Verification of Receiving Equalizer Block for Mixed Signal IP

A Major Project

Submitted in Partial Fulfillment of the Requirements for the degree of

MASTER OF TECHNOLOGY

IN

VLSI DESIGN

By

Param Solanki

21MECV18

Department of Electronics & Communication Engineering

Institute of Technology, Nirma University

Ahmedabad - 382 481

May 2023

Verification of Receiving Equalizer Block for Mixed Signal IP

A Major Project

Submitted in Partial Fulfillment of the Requirements for the degree of

MASTER OF TECHNOLOGY
IN
VLSI DESIGN

By

Param Solanki
21MECV18

Internal Guide:
Dr. Manish Patel
Nirma University

External Guide:
Ganesh B P
Intel Technology India Pvt Ltd

Department of Electronics & Communication Engineering
Institute of Technology, Nirma University
Ahmedabad - 382 481

May 2023

Certificate

This is to certify that the Major Project entitled **"Verification of Receiving - Equalizer block for mixed signal IP"** submitted by **Param Solanki (21MECV18)**, towards the partial fulfillment of the requirements for the Masters of Technology in VLSI Design, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this Project, to the best of our knowledge haven't been submitted to any other university or institution for award of any degree or diploma.

Date:
Place: Ahmedabad

Dr. Manish Patel
Internal Guide,
Assistant Professor,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Usha Mehta
Program coordinator,
Professor,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Usha Mehta
Professor and Head,
EC Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Rajesh Patel
Director,
Institute of Technology,
Nirma University, Ahmedabad.

Certificate

This is to certify that the Major Project Report entitled **"Verification of Receiving - Equalizer block for mixed signal IP"** submitted by **Param Solanki** (Roll No. **21MECV18**) as the partial fulfillment of the requirements for the award of the degree of Master of Technology in VLSI Design, Electronics and Communication Engineering, Institute of Technology, Nirma University is the record of work carried out by him under my supervision and guidance. The work submitted in our opinion has reached a level required for being accepted for the examination.

Date:

Ganesh B P

Engineering Guide

Intel Corporation

Statement of Originality

I, **Param Solanki**, Roll No: **21mecv18**, give undertaking that the M.Tech thesis entitled "**Verification of Receiving - Equalizer block for mixed signal IP**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Electronics & Communication Engineering (VLSI Design)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and Igive assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student

Date:

Place: Ahmedabad

Endorsed by

Dr. Manish Patel

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr. Manish Patel**, Assistant Professor, EC Department, Institute of Technology, Nirma University for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me a great pleasure to thank **Dr. Usha Mehta**, PG Coordinator of VLSI Design, EC Department, Institute of Technology, Nirma university for his guidance throughout whole M.Tech and in this project thesis.

It gives me an immense pleasure to thank **Dr. Usha Mehta**, Hon'ble Head of EC Department, Institute of Technology, Nirma University for his kind support and providing basic infrastructure and healthy research environment.

I would also like to thank my Intel colleague **Ganesh B P** (SOC design engineer), **Kanika Nayak** (SOC design engineer) and **Pareppady Shishira** (Engineering Manager - SOC design engineer) for their continuous support and guidance throughout my M.Tech thesis projectat Intel Corporation.

I would also thank the Institution, all faculty members of Electronics and Communication Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

Param Solanki

21MECV18

Abstract

The Universal Serial Bus is a ubiquitous wired interconnect and has been around for almost two decades. The latest revision of the USB specification has introduced Super-Speed-Plus mode, double the rate of the existing SuperSpeed mode. Operation at this speed poses quite a few challenges in the design of the PHY layer, both in electrical and in logical sub-blocks.

USB-C is the emerging standard for charging and transferring data. Right now, it's included in devices like the newest laptops, phones, and tablets and—given time—it'll spread to pretty much everything that currently uses the older, larger USB connector. UVM-based HVL verification environment is used for the mixed signal verification of Type-C USB PHY. Verification environment is divided into sub-level verification environments for USB 2.0 and USB 3.0 PHYs.

The UVM-based HVL verification environment of the Type-C USB PHY used for digital verification has been effectively reused for the mixed-signal verification. The scoreboard and protocol checkers are enabled in mixed-signal verification. Additional tests and functional coverage were added to the existing digital environment for robust verification of the RX-EQ of mixed signal IP. The overall functional coverage is improved by merging digital regression data with mixed-signal regression data. Thus, this mixed-signal verification methodology improves the verification quality and confidence in the design.

Although constrained-random stimulus generation produces many tests very quickly, results checking is needed to ensure that the design executes each test properly. Results checking can be subdivided into data checking and protocol checking. Data checking relies on the ability of the testbench to account for variability in the delay and/or order in which results come out of the design being tested. This variability is critical for covering all possible scenarios.

CONTENTS

List of Figures

Chapter 1

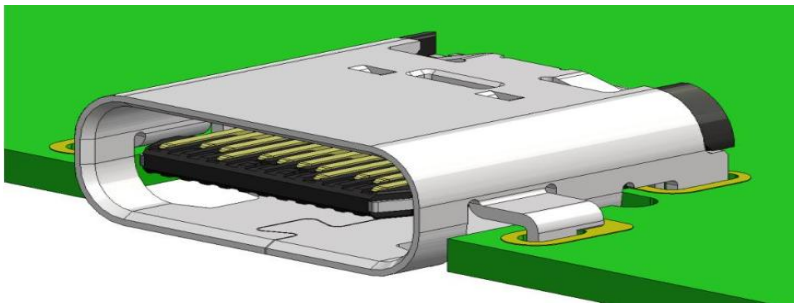
Introduction

Verification in pre-silicon stage is considered as an important concern to avoid chip defects due to design errors, specification errors. In today's SoC chips where there are several modules such as processor, memories, caches, Bluetooth, GPS, etc. in a single chip, verification plays a major role in verifying the design against the desired specification before tape-out. Any bugs in design not identified before tape-out will be a serious issue, hence verification is considered as a critical process in design cycle. Apart from verifying the functional correctness of the design, other aspects such as security, safety, performance, power targets are also verified during Verification process. Verification is in front end part of ASIC design flow. RTL design consumes about 10 – 20% of the time in entire design cycle while Verification takes around 80 – 90% of the time. The functionality of the modules is designed using HDL languages like VHDL, Verilog or System Verilog. Verification Plan is developed by the Verification Team.

1.1 USB – C:

USB-C (properly known as USB Type-C) is a 24-pin USB connector system with a rotationally symmetrical connector. The designation **C** refers only to the connector's physical configuration or form factor and should not be confused with the connector's specific capabilities, which are designated by its transfer specifications (such as USB 3.2). A notable feature of the USB-C connector is its reversibility; a plug may be inserted into a receptacle in either orientation.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
GND	TX1+	TX1−	VBUS	CC1	D+	D−	SBU1	VBUS	RX2−	RX2+	GND
GND	RX1+	RX1−	VBUS	SBU2	D−	D+	CC2	VBUS	TX2−	TX2+	GND
B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1



Source: PHY Interface for the PCIe, SATA and USB 3.0 Architectures, Version 4.0

Figure 1.1: Type-C

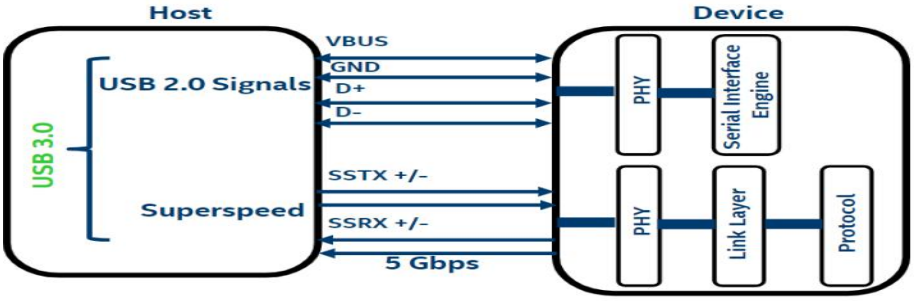
Unlike existing USB Type - A and USB Type - B receptacles and plugs, the mechanical characteristics of the USB Type - C receptacle and plug do not inherently establish the relationship of USB host and device ports. The CC pins on the receptacle also serve to establish an initial power (Source -To - Sink) and data (Host to Device) relationships prior to the normal USB enumeration process. For the purpose of defining how the CC pins are used to establish the initial power relationship, the following port power behavior modes are defined.

- Source - only – for this mode, the port exclusively behaves as a Source
- Sink-only – for this mode, the port exclusively behaves as a Sink
- Dual-Role-Power (DRP) –for this mode, the port can behave either as a Source or Sink Additionally

when a port supports USB data operation, a port's data behavior modes are defined.

- DFP only–for this mode, the port exclusively behaves as a DFP
- UFP-only– for this mode, the port exclusively behaves as a UFP
- Dual-Role-Data (DRD) – for this mode, the port can behave either as a DFP or UFP

1.1.1 USB 3.0:



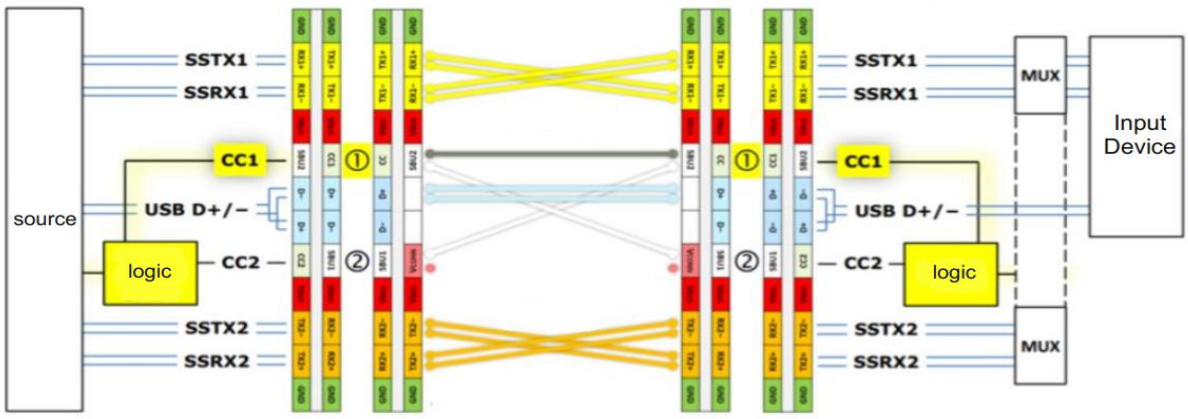
Source: PHY Interface for the PCIe, SATA and USB 3.0 Architectures, Version 4.0

Figure 1.2 USB3.0

- USB 3.0 PHY includes
  - 8b/10b encoding/ decoding,
  - Data scrambling and descrambling,
  - Serializing and De Serializing functions
  - 5Gbps A round number
- Link layer maintains link connectivity
  - Data integrity by implementing error detection
  - Packets are created in the link layer with link commands
- Protocol layer manages end to end data flow
  - Four packet types are defined – Two of them,
  - Transaction (TP) and Data Packet (DP) same as USB 2.0
  - Two additional packet types of Isochronous Timestamp
  - Packet (ITP) and Link Management Packet (LMP) are
  - newly introduced by USB 3.0.

1.1.2. USB 3.2:

USB 3.2, released in September 2017, replaces the USB 3.1 standard. It preserves existing USB 3.1 SuperSpeed and SuperSpeed+ data modes and introduces two new SuperSpeed+ transfer modes over the USB-C connector using two-lane operation, with data rates of 10 and 20 Gbit/s (~1.2 and 2.5 GB/s).



Source: PHY Interface for the PCIe, SATA and USB 3.0 Architectures, Version 4.0

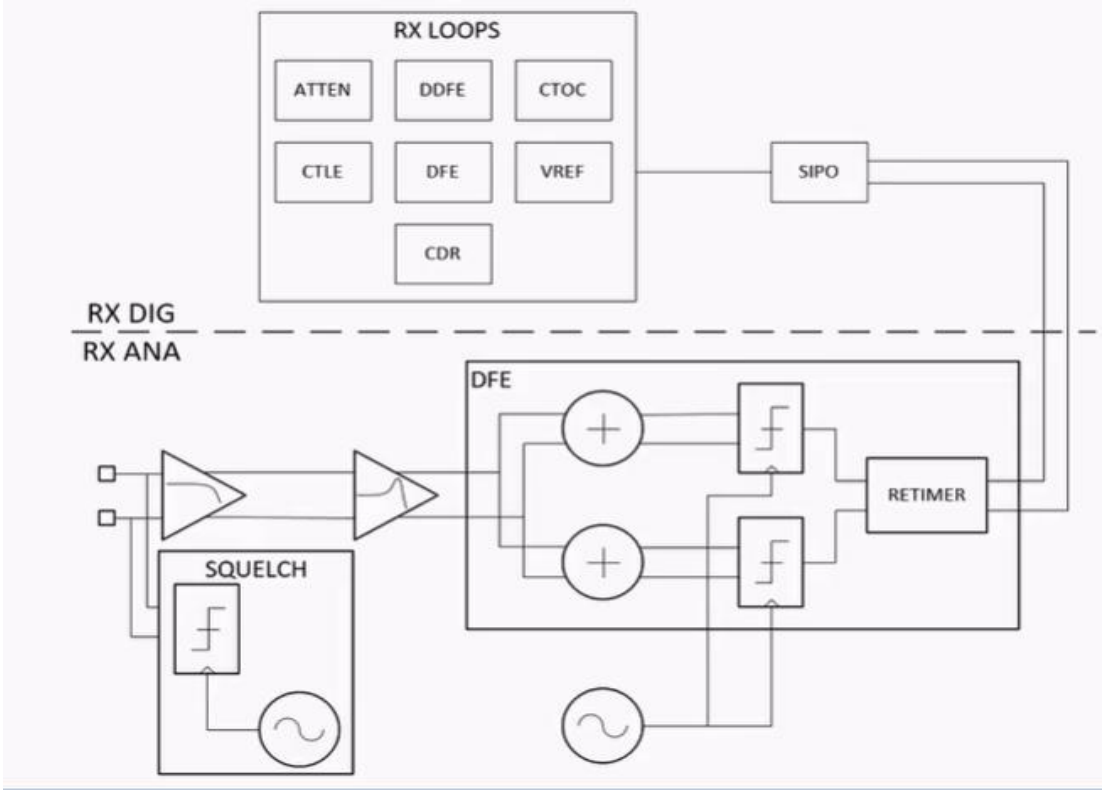
Figure 1.3 Architecture of PHY USB

Chapter 2

Receiving - Equalizer architecture

In this chapter, we will understand the concept of RX-EQ block in USB-C and also, we will understand each sub-block. Adaption flow will explain by this chapter that how actually data flows.

2.1. Receiving - Equalizer Block diagram



Source: PHY Interface for the PCIe, SATA and USB 3.0 Architectures, Version 4.0

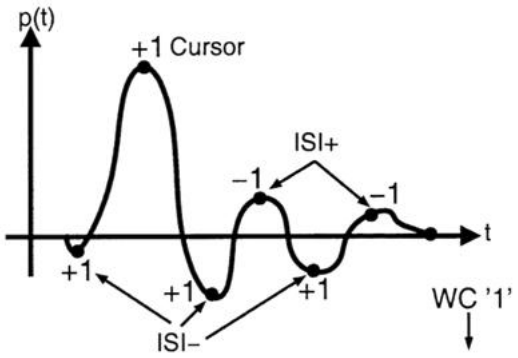
Figure 2.1 RX-EQ Architecture

The main goal of the receiver is to, as the name implies, Receive the data. As speeds get

higher and channels longer, the ability to easily send data from one chip to another becomes more difficult. Things such as inter-symbol interference (ISI), crosstalk, jitter, noise, etc. Make it harder to distinguish a digital 1 from a 0. The receiver's job is to cancel out or compensate for as many of these effects as possible to receive the correct data.

2.1.1. INTER-SYMBOL INTERFERENCE:

If the TX was to send a single pulse across the channel, the signal received by the RX would not be as clean. Due to non-idealities in the channel, the signal would look distorted. There would even be portions of the signal that arrive at different times. The portions of the signal that arrive more than 1 UI away from the initial pulse will influence the other data bits. The affect that this has on future data bits are referred to as inter symbol interference.

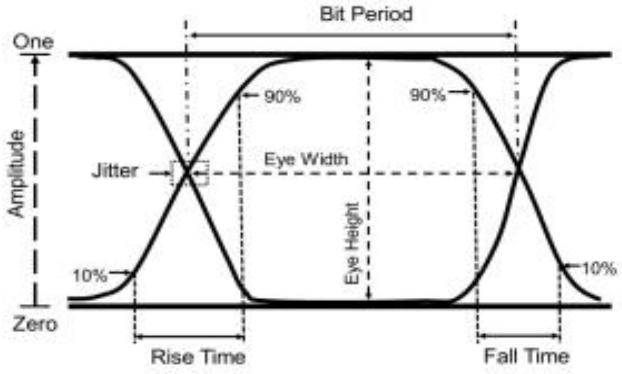


Source: www.onsemi.com

Figure 2.2 Inter symbol Interference

2.1.2. EYE DIAGRAM:

An eye diagram is used to measure how well the RX can tell the difference between a 1 and a 0. It is re-constructed, usually with a DFX hook, by measuring the analog signal during many data transitions. The "height" of the eye will tell how easy it would be for the RX to tell the difference between a 1 and a 0. The "width" of the eye will tell how much error in the sampling position the RX could tolerate before misinterpreting, or even slipping, a data bit. The RX tries to make the eye as "open" as possible.



Source: www.onsemi.com

Figure 2.3 Eye Diagram

2.1.3. Cross talk:

crosstalk refers to the unwanted coupling of signals between adjacent wires or components on an integrated circuit (IC). It is a significant issue in VLSI design that can lead to signal integrity problems and performance degradation.

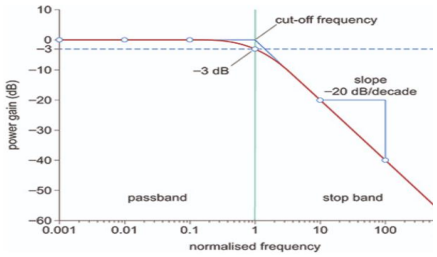
When multiple wires or components are placed closely together on an IC, they can interact with each other through parasitic capacitance and inductance. This interaction causes the signals on one wire to induce an undesired effect on neighboring wires, leading to crosstalk.

Crosstalk can manifest in two forms: capacitive crosstalk and inductive crosstalk. Capacitive crosstalk occurs when the changing voltage on one wire induces an unwanted voltage on an adjacent wire through the parasitic capacitance between them. Inductive crosstalk, on the other hand, happens when the changing current on one wire induces a voltage on an adjacent wire due to the parasitic inductance between them.

2.2. Attenuator:

In USB communication, an equalizer is commonly used to counteract signal distortions that arise from transmitting data over a physical medium, like a cable. The equalizer's purpose is to restore the signal's integrity by mitigating the effects of inter-symbol interference (ISI) and other forms of distortion. Also referred to as an AGC (automatic gain control) ckt. Changes the gain in the RX data path to change the size of the incoming signal. Used to increase the signal when it is too small due to long channel. Also used to decrease

the signal when it is large due to short channel.



Source: www.onsemi.com

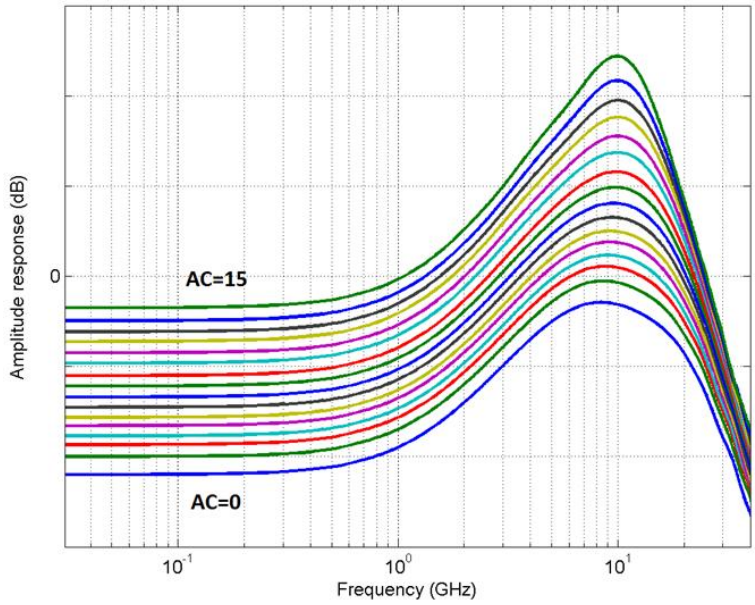
Figure 2.4 Attenuation graph

2.3. CTLE:

In the USB protocol, CTLE stands for Continuous-Time Linear Equalization. It is a key component found in the receiver side of USB interfaces, specifically in USB 3.0 and USB 3.1 standards. The purpose of the CTLE block is to compensate for signal distortion and attenuate the effects of inter-symbol interference (ISI) caused by transmission over physical media like cables. Continuous Time equalizers are implemented to correct for losses and distortions caused by high frequency transmission lines. When signals are transmitted through a physical medium, such as cables or wires, they can be affected by various factors that degrade the signal quality. One such factor is the low-pass characteristics of the medium, which means that high-frequency components of the signal tend to be attenuated or weakened.

To compensate for this effect, the receiver buffer employs equalization circuits. These circuits are programmable, meaning that they can be adjusted or configured to meet specific requirements. There are five such circuits in this receiver buffer, and each operates independently.

Creates “peaking” by increasing the gain at high frequencies. Peaking is centered around the Nyquist Frequency Used in conjunction with the DFE to help equalize or clean up the incoming signal.



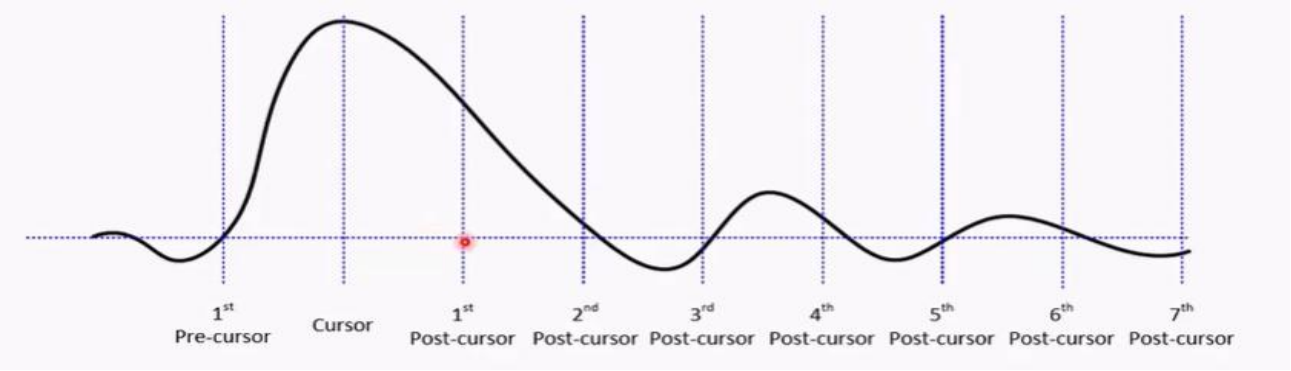
Source: www.onsemi.com
Figure 2.5 CTLE graph

2.4. DFE:

In the USB protocol, DFE stands for Decision Feedback Equalization. It is an important component found in the receiver side of USB interfaces, particularly in USB 3.0 and USB 3.1 standards. The purpose of the DFE block is to further compensate for signal distortion and mitigate the effects of inter-symbol interference (ISI) caused by transmission over physical media like cables.

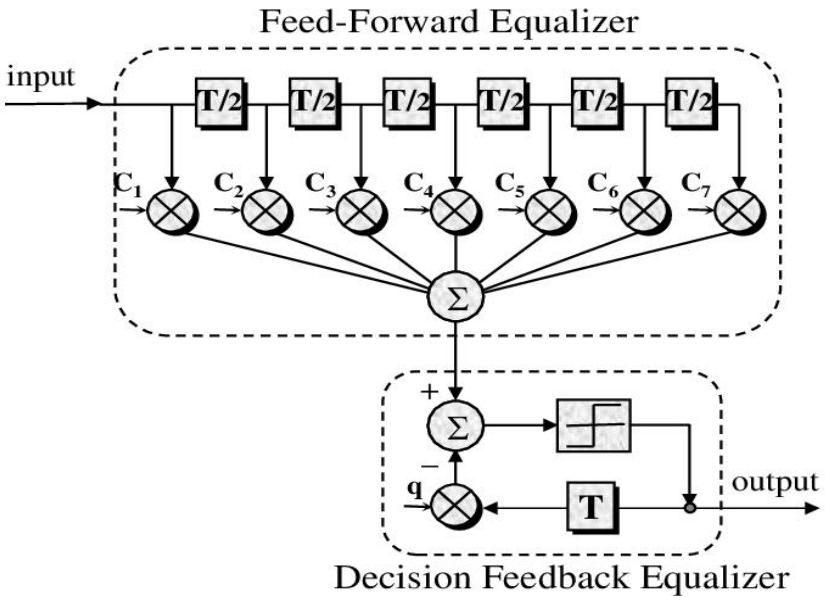
DFE operates by using feedback from previously detected symbols to improve the accuracy of symbol detection for the current symbol. It uses a tap-weighted filter that examines the received signal and makes decisions based on the sequence of symbols it has previously detected. By considering the history of detected symbols, the DFE block can effectively "look ahead" and make more precise decisions for the current symbol, compensating for ISI.

Decision feedback equalizer Focuses on canceling out the ISI at discrete points in the time after the cursor. Uses the values of previous data bits as feedback for equalizing the signal.



Source: PHY Interface for the PCIe, SATA and USB 3.0 Architectures, Version 4.0

Figure 2.6 Example of ISI signal



Source: PHY Interface for the PCIe, SATA and USB 3.0 Architectures, Version 4.0

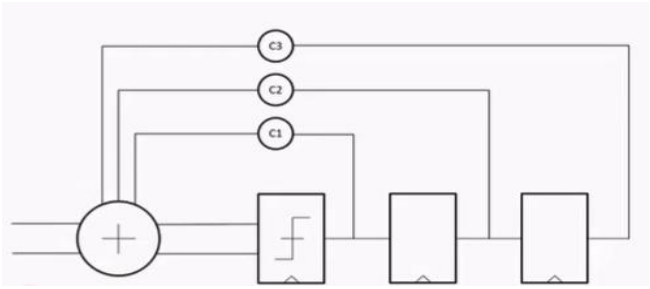
Figure 2.7 DFE block diagram

The main advantage of the decision-feedback equalizer (DFE) over linear equalizers is its ability to cancel inter-symbol interference (ISI) without amplifying the noise. ISI refers to the interference caused by neighboring symbols that overlap with the current symbol in a communication system. The DFE operator plays a crucial role in validating hardware DFE designs and observing their effects on eye openings. The "eye opening" refers to the clear space or region in a received signal where the decision threshold can be set to accurately determine the symbol being transmitted.The tap values of the DFE are carefully chosen to correct for the portion of the previous symbol that lingers and distorts the current symbol. This distortion is caused by the interference from adjacent symbols, resulting in a loss of signal integrity. By applying appropriate tap values, the DFE compensates for this distortion and

helps restore the original symbol waveform.To put it simply, the DFE is a specialized equalization technique that not only mitigates inter-symbol interference but also minimizes the amplification of noise. It achieves this by using taps, which are applied to normalized voltages based on the symbol slicer decision. These tap values are designed to correct any lingering distortion from previous symbols, thereby improving the accuracy and reliability of symbol detection in communication systems.

2.4.1. Summer:

Combines the feedback with the incoming data to equalize the signal. Requires a feedback data bit for each tap that is going to be equalized.Tap correspond to the 1st post cursor 2nd post cursor so on.

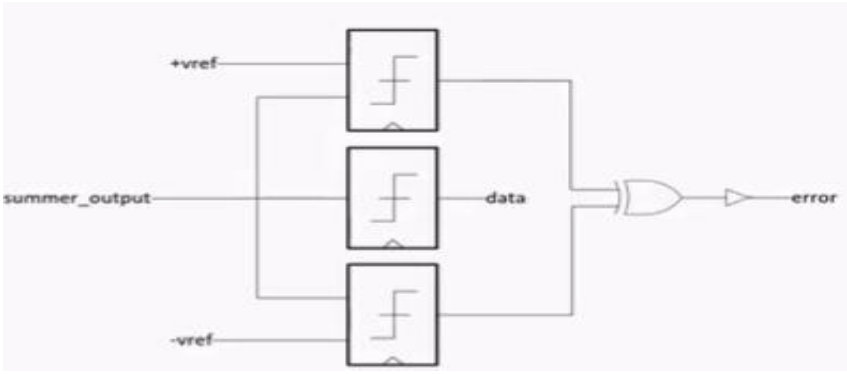


Source: PHY Interface for the PCIe, SATA and USB 3.0 Architectures, Version 4.0

Figure 2.8 Summer and sampler with tap

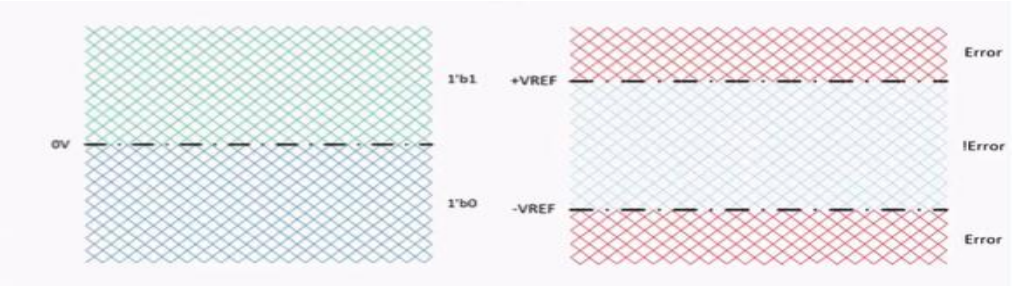
2.4.2. Samplers:

Analog FFs that are used to covert the incoming analog voltage into digital bit. Comperes the incoming signals to the reference voltage to determine if the signal is 1 or 0.



Source: PHY Interface for the PCIe, SATA and USB 3.0 Architectures, Version 4.0

Figure 2.9 Sampler



Source: PHY Interface for the PCIe, SATA and USB 3.0 Architectures, Version 4.0

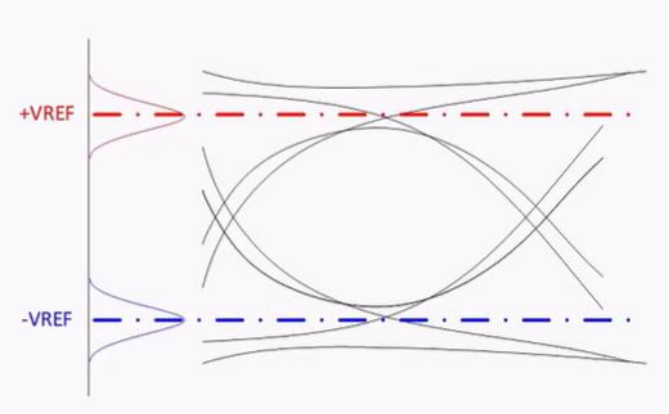
Figure 2.10 Sampler example

2.5. Vref loop:

In the USB protocol, the "voltage reference loop" refers to a circuit or subsystem that provides a stable and accurate voltage reference for various components within the USB system. It is not a specific block or feature defined by the USB specification but rather a general concept related to maintaining consistent voltage levels.

A voltage reference is crucial in many electronic systems, including USB, as it serves as a baseline against which other voltages are measured and regulated. The voltage reference loop ensures that the reference voltage remains constant and within the specified tolerance levels, which is essential for reliable and accurate operation of the USB interface.

A measurement is needed to see the relative size of the incoming signal. This value is used as a reference for the rest of the loops in the RX. The reference voltage is measured by moving the vRef code until it sits at the average height of all valid data transitions. An accurate measurement can be crucial to maintaining stable loops. vRef will be at the average of the 1's and 0's distribution to detect 1's and 0's easily. vRef is the basic loop that will going to use in every loop.

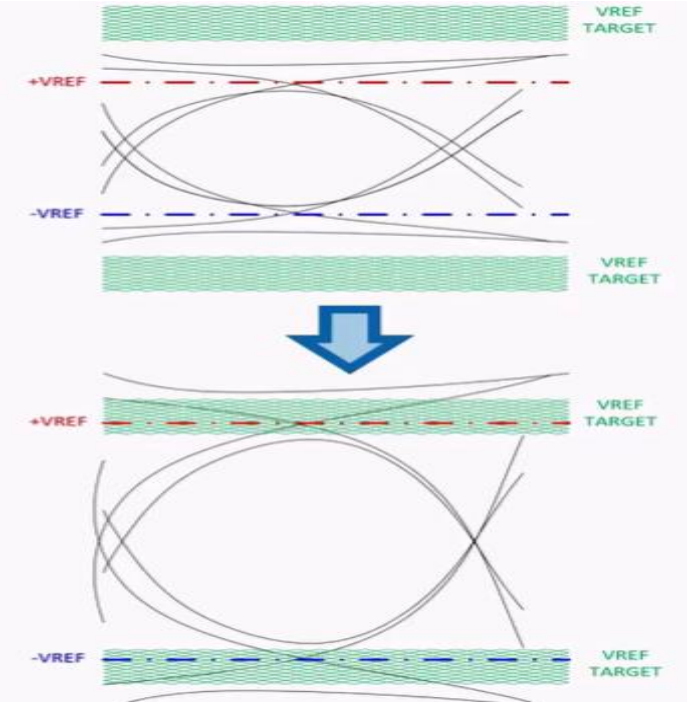


Source: PHY Interface for the PCIe, SATA and USB 3.0 Architectures, Version 4.0

Figure 2.11 EYE diagram with Vref loop

2.6. ATTEN loop:

The atten loop is used to control the gain in the attenuator circuit. The desired amount of gain is based on getting VREF to a specific target range. If Vref is too low, then the gain needs to be increased to increase the signal. If VREF is too high, then the gain would need to be decreased. The gain is updated until the average VREF code is within a certain tolerance of the target.



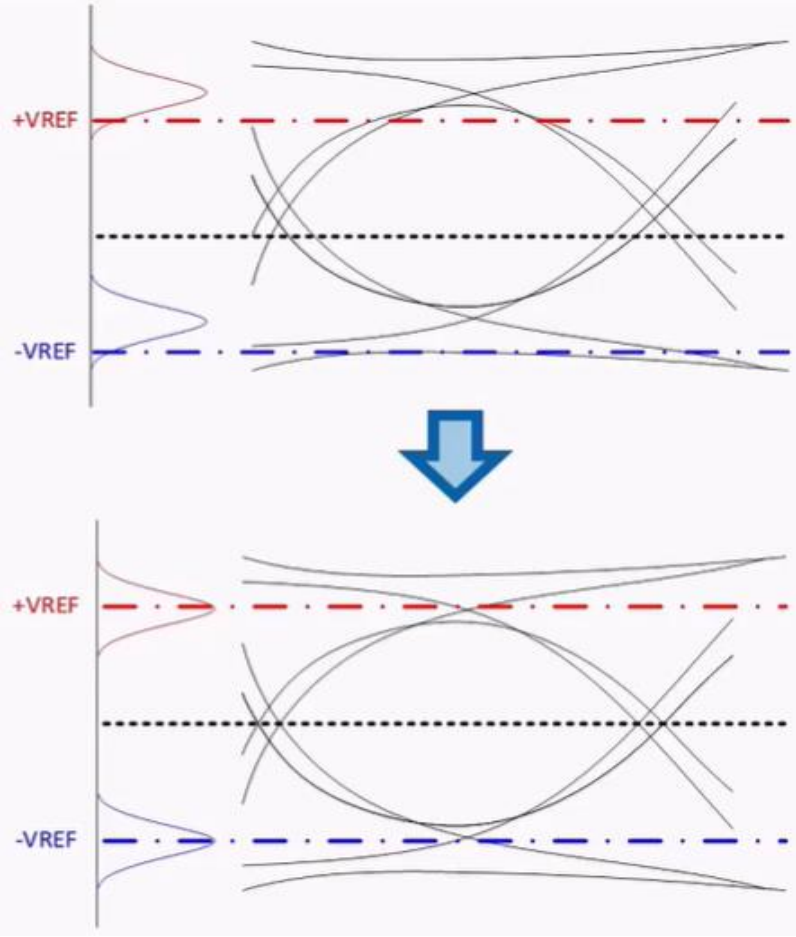
Source: PHY Interface for the PCIe, SATA and USB 3.0 Architectures, Version 4.0

Figure 2.12 Example of Attenuation loop work

2.7. CTOC loop:

In the context of the USB protocol, the continuous time offset correction (CTOC) loop is a mechanism used to compensate for timing variations and inaccuracies in the received signal. It is an important component found in the receiver side of USB interfaces, specifically in high-speed USB versions such as USB 3.0 and USB 3.1.

The CTOC loop operates by continuously monitoring and adjusting the sampling time of the received signal to align it with the optimal sampling point. Timing variations can occur due to various factors such as transmission channel characteristics, cable length, and signal distortion. These variations can result in inter-symbol interference (ISI) and cause errors in symbol detection. Continuous Time Offset Correction. Maintains the offset correction in the sampler



Source: PHY Interface for the PCIe, SATA and USB 3.0 Architectures, Version 4.0

Figure 2.13 Example of CTOC loop work

2.8. CDR loop:

Clock and data recovery. The RX only receives data from the other TX, but not the clock the TX uses to send the data. The RX needs a clock that is properly aligned to the incoming data to ensure that the samplers in the DFE are sampling at the correct time. So, the RX attempts to reconstruct the TX's clock by adjusting the phase of a local clock based on data transitions. It Will receive "early" or "late" signals from a phase detector to help adjust clock

2.9. CTLE loop:

In the same way that the ATTEN loop controls the attenuator's gain, the CTLE loop is used to control the CTLE gain. The CTLE works as a high-pass filter, which will boost high frequency signals. At higher frequency curves of the input data could be very nearer or distorted due to ISI so it would be difficult of the receiver to identify the data pattern.To sharpen that curve CTLE will give high gain at the higher frequency.

2.10. DFE loop:

The DFE attempts to remove ISI revery integer number of UI after the cursor. This means that , if there is a positive amount of interference, it will pull the curve down until it crosses 0. There is still interference after the last tap correction point. There is also still interference at non integer time points. Since the rising clock edge for the samplers occurs every UI, they should only see the signal corresponding to the cursor, and all uncorrected pre/post cursors.

2.11. DDFE loop:

The directive DFE loop operates by utilizing decision feedback equalization to mitigate the effects of inter-symbol interference (ISI) and improve the accuracy of symbol detection. It is responsible for making decisions about the received symbols based on feedback from previously detected symbols.

The DFE loop incorporates a tap-weighted filter that examines the received signal and adjusts its tap weights based on the history of previously detected symbols. By considering the sequence of previously detected symbols, the DFE loop can effectively cancel out or reduce the impact of ISI and make more accurate decisions for the current symbol.

The tap weights of the DFE loop are continuously adjusted based on the received signal characteristics and the estimated ISI. By dynamically adapting the equalization parameters, the DFE loop optimizes symbol detection and compensates for channel distortions. This helps improve the overall signal quality, increase the link's robustness, and enhance the receiver's performance.

The primary objective of the directive DFE loop in the USB protocol is to enhance the receiver's ability to accurately detect symbols in the presence of ISI. By leveraging decision feedback equalization, the directive DFE loop contributes to achieving reliable high-speed data transmission and maintaining signal integrity.

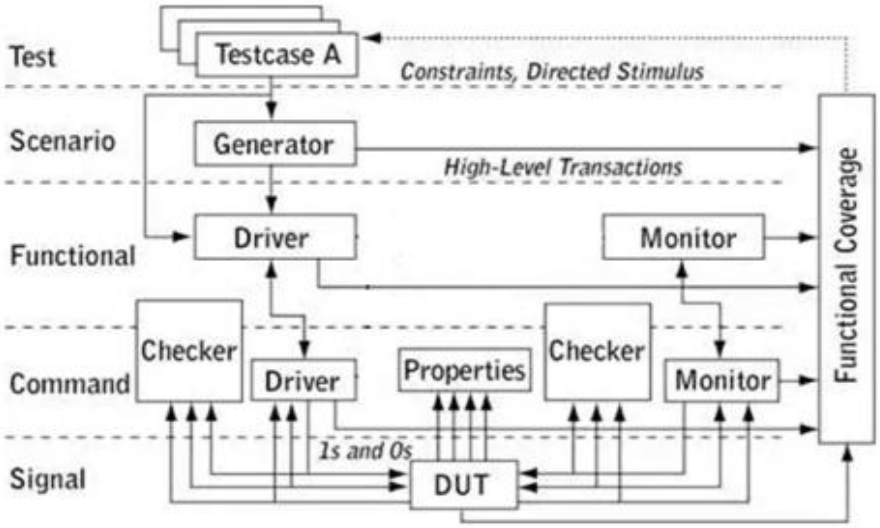
Derivative DFE. The CDR is not guaranteed to set the sampling position at the optimal location. DDFE will try to find the sampling point to sample at the correct location where eye is most open. That should increase the eye opening by finding maximum point for the lower part of the eye.

Chapter 3

Test Bench Environment and Test plan

To have a common verification environment that facilitates reuse and extension to take full advantage of automation, a layered testbench architecture is required. This approach supports both top-down and bottom-up verification within a project and makes it easier to share common components between projects.

3.1 Test Bench Environment:



Source: www.eetimes.com
Figure 3.1 Exhaustive test bench environment

The lowest layer is the signal layer that connects the testbench to the RTL design. It consists of interface, clocking, and mod port constructs. The command layer contains lower-level driver and monitor components, as well as the assertions (properties) that check design intent. This layer provides a transaction-level interface to the layer above and drives the physical pins via the signal layer. The functional layer contains higher-level driver and monitor components, as

well as the self-checking structure that determines whether tests pass or fail. Additional checking, for example protocol checkers, can span the command and functional layers.

The scenario layer uses generators to produce streams or sequences of transactions that are applied to the functional layer. The generators have a set of weights, constraints or scenarios specified by the test layer. The randomness of constrained-random testing is introduced within this layer. Finally, the test layer is where the tests are located. The tests can define new sequences of transactions using the scenario layer, synchronize multiple transaction streams, generate sequences by interacting directly with the functional or command layers, or supply directed stimulus directly to the command layer. Although this layered testbench is designed primarily for using constrained-random stimulus generation, it supports manual directed tests as well. The upper-left portion of Figure 1 shows a path running directly from the tests to the driver, bypassing the generator entirely. This allows a verification engineer to generate transactions directly without setting up constrained-random scenarios.

Although constrained-random stimulus generation produces many tests very quickly, results checking is needed to ensure that the design executes each test properly. Results checking can be subdivided into data checking and protocol checking. Data checking relies on the ability of the testbench to account for variability in the delay and/or order in which results come out of the design being tested. This variability is critical for covering all possible scenarios.

Building results-checking functionality into the testbench is often a substantial part of the difficulty in creating the testbench in the first place. System Verilog was designed with language constructs and primitives to help implement the communication between the stimulus and response checking of the testbench and help manage the expected results in such a way as to account for the variability of the possible output.

Including data coverage recording in the response checkers ensures that, for all the data combinations in the input, the appropriate output combinations were received. It also allows the verification engineer to analyze the coverage data and evaluate whether the right input stimulus combinations were generated to verify all possible output conditions.

Protocol checking typically requires the monitoring of behavior over time, establishing a temporal relation. Some high-level protocols are most naturally specified with System Verilog verification constructs and monitored within the testbench. Other implementation-specific protocol checking of design assumptions is most naturally checked by System Verilog assertions within the design and on its interfaces.

3.2 Test-Plan:

a test plan is a document that outlines the overall strategy, objectives, and scope of the testing effort. It serves as a blueprint for the verification process, guiding the development of test cases, defining test goals, and documenting the expected results.

A well-defined test plan helps ensure that the verification process is comprehensive, organized, and efficient. It typically includes the following key elements:

1. Objective and Scope: The test plan begins by clearly stating the overall objective of the verification effort. This includes defining the specific features, functionality, or aspects of the design that need to be tested. The scope of the test plan outlines the boundaries and limitations of the testing effort.
2. Test Goals and Strategies: The test plan describes the specific goals and objectives for the testing process. It outlines the strategies and methodologies to be employed, such as directed testing, random testing, or constrained random testing. The plan also includes any specific focus areas, corner cases, or critical scenarios to be tested.
3. Test Environment: The test plan specifies the simulation environment and tools to be used for testing. It outlines the setup requirements, including the testbench components, test harness, and any external interfaces or stimuli that may be necessary.
4. Test Cases: The test plan details the specific test cases to be executed. It includes a list of different scenarios, inputs, and expected outputs. Each test case should have a clear objective and expected results to determine whether the DUT is functioning correctly.
5. Coverage Analysis: The test plan outlines the coverage goals and metrics to be used for assessing the completeness of the testing process. It defines the coverage points, such as code coverage, functional coverage, or assertion coverage, and specifies the desired coverage targets.
6. Schedule and Resources: The test plan includes a timeline or schedule for executing the test cases and completing the verification process. It also outlines the required resources, such as simulation platforms, hardware resources, or human resources.
7. Reporting and Documentation: The test plan specifies the reporting and documentation requirements. It outlines how the results will be documented, including any required logs, reports, or summaries. It also defines the process for tracking and managing issues or defects found during testing.

Basically, first we read the specifications of the design properly and study about the design and know the features of the designs that need to be verified by verification engineer. According to the specification and features of the design engineer will make one HVP plan. HVP plan is nothing but **High-Level Verification** plan which will include features of the design.

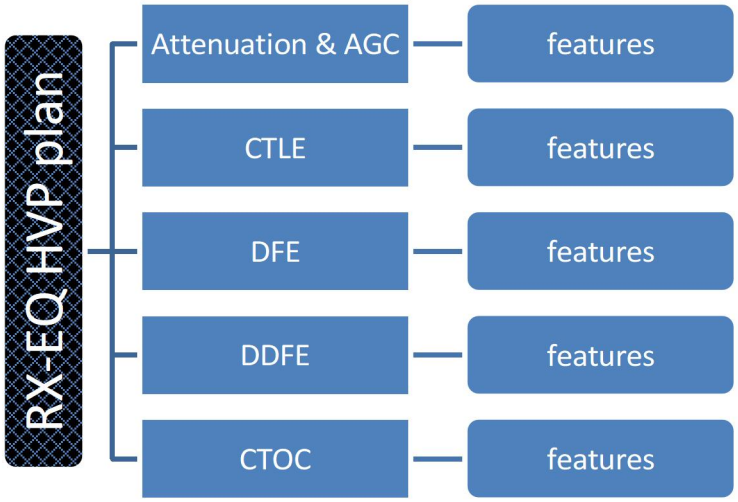


Figure 3.2 RX-EQ HVP Plan

In a validation project, Tests Plans or Test Protocols are used to demonstrate that a system meets requirements previously established in specification, design, and configuration documents. Test Plans document the general testing strategy; Test Protocols are the actual testing documents. In many cases, the Test Plan and Test Protocol are combined into a separate document.

The Test Plan outlines the testing requirements and strategy. It should include the general process for performing the testing, documenting evidence of testing and the process for handling testing failures. The Test Plan may also include the types of testing, descriptions of environments where testing will be performed, who is responsible for testing, equipment or testing that will be used in testing, or other organizational requirements for testing.

Chapter 4

Results of checker and coverage

4.1 Checker:

a checker plays a critical role in validating the behavior and correctness of the design under test (DUT). The checker acts as an independent monitoring component that examines the DUT's outputs or internal signals and compares them against expected values or properties. It helps ensure that the DUT functions correctly and adheres to the specified requirements.

checkers enhance the effectiveness and reliability of the verification process by independently monitoring and verifying the behavior of the DUT. They contribute to error detection, protocol compliance, assertions validation, coverage analysis, debugging, and functional verification, helping to identify and address potential issues early in the development cycle.

4.1.1. PAM Enable Check:

There are mainly three signals in which this checker is focused. Importance of this checker is to check whether the pam2 feature is enabled for attenuator block or not. First signal is to check that when atten fsm done is going high at that moment atten enable is going low or not. As atten enable describe that when attenuation block will work. There are two signals after that which represent that pam2 is enabled or not. So, when pam2 enable signal is enabled(high) after that only atten enable signal should go high.

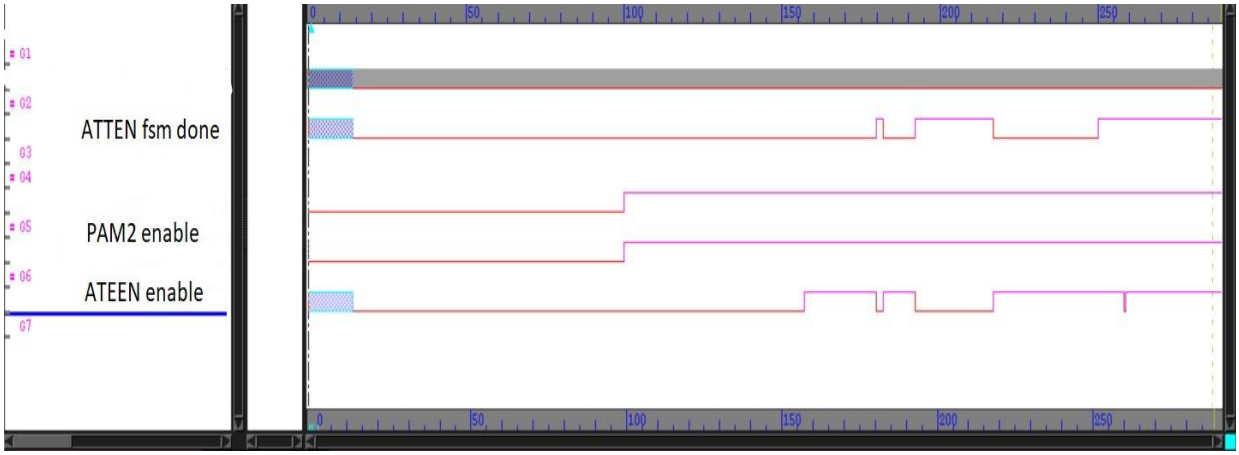


Figure 4.1 Pam2 enable check

4.1.2. Attenuation loop enable during CTLE period Check:

There are four main signals in which we must focus for this scenario. During CTLE period attenuation is again initiating attenuation loop by fsm or not that scenario we need to check.

First is Atten CTLE en signal which will indicate to fsm_eq signal to enable the attenuation during CTLE period. Aten_enable signal will notify to attenuator block to do attenuation whenever it is high it means attenuation loop is running. Fsm_eq is showing that in which fsm state we are currently. State_eq is showing that equalizer is in which state either it will be in READLUT state or RESET state. READLUT state means equalizer is ready to take inuput values and in RESET state it will remain off.

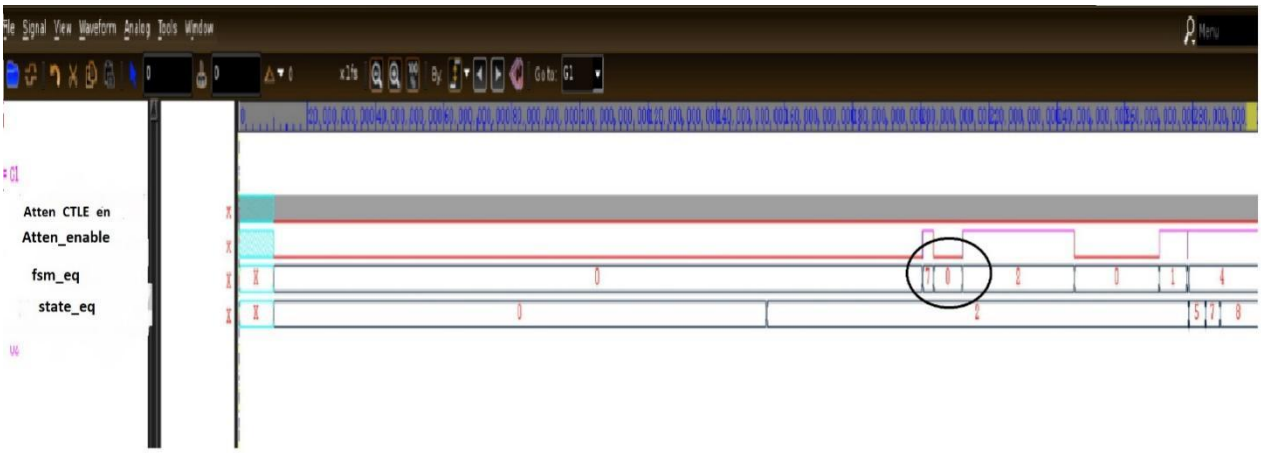


Figure 4.2 Attenuation enable during CTLE check

4.1.3. Reset eq when rate is changing without standby signal:

Mainly in two signal this checker is dependent. Reset_fsm_eq should go high when rate is changing. It means that whenever rate_change signal is changing its value reset_fsm_eq should go high to reset the fsm of equalizer so that it can start from beginning. As rate change means design is changing its operation from display to charging for example. Whenever state_eq is in READLUT state at that time only reset_fsm_eq should go high otherwise equalizer would miss the notation from that signal that fsm must restart.

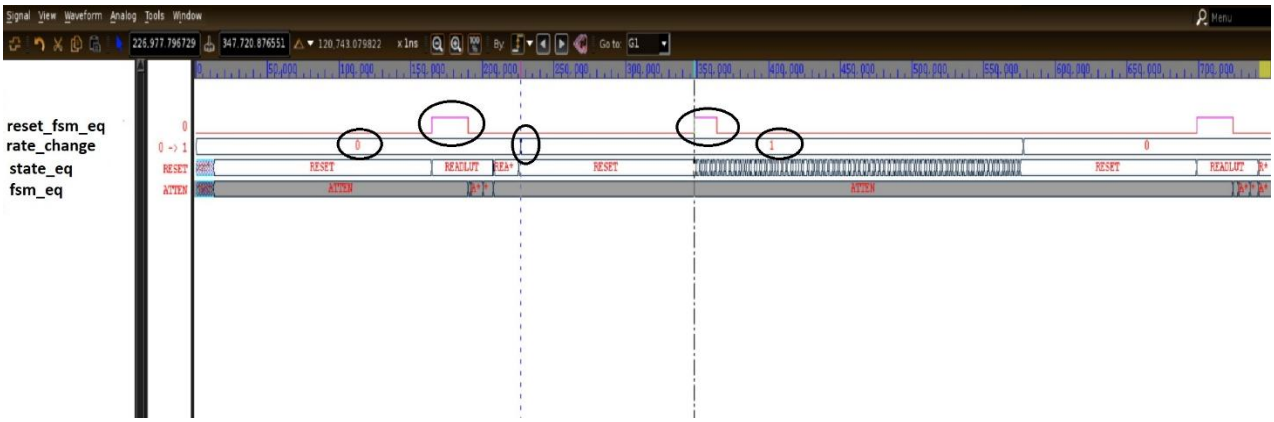


Figure 4.3 Reset eq when rate is changing without standby signal check

4.1.4 Reset eq when rate is changing with standby signal:

When I was checking for rate change checker, I faced that sometimes it could happen that Type-C is in standby mode and it is not ready to take any action when the rates are changing. So I come to know that there is one more signal call standby signal it defines that whenever that signal is high that means out type-c is in standby mode and not ready to take any action during that period.

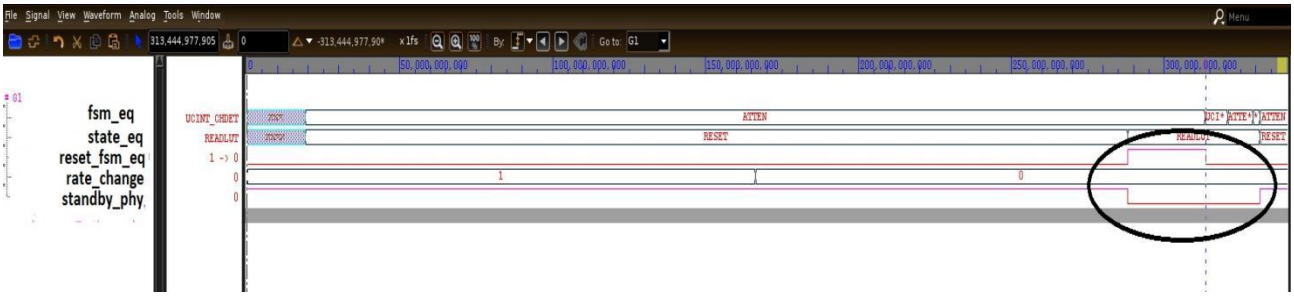


Figure 4.4 Reset eq when rate is changing with standby signal check

Figure 4.5 Rate change checker is passing

4.2 Cover group:

a covergroup is a construct used to measure and track code coverage metrics during simulation-based testing. It plays a vital role in assessing the completeness of the verification process by monitoring which parts of the design or testbench code have been exercised.

The covergroup collects coverage information based on specified coverage points, which can include conditions, branches, expressions, state transitions, or any other desired events or elements within the design. It allows verification engineers to define coverage goals and measure the progress towards achieving them.

4.2.1 Filter Select Coverage:

Code coverage is a basic coverage type which is collected automatically. It tells you how well your HDL code has been exercised by your test bench. In other words, how thoroughly the design has been executed by the simulator using the tests used in the regression. Functional coverage measures how well the functionality of the design has been covered by your test bench. In functional coverage user must define the functionality to be measured through coverage.

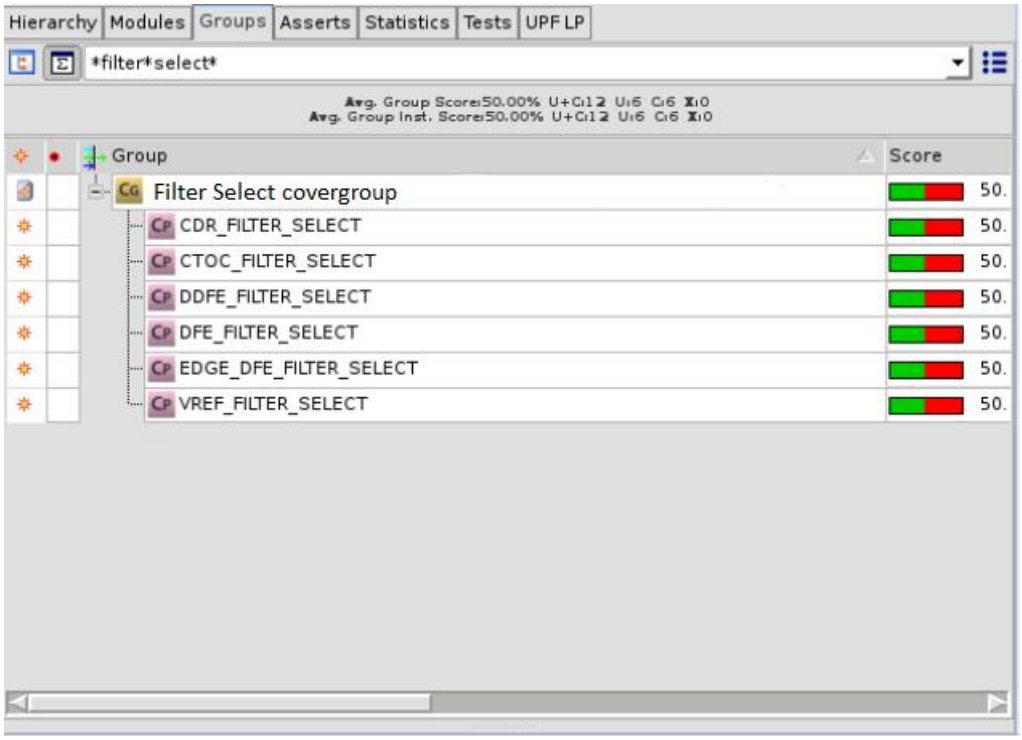


Figure 4.6 Filter select cover group

Cover group is written because to check that that portion of code is utilizing or not and whatever bin which we have created inside the cover group is covered or uncovered. Here we can see that there is 50% of score for that cover group it means all bins of every cover point have been hit 50%.

4.2.2. ts_controll_ff Coverage:

Ts_controll_ff is the signal for which we must measure the coverage as this signal is responsible to select filters which will work parallelly. It means that it will select four filter and select that from that four filter which filter will work parallelly and which will be in time sharing. There will be total of 16 combinations. So 16 bins will be created under one cover group which is ts_controll_ff covergroup. Only one combination is matching at a moment that's why it is showing 6.67%.

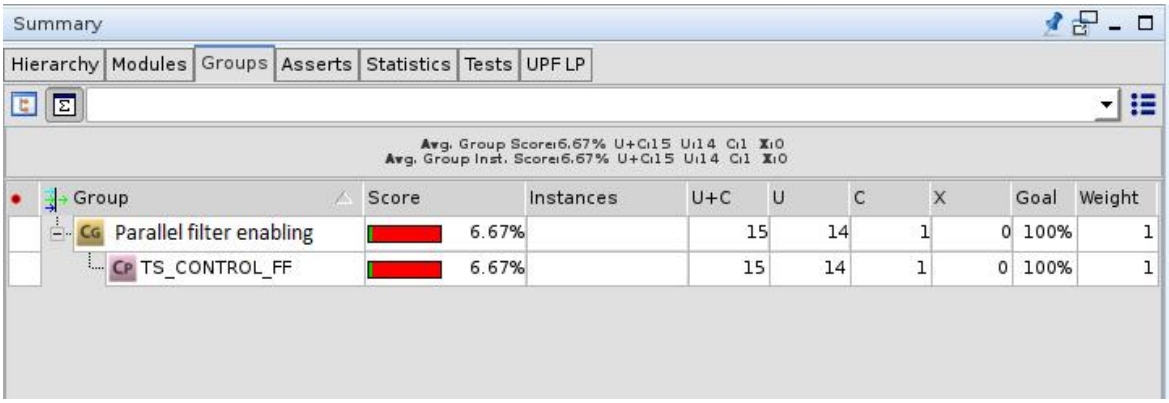


Figure 4.7 ts_control_ff cover group

4.2.3. Channel select Coverage:

Channel select will have different tap values in it according to the channel which is selected. It means that some channel has more noise and some has low noise according to that we have to remove noise and we have to give certain amount of tap value to the DFE block to remove the noise so those values are already predefined and expected from the channel. Different rate has different values of tap accordingly we have to measure coverage for that channel.

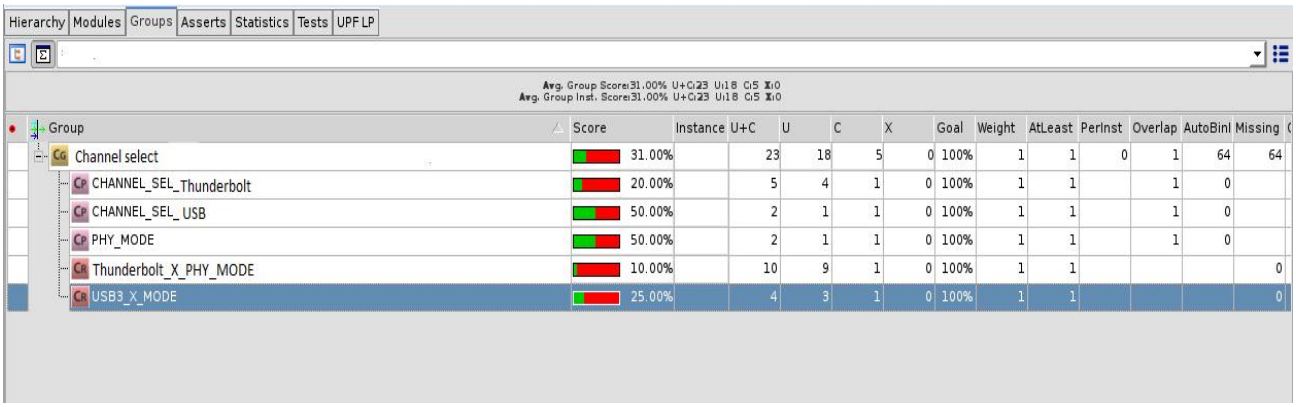


Figure 4.8 channel select cover group

So here you can see that thunderbolt is charging rate for the USB and other is USB mode and phy mode is the signal which will select that our phy is working in which mode is it display or USB or thunderbolt. In the last two coverpoint I have crossed the covergroup to have wide coverage scenario.

Conclusion

The verification process for the RX-EQ plays a crucial role in ensuring the reliable and accurate operation of USB interfaces. Through the use of comprehensive test bench environments, the functionality, performance, and compliance of USB devices can be thoroughly assessed.

Verification efforts include various components such as validating the physical layer, ensuring proper signal integrity, testing data transmission and reception, verifying protocol compliance, and evaluating interoperability with different RX-EQ block.

The verification process aims to identify and rectify any issues or potential sources of error, ensuring that RX-EQ block meet the desired specifications and adhere to the USB protocol standards. By conducting extensive testing, including scenario-based simulations and real-world tests, the robustness, stability, and interoperability of USB implementations can be assured.

Understood each RX-EQ sub-blocks for mixed signal IP. Prepared HVP plan for RX-EQ. WHile working on this project I had obtained good knowledge of regression, coverage and Mixed signal IP. Exhaustive verification environment is build to verify IP. Checkers and Cover groups are written to verify scenario based feature. Formal verification training has been taken to understand jashpher gold tool.

Bibliography

[1] R. J. Ruiz-Urbina, F. E. Rangel-Patino, J. E. Rayas-Sanchez, E. A. Vega-Ochoa, and O. H. Longoria-Gandara, “Transmitter and receiver equalizers optimization for PCI express Gen6.0 based on PAM4,” in *2020 IEEE MTT-S Latin America Microwave Conference (LAMC 2020)*, 2021.

[2] G. Rohilla, D. Mathur, and U. Ghanekar, “Functional verification of MAC-PHY layer of PCI express Gen5.0 with PIPE interface using UVM,” in *2020 International Conference for Emerging Technology (INCET)*, 2020.

[3] PHY Interface for PCI Express, SATA, and USB 3.1, DisplayPort and Converged IO Architectures,ver 5.1 by Intel corporation.

[4] Intel confidential documents.

[5] PHY Interface for the PCIe, SATA and USB 3.0 Architectures, Version 4.0

ORIGINALITY REPORT

7%	7%	4%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	docplayer.net Internet Source	3%
2	Submitted to Manipal University Student Paper	2%
3	www.ednasia.com Internet Source	1%
4	tekdevredesign.tek.com Internet Source	<1%
5	rei.iteso.mx Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		