    end

    phi_out_step(n) = phi_out_step(n-1) + (K_DCO_Hz * T_clk) * d_ctrl_step(n-1);
end

phase_err_val  = abs(phi_in_step - phi_out_step);
settle_indices = find(phase_err_val <= 0.02);
if ~isempty(settle_indices)
```

```
    lock_cycle = settle_indices(1);
    t_lock_ns  = t_ns(lock_cycle);
else
    lock_cycle = N_sim;
    t_lock_ns  = t_ns(end);
end

% --- 5. GENERATE & SAVE INDIVIDUAL FIGURES SAFELY ---

% FIGURE 1: Bode & PM
fig1 = figure('Name', 'Figure 1: Open-Loop Bode Plot', 'Color', 'w');
subplot(2, 1, 1);
semilogx(f_freq / 1e6, mag_T_dB, 'b-', 'LineWidth', 2.0); hold on; grid on; grid
minor;
yline(0, 'k--', '0 dB Crossover', 'LineWidth', 1.0);
xline(f_bw_sim / 1e6, 'm--', sprintf('f_{bw} = %.2f MHz', f_bw_sim/1e6),
'LineWidth', 1.2);
xline(f_z_calc / 1e6, 'g:', sprintf('f_z = %.3f MHz', f_z_calc/1e6), 'LineWidth',
1.5);
xlim([0.1, f_clk/2/1e6]); ylim([-40, 60]);
xlabel('Frequency (MHz)', 'FontSize', 10, 'FontWeight', 'bold');
ylabel('Magnitude |T(z)| (dB)', 'FontSize', 10, 'FontWeight', 'bold');
title('112-Gb/s ADCDR Open-Loop Bode Magnitude Response', 'FontSize', 11,
'FontWeight', 'bold');

subplot(2, 1, 2);
semilogx(f_freq / 1e6, phase_T_deg, 'r-', 'LineWidth', 2.0); hold on; grid on;
grid minor;
yline(-180, 'k--', '-180^\circ Limit', 'LineWidth', 1.0);
xline(f_bw_sim / 1e6, 'm--', sprintf('f_{bw} = %.2f MHz', f_bw_sim/1e6),
'LineWidth', 1.2);
plot(f_bw_sim / 1e6, phase_cross, 'ro', 'MarkerSize', 7, 'MarkerFaceColor', 'r');
xlim([0.1, f_clk/2/1e6]); ylim([-240, 0]);
xlabel('Frequency (MHz)', 'FontSize', 10, 'FontWeight', 'bold');
ylabel('Unwrapped Phase \angle T(z) (deg)', 'FontSize', 10, 'FontWeight', 'bold');
title(sprintf('Open-Loop Phase Response & Margin (PM = %.2f^\\circ)', PM_sim),
'FontSize', 11, 'FontWeight', 'bold');

% Safe Save Rule: save active figure (gcf) or verify graphics handle
if isgraphics(fig1)
    saveas(fig1, 'fig1_bode_phase_margin.png');
end

% FIGURE 2: Jitter Transfer
fig2 = figure('Name', 'Figure 2: Jitter Transfer', 'Color', 'w');
semilogx(f_freq / 1e6, mag_H_dB, 'b-', 'LineWidth', 2.2); hold on; grid on; grid
minor;
yline(0.10, 'r--', 'OIF-CEI Peaking Spec Limit (< 0.10 dB)', 'LineWidth', 1.5);
yline(0.00, 'k:', '0 dB Passband', 'LineWidth', 1.0);
xline(f_bw_sim / 1e6, 'm--', sprintf('f_{bw} = %.2f MHz', f_bw_sim/1e6),
'LineWidth', 1.2);
xlim([0.1, f_clk/2/1e6]); ylim([-1, 1]);
xlabel('Jitter Frequency (MHz)', 'FontSize', 11, 'FontWeight', 'bold');
ylabel('Jitter Transfer |H(f)| (dB)', 'FontSize', 11, 'FontWeight', 'bold');
```

```
title(sprintf('Closed-Loop Jitter Transfer H(f) (Peaking M_p = +%.4f dB)',
peaking_dB), 'FontSize', 12, 'FontWeight', 'bold');

if isgraphics(fig2)
    saveas(fig2, 'fig2_jitter_transfer.png');
end

% FIGURE 3: JTOL
fig3 = figure('Name', 'Figure 3: Jitter Tolerance', 'Color', 'w');
loglog(f_jtol / 1e6, JTOL_modeled_UI, 'b-', 'LineWidth', 2.2); hold on; grid on;
grid minor;
loglog(f_jtol / 1e6, JTOL_spec_UI, 'r--', 'LineWidth', 2.0);
xlim([0.1, 500]); ylim([0.02, 20]);
xlabel('Jitter Frequency (MHz)', 'FontSize', 11, 'FontWeight', 'bold');
ylabel('Jitter Tolerance (UI_{pp})', 'FontSize', 11, 'FontWeight', 'bold');
title('Jitter Tolerance (JTOL) vs. OIF-CEI-112G-XSR Spec Mask', 'FontSize', 12,
'FontWeight', 'bold');

if isgraphics(fig3)
    saveas(fig3, 'fig3_jitter_tolerance.png');
end

% FIGURE 4: Transient Step
fig4 = figure('Name', 'Figure 4: Step Response', 'Color', 'w');
plot(t_ns, phi_in_step, 'k--', 'LineWidth', 1.5); hold on; grid on; grid minor;
plot(t_ns, phi_out_step, 'b-', 'LineWidth', 2.0);
yline(1.02, 'r:', 'LineWidth', 1.0);
yline(0.98, 'r:', '\pm 0.02 UI Settling Bounds', 'LineWidth', 1.0);
xline(t_lock_ns, 'm-.', sprintf('Settled = %.2f ns', t_lock_ns), 'LineWidth',
1.5);
xlim([0, max(t_ns)]); ylim([-0.1, 1.25]);
xlabel('Time (ns)', 'FontSize', 11, 'FontWeight', 'bold');
ylabel('Clock Phase (UI)', 'FontSize', 11, 'FontWeight', 'bold');
title('Time-Domain Transient Step Phase Response Acquisition', 'FontSize', 12,
'FontWeight', 'bold');

if isgraphics(fig4)
    saveas(fig4, 'fig4_transient_step_response.png');
end

disp('All figures plotted and saved successfully without handle errors!');
```

```
% ========================================================================
%
% Title : Discrete-Time z-Domain Step Phase and Step Frequency Response
%         Modeling for 112-Gbps PAM-4 All-Digital CDR (ADCDR)
% Parameters:
%   K_voter = 0.7136, K_p = 0.125, K_i = K_p / 4096 = 3.05176e-5
%   K_dco = 500 MHz/LSB, Delay Operator D = z^-4 (D = 4)
% ========================================================================

clear; clc; close all;

fprintf('========================================================================\n');
fprintf('  112-Gbps PAM-4 ADCDR z-Domain Step Phase & Frequency Response    \n');
fprintf('========================================================================\n');

%% 1. SYSTEM PARAMETERS & ARCHITECTURAL DEFINITIONS
f_baud        = 56.0e9;               % Symbol Rate: 56.0 GBaud PAM-4 (112 Gbps)
R_b           = 2 * f_baud;           % Bit Rate: 112.0 Gbps
T_UI          = 1 / f_baud;           % Unit Interval: 17.8571 ps
f_clk         = f_baud / 4;           % Quarter-Rate Clock: 14.0 GHz
T_clk         = 1 / f_clk;            % Quarter-Rate Clock Period: 71.4286 ps

% Phase Detector, Voter & DCO Gains
K_MMPD_unbal  = 1.241;                % Linearized Unbalanced MMPD Gain (LSB/rad)
K_voter       = 0.7136;               % Spatial Majority Voter Gain
K_PD_eff      = K_MMPD_unbal * K_voter; % Effective Detector Gain = 0.88558
LSB/rad

K_DCO_Hz      = 500.0e6;              % Fine DCO Sensitivity: 500 MHz/LSB
K_DCO_rad     = 2 * pi * K_DCO_Hz;    % Fine DCO Gain: 3.1416e9 rad/(s*LSB)
K_DCO_z       = K_DCO_rad * T_clk;    % Normalized DCO Phase Step = 0.2244 rad/LSB

% Digital Loop Filter (DLF) Gains
K_p           = 0.125;                % Proportional Gain (2^-3)
K_i_ratio     = 1 / 4096;             % Integral-to-Proportional Ratio (2^-12)
K_i           = K_p * K_i_ratio;      % Integral Gain = 3.0517578e-5 LSB/LSB (2^-
15)

% Retiming Pipeline Latency (Delay Operator D = z^-4, D = 4)
D             = 4;                    % Delay operator D = z^-4 (4 cycles @ 14
GHz)
tau_lat       = D * T_clk;            % Physical transport delay = 285.71 ps

% Closed-Loop System Bandwidth & Time Constant
f_bw          = (K_PD_eff * K_p * K_DCO_rad) / (2 * pi); % f_bw = 55.35 MHz
f_z           = K_i_ratio / (2 * pi * T_clk);           % Loop Zero = 0.544 MHz
tau_loop      = 1 / (2 * pi * f_bw);                    % Closed-loop time
constant = 2.876 ns

fprintf('Quarter-Rate Clock Period (T_clk): %.3f ps\n', T_clk*1e12);
fprintf('Effective Detector Gain (K_eff)  : %.5f LSB/rad\n', K_PD_eff);
fprintf('Proportional Loop Gain (K_p)     : %.6f LSB/LSB\n', K_p);
fprintf('Integral Accumulator Gain (K_i)   : %.9f LSB/LSB\n', K_i);
fprintf('Fine DCO Sensitivity (K_dco)     : %.1f MHz/LSB\n', K_DCO_Hz/1e6);
fprintf('Pipeline Delay Operator (D)       : z^-%d (%.2f ps)\n', D, tau_lat*1e12);
```

```
fprintf('Loop Crossover Bandwidth (f_bw)   : %.2f MHz\n', f_bw/1e6);
fprintf('Closed-Loop Time Constant (tau)   : %.3f ns\n', tau_loop*1e9);
fprintf('----------------------------------------------------------------------
\n\n');


%% 2. Z-DOMAIN CHARACTERISTIC POLE MAP & STABILITY
K_gain = K_PD_eff * K_DCO_z;

% Characteristic Polynomial P(z) = z^4 * (z - 1)^2 + K_gain * [(K_p + K_i)z - K_p]
= 0
% P(z) = z^6 - 2*z^5 + z^4 + K_gain*(K_p + K_i)*z - K_gain*K_p = 0
P_coeffs = [1.0, -2.0, 1.0, 0.0, K_gain*(K_p + K_i), -K_gain*K_p, 0.0];
z_roots  = roots(P_coeffs);

fprintf('z-Domain Characteristic Roots Summary (D = 4):\n');
for m = 1:length(z_roots)
    fprintf('  z_%d = %+.6f + j%+.6f  (|z| = %.6f)\n', ...
        m, real(z_roots(m)), imag(z_roots(m)), abs(z_roots(m)));
end
fprintf('----------------------------------------------------------------------
\n\n');


%% 3. TIME-DOMAIN DISCRETE-TIME RECURSIVE SIMULATIONS
N_sim     = 500;                    % Simulation length: 500 clock cycles
n_vec     = 0:N_sim-1;
t_ns      = n_vec * T_clk * 1e9;    % Time in nanoseconds

% --- A. STEP PHASE RESPONSE SIMULATION (\Delta\phi_0 = +0.35 UI) ---
delta_phi_0      = 0.35;            % Phase Step Amplitude in UI
phi_in_phase     = delta_phi_0 * ones(1, N_sim);
phi_out_phase    = zeros(1, N_sim);
phi_err_phase    = zeros(1, N_sim);
phi_err_phase(1) = delta_phi_0;
accum_phase      = 0.0;
d_ctrl_phase     = zeros(1, N_sim);

for n = 2:N_sim
    phi_err_phase(n) = phi_in_phase(n) - phi_out_phase(n-1);

    e_pd_lsb = K_PD_eff * (2 * pi * phi_err_phase(n));
    accum_phase = accum_phase + K_i * e_pd_lsb;
    accum_phase = max(min(accum_phase, 256.0), -256.0); % Clamping

    % Delay operator D = z^-4 (4 cycles delay)
    n_del = n - D;
    if n_del >= 1
        e_pd_del_lsb = K_PD_eff * (2 * pi * phi_err_phase(n_del));
        d_ctrl_phase(n) = K_p * e_pd_del_lsb + accum_phase;
    else
        d_ctrl_phase(n) = 0.0;
    end
```

```
    phi_out_phase(n) = phi_out_phase(n-1) + d_ctrl_phase(n-1) * (K_DCO_Hz * T_clk
* 2.0);
end

% Overdamped Exponential Envelope Model
phi_err_envelope = delta_phi_0 * exp(-t_ns / (tau_loop * 1e9));

% Phase Lock Settling Index (+/- 0.02 UI bounds)
lock_idx_phase = find(abs(phi_err_phase(10:end)) <= 0.02, 1, 'first') + 9;
if isempty(lock_idx_phase), lock_idx_phase = N_sim; end
t_lock_phase_ns = t_ns(lock_idx_phase);


% --- B. STEP FREQUENCY RESPONSE SIMULATION (\Delta f = +500 ppm) ---
delta_f_ppm     = 2000.0/1.12;             % Frequency Step in ppm
delta_f_Hz      = delta_f_ppm * 1e-6 * 28.0e9; % +14.0 MHz Nyquist offset
phi_ramp_inc    = (delta_f_Hz / 28.0e9) * (T_clk / T_UI); % Phase ramp per cycle

phi_in_freq     = cumsum(phi_ramp_inc * ones(1, N_sim));
phi_out_freq    = zeros(1, N_sim);
phi_err_freq    = zeros(1, N_sim);
accum_freq      = 0.0;
d_ctrl_freq     = zeros(1, N_sim);
f_dco_offset_MHz= zeros(1, N_sim);

target_ftw_shift = delta_f_Hz / K_DCO_Hz; % +0.028 LSB FTW shift at 500 MHz/LSB

for n = 2:N_sim
    phi_err_freq(n) = phi_in_freq(n) - phi_out_freq(n-1);

    e_pd_freq_lsb = K_PD_eff * (2 * pi * phi_err_freq(n));
    accum_freq = accum_freq + K_i * e_pd_freq_lsb;
    accum_freq = max(min(accum_freq, 256.0), -256.0);

    % Delay operator D = z^-4
    n_del = n - D;
    if n_del >= 1
        e_pd_del_freq_lsb = K_PD_eff * (2 * pi * phi_err_freq(n_del));
        d_ctrl_freq(n) = K_p * e_pd_del_freq_lsb + accum_freq;
    else
        d_ctrl_freq(n) = 0.0;
    end

    f_dco_offset_MHz(n) = d_ctrl_freq(n) * (K_DCO_Hz / 1e6); % Frequency offset in
MHz
    phi_out_freq(n)     = phi_out_freq(n-1) + d_ctrl_freq(n-1) * (K_DCO_Hz * T_clk
* 2.0);
end

% Frequency Lock Settling Index (+/- 0.5 MHz bounds)
lock_idx_freq = find(abs(f_dco_offset_MHz(10:end) - (delta_f_Hz/1e6)) <= 0.5, 1,
'first') + 9;
if isempty(lock_idx_freq), lock_idx_freq = N_sim; end
t_lock_freq_ns = t_ns(lock_idx_freq);
```

```
%% 4. PUBLICATION-QUALITY FIGURE GENERATION
fig = figure('Name', 'z-Domain Step Phase and Step Frequency Response', ...
             'Color', 'w', 'Position', [100, 100, 1000, 750]);

% ------------------------------------------------------------------------
% SUBPLOT 1: Step Phase Response - Phase Error Trajectory
% ------------------------------------------------------------------------
subplot(2, 2, 1);
plot(t_ns, phi_err_phase, 'b-', 'LineWidth', 2.0, 'DisplayName', 'Discrete
Recursive Simulation'); hold on;
plot(t_ns, phi_err_envelope, 'k:', 'LineWidth', 1.5, 'DisplayName', 'Exponential
Envelope (\tau = 2.88 ns)');
yline(0.02, 'm--', '\pm 0.02 UI Lock Bounds', 'LineWidth', 1.0);
yline(-0.02, 'm--', 'LineWidth', 1.0, 'HandleVisibility', 'off');
xline(t_lock_phase_ns, 'g-.', sprintf('Phase Lock = %.2f ns', t_lock_phase_ns),
'LineWidth', 1.5);
grid on; grid minor;
xlim([0, 15]); ylim([-0.05, 0.40]);
xlabel('Simulation Time (ns)', 'FontSize', 10, 'FontWeight', 'bold');
ylabel('Phase Error \phi_{err}[n] (UI)', 'FontSize', 10, 'FontWeight', 'bold');
title('(a) Step Phase Response (\Delta\phi_0 = +0.35 UI)', 'FontSize', 11,
'FontWeight', 'bold');
legend('Location', 'northeast', 'FontSize', 8.5);

% ------------------------------------------------------------------------
% SUBPLOT 2: Step Phase Response - Recovered Clock Phase Output
% ------------------------------------------------------------------------
subplot(2, 2, 2);
plot(t_ns, phi_in_phase, 'k--', 'LineWidth', 1.5, 'DisplayName', 'Target Phase
Step (+0.35 UI)'); hold on;
plot(t_ns, phi_out_phase, 'r-', 'LineWidth', 2.0, 'DisplayName', 'Recovered Clock
Phase \phi_{out}[n]');
xline(t_lock_phase_ns, 'g-.', sprintf('Phase Lock = %.2f ns', t_lock_phase_ns),
'LineWidth', 1.5);
grid on; grid minor;
xlim([0, 15]); ylim([-0.05, 0.42]);
xlabel('Simulation Time (ns)', 'FontSize', 10, 'FontWeight', 'bold');
ylabel('Output Phase \phi_{out}[n] (UI)', 'FontSize', 10, 'FontWeight', 'bold');
title('(b) Recovered Clock Phase Tracking', 'FontSize', 11, 'FontWeight', 'bold');
legend('Location', 'southeast', 'FontSize', 8.5);

% ------------------------------------------------------------------------
% SUBPLOT 3: Step Frequency Response - DLF FTW Control Word Output
% ------------------------------------------------------------------------
subplot(2, 2, 3);
plot(t_ns, d_ctrl_freq, 'b-', 'LineWidth', 2.0, 'DisplayName', 'DLF FTW Control
Output D_{ctrl}[n]'); hold on;
yline(target_ftw_shift, 'r--', sprintf('Target Shift (+%.3f LSB)',
target_ftw_shift), 'LineWidth', 1.4);
xline(t_lock_freq_ns, 'g-.', sprintf('Freq Lock = %.2f ns', t_lock_freq_ns),
'LineWidth', 1.5);
grid on; grid minor;
xlim([0, 25]); ylim([-0.2, 0.2]);
xlabel('Simulation Time (ns)', 'FontSize', 10, 'FontWeight', 'bold');
```

```
ylabel('Control Word Output D_{ctrl}[n] (LSB)', 'FontSize', 10, 'FontWeight',
'bold');
title('(c) Step Frequency Response', 'FontSize', 11, 'FontWeight', 'bold');
legend('Location', 'southeast', 'FontSize', 8.5);

% ------------------------------------------------------------------------
% SUBPLOT 4: Step Frequency Response - Recovered Nyquist Frequency Offset
% ------------------------------------------------------------------------
subplot(2, 2, 4);
plot(t_ns, f_dco_offset_MHz, 'r-', 'LineWidth', 2.0, 'DisplayName', 'Recovered DCO
Nyquist Frequency Offset'); hold on;
yline(delta_f_Hz/1e6, 'k--', sprintf('Target Offset (+%.1f MHz)', delta_f_Hz/1e6),
'LineWidth', 1.4);
yline(0, 'k:', 'LineWidth', 1.0, 'HandleVisibility', 'off');
xline(t_lock_freq_ns, 'g-.', sprintf('Freq Lock = %.2f ns', t_lock_freq_ns),
'LineWidth', 1.5);
grid on; grid minor;
xlim([0, 25]); ylim([40, 60]);
xlabel('Simulation Time (ns)', 'FontSize', 10, 'FontWeight', 'bold');
ylabel('Nyquist Frequency Offset f_{DCO} (MHz)', 'FontSize', 10, 'FontWeight',
'bold');
title('(d) DCO Frequency Pull-In & Lock Acquisition', 'FontSize', 11,
'FontWeight', 'bold');
legend('Location', 'southeast', 'FontSize', 8.5);

% Save Figure
output_filename = 'adcdr_zdomain_step_responses.png';
if exist('exportgraphics', 'file')
    exportgraphics(fig, output_filename, 'Resolution', 300);
else
    saveas(fig, output_filename);
end
fprintf('Figure saved as "%s" successfully.\n', output_filename);
```

B Verilog–A

```
// ============================================================================
// Module: xsr_channel_model.va
// Language: Verilog-A
// Description: Numerically Stabilized Differential XSR Channel Model.
//              Guarantees exact target insertion loss at Nyquist (28 GHz).
// ============================================================================
`include "disciplines.vams"
`include "constants.vams"

module xsr_channel_model (vin_p, vin_n, vout_p, vout_n);
   input vin_p, vin_n;
   output vout_p, vout_n;
   electrical vin_p, vin_n, vout_p, vout_n;

   // Architectural Parameters
   parameter real loss_28g_db  = 10.0;        // Target insertion loss at 28 GHz in dB
   parameter real f_nyquist    = 28.0e9;      // Nyquist Frequency (28 GHz)
   parameter real f_p2         = 56.0e9;      // Second pole (56 GHz)
   parameter real f_p3         = 112.0e9;     // Parasitic roll-off pole (112 GHz)
   parameter real r_diff       = 100.0;       // Termination resistance (Ohms)
   parameter real prop_delay   = 80.0e-12;    // Propagation delay (80 ps)
   parameter real v_cm_out     = 0.0;         // Common-mode reference voltage (V)

   // Internal Variables
   real num[0:0];
   real den[0:3];
   real w_nyq, w_p1, w_p2, w_p3;
   real target_mag, fixed_poles_mag, ratio;
   real v_in_diff, v_delayed, v_channel_out;
```

```
analog begin
   @ (initial_step) begin
      w_nyq = 2.0 * `M_PI * f_nyquist;
      w_p2  = 2.0 * `M_PI * f_p2;
      w_p3  = 2.0 * `M_PI * f_p3;

      // Target linear magnitude attenuation at Nyquist
      target_mag = pow(10.0, -abs(loss_28g_db) / 20.0); // 10^(-8/20) = 0.3981

      // Attenuation contributed by fixed higher-order poles (56 GHz and 112 GHz)
      fixed_poles_mag = sqrt(1.0 + pow(w_nyq / w_p2, 2.0)) *
                        sqrt(1.0 + pow(w_nyq / w_p3, 2.0));

      // Solve for w_p1 such that |H(j*w_nyq)| = target_mag
      ratio = pow(1.0 / (target_mag * fixed_poles_mag), 2.0) - 1.0;
      if (ratio > 0.0) begin
         w_p1 = w_nyq / sqrt(ratio); // Solves to f_p1 = 14.458 GHz for 8 dB loss
      end else begin
         w_p1 = w_nyq;
      end

      // Normalized Laplace coefficients (den[0] = 1.0)
      num[0] = 1.0;

      den[0] = 1.0;
      den[1] = (1.0 / w_p1) + (1.0 / w_p2) + (1.0 / w_p3);
      den[2] = (1.0 / (w_p1 * w_p2)) + (1.0 / (w_p2 * w_p3)) + (1.0 / (w_p1 * w_p3));
      den[3] = 1.0 / (w_p1 * w_p2 * w_p3);
   end
```

```
    // Differential Input Termination (100 Ohms)
    I(vin_p, vin_n) <+ V(vin_p, vin_n) / r_diff;

    // Differential Voltage Sensing
    v_in_diff = V(vin_p, vin_n);

    // Transmission Line Delay
    v_delayed = absdelay(v_in_diff, prop_delay);

    // Continuous S-Domain Filtering
    v_channel_out = laplace_nd(v_delayed, num, den);

    // Differential Output Generation
    V(vout_p) <+ v_cm_out + 0.5 * v_channel_out;
    V(vout_n) <+ v_cm_out - 0.5 * v_channel_out;
  end
endmodule
```

```
// ============================================================================
// Module: xsr_afe_ctle.va
// Language: Verilog-A
// Description: Dynamic Continuous-Time Linear Equalizer (CTLE) Model for XSR.
//              Dynamically solves for zero (f_z) to match peaking_db target
//              at Nyquist frequency while maintaining numerical stability.
// ============================================================================
`include "disciplines.vams"
`include "constants.vams"

module xsr_afe_ctle (vin_p, vin_n, vout_p, vout_n);
   input vin_p, vin_n;
   output vout_p, vout_n;
   electrical vin_p, vin_n, vout_p, vout_n;

   // Architectural Parameters
   parameter real g_dc_db     = -3.0;     // DC Gain in dB (-3.0 dB)
   parameter real peaking_db  = 10.0;      // Target Peaking Boost relative to DC gain at 28 GHz (+8.0 dB)
   parameter real f_nyquist   = 28.0e9;    // Nyquist Frequency (28 GHz)
   parameter real f_p1        = 28.0e9;    // Pole 1 Frequency (28 GHz)
   parameter real f_p2        = 56.0e9;    // Pole 2 Frequency (56 GHz)
   parameter real f_p3        = 112.0e9;   // Parasitic Pole 3 Frequency (112 GHz)
   parameter real r_in        = 100.0;     // Input Differential Resistance (100 Ohms)
   parameter real v_cm_out    = 0.0;       // Output Common-Mode Voltage (V)

   // Internal Variables
   real num[0:1];                      // Numerator array [n0, n1] for (n0 + n1*s)
   real den[0:3];                      // Denominator array [d0, d1, d2, d3] for (d0 + d1*s + d2*s^2 + d3*s^3)
   real w_nyq, w_z, w_p1, w_p2, w_p3;
```

```
real g_dc, m_poles, ratio_sq;
real v_in_diff, v_ctle_out;

analog begin
   @ (initial_step) begin
      w_nyq = 2.0 * `M_PI * f_nyquist;
      w_p1  = 2.0 * `M_PI * f_p1;
      w_p2  = 2.0 * `M_PI * f_p2;
      w_p3  = 2.0 * `M_PI * f_p3;

      // Convert DC gain from dB to linear magnitude
      g_dc = pow(10.0, g_dc_db / 20.0); // 10^(-3/20) = 0.70795

      // Calculate pole attenuation magnitude factor at Nyquist
      m_poles = sqrt(1.0 + pow(w_nyq / w_p1, 2.0)) *
            sqrt(1.0 + pow(w_nyq / w_p2, 2.0)) *
            sqrt(1.0 + pow(w_nyq / w_p3, 2.0)); // = 1.6298

      // Solve for w_z to yield exactly peaking_db boost at f_nyquist
      ratio_sq = pow(pow(10.0, abs(peaking_db) / 20.0) * m_poles, 2.0) - 1.0;

      if (ratio_sq > 0.0) begin
         w_z = w_nyq / sqrt(ratio_sq); // Calculates f_z = 7.053 GHz
      end else begin
         w_z = w_nyq;
      end

      // Numerator Polynomial Coefficients N(s) = n0 + n1*s
      num[0] = g_dc;
      num[1] = g_dc / w_z;
```

```
        // Denominator Polynomial Coefficients D(s) = 1 + d1*s + d2*s^2 + d3*s^3
        den[0] = 1.0;
        den[1] = (1.0 / w_p1) + (1.0 / w_p2) + (1.0 / w_p3);
        den[2] = (1.0 / (w_p1 * w_p2)) + (1.0 / (w_p2 * w_p3)) + (1.0 / (w_p1 * w_p3));
        den[3] = 1.0 / (w_p1 * w_p2 * w_p3);
    end

    // Input Termination Resistance
    I(vin_p, vin_n) <+ V(vin_p, vin_n) / r_in;

    // Differential Input Voltage Sensing
    v_in_diff = V(vin_p, vin_n);

    // Continuous S-Domain CTLE Filtering
    v_ctle_out = laplace_nd(v_in_diff, num, den);

    // Differential Output Driver
    V(vout_p) <+ v_cm_out + 0.5 * v_ctle_out;
    V(vout_n) <+ v_cm_out - 0.5 * v_ctle_out;
  end
endmodule
```

```
// Real-time Variable Gain Amplifier (VGA) & Level Shifter for 112G PAM-4 XSR AFE
`include "disciplines.vams"
`include "constants.vams"

module vga_xsr_afe (
   vin_p, vin_n,
   vctrl,
   vout_p, vout_n
);

   // Port Declarations
   input vin_p, vin_n, vctrl;
   output vout_p, vout_n;
   electrical vin_p, vin_n, vctrl, vout_p, vout_n;

   // Architectural Parameters
   parameter real vcm_in      = 0.0;    // Input DC Common-Mode (from CTLE)
   parameter real vcm_out     = 0.5;    // Output DC Common-Mode (for Slicers)
   parameter real gain_min_db  = -6.0;   // Minimum Gain setting (dB)
   parameter real gain_max_db  = 14.0;   // Maximum Gain setting (dB)
   parameter real vctrl_min    = 0.0;    // Control voltage for min gain (V)
   parameter real vctrl_max    = 1.0;    // Control voltage for max gain (V)
   parameter real bw_3db      = 56.0e9;  // -3dB Bandwidth (40 GHz)
   parameter real vsat_diff    = 0.35;   // Differential saturation voltage (V peak)
   parameter real tr_tf       = 1.0p;   // Propagation transition time (ps)

   // Internal Variables & Coefficients
   real v_in_diff;
   real ctrl_clamped;
   real gain_db;
```

```
real gain_lin;
real v_out_linear;
real v_out_sat;
real w_n, Q;

// Laplace transfer function arrays for 2nd-Order Butterworth Bandwidth Roll-off
real num[0:0];
real den[0:2];

analog begin
  @(initial_step("dc", "tran")) begin
    // 2nd-Order Butterworth Filter setup: H(s) = w_n^2 / (s^2 + (w_n/Q)*s + w_n^2)
    w_n = `M_TWO_PI * bw_3db;
    Q   = 0.7071; // Butterworth maximally flat response

    num[0] = w_n * w_n;
    den[0] = w_n * w_n;
    den[1] = w_n / Q;
    den[2] = 1.0;
  end

  // 1. Calculate Differential Input Voltage
  v_in_diff = V(vin_p) - V(vin_n);

  // 2. Bound Control Voltage within limits
  if (V(vctrl) < vctrl_min)
    ctrl_clamped = vctrl_min;
  else if (V(vctrl) > vctrl_max)
    ctrl_clamped = vctrl_max;
  else
```

```
        ctrl_clamped = V(vctrl);

    // 3. Linear-in-dB Gain Calculation
    gain_db  = gain_min_db + ((ctrl_clamped - vctrl_min) / (vctrl_max - vctrl_min)) * (gain_max_db - gain_min_db);
    gain_lin = pow(10.0, gain_db / 20.0);

    // 4. Compute Linear Output Differential Signal
    v_out_linear = v_in_diff * gain_lin;

    // 5. Non-linear Compression (Hyperbolic Tangent Saturation for HD3 modeling)
    // If v_out_linear is small, tanh(x)/x ~ 1. Under large swing, it compresses smoothly to vsat_diff.
    v_out_sat = vsat_diff * limexp(v_out_linear / vsat_diff - pow(v_out_linear / vsat_diff, 3) / 3.0);
    v_out_sat = vsat_diff * tanh(v_out_linear / vsat_diff);

    // 6. Apply Bandwidth Limitation via 2nd-Order Laplace Filter
    v_out_sat = laplace_nd(v_out_sat, num, den);

    // 7. Reconstruct Single-Ended Outputs with Level Shift (DC = 0.5 V)
    V(vout_p) <+ transition(vcm_out + 0.5 * v_out_sat, 0.0, tr_tf, tr_tf);
    V(vout_n) <+ transition(vcm_out - 0.5 * v_out_sat, 0.0, tr_tf, tr_tf);
  end

endmodule
```

```
// Verilog-A Model: 4-Phase Majority Voter for Baud-Rate CDR Loop
// Summarizes 4-phase Early/Late decision outputs into a single E/L decision.

`include "disciplines.vams"
`include "constants.vams"

module majority_voter_4ph (
   input electrical clk,       // Sampling Clock (~14 GHz for 112 Gbps)

   // 4-Phase Early Inputs (0, 90, 180, 270 deg)
   input electrical early0,
   input electrical early90,
   input electrical early180,
   input electrical early270,

   // 4-Phase Late Inputs (0, 90, 180, 270 deg)
   input electrical late0,
   input electrical late90,
   input electrical late180,
   input electrical late270,

   // Majority Decision Outputs to DLF
   output electrical early_out,
   output electrical late_out
);

   // Parameter Declarations
   parameter real vh = 1.0;          // High Logic Voltage (V)
   parameter real vl = 0.0;          // Low Logic Voltage (V)
   parameter real vth = 0.5;         // Digital Input Threshold (V)
```

```
parameter real trise = 1p;         // Output Rise Time
parameter real tfall = 1p;         // Output Fall Time

// Internal Variables
integer num_early;
integer num_late;
real v_early_val;
real v_late_val;

analog begin
   @ (initial_step) begin
      v_early_val = vl;
      v_late_val = vl;
   end

   // Evaluate Majority Vote on the rising edge of system clock
   @ (cross(V(clk) - vth, +1)) begin

      // Tally active Early votes across 4 phases
      num_early = (V(early0)   > vth ? 1 : 0) +
                  (V(early90)  > vth ? 1 : 0) +
                  (V(early180) > vth ? 1 : 0) +
                  (V(early270) > vth ? 1 : 0);

      // Tally active Late votes across 4 phases
      num_late  = (V(late0)    > vth ? 1 : 0) +
                  (V(late90)   > vth ? 1 : 0) +
                  (V(late180)  > vth ? 1 : 0) +
                  (V(late270)  > vth ? 1 : 0);
```

```
        // Decision Logic
        if (num_early > num_late) begin
            // Majority Early -> Signal DLF to retard phase / decrease frequency
            v_early_val = vh;
            v_late_val  = vl;
        end else if (num_late > num_early) begin
            // Majority Late -> Signal DLF to advance phase / increase frequency
            v_early_val = vl;
            v_late_val  = vh;
        end else begin
            // Tie or Zero Votes (Stay condition) -> Maintain loop state
            v_early_val = vl;
            v_late_val  = vl;
        end
    end

    // Continuous signal assignment with transition filtering
    V(early_out) <+ transition(v_early_val, 0, trise, tfall);
    V(late_out)  <+ transition(v_late_val, 0, trise, tfall);
  end

endmodule
```

```
// Verilog-A Model: Digital Loop Filter (DLF) scaled for 112 Gbps PAM-4

`include "disciplines.vams"
`include "constants.vams"

module dlf_112g (
   input electrical clk,      // Recovered/System Clock (~14 GHz)
   input electrical early_in,  // Early signal from MV/PD
   input electrical late_in,   // Late signal from MV/PD
   output electrical v_pro,    // Proportional control output to DCO
   output electrical v_int     // Integral control output to DCO
);

   // Parameter Declarations
   parameter real vh = 1.0;               // High logic level (V)
   parameter real vl = 0.0;               // Low logic level (V)
   parameter real vth = 0.5;              // Threshold voltage (V)
   parameter integer DESER_RATIO = 16;      // Integral path accumulation ratio (2:16 deser)

   // Gain parameters tuned for 112 Gbps (56 GBaud)
   parameter real K_PRO = 0.05;           // Proportional gain step (V)
   parameter real K_INT = 0.005;          // Integral gain step (V)
   parameter integer INT_MAX = 63;         // 63-code thermometer depth
   parameter integer INT_MIN = 0;

   // Internal Variables
   integer deser_cnt;
   integer int_code;
   real pro_val;
   real int_val;
```

```
integer pd_status; // -1: Early, 0: Stay, +1: Late

analog begin
   @ (initial_step) begin
      deser_cnt = 0;
      int_code = 0;//INT_MAX / 2; // Mid-scale initialization
      pro_val = 0.0;
      int_val = (int_code / (1.0 * INT_MAX)) * vh;
   end

   // Clock rising edge trigger
   @ (cross(V(clk) - vth, +1)) begin
      // Decode Phase Detector status
      if (V(early_in) > vth && V(late_in) <= vth) begin
         pd_status = -1; // Clock is Early -> Decrease frequency
      end else if (V(late_in) > vth && V(early_in) <= vth) begin
         pd_status = 1;  // Clock is Late  -> Increase frequency
      end else begin
         pd_status = 0;  // Stay status
      end

      // Direct Proportional Path update
      pro_val = pd_status * K_PRO;

      // Integral Path with 2:16 Deserializer accumulation
      deser_cnt = deser_cnt + 1;
      if (deser_cnt >= DESER_RATIO) begin
         int_code = int_code + pd_status;

         // Saturate 63-code thermometer range
```

```
            if (int_code > INT_MAX) int_code = INT_MAX;
            if (int_code < INT_MIN) int_code = INT_MIN;

            int_val = (int_code / (1.0 * INT_MAX)) * vh;
            deser_cnt = 0;
          end
        end

        // Output control voltages with fast transition times
        V(v_pro) <+ transition(pro_val, 0, 1p, 1p);
        V(v_int) <+ transition(int_val, 0, 1p, 1p);
    end

endmodule
```

```
// Verilog-A Model: 4-Phase Square-Wave DCO with Phase Jitter for 12nm FinFET
`include "disciplines.vams"
`include "constants.vams"

module dco_12nm_square (
   input electrical v_pro,    // Proportional control input (1V = 1 LSB)
   input electrical v_int,    // Integral control input (1V = 1 LSB)
   output electrical clk0,    // Phase 0 deg
   output electrical clk90,   // Phase 90 deg
   output electrical clk180,  // Phase 180 deg
   output electrical clk270   // Phase 270 deg
);

   // 12nm FinFET Technology & DCO Parameters
   parameter real F0 = 14.0e9;           // Center Frequency: 14 GHz
   parameter real KDCO = 500.0e6;        // Tuning Gain: 500 MHz / LSB
   parameter real VDD = 1;             // 12nm Nominal Supply Voltage (V)
   parameter real VSS = 0.0;             // Ground Reference (V)
   parameter real TRISE = 1.0e-12;       // 6 ps Rise Time (12nm FinFET buffer)
   parameter real TFALL = 1.0e-12;       // 6 ps Fall Time (12nm FinFET buffer)

   // Noise / Jitter Parameters
   parameter real RJ_RMS = 150.0e-15;     // 150 fs RMS Random Period Jitter
   parameter integer EN_NOISE = 1;       // Enable Noise: 1 = ON, 0 = OFF

   // Internal Variables
   real freq;
   real phase;
   real dt;
   real last_time;
```

```
real phase_jitter;
integer seed;

// Logic State Variables
real q0, q90, q180, q270;

analog begin
    @ (initial_step) begin
        phase = 0.0;
        last_time = $abstime;
        seed = 314159; // Initialize random seed
    end

    // Calculate instantaneous frequency f(t)
    freq = F0 + KDCO * (V(v_pro) + V(v_int));

    // Frequency limiting (12.0 GHz to 16.0 GHz)
    if (freq < 12.0e9) freq = 12.0e9;
    if (freq > 16.0e9) freq = 16.0e9;

    // Integrate phase over time
    dt = $abstime - last_time;
    last_time = $abstime;
    phase = phase + 2.0 * `M_PI * freq * dt;

    // Phase wrap and Gaussian Phase Noise / Jitter Injection
    if (phase >= 2.0 * `M_PI) begin
        phase = phase - 2.0 * `M_PI;

        if (EN_NOISE == 1) begin
```

```
            // Convert RMS time jitter (RJ_RMS) to phase jitter (rad):
            // sigma_phase = 2 * PI * f0 * RJ_RMS
            phase_jitter = 2.0 * `M_PI * freq * $rdist_normal(seed, 0, 1) * RJ_RMS;
            phase = phase + phase_jitter;
          end
        end

        // Quadrature Square Wave Thresholding (0 to VDD)
        q0   = (phase >= 0.0 && phase < `M_PI) ? VDD : VSS;
        q90  = (phase >= `M_PI_2 && phase < 3.0 * `M_PI_2) ? VDD : VSS;
        q180 = (phase >= `M_PI) ? VDD : VSS;
        q270 = (phase >= 3.0 * `M_PI_2 || phase < `M_PI_2) ? VDD : VSS;

        // Drive outputs with 12nm FinFET edge transitions
        V(clk0)   <+ transition(q0,   0, TRISE, TFALL);
        V(clk90)  <+ transition(q90,  0, TRISE, TFALL);
        V(clk180) <+ transition(q180, 0, TRISE, TFALL);
        V(clk270) <+ transition(q270, 0, TRISE, TFALL);
    end

endmodule
```