Pre Silicon RTL Verification of Mixed Signal IPs through Real Value Modelling

A Thesis Submitted in Partial Fulfilment of the Requirement for the Award of the Degree of

MASTER OF TECHNOLOGY

in VLSI Design

Submitted By
PALLAVI SHARMA
601662012

Under Supervision of
Dr. ALPANA AGARWAL
Associate Professor

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY
(A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB
JUNE, 2018

INTEL TECHNOLOGY INDIA PVT. LTD.
Bangalore – 560103, Karnataka, India

Date: June 11, 2018

CERTIFICATE

This is to certify that **Pallavi Sharma (Regn. No. 601662012)**, a student of M.Tech.(VLSI Design), **Thapar Institute of Engineering and Technology, Patiala** has successfully completed one year (August 2017 – June 2018) internship programme in **Intel Technology India Pvt. Ltd, Bangalore**. Her title of dissertation is "**Pre Silicon RTL Verification of Mixed Signal IPs through Real Value Modelling**".

During the period of her internship programme, she was punctual and hardworking.

I wish her every success in life.

Jeyenth Vijayaraghavan
Engineering Manager
MIG, India

DECLARATION

I, Pallavi Sharma hereby declare that the work presented in this thesis entitled **"Pre Silicon RTL Verification of Mixed Signal IPs through Real Value Modeling"** in partial fulfillment of the requirement for the award of degree of Master of Technology (VLSI Design) submitted at Electronics and Communication Engineering Department, Thapar Institute of Engineering & Technology (Deemed to be University), Patiala is an authentic record of work carried out under supervision of **Dr. Alpana Agarwal (Associate Professor, ECED)** from June, 2017 to June, 2018. The matter presented in this has not been submitted either in part or full to any other university or institute for the award of any other degree.

Date: .....

Pallavi Sharma
601662012

It is certified that the above statement made by the student is correct to best of my knowledge and belief.

Date: .....

Dr. Alpana Agarwal
Associate Professor
Electronics and Communication Engineering Department
Thapar Institute of Engineering & Technology
(A Deemed To Be University), Patiala, Punjab

ACKNOWLEDGEMENT

I would like to convey my deep sense of gratitude to my project guide, **Dr. Alpana Agarwal, Associate Professor (Head), ECED** who is a constant source of motivation and firm support in carrying out this project. The support and supervision that she gave has helped me to progress in the project. Her co-operation is highly appreciated and I highly oblige to her for her valuable comments and moral support during this research period.

I would also like to thank my Mentor, **Subodh Kopay, Intel Technology India Private Limited, Bangalore** and my manager **Jeyenth Vijayaraghavan, Engineering Manager, Intel India Private Limited, Bangalore** for their esteemed guidance, Valuable suggestions and time throughout my internship. Their guidance and vast knowledge directed me to accomplish critical tasks smoothly.

Also, my special gratitude to my family and friends for their constant support and motivation.

DATE:

PLACE:

PALLAVI SHARMA

601662012

ABSTRACT

Validation remains an unavoidable, yet truly difficult and prolonged part of Intellectual Property (IP) configuration and manufacture process. With shortening timelines and expanding time-to-market necessity, IP producing houses are compelled to put more resources into validation. Hence validation remains an indispensable and essential period of today's IP design and integration procedure. Validation techniques can be partitioned into two stages: pre-silicon validation and post-silicon validation. Pre-silicon validation refers to validation activities that has been performed on a simulation or emulation model prior to fabrication of actual silicon. Pre-silicon validation verifies the correctness and efficiency of the design. Verification of Mixed Signal IP has been always a major point of concern. Though Mixed Signal IP have both analog as well as digital part, verifying them individually have become a tedious task. For modern IP, a technique which provide fast solution for verification in pre-silicon stage is necessary. The verification has been done at pre-silicon stage to make post silicon stage bug free. Because verifying a design at post silicon stage takes a lot time and costly re-spin process. As time to market has become a crucial factor, a solution must be there with a verifying IP having both analog and digital block with checking behavior of analog block in a real manner. Verification of anlog blocks have been done in digital environment which make simulation faster and get better performance. A methodology for verifying analog and mixed signal IP has been implemented through real value modelling. In this method, an anlog block behavior has been cloned in a digital environment. This real value modelling has been done with the help of hardware verification language such as System Verilog with OVM methodology

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Abbreviations

AMS	Analog and Mixed Signal
DFD	Design for Debug
DFT	Design for Testing
DFV	Design for Validation
DFx	Design for x
HSIO	High Speed Input Output
IP	Intellectual Property
JTAG	Joint Test Action Group
NLP	Native Low Power
OVM	Open Verification Methodology
PCS	Physical Coding Sub-layer
PCIE	Peripheral Component Interconnect Express
PRBS	Pseudorandom Binary Sequence
PHY	Physical Layer
PLL	Phase Locked loop
RTL	Register Transfer Logic
Rx	Receiver
SoC	System on Chip
SPICE	Simulation Program with Integrated Circuits Emphasis
TAP	Test Access Port
Tx	Transmitter

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

Mixed signal IPs are made of both analog as well as digital block. Conventional verification of mixed signal IPs are being done by individual verification. This individual verification for an SoC level design become a humongous task. Thus, verification of mixed signal IPs in an efficient way has become much more important.

Verification of mixed signal IPs is done at pre-silicon as well as post silicon level. Pre-silicon verification is the activity of verifying the task performed on simulation model. It is basically done before fabrication. Thus, pre-silicon verification helps verifying the correctness of the design with zero bugs before implementation on Silicon. This is important to verify in pre-silicon stage as detection and removal of bugs in post-silicon stage is very costly. Still there are some works needed to be done in post-silicon verification. Ideally all bugs are detected at pre-silicon stage.

For mixed signal IPs, verification of analog part takes a lot of time in an analog solver. This creates a need of an environment through which a mixed signal IP can verify in least amount of time. A solution is to mimic the behavior of analog part in digital environment.

1.2 OBJECTIVE

The main objective of dissertation is to implement an efficient verification methodology for mixed signal IPs. This is divided into three parts,

1) To implement an effective methodology such as to ensure that the PCIE PHY has no error making sure that it works properly.
2) Using Real Value Modelling, an efficient and accurate environment to verify mixed signal IPs is implemented for better performance.
3) Speed enhancement in verification through real value modelling by mimicking the behavior of analog circuit as analog simulator like SPICE takes hours to run.

1.3 PROBLEM STATEMENT

Mixed signal IPs are those that have analog and digital block in same chip. Mixed signal IPs has been used on PHY layer working on PCIE protocol. Verification of this PHY is main point of concern. Traditionally verification of analog and digital part is done separately. They follow the black box approach. In this approach verification of individual digital block is being done with the assumption of analog blocks are pre-verified and vice versa. Thus, for full chip SoC verification only concerned with connection between analog and digital blocks. As functionality and complexity of SoC design is increased day by day with deep

submicron process comes into picture traditional approach becomes obsolete. Verification of analog blocks along with its digital counterpart is essential for proper verification. One way to felicitate this feature is to use real valued signal representing voltage and current as verification parameter. This is commonly known as Real Value Modelling.

In Real value modelling, the behavior of analog circuit is mimicked in a complete digital environment. Main purpose of using this approach is to reduce time to market in terms of verification as analog SPICE simulation consumes a huge amount of time. Also, any bug found in post silicon verification could lead to re-spin, which costs a lot. Because any bug which get unavoidable could lead to loss in terms of cost and reputation of industry. Also, all the functionality of PHY layer must be verified as missing those can results into missing coverage or mismatch issues.

1.4 ORGANIZATION OF THESIS

The thesis has been divided into 8 chapters.

Chapter 2 presents the literature survey related to verification and real value modelling.

Chapter 3 deals with the background of mixed signal IP verification.

Chapter 4 specifies industry standard verification technique, emphasizing on analog circuit simulation.

Chapter 5 specifies the PHY and methodologies used in digital verification.

Chapter 6 describes an IP level verification with real value modelling.

Chapter 7 and **Chapter 8** describes results, conclusion and future scope.

CHAPTER 2

LITERATURE SURVEY

2.1 OVERVIEW

In this section survey is presented where verification process proposed by authors is defined. Also, it should be completely verified with all corner cases and that is done using regression. Also, in this section with the specifying the need of real value modelling in an analog circuit.

B. Vermeulen, *et al.* [1] described the need for improvement in verification due to complex systems. Because static time analysis, formal verification cannot assure that first silicon comes is bug free. As this error can go undetected as verification done at module of integrated circuit not to the silicon. And with the time to market reduction importance it is any error before tape out need to be detected. Thus, a design for debugging is a technique to found error with maintaining high importance to time to market come into the picture. And the task of design for debug is difficult for the designers as it is difficult to check the interior of chip. traditionally needle probe and other was technique used but as size decreased complexity increased in detecting interior chip so other methods came that support debugging software and those software interacts through TAP. Thus, concluded that debug system should provide access to chip functionality pin, access to memories and on chip signal and control execution of chip.

T.B Tarim. *et al.* [2] addresses today's mixed signal and SoC design flow and methodology issues and discusses the requirements for a successful design flow and methodology from industry point of view. Also, emphasized the importance of using behavioral models during TLV to reduce simulation time in large designs.

J. A. Lear, *et al.* [3] describes the differences between mixed signal verification and digital verification with highlighting the importance of mixed signal verification. In it verification approach is top-down it begin after block level are design. Then design under test are debugged after assembling. As it does not stop with production means test cases can be re-used with different methodologies and language driven verification.

P.D. Mulani, *et al.* [4] describes the use of Sytem Verilog to put up a verification environment which mainly focuses on assertions and monitor development. The I2C DUT has been verified for its functionality using the verification environment developed in System Verilog.

Peng Gang, *et al.* [5] proposed the behavioral modelling and simulation of analog circuit using Verilog-AMS. Verilog –AMS is derived from IEEE1364-2005 standard it combines the feature of Verilog-A and

Verilog-HDL which means it has the capability of handling both analog and digital signals. The main advantage is it provides signal simulation environment and single language for both analog and digital designers. Thus, making it easier for designer to work together having same simulation environment as they provide straightforward way to write behavioral model and for analog simulation brings strong event driven capabilities.it improves design cycle and thereby reducing top level simulation time.

X. Yang, *et al.* [6] described the difficulties in verifying mixed signal SoC because it is difficult to create verification environment of complex SoC with maintain accuracy of design with minimum simulation run time. Thus, to overcome this issue industry proposed several simulation tools but those run at transistor level so consume much time on comparing to digital RTL simulations. Thus, the author implemented circuit using system Verilog the contribution of it is that it employs code coverage verification, random data generation also adding assertion in the verification environment. And it is a faster approach as it can speed up regression period.

P. Y. Kuo, *et al.* [7] proposed the behavior model of low pass filter which is analog circuit is analyzed based on Verilog-A. However, the circuit become more complex as technology is improving more number of transistor are used to construct an analog circuit as transistors does not reach millions as digital circuits. In analog to analyze performance SPICE is sued which takes time to run. Thus, using Verilog-A to enhance the performance by upgrading the level of abstraction. And hence speed up the circuit simulation.

D. Juneja, *et al.* [8] emphasized the effect of discreet modelling on continuous system. Author main purpose is to emphasis on fact of need of improving mixed signal verification. The concern is to ensure signal integrity within the chip even if the complexity has been increasing in terms of functionality and size. And second to improve time to market as it takes time to simulate in analog environment thus creating digital environment to save time on verifying any bug upfront thus event driven modelling came into picture which is called as real n modelling. The strength of using real value modelling is that their evaluation is triggered based on discrete events. Another advantage of event driven model is easy to avenue digital verification technique. In this paper event driven modelling is discussed in reference to different circuits like comparator, filter and PLL also usage of assertion for modelling is highlighted and it is concluded by author that using real value modelling speed is increased up to 100x.

A. Jakobsson, *et al.* [9] described the mixed signal verification need by analog circuit extensions to HDL example Verilog –AMS, VHDL-AMS and System C-AMS. Due to increasing complexity of modern SoCs contribute to verification challenges. Not only the connection between digital and anlog blocks needs to be verified but also SoC performance to be maintain within specified requirements. Traditional analog and digital blocks are simulated individually which is accurate for analog need SPICE and for digital block

event-based HDL languages are used. In this paper PLL is implemented in Verilog-AMS and is compared with traditional analog simulation. On comparing with traditional method, it was observed that implemented circuit takes less simulation time by maintaining accuracy.

P. Das, *et al.* [10] prosed real value modelling approach which facilitates analog accuracy in digital simulation. In this paper implemented PLL using behavioral models to get results faster and accurate. Traditionally analog simulators were used but due to shorter time to market verification needs to be improved and is done by providing digital solvers with analog real values. Modelling of PLL is done by replacing charge pump and frequency detector by their digital real value models. By doing this simulation time has been reduced up to 30% as compare to AMS models. Also, paper introduces corner variation that is observed from SPICE and motivation to introduce in real model to cover all aspects that is full coverage with respect to SPICE.

A. V. Craciun, *et al.* [11] describes the benefit of using real number model. As mixed signals are the fastest growing segment in semiconductor industry so challenges for verifying the complex chip also increased. And to reach reasonable speed which is a troublesome with analog simulators behavioral modeling of analog circuit is must. This approach is used because it is 5 to 100 times faster than analog simulator like SPICE also speed also depends upon the application. Behavioral modelling is done in different languages like Verilog-AMS, Verilog-A, VHDL-AMS and System Verilog. Main challenge is one that is creation of behavioral model as analog designer are similar to circuit but less with coding skills and similarly digital designers know more of coding than analog circuit. Analog blocks are model as discreet data using real number models. The main benefit by using real number model is that they are nearly as fast as purely digital simulation environment and takes less time compare to analog simulation environment. In the paper author describes op-amp first in s-domain then implement the function in VHDL. On comparing the result of SPICE and VHDL he concluded the behavior of both model is same also VHDL implementation does not contain any second order effects.

N. Georgoulopoulos, *et al.* [12] proposed real number modeling for the implementation of analog and mixed circuit. Due to mixed signal challenges verification is a major concern with achieving high accuracy. A mixed signal rely on simulation runs as they take long time to run which leads to delay in manufacturing process. Thus, solution to this problem is modelling an analog circuit using real number modelling. In this paper an analog circuit i.e., three-bit flash ADC which is used in numerous applications like electronic test equipment is implemented using real number modelling. The implemented circuit is compared with three different models: Verilog-A and Verilog –AMS model and SPICE.

CHAPTER 3

MIXED SIGNAL VERIFICATION BACKGROUND

3.1 VERIFICATION

Verification is the process of determining whether the implemented design is correct. Verification is done to make sure that it works as expected requirement's i.e., it meets the required initial specification of the product such that to ensure that it will be used efficiently by intended user or customers.

In other word, to ensure that product is accurate and correct or information provided is true on providing a comparison. Verification become more important day by day as the complexity increases so any products if have any sort of bug creates re spin and impact cost also to prevent time to market delay so it is important to verify the design in design flow. As fast a bug is detected faster, more time which could get wasted in re-spin will be saved.

3.1.1 Need for verification

Consider a case in industry where product is sent out to customer without proper ensuring that the product was working as intended. If could be a possibility that product which is sent out is having some issue which will lead to insufficient loss as re spin costs much also will impact reputation.

Efforts taken in verification is more than design effort, the design is construct based on specifications and then that specification is converted to by design engineer into implementation which has architect specifications which is then must be verified by a design engineer including all cases and thus it is the work of Verif engineer to prove that the design which is implemented is actually work as according to specifications.

Verification effort should be independent of design effort. If both design and Verif engineer both follows same rule of design they may commit same error. Also, cannot avoid verification as it cost too high and takes too much time. Thus, it is recognized as important part in industries. Thus statistically 70% of ASIC design budget is devoted to verification only.

3.1.2 Overview of Typical ASIC Design Process

Functional verification is the activity where the design or product is tested to make sure that all the functions of the device are indeed working as stated [13]. This activity ensures that the features functions as specified. It is noted that some of the device features may or may not be visible to the user and may be internal to the design itself. However, it is imperative that all the features are verified to operate correctly as specified. This verification activity usually consumes the most time in the design cycle. The verification process in ASIC flow is shown below in Figure 3.1,

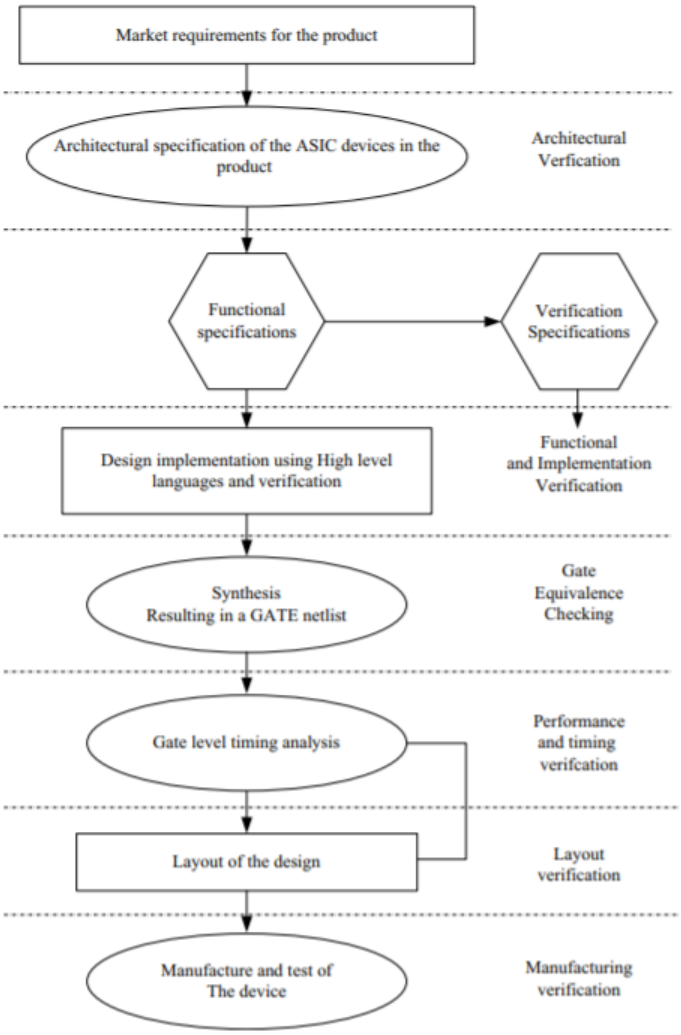


Figure 3.1 Verification process in ASIC flow [13]

The ASIC flow starts with defining the features of the system incorporating the device to be designed. At this step factors like cost, time to market, feasibility, performance and profit margin are determined.

The next step in the design process is define architecture such as to address all the product needs. Then the architecture requirement is mapped into software and hardware components. Then all the architecture specifications are translated into functional specification and which leads to planning for implementation and at this stage all necessary license and needed resources are identified. Once functional specification is ready then design and Verif team work is divide in parallel forms.

Design effort starts with description of device and is done by using RTL whereas Verif starts with creating module and chip environment [13]. The design effort and Verif effort are verified against the functionality specifications. And if any mismatch found then it is a bug. In verification process regressions are run to cover all cases and when verification is done then design is synthesized into gate netlist. Simulations are also run on gate netlist ton ensure the integrity of device which is then run with delays models.

The gate netlist used to generate the pattern which is then used to verify if the fabricated device is good or bad. This gate netlist is then translated into layout and then the device is fabricated. After this process manufacture verification is involved. Manufacture verification is done to check that the device is manufactured correctly with no flaws.

3.1.3 Analog verification

Verification of analog circuit at device level is done using simulator like SPICE which uses compact model such as to get the behavior of circuit schematic. The circuit is said to be verified if required performance is achieved and for that analog designer are interested in measuring some characteristics of analog circuit output [14]. The circuit are simulated with various analyses available in SPICE like AC analysis, DC, transient analysis and then inspect that all result came meet the spec or not.

3.1.4 Digital Verification

Digital verification goal is different from analog verification. In analog verification try to extract number of parameter of circuit. But in digital verification, digital designer wants to know that large collection of test cases meets the correct function. A digital design with N states variable has 2^N states which is mutually independent. However digital verification is not smooth as analog as it takes care of each states possibility. Thus, validating a digital circuit every possibility, it goes through. Commonly used digital simulators are VCS by Synopsys and ModelSim.

3.2 ANALOG CIRCUIT SIMULATION

Over the last decades, analog circuit are simulated using analog simulator like SPICE. SPICE is Simulation Program with Integrated circuit Emphasis developed at University of California and it's become standard simulator worldwide. To ensure the correctness of design before manufacturing, designer verify circuit using SPICE simulator. They are mostly used due to reason that they cover wide range of component from

passive component to active components. The behavior of analog circuit in SPICE is predicted by writing its text description of the circuit. But SPICE simulation takes a week or more to run even if it is a smaller portion of the design. Even the speed of SPICE is improved in recent years but still they are not liable according to industry standards as accuracy is bit compromised [15].

In the design development cycle, it is significant to have different simulation tools such that they can be utilized so well to get high level of productivity. Modelling an analog in high level s a crucial task for designer to gain productivity.

For a top-down approach as discussed in [15] it can be possible to have wrong assumptions with wasting lot of effort and time to design. Thus, by increasing complexity it is important to provide a solution for analog mixed signal integrating circuits thus an abstract level modelling in beginning will be helpful. Abstract model is shown below in Figure 3.2 which has digital model with analog model where digital block is written thorough HDL and analog model is written through AMS language and some with SPICE netlists.

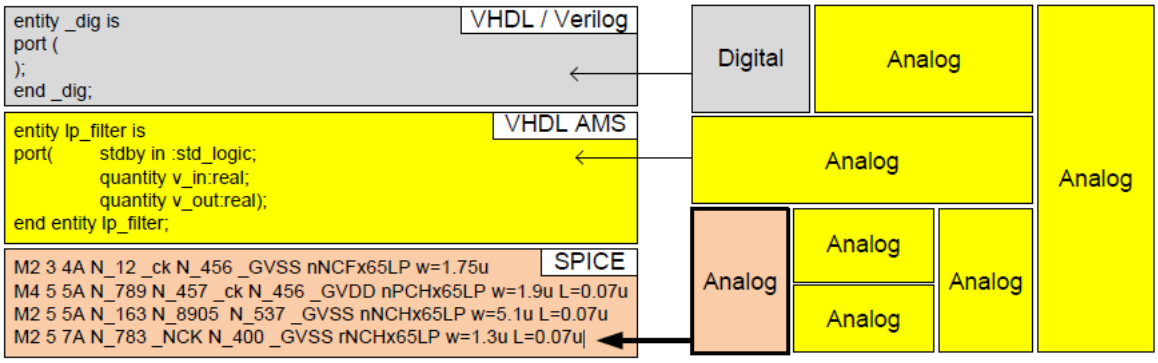


Figure 3.2 Top down approach in analog centric designs [16]

This approach is analog centric while digital centric design top-down approach is used. Verification of analog centric circuit are done using SPICE simulators while digital simulation can be done using behavioral model of circuit [16]. Figure 3.3 shows that an analog circuit can be modelled using real value modelling while digital circuit is implemented using HDL languages.

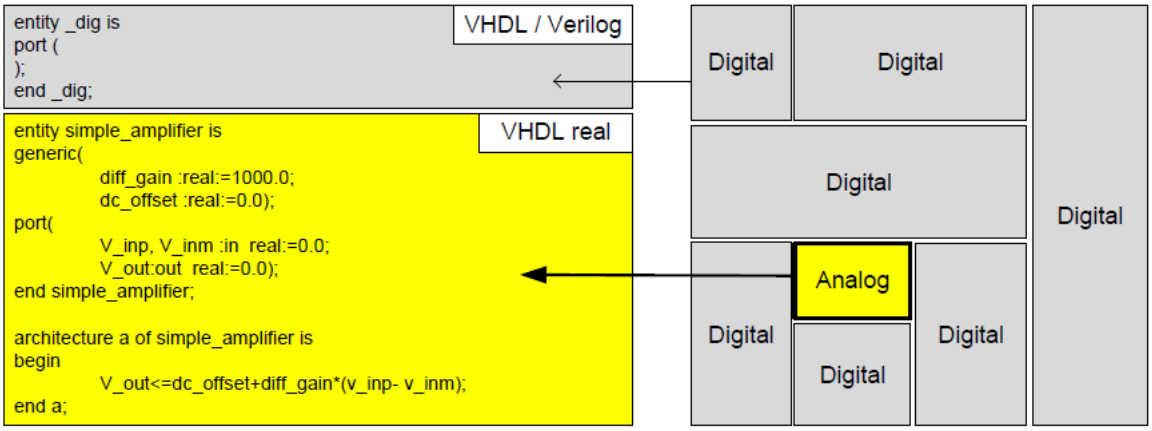


Figure 3.3 Bottom up approach in digital centric designs [16]

With this above approach can have the possibility to provide event driven simulation for full-chip. On concluding one can say that as the process node shrinks with increasing complexity as millions of transistors are on same chip this verifying analog circuit with SPICE simulator is not good as time to market is being significant from industry point of view thus providing abstract model for analog circuit is important is more reliable.

Thus, thesis majority provides the idea of such different ways in which can provide analog simulator with minimum simulation run time while maintaining performance of circuit is shown.

3.3 GAP BETWEEN ANALOG AND DIGITAL VERIFICATION PROCESS

Verification process including creation of test-plan, developing test benches and simulating and comparing results with the specification that are provided in beginning of design flow by designer [17].

In past verifying analog part and digital part was an easy task and only the connection between block was the main thing of concern but now with modern technology complex chip has both analog as well as digital part and is called as Mixed Signal IPs that has both functionality of analog and digital. And thus, the verification cannot be done separately [18] thus the full chip verification is needed. Complexity of mixed signal design is shown below in Figure 3.4.

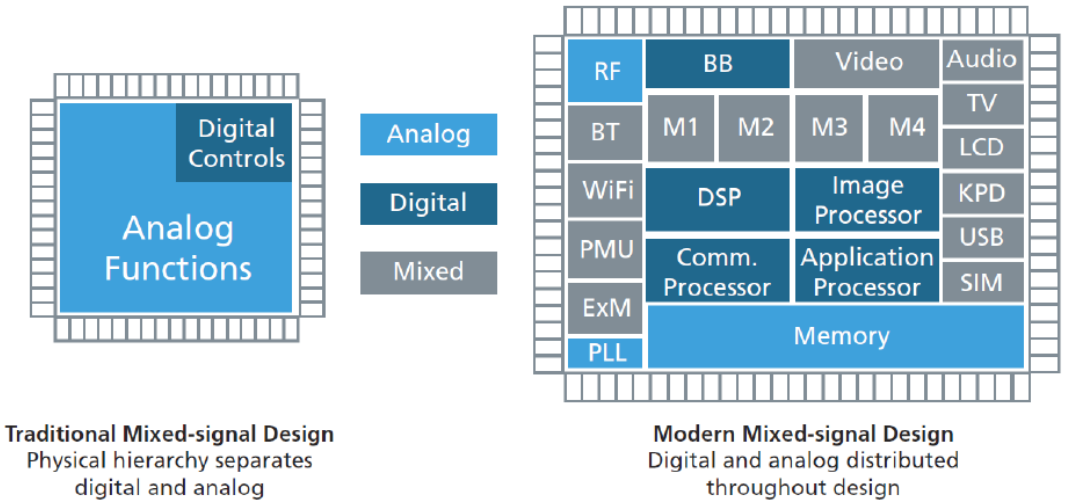


Figure 3.4 Complexity of mixed-signal design [19]

The verification of analog block is done in bottom-up flow where every module is implemented from its respective specification and verification is done using SPICE simulator. And this implementation is done using transistor level and verified after verification blocks are integrated with similar blocks. But the approach is good only for smaller design. As design become complex, the process of verification becomes more tedious [19]. On the other hand, digital verification is a top-down flow. Where the verification is done using different methodologies using coverage. Assertion and regression are run to cover each and every aspect of design.

3.4 CHALLENGES OF MIXED SIGNAL VERIFICATION

With increasing complexity verification for mixed IPs are growing in industry sector. The main challenge is direct testing methods is become harder to apply nowadays [17].

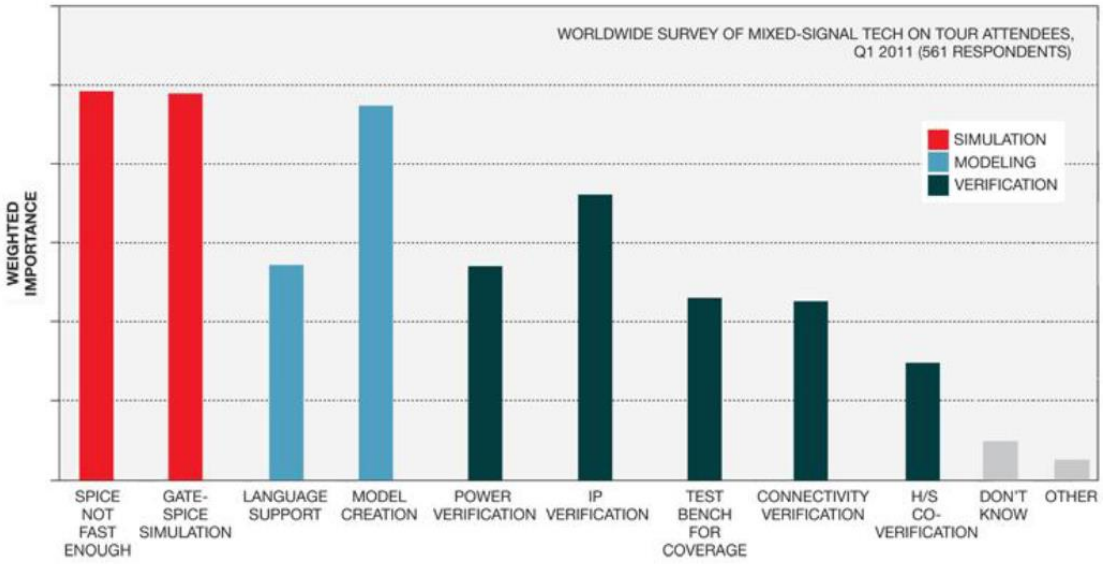


Figure 3.5 Results of a 2011 survey showing biggest mixed-signal verification challenges [20]

From the Figure 3.5, it is concluded that analog verification for mixed signal IPs is a bottleneck. Even though advancement in SPCIE has increased speed with using FAST-SPICE but at cost of accuracy. Because still they take a lot time to run.

To overcome this problem with SPICE, now teams in industry are turning towards behavioral modelling of analog model. The advantage of this model is that they increase the speed but creating such an environment is challenging [20].

In 2011 conference of DAC, discussed the need of digital verification environment for analog models which is more structured and verifying in to-down approach is easy. For analog mixed signal verification methodologies like assertion, metric driven and open verification methodology (OVM) are needed [18]. Because of high costs of product re-spin and time waste leads for opting new strategies which could lead to high productivity of product. The overview of older fashion of mixed IPs in two ways are stated in coming section.

Black Box Approach: Black box approach is a technique used to verify analog and mixed signal in a digital dominant environment. Basically, an analog block which is pre-verified is integrated into a design and this integration is done using its high-level abstraction model. It contains interface description. By this method simulation speed is increased but degrades the quality. But today chips are becoming more complex because chip not have analog and digital block but feedback between both block and interconnection between them makes it more difficult to verify. Thus, replacing an analog block with black box approach is not reliable way nowadays [21].

Capture Approach: the analog part is specified and checked in simulation with some boundaries at simulation times and this waveform are then replayed as input of AMS simulator. But this is not reliable as still miss the feedback path between analog and digital blocks.

CHAPTER 4

METHODS FOR MODELLING MIXED SIGNAL IPS

To verify mixed signal IPs various technique has been introduced. This section covers all methods to verify anlog and mixed signal design using different methods.

4.1 BEHAVIORAL MODELLING

Describing an analog circuit in higher abstraction level makes simulation of this model more effective thus behavioral modelling is a key component in verification methodology for mixed signal. Behavioral modelling is where actually one mimics the analog circuit behavior.

While creating model for such analog and mixed signal model is not an easy task there are several challenges:

- The analog circuit which is implemented should be verified to make sure that the implemented model is working as specification with high accuracy.
- If there is any specification changes it must be synchronized with circuit.
- For any issue while simulating, the circuit should be written in such way that it doesn't not converge any issues.
- The main purpose of model must be understood and according to that model, or architecture must be chosen. Models are developed before circuit availability in a top-down approach. While in bottom-up approach for verification model needs to be verified with an already implemented block.
- For implementing such model must have good understanding of analog and mixed signals, coding and debugging

Analog and mixed signal can be modelled using several possible features. Depending on the application and complexity of design it can take from minutes to week or month to simulate for a design [22].

4.1.1 Types of Modelling

It is important to use different modelling schemes for verification of a large integrated chips. Modelling can be done in different manner depending upon the application are and time to market need of industry.

Device based design (Spectre, SPICE) – in this circuit is build using standard transistor level technique.

Analog modelling (Verilog-A) – Verilog-A is the subset of Verilog-AMS which describes the current and voltage relation of analog part.

Mixed Signal modeling (Verilog-AMS, VHDL-AMS) - it is the extension of VHDL known as IEEE 1076 VHDL. It describes the behavior of analog and digital blocks.

Real Value modelling (VHDL, System Verilog) – in this analog block is model as a discrete real number data. It provides extension to support full chip verification of SoC having analog blocks.

The analog block can be abstracted using transistor level netlist or FAST SPICE depending on application and according to what form of abstraction has been used the accuracy and performance and effort taken while modelling an analog circuit varies.

The graph with taking account of accuracy and performance is plotted in Figure 4.1 shown below where analog block is abstracted in SPICE, FAST-SPICE, Verilog-AMS, VHDL-AMS and Real value modelling and pure digital simulation environment for analog block is plotted [22]. The performance can varies depending on applications.

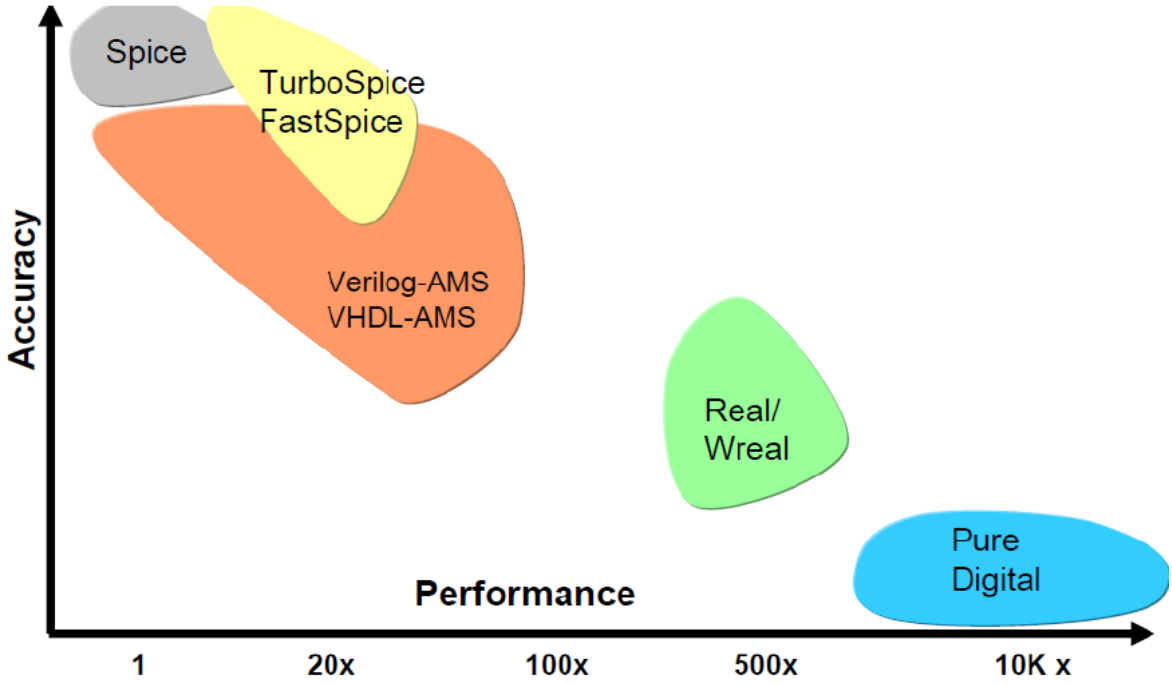


Figure 4.1 Model accuracy vs. performance gain for mixed-signal simulation [22]

Depending on application each has their own advantages and disadvantages. For an analog circuit SPICE is used as golden reference for simulating. Then FAST SPICE where used which were advanced version of SPICE by speeding up simulation but with compromise with accuracy. Whereas Real value Modelling provide high performance with not much high accuracy but when it comes of performance can go with real modelling. Pure digital simulation is less accurate and does not provide good performance but they can be used to check the connectivity or other verification tasks.

Now how much effort does need for a model to create is plotted in next figure. From the Figure 4.2 shown below can be concluded that SPICE can be easily modelled but they run slowly. And to create model can take time from a week to months. For real modelling convergence is not an issue and same effort is with digital model [22].

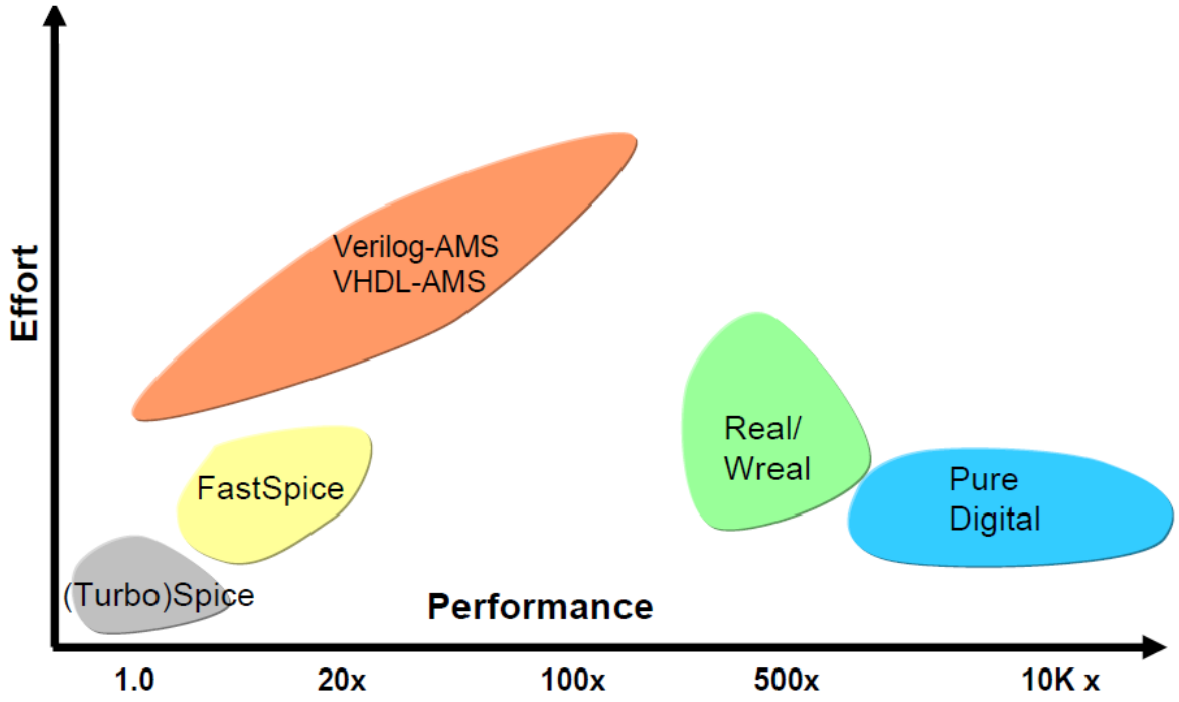


Figure 4.2 Required effort vs. performance gain for mixed-signal simulation [22]

It is important to understand for what purpose what model must be used which is covered in next section.

4.2 ANALOG MODELLING

For modelling electrical behavior of analog circuit pure analog languages like SPICE are used. Thus, for modelling analog behavior Verilog-A is a standard language. Verilog-A language creates the relation between voltage and current of the system. Can directly write impedance characteristics and integral and derivatives using this language. Then Verilog-A model converts this in some set of equations whether linear or differential which is suitable for simulator.

On comparing to a transistor level this well-defined model can sped up to 10 to 70 times of SPICE simulation [22]. But speed of simulation also depends upon the application for which the equations are formed. By using behavioral modelling of analog circuit performance is increased.

4.3 DIGITAL MODELLING

For a digital circuit with digital input and output characteristics a pure digital solver is used. VHDL, Verilog and System Verilog are the language through which a digital circuit is modelled. These are efficient in handling logic and timing relationships of input/output using a digital simulation environment. Although this approach is not for analog signals. Thus, this model is used for an entirely digital model where an

analog block is assumed to be black box means, assuming that analog block is pre-verified. But this digital approach is extended where an analog bock can be models in a digital environment taking advantage of modelling.

4.4 MIXED SIGNAL MODELLING

In a mixed signal modelling and analog and digital model solver come in single simulator. For modelling mixed signal languages like Verilog-AMS, VHDL-AMS can be used [22]. These languages allow modelling of mixed signal in a simple manner as analog block can be model through its electrical behavior modelling approach and digital block can be modelled using discrete or event modelling technique. With this two simulation algorithm data can be transparently passed. And is very efficient way to write test benches for mixed signal signals. For analog and mixed signal, Verilog-AMS can be used to write read both analog and digital blocks which makes it an efficient environment in which one can implement an entire system. And further real number modelling is a technique used to write mixed signal models. The simulation speed for AMS mode depends on application and replacing transistor level circuit.

4.5 REAL VALUE MODELLING

Real Value Modelling (RNM) is a modelling technique which actually mimic the analog circuit behavior. This is a special technique where electrical signals of analog blocks are represented using time varying real values. For analog simulator, some set of equations are defined in a model. And this model which is defined by equations are enforced by simulator by adding constraint by applying Kirchhoff's law and then solve this constraint of equations at every step to determine the voltage and current from those equations.

But in a digital environment there is no set of equations no Kirchhoff's law. In this output is directly evaluated using input disregarding feedback or any voltage or current which can create interdependencies [22]. Real value model procedure is easy where just using input/ output transfer function can write mathematic algorithm and check whether the corresponding output is changes on change of input. Thus, it is a straight forward concept. Also, the inputs like voltage, current and power are pass as a real value in model and later check whether they are coming in with some specified tolerance. Thus, biasing should be proper ad checking of biasing is easy task. And it is not difficult to create behavior model of analog circuit as an analog circuit verification on transistor level to behavior in higher level of abstraction is already a known process using VHDL-AMS thus using Real model is not a difficult task.

For Real Number Modelling languages like Verilog, System Verilog is used which has the property where a signal can be represented as a real i.e., floating point value which makes them behaves more like an analog circuit values as in digital actually deal with four vales that is 0, 1, x and z but not with the real modelling case where actual float values called real values are used.

CHAPTER 5

METHODOLOGIES FOR VERIFICATION

5.1 DESIGN FOR x (DFx)

DFx is design for x where x can be testability, validation, debug or manufacturability. Figure 5.1 shows an overview of working features of DFx.

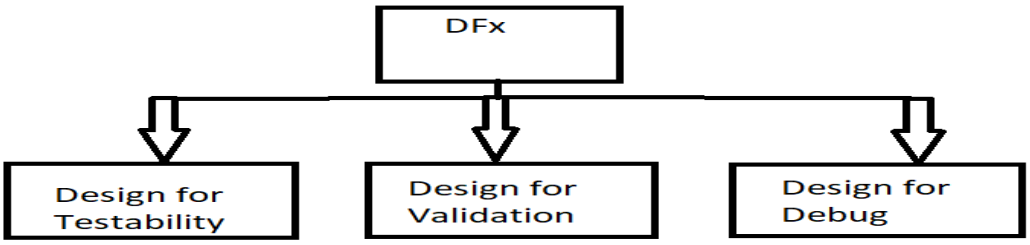


Figure 5.1 Flow for DFx

5.1.1 Why need DFx

As today with the rapid growth of microelectronics technology, with the need of higher performance and low cost of system other factor which comes is complexity as the as with more functions the integrated circuits become more complex and with more complexity it is become difficult to test complex chip. With more complexity testing cost become more so as to reduce testing cost and to make it easy to test as chip and extra circuit is added during design stage which is called DFx.

5.1.2 Design for Testability (DFT)

Design for Testability (DFT) consists of two very important terms. "Testability" is a condition of a circuit that makes it possible, easy, and cost-effective to test and diagnose the circuit under test. There is a wide acceptance that such a characteristic should be part of electronics integrated circuits, boards and systems, too. Without having DFT as a part of integrated circuit fault can go undetected which makes it more difficult to redesign and repair the design.

Thus, testing is the method to check if the chip is working correctly after manufactured. Testing means to check whether design is working properly that is if it meets the desired functionality. Unfortunately manufacturing process of integrated chips is not perfect thus needed to check if design meets original requirements which is done using design for testability another reason is complexity which makes cost for testing high. In the customer's environment tests are applied at many steps in the hardware manufacturing flow and, for certain products, may also be used for hardware. There are tests which are executed using Automatic Test Equipment (ATE) and tests are driven by this test program with ate or, in the case of system maintenance, inside the assembled system itself. In addition to find the defect and indicate that test is failing test should be able to diagnose the defect and should be able to get information about the reason of the encountered test failures. Thus, this information can be used to locate the source of the failure.

In other words, a circuit which is under test called DUT thus applying some pattern to get the output from it to check whether it is working correct. To check the correctness, one compares this circuit with some other. Thus, the response of vectors (patterns) of design under test is compared from a reference circuit using the same patterns. On compare if the response matches, then the circuit is correct that is it is behaving as expected. Otherwise, the circuit is not manufactured as it was intended.

5.1.3 Design for Debug (DFD)

One of the key components to enabling the debug process to go smoothly is the implementation of features in the design to aid debug. This is known as design for debug, or DFD. In addition, there are a number of tools and methods which are used in concert with these design features once silicon has arrived to accelerate the debug process.

Typical post silicon methodology consists of booting low-level console user interface, running legacy tests, doing a boot of the favorite operating system, running pre-silicon tests, running post silicon tests, locating and diagnosing bugs, reproducing and analyzing bugs on the RTL model, and using microcode patches to provide a workaround so that validation may proceed. This is typically done with low observability using the small amount of DFT features available on the chip. Post silicon validation effort is done along two major directions: system validation and compatibility validation. System validation is primarily focused on validating the CPU and chip sets in an embedded systems environment that uses the new silicon in a multi-way configuration and has special monitor software to download the validation tests into the platform for execution on bare silicon.

There are two main goals of validation testing: first, to thoroughly test the functionality of the chip; second, to structure the code and plan the test environment to quickly and efficiently debug, isolate, and report problems and issues that come up in the lab while testing. This means putting together an organized system of debug statements, logic analyzer triggers, event logging, and so on. There are three types of validation tests: directed diagnostic tests, stress tests, and real-world tests. Although there is some duplication in the

functionality tested, these types represent different test dimensions that are all needed to ensure the quality of the silicon device.

5.1.4 Design for Validation (DFV)

Design verification and validation are very important as they directly influence performance of design and which defines functionality of product that is correct and also customer's perception.

It is process where actually check build version of product and check whether it is as accordance to what verified. The things involved in Design for validation are jitter analysis, Pattern generation, Tap security etc.

5.2 DIGITAL VERIFICATION METHODOLOGY

For a robust verification environment methodology is a way. Basically, methodology helps us to do a task in a well-founded system by taking benefits of predefined and some set of mandate. For constructing test benches standard libraries from methodology helps in making a well-defined environment. For any company with this time to market importance and increasing competition adoption of such methodology is very important also to improve product utilization.

Thus, for verifying integrated circuits various methodology has been introduced. Thus, in terms of reliability digital verification has been progressed. The test benches are implemented using hardware description languages. But due to some drawback there comes the need for hardware verification languages. And these are System C, System Verilog. By using this high verification language, a semiconductor with no bug and high quality can be made. Table 5.2 shows different methodologies from different companies to magnify verification process has been shown.

Table 5.2: Different verification methodologies

Verification Methodologies	**Vendor**	**Year**	**HVL**
e Reuse methodology (eRM)	Cadence	2002	e
Reference verification methodology	Synopsys	2003	Vera
Advanced Verification Methodology (AVM)	Mentor	2004	SV/SC
Universal Reuse Methodology (URM)	Cadence	2006	SV/e
Verification Methodology Manual (VMM)	Synopsys	2004	SV
Open Verification Methodology (OVM)	Cadence/Mentor	2008	SV

5.2.1 Open Verification Methodology (OVM)

Open verification methodology is a shell for verifying digital hardware using System Verilog. It helps in verifying a design in less amount of time by providing self-checking mechanism. Main purpose is to replace old hardware description language used for writing test benches with robust methodology based on reusable verification component in well-defined way. The key features which are needed to simulate a design under test that are data items, drivers, generators, monitors, scoreboards are modelled through system Verilog classes same way is done by OVM components. Basically, this test bench is ready to use in a configurable verification environment for a system [23]. For a specific design each OVM component follows some set of flow or architecture to verify the design under test.

5.2.2 OVC Overview

The following sections describe the detail of OVM components. The Figure 5.2 shows the OVM topology.

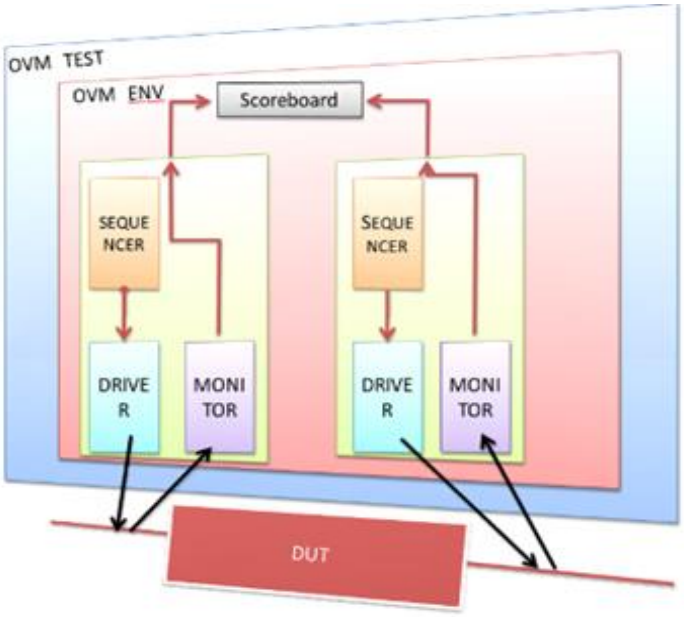


Figure 5.2 OVM topology [23]

Data item (Transaction)

Data items are the stimulus to the design under test. Basically, a driver needs to check the signal it drives to design under test and is done at bit level but if the Verif person are away from design under test this verification is of no worth. So, this OVM is focused on what is known as transaction level modelling. And these transactions are the smallest data in the model. Examples are networking packets etc.

Driver (BFM)

Driver is an entity which follows the logic which operates the design under test. Driver receives the data from the sequencer or generator and drives it to design under test by sampling and drive the signal. Example read/write signal.

Sequencer

Sequencer control the item given to driver. It is an advanced generator. It behaves the same way a normal generator does by generating input and returning some stimulus on having any request from driver.

Monitor

The other OVM component is monitor where one directly writes the code that directly connect with DUT. Here you don not drive the signal only sample them thus it is a passive entity. The signal which is sed for verifying at higher level are basically converted into an effective information by monitor. It collects data items. Also check the availability of information and notify other components [24]. Also, do checking such that DUT outputs are according to protocol specification. In general, interest is in getting the packets received from sequencer and give them to scoreboard. Later scoreboard can compare and check this response from design under test.

Agent

Agent where connects DUT, sequencer, driver and monitor. Although each component work indecently but still you need an environment to known the roles and configuration. An OVC can have more than one agent. For example, some are transmitter agent and other is receiver agent which react to data items requested.

Environment and Scoreboard

It is the top of OVM component and it has one or more agents. It contains configuration property which helps to enable to customize the behavior and topology and make it reusable [24].

5.2.3 System Verilog OVM Class Library:

For constructing a well-defined, reusable verification object and environment all the building blocks are provided from system Verilog OVM library. The Figure 5.3 shows OVM class hierarchy.

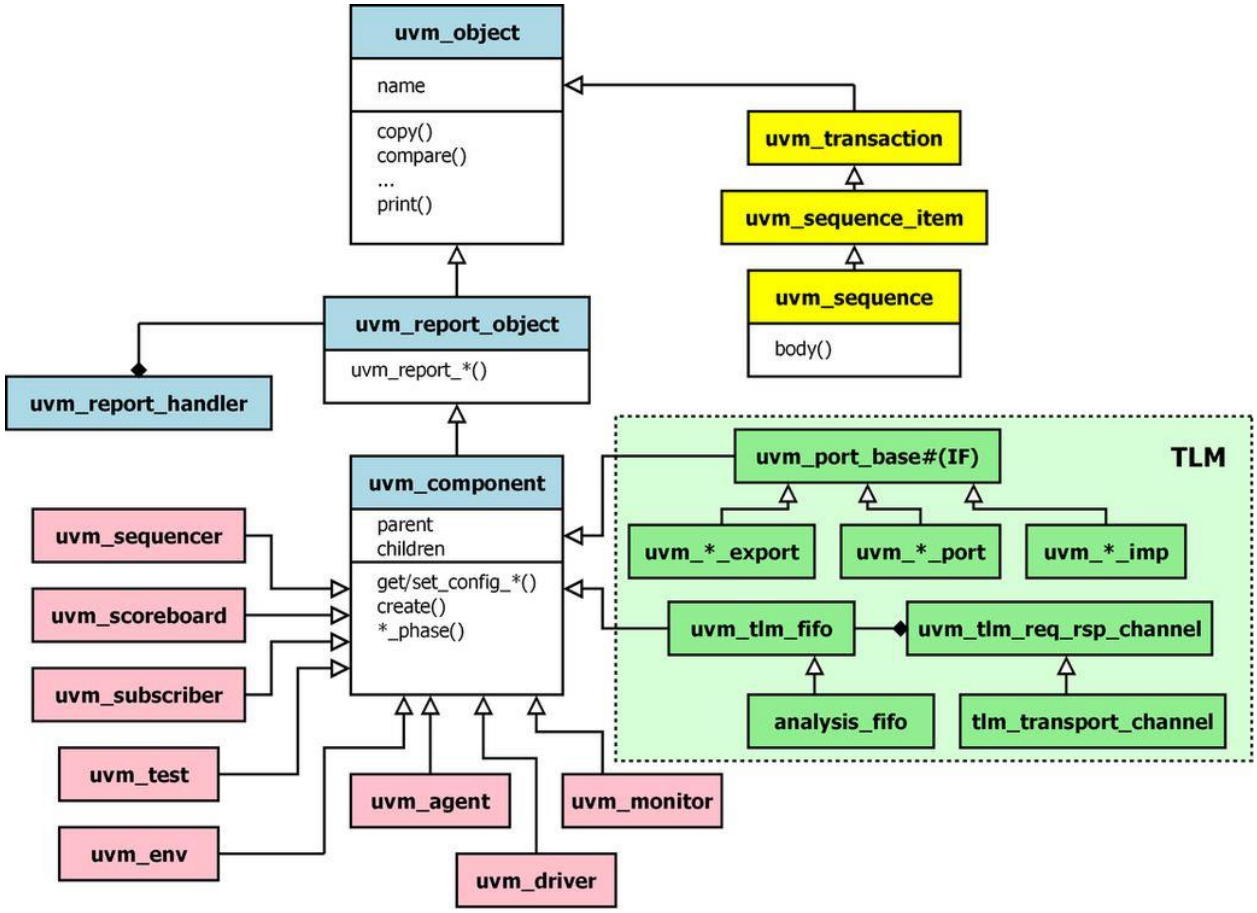


Figure 5.3 OVM class hierarchy [24]

Libraries have macros, base class, utilities. The main advantages of using OVM library is

- A robust set of built-in features
- Correctly-implemented OVM concepts

5.3 VERIFICATION THROUGH REGRESSION

As a design under test is surrounded by many sub modules like bus functional model and simulator controller and other log files which help to create and environment so verifying each and every aspect which makes a complete design is of major concern the design under test with different sub module is shown below in Figure 5.4.

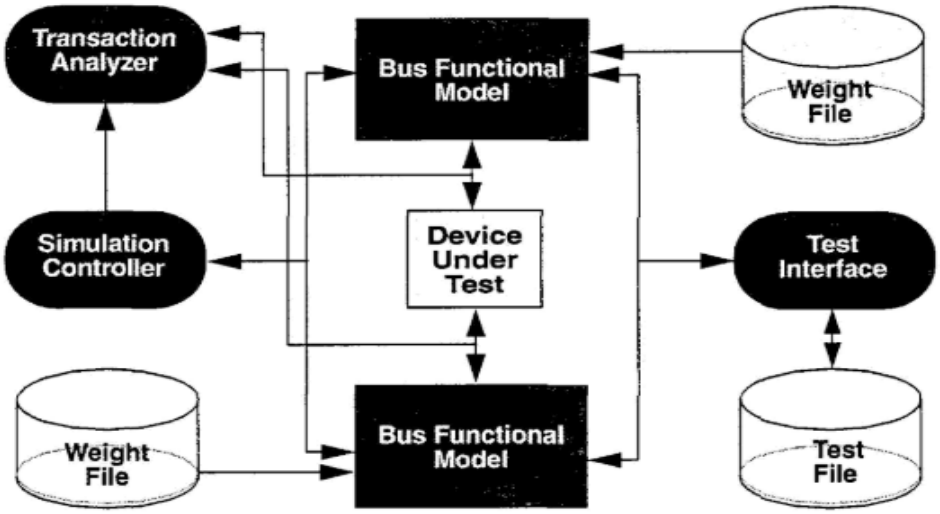


Figure 5.4 Design under test environment

Design under test can be verified in a proper manner as shown below in Figure 5.5.

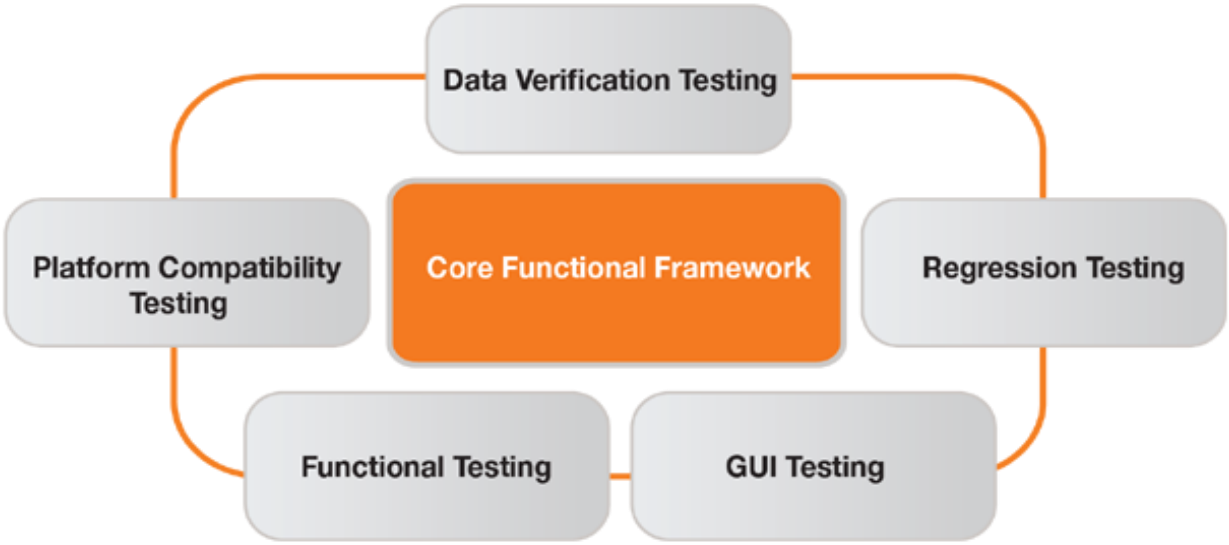


Figure 5.5 Testing flow of a design

Regressions is the process of re-running a modified test to check that faults which are being fixed have re-merged. If any changes in RTL or due to new changes some hidden bug which can be masked in design under test impact the tests and it will fail so to check that with RTL modification or any model changes

tests won't get effected. Another factor is that regression covers all corners of DUT. As a test may be so many vectors which can't be run interactively as running integrative each seed takes so many times so to save time regression is the way to get result with covering all corners. Thus, for equalizations test as it has number of slice count and running each case is not possible thus launch regression to check if any slice is not behaving as expected. The output for regression result is shown in next section. The Figure 5.6 shows the process of regression.

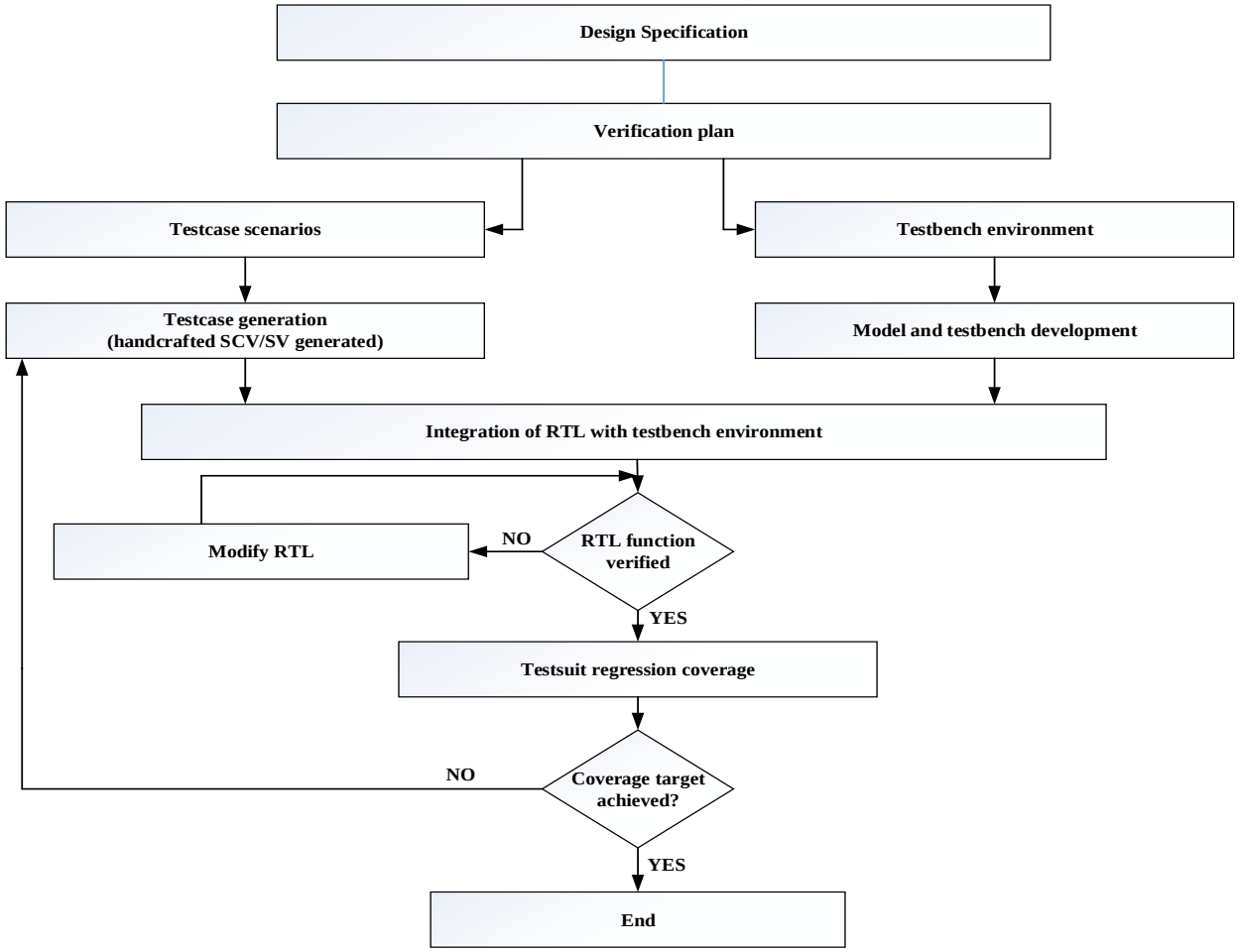


Figure 5.6 Flowchart of regression process

Regression are run in different modes:

5.3.1 Native Low Power and NON-NLP

NLP stands for native low power. A design usually has different power domains. In industry and RTL simulation is done assuming all domains are on. If could possible you have to domains one is on and other is off thus to run in real behavior NLP is run with UPF. The behavior is shown in Figure 5.7.

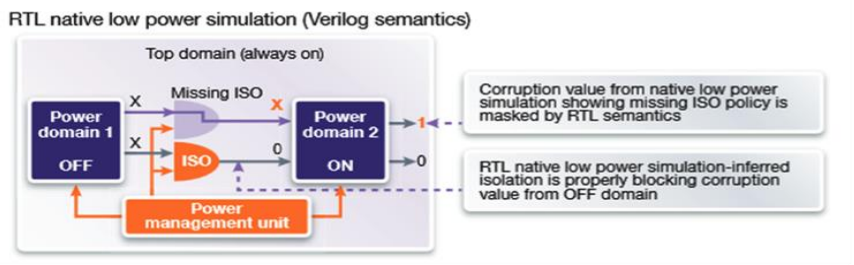


Figure 5.7 NLP simulation with missing ISO [25]

Figure shows two domains connecting through some isolation cell to sure the behavior. Suppose if isolation cell is off in that case values X from shutdown downstream off state may propagate to on state thus Verilog RTL could convert this scenario. Thus, through NLP VCS simulation this behavior can be removed [25]. Thus, behavior using NLP is shown below in Figure 5.8.

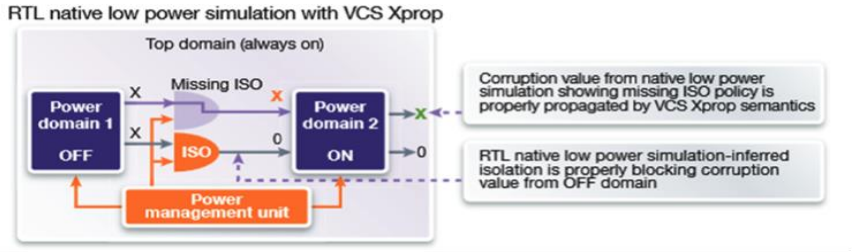


Figure 5.8 NLP simulation with ISO [25]

Here corrupted values are accurately propagated to second domain.

5.3.2 Isolation and Non-Isolation mode

A design under test is run interactively to check the behavior or design if working as expecting. A design inside has so many sub-designs and they are run separately and check by owners whether it's working or not. If a design is run individual not concerning whole design is what call running a test in isolation mode. But when a design is run with respect to other design to check if design with margining with other design do not affecting other design and others design also not changing our design intent is called running a test in non-isolation mode.

5.3.3 Meta-stability

When the timing constraints are violated the design comes in met stability state. For addressing Meta stability issues synchronizer are the solution. Thus, for a synchronizer flop it can give stable output between

say X_N to X+N. thus it is divided in two formats meta plus and meta minus. When the design is tested between ranges X-N to X where X is average. And in Meta plus its run between X to X+N.

5.4 CHALLENGES OF HIGH SPEED IO

For a modern computer platforms post silicon testing in industries is becoming difficult task. And thus, release of any product become critical for high speed IO link. Thus, need for adoption of an architecture is important. There are number of digital changes needed to be consider for physical layer as well as at the protocol layer due to the need of high throughput and lower latency. Thus, highlight the challenges in both layer.

Physical Layer Test Challenges: when implemented a real system with mimic analog parameters problem using digital signal with PC traces, connectors. Then certain factor impacts the physical layer performance and factors are impedance mismatch and signal integrity. Signal integrity can be damage due to noise and inter symbol interference due to high data rates as at high data rate bit unit become smaller. For a physical layer testing oscilloscope are used but due to degrading signal integrity, performance at receiver device also degrades which creates negative impact on robustness and reliability.

Protocol Layer Test Challenges: physical layer issues are reflected at protocol layer itself. The jitter rate could impact the performance at higher data rate which impact again the reliability of product. And digital link reliability's critical for application like in defense. Thus, there is a need for protocol analyzer which helps in ensuring the functionality of the design. Thus, for multi links PCIe is the solution with the ability to define complex, serial packet-oriented triggers.

Addressing Test Challenges Solution for COTS:

- JTAG
- PCIe

5.4.1 Peripheral Component Interconnect Express (PCIe)

For having high speed communication with maximum throughput PCIe is the backbone for complex chips. Formerly it is known as 3GIO. PCIe is extensively used in different applications like graphic cards, computer cards, industrial and consumer applications. For verifying high speed interface liked PCIe verification methodology such as functional verification is required. As said it is a 3GIO means thirds generation high performance input/output and is extensively used to connect peripheral devices of application platform. Application like mobiles, laptop which has interconnect work employs PCIe [26].

They are also used to provide a connection between graphic card to main memory and CPU. And provide extension to attach to motherboard like extended graphic cards. Thus, PCIE provides high performance, maximum speed of data communication, and differential link. Over decade ago PCIe came at 33MHz, with a 32-bit bus and a peak theoretical bandwidth of 132MB/s which was good for old days. After this some

combinations were introduced 64-bit, 33MHz bus with bandwidth of 264MB/s; a more recent a combination of 64-bit, 66MHz c with a bandwidth of 512MB/s they are introduces as nowadays with complex system when demand for high speed and more bandwidth speed and width improved.

PCI vs PCIE: Shared bus topology for PCI which is an older version is such that every device connected can shared the bus. The bus used to communication with it all devices like sound card or a network cards are attached. The shared bus topology is shown below in Figure 5.9 how different device which wants to communicate with CPU are connected though same bus.

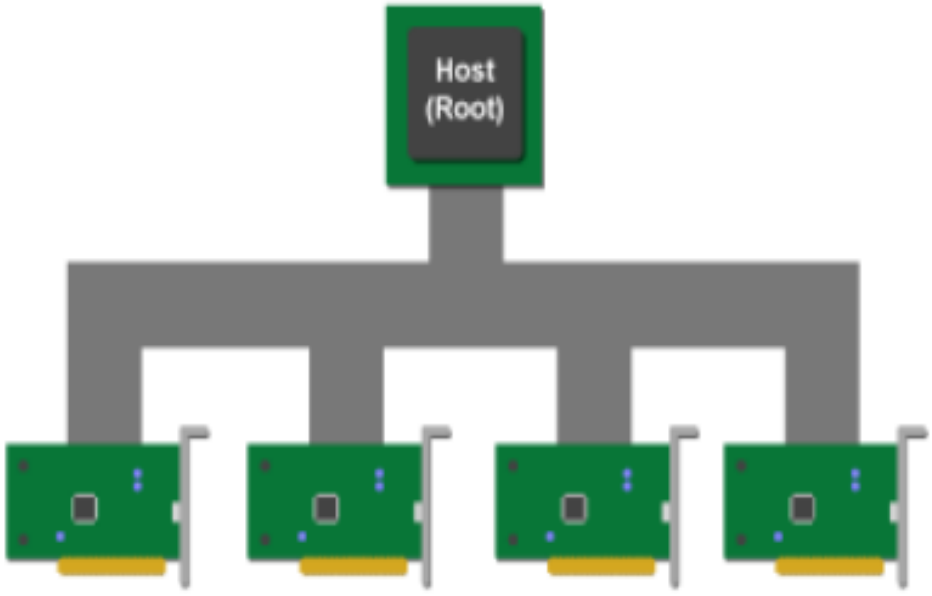


Figure 5.9 Shared bus topology of PCI [26]

This approach holds good when less number of devices are attached. But as number of devices connects to bus such this topology in not reliable thus moved to extended version which is known as PCIE which has more bandwidth and high speed.

5.4.1.1 PCIE: The Next Generation

PCIe is improved version of PCI because of its point to point topology. The figure below shows the topology of PCIe from this figure 5.10 can compare the difference of PCI and why moved to PCIe.

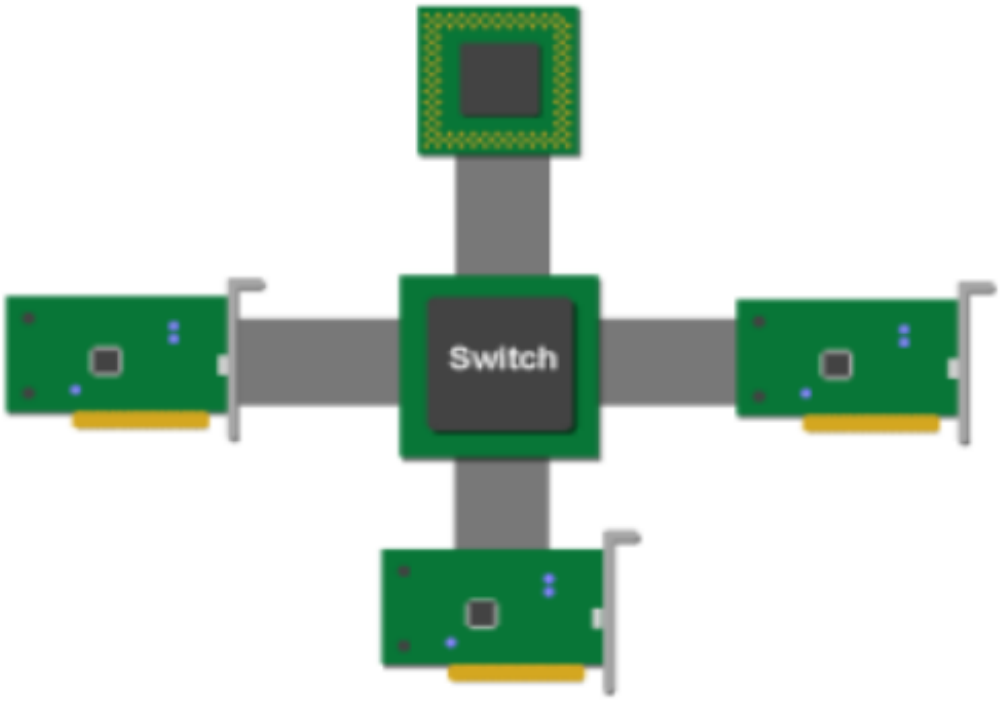


Figure 5.10 Point to point bus topology of PCIe [26]

In this topology each device has its own direct access to main switch. In this topology shared switch is replaced unlike in shared topology. In shared where device has to negotiate to who use switch in point to point they need not to do same. The main concept of point to point topology is that it controls traffic between devices thus making it resources sharing device. Also, quality of service is improved in PCIe [26].

5.4.1.2 PCIE protocol details

This section focusses more on protocol of PCIe that its layers and functionality of each layers.

1) PCIe LINK

Link is a connection between two components. Thus, it is a physical connection between the devices. PCIe links has signal in both directions. 1 link has one lane or differential signal for total of 4signals. Similarly, 32 link has 32 lanes for total of 128 signals. Figure 5.11 shows the link.

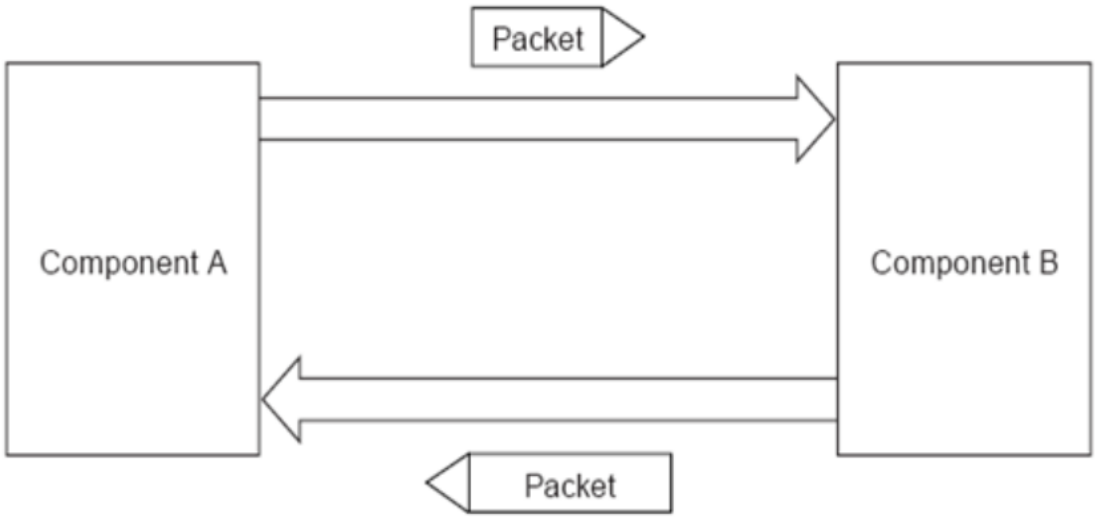


Figure 5.11 Block diagram of PCIe link [26]

2) PCIe Topology:

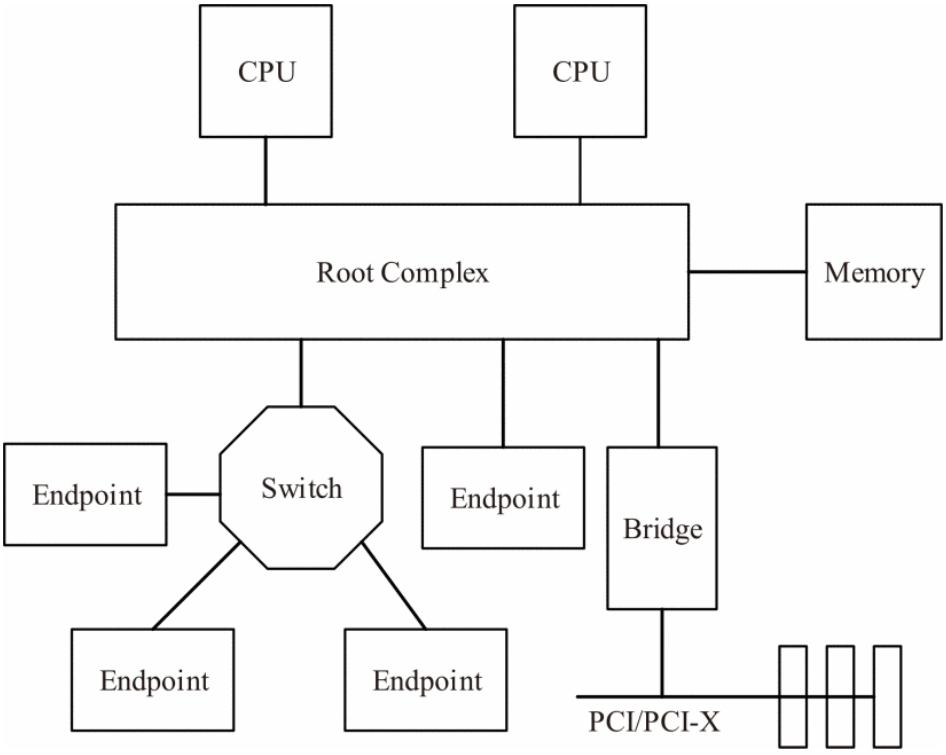


Figure 5.12 Block diagram of PCIe topology [27]

The PCIe topology shown in Figure 5.12 mainly has a root complex, endpoint, PCIx. Thus, look in each of it.

Root Complex: it connects the CPU or main memory to the input/output. Basically, it represents the root of input/output hierarchy.

Endpoint: endpoint can be a requester requesting for switch or a completer it could do it on its own behalf or on the behalf or some other device which is a non-PCIe. It could be a graphic card or a USB controller.

PCIe to PCIx Bridge: It is a connection between a PCIe and a PCI-X/PCI hierarchy.

3) PCIe layering:

PCIe is a three-layer architecture. Thus, for communication between any two devices follow this three-layering protocol. Three layers are transaction, data and physical layer. Layering of PCIe is shown below in Figure 5.13.

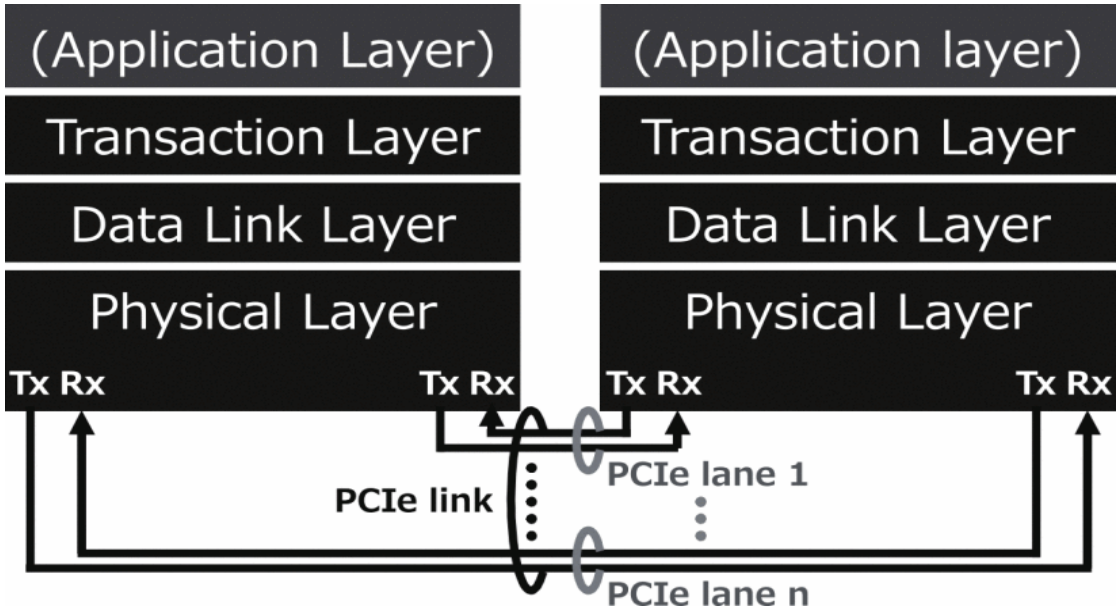


Figure 5.13 Diagram of PCIe layering [28]

Each layer is divided into two portions. One portion transmits the information and other portion receives it.

Transaction layer: packets which are foundation of data transfer between devices being created and requested by this layer. It functions for receiving and transmitting the data. Basically, on receiving

requesting data from the core it turns them into PCIe transaction. On receiving end receives data from data transaction layer. This layer relies on data layer.

Data Link layer: it is intermediate stage between physical and transaction layer. Its main purpose is to transfer data between two ends. It basically converts frames into bits and give them to physical layer. Error detection is also done at this layer [27].

Physical Layer: it is lowest layer of OS model and is responsible for sending bits from one compute to another. This layer is responsible for resource initialization and transmission or reception of signals.

CHAPTER 6

TEST IMPLEMENTATION

Verifying mixed signal IP and designing becoming complex. To avoid any re-spins or re-design cost it is very important to detect any form of bug whether in functional or in architectural bugs which can come early in design. Mixed signal IP which has both analog and digital block required both designer such that they can have confidence in their design before proceeding the design to design level. Also, they must have clarity of the design. For a mixed signal IP, the design both in RTL and Analog Front End depends on each other heavily. Thus, verifying interface correctness of analog and digital block is also a major concern while designing. In thesis basically present a method to design and verifying a Mixed signal IP. And tis verification is done using language System Verilog in VCS compiler. To explain the method, a design of 16Gbps transmitter (TX) which support serial PCIe protocol has been taken. For implementing features like equalization, output swing and power management interaction of analog and digital is way to do. The task to model all possible combination of settings and verifying them with minimum time has been taken. Those challenges have been addressed in this thesis.

6.1 HIGH SPEED PCIE CHANNEL

The purpose of post silicon validation is to qualify a product over all possible corners and then industry testing in physical platforms. But with the increased complexity, time to market need and customers performance specification created pressure for post silicon validation. Thus, a major portion of circuits goes to validate high speed IO (HSIO) links. Validation of these HSIO links is critical for product release qualification and should be taken care. Also providing best receiver circuit setting is important in HSIO link is time consuming process.

Thus, PCIe is the best HSIO interface. Thus, for customers' demands in modern industry PCIe is the solution as it is a packet based high speed point to point technology. The bandwidth is increased with multiple lanes i.e. x2, x4, x8, x16 and x32. And the rates also increased from 2.5Gbps for PCIe1 to 5Gbps PCIe2 and 8Gbps PCIe3 to 16Gbps for PCIe4. The end to end channel is shown below in Figure 6.1.

Figure 6.1 PCIe channel [29]

Due to speed increased at transmission channel effects like reflection and electromagnetic coupling are more leads to signal errors. Also, channel caused attenuation at high frequency generating distortion leads to inter symbol interference which creates signal undetectable at receiver end. Also, too close eyes diagram at receiver end with bit errors. Figure 6.2 shows the signal behavior from Tx to Rx when the frequency increases [30].

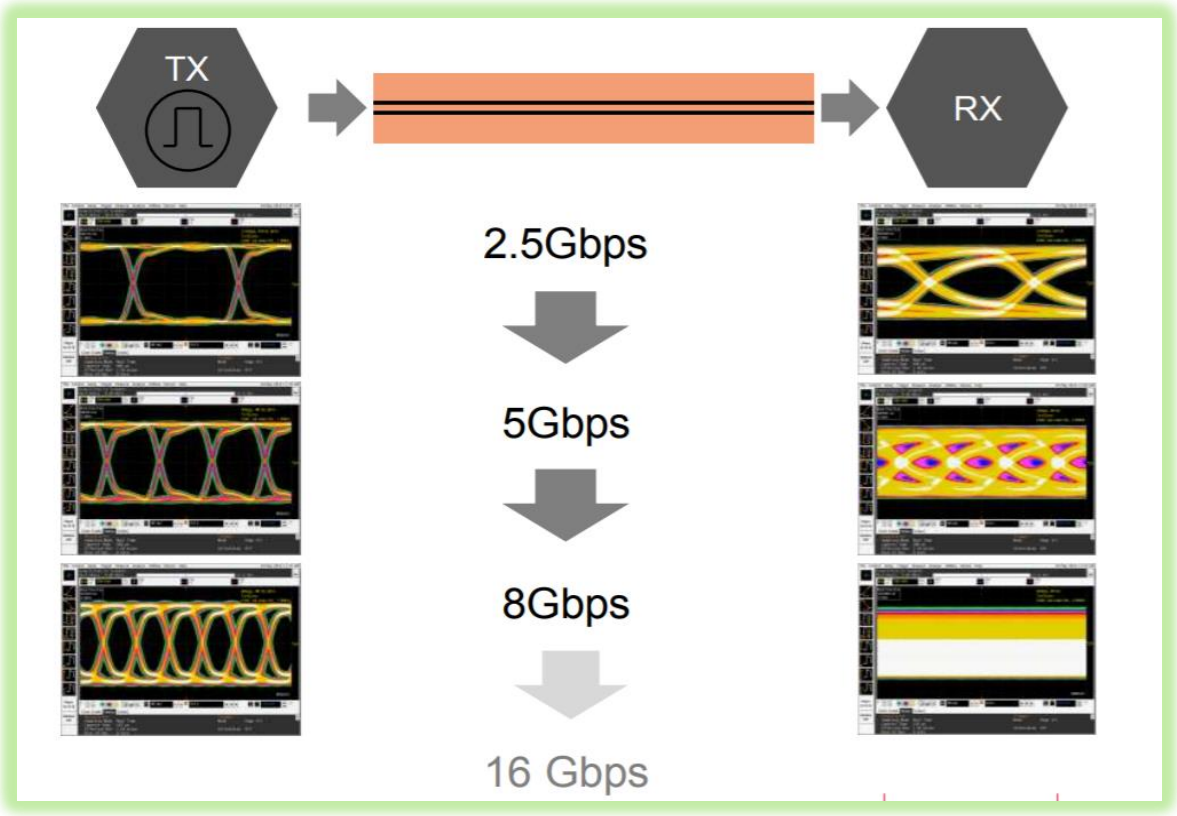


Figure 6.2 Signal behavior from TX to RX [30]

Thus, problem to this solution is to open the eye diagram and this is to be done before receiver samples data. Thus, equalization is the solution to compensate channel attenuation. Thus, to minimize bit error rate PCIe spec defines the requirement for equalization at TX or RX to remove the effect.

6.2 MODELLING THROUGH REAL VALUE

IP created manually have more chances of error and also debugging them is difficult task. Thus, a model where only some special cell in an AFE are modelled and from analog schematic a top-level hierarchy is created. By using this method can make sure that design exactly mimics the analog schematic. Thus, while modelling an IP System Verilog allows level of detailing which is required. Thus, to model behavior of analog block which is exactly closer to analog circuit is done using Real Value Modelling which is done through System Verilog. The hierarchy of project is shown below in Figure 6.3.

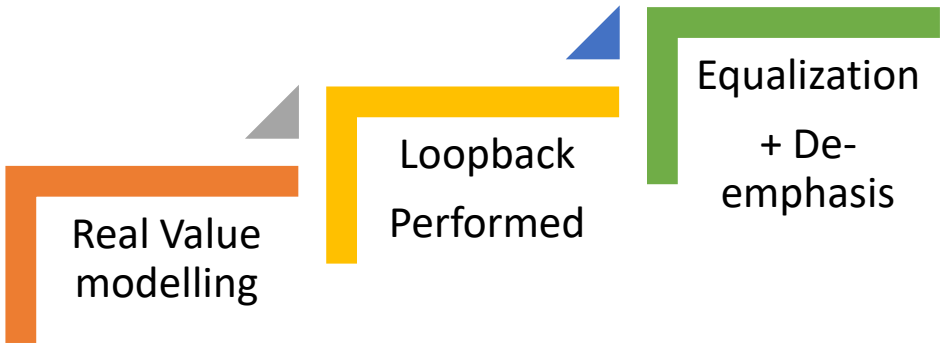


Figure 6.3 IP verification hierarchy

6.3 IDEAL SETUP FOR MIXED SIGNAL VERIFICATION

With the increased complexity verifying such mixed IPs are becoming tough and challenging task with including analog and digital dependencies. Thus, some inter domain feedback like loopback between analog and digital domain, trim settings and trimming of termination resistance of transmitter. In most cases not only verifying correctness of RTL design but also verifying interface of analog and digital block must be of concern. Thus, verification is not only to RTL but now it is to verify entire IP. Thus, there must be an ideal mixed signal setup to verify and observe analog inputs such as pad voltages, power domain, supply requirements etc.

Traditionally these requirements are verified through analog and mixed signal (AMS) simulation. But they take time to run such that it is not possible to cover all aspects like coverage including all cases in less time. Also, a golden standard is SPICE simulation which is used but they take again lot time could lead from

week to month which is critical for a time to market. Thus, to reach reasonable simulation speed analog model can be used with RTL. The advantage if it is that it creates digital environment means near digital simulation for an entire analog circuit.

Thus, Real value modelling of analog circuit using System Verilog and providing simulation in VCS compiler provides high speed and accuracy. But how much the model is efficient is depend on analog model quality which goes into simulation with RTL.

Analog and digital block are modelled parallel in a typical mixed signal IP design. Analog blocks are handwritten in initial stages and is verified in usually RTL flow. But this method is not accurate. Thus, one cannot guarantee that this model is bug free. After designing analog block are verified using AMS but they don't give much confidence to designer such that it can freeze the model. Thus, AFE models needs to be available in early design cycle and has to contain all enough detail required. Thus, ideally it is important to have full visibility of AFE signals and if error comes can verify the blocks. But this method is hard to maintain. Thus, proposed a method to directly create an AFE hierarchy from AFE schematics. This makes the full visibility of signal but creating an exact model of real analog circuit which makes it easier to debug.

6.4 VERIFICATION FOR LOOPBACK

Loopback mode in physical layer allow verification independent of post-silicon validation and also characterization of physical layer without the need of help from higher layers.

In loopback mode two PHY can be connected as master and slave or transmitter of PHY can be connected to its receiver. Transmitter send out a set of pattern sequence and compare this signal received on receiver side. On comparison if PHY is working properly it will pass otherwise it will fail. And thus, covers critical circuit and logic within PHY.

The loopback is performed to ensure the signal integrity as work on transmitter but to know whether the received signal is correct at receiver without channel is done through loopback. PCIe has this feature of performing loopback. Basically, in this mode actually connect the transmit lanes of PCIe to receiver lane of PCIe which is output of transmitter is connected to input of transmitter. Thus, signal sent from our PHY is received by PHY itself. The model of loopback is shown below in Figure 6.4.

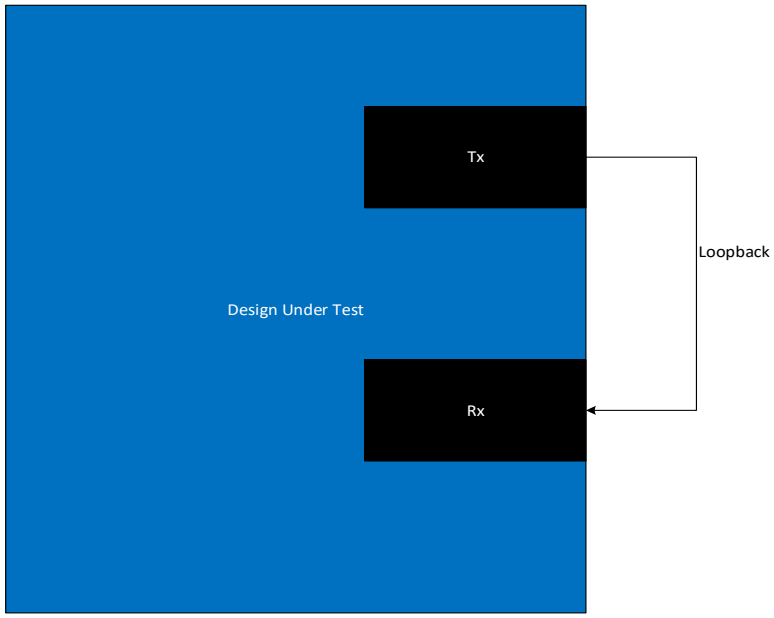


Figure 6.4 Loopback

If the pad at lanes are getting same value means received vale is correct. Loopback is of different type it can be analog loopback or digital loopback. The results with observation is described in next section. Loopback is of four types,

1. Digital near end loopback.
2. Analog near end loopback.
3. Digital far end loopback.
4. Analog far end loopback.

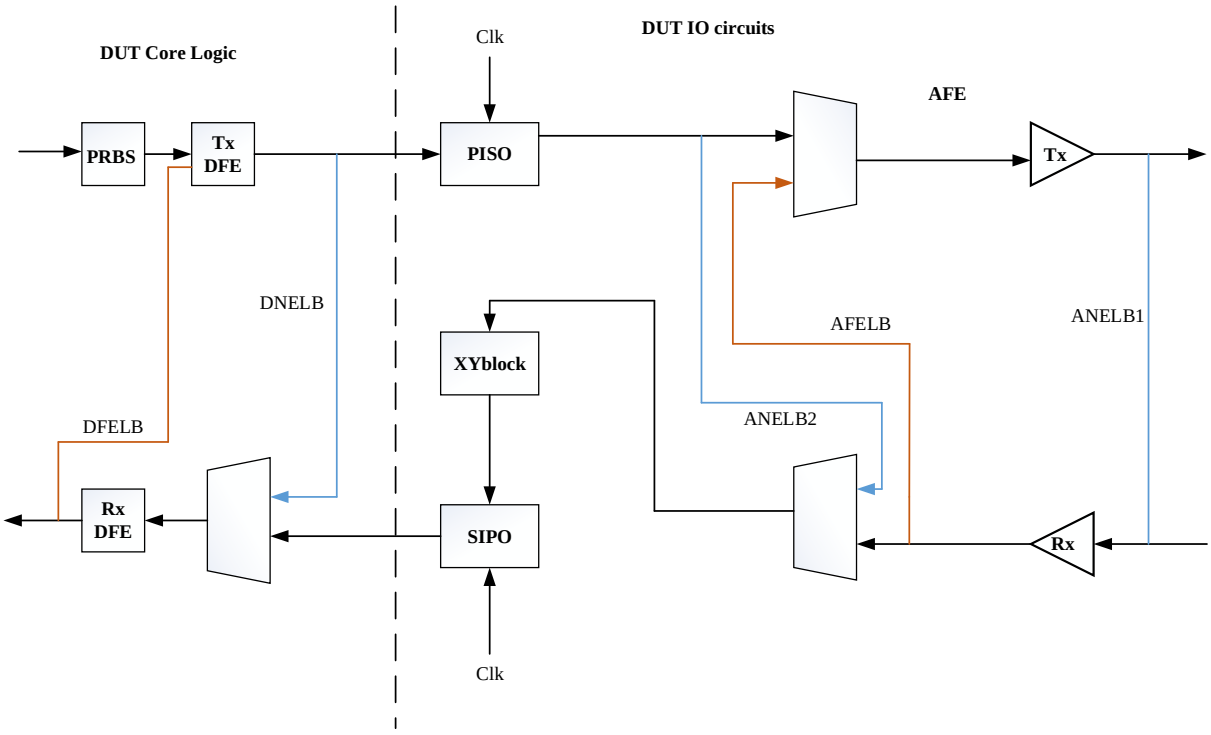


Figure 6.5 Types of Loopback Mode

Figure 6.5 above shows the loopback types. Digital near end loopback is when loopback is performed within the in digital part where near end loopback is when loopback is performed within the lane. Similarly, analog near end loopback is that is performed in analog part within the lane. Means both transmitter data and clock are to will be loopback to receiver path. And both RX and TX block will be running on same clock generated by PLL. Far end loopback is performed between different lanes.

6.5 OVERVIEW OF EQUALIZATION TECHNIQUE

Equalization is the process of distorting signal and this is done with transformation which is the reverse of the channel. Equalization can be applied both at tram sitter and at receiver side. As in PCIe1 and PCIEx2 form of equalization is de-emphasis at transmitter where data is sent at full swing after each polarity transition and done to decrease the swing for bits which is having same polarity [31]. At high frequency channel experience greater loss so to reduce this loss equalization is done. Equalization can be done using two algorithm one is linear equalization done at transmitter or receiver side other is decision feedback which is done at receiver side. Equalization is refer as adjusting Tap coefficient. Every channel, Tx and Rx has some coefficient combination for having optimum signal to noise ratio. The need for equalization with the importance of equalization is shown in Figure 6.6 depicts Tx output with no equalization and other with equalization such that if don't do it the signal at Rx will be unreadable.

Figure 6.6 No equalization [31]

Thus, equalization restores the information either by boosting high frequency or by attenuating low frequency. Thus, it is done to remove the effect of inter symbol interference hence to remove the effect of error rate. Thus, a signal is passed through a filter having its own frequency and inverse of channel frequency. The behavior of signal from transmitter to receiver is shown below in Figure 6.7.

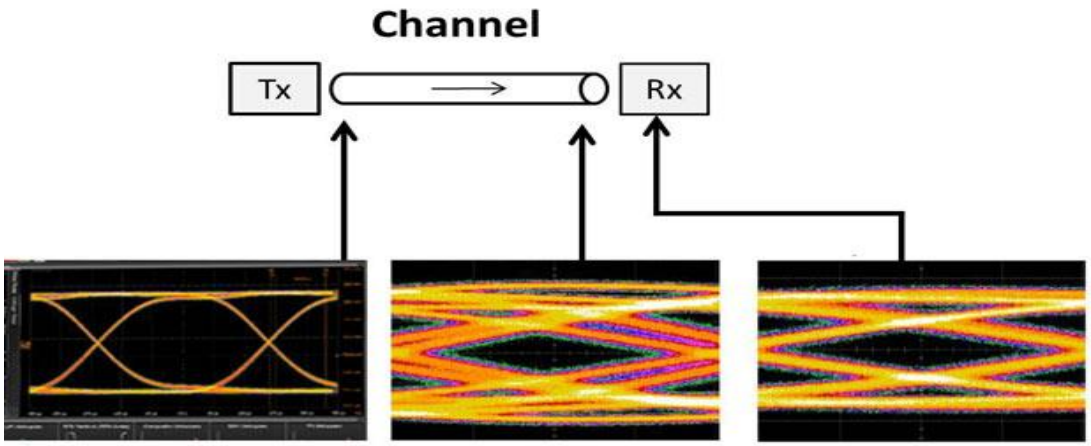


Figure 6.7 Behavior of signal from Tx to Rx with equalization [31]

As mention equalization can be done at either side at Tx or Rx. But at receiver side the coming signal could mix with channel noise which can degrade signal which is unreadable and even signal to noise ratio degrades more and leads to noise enhancement [32]. At transmitter equalization is free from noise enhancement. And making it more effective at transmitter side.

6.5.1 PCIe Equalization

PCIe gives 16Gbps bit rate but due to bandwidth limited channel at high frequency signal is distorted thus to remove the effect of inter symbol interference and noise impairment. Thus, PCIe specification provides the provision of performing equalization at the trimester side. Equalization at transmitter is based on 3-tap

filter as shown below in Figure 6.8 which is a 3-tap finite impulse response. It has three filter coefficient C0, C_m and C_p.

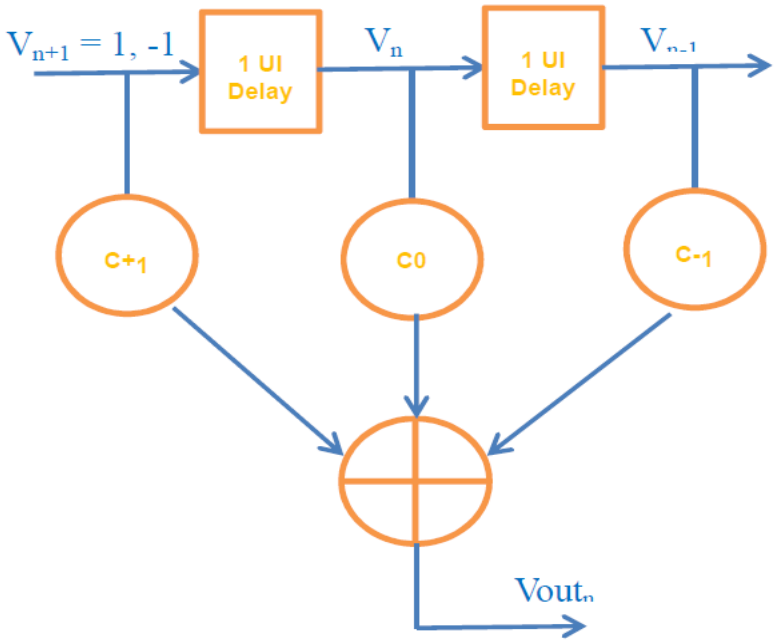


Figure 6.8 3-tap Filter [32]

Three consecutive pulse are multiplied with three different filter coefficients and then multiplier outputs are added together to produce the final filter output.

The 3-tap FIR filter output is given following eq.

$$\mathrm{Vout} = \mathrm{Vnm} * \mathrm{Cm} + \mathrm{Vn} * \mathrm{C0} + \mathrm{Vnp} * \mathrm{Cp} \ldots\ldots (1)$$

Where C_m, C0 and C_p are pre-cursor, cursor and post-cursor coefficient respectively and these pre and post coefficients are for filter work on delay and advance signal with respect to time. [32] Through this filter, the serial data input signal is delay by flip flop which are used to implement the taps. The filter response can be control by varying coefficient values.

6.5.2Transmitter Equalization Coefficient Matrix

With the protocol constraint the values of transmitter coefficient:

$$|\mathrm{Cm}| + |\mathrm{C0}| + |\mathrm{Cp}| = 1 \text{ subject to } \mathrm{C0} > 0, \mathrm{Cm} < 0 \text{ and } \mathrm{Cp} < 0 \ldots\ldots (2)$$

These constraints are obtained by determining only C_m and C_p to Vout from equation (1) and C0 applied to equation (2). Also, coefficient range and tolerance are constraint by specification given by the team.

FS is the full swing refer as the maximum differential voltages and is given as [32]

$$FS = |Cm| + |C0| + |Cp| \ldots\ldots\ldots (3)$$

Another parameter is low frequency (LF) which is flat voltage and it should be always greater than differential voltage.

$$|Cm| - |C0| - |Cp| \geq LF \ldots\ldots\ldots (4)$$

6.5.3 Verification of 3-TAP FFE for Transmitter

For transmitter IP only, model logic gates and analog block resides in the design. After modelling which is done in detail check them by simulating in minimum and maximum ranges in VCS compiler. Using Real Value Modelling pad of transmitter are modelled to get the analog voltage and output impedance based on the signals which are received from the RTL. From this can conclude that digital designer can get the exact behavior of circuit using actual values happening during equalization transmission. The settings are provided from spec as a reference from PCIe spec [31] requirements are shown below in Table 6.5.

Table 6.5 Coefficient's for Pre and Post Cursor Reference from PCIe 3 Spec [29]

Preshoot (dB)	**De-Emphasis (dB)**	**Pre_bits_DeEmphasis (V)**	**Post_bits_DeEmphasis (V)**
0.0	0.0	1.0	1.0
0.0	-3.5 ± 1	0.66	0.66
3.5 ± 1	-3.5 ± 1	0.75	0.75

Figure 6.9 shows equalization example not intending signal appear for measurement. The pre-cursor is referred to as pre-shoot and post-cursor is refer to as de-emphasis. In 2.5Gbps and 5.0Gbps only de-emphasis is specified hold for 8Gbps and so on and pre-shoot and de-emphasis both are independent.

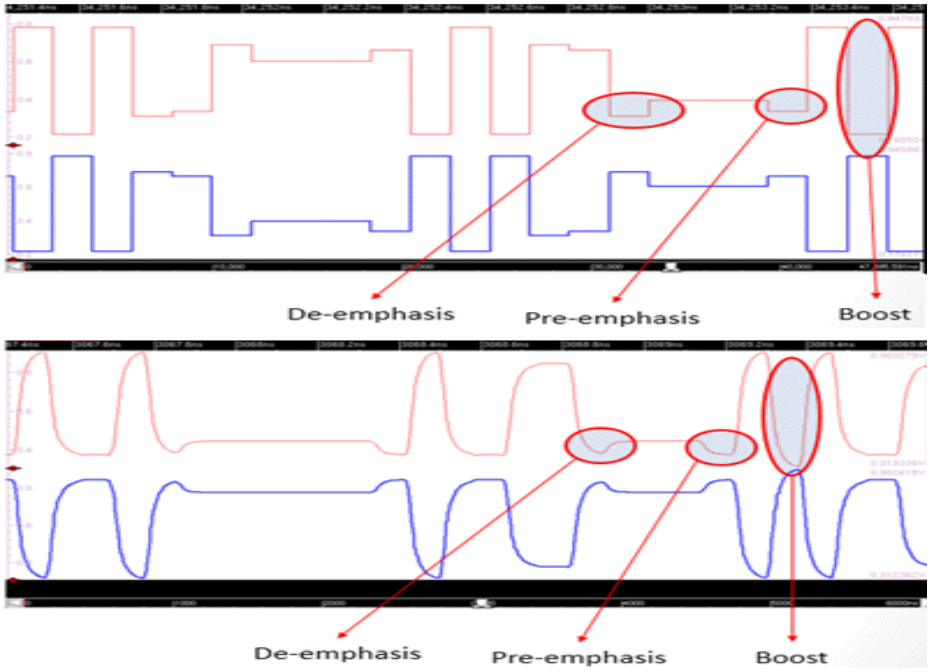


Figure 6.9 Pre-shoot, Boost and De-emphasis [32]

Flowchart for equalization setting for full swing

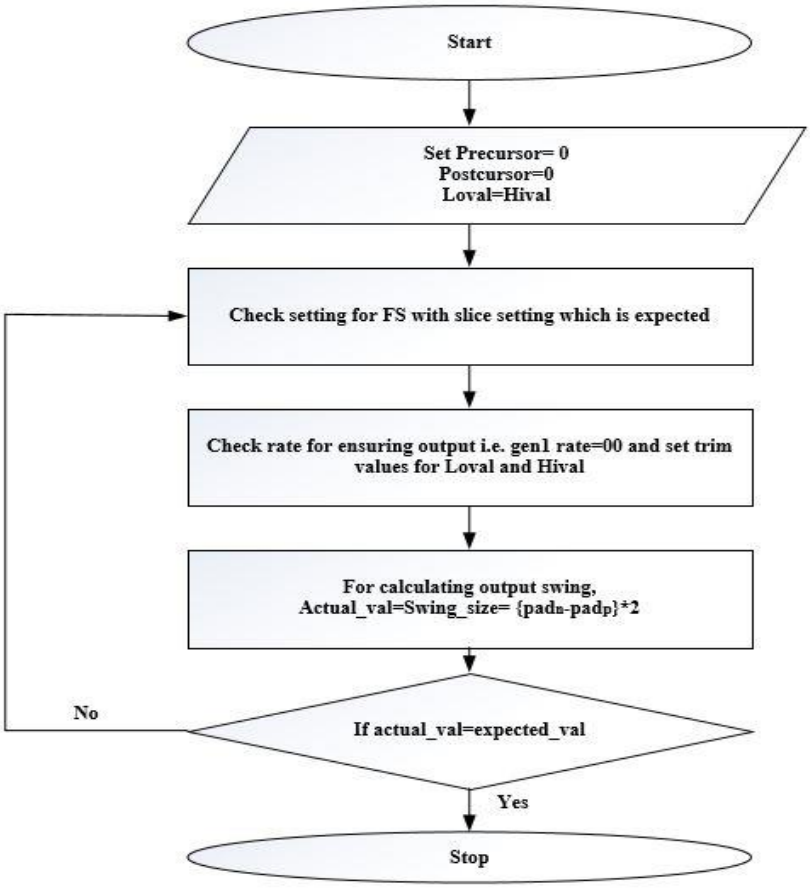


Figure 6.10 Flow chart for equalization

6.5.3.1 De-emphasis

When multiple bits of same polarity are being output then to remove the effect of inter symbol interference de-emphasis is performed. Individual bit is driven by full voltage level whereas multiple bits are driven by differential voltage level. An example of de-emphasis is shown in Figure 6.10 with showing -3.5 dB de-emphasis around a common mode of 0.5 V.

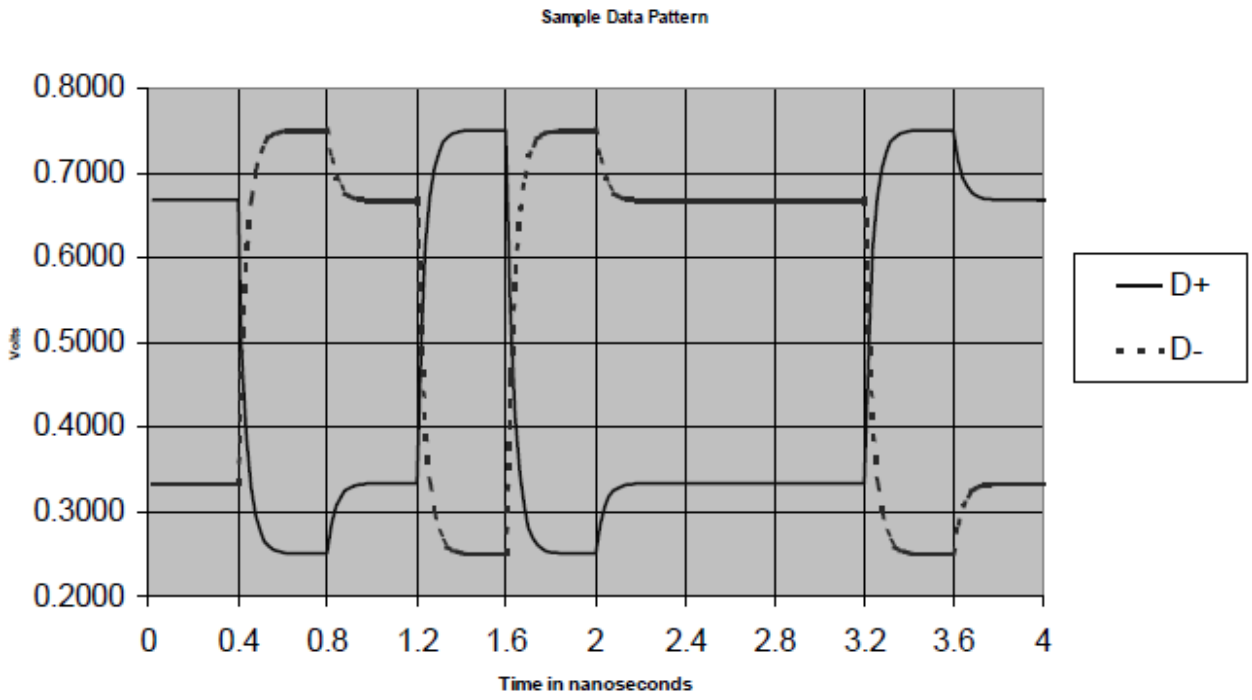


Figure 6.11 De-emphasis [31]

6.5.3.2 Regression

Regression are run to check the all possible cases of test that is covering all corners if any cases is creating bug or not thus the flow for regression is shown below in Figure 6.11 and the output for same is shown in coming section.

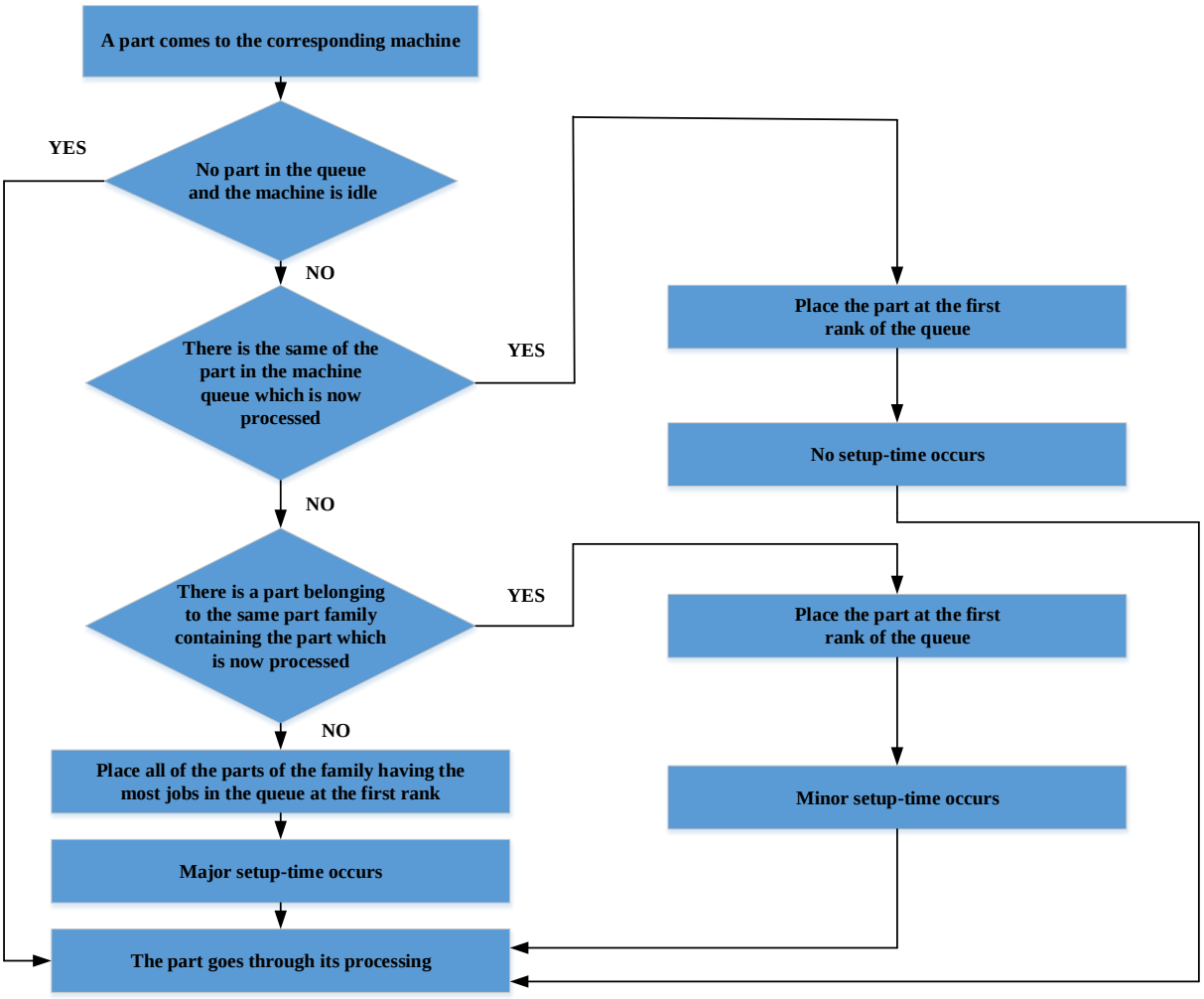


Figure 6.12 Regression flow

CHAPTER 7

RESULT AND DISCUSSION

7.1 MEASUREMENT METHODS

Measurement has been done in two ways. One with full swing and equalization **VTX-FS-EQ** and other one with half swing with equalization **VTX-HS-EQ.** The tolerance for transmitter equalization for 2.5 and 5.0Gbps signal has been relaxed from ±0.5 dB to ±1.0 dB.

7.1.1 VTX-FS-EQ

Full swing with equalization has been done by making C_m and C_p values equal to expected values. It has been needed to set trim to get exact values for performing equalization with specified tolerance and if don't work in those specify ranges the output won't come as expected. Unexpected values will come out of tolerance range. The output with undesired trim has been observed to get values at pad out of order which has been shown below in Figure 7.1.

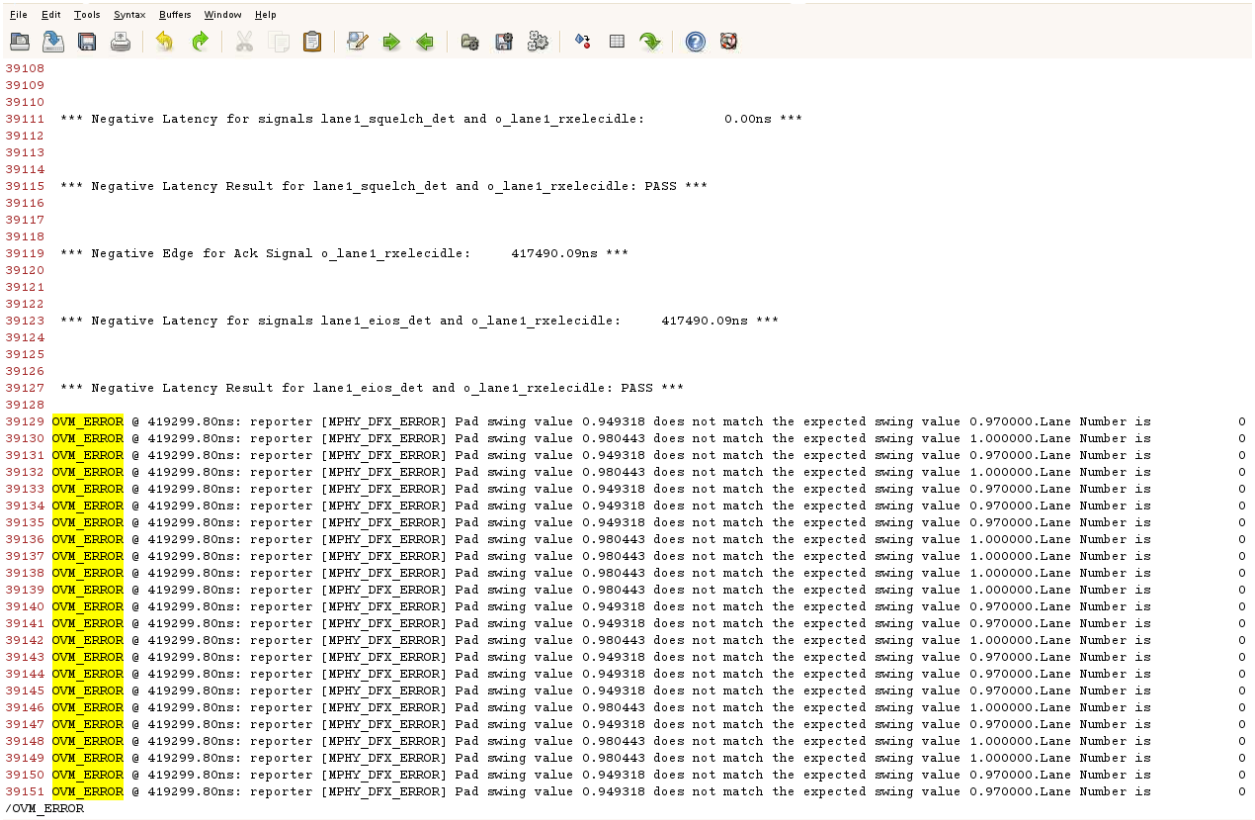


Figure 7.1 Value Observe at output pad with fail case

Another problem could occur with digital signal that have been used to compare analog signal in terms of digital signal through checker. In that checker actual specification of the limits of swing and how much they have been relaxed with specified tolerance given. If digital signal does not propagate properly then output won't come as checker won't work. The waveform with error has been shown in Figure 7.2.

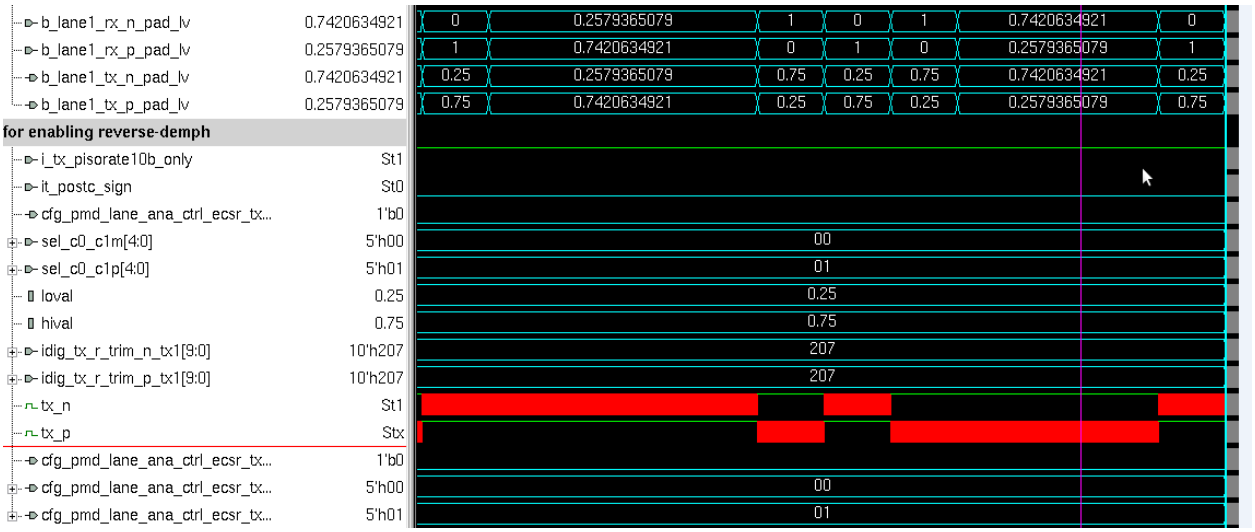


Figure 7.2 Signal at digital pad with high-z value

On debugging it has been known that signal was not properly propagated due to some buffer files from TAP need modification as JTAG have been not able to translate proper value from its input to output thus, on performing correction, the rectified output has been shown below in Figure 7.3.

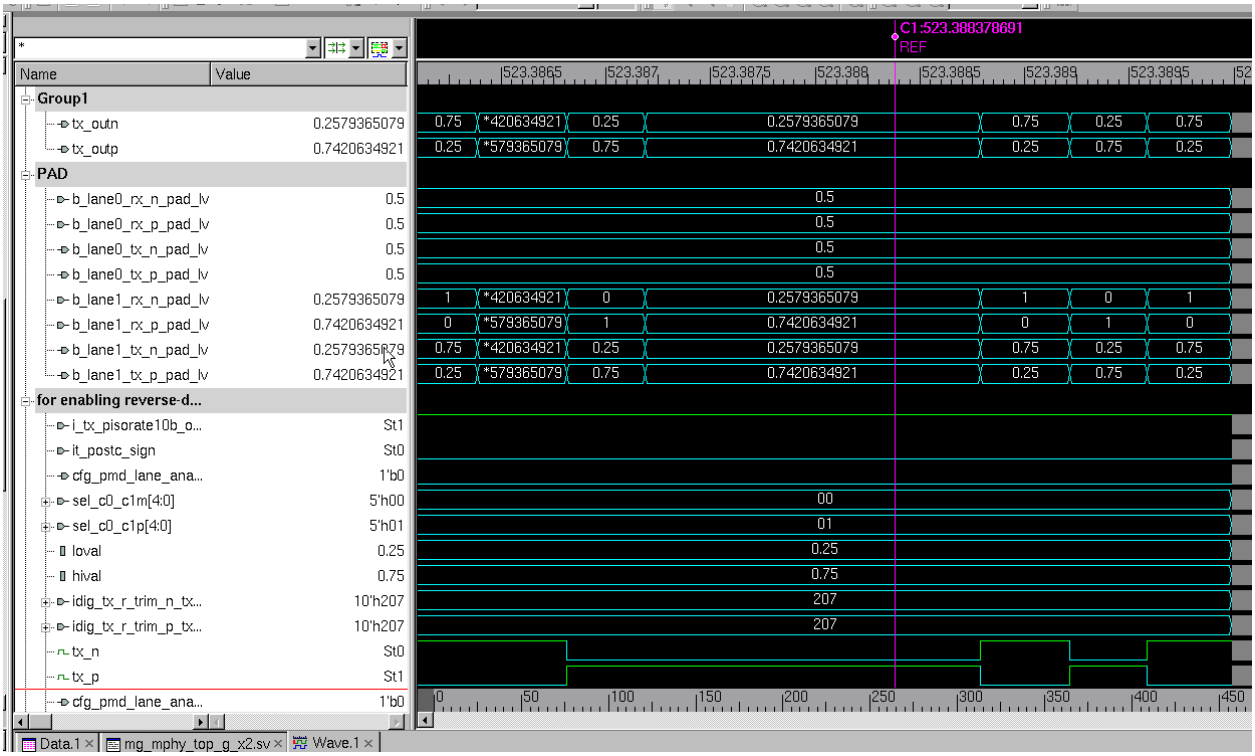


Figure 7.3 Corrected value at digital pad on modification

Though from above Figure 7.3 it has been seen that improper values on pads means pads not getting the good values. Thus, for the same by updated trim value as the environment for project has not able to propagate proper trim with this setting to get proper value which can be check in above figure where with no loopback propagation of signal on pads has not coming proper.

Loopback performed: In loopback output should be same at pad coming from transmitter of PHY and receiver pin of same PHY. The values at pad on performing loopback has been shown in Figure 7.4.

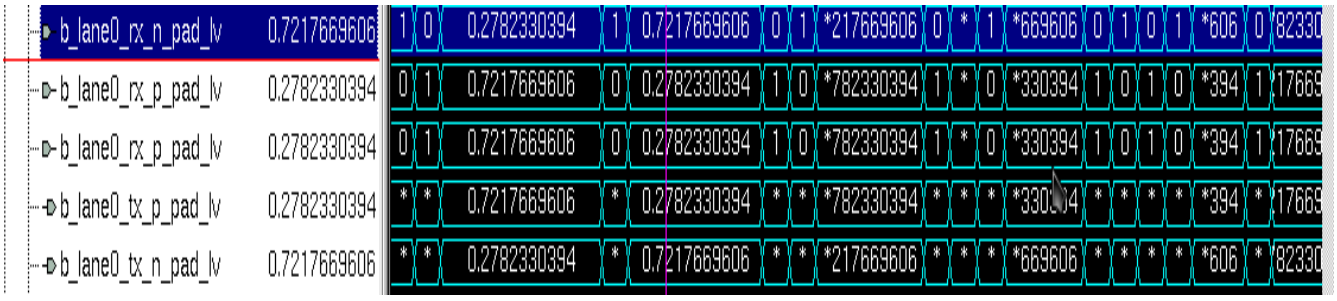


Figure 7.4 Output at pad with loopback performed

Equalization performed

Figure 7.5 below has shown the equalization for post cursor value showing OVM error zero with no fatal error in the notification window.

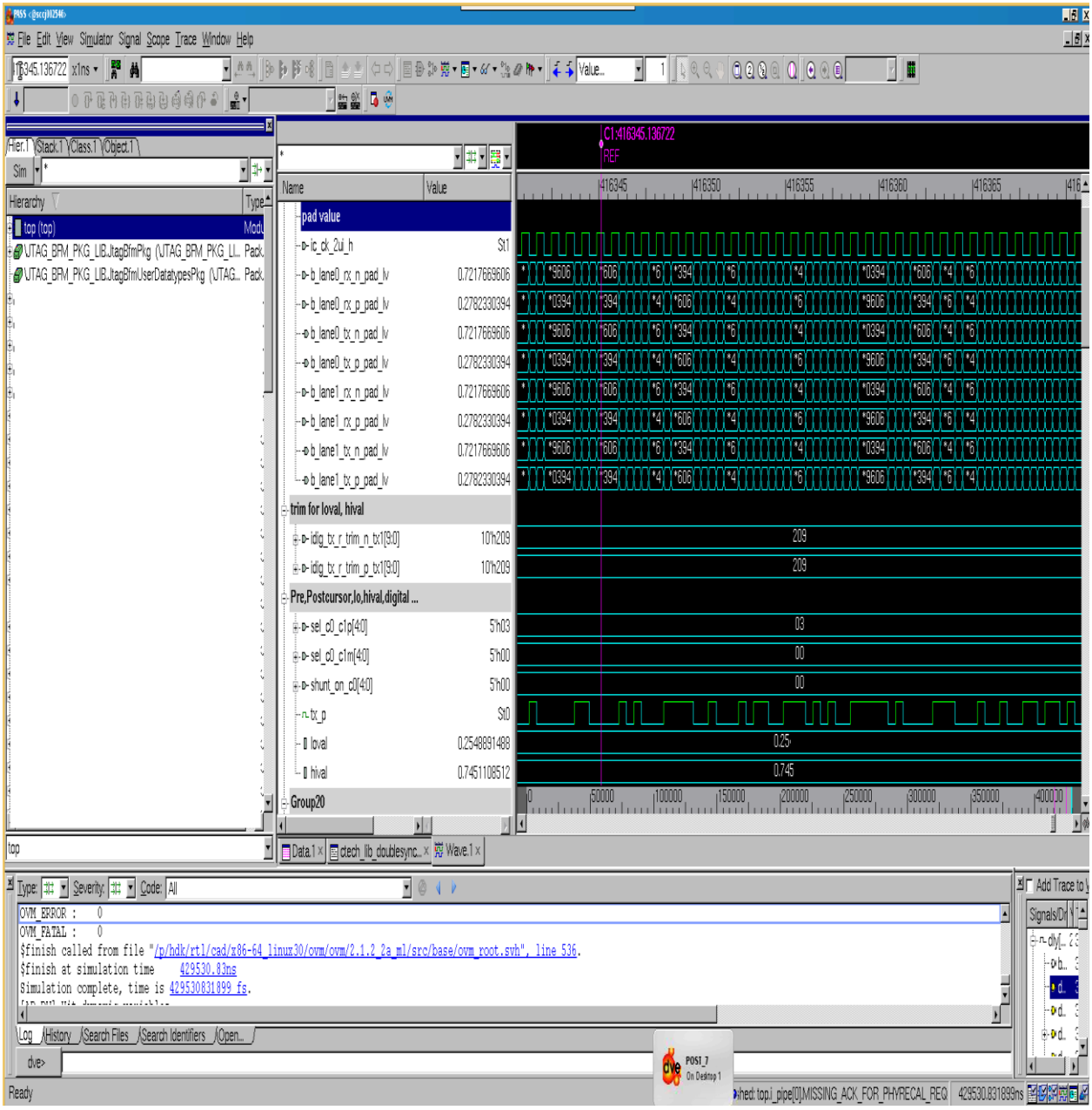


Figure 7.5 Equalization performed

Figure 7.6 has shown post cursor with slice 08 and value expecting on pad is 0.75 also pad on Tx and Rx has the same value due to loopback performed.

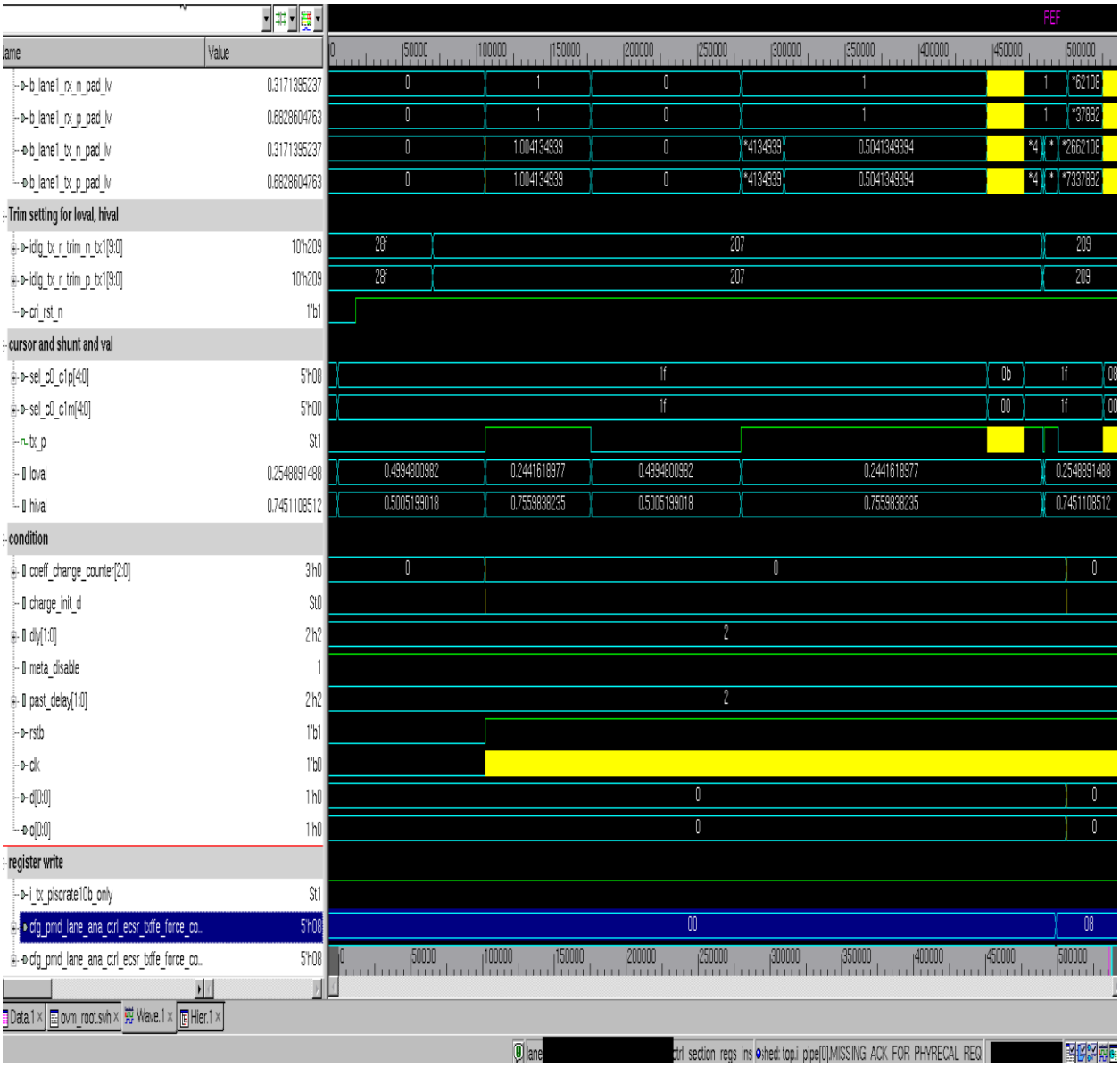


Figure 7.6 Equalization with post cursor

OVM stages: the waveform has shown the OVM stages in the log file when the test started it calls the initialize bring up from there the proposed test starts as shown in figure above and from that bring up the test actual start according to the need of test. The stages of OVM has been shown in figure below.

```
OVM_INFO /p/hdk/rtl/ip_releases/:     ↓/CHASSIS_JTAGBFM_2015WW41_R2.4_PIC1/verif/tb/JtagBfm/JtagBfmDriver.sv                    ovm_test_top.m_tb_env.mphy.jtag.PriMasterAgent.Driver [JtagBfmDriver]
Exiting GOTO Task:
Current State: RUTI
OVM_INFO /p/hdk/rtl/ip_releases/          mphytb_y17ww26.3/mphytb/verif/tests/test_base/test_base.svh                    ovm_test_top [mphy_dfx_tx_eqn_ctrl_test] ** HEARTBEAT **
The total objection count is 5
---------------------------------------------------------
Source  Total
Count   Count   Object
---------------------------------------------------------
0       5       ovm_top
0       5         ovm_test_top
0       5           m_tb_env
0       5             m_tb_env_sequencer
1       1               dlane_seq_0
1       1               dlane_seq_1
1       1               dlane_seq_2
1       1               dlane_seq_3
1       1               vseq
---------------------------------------------------------
```

Figure 7.7 OVM stages in equalization

Output log files: when tests run properly on full propagation of signals at the output this is how the output comes and which can be checked in the respective log files. The waveform for the log has been shown below.

```
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.980443 matches the expected swing value 1.000000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
OVM_INFO @ 721554.9  :: reporter [MPHY_DFX_PASS] Pad swing value 0.731442 matches the expected swing value 0.750000.Lane Number is    0
```

Figure 7.8 Output at pad for post-cursor value 0.75

Reverse de-emphasis

For performing reverse de-emphasis, bit has been needed to set to make de-emphasis in effect if that bit has not set that test won’t come in effect. Trim setting has to be done. Run de-emphasis only for post-cursor values. The waveform for the post cursor value has been shown below Figure 7.9.

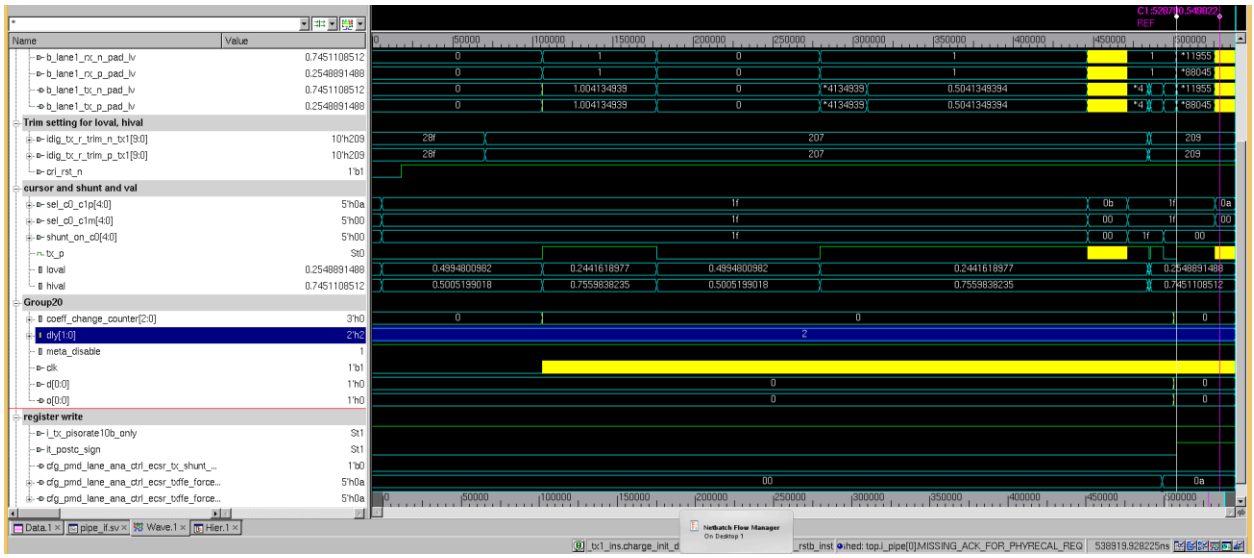


Figure 7.9 Post-cursor with reverse De-emphasis

Figure 7.10 has shown fitted from of above with showing reverse de-emphasis bit to set at high and when it gets high at that time values comes on pad with specified value on post cursor with pre-cursor to be zero also shunt values should remain zero at the same time.

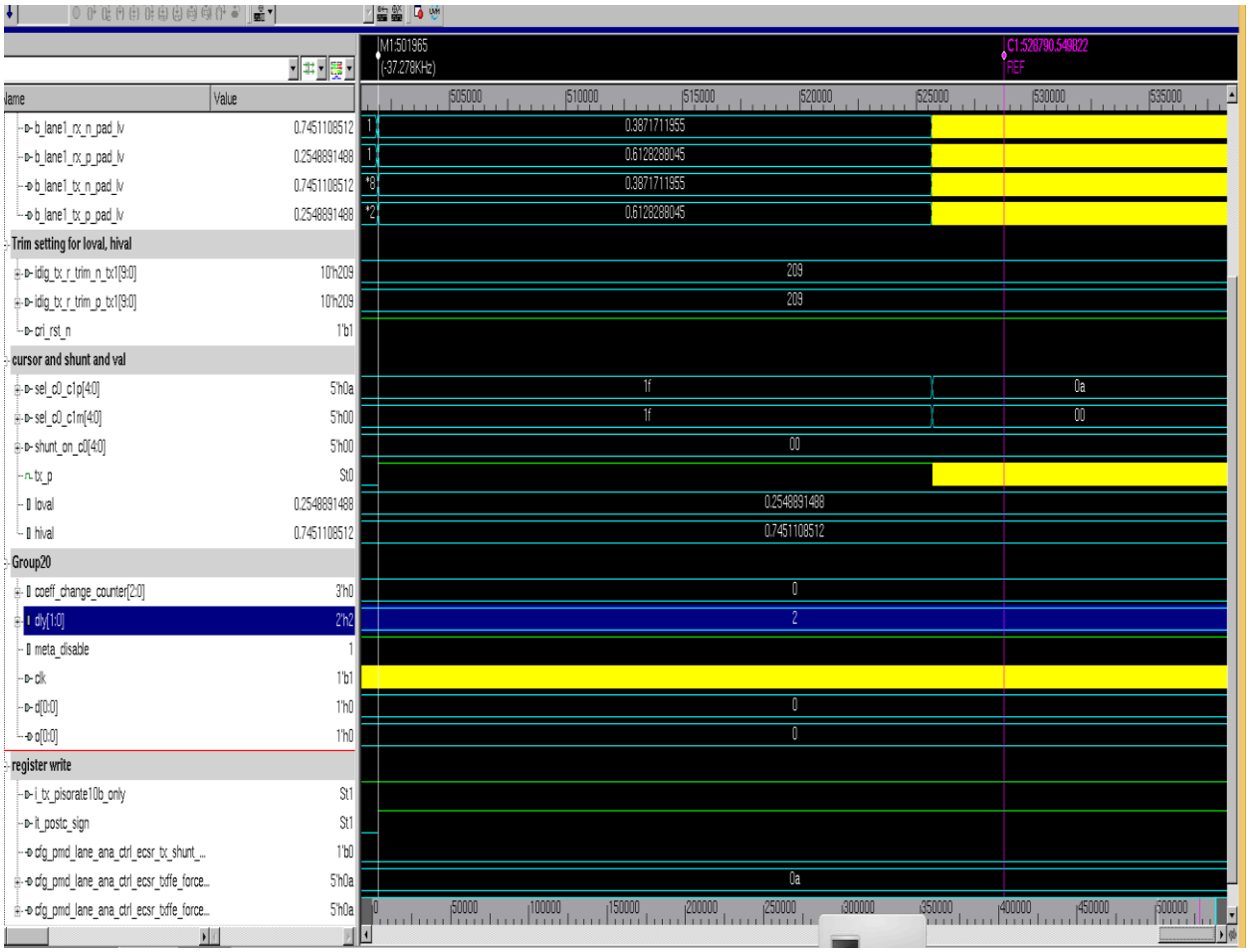


Figure 7.10 Post-cursor with reverse De-emphasis showing meta state disable

7.1.2 VTX-HS-EQ

For half swing with equalization has been done only to check the wave with global variables that has been performing equalization by making C_m and C_p values equal to zero. Here a shunt signal comes which has been modified to get the output according half swing setting given by designer. This has resulted into run test with half swing with min and max ranges. If the output has come between those ranges than test will pass with no bug. The output with undesired trim has been observed to have values at pad out of order which has been shown below.

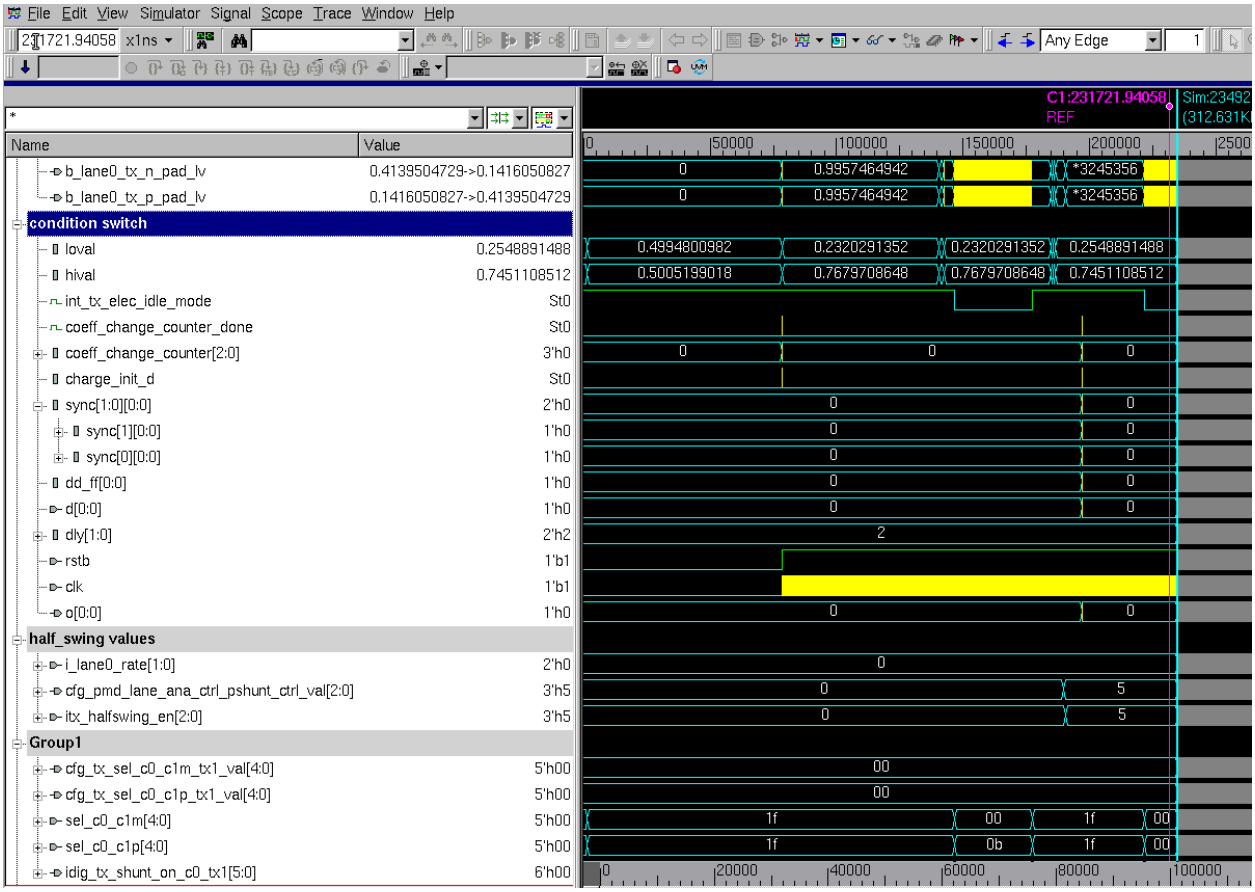


Figure 7.11 Pshunt with half swing enable

7.2 Verifying tests through regression

The regression has been launched to cover all the corners the result with fail and pass scenario as shown below. The test in initial stage has been launched to check at what stage they are failing after that modifications have been planned to performed. Figure 7.12 has shown regression fail case below.

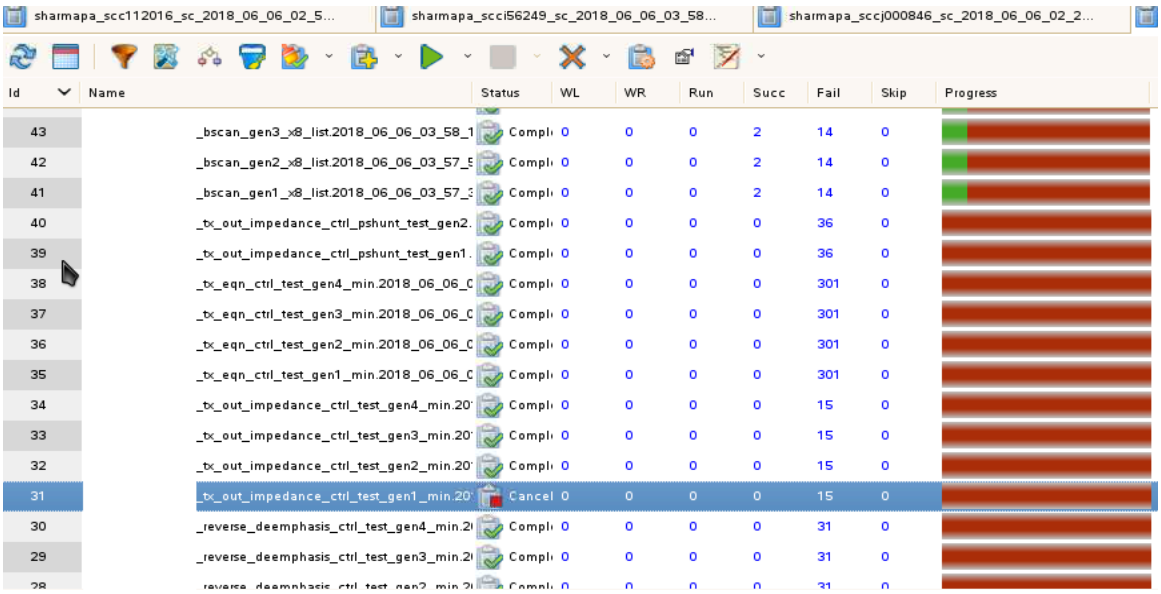


Figure 7.12 Regression with fail case

When write a test bench either check the test by running iterative in two or more seeds or directly launch regression to see if all cases are failing. After modification the and addition in test though test starts to pass but they stayed fail due to again some missing and bugs occurring in different cases. As Figure 7.13 has shown below, one can see equalization has been passing for some seeds and failing for others. Thus those seeds which ran interactively have been checked and modified.

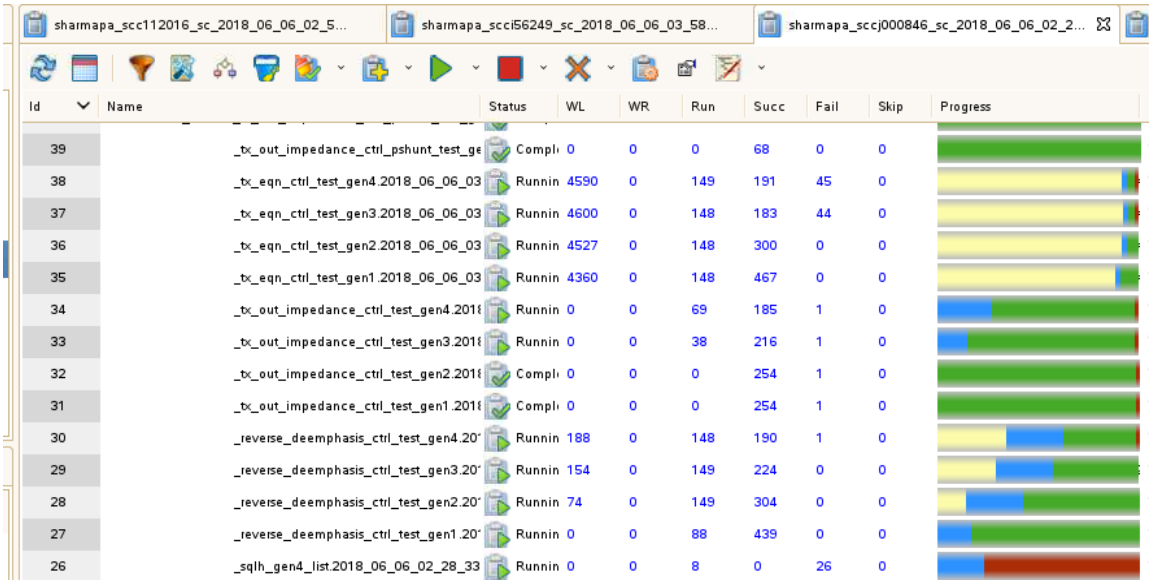
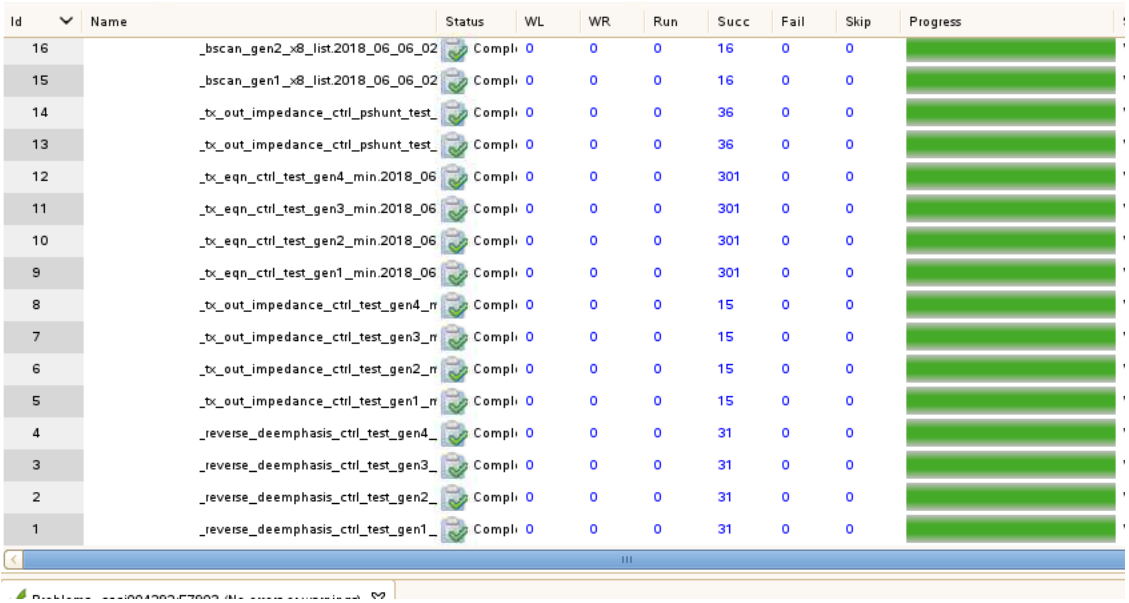


Figure 7.13 Regression with some stages with pass and fail case

At final stage all modifications have been included and model verified all tests have been passed as shown below in Figure 7.14. The design with 100% pass rate has been given to post work.



Id	Name	Status	WL	WR	Run	Succ	Fail	Skip	Progress
16	_bscan_gen2_x8_list.2018_06_06_02	Compl	0	0	0	16	0	0	
15	_bscan_gen1_x8_list.2018_06_06_02	Compl	0	0	0	16	0	0	
14	_tx_out_impedance_ctrl_pshunt_test_	Compl	0	0	0	36	0	0	
13	_tx_out_impedance_ctrl_pshunt_test_	Compl	0	0	0	36	0	0	
12	_tx_eqn_ctrl_test_gen4_min.2018_06	Compl	0	0	0	301	0	0	
11	_tx_eqn_ctrl_test_gen3_min.2018_06	Compl	0	0	0	301	0	0	
10	_tx_eqn_ctrl_test_gen2_min.2018_06	Compl	0	0	0	301	0	0	
9	_tx_eqn_ctrl_test_gen1_min.2018_06	Compl	0	0	0	301	0	0	
8	_tx_out_impedance_ctrl_test_gen4_m	Compl	0	0	0	15	0	0	
7	_tx_out_impedance_ctrl_test_gen3_m	Compl	0	0	0	15	0	0	
6	_tx_out_impedance_ctrl_test_gen2_m	Compl	0	0	0	15	0	0	
5	_tx_out_impedance_ctrl_test_gen1_m	Compl	0	0	0	15	0	0	
4	_reverse_deemphasis_ctrl_test_gen4_	Compl	0	0	0	31	0	0	
3	_reverse_deemphasis_ctrl_test_gen3_	Compl	0	0	0	31	0	0	
2	_reverse_deemphasis_ctrl_test_gen2_	Compl	0	0	0	31	0	0	
1	_reverse_deemphasis_ctrl_test_gen1_	Compl	0	0	0	31	0	0	

Figure 7.14 Regression with pass case

CHAPTER 8

CONCLUSION AND SCOPE OF FUTURE WORK

8.1 CONCLUSION

Verifying mixed signal SoC's is a difficult task because of interaction between analog and digital circuits. Traditional verification of mixed signal IPs takes millions of test vectors and each of them require millisecond. Simulator like SPICE and VCS for verifying analog and digital circuits respectively cannot directly used for mixed signal verification as shown in Chapter 3. From the various solutions, behavioral modelling is the one to be used and has chance of success. It provides the solution for side stepping analog solver which slows the entire system simulation. Also, this modelling has in built mathematical operators that can aid in emulation of analog circuits.

The proposed method uses the real value stimulus generation for designing the analog circuits. The verification environment is built by using OVM methodology. The result testbench provides the real value inputs in transaction level. Thus, generating transaction in an OVM environment which validated the verification process. Using this approach has one benefit of verification of IPs to an early stage in design cycle. Proposed methodology has been successfully applied for verifying 16Gbps PCIe transmitter IP.

8.2 FUTURE WORK

assertion-based technique could be used as assertion-based verification which will be useful for debugging analog and mixed signal circuits. From waveform inspection it could possible to skip some bug but with assertion technique can capture various properties whether it is small or complex. It could possible to not observe the violation if occur in waveform thus assertion can cover those cases.

Regression launch through isolation and non-isolation mode where meta stability stage is an inbuilt function. While running regression for Meta stability stage an issue with synchronizer has been encountered as the flop were dynamically changed which effect the performance. This requires an enhancement in synchronizer and delay performance.

System Verilog feature can be enhanced by taking advantage of multiple inheritance offered by interface classes the goal is to simplify the implementation of proposed technique and implement more robust verification environment.

REFERENCES

[1] Vermeulen, B. and Goel, S.K., 2002. Design for debug: Catching design errors in digital chips. IEEE Design & Test, 19(3), pp.37-45.

[2] Tarim, T.B., 2005, May. Mixed signal and SoC design flow requirements. In Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on (pp. 5974-5977). IEEE.

[3] Lear, J.A., 2009, December. Digital and Mixed-Signal Verification Differences. In Microprocessor Test and Verification (MTV), 2009 10th International Workshop on (pp. 91-94). IEEE.

[4] Mulani, P.D., 2009, December. SoC level verification using System Verilog. In Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on(pp. 378-380). IEEE.

[5] Gang, P., 2009, September. Behavioral modeling and simulation of analog/mixed-signal systems using verilog-ams. In Information, Computing and Telecommunication, 2009. YC-ICT'09. IEEE Youth Conference on (pp. 383-386). IEEE.

[6] Yang, X., Niu, X., Fan, J. and Choi, C., 2013, March. Mixed-signal system-on-a-chip (SoC) verification based on systemverilog model. In System Theory (SSST), 2013 45th Southeastern Symposium on (pp. 17-21). IEEE.

[7] Kuo, P.Y. and Sie, L.F., 2015, June. Analyze the behavior model based on Verilog-A for Sallen-Key low-pass filter. In Consumer Electronics-Taiwan (ICCE-TW), 2015 IEEE International Conference on (pp. 460-461). IEEE.

[8] Juneja, D., 2015, January. On Event Driven Modeling of Continuous Time Systems. In VLSI Design (VLSID), 2015 28th International Conference on (pp. 198-203). IEEE.

[9] Jakobsson, A., Serban, A. and Gong, S., 2015. Implementation of Quantized-State System Models for a PLL Loop Filter Using Verilog-AMS. IEEE Transactions on Circuits and Systems I: Regular Papers, 62(3), pp.680-688.

[10] Das, P., Yadav, J.K. and Deb, S., 2016, January. Mixed Mode Simulation and Verification of SSCG PLL through Real Value Modeling. In VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), 2016 29th International Conference on (pp. 591-592). IEEE.

[11] Craciun, A.V., 2017, May. Real number models for op amp filters implemented in VHDL. In Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP), 2017 International Conference on (pp. 977-982). IEEE.

[12] Georgoulopoulos, N. and Hatzopoulos, A., 2017, November. Real number modeling of a flash ADC using SystemVerilog. In Electronics and Telecommunications (PACET), 2017 Panhellenic Conference on (pp. 1-4). IEEE.

[13] Mehta, A.B. ASIC/SoC Functional Design Verification. USA:Springer,2017.

[14] Little, S., Walter, D., Jones, K., Myers, C. and Sen, A., 2010. Analog/mixed-signal circuit verification using models generated from simulation traces. International Journal of Foundations of Computer Science, 21(02), pp.191-210.

[15] Kundert, K., 1998. Simulation of analog and mixed-signal circuits. IEEE BCMT Tutorial on Circuit Simulation.

[16] Ahmed Eisawy, Mentor Graphic. Improving Analog/Mixed-Signal Verification Productivity. Available at https://s3.amazonaws.com/verificationhorizons.verificationacademy.com/volume-8_issue-3/articles/stream/improving-analog-mixed-signal-verification-productivity_vh-v8-i3.pdf

[17] Nizic, M., 2012. Mixed-Signal Design Trends and Challenges. Mixed-Signal Methodology Guide–Advanced Methodology for AMS IP and SoC Design, Verification and Implementation, pp.1-10.

[18] Rashinkar, P., Paterson, P. and Singh, L., 2007. System-on-a-chip verification: methodology and techniques. Springer Science & Business Media.

[19] Karnane, K., Curtis, G. and Goering, R., 2009. Solutions for mixed-signal SoC verification. Cadence Design Systems.

[20] Wang, Q., 2012. Mixed-signal SoC verification using analog behavioral models. EDN-Electronic Design News, 57(14), p.33.

[21] Karnane, K., Curtis, G. and Goering, R., 2009. Solutions for mixed-signal SoC verification. Cadence Design Systems.

[22] Walter Hartong, Scott Cranston.,2009. Real Valued Modeling for Mixed Signal Simulation, Cadence Design Systems.

[23] Introduction to OVM. Available at http://testbench.in/OT_02_OVM_TESTBENCH.html

[24] Cadenas, O. and Todorovich, E., 2009, April. Experiences applying OVM 2.0 to an 8B/10B RTL design. In Programmable Logic, 2009. SPL. 5th Southern Conference on (pp. 1-8). IEEE.

[25] David HSU. Low Power Verification – "X" Marks The Spot. Available at http://semiengineering.com/power-verification-x-marks-spot/

[26] Arjun Shetty. Introduction to PCI Express. Available at: - http://www.testbench.in/introduction_to_pci_express.html

[27] Liu, Y., Xu, D., Cai, H. and Yang, C., 2017, July. System architecture design of PCIe root complex based on SOPC. In Control Conference (CCC), 2017 36th Chinese (pp. 5490-5495). IEEE.

[28] Nakamura, H., Takayama, H., Yamaguchi, Y. and Boku, T., 2017, December. Thorough analysis of PCIe Gen3 communication. In ReConFigurable Computing and FPGAs (ReConFig), 2017 International Conference on (pp. 1-6). IEEE.

[29] PCI Express Base Specification Revision. Available at:- http://composter.com.ua/documents/PCI_Express_Base_Specification_Revision_3.0.pdf

[30] Tim Fairfield. PCI Express 3.0 and 4.0 Test Challenges: Link Equalization Testing 8Gbps vs 16Gbps Rx Testing. Available at https://www.keysight.com/upload/cmc_upload/All/HSDSeminarPaper3.pdf

[31] Brian Bailey. PCIe 3.0 Equalization. Available at https://www.edn.com/electronics-blogs/practical-chip-design/4405647/PCIe-3-0-Equalization

[32] Rangel-Patiño, F.E., Rayas-Sánchez, J.E., Vega-Ochoa, E.A. and Hakim, N., 2018, March. Direct optimization of a PCI express link equalization in industrial post-silicon validation. In Test Symposium (LATS), 2018 IEEE 19th Latin-American (pp. 1-6). IEEE.

[33] PCI Express 3.0 Equalization: The Mystery Unsolved. Available at https://www.ashtbit.net/applications/lfrunew/resource/PCIe_Equalization_v01.pdf

601662012_by_pallavi_sharma.docx

by

Submission date: 15-Jun-2018 01:35PM (UTC+0530)
Submission ID: 975897378
File name: 601662012_by_pallavi_sharma.docx (3.03M)
Word count: 12461
Character count: 65639

601662012_by_pallavi_sharma.docx

ORIGINALITY REPORT

6%	%	6%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	"An Introduction to IC Verification", Effective Functional Verification, 2006 Publication	1%
2	H. Rotithor. "Postsilicon validation methodology for microprocessors", IEEE Design & Test of Computers, 2000 Publication	1%
3	Babin, Steve. "Diagnostics for Design Validation -- Whether you are testing a new microcontroller or an ASIC, post-", Embedded Systems Programming, April 1 2003 Issue Publication	1%
4	Doug Josephson. "The good, the bad, and the ugly of silicon debug", Proceedings of the 43rd annual conference on Design automation - DAC 06 DAC 06, 2006 Publication	1%
5	"Design-for-Test", Electronic Design Automation for IC System Design Verification and Testing, 2016. Publication	<1%

Pallavi 16/6/18