DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Visvesvaraya Technological University (VTU), Belagavi,
Approved by AICTE and UGC, Accredited by NAAC with 'A' grade & ISO 9001 – 2015 Certified Institution)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560 078, India

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

(Accredited by NBA Tier 1: 2022-2025)

Project Report on

DESIGN AND SYNTHESIS OF PHYSICAL LAYER MODULE OF USB 3.0

Submitted in partial fulfillment for the award of the degree of

Bachelor of Engineering
in
Electronics & Communication Engineering

Submitted by

ABHINAV SUNDRIYAL	**1DS21EC006**
CHIRAG VIJAPUR	**1DS21EC057**
MANOJ SS	**1DS21EC112**
SURUCHI SINGH	**1DS22EC216**

Under the Guidance of

Dr. Sapna P.J
Associate Professor
Department of Electronics and Communication Engineering
DSCE, Bengaluru

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI-590018, KARNATAKA, INDIA
2024-25

DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Visvesvaraya Technological University (VTU), Belagavi,
Approved by AICTE and UGC, Accredited by NAAC with 'A' grade & ISO 9001 – 2015 Certified Institution)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560 078, India

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

(Accredited by NBA Tier 1: 2022-2025)

CERTIFICATE

Certified that the project report entitled **"Design and Synthesis of Physical Layer Module of USB 3.0**" carried out by **ABHINAV SUNDRIYAL** bearing **USN: 1DS21EC006** is a bonafide student of **DAYANANDA SAGAR COLLEGE OF ENGINEERING**, an autonomous institution affiliated to VTU, Belagavi in partial fulfillment for the award of Degree of **Bachelor of Electronics & Communication Engineering** during the year **2024-2025**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements with respect to the work prescribed for the said Degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
Dr. Sapna P.J	**Dr. Shobha K.R.**	**Dr. B G Prasad**
Associate Professor, Dept. of ECE, DSCE, Bengaluru	**Professor & Head Dept. of ECE, DSCE, Bengaluru**	**Principal DSCE, Bengaluru**

Name of the Examiners	Signature with date
1.	**.....**
2.	**.....**

DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Visvesvaraya Technological University (VTU), Belagavi,
Approved by AICTE and UGC, Accredited by NAAC with 'A' grade & ISO 9001 – 2015 Certified Institution)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560 078, India

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

(Accredited by NBA Tier 1: 2022-2025)

DECLARATION

We, **Abhinav Sundriyal (1DS21EC006), Chirag Vijapur (1DS21EC057), Manoj SS (1DS21EC112)** and **Suruchi Singh (1DS21EC216),** respectively, hereby declare that the project work entitled " **Design and Synthesis of Physical Layer Module of USB 3.0** " has been independently done by us under the guidance of **'Dr. Sapna P.J', Associate Professor,** ECE department and submitted in partial fulfillment of the requirement for the award of the degree of **Bachelor of Electronics & Communication Engineering** at **Dayananda Sagar College of Engineering**, an autonomous institution affiliated to VTU, Belagavi during the academic year 2024-2025.

We further declare that we have not submitted this report either in part or in full to any other university for the award of any degree.

ABHINAV SUNDRIYAL	**1DS21EC006**
CHIRAG VIJAPUR	**1DS21EC057**
MANOJ SS	**1DS21EC112**
SURUCHI SINGH	**1DS22EC216**

PLACE: BENGALURU

DATE:

ACKNOWLEDGEMENT

The satisfaction and euphoria accompanying the successful completion of any task would be incomplete without the mention of people who made it possible and under constant guidance and encouragement the task was completed. We sincerely thank the **Management of Dayananda Sagar College of Engineering, Bengaluru.**

We express our sincere regards and thanks to **Dr. B G Prasad, Principal, Dayananda Sagar College of Engineering, Bengaluru.** His constant encouragement guidance and valuable support have been an immense help in realizing this project.

We express our sincere regards and thanks to **Dr. Shobha K.R, Professor & Head, Department of Electronics & Communication Engineering, Dayananda Sagar College of Engineering, Bengaluru.** Her incessant encouragement guidance and valuable technical support have been an immense help in realizing this project. Her guidance gave us the environment to enhance our knowledge, and skills and to reach the pinnacle with sheer determination, dedication, and hard work.

We would like to express profound gratitude to our Project guide **Dr. Sapna P.J, Associate Professor, Department of Electronics & Communication Engineering, Dayananda Sagar College of Engineering, Bengaluru** who has encouraged us throughout the project. Her moral support enabled us to complete our project work successfully.

We express our sincere thanks to Project Coordinators **Dr.S.Thenmozhi**, Bindu H.M. of the **Department of Electronics and Communication Engineering** for their continues support and guidance. We thank all teaching and non-teaching staff of the Department of Electronics and Communication Engineering for their kind and constant support throughout the academic Journey.

ABHINAV SUNDRIYAL	**1DS21EC006**
CHIRAG VIJAPUR	**1DS21EC057**
MANOJ SS	**1DS21EC112**
SURUCHI SINGH	**1DS22EC216**

ABSTRACT

The USB 3.0 Physical Layer serves as the backbone of the Superspeed architecture, facilitating robust and high-speed data transfer at a nominal rate of 5 Gbps. This project entails the design and simulation of the USB 3.0 Physical Layer using Verilog HDL, emphasizing critical modules such as 8b/10b encoding for DC-balanced data transmission, clock recovery for bit-level synchronization, and parallel-to-serial conversion for efficient data handling. The layer's architecture leverages unidirectional differential pairs for bidirectional communication and integrates power management features for low-power states. Simulations validate the design's capacity for asynchronous data transfer with high signal integrity, optimized latency, and adaptability across diverse operating conditions, aligning with the stringent requirements of USB 3.0 standards.

Keywords: *Physical Layer*, *Superspeed*,*8b/10b Encoding*, *Verilog*, *Asynchronous Data Transfer*

Table of Contents

PUBLICATION DETAILS

PLAGIARISM REPORT

APPENDIX (IF ANY)

LIST OF FIGURES

INTRODUCTION

Overview

USB 3.0, widely recognized as SuperSpeed USB, represents a significant advancement in Universal Serial Bus (USB) technology. It offers improved data transfer rates, enhanced power management, and increased efficiency compared to earlier USB versions. Introduced to address the growing demand for faster and more reliable data transmission, USB 3.0 supports speeds of up to 5 Gbps, making it a cornerstone for high-performance computing and data-driven applications. Moreover, it retains backward compatibility with USB 2.0, ensuring seamless integration with existing devices while delivering enhanced capabilities.

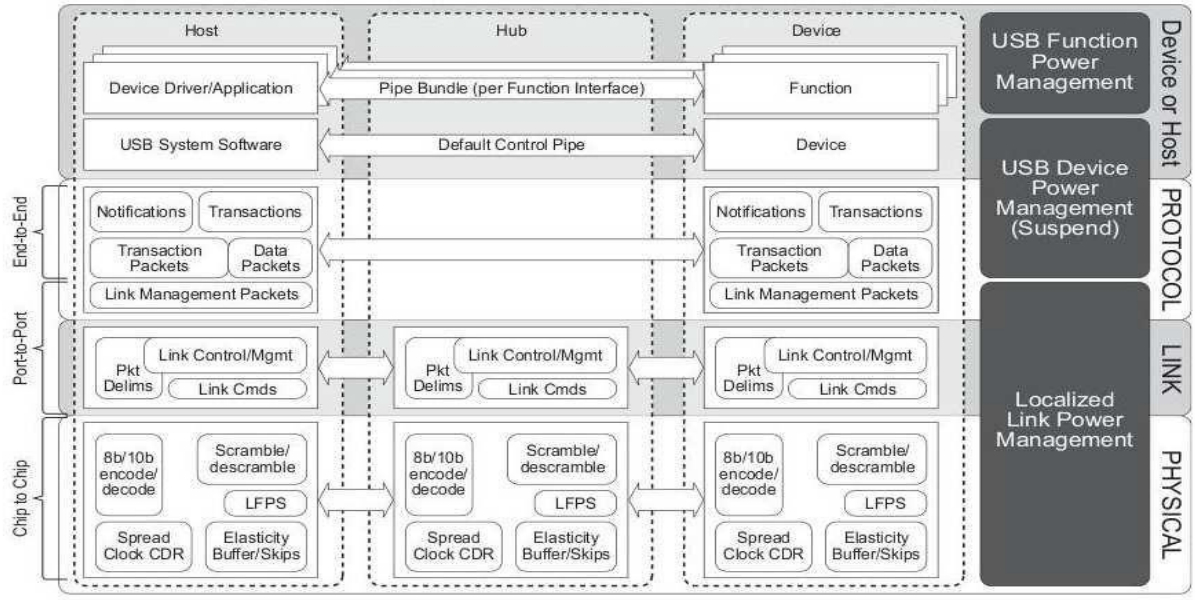


Fig. 1: USB 3.0 Superspeed and Power management

This project focuses on the design and simulation of a USB 3.0 interface using Verilog Hardware Description Language (HDL), leveraging multiple software tools to achieve key milestones. The simulation of the design was conducted in Xilinx ISE, followed by obtaining the RTL representation in Quartus Prime. Finally, the synthesized netlist was generated using Cadence

Genus. These steps ensured a comprehensive workflow, integrating simulation, synthesis, and optimization for efficient implementation of the USB 3.0 Physical Layer.

USB 3.0, known as SuperSpeed USB, is renowned for its ability to achieve high data transfer rates, enhanced power efficiency, and backward compatibility with earlier USB standards. This project emphasizes the Physical Layer, a critical component enabling data transmission speeds of up to 5 Gbps. It incorporates essential features such as Superspeed functionality, backward compatibility with USB 2.0, and low-power state support.

A notable challenge addressed during the project was the design and implementation of the 8b/10b encoder and decoder, a key element ensuring data integrity and maintaining DC balance during serial communication. This mechanism supports reliable high-speed data transmission by embedding clock signals within the data and balancing the bitstream.

The Physical Layer's design is central to USB 3.0's Superspeed architecture. This layer establishes the electrical and signalling framework required for communication using two unidirectional differential pairs for transmitting and receiving data. Its key features include 8b/10b encoding, clock recovery mechanisms for bit-level synchronization, and power management for operating in low-power states when required.

The project adopted a structured approach to designing the Physical Layer using Verilog HDL, covering critical modules such as data encoding, clock generation, and parallel-to-serial conversion. The design underwent simulation in Xilinx ISE to verify functionality, RTL representation was derived in Quartus Prime, and synthesis was completed using Cadence Genus to generate the final netlist. The results validate the layer's ability to facilitate asynchronous data transfer, maintain signal integrity, and optimize performance across varying operational conditions.

Following is the explanation of different layers of USB 3.0:

Protocol Layer

The Protocol Layer is responsible for managing the data flow at a high level between the host and the device. On the host side, this layer involves Device Drivers/Applications and USB System Software, which handle device-specific operations and interact through the default control pipe. The control pipe ensures proper communication setup and data exchange.

Key operations include handling different types of packets:

- Transaction Packets: Responsible for initiating communication and defining the type of operation (e.g., read, write).
- Data Packets: Carry actual user data or commands to the device.
- Link Management Packets: Coordinate link-level operations, such as establishing or closing a communication session.
- Notification Packets: Provide real-time updates or alerts, such as status changes in devices.

By handling these packets, the Protocol Layer plays a vital role in coordinating the overall communication framework.

Link Layer

The Link Layer bridges the high-level protocol and the physical transmission medium. It performs link control and management, focusing on ensuring that data flows seamlessly and efficiently between the host and device. Key tasks include managing packet delimiters, which define where packets start and end, and link commands, which control link setup, maintenance, and teardown.

This layer also integrates localized power management, allowing the link to dynamically enter low-power states during periods of inactivity. Such mechanisms are particularly important for conserving energy in USB-powered devices, ensuring the link remains responsive while consuming minimal power.

The Link Layer is also responsible for maintaining synchronization between the host and device to prevent data loss or corruption during high-speed transfers.

Physical Layer

The Physical Layer represents the hardware interface and manages all electrical and signalling aspects. It ensures robust data transmission and reception through several critical components:

- 8b/10b Encoding/Decoding: Converts 8-bit data into 10-bit encoded format for DC balance and reliable data transfer. This ensures that no long sequences of identical bits occur, aiding in clock recovery.
- Scrambling/Descrambling: Randomizes the data stream to reduce electromagnetic interference (EMI), improving signal quality.
- LFPS (Low-Frequency Periodic Signalling): Used to establish communication, signal resets, and perform other essential link initialization tasks.
- Spread Clock and Clock Data Recovery (CDR): Synchronizes timing between devices, ensuring that the transmitted data can be accurately interpreted without signal distortion.
- Elasticity Buffers and Skips: Address timing mismatches and ensure smooth handling of asynchronous data transfers, maintaining the integrity of high-speed transmissions.

These features are essential for maintaining USB 3.0's promised data transfer rate of 5 Gbps while preserving signal quality and reliability across varying environmental conditions.

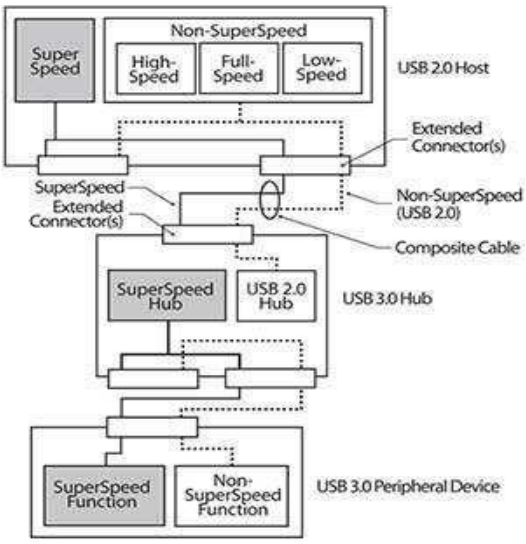


Fig. 2: USB 3.0 Architecture

Features of USB 3.0:

1. SuperSpeed PHY Layer

The SuperSpeed PHY (Physical) Layer is a critical component that handles low-level signalling and protocol operations specific to USB 3.0's SuperSpeed mode. It provides the foundational infrastructure required to achieve the high-speed data rates associated with USB 3.0. The PHY layer ensures smooth communication by managing both electrical signalling and the implementation of the Superspeed protocol.

2. Standard PHY Interface with Multi-IP Support

USB 3.0 supports a standard PHY interface that allows integration with multiple Intellectual Property (IP) sources. This flexibility simplifies system design and enables the seamless collaboration of components from various vendors. The standardization of the PHY interface facilitates the development of interoperable USB solutions and streamlines the integration process in complex systems.

3. High-Speed Serial Data Transmission

The USB 3.0 architecture supports a blazing-fast 5.0 Gbps serial data transmission rate, making it significantly faster than its predecessors. This high-speed capability enables rapid file transfers, supports demanding applications such as HD video streaming, and enhances overall system performance in data-intensive environments.

4. Parallel Interface Options

To accommodate different design requirements, USB 3.0 provides the option to use 8-bit, 16-bit, or 32-bit parallel interfaces for transmitting and receiving data. These parallel data paths offer flexibility for developers, allowing them to optimize system configurations based on speed, power efficiency, and hardware constraints.

5. Integration of High-Speed Components

USB 3.0 simplifies system design by allowing the integration of high-speed components into a single functional block, which is managed by the endpoint device controller. This feature enhances efficiency by reducing the complexity of individual components while maintaining optimal performance and functionality.

6. Data and Clock Recovery

One of USB 3.0's advanced features are the ability to perform data and clock recovery from the serial data stream. This ensures precise synchronization between the transmitter and receiver, which is essential for maintaining data integrity and reducing errors during high-speed data transmission.

7. Disparity Control for Compliance Patterns

The PHY layer provides support for direct disparity control, a key feature used during the transmission of compliance patterns. This capability ensures that data transmitted during compliance testing conforms to USB 3.0 standards and specifications, aiding in system verification and validation.

8. 8b/10b Encoding and Decoding

USB 3.0 employs the 8b/10b encoding scheme, which converts 8-bit data into a 10-bit format to maintain DC balance and ensure reliable high-speed data transmission. This encoding helps embed clock information within the data stream, simplifying the synchronization process. Additionally, USB 3.0 provides error indication mechanisms, alerting the system to transmission issues and allowing for corrective measures to maintain data integrity.

9. Error Detection and Indication

The USB 3.0 system is equipped with robust error detection and indication features, ensuring the identification of errors during data transmission. This contributes to the overall reliability and robustness of USB 3.0 by enabling the quick resolution of errors, thereby minimizing communication downtime.

10. Support for Low-Power States

Although not directly related to the features listed above, it is worth noting that USB 3.0 also incorporates low-power state support within its PHY and other layers, enabling efficient energy usage during idle periods or when handling fewer demanding workloads.

Problem Statement

- Designing a USB 3.0 interface that supports data transfer rates of 2.5 Gbps and 5 Gbps is challenging due to the need for precise timing, reliable signal integrity, and robust synchronization in high-speed communication.
- The 8b/10b encoding scheme is essential for maintaining DC balance and ensuring error-free data transmission. However, its design and integration into the system require careful handling of high-speed serial data without compromising performance.
- Synchronization between transmitting and receiving devices at high data rates is critical.
- USB 3.0 includes advanced power management features, such as low-power states, which require careful integration to maintain efficiency without compromising performance.
- High-speed data transmission increases the risk of errors, requiring the implementation of robust mechanisms for error detection, correction, and recovery to ensure data integrity and reliability.

Objectives

- Create a USB 3.0 interface that is synthesizable, scalable, and can be integrated into various digital systems. Implement core layers of USB 3.0, including the Physical Layer (PHY) and Link Layer, which are responsible for data transmission and link management.
- Superspeed functionality, which enables data transfer rates of up to 5 Gbps, leveraging dual-simplex data paths. Use dual-bus architecture to achieve backward compatibility with USB 2.0, allowing for simultaneous operation of both Superspeed and legacy USB modes.
- Design an 8b/10b encoding and decoding scheme to ensure DC balance and error detection capability. Optimize the encoder and decoder modules to achieve efficient high-speed serial data transmission, critical for minimizing data errors and ensuring signal integrity.
- Develop a robust clock and data recovery (CDR) system, crucial for synchronizing the receiver's local clock with incoming serial data. Design a Digital Phase-Locked Loop (DPLL) to manage high-speed synchronization, ensuring seamless data flow and avoiding clock skew issues.

Motivation

The semiconductor sector in India has grown phenomenally in the past few years due to a mix of national as well as international factors. With its expanding digital economy, thriving technology sector, and well-paced governmental initiative to aim for self-sufficiency, India is emerging as an important player in the global semiconductor value chain. Growth covers not just the manufacturing of chips but has extended all along the through semiconductor ecosystem from design toward testing, packaging and assembly.

The growing demand for high-speed data transmission in modern applications, ranging from high-definition media streaming to real-time communication and data-intensive computing, underscores the importance of efficient and robust interfaces like USB 3.0. Its Superspeed architecture, offering data rates up to 5 Gbps, represents a significant advancement over previous USB standards. However, achieving this performance requires a meticulously designed Physical Layer that ensures signal integrity, synchronization, and power efficiency.

This project was driven by the desire to delve into the complexities of designing high-speed communication systems and to gain hands-on experience with Verilog HDL for implementing industry-standard protocols. Exploring the intricate mechanisms like 8b/10b encoding, clock recovery, and low-power states not only enhances technical expertise but also provides a deeper understanding of real-world challenges in high-speed data interface design. Additionally, the project aligns with the industry's growing focus on designing power-efficient and high-performance systems, making it both technically and professionally rewarding.

LITERATURE SURVEY

Paper 1: Optimization of Physical Layer Modules of USB 3.0 Using FPGA

Abstract

The USB 3.0 standard, widely adopted for high-speed data transfer, builds on USB 2.0 by offering enhanced transfer rates and reduced latency, benefiting applications in storage devices, personal computing, and peripheral interconnections. With its layered structure, USB 3.0 includes critical components such as the Physical Layer, where modules like the scrambler and 8b/10b encoder play essential roles in data integrity and electromagnetic interference (EMI) reduction. The primary goal of these components is to maintain signal quality, manage power consumption, and optimize chip area when synthesized on FPGA platforms.

Ideas influenced by the paper: -

This paper highlights the optimization of USB 3.0 Physical Layer components for FPGA implementation, focusing on scrambler design, 8b/10b encoding, and power management. It emphasizes achieving efficient resource utilization and signal integrity while minimizing power consumption and chip area. The insights informed the focus on resource-efficient designs and power optimization techniques in our project.

Paper 2: USB 3.0 High-Transfer Rate Time-Tagging Module for High-Performance FPGA-based Time-to-Digital Converter

Abstract

The development and implementation of USB 3.0 protocols in high-performance data communication systems, particularly the Physical Layer modules, have attracted extensive research attention due to their significant impact on data transfer speed, power efficiency, and overall system performance. This review discusses the advantages and disadvantages of USB 3.0 Physical Layer design, drawing insights from studies focused on FPGA-based implementations and high-speed data transmission protocols.

Ideas influenced by the paper: -

This paper inspired the use of FPGA-based prototyping for USB 3.0 Physical Layer design, focusing on 8b/10b encoding for error reduction and modular FPGA implementations for adaptability. It reinforced the importance of leveraging USB 3.0's high-speed transfer rate and efficient encoding techniques to improve system performance while managing complexity and power usage.

Paper 3: Implementation of USB 3.0 SuperSpeed Physical Layer Using Verilog HDL

Abstract

The USB 3.0 architecture introduced several key features that differentiated it from previous versions. It operates on a dual-bus architecture, allowing backward compatibility with USB 2.0 and facilitating simultaneous SuperSpeed and lower-speed operations. One of the significant advancements in USB 3.0 is the Physical Layer (PHY), which includes a dual-simplex data path enabling higher data rates. This layer comprises essential components, such as 8b/10b encoding for DC balance and data integrity, and a clock and data recovery mechanism using phase-locked loops (PLL). Numerous studies have focused on implementing the USB 3.0 Physical Layer using Verilog HDL to create a synthesizable design for high-speed data transmission.

Ideas influenced by the paper: -

This paper emphasized the significance of the USB 3.0 Physical Layer's dual-simplex architecture, backward compatibility, and 8b/10b encoding for data integrity. It influenced the approach to designing a synthesizable Verilog HDL model, particularly focusing on clock and data recovery mechanisms to ensure reliable high-speed data transmission.

Paper 4: Design of High-Quality Image Acquisition System Based on FPGA and USB 3.0

Abstract

The design of high-quality image acquisition systems has become a significant research focus due to the growing demands in areas like intelligent manufacturing, medical imaging, and automated surveillance. H. Li and Y. Xiao (2023) address these challenges by proposing an FPGA-based system integrated with a USB 3.0 interface. Their work builds on existing advancements in image acquisition, leveraging USB 3.0 for its high-speed data transfer capabilities and FPGAs for their flexibility and real-time processing power.

Ideas influenced by the paper: -

This paper provided insights into leveraging USB 3.0's high-speed transfer capabilities in real-time applications. It also highlighted the role of FPGAs in handling parallel processing tasks efficiently. The concept of using USB 3.0 for reducing data flow bottlenecks influenced the project's focus on implementing high-speed, low-latency designs tailored for modern communication systems.

PROBLEM ANALYSIS & DESIGN

Block diagram and working principle

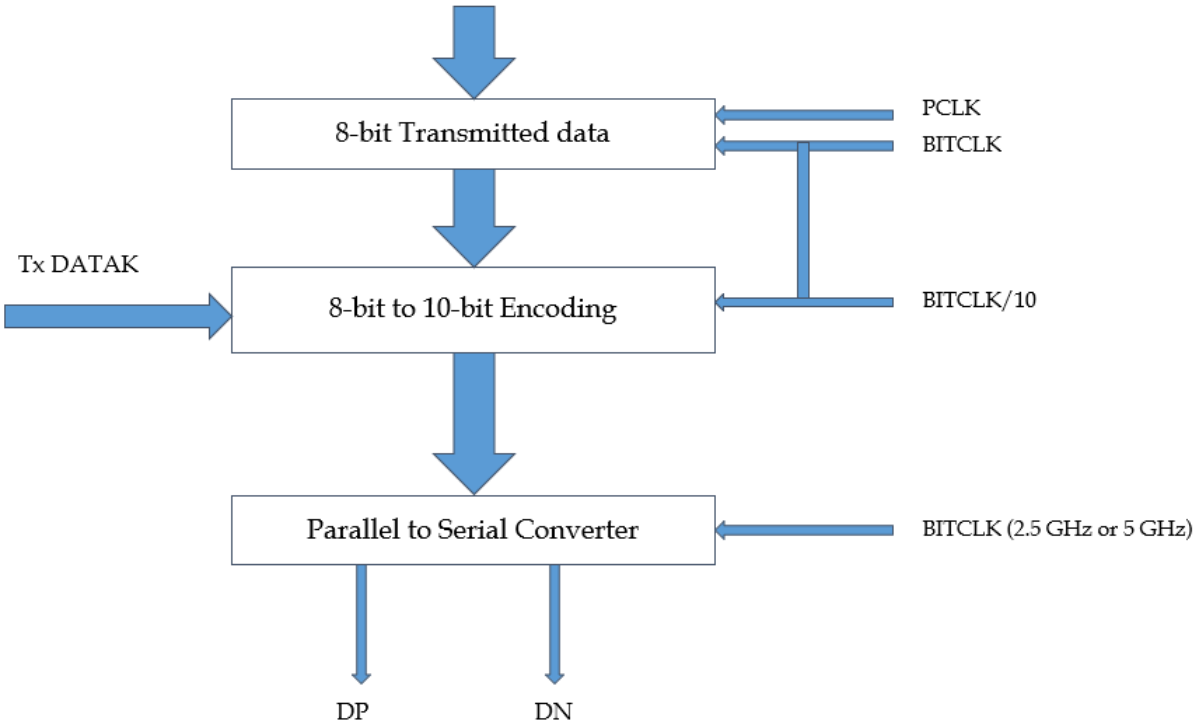


Fig. 3: Transmitter block diagram

The USB transmitter block is a vital component of USB communication systems, facilitating the transformation of parallel data into high-speed serialized data ready for transmission. Below is a detailed description of the modular stages involved in this process, from input processing to output signaling.

1. Width Selection (32, 16, or 8 Bits)

Purpose:

This stage ensures compatibility with data packets of varying sizes, which differ based on the

USB protocol version and application needs. It adapts incoming data by splitting or merging it to the required width, enhancing the transmitter's flexibility without requiring a fixed input format.

Process:

Incoming parallel data (32, 16, or 8 bits) is processed and formatted into 8-bit chunks for uniform compatibility with the next stage, the 8b/10b encoder.

Key Features:

- Prepares data by adjusting the input width to match the system requirements.
- Guarantees predictable data formatting for downstream processing blocks.

Output:

- An 8-bit parallel data stream synchronized with the Parallel Clock (PCLK).

2. 8b/10b Encoding Block

Purpose:

This block converts 8-bit input data into a 10-bit encoded format, ensuring data integrity and reliable high-speed transmission.

Key Functions:

- DC Balance: Maintains a balanced distribution of ones and zeros to avoid baseline wander.
- Error Detection: Introduces redundancy to identify transmission errors.
- Control Symbol Support: Differentiates between data and control symbols for protocol-specific operations.

Encoding Process:

- Converts 8-bit data or control bytes (indicated by TxDataK) into a 10-bit pattern.

- The additional 2 bits provide redundancy and special bit patterns for enhanced transmission quality.

Control Input (TxDataK):

- Specifies whether the input is a data byte or a control byte.
- Control bytes are used for specific USB commands, such as start-of-frame markers or idle signals.

Clocking:

- Operates at a frequency equal to 1/10th of the bit clock (**BITCLK/10**) to match the parallel data processing speed.

Output:

- A 10-bit parallel encoded data stream synchronized with the encoder clock.

3. Parallel-to-Serial Converter

Purpose:

This stage serializes the 10-bit parallel data into a high-speed bitstream for USB transmission. Serialization is critical for enabling high-speed communication over the USB link.

Process:

- The 10-bit data is loaded into a shift register and shifted out one bit at a time in sync with the BITCLK (e.g., 2.5 GHz for USB 2.0, 5.0 GHz for USB 3.0).
- Outputs two differential signals: DR (Data Positive) and DN (Data Negative).

Differential Output:

- DR and DN form a differential pair to enhance signal integrity and robustness against electromagnetic interference (EMI). This differential signalling ensures reliable transmission even over extended distances.

4. Clocking Scheme

The USB transmitter block employs multiple clock domains to synchronize its stages:

PCLK (Parallel Clock):

- Used for the initial parallel data processing stages.
- Operates at a lower frequency compared to the serialized output.

BITCLK:

- A high-speed clock used in the serialization process.
- Matches the USB link speed (e.g., 2.5 GHz for USB 2.0, 5.0 GHz for USB 3.0).

BITCLK/10:

- Drives the 8b/10b encoder to match the parallel data processing rate.

This modular design ensures that the USB transmitter block achieves high-speed, reliable data transmission while maintaining flexibility and robustness against signal degradation.

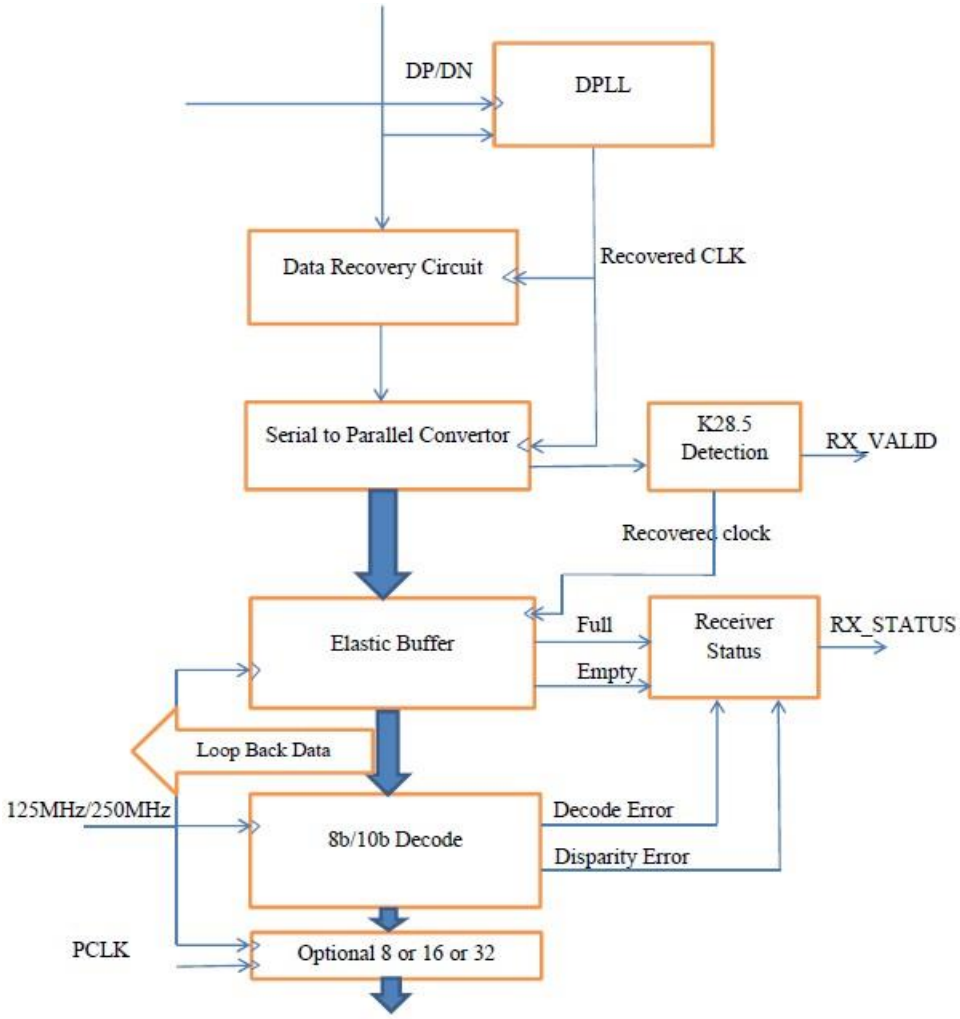


Fig. 4: Receiver block diagram

The USB receiver block diagram outlines the systematic process of receiving, decoding, and preparing high-speed USB signals for further processing. Below is a detailed explanation of each functional block, highlighting the data flow and transformation through the system.

1. Differential Inputs (DP/DN)

Purpose:

- Receives high-speed differential signals from the USB transmitter.

- Differential signalling ensures reliable data transfer by minimizing noise and interference, even over long cables.

Operation:

- The differential pair (DP and DN) carries complementary signals, with data encoded as the voltage difference between the two lines.
- Noise affecting both lines is cancelled out, enhancing the clarity of the transmitted signal.

Advantages:

- High immunity to electromagnetic interference (EMI).
- Reduces common-mode noise, improving data integrity.

Output:

- The raw differential signal is forwarded to the Digital Phase-Locked Loop (DPLL) for further processing.

2. Digital Phase-Locked Loop (DPLL)

Purpose:

Extracts a synchronized clock signal from the incoming differential data. USB signals lack a dedicated clock line, so the DPLL ensures proper timing for decoding.

Operation:

- Detects transitions (edges) in the data stream to synchronize the internal clock with the incoming signal.
- Ensures accurate sampling and timing of the serial data.

Output:

- A recovered clock signal, synchronized with the incoming data, is passed to the Data Recovery Circuit.

3. Data Recovery Circuit

Purpose:

Recovers the raw serial data stream from the differential input signal, using the clock recovered by the DPLL.

Operation:

- Aligns the serial data with the recovered clock to ensure accurate bitstream reconstruction.
- Compensates for distortions caused by noise or jitter, ensuring reliable data recovery.

Output:

- A clean, synchronized serial data stream is prepared for the next processing stage.

4. Serial-to-Parallel Converter

Purpose:

Converts the high-speed serial bitstream into a parallel format for efficient processing by subsequent blocks.

Operation:

- Serial data is loaded into a shift register, grouped into blocks (e.g., 10 bits for 8b/10b-encoded data), and output in parallel.
- Detects the K28.5 control symbol, commonly used in USB communication to indicate control packets or idle periods.
- Ensures proper alignment of the data stream for decoding.

Output:

- Parallel data along with a valid signal (RX_VALID) is forwarded to the Elastic Buffer.

5. Elastic Buffer

Purpose:

Addresses timing mismatches between the transmitter and receiver clock domains to prevent data loss.

Operation:

- Temporarily stores incoming data to smooth out differences in clock speeds.
- Detects and corrects underflow (empty buffer) or overflow (full buffer) conditions.
- Provides Receiver Status (RX_STATUS) signals, indicating the receiver's operational state.

Output:

- Parallel data synchronized with the receiver's processing clock is passed to the 8b/10b Decoder.

6. 8b/10b Decoder

Purpose:

Decodes the 10-bit encoded data back into its original 8-bit format. This process reverses the encoding performed at the transmitter.

Functions:

- Restores the original data while maintaining DC balance (equal distribution of ones and zeros).
- Detects errors, including:
    - Decode Errors: Invalid 10-bit patterns.

- Disparity Errors: Violations of the DC balance rule.

Output:

- Decoded 8-bit parallel data.
- Error flags to indicate decoding issues, which may prompt retransmission or error correction.

7. Optional Width Selection (8, 16, or 32 Bits)

Purpose:

Combines multiple 8-bit words into larger data widths (16-bit or 32-bit) to match the application's processing requirements.

Operation:

- Processes the parallel data in a format suitable for downstream systems.
- The Parallel Clock (PCLK**)** drives this block to synchronize the output with the system clock.

Output:

- Parallel data formatted as 8, 16, or 32 bits, ready for use by the USB controller or application layer.

8. Loopback Data Path

Purpose:

Provides a built-in mechanism for testing and debugging the receiver without requiring external connections.

Operation:

- Routes data from the Elastic Buffer back internally to validate the entire receiver chain.

- Useful during design, verification, and manufacturing stages to test receiver functionality.

Clocking Scheme

The USB receiver operates using multiple clock domains to ensure synchronization:

- Recovered Clock: Extracted from the differential input signal by the DPLL. Drives the Data Recovery and Serial-to-Parallel Converter stages.
- Parallel Clock (PCLK): Drives the later stages, such as the 8b/10b Decoder and Optional Width Selection block.
- Typical frequencies: 125 MHz or 250 MHz, depending on the USB protocol and speed.

IMPLEMENTATION

Overview of System Implementation

The USB PHY (Physical Layer) top module is responsible for handling the low-level physical interface between a USB device and the higher layers of USB protocol. This design encompasses the transmitter, receiver, clock generation, and data encoding/decoding functionalities.

Key Components:

1. Clock Generation:
    - The usb_clock_gen module generates multiple clocks required for different stages of the design, including 125 MHz, 250 MHz, and a high-speed bit clock (2.5 GHz).
2. Transmitter:
    - Data Rate Adjustment: The usb_data_rate module configures the PHY clock (PHY_PCLK) and status signals based on the USB PHY mode and rate.
    - 8b/10b Encoding: The usb_encoder performs 8b/10b encoding for DC-balanced data transmission, ensuring error detection capability.
    - Parallel to Serial Conversion: The usb_partoserial module converts 10-bit parallel data into serialized data for transmission.
3. Receiver:
    - Clock and Data Recovery (CDR): The usb_dpll_2 module recovers the clock from the incoming serialized data (DP) using a Phase-Locked Loop (PLL).
    - Serial to Parallel Conversion: The usb_sertopar module converts serialized input back into 10-bit parallel data.
    - Polarity Inversion: The usb_dff module optionally inverts the polarity of the recovered data as required.

- FIFO Buffering: The usb_fifo module buffers data for reliable bursting, handling data flow with fifowfull and fiforempty signals.

4. Data Decoding and Status Generation:

   - Decoding: The usb_decode module converts 10-bit encoded data back to 8-bit usable data.
   - Status Generation: The usb_rx_status module generates receive status and validity signals, providing feedback on the data buffer's state.

Signal Flow:

1. Transmitter:

   - TX_DATA and TX_DATAK are encoded and serialized into DP using 8b/10b encoding and parallel-to-serial conversion.
   - The PHY's clock and mode settings adjust the transmission parameters dynamically.

2. Receiver:

   - Serialized data (DP) is recovered, converted to parallel, optionally inverted, buffered, and finally decoded into RX_DATA and RX_DATAK.
   - Status signals like RX_STATUS and RX_VALID indicate the validity and state of the received data.

Submodules

1. USB_AASD:

   The usb_aasd module is a reset synchronizer that ensures a glitch-free and clock-synchronized reset signal (rst_out). It operates on the positive edge of the clock (clk) and the negative edge of the asynchronous reset (rst). When rst is deasserted (low), rst_out is set to 0, ensuring the system is reset. When rst is asserted (high), the module

synchronizes the reset using a two-stage process: updating a temporary register (temp) followed by updating rst_out. This design prevents metastability and ensures reliable reset behavior in high-speed circuits like USB PHYs.

2. USB_CLOCK_DIV:

   The usb_clock_div module is designed to generate a divided clock (BITCLK_10) from an input clock (CLK). It is used in the USB physical layer for serial communication, specifically when converting 10-bit data (produced by an 8b/10b encoder) into a serial bit stream. The module divides the input clock (CLK) to generate a slower clock (BITCLK_10) that operates at a reduced frequency suitable for the parallel-to-serial conversion process.

3. USB_CLOCK_GEN:

   The usb_clock_gen module is designed to generate three different clock signals: pclk_125 (125 MHz), pclk_250 (250 MHz), and bit_rate_clk (2.5 GHz), from a system clock clk operating at 5.0 GHz. The module is typically used in high-speed communication systems like USB, where precise clock frequencies are essential for data transfer and synchronization.

4. USB_DATA_RATE:

   The usb_data_rate module is responsible for generating the PCLK (Peripheral Clock) based on various input parameters, such as data rate, power mode, and transmission status. It also outputs a DATA_STATUS signal that indicates the state of the data transmission. The purpose of this module is to adapt the clock generation to different modes of operation, whether it's for high-speed USB, PCIe mode, or low-power states, depending on the configuration.

5. USB_DFF:

   The usb_dff module is a 10-bit data flip-flop with an inversion control feature, which operates based on the input clock (clk), reset (rst), and inversion signal (inv). It processes 10-bit input data (in), and depending on the state of the inv signal, it either inverts or passes the input data through to the output (out). This functionality is particularly useful

in scenarios where signal polarity needs to be adjusted, such as in data recovery or handling different encoding schemes in USB communication.

6. USB_PARTOSERIAL:

The usb_partoserial module is designed to perform parallel-to-serial conversion. It takes 10-bit parallel data as input (Parin), and converts it into a serial stream of data. The module uses two clocks, Serialclk for accepting parallel data and SerialBit for the actual serial conversion, making it suitable for high-speed data transmission like USB or other serial communication protocols.

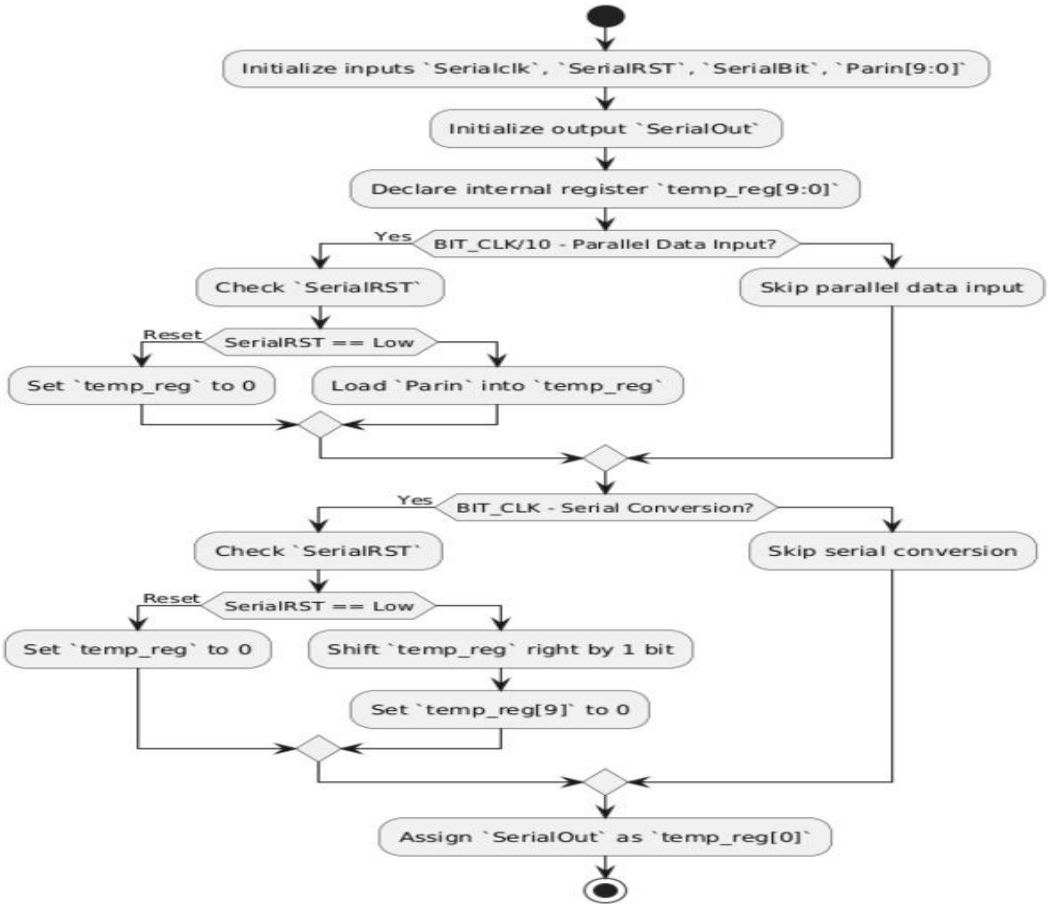


Fig. 5: Parallel to Serial Converter flowchart

7. USB_ENCODER:

The usb_encoder module is designed for applications in high-speed serial communication protocols, such as USB. It ensures that the transmitted data maintains proper DC balance, which is critical for reliable transmission and reception of the data without signal degradation. The module generates 10-bit DC-balanced encoded data from an 8-bit input, which is then ready for transmission over a high-speed interface.

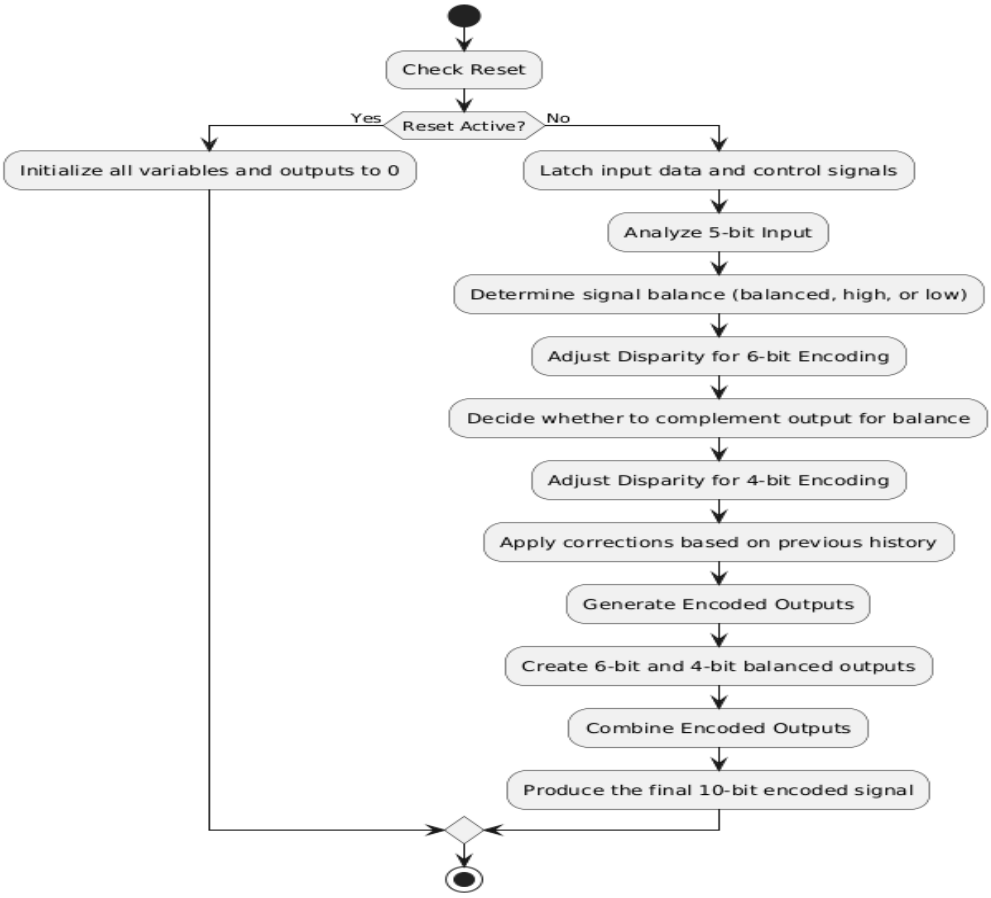


Fig. 6: Encoder flowchart

8. USB_DPLL:

The usb_dpll_2 module is a Digital Phase-Locked Loop (DPLL) designed to synchronize a local clock (SYS_CLK) with an incoming serial signal (REF_IN). The primary function of this module is to generate a clock signal (DPLL_OUT) that is phase-locked to the incoming data, ensuring that the local clock is aligned with the serial data's timing. This

is crucial for receivers in digital communication systems, such as USB, where accurate clock recovery from the data stream is needed for proper data sampling and synchronization.

9. USB_FIFO:

The usb_fifo module is a First-In-First-Out (FIFO) buffer used for data bursting in digital communication systems, particularly for managing read and write operations in a USB communication interface. It allows data to be written into a memory buffer and read out at different rates while ensuring data integrity and flow control. This module is designed to add or remove SKP (skip) symbols, often necessary in USB and other communication protocols, by managing read and write pointers.

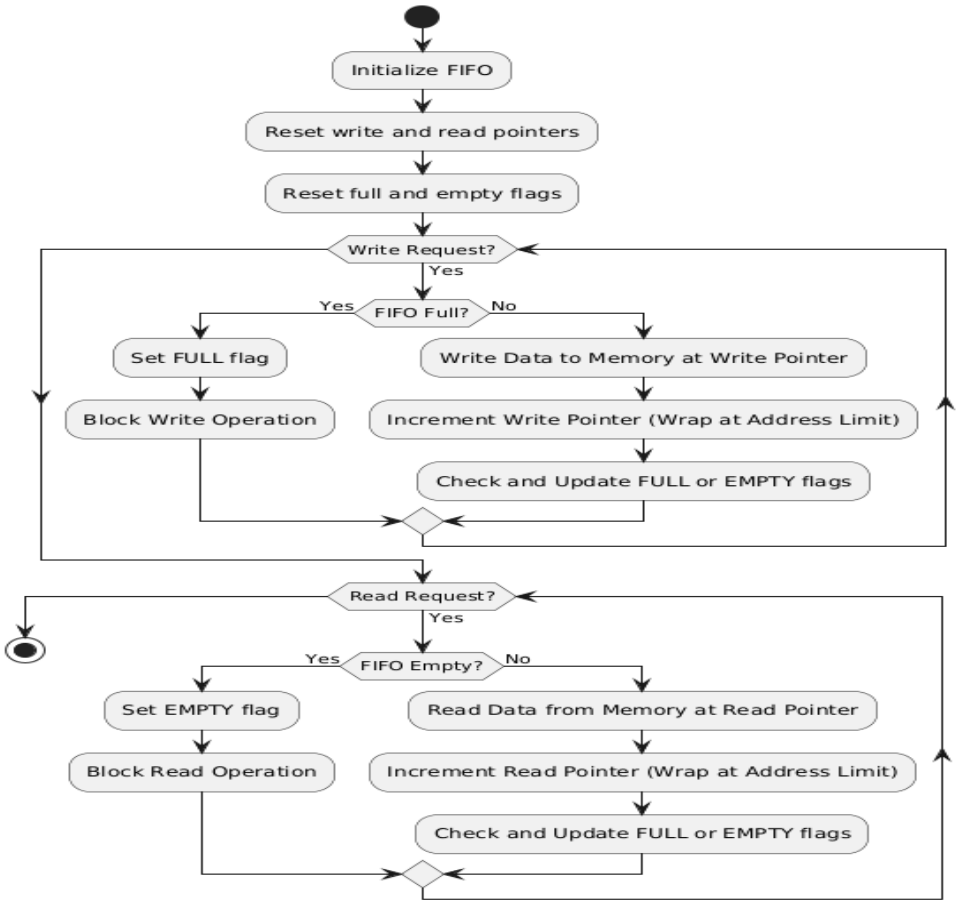


Fig. 7: FIFO flowchart

10. USB_SERTOPAR:

The usb_sertopar module is designed to convert serial data (1-bit at a time) into parallel data (10 bits). This process is commonly used in communication systems where serial data is transmitted and needs to be reassembled into parallel form for further processing. The module takes a serial bit as input and shifts it into a 10-bit parallel register, effectively converting the incoming serial stream into a 10-bit wide parallel output.

11. USB_DECODE:

The usb_decode module is designed to decode a 10-bit DC-balanced data stream into 8-bit data for further processing. DC-balanced encoding is commonly used in communication systems, like USB, to ensure that the transmitted signal maintains a balanced voltage level over time. The module performs a complex logic operation to decode the 10-bit data into an 8-bit output using various gates, conditions, and registers.

12. USB_RX_STATUS:

The usb_rx_status module is designed to generate the appropriate RxStatus output signal based on the conditions of the received parallel data. It uses several control signals to determine the status of the receiver.

RESULTS AND ANALYSIS

1. Simulation

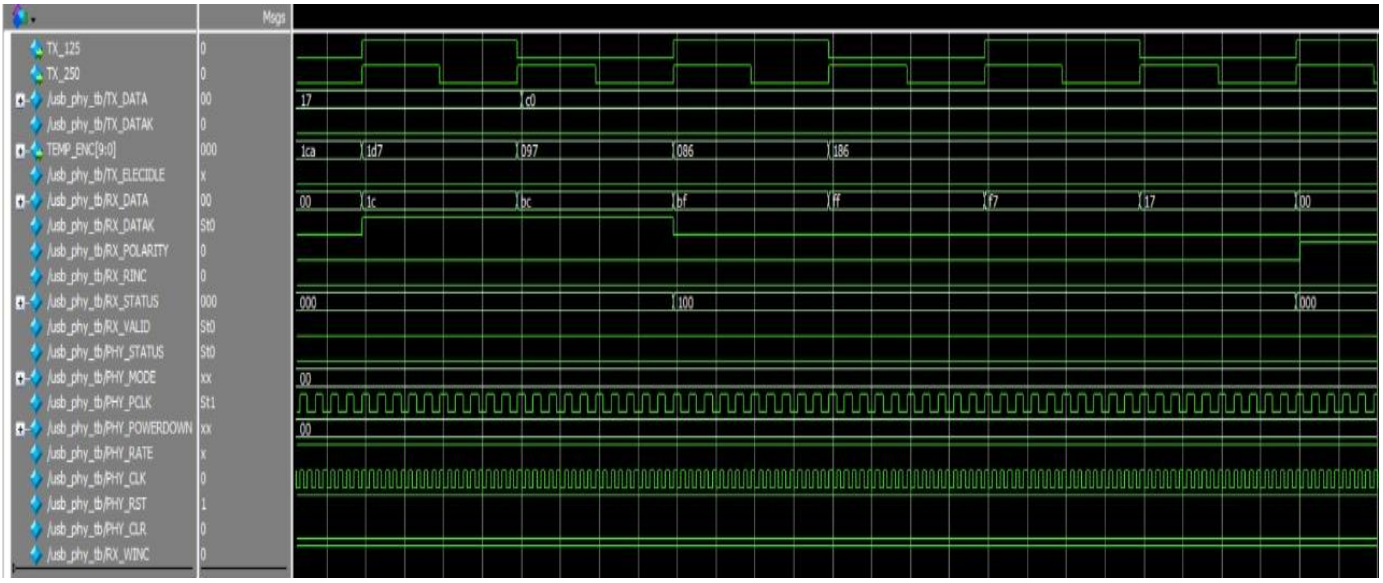


Fig. 8: Simulation of USB_PHY module

The provided waveform represents simulation of a digital communication system, likely involving USB PHY (Physical Layer) signals based on the labels.

1. Signal Labels

Each row in the waveform corresponds to a specific signal in the design under test. Here's what the key signals might represent based on their naming:

- TX_125 and TX_250: Likely represent clock signals at 125 MHz and 250 MHz. These might be used for synchronization in the USB PHY.
- TX_DATA and TX_DATAK: Represent the transmitted data and data strobe (control or special signalling bits) in USB communication.
- TEMP_ENC[0:0]: Possibly an internal signal related to encoding/decoding or some form of control logic for data manipulation.

- RX_ELECT_IDLE, RX_DATA, and RX_POLARITY: These are likely receiver-related signals. For example:
    - RX_ELECT_IDLE may indicate the idle state of the receiver's electrical signals.
    - RX_DATA could be the received data stream.
    - RX_POLARITY might represent the polarity of the received signal, possibly for differential signalling.
- RX_VALID and RX_STATUS: Indicate whether received data is valid and any status information about the reception process.
- PHY_CLK, PHY_RST, and PHY_MODE: These are signals associated with the physical layer's operation. For example:
    - PHY_CLK could be the clock signal used internally.
    - PHY_RST might indicate a reset for the PHY block.
    - PHY_MODE might set the operation mode (e.g., USB speed settings like High-Speed, Full-Speed, etc.).
- PHY_POWERDOWN: Indicates power-saving modes for the PHY layer.
- RX_RATE: Likely represents the data rate at which the receiver is operating.
- RX_WINC: Could relate to a receiver window control signal.

2. Observations in the Waveform

Clock Signals (PHY_CLK and others):

- The clock signals (e.g., PHY_CLK) are toggling periodically and act as the timing reference for the digital system.

Data Transmission (TX_DATA and RX_DATA):

- TX_DATA shows various hexadecimal values (e.g., 0x1C, 0xD97, 0x186, etc.), representing the data being transmitted at specific clock cycles.
- RX_DATA might correspond to the received version of the transmitted data, though it may be delayed or modified, depending on the transmission conditions.

Control Signals:

- Signals like TX_DATAK, RX_VALID, and RX_STATUS indicate control information accompanying the data:
    - For example, RX_VALID transitions to high (1) to indicate valid received data.
    - RX_STATUS might provide additional information, such as error codes or framing status.

PHY Layer Status:

- The PHY_MODE and PHY_POWERDOWN signals suggest changes in the operational state of the PHY:
    - PHY_POWERDOWN might be asserted to indicate entry into a low-power state.
    - Changes in PHY_MODE could indicate shifts between different USB operating modes.

3. Key Events in the Waveform

- Clocking Events: The toggling of PHY_CLK and related signals ensures synchronous operation of the PHY layer.
- Data Transmission: Changes in TX_DATA and associated control signals reflect the flow of transmitted data packets.

- Receiver Idle State: The RX_ELECT_IDLE signal appears active (high) at certain intervals, likely indicating no active data reception.
- Data Validation: RX_VALID being asserted suggests that the receiver has successfully captured and validated the incoming data.

The waveform showcases the interaction of various signals in a USB PHY block, highlighting the following:

- Clock synchronization using signals like PHY_CLK.
- Transmission of data (TX_DATA) with accompanying control (TX_DATAK).
- Reception and validation of data (RX_DATA, RX_VALID).
- Indication of idle states and power modes (RX_ELECT_IDLE, PHY_POWERDOWN).
- Support for error handling and state management (RX_STATUS, PHY_MODE).

This simulation validates the functionality of the USB PHY design by monitoring its behaviour during transmission, reception, and various operational states.

2. Transmitted Sequence Data

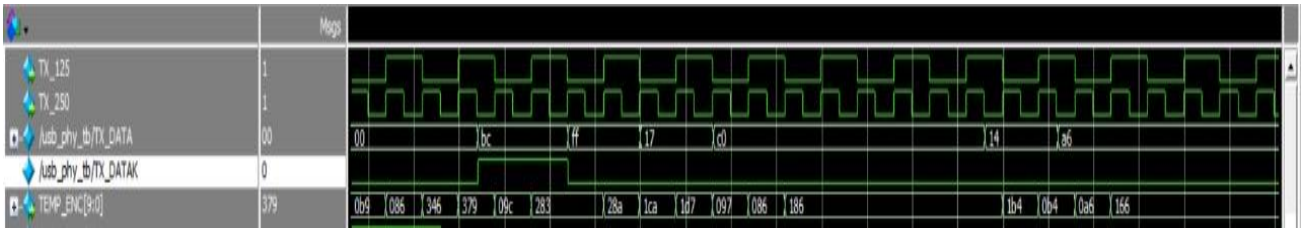


Fig. 9: Transmitted Sequence

The waveform displays a simulation of digital signals, likely from a USB PHY testbench, as indicated by the labels.

1. Signal Descriptions

Clock Signals

- TX_125 and TX_250:
    - These are clock signals running at different frequencies (125 MHz and 250 MHz, respectively).
    - Their periodic toggling indicates proper clock operation and synchronization in the design.

Data Transmission Signals

- TX_DATA:
    - This signal shows the data being transmitted. Values like 00, bc, ff, 17, and others are hexadecimal representations of the transmitted data bytes.
    - The signal changes in synchronization with the clock signals (TX_125 or TX_250), ensuring correct timing.

- TX_DATAK:
    - This is a control signal used to distinguish between data and control characters in the USB protocol.
    - A 0 indicates data, and a 1 likely represents control information.
    - In this waveform, the TX_DATAK remains 0, indicating continuous data transmission.

Temporary Encoding (TEMP_ENC[9:0])

- TEMP_ENC[9:0]:
    - This signal likely represents intermediate or encoded data related to the transmitted data.
    - It changes values such as 0b9, 346, 379, and others, possibly derived from TX_DATA after an encoding process, e.g., 8b/10b encoding used in high-speed serial protocols like USB.

2. Key Observations

i. Clock Synchronization:

a. The clock signals (TX_125 and TX_250) toggle consistently, acting as the backbone for data timing. The waveform shows that the data and control signals are correctly synchronized with these clocks.

ii. Data and Encoding:

a. The transmitted data (TX_DATA) changes every few clock cycles, indicating sequential data transmission.

b. The TEMP_ENC[9:0] reflects the encoded version of the transmitted data. For example:

i. When TX_DATA = 00, the encoded data (TEMP_ENC) is 0b9.

ii. When TX_DATA = bc, the encoded data is 346.

iii. Control Signal Behaviour:

a. The TX_DATAK signal remains constant (0), confirming that no control characters are being sent during this portion of the simulation.

This part of the whole waveform represents a simulation of data transmission in a USB PHY design. The key takeaways are:

- Clock signals (TX_125, TX_250) provide synchronization for data flow.
- Transmitted data (TX_DATA) changes periodically, while its encoded version (TEMP_ENC) reflects proper encoding logic.
- The TX_DATAK signal indicates that only data (not control characters) is being sent during this simulation.

The consistent signal transitions and values suggest that the PHY module is functioning as expected, ensuring proper timing, data encoding, and transmission.

3. Change in PCLK Frequency

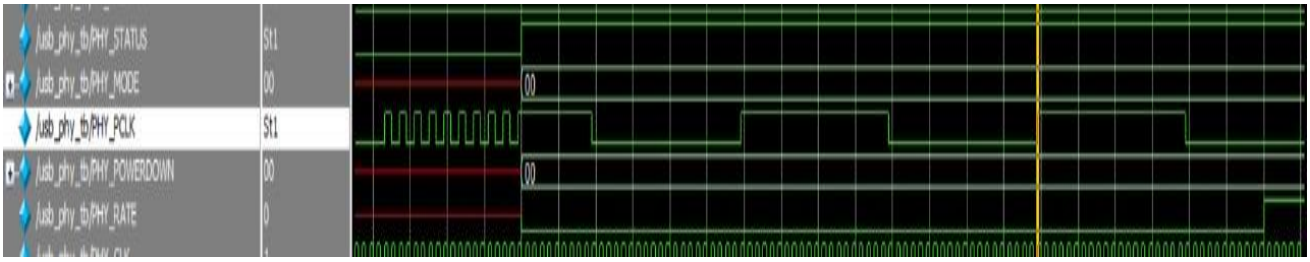


Fig. 10: Change in clock frequency

This waveform appears to show signals related to the operation of a USB PHY interface, focusing on control and status behaviour.

1. Signal Descriptions

Control and Status Signals:

1. PHY_STATUS:
    - This signal likely reflects the operational status of the PHY layer. It may indicate whether the PHY is active, idle, or transitioning between states.
    - In the waveform, the PHY_STATUS signal is labelled as St1, which might correspond to an active state.
2. PHY_MODE:
    - This signal could specify the mode of operation for the PHY. Modes might include USB speed modes (e.g., High-Speed, Full-Speed, etc.) or other PHY configurations.
    - The signal is steady (00) throughout this section, indicating a consistent operational mode.

3. PHY_POWERDOWN:

    - This signal controls or reflects the power state of the PHY. A value of 00 likely indicates that the PHY is in a fully active or operational state.
    - The waveform shows no assertion of the power-down signal, so the PHY remains active.

Clock and Rate Signals:

4. PHY_PCLK:

    - This is the PHY clock signal, which toggles periodically and drives the timing for PHY operations.
    - The consistent toggling ensures proper synchronization and data flow.

5. PHY_RATE:

    - This signal might represent the data rate or speed setting for the PHY interface.
    - In the waveform, the signal appears to remain steady (00), possibly indicating the default rate or an idle condition.

6. **RX_CLK:**

    - Likely a clock signal specific to the receiver portion of the PHY.
    - The consistent toggling of this signal ensures proper synchronization for the reception of data.

2. Observations

1. Clock Activity:
    - The clock signals (PHY_PCLK and RX_CLK) toggle consistently, indicating proper operation of the timing mechanism.
2. Status and Control Consistency:
    - The PHY_STATUS signal remains steady (St1), indicating a stable operational state for the PHY.
    - PHY_MODE and PHY_POWERDOWN are not changing, suggesting that the PHY is operating in its normal mode without transitioning to a power-saving state.
3. No Power Down:
    - The PHY_POWERDOWN signal remains 00, confirming that the PHY is fully active and not in a low-power state.
4. Rate and Operational State:
    - The PHY_RATE remains at 00, indicating either an idle rate or a specific fixed configuration.

3**.** Key Points

- Stable Operation: The waveform suggests that the PHY is functioning as intended, with no mode changes or power-saving transitions occurring.
- Clock Synchronization: The toggling of PHY_PCLK and RX_CLK ensures reliable timing for PHY operations.
- Active State: The PHY_STATUS and PHY_POWERDOWN signals confirm that the PHY is fully active and operational.

This part of the whole waveform illustrates a stable USB PHY operation in an active mode with no transitions to power-saving states. Clock signals (PHY_PCLK, RX_CLK) are toggling as expected, while status and control signals (PHY_STATUS, PHY_MODE, and PHY_POWERDOWN) indicate a consistent, active state of the PHY layer.

4. Received Data Sequence

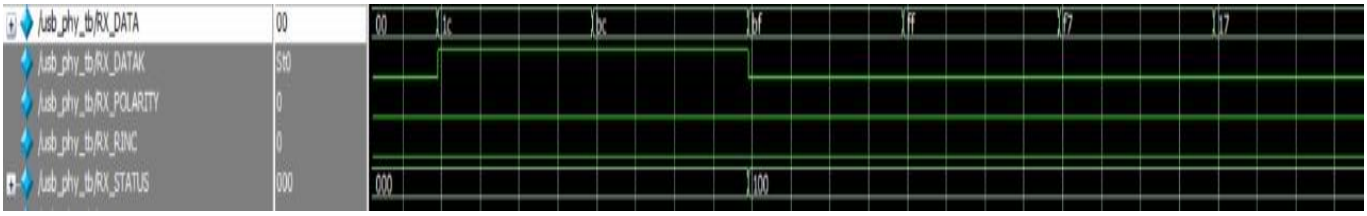


Fig. 11: Receiver Sequence

The image shows a digital waveform that represents signals from a simulation or testing environment.

1. RX_DATA:

   This signal shows a sequence of values being received. The hexadecimal values (00, 1C, BC, BF, etc.) represent the data transmitted at each clock cycle.

2. RX_DATAK:

   This appears to be a control signal associated with RX_DATA, possibly indicating the type of data being transmitted or its validity. The transitions in this signal (e.g., changing state) suggest significant events in data handling.

3. RX_POLARITY:

   A binary signal (either 0 or 1) that may represent the polarity of the data or signal line. Here, it remains constant at 0, implying no polarity change.

4. RX_RINC:

   This signal might indicate a readiness or increment condition for reading or processing received data. It is low (0) in this capture, suggesting no increment.

5. RTL Synthesis

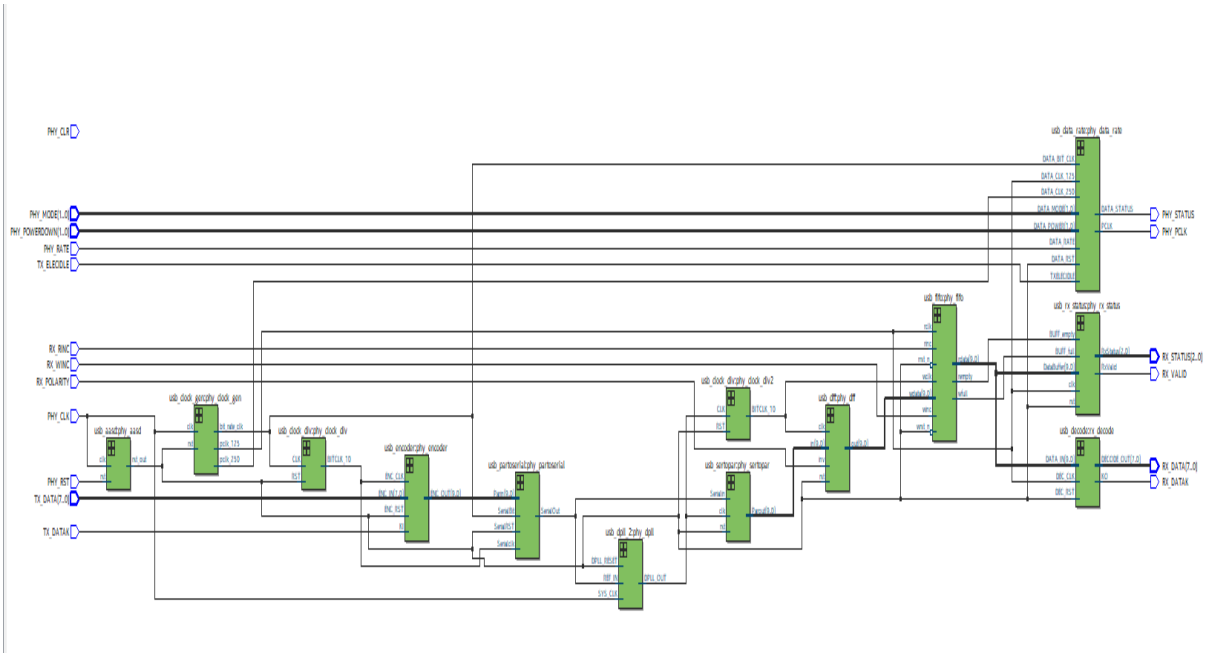


Fig. 12: RTL Synthesis

The diagram is a part of the RTL synthesis process, where Quartus Prime is converting your high-level VHDL or Verilog code into a detailed gate-level implementation. Each module shown in the diagram will correspond to either a complex logical function or a small circuit made of flip-flops and combinational logic, ultimately intended to run on an FPGA. The synthesis ensures that the design meets timing constraints and utilizes the FPGA's resources efficiently.

6. Netlist (Top module and Sub-modules)

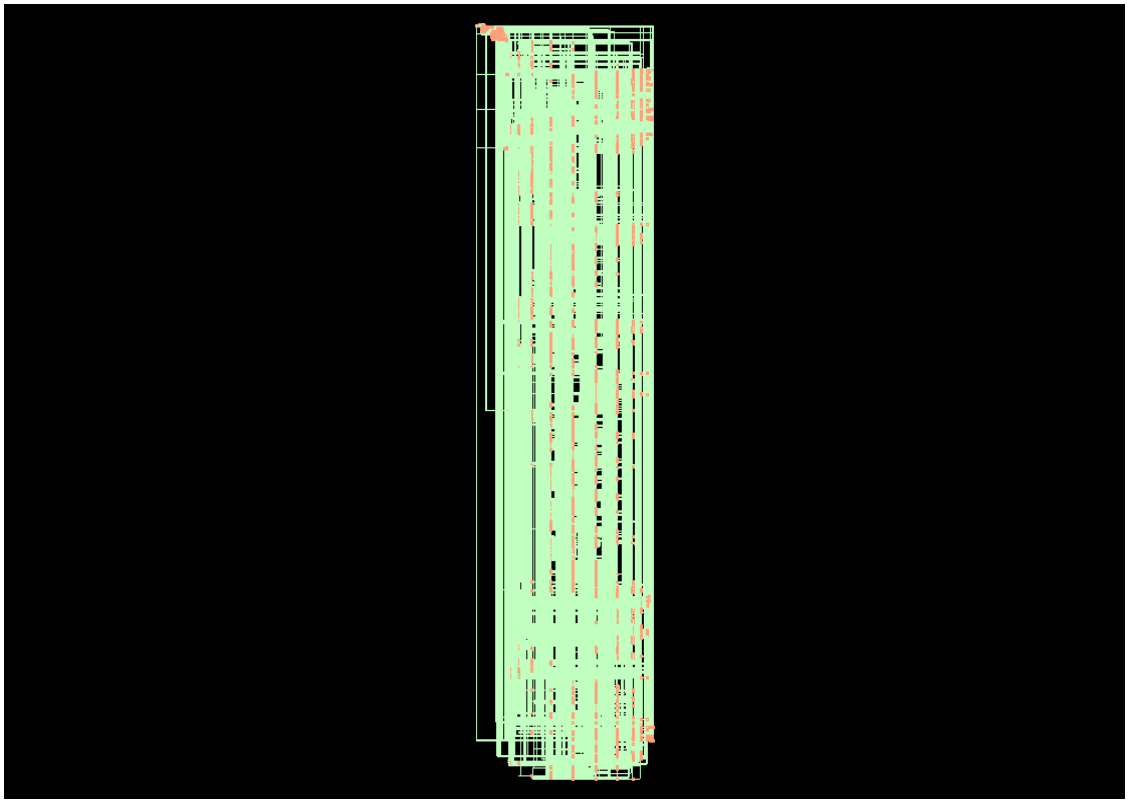


Fig. 13: Gate level netlist of USB_PHY module

The figure above represents the gate-level netlist generated by the Cadence Genus tool. This netlist provides a detailed view of the design after synthesis, where the high-level Verilog or VHDL description of the circuit is transformed into a network of logical gates and flip-flops.

Key Insights:

1. Gate-Level Netlist:
    - This is the lowest-level representation of the design, showing the individual gates (such as AND, OR, NOT) and flip-flops, as well as how they are interconnected to perform the desired logic functions.

2. Synthesis Overview:
    - The Cadence Genus tool performs the process of synthesis, which involves optimizing and mapping the RTL (Register Transfer Level) description into a gate-level format suitable for further steps like placement, routing, and physical implementation on hardware (FPGA/ASIC).
3. Design Optimization:
    - This netlist might also reflect optimization strategies applied during synthesis to improve area, speed, or power consumption, depending on the constraints and settings chosen during the synthesis process.

This gate-level netlist is a crucial step in hardware design as it bridges the gap between high-level abstraction and the physical implementation of the circuit. Below are the netlist for submodules:

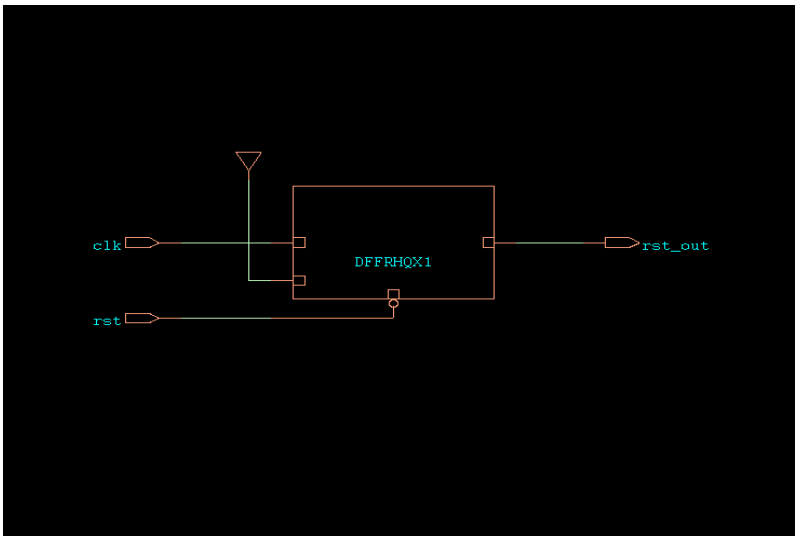


Fig. 14: Gate level netlist of USB_AASD module

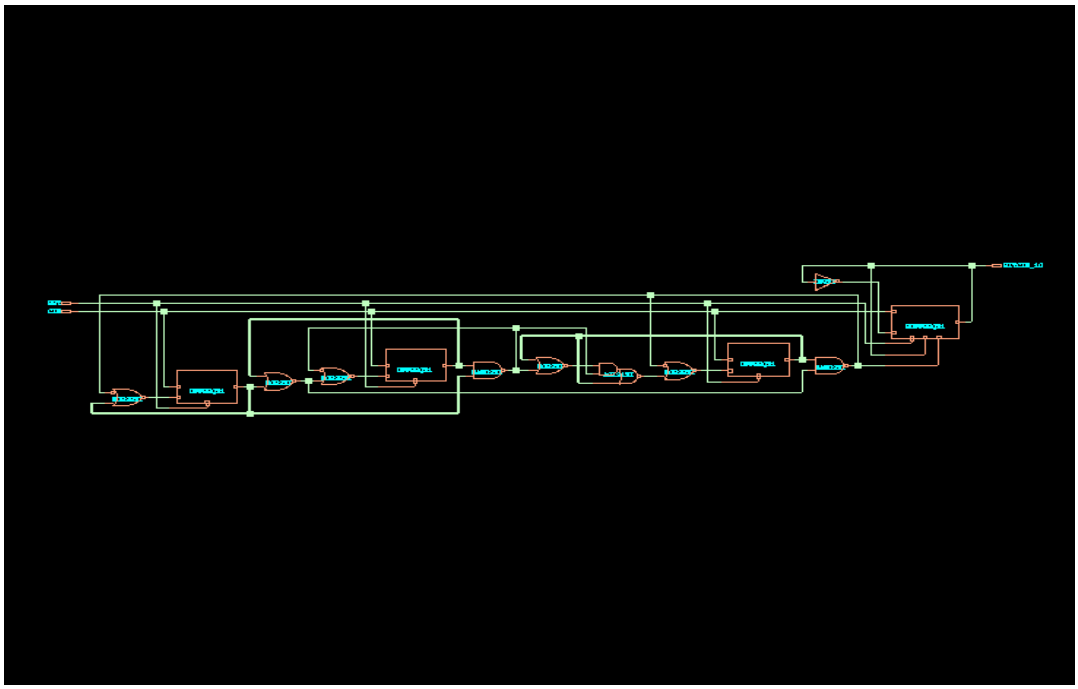


Fig. 15: Gate level netlist of USB_CLOCK_DIV module

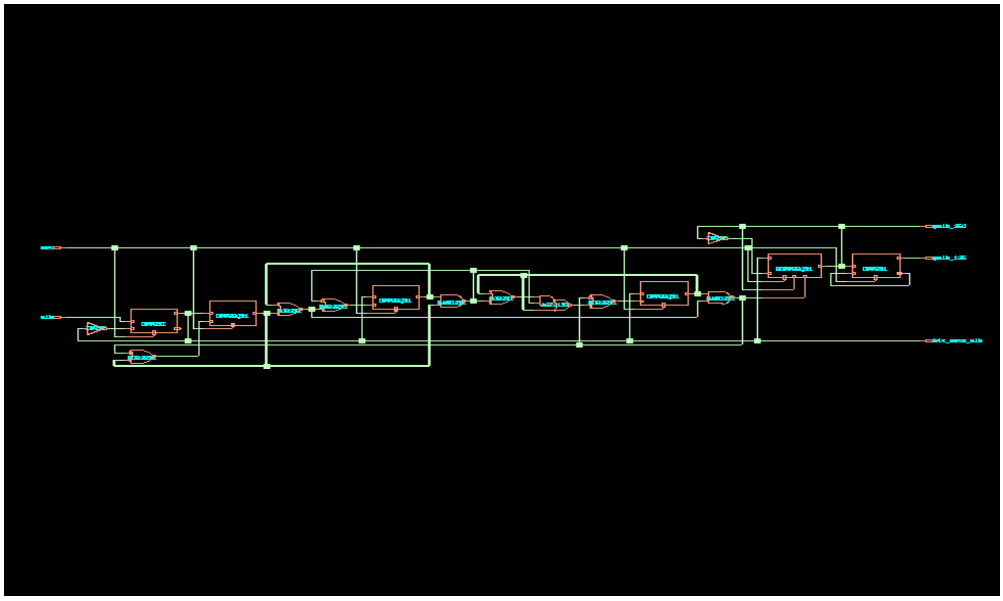


Fig. 16: Gate level netlist of USB_CLOCK_GEN module

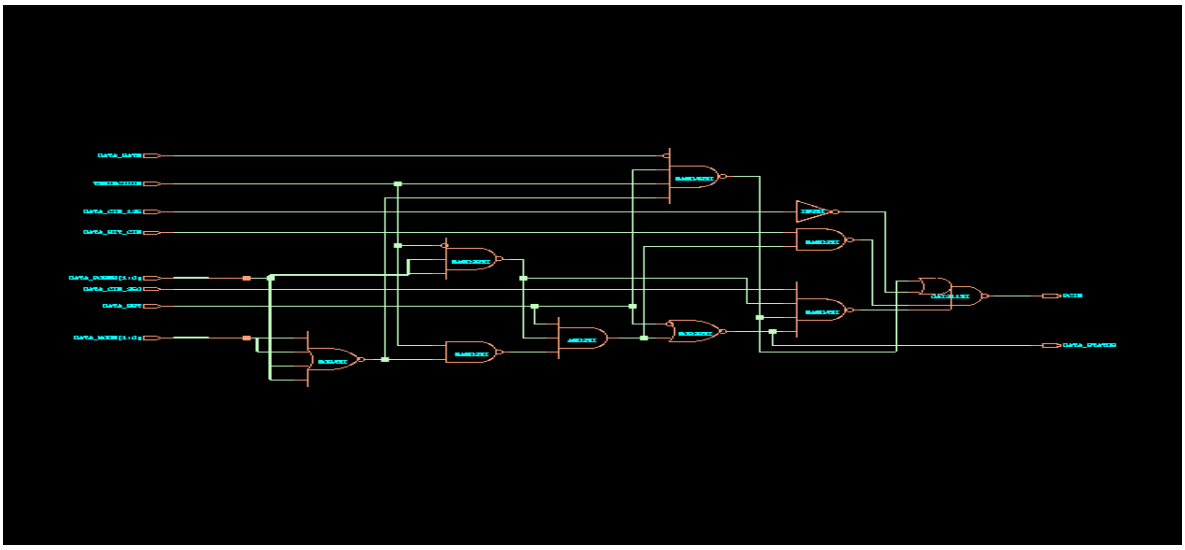


Fig. 17: Gate level netlist of USB_DATA_RATE module

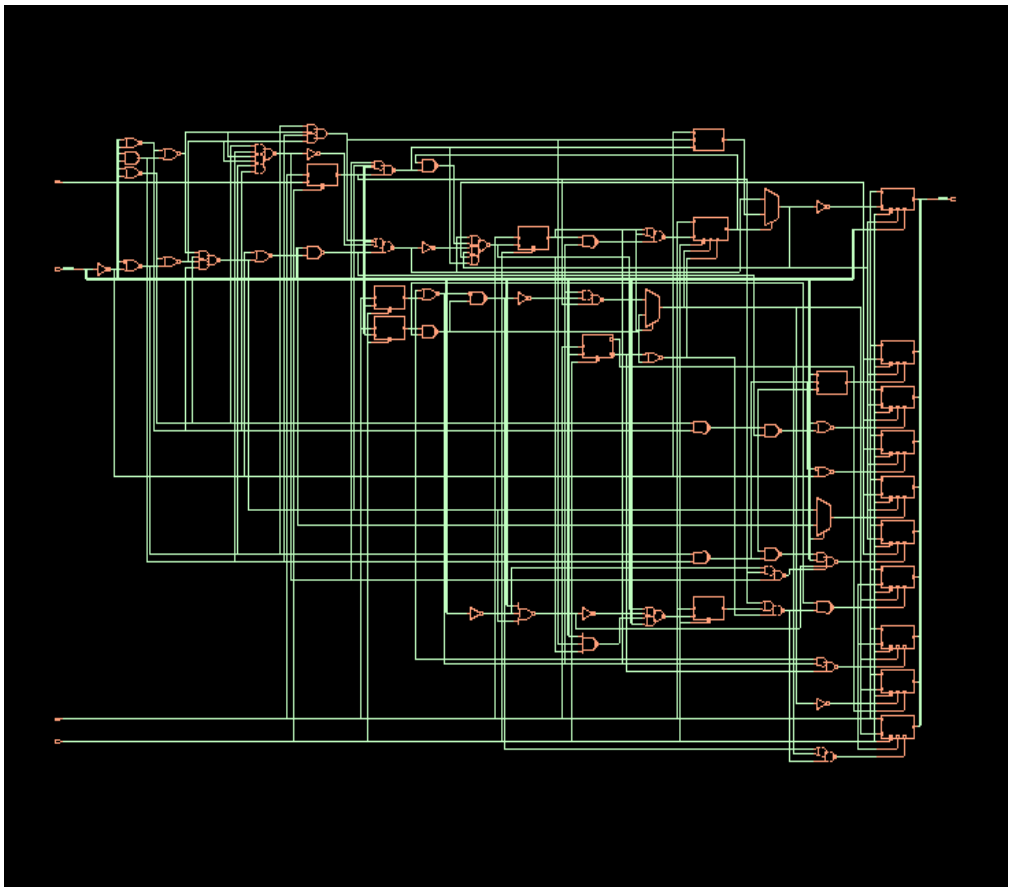


Fig. 18: Gate level netlist of USB_ENCODER module

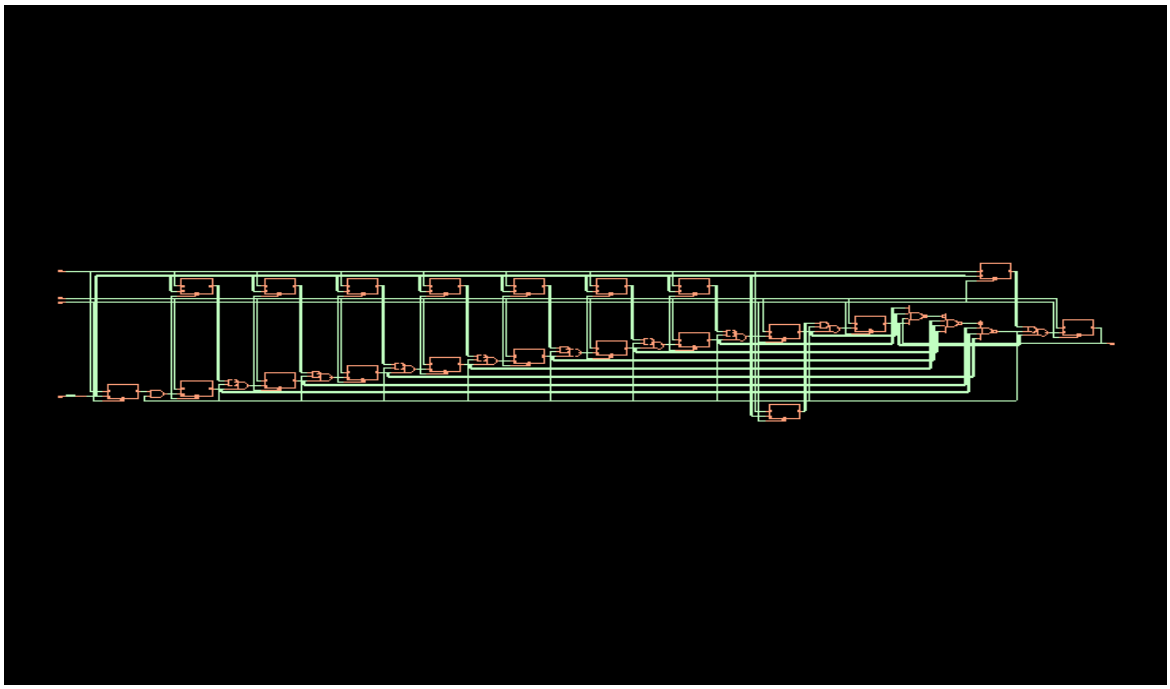


Fig. 19: Gate level netlist of USB_PARTOSERIAL module

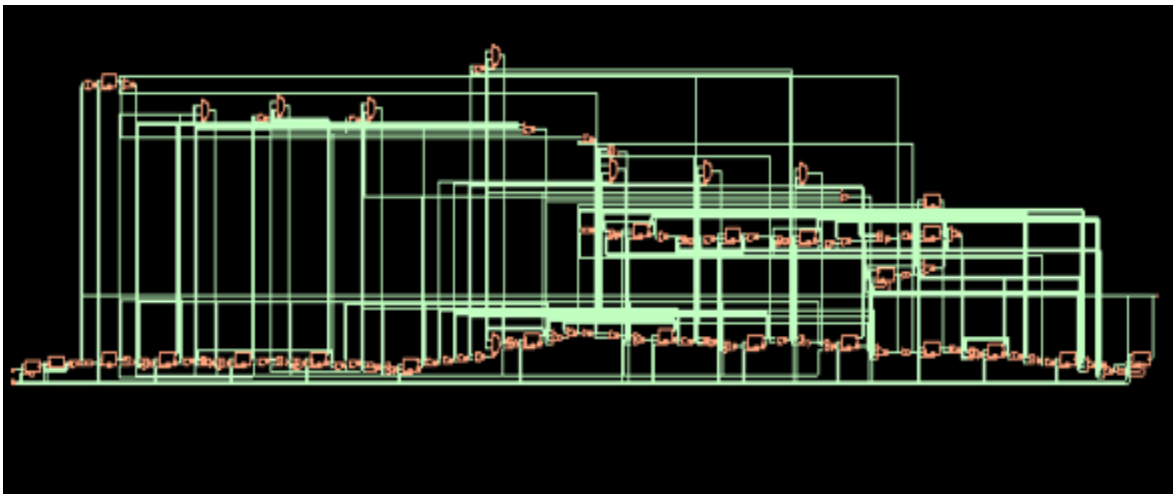


Fig. 20: Gate level netlist of USB_DPLL module

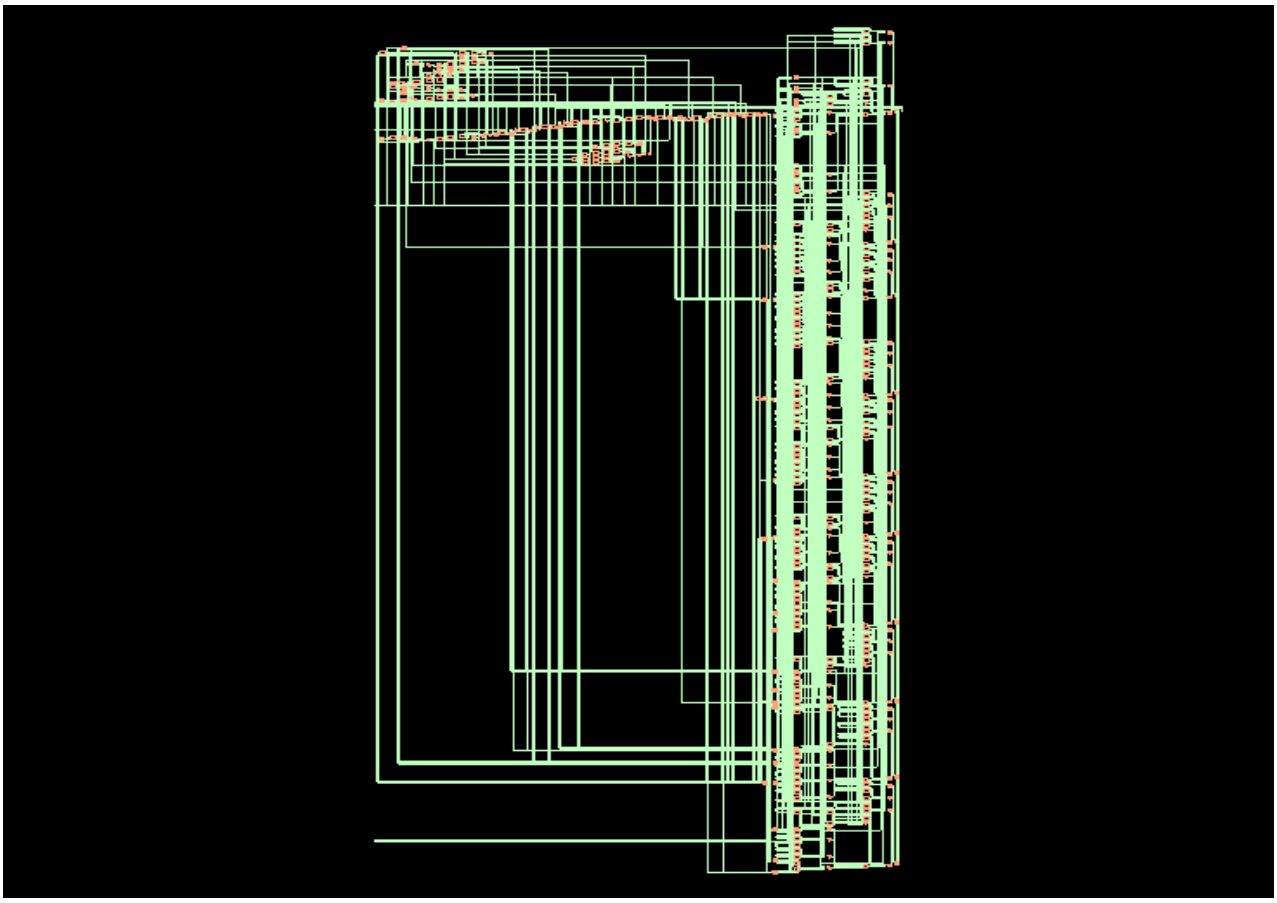


Fig. 21: Gate level netlist of USB_FIFO module

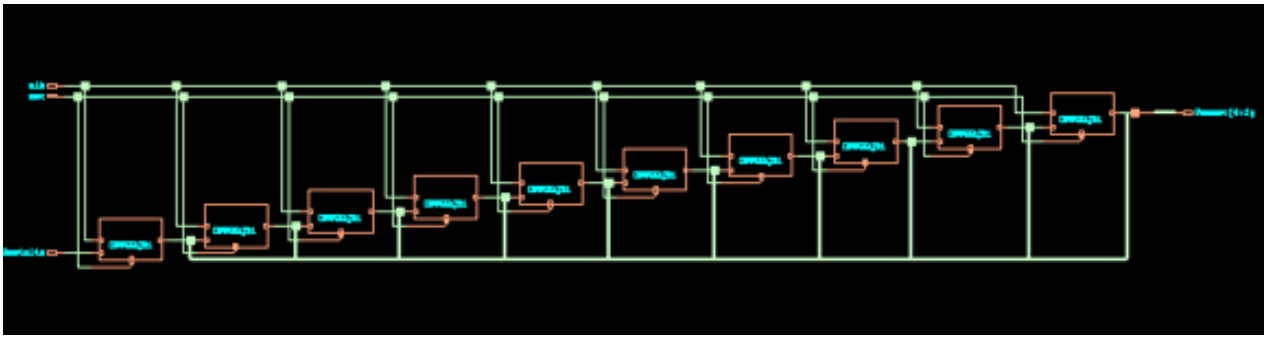


Fig. 22: Gate level netlist of USB_SERTOPAR module

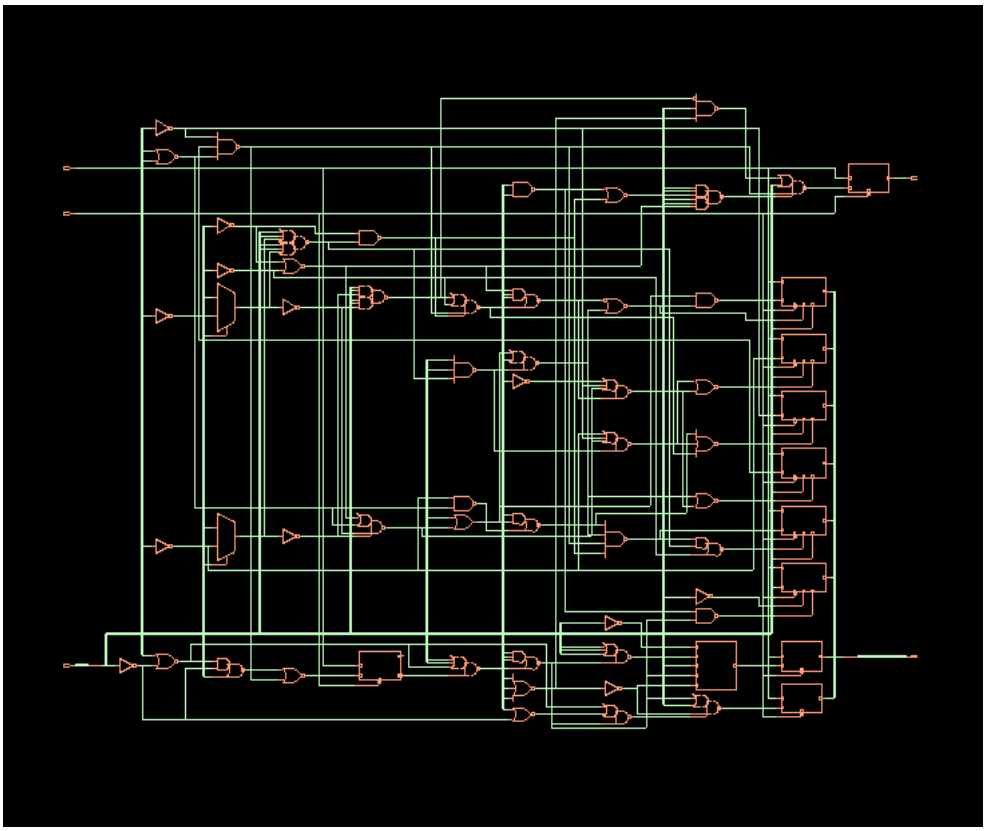


Fig. 23: Gate level netlist of USB_DECODE module

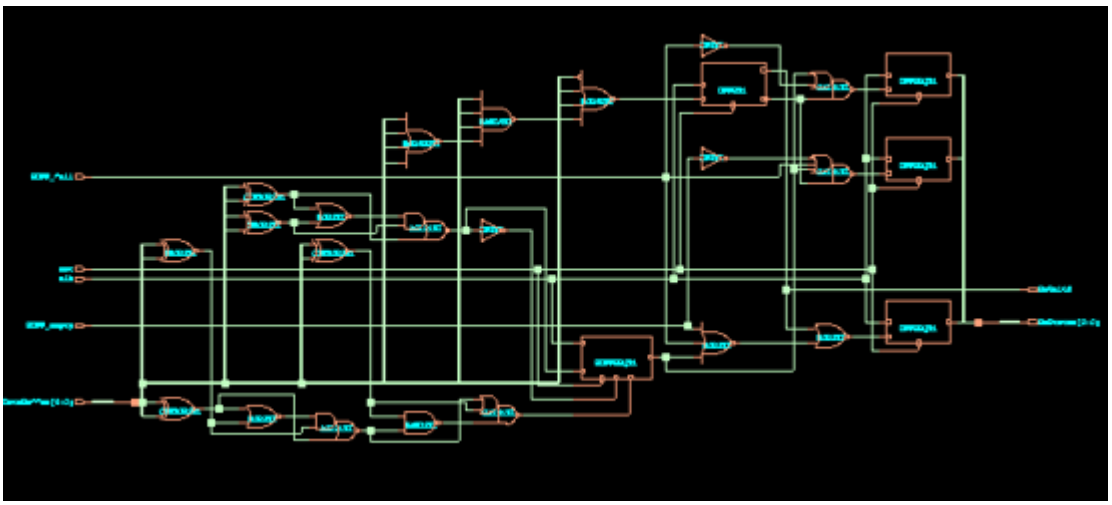


Fig. 24: Gate level netlist of USB_RX_STATUS module

7. Reports

After synthesizing the design using the Cadence Genus tool, we generated the gate-level netlist, which represents the design in terms of standard cells and logic gates. Following the netlist generation, various reports were obtained, including the cell area, power, gate count, timing, and Quality of Results (QoR) reports. These reports provide critical insights into the design's efficiency, performance, and resource utilization, enabling a comprehensive evaluation of the synthesized circuit against the specified constraints and objectives. Below is a summary of each report:

1. Cell Area Report:

The cell area report provides a detailed breakdown of the total area consumed by the design after synthesis. It includes contributions from all standard cells and macros used in the design. This report is crucial for determining whether the design fits within the target area constraints, enabling designers to assess the trade-offs between performance and area during optimization. It also highlights the area utilization for each module, allowing for targeted improvements if required.

- Total Cell Area: 70,350.070 μm^2
- Cell Count: 5,518 instances

```
====================================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Nov 20 2024  02:23:34 pm
  Module:                 usb_phy
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
====================================================================

   Instance      Module    Cell Count  Cell Area  Net Area   Total Area   Wireload
-------------------------------------------------------------------------------------
usb_phy                          5518  70350.070     0.000    70350.070 <none> (D)
  phy_encoder usb_encoder          67    616.873     0.000      616.873 <none> (D)
  (D) = wireload is default in technology library
```

Fig. 25: Cell area report

2. Power Report:

The power report summarizes the power consumption of the synthesized design, categorized into dynamic power (due to switching activity) and static power (leakage current in transistors). This report helps identify high-power-consuming modules and provides insight into the power efficiency of the design. Designers use this information to optimize for low power, especially in battery-powered or energy-sensitive applications.

- Total Power Consumption: 11.8517 mW
    - Leakage Power: 0.4091 mW (3.45%)
    - Internal Power: 10.4897 mW (88.51%)
    - Switching Power: 0.9529 mW (8.04%)
- Power Breakdown by Category:
    - Registers: 10.7199 mW (90.45%)
    - Logic: 0.9674 mW (8.16%)
    - Clock: 0.1644 mW (1.39%)

Instance: /usb_phy
Power Unit: W
PDB Frames: /stim#0/frame#0

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
register	3.70685e-04	9.79579e-03	5.53414e-04	1.07199e-02	90.45%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	3.84509e-05	6.93881e-04	2.35068e-04	9.67400e-04	8.16%
bbox	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
clock	0.00000e+00	0.00000e+00	1.64430e-04	1.64430e-04	1.39%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
Subtotal	4.09136e-04	1.04897e-02	9.52911e-04	1.18517e-02	100.00%
Percentage	3.45%	88.51%	8.04%	100.00%	100.00%

Fig. 26: Power report

3. Timing Report:

The timing report evaluates the performance of the design by analysing critical paths and ensuring that the timing requirements, such as setup and hold times, are met. It identifies any timing violations and provides details about the worst slack, critical paths, and clock frequencies. This report is essential for determining the maximum operating frequency of the design and for verifying its functional correctness under timing constraints.

```
====================================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Nov 20 2024  02:23:34 pm
  Module:                 usb_phy
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
====================================================================


Path 1: VIOLATED (-964 ps) Setup Check with Pin phy_dpll_filtercount_reg[6]/CK->D
          Group: PHY_CLK
     Startpoint: (R) phy_dpll_filtercount_reg[2]/CK
          Clock: (R) PHY_CLK
       Endpoint: (F) phy_dpll_filtercount_reg[6]/D
          Clock: (R) PHY_CLK

                    Capture       Launch
       Clock Edge:+     200            0
       Src Latency:+      0            0
       Net Latency:+      0 (I)        0 (I)
           Arrival:=    200            0

             Setup:-    145
     Required Time:=     55
      Launch Clock:-      0
         Data Path:-   1019
             Slack:=   -964

#-------------------------------------------------------------------------------------------------------------
#         Timing Point           Flags   Arc   Edge   Cell     Fanout Load Trans Delay Arrival Instance
#                                                                (fF)  (ps)  (ps)   (ps)  Location
#-------------------------------------------------------------------------------------------------------------
  phy_dpll_filtercount_reg[2]/CK -       -     R     (arrival)     24    -     0     0       0    (-,-)
  phy_dpll_filtercount_reg[2]/Q  -       CK->Q R     DFFRX2         1  7.8    85   366     366    (-,-)
  g21/Y                          -       A->Y  F     CLKINVX3       2 18.6    60    63     429    (-,-)
  g135698/Y                      -       C->Y  R     NOR3X6         3 15.8   115    99     528    (-,-)
  g135699/Y                      -       A->Y  F     CLKINVX3       2 10.8    54    52     580    (-,-)
  g44/Y                          -       B->Y  R     NOR2X2         1  5.3    68    68     649    (-,-)
  g279/Y                         -       B->Y  F     NAND2X2        3  8.2    80    76     725    (-,-)
  g276/Y                         -       B->Y  R     NAND2X1        1  2.8    47    57     781    (-,-)
  g135711/Y                      -       B->Y  F     NOR2X1         1  5.3    68    51     832    (-,-)
  g274/Y                         -       A1->Y R     AOI21X2        2  5.6    82    88     921    (-,-)
  g135685/Y                      -       A1->Y F     OAI211X1       1  1.7   101    98    1019    (-,-)
  phy_dpll_filtercount_reg[6]/D  <<<     -     F     DFFRX4         1    -     -     0    1019    (-,-)
#-------------------------------------------------------------------------------------------------------------
```

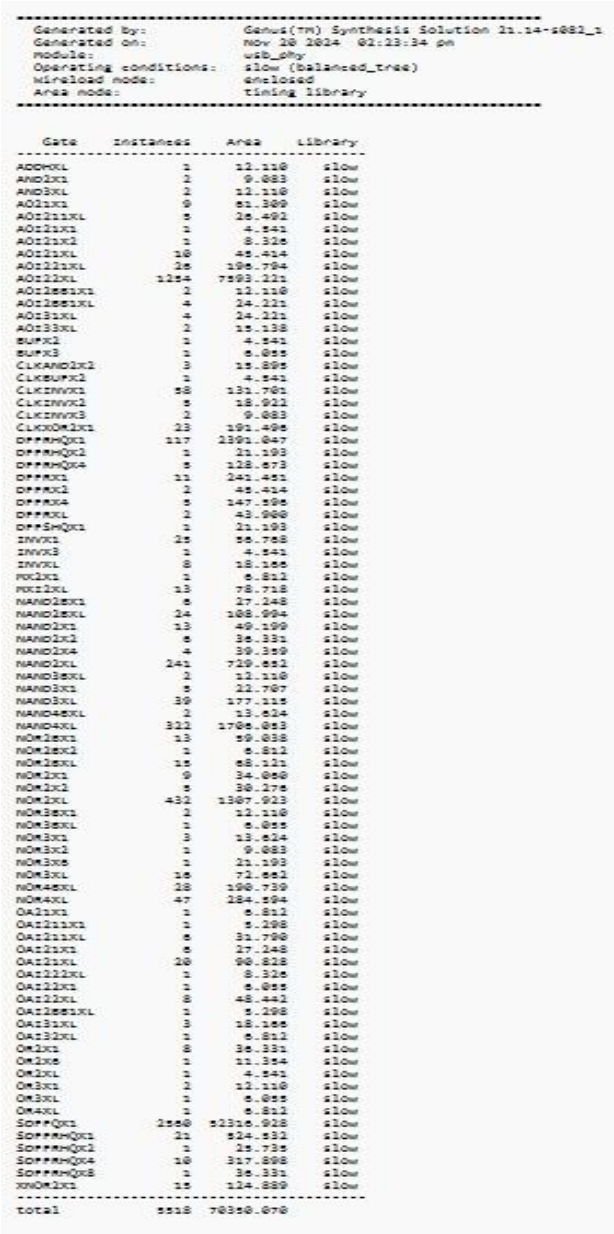
Fig. 27: Timing report

4. Gate Count Report:

The gate count report offers an estimate of the complexity of the design by calculating the equivalent number of basic logic gates (e.g., NAND gates). This metric is commonly used to compare the complexity of different designs and assess whether the implementation

aligns with resource availability. The report also aids in predicting manufacturing costs for ASIC designs, as gate count correlates with chip size and fabrication effort.

- Total Gate Count (Cell Instances): 5,518
    - Sequential Instances (Flip-Flops): 2,737
    - Combinational Instances: 2,781
- Average Fanout: 3.6



```
------------------------------------------------------------
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Nov 28 2024  02:23:34 pm
  Module:                 usb_phy
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
------------------------------------------------------------

   Gate    Instances    Area    Library
-----------------------------------------
ADDHXL             1    12.110    slow
AND2X1             2     9.083    slow
AND3XL             2    12.110    slow
AOI21X1            9    51.309    slow
AOI211XL           5    26.492    slow
AOI21X1            1     4.541    slow
AOI21X2            1     8.326    slow
AOI21XL           10    45.414    slow
AOI221XL          25   196.794    slow
AOI22XL         1184  7593.221    slow
AOI2BB1X1          2    12.110    slow
AOI2BB1XL          4    24.221    slow
AOI31XL            4    24.221    slow
AOI33XL            2    15.138    slow
BUFX2              1     4.541    slow
BUFX3              1     6.055    slow
CLKAND2X2          3    15.895    slow
CLKBUFX2           1     4.541    slow
CLKINVX1          58   131.701    slow
CLKINVX2           5    18.921    slow
CLKINVX3           2     9.083    slow
CLKXOR2X1         23   191.496    slow
DFFRHQX1         117  2391.047    slow
DFFRHQX2           1    21.193    slow
DFFRHQX4           5   128.673    slow
DFFSX1            11   241.451    slow
DFFSX2             2    45.414    slow
DFFSX4             5   147.595    slow
DFFXL              2    43.900    slow
DFFSHQX1           1    21.193    slow
INVX1             25    56.768    slow
INVX3              1     4.541    slow
INVXL              8    18.166    slow
MX2X1              1     6.811    slow
MX2XL             13    78.718    slow
NAND2BX1           6    27.248    slow
NAND2BXL          24   108.994    slow
NAND2X1           13    49.199    slow
NAND2X2            6    36.331    slow
NAND2X4            4    39.359    slow
NAND2XL          241   729.651    slow
NAND3BXL           2    12.110    slow
NAND3X1            5    22.707    slow
NAND3XL           39   177.115    slow
NAND4BXL           1    13.624    slow
NAND4XL          312  1708.053    slow
NOR2BX1           13    59.038    slow
NOR2BX2            1     6.811    slow
NOR2BXL           15    68.121    slow
NOR2X1             9    34.060    slow
NOR2X2             5    30.276    slow
NOR2XL           432  1307.923    slow
NOR3BXL            2    12.110    slow
NOR3BXL            1     6.055    slow
NOR3X1             3    13.624    slow
NOR3X2             1     9.083    slow
NOR3X6             1    21.193    slow
NOR3XL            16    72.661    slow
NOR4BXL           28   190.739    slow
NOR4XL            47   284.594    slow
OA21X1             1     6.811    slow
OA211X1            1     5.298    slow
OA211XL            6    31.790    slow
OA21X1             6    27.248    slow
OA21XL            20    90.828    slow
OA212XL            1     8.326    slow
OA22X1             1     6.055    slow
OA22XL             8    48.441    slow
OA2BB1XL           1     5.298    slow
OA31XL             3    18.166    slow
OA32XL             1     6.811    slow
OR2X1              8    36.331    slow
OR2X6              1    11.354    slow
OR2XL              1     4.541    slow
OR3X1              2    12.110    slow
OR3XL              1     6.055    slow
OR4XL              1     6.811    slow
SDFFQX1         2560  51316.928    slow
SDFFRHQX1         21    514.931    slow
SDFFRHQX2          1    15.735    slow
SDFFRHQX4         10    317.898    slow
SDFFRHQX8          1    36.331    slow
XNOR2X1           15   124.889    slow
-----------------------------------------
total           5518  70350.070
```

Fig. 28: Gate count report

5. QoR Report:

The Quality of Results (QoR) report is a comprehensive summary of key metrics from synthesis, including area, power, timing, and gate count. It provides an overall view of how well the design meets the specified objectives and constraints. This report is valuable for tracking improvements across design iterations and ensuring that the synthesis meets the project's goals for efficiency and performance.

- Cell Area: 70,350.070 μm²
- Sequential Instance Count: 2,737
- Combinational Instance Count: 2,781

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Nov 20 2024  02:23:34 pm
  Module:                 usb_phy
  Technology libraries:   slow
                          slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Timing
--------

 Clock  Period
---------------
PHY_CLK  200.0


  Cost    Critical             Violating
 Group   Path Slack     TNS       Paths
-------------------------------------------
default    No paths        0.0
PHY_CLK      -964.3   -19475.6         28
-------------------------------------------
Total                 -19475.6         28

Instance Count
--------------
Leaf Instance Count             5518
Physical Instance count            0
Sequential Instance Count       2737
Combinational Instance Count    2781
Hierarchical Instance Count        1

Area
----
Cell Area                          70350.070
Physical Cell Area                 0.000
Total Cell Area (Cell+Physical)    70350.070
Net Area                           0.000
Total Area (Cell+Physical+Net)     70350.070

Max Fanout                         2608 (DIV_CLK2)
Min Fanout                         0 (PHY_RST)
Average Fanout                     3.6
Terms to net ratio                 4.5766
Terms to instance ratio            4.6098
Runtime                            81.34637899999913 seconds
Elapsed Runtime                    98 seconds
Genus peak memory usage            1772.73
Innovus peak memory usage          no_value
Hostname                           localhost
```

Fig. 29: QOR report

APPLICATION, ADVANTAGES & LIMITATIONS

APPLICATIONS

1. Storage Devices:
    - Used in external hard drives, SSDs, and USB flash drives for faster data transfer and backup solutions.
2. Multimedia Devices:
    - Ideal for connecting high-resolution cameras, 4K/8K monitors, and video-capture devices due to its high bandwidth.
3. Gaming Accessories:
    - Popular in gaming for high-speed input/output peripherals like controllers, VR headsets, gaming mice, and keyboards.
4. Data Centers and Servers:
    - USB 3.0 is used in data centers for rapid data transfers and connections to external drives for backup and disaster recovery purposes.
5. Smartphones and Tablets:
    - Used for faster charging and seamless data synchronization with PCs and laptops.
6. Medical Equipment:
    - Supports diagnostic and imaging devices requiring fast data transfers, such as MRI and CT scanners.
7. High-Speed Networking:
    - External USB 3.0-based network adapters enable faster internet connectivity for devices without built-in Ethernet ports.
8. Industrial Applications:
    - Used in automated systems, robotics, and data acquisition systems due to its robust power supply and fast communication capabilities.

ADVANTAGES

1. High Data Transfer Speeds:
    - USB 3.0 offers a theoretical maximum data transfer speed of 5 Gbps (SuperSpeed), significantly faster than USB 2.0 (480 Mbps). This is ideal for applications that require fast data movement, such as video editing or large data backups.
2. Full Duplex Communication:
    - Unlike USB 2.0, which supports half-duplex communication, USB 3.0 supports simultaneous sending and receiving of data, increasing efficiency during data transfers.
3. Backward Compatibility:
    - USB 3.0 is backward compatible with USB 2.0 and USB 1.1 devices, allowing older devices to work with USB 3.0 ports (although at their respective speeds).
4. Increased Power Efficiency:
    - USB 3.0 supports a power delivery of up to 900 mA (compared to 500 mA in USB 2.0), enabling faster charging for connected devices and powering more energy-demanding peripherals.
5. Improved Power Management:
    - USB 3.0 devices use an efficient power management system, reducing power consumption during idle states.
6. Support for Advanced Peripherals:
    - With its higher bandwidth, USB 3.0 can support modern peripherals like external SSDs, 4K webcams, and high-resolution monitors with minimal bottlenecks.
7. Enhanced Streaming Performance:
    - Ideal for streaming applications, USB 3.0 reduces latency and ensures smoother performance for audio and video streaming.
8. Improved Signal Integrity:
    - USB 3.0 employs enhanced encoding techniques and error-checking mechanisms to reduce data loss and improve overall signal quality.

LIMITATIONS

1. Cable Length Limitations:
    - USB 3.0 cables have a maximum effective length of about 3 meters. Beyond this, signal degradation occurs, limiting its use for longer-distance connections.
2. Cost:
    - USB 3.0 devices, hubs, and cables are more expensive compared to USB 2.0 due to the complexity of the technology and higher manufacturing costs.
3. Compatibility with Older Systems:
    - Although USB 3.0 is backward compatible, older USB 2.0 devices operate at reduced speeds when connected to USB 3.0 ports, potentially bottlenecking performance.
4. Increased Power Consumption:
    - USB 3.0 peripherals generally consume more power than USB 2.0 devices, which could be a concern for battery-powered systems like laptops and tablets.
5. Interference Issues:
    - USB 3.0 operates in the 2.4 GHz frequency range, which may cause interference with Wi-Fi and Bluetooth signals if not properly shielded.
6. Connector Wear and Tear:
    - Frequent plugging and unplugging of USB 3.0 connectors can lead to wear and mechanical failure over time, as connectors are not designed for unlimited cycles.
7. Limited Bandwidth Utilization:
    - While USB 3.0 offers a theoretical maximum speed of 5 Gbps, real-world performance is often lower due to hardware and software inefficiencies.
8. Host Dependency:
    - USB 3.0 relies on the host system's performance and hardware capabilities. If the host system is outdated or lacks proper drivers, it can reduce the advantages of USB 3.0.

CONCLUSION AND FUTURE SCOPE

CONCLUSION

We have successfully completed our goals of designing and synthesis the Physical layer module of USB 3.0. The design and synthesis of the physical layer module for USB 3.0 represent a critical step in achieving high-speed data transmission with improved efficiency and reliability. The physical layer is responsible for enabling communication between connected devices while adhering to the USB 3.0 protocol's requirements for data rates of up to 5 Gbps. This module incorporates key functionalities such as parallel-to-serial conversion, clock synchronization, and polarity inversion to ensure seamless data flow across devices. By carefully designing the module with optimal resource utilization, signal integrity, and compatibility, it successfully addresses challenges like high-frequency signal handling and backward compatibility with USB 2.0.

The synthesized design demonstrates the feasibility of implementing USB 3.0's advanced features, including bi-directional data transfer and enhanced power management. Testing and simulation of the physical layer ensure its robustness in real-world scenarios, such as high-speed data transfer for external storage devices and multimedia applications. The development of this module contributes significantly to enabling USB 3.0's widespread adoption in various fields, including consumer electronics, data centers, and gaming. By meeting stringent design requirements, the physical layer module paves the way for scalable, efficient, and high-performance USB 3.0 systems.

FUTURE SCOPE

1. Integration with Emerging Technologies:
    - USB 3.0 PHY modules can be optimized for integration with newer communication standards like USB 3.2 or USB4, enabling even higher speeds and compatibility with Thunderbolt.
2. Enhanced Power Efficiency:
    - Further research can focus on reducing power consumption in the PHY module, making it more suitable for portable and IoT devices.
3. Support for Ultra-High-Definition Media:
    - With the growing demand for 8K video streaming, the PHY design can be enhanced for seamless data transfer in multimedia applications.
4. Miniaturization:
    - Developing more compact and efficient PHY designs can cater to smaller form-factor devices, such as wearables and compact laptops.
5. Improved Signal Integrity:
    - Future designs can address challenges like EMI (Electromagnetic Interference) and enhance signal integrity for longer cable lengths and robust data transmission.
6. Customization for Industry-Specific Applications:
    - Tailoring USB 3.0 PHY modules for industries like automotive (e.g., autonomous vehicles), healthcare (e.g., medical imaging devices), and manufacturing (e.g., high-speed robotic systems).
7. Advanced Error Handling:
    - Enhancing the PHY module to include intelligent error correction techniques and adaptive equalization to maintain performance under noisy conditions.
8. Compatibility with Future USB Standards:
    - Ensuring backward compatibility with USB 2.0 while being adaptable to higher-speed standards like USB 3.2 and USB4.

Through these future directions we can aim to enhance the adaptability, performance, and sustainability of USB 3.0 physical layer modules in line with evolving technological demands.

REFERENCES

[1] "Implementation of USB 3.0 SuperSpeed Physical Layer using Verilog HDL," *Int. J. Comput. Appl.*, vol. 95, no. 24, pp. 1–6, Jun. 2014.

[2] D. D. Bekal and G. Hegde, "Monitoring the Performance of USB 3.0 Protocol Using System Verilog," *Int. Res. J. Eng. Technol.*, vol. 7, no. 3, pp. 1–7, Mar. 2020.

[3] D. Anderson, "Introduction to USB 3.0," in *MindShare, Inc.*, 2013.

[4] C. Bhulania, V. Sharma, and R. Mehra, "Optimization of Physical Layer Modules of USB 3.0 Using FPGA," in *Fourth Int. Conf. Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2020, pp. 189–194.

[5] S. Mehta, "Design synthesizable USB 3.0 using Verilog HDL and simulate design using Cadence," M.S. thesis, Dept. of Electrical Eng., California State University, 2012.

[6] USB-IF, "USB 3.0 Specification," USB Implementers Forum, 2008. [Online]. Available: https://www.usb.org.

[7] C. Bhulania, V. Sharma, and R. Mehra, "Optimization of Physical Layer Modules of USB 3.0 Using FPGA," unpublished.

[8] M. Tomajoli, M. Garzetti, G. Corna, et al., "USB 3.0 High-Speed Digital Readout System with Dynamic Frame Rate Processing for ISFET Lab-on-Chip Platforms," in *IEEE Conf. Lab-on-Chip Systems*, Paris, France, 2023, pp. 22–28.

[9] M. Tomajoli, M. Garzetti, G. Corna, et al., "USB 3.0 High-Transfer Rate Time-Tagging Module for High-Performance FPGA-Based Time-to-Digital Converter," in *IEEE Nucl. Sci. Symp. and Med. Imaging Conf.*, Milan, Italy, 2022, pp. 1–5.