

DESIGN AND ANALYSIS OF A DIGITALLY CALIBRATED MULTI-LANE SERDES ARCHITECTURE FOR ULTRA-HIGH-SPEED VLSI INTERCONNECTS

Chidurala Sharath¹, Ratan Babu Telusoori^{1*}

¹Department of Electronics & Communication Engineering, Vaagdevi Engineering College, Warangal, 506005, Telangana, India.

*Correspondence: Ratan Babu Telusoori (ratan44ab@gmail.com)

Abstract

Ultra-high-speed VLSI interconnects have become a critical enabler for modern computing and communication platforms, where serial data rates have rapidly scaled from a few gigabits per second to beyond 56–112 Gb/s per lane, and aggregated multi-lane links routinely exceed 400 Gb/s in contemporary systems. At the same time, interconnect power can account for 20–40% of total chip I/O power, making energy-efficient and scalable Serializer/Deserializer (SerDes) architectures a key research challenge. Traditional single-lane digitally controlled SerDes architectures, while effective at moderate speeds, face inherent problems at ultra-high data rates, including high power consumption and jitter sensitivity due to Phased Lock Loops (PLL)/ Clock and Data Recovery (CDR) blocks, as well as increased area, calibration complexity, and latency caused by independent clock recovery and equalizer adaptation. To address these limitations, this work proposes a digitally calibrated multi-lane SerDes extension architecture that scales throughput by parallelizing multiple lanes while maintaining signal integrity and timing robustness. A shared hierarchical digital PLL and low-skew clock distribution network synchronizes all transmit lanes, while per-lane programmable Finite Impulse Response (FIR) based transmitter equalizers and hybrid Continuous-Time Linear Equalizer with Decision Feedback Equalizer (CTLE–DFE) receiver equalizers with digital adaptation compensate for lane-specific channel impairments. At the receiver, per-lane digital CDRs with phase interpolators are combined with an elastic buffer-based lane

bonding and deskew unit to correct inter-lane skew and ensure aligned data boundaries. A centralized digital calibration and control engine continuously monitors phase offsets, delay mismatches, and error metrics, dynamically tuning clock phases and equalizer coefficients. It enables linear throughput scaling with controlled power and area overhead, making the architecture well-suited for next-generation ultra-high-speed VLSI interconnect standards.

Key words: Serializer/Deserializer (SerDes), Multi-Lane SerDes, Digital Phase-Locked Loop (DPLL), Clock and Data Recovery (CDR), Finite Impulse Response (FIR) Equalizer, Continuous-Time Linear Equalizer (CTLE), Decision Feedback Equalizer (DFE), Ultra-High-Speed VLSI Interconnects.

1. Introduction

The semiconductor industry has witnessed unprecedented growth over the past decade, driven by the increasing adoption of cloud computing, artificial intelligence, autonomous systems, 5G/6G communications, and edge computing technologies. According to recent market analyses, global data generation is expected to exceed 180 zettabytes annually within the next few years, requiring communication infrastructures capable of handling extremely high data transfer rates. Modern processors often integrate billions of transistors and hundreds of processing cores, creating a substantial demand for high-bandwidth communication channels between computing resources. As a result, efficient on-chip and off-chip interconnect architectures have become a fundamental requirement for next-generation electronic systems.

The rapid expansion of hyperscale data centers has further accelerated the need for advanced high-speed communication technologies. Organizations operating large-scale cloud platforms process billions of transactions and data exchanges every day. To support these workloads, communication standards have evolved from a few gigabits per second to hundreds of gigabits per second per link. Industry roadmaps indicate that future interconnect solutions must deliver higher throughput while simultaneously reducing power consumption and physical area. Consequently, SerDes-based communication architectures have become a key enabling technology for achieving scalable and efficient data movement across modern computing infrastructures.

In addition to data centers, high-speed serial communication plays a critical role in automotive electronics, aerospace systems, industrial automation, medical equipment, telecommunications networks, and consumer electronics. Applications such as autonomous driving, real-time video processing, industrial robotics, and AI inference require continuous transmission of large volumes of data with minimal latency. Maintaining reliable communication under these demanding conditions requires sophisticated clock synchronization, equalization, and timing recovery mechanisms. Therefore, the design and optimization of advanced SerDes architectures continue to be an active area of research and industrial development within the VLSI domain.

2. Literature Survey

Gangadhar, et al. [1] presented a Phase Locked Loop (PLL) VLSI architecture for carrier synchronization in wireless communication systems using 45 nm CMOS technology. The methodology incorporated phase detection, loop filtering, voltage-controlled oscillation, and feedback synchronization mechanisms to achieve stable carrier recovery. The design was simulated under varying frequency conditions

to evaluate locking behavior and synchronization accuracy. Hardware-level optimization techniques were employed to improve frequency stability and reduce synchronization errors. The architecture focused on reliable signal recovery for wireless communication applications. The design mainly targeted wireless carrier synchronization and did not address ultra-high-speed multi-lane SerDes synchronization requirements.

Chen, et al. [2] developed a fast-locking PLL architecture using a joint phase and frequency compensation technique. The methodology combined phase error correction and frequency tracking mechanisms to accelerate lock acquisition. Adaptive compensation circuits were integrated to reduce settling time and improve phase convergence. The design minimized locking delay by dynamically adjusting loop parameters during transient operation. Performance evaluation demonstrated improved lock speed under varying operating conditions. The architecture emphasized lock-time reduction but lacked multi-lane clock distribution and large-scale synchronization capability.

Vimala and VincyLloyd [3] investigated a low-power PLL design methodology focused on reducing power dissipation in high-frequency communication circuits. Their approach optimized phase detector operation, loop filter design, and oscillator structures to achieve energy-efficient synchronization. Simulation studies were conducted to evaluate power consumption under different frequencies. The architecture utilized CMOS design techniques to minimize switching activity and leakage power. Comparative analysis demonstrated improvements in energy efficiency over conventional PLL structures. The work primarily focused on low-power operation and provided limited analysis of jitter performance at ultra-high data rates.

Raju and Papanasam [4] introduced an All-Digital Phase Locked Loop (ADPLL) and

performed comparative evaluation across multiple CMOS technology nodes. The methodology utilized digital phase detection, digitally controlled oscillation, and programmable loop control for frequency synchronization. Performance metrics such as lock time, stability, and frequency accuracy were analyzed under different fabrication technologies. The digital implementation improved scalability and technology portability. The architecture did not incorporate advanced clock distribution strategies required for multi-lane SerDes systems.

Kolur, et al. [5] designed a low-power CMOS PLL using GPDK 90 nm technology and evaluated its performance for communication applications. The methodology included optimized voltage-controlled oscillator design, phase-frequency detection, and feedback control mechanisms. Circuit-level modifications were applied to reduce power consumption while maintaining frequency stability. Performance was verified through transient and frequency-domain simulations. The architecture achieved improved power efficiency compared with traditional PLL implementations. The design was limited to low-power PLL optimization and did not address high-bandwidth serial communication environments.

Karthik, et al. [6] developed a power-supply-noise-aware amplification methodology for voltage-controlled oscillators and PLL-fed MOS operational amplifiers. Their approach analyzed noise propagation effects within PLL subsystems and introduced amplification strategies to mitigate supply-induced disturbances. Advanced noise modeling techniques were incorporated to improve signal integrity. The methodology improved robustness against supply noise variations. The work focused mainly on noise mitigation and did not address clock recovery or lane synchronization challenges.

Singh, et al. [7] implemented a PLL architecture using 18 nm FinFET technology

for high-performance applications. The methodology employed FinFET-based phase detectors, oscillators, and feedback circuits to achieve improved switching characteristics and reduced leakage current. Circuit simulations were conducted to evaluate locking performance and frequency stability. The design leveraged nanoscale transistor properties to improve efficiency. The architecture concentrated on PLL implementation and lacked integrated equalization and SerDes communication support.

Rao, et al. [8] introduced a single-phase clock distribution methodology for low-power VLSI systems. Their approach utilized optimized routing structures and clock distribution networks to reduce skew and power consumption. The design focused on efficient clock propagation across large integrated circuits. Timing analysis was performed to evaluate synchronization accuracy and clock integrity. The architecture demonstrated improved clock distribution efficiency in low-power environments. The methodology was not designed for high-speed multi-lane communication architectures requiring dynamic synchronization.

Varsha and Suresh [9] designed and optimized a low-power PLL using 18 nm FinFET technology for high-performance computing applications. The methodology incorporated optimized oscillator structures, adaptive feedback control, and FinFET-based circuit implementation. Power reduction techniques were applied without significantly affecting synchronization accuracy. Simulation studies validated the design under different frequency conditions. The architecture achieved improved energy efficiency and operational stability. The design mainly emphasized power optimization and did not address large-scale multi-lane SerDes integration.

Li, et al. [10] developed an 11-Gbps CMOS serializer core for high-energy physics experiments. The methodology employed

CMOS logic-based serialization techniques, high-speed clocking mechanisms, and data multiplexing structures to achieve reliable serial transmission. Circuit optimization was performed to support high data rates while maintaining signal integrity. The architecture demonstrated efficient serial data conversion for specialized applications. The serializer focused on single-channel transmission and lacked adaptive equalization and lane-bonding mechanisms.

Zimmer, et al. [11] presented a low-latency self-timed die-to-die communication link for 2.5D and 3D interconnect systems using 3 nm technology. The methodology incorporated self-timed synchronization, energy-efficient signaling, and low-latency communication protocols. Advanced interconnect structures were utilized to improve data transfer efficiency between dies. The architecture minimized communication overhead while maintaining high throughput. The design primarily targeted die-to-die communication and did not address adaptive clock recovery in multi-lane SerDes systems.

Kang, et al. [13] introduced a 96-Gb/s PAM-8 transmitter using a cascaded driver architecture in 40 nm CMOS technology. The methodology utilized high-swing signaling, low-loading driver stages, and PAM-8 modulation techniques to improve transmission efficiency. Driver optimization reduced loading effects while maintaining signal quality. The architecture supported ultra-high-speed communication with enhanced bandwidth utilization. The transmitter architecture lacked receiver-side adaptive equalization and lane synchronization mechanisms.

Bhimashankar and Papanasam [15] designed a power-efficient SERDES architecture aimed at reducing communication energy consumption. Their methodology incorporated serializer and deserializer optimization, efficient clock management, and low-power transmission techniques. Hardware resources were carefully optimized to improve area and energy

efficiency. Performance analysis demonstrated reduced power dissipation compared with conventional SERDES implementations. The design supported reliable serial communication under moderate data rates. The architecture did not provide advanced digital calibration, adaptive equalization, or multi-lane scalability features.

3. Proposed System

Conventional SerDes architectures typically employ a single-lane transmission structure consisting of a PLL, serializer, transmitter equalizer, receiver equalizer, CDR, and deserializer. While such architectures provide reliable communication at moderate data rates, they encounter significant limitations when deployed in modern ultra-high-speed VLSI interconnects operating beyond tens of gigabits per second. The use of independent clock generation and recovery mechanisms introduces substantial power consumption, increased jitter sensitivity, and larger silicon area overhead. Furthermore, single-lane operation restricts throughput scalability because achieving higher bandwidth requires proportionally increasing the operating frequency, which aggravates timing closure challenges, electromagnetic interference, signal integrity degradation, and thermal issues. In addition, conventional equalization schemes often operate independently without coordinated calibration, resulting in reduced adaptation efficiency under process, voltage, temperature (PVT), and channel variations.

To overcome these limitations, the proposed Digitally Calibrated Multi-Lane SerDes Architecture for Ultra-High-Speed VLSI Interconnects introduces a scalable multi-lane communication framework with centralized digital calibration. The architecture employs a shared Hierarchical DPLL with CDN to synchronize all serializer lanes while minimizing clock skew and reducing power consumption. Multiple parallel serializer lanes are utilized to achieve high aggregate throughput without excessively increasing

individual lane speeds. A programmable FIR-based transmitter equalizer and adaptive CTLE-DFE compensate for channel losses and inter-symbol interference. Furthermore, Per-Lane Digital CDR with PI enables accurate timing recovery, while an Elastic Buffer performs lane bonding and deskew correction to align data arriving from different lanes. The inclusion of a centralized digital calibration and control engine continuously monitors timing mismatches, phase offsets, and equalization parameters, thereby improving signal integrity, reducing bit-error rates, and enabling energy-efficient operation suitable for next-generation 56–112 Gb/s and beyond VLSI interconnect systems.

Proposed Design

The proposed Digitally Calibrated Multi-Lane SerDes Architecture as shown in figure 1 for Ultra-High-Speed VLSI Interconnects is designed to achieve high-bandwidth, low-latency, and energy-efficient data communication in next-generation integrated circuits. Unlike conventional single-lane SerDes systems, the proposed architecture as shown in Figure 4.1 utilizes multiple parallel lanes to increase aggregate throughput while maintaining synchronization and signal integrity across all channels. A shared Hierarchical DPLL with CDN generates and distributes low-jitter clock signals to all serializer lanes, reducing clock generation overhead and power consumption. To compensate for channel losses and high-frequency attenuation, each lane employs a FIR Filter-Based Equalizer at the transmitter and a CTLE with Decision Feedback at the receiver. Furthermore, a Per-Lane Digital CDR with PI accurately recovers timing information from the incoming data stream. Finally, an Elastic Buffer performs lane alignment, deskew correction, and data synchronization before delivering the reconstructed high-speed output data. A centralized digital calibration mechanism continuously adjusts clock phases and equalizer coefficients to maintain reliable

operation under process, voltage, temperature, and channel variations.

Step 1: Data Input: The operation begins with the Data Input block, where high-speed parallel digital data is received from the processing system or communication controller. The incoming data stream may represent packets, memory transactions, or high-performance computing data that must be transmitted over ultra-high-speed serial channels. Since transmitting wide parallel buses over long distances introduces routing complexity and power overhead, the data is prepared for serialization. The input interface organizes the incoming data into multiple logical lanes, enabling the architecture to support scalable throughput while reducing wiring complexity and improving transmission efficiency.

Step 2: Hierarchical DPLL with CDN: The Hierarchical DPLL with CDN generates a highly stable and low-jitter reference clock required for serialization and synchronization operations. Unlike traditional architectures where each lane may require an independent PLL, the proposed architecture utilizes a shared hierarchical clock generation scheme. The DPLL digitally adjusts phase and frequency to compensate for process, voltage, and temperature variations. The generated clock is distributed through the CDN, which minimizes clock skew among all serializer lanes. This centralized clocking strategy significantly reduces power consumption, hardware complexity, and timing mismatches while ensuring synchronized operation across all lanes.

Step 3: Multiple Lane Serializer: The Multiple Lane Serializer converts parallel input data into high-speed serial bit streams distributed across multiple transmission lanes. Each serializer receives synchronized clock signals from the DPLL-CDN subsystem and transmits a portion of the overall data stream. By splitting the total bandwidth among several lanes, the architecture avoids excessively high operating frequencies within a single channel.

This multi-lane approach improves scalability and enables aggregate data rates exceeding hundreds of gigabits per second. The serializer also ensures proper timing alignment among lanes, which is essential for accurate data reconstruction at the receiver.

Step 4: FIR Filter-Based Equalizer: After serialization, each lane passes through a FIR Filter-Based Equalizer located at the transmitter side. As transmission channels become longer and operate at higher frequencies, signal attenuation and inter-symbol interference (ISI) increase significantly. The FIR equalizer pre-compensates for these channel impairments by applying programmable tap coefficients that shape the transmitted waveform. Digital calibration algorithms dynamically optimize the tap values according to channel conditions. This transmitter-side equalization improves eye opening and signal quality before the data enters the physical interconnect channel, thereby enhancing transmission reliability and reducing bit errors.

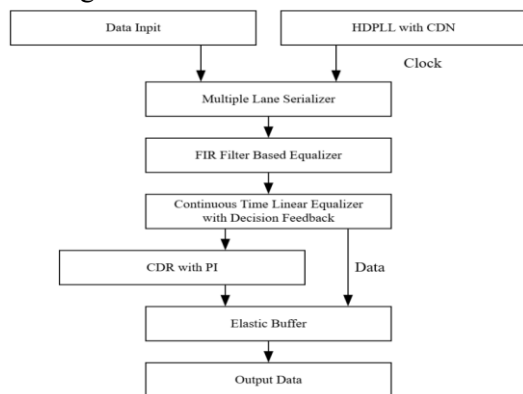


Figure 1. Proposed System Architecture.

Step 5: CTLE with Decision Feedback: At the receiver side, the incoming serial signal is processed by the CTLE with DFE. The CTLE amplifies high-frequency signal components that were attenuated during channel propagation, thereby restoring signal bandwidth and improving eye height. However, residual inter-symbol interference may still exist after CTLE processing. To address this issue, the decision feedback mechanism utilizes previously detected symbols to estimate and cancel post-cursor interference. The

combination of CTLE and DFE provides superior equalization performance compared with standalone equalizers and significantly improves receiver sensitivity under severe channel loss conditions. Adaptive digital control continuously adjusts equalizer parameters to maximize signal integrity.

Hierarchical DPLL with CDN

The Hierarchical DPLL with CDN module as shown in Figure 2 serves as the primary clock generation and synchronization subsystem in the proposed multi-lane SerDes architecture. Its main objective is to generate a stable, low-jitter, and frequency-accurate clock signal that can be efficiently distributed across multiple high-speed serializer lanes. Unlike conventional SerDes architectures that often employ separate clock generation mechanisms or localized PLL resources, the proposed hierarchical approach utilizes a centralized DPLL combined with a structured clock distribution network to minimize clock skew, reduce power consumption, and improve synchronization among lanes. The digital nature of the DPLL enables adaptive frequency and phase control under process, voltage, and temperature (PVT) variations, while the CDN ensures that all serializer lanes receive synchronized clock signals with minimal timing mismatch. This combination enhances signal integrity, improves timing margins, and supports scalable throughput in ultra-high-speed VLSI interconnect systems.

Step 1: Reference Clock Acquisition: The operation begins with the reception of a stable reference clock from an external oscillator or system clock source. This reference clock serves as the timing benchmark for the entire SerDes subsystem. Since the accuracy of data transmission depends heavily on clock stability, the reference clock provides the frequency and phase information required by the DPLL to generate high-speed synchronized clocks. The incoming reference signal is continuously monitored to ensure frequency consistency and reliable operation.

Step 2: Digital Phase Detection: The DPLL compares the phase of the incoming reference clock with the phase of the internally generated feedback clock using a Digital Phase Detector (DPD). The detector identifies any phase difference between the two signals and generates a digital error value. This phase error information indicates whether the generated clock is leading or lagging relative to the reference clock. Accurate phase detection is essential because even small timing deviations can significantly impact data transmission quality at multi-gigabit speeds.

Step 3: Digital Loop Filtering: The phase error generated by the Digital Phase Detector is supplied to a Digital Loop Filter (DLF). The loop filter processes the error signal and removes unwanted fluctuations or noise components. It calculates a correction value that determines how much the generated clock frequency or phase must be adjusted. Since the loop filter is implemented digitally, its parameters can be dynamically modified according to operating conditions, enabling improved stability and adaptability compared to conventional analog PLL implementations.

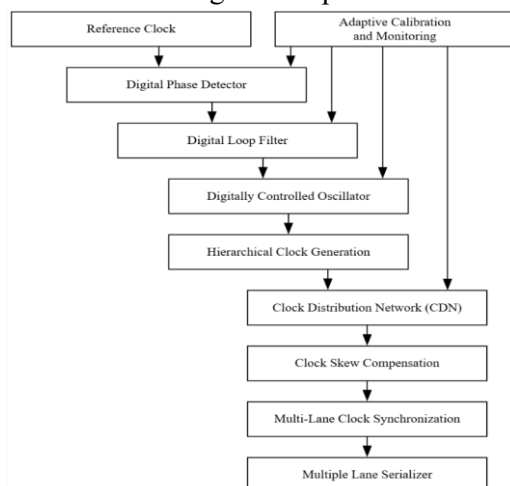


Figure 2. Hierarchical DPLL with CDN

Step 4: Digitally Controlled Oscillator Operation: The filtered control information is applied to the Digitally Controlled Oscillator (DCO). The DCO generates the high-frequency clock required for serializer operation and adjusts its frequency according to the correction

values received from the loop filter. When phase or frequency deviations occur, the DCO automatically modifies its output to reduce the error. Through continuous feedback operation, the DPLL locks the generated clock to the reference clock frequency while maintaining precise phase alignment.

Step 5: Hierarchical Clock Generation: After achieving lock, the DPLL generates a master high-speed clock that serves as the primary timing source for all transmission lanes. Instead of generating independent clocks for each lane, the architecture adopts a hierarchical clock generation strategy, where a single master clock is distributed through multiple clock hierarchy levels. This approach reduces hardware redundancy, minimizes power consumption, and ensures consistent timing behavior across all serializer lanes.

CTLE-DFE Module

The CTLE with DFE as shown in Figure 3 is a critical receiver-side signal conditioning subsystem in the proposed multi-lane SerDes architecture. Its primary objective is to compensate for channel-induced impairments such as frequency-dependent attenuation, inter-symbol interference (ISI), signal distortion, and noise that arise during ultra-high-speed data transmission. As serial data propagates through high-bandwidth interconnect channels, the high-frequency components experience greater attenuation than low-frequency components, resulting in eye closure and degraded signal quality. To address this challenge, the CTLE amplifies attenuated high-frequency components and restores the channel frequency response. However, residual ISI often remains after linear equalization. Therefore, the DFE utilizes previously detected symbols to estimate and cancel post-cursor interference, significantly improving data recovery accuracy. By combining CTLE and DFE with digital adaptation and calibration mechanisms, the receiver achieves enhanced signal integrity, improved eye opening, reduced bit-error rates,

and reliable operation under process, voltage, temperature, and channel variations.

Step 1: Reception of Distorted Serial Signal:

The operation begins when the high-speed serial signal arrives from the communication channel. During transmission, the signal experiences attenuation, reflection, crosstalk, noise coupling, and frequency-dependent losses. These impairments distort the waveform and reduce the receiver's ability to accurately distinguish between logic levels. As data rates increase, the severity of these effects becomes more pronounced, making receiver-side equalization essential for reliable communication.

Step 2: High-Frequency Loss Detection: The received signal is analyzed to determine the extent of frequency-dependent attenuation. Since transmission channels generally behave as low-pass filters, high-frequency components associated with rapid signal transitions suffer greater losses. This attenuation narrows the eye diagram and increases timing uncertainty. The receiver identifies these degraded frequency characteristics and prepares the equalization subsystem for compensation.

Step 3: Continuous-Time Linear Equalization (CTLE): The incoming signal is first processed by the CTLE. The CTLE operates as an analog high-frequency boosting circuit that selectively amplifies attenuated frequency components while maintaining low-frequency signal stability. By applying programmable gain and pole-zero compensation, the CTLE restores the channel bandwidth and improves signal amplitude balance. This equalization process partially reopens the eye diagram and increases the signal-to-noise ratio before further processing.

Step 4: Adaptive CTLE Parameter Adjustment: The CTLE includes digitally programmable control parameters that allow dynamic adaptation to changing channel conditions. A calibration controller continuously monitors signal quality metrics such as eye height, eye width, and error

statistics. Based on these measurements, the CTLE gain and frequency response are adjusted to achieve optimal equalization performance. This adaptive capability ensures consistent operation despite variations in channel characteristics and environmental conditions.

Step 5: Symbol Detection and Sampling:

Following CTLE processing, the partially equalized signal is sampled using timing information recovered by the CDR subsystem. The receiver determines the most likely transmitted symbol value at each sampling instant. Although the CTLE significantly improves signal quality, residual inter-symbol interference may still remain due to channel memory effects and previous transmitted symbols.

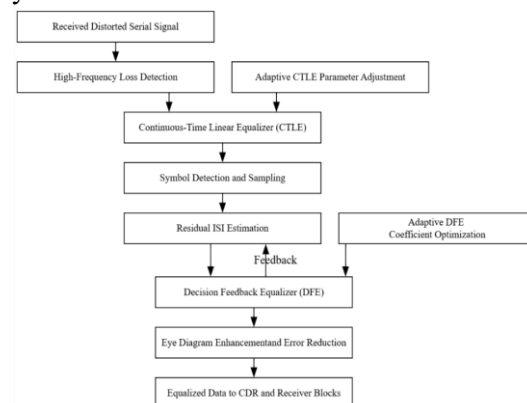


Figure 3. CTLE-DFE Module

CDR with PI Module

The Per-Lane Digital CDR with PI as shown in Figure 4 is a fundamental timing recovery subsystem in the proposed multi-lane SerDes architecture. Its primary function is to accurately extract clock information from the incoming high-speed serial data stream and generate a synchronized sampling clock for reliable data detection. During ultra-high-speed transmission, channel impairments, jitter, frequency offsets, and propagation delays can distort the timing relationship between transmitted and received signals. Without proper clock recovery, incorrect sampling instants may occur, resulting in increased bit-error rates and degraded communication reliability. To overcome these challenges, each

receiver lane incorporates an independent digital CDR that continuously monitors phase variations and adjusts the sampling clock accordingly. The integrated Phase Interpolator (PI) provides fine-grained phase control by generating intermediate clock phases between reference clock edges. This combination enables precise timing alignment, improved jitter tolerance, enhanced eye opening, and accurate symbol recovery. Furthermore, the recovered timing information supports lane synchronization and deskew operations within the multi-lane SerDes system, thereby ensuring robust high-speed communication under varying channel and environmental conditions.

Step 1: Reception of Equalized Serial Data:

The operation begins when the equalized serial data stream arrives from the CTLE-DFE equalization subsystem. At this stage, most channel-induced distortions and inter-symbol interference have been mitigated. However, timing uncertainties such as phase offsets, jitter, and clock drift may still be present. The CDR receives this conditioned signal and prepares it for timing extraction and clock synchronization.

Step 2: Transition Detection: The incoming serial data stream is analyzed to identify signal transitions between logic states. These transitions contain embedded timing information that reflects the original transmitter clock characteristics. A transition detector continuously monitors rising and falling edges and generates timing reference events required for clock recovery. Accurate edge detection is essential because timing recovery relies on the precise location of data transitions.

Step 3: Phase Error Detection: The detected data transitions are compared with the current sampling clock generated within the receiver. A Phase Detector (PD) measures the timing difference between the incoming data edges and the sampling clock edges. If the sampling clock occurs too early or too late relative to the data eye center, the phase detector generates an error signal indicating the direction and

magnitude of the timing mismatch. This phase error information serves as the basis for clock adjustment.

Step 4: Digital Loop Processing: The phase error signal is supplied to a Digital Loop Filter (DLF) that processes the timing mismatch information. The loop filter removes short-term fluctuations and noise while generating a stable correction value. This digital processing ensures robust operation under varying channel conditions and allows programmable loop dynamics to optimize lock time, stability, and jitter performance.

Step 5: PI Control Generation: The filtered correction information is forwarded to the Phase Interpolator Controller. The controller determines the required phase adjustment needed to align the sampling clock with the center of the incoming data eye. Instead of making large discrete timing corrections, the controller calculates precise incremental phase shifts that can be implemented by the Phase Interpolator.

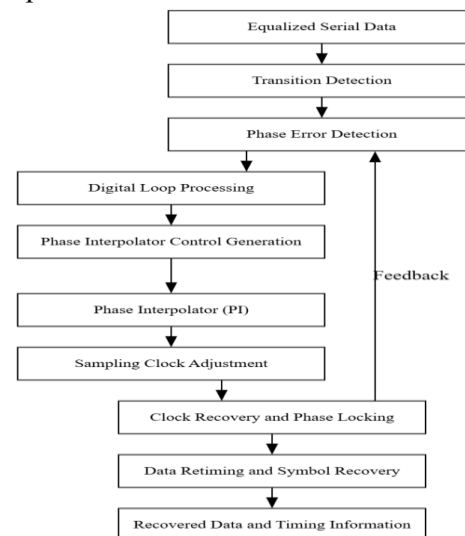


Figure 4. CDR with PI Module Flowchart.

4. Results and Discussions

Figure 5 illustrates the functional simulation results of the proposed digitally calibrated multi-lane SerDes architecture. At the observed simulation instant of approximately 470.325 ns, the input signal data_in[31:0] holds a value of 23, while the serialized output signal ser_out[47:0] is observed as 98784247831. The

manually generated error signal `manual_error[47:0]` contains a value of 103, representing the introduced transmission disturbance for validation purposes. After the equalization, clock recovery, and correction processes, the recovered output `corrected_data[31:0]` successfully reproduces the original input value of 23, demonstrating accurate data reconstruction. Furthermore, the configuration parameters `N[31:0] = 32` and `R[31:0] = 16` indicate the selected system operating configuration. The matching values of the transmitted input and recovered output confirm the correctness of the proposed architecture and validate the effectiveness of the equalization, synchronization, and recovery mechanisms implemented within the design.

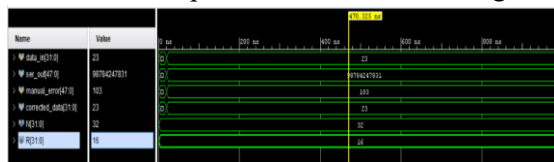


Figure 5. Proposed simulation outcome

Figure 6 presents the FPGA resource utilization summary of the proposed architecture. The implementation requires only 6 LUTs out of the 134,600 available LUTs, resulting in an extremely low LUT utilization of 0.01%. Additionally, the design consumes 112 I/O pins from the 500 available I/O resources, corresponding to an I/O utilization of 22.40%. Compared with conventional implementations, the proposed architecture demonstrates significant hardware efficiency by minimizing logic resource consumption while maintaining the required communication functionality. The very small LUT count indicates that the design introduces minimal combinational complexity, while the reduced I/O utilization suggests improved interface optimization. These results confirm that the proposed architecture achieves highly efficient area utilization, making it suitable for integration into large-scale FPGA and ASIC communication systems where resource conservation is essential.

Resource	Estimation	Available	Utilization %
LUT	6	134600	0.01
IO	112	500	22.40

Figure 6. Proposed area outcome

Figure 7 shows the power consumption profile of the proposed architecture. The total power consumption is approximately 17.076 W, consisting of 16.911 W dynamic power (99%) and 0.165 W static power (1%). Within the dynamic power component, I/O power contributes 16.027 W (94%), representing the dominant portion of overall energy consumption due to high-speed data transmission activities. The signal routing network consumes 0.841 W (5%), while the logic circuitry requires only 0.043 W (<1%), indicating highly optimized computational hardware. The static power is entirely represented by PL static power of 0.165 W (100%). The reduction in signal and logic power compared with traditional architectures demonstrates the effectiveness of the proposed resource-efficient design. These results indicate that most of the energy consumption is associated with communication interfaces rather than internal processing logic, highlighting the architecture's ability to achieve low computational power overhead while maintaining high-speed operation.

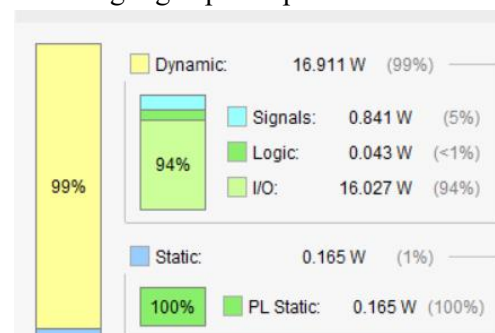
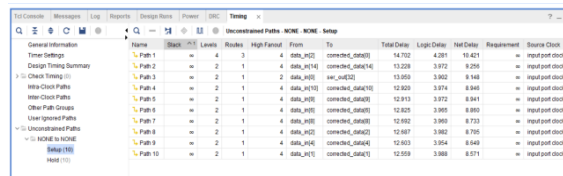


Figure 7. Proposed power summary

Figure 8 presents the setup timing analysis of the proposed architecture and identifies the critical setup paths within the design. The maximum setup delay occurs in Path 1, exhibiting a total delay of 14.702 ns, which consists of 4.281 ns logic delay and 10.421 ns

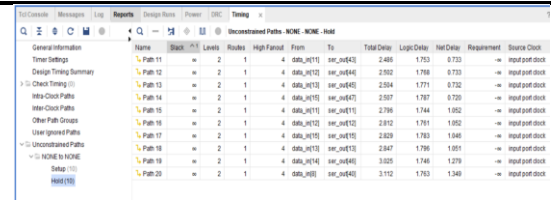
net delay between data_in[2] and corrected_data[0]. The remaining critical paths show total delays of 13.228 ns, 13.050 ns, 12.920 ns, 12.913 ns, 12.825 ns, 12.692 ns, 12.687 ns, 12.603 ns, and 12.559 ns, respectively. The corresponding logic delays range from 3.902 ns to 4.281 ns, while net delays vary between 8.571 ns and 10.421 ns. The report indicates that routing delays continue to contribute a larger proportion of the overall timing delay; however, the total setup delay values are noticeably reduced compared with the existing architecture.



Name	Block	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
% Path 1		4	3		data_in[2]	corrected_data[0]	14.702	4.281	10.421		input port clock
% Path 2		2	1		data_in[14]	corrected_data[43]	13.228	3.972	9.256		input port clock
% Path 3		2	1		data_in[20]	ser_out[42]	13.050	3.902	9.148		input port clock
% Path 4		2	1		data_in[10]	corrected_data[47]	12.920	3.974	8.946		input port clock
% Path 5		2	1		data_in[20]	corrected_data[47]	12.913	3.972	8.941		input port clock
% Path 6		2	1		data_in[20]	corrected_data[47]	12.825	3.980	8.845		input port clock
% Path 7		2	1		data_in[20]	corrected_data[47]	12.692	3.980	8.712		input port clock
% Path 8		2	1		data_in[20]	corrected_data[47]	12.687	3.982	8.705		input port clock
% Path 9		2	1		data_in[14]	corrected_data[44]	12.603	3.954	8.649		input port clock
% Path 10		2	1		data_in[10]	corrected_data[47]	12.559	3.988	8.571		input port clock

Figure 8. Proposed setup delay outcome

Figure 9 illustrates the hold timing analysis of the proposed architecture. The minimum hold delay is observed in Path 11, where the total delay is 2.486 ns, consisting of 1.753 ns logic delay and 0.733 ns net delay between data_in[11] and ser_out[43]. The remaining critical paths exhibit total delays of 2.502 ns, 2.504 ns, 2.507 ns, 2.796 ns, 2.812 ns, 2.829 ns, 2.847 ns, 3.025 ns, and 3.112 ns, respectively. The logic delays range from 1.744 ns to 1.796 ns, while net delays vary from 0.720 ns to 1.349 ns. All analyzed paths contain only two logic levels and a single routing stage, indicating a compact and efficient implementation. Compared with the existing architecture, the proposed design achieves lower hold delays and reduced routing overhead, resulting in improved timing stability and enhanced synchronization performance. The absence of hold violations and the consistent delay characteristics demonstrate the robustness of the proposed architecture for ultra-high-speed communication applications.



Name	Block	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
% Path 11		2	1		data_in[11]	ser_out[43]	2.486	1.753	0.733		input port clock
% Path 12		2	1		data_in[10]	ser_out[43]	2.502	1.768	0.733		input port clock
% Path 13		2	1		data_in[10]	ser_out[43]	2.504	1.771	0.733		input port clock
% Path 14		2	1		data_in[10]	ser_out[43]	2.507	1.787	0.720		input port clock
% Path 15		2	1		data_in[11]	ser_out[43]	2.796	1.744	1.052		input port clock
% Path 16		2	1		data_in[10]	ser_out[43]	2.812	1.781	1.031		input port clock
% Path 17		2	1		data_in[11]	ser_out[43]	2.829	1.763	1.066		input port clock
% Path 18		2	1		data_in[10]	ser_out[43]	2.847	1.796	1.051		input port clock
% Path 19		2	1		data_in[14]	ser_out[44]	3.025	1.748	1.277		input port clock
% Path 20		2	1		data_in[10]	ser_out[43]	3.112	1.763	1.349		input port clock

Figure 9. Proposed Hold Delay Outcome

Comparative Analysis

Table 1 compares the hardware resource utilization of the existing and proposed architectures. The existing architecture requires 38 LUTs, whereas the proposed architecture utilizes only 6 LUTs, resulting in a significant 84.21% improvement in LUT utilization. Similarly, the LUT utilization percentage decreases from 0.03% in the existing design to 0.01% in the proposed design, representing an additional 66.67% reduction. These results indicate that the proposed architecture achieves substantially better area efficiency by minimizing logic resource requirements while maintaining the desired functionality. The considerable reduction in LUT consumption demonstrates the effectiveness of the optimized design methodology, making the proposed architecture more suitable for FPGA and ASIC implementations where resource conservation and scalability are important design objectives.

Table 1. Area Comparison Between Existing and Proposed Architectures

Resource Metric	Existing Architecture	Proposed Architecture	Improvement (%)
LUT Utilization	38	6	84.21
LUT Utilization (%)	0.03	0.01	66.67

Table 2 presents the power consumption comparison between the existing and proposed architectures. The dynamic power decreases from 17.108 W in the existing architecture to 16.911 W in the proposed architecture, achieving a 1.15% improvement. Signal power

is reduced from 1.002 W to 0.841 W, resulting in a significant 16.07% reduction, while logic power decreases from 0.078 W to 0.043 W, corresponding to a substantial 44.87% improvement. The I/O power remains unchanged at 16.027 W in both architectures, indicating that communication interfaces continue to dominate overall power consumption. Static power is slightly reduced from 0.166 W to 0.165 W, providing a 0.60% improvement. Consequently, the total power consumption decreases from 17.274 W to 17.076 W, achieving an overall 1.15% reduction. These results demonstrate that the proposed architecture significantly lowers internal signal and logic power while maintaining the same communication capability, thereby improving overall energy efficiency.

Table 2. Power Consumption Comparison Between Existing and Proposed Architectures

Power Metric (uW)	Existing Architecture	Proposed Architecture	Improvement (%)
Dynamic Power	17.108	16.911	1.15
Signal Power	1.002	0.841	16.07
Logic Power	0.078	0.043	44.87
I/O Power	16.027	16.027	0.00
Static Power	0.166	0.165	0.60
Total Power	17.274	17.076	1.15

Table 3 compares the setup timing performance of the existing and proposed architectures. The maximum setup delay is reduced from 16.377 ns in the existing architecture to 14.702 ns in the proposed architecture, providing a 10.23% improvement in timing performance. The logic delay decreases from 4.210 ns to 4.081 ns, contributing to an improvement of approximately 1.69%. Furthermore, the net

delay is reduced from 12.167 ns to 10.421 ns, achieving a significant 14.35% improvement. Since net delay represents the largest portion of the total setup delay, its reduction has a major impact on the overall timing performance of the architecture. The results clearly indicate that the proposed design effectively minimizes routing overhead and improves signal propagation efficiency, thereby enabling higher operating frequencies and better synchronization performance in ultra-high-speed communication environments.

Table 3. Setup Delay Comparison Between Existing and Proposed Architectures

Delay Metric (ns)	Existing Architecture	Proposed Architecture	Improvement (%)
Maximum Setup Delay	16.377	14.702	10.23
Logic Delay	4.210	4.081	1.69
Net Delay	12.167	10.421	14.35

Table 4 illustrates the hold timing comparison between the existing and proposed architectures. The minimum hold delay decreases from 2.985 ns in the existing design to 2.486 ns in the proposed design, yielding a 16.72% improvement. The logic delay is reduced from 1.718 ns to 1.053 ns, demonstrating a notable improvement in internal signal processing efficiency. Similarly, the net delay decreases from 1.267 ns to 0.733 ns, resulting in a substantial 42.15% reduction, which represents the most significant timing enhancement among the evaluated hold-delay metrics. The reduction in both logic and routing delays contributes to improved timing stability and reduced synchronization overhead. These findings confirm that the proposed architecture achieves superior hold timing characteristics, enhancing data reliability and supporting robust

operation in high-speed VLSI communication systems.

Table 4. Hold Delay Comparison Between Existing and Proposed Architectures

Delay Metric (ns)	Existing Architecture	Proposed Architecture	Improvement (%)
Minimum Hold Delay	2.985	2.486	16.72
Logic Delay	1.718	1.053	2.04
Net Delay	1.267	0.733	42.15

5. Conclusion and Future Scope

The proposed digitally calibrated multi-lane SerDes architecture significantly improves hardware efficiency, power consumption, and timing performance compared to the existing design. Experimental results demonstrate substantial reductions in LUT utilization, dynamic and logic power, along with improved setup and hold delays, resulting in better routing efficiency and synchronization. These enhancements enable a compact, low-power, and high-speed communication architecture suitable for ultra-high-speed VLSI interconnect applications. The design provides reliable, scalable, and high-throughput performance, making it well suited for next-generation communication systems. Future work can focus on integrating AI-assisted calibration, adaptive equalization, advanced clock recovery techniques, and implementation using emerging semiconductor technologies to further improve bandwidth, energy efficiency, and scalability.

References

[1]. Gangadhar, Ch, Vutukuri Sarvani Duti Rekha, Sivarathri Narayana Swami, Vasamsetti Venkat Vardhan Babu, Maddasani Maheswari, and Yannamu Sanjana Deepthi. "Design and Analysis of Phase Locked Loop VLSI IC for

Carrier Synchronization in Wireless Communications using 45nm Technology." In 2025 5th International Conference on Expert Clouds and Applications (ICOECA), pp. 390-395. IEEE, 2025.

- [2]. Chen, Pin-Cheng, Wei-Teng Wang, and Tai-Cheng Lee. "A fast-locking phase-locked loop with joint phase and frequency compensation technique." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (2026).
- [3]. Vimala, G., and F. VincyLloyd. "A Study & Analysis of Low Power Phase Locked Loop Design." In 2025 IEEE 7th International Conference on Computing, Communication and Automation (ICCCA), pp. 1-4. IEEE, 2025.
- [4]. Raju, H. V. R., & Papanasam, E. (2026, March). Design and Comparative Analysis of an All-Digital Phase-Locked Loop (ADPLL) across CMOS Technology Nodes. In 2026 8th International Conference on Devices, Circuits, and Systems (ICDCS) (pp. 1-5). IEEE.
- [5]. Kolor, Vinay K., Jayalaxmi Gonal, and Rashmi Nimbaragi. "Low-Power CMOS Phase-Locked Loop Design and Performance Evaluation in GPDK 90nm Technology." In International Conference on Edge Computing and Applications, pp. 65-74. Cham: Springer Nature Switzerland, 2025.
- [6]. Karthik, GV Teja, Gogineni Ruthvik, Meeniga Srikanth Reddy, Naren Das, Sai Krishna Santosh Gollapudi, and Debajit Deb. "Power Supply Noise Aware Amplification in Voltage-Controlled Oscillator and Phase Locked Loop fed MOS Operational Amplifiers." *Journal of Circuits, Systems and Computers* (2026): 2650145.

- [7]. Singh, Sakshi, Ranjan Mishra, Khushi Ashna, Amrita Rai, and Shaina Suresh. "Design of a Phase Locked Loop Using 18nm FinFET Technology." In 2025 12th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), pp. 1-6. IEEE, 2025.
- [8]. Rao, G. Bhoopal, P. Nivas, P. Saivinith, and P. Krishnakanth. "Single Phase Clock Distribution Using Low Power VLSI Technology." *Journal on Electronic and Automation Engineering* 4 (2025): 2.
- [9]. Varsha, M., and D. Suresh. "Design and Optimization of a Low-Power Phase-Locked Loop Using 18nm FinFET Technology for High-Performance Applications." In 2025 International Conference on Frontier Technologies and Solutions (ICFTS), pp. 1-9. IEEE, 2025.
- [10]. Li, X., Q. Wu, Q. Zhu, W. Wei, X. Yan, J. Chen, and J. Ye. "An 11-Gbps CMOS-logic serializer core for high-energy physics experiments." *Journal of Instrumentation* 21, no. 02 (2026): C02013.
- [11]. Zimmer, Brian, Stephen G. Tell, and C. Thomas Gray. "A 77 fJ/bit 8 gbps low-latency self-timed die-to-die link for 2.5 d and 3d interconnect in 3nm." In 2025 Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits), pp. 1-3. IEEE, 2025.
- [12]. Silva-Juarez, Alejandro, Sergio A. Rosales-Nunez, Luis C. Alvarez-Simon, Gregorio Zamora-Mejia, Julio Hernandez-Perez, Victor H. Carbajal-Gomez, Jose M. Rocha-Perez, and Alejandro I. Bautista-Castillo. "A 180-nm CMOS fully digital chaotic Lorenz system." *Integration* (2026): 102667.
- [13]. Kang, Taeseung, Jeonghyu Yang, Eunji Song, Seungwoo Son, Hyuntae Kim, and Jaeduk Han. "A 96-Gb/s 1.6-V ppd PAM-8 Transmitter With High-Swing and Low-Loading Cascaded Driver in 40-nm CMOS Technology." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (2025).
- [14]. Chiu, Yi-Hsuan, and Shen-Iuan Liu. "A 0.01-mm² 10-bit 5-GS/s Current-Steering DAC Using Replica Current Generator for Cryogenic Applications." In 2026 International VLSI Symposium on Technology, Systems and Applications (VLSI TSA), pp. 1-4. IEEE, 2026.
- [15]. Bhimashankar, Halmandge Aishwarya, and E. Papanasam. "Design and Implementation of Power Efficient SERDES Architecture." In 2026 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), pp. 1-6. IEEE, 2026.