POST-SILICON GENERAL BUS FUNCTION MODEL MODELLING APPROACH USING C CODE FOR TEST SCHEME REUSE AND SHARING

By

CHANG LING KWAI

A Dissertation submitted for partial fulfillment of

the requirements for degree of

Master of Science (Micro-electronics)

JULY 2014

ACKNOWLEDGEMENT

Along the process in carrying out this research project, various knowledge and experiences gained, especially the process of learning in research skill. The whole learning and implementing processes are very fruitful and worthwhile. In addition to research skill, development in technical skills and communication skill also have been achieved. There are many people who should be thanked for helping in the successful completion of this project.

First and foremost, I would like to express my gratitude and appreciation to all those who gave me the possibility to complete this dissertation. A special thanks my supervisor, Mr. Zulfiqar, whose help, stimulating suggestions and encouragement, helped me to complete this research.

Furthermore, I would also like to thank my seniors and colleagues, especially Mr. Lim Chen Tatt for his immense support and advice. I am glad to get his help all the time as a lot of knowledge has been gained and guidance in doing this project.

Last but not least, I would like to extend my heartiest gratitude to all my friends and family who support and help me, directly or indirectly, in the process of completion of this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBEVIATION

ASIC	Application Specific Integrated Circuit
BFM	Bus Function Model
CPU	Central Processing Unit
DUT	Design Under Test
EDA	Electronic Design Automation
FIFO	First In First Out
FSL	Fast Simplex Link
FPGA	Field Programming Gate Array
GPIO	General Purpose Input Output
GUI	Graphic User Interface
HDL	Hardware Description Language
HLS	High Level Synthesis
I2S	Inter IC sound
IC	Integrated Circuit
ILA	Internal Logic Analyzer
IP	Intellectual Property
ISR	Interrupt Service Routine
I/O	Input and Output

LA	Logic Analyzer
LMB	Local Memory Bus
OPB	On-chip Peripheral Bus
OVM	Overall Verification Methodology
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
RTL	Register Transfer Level
SDK	Software Development Kit
SoC	System on Chip
SUT	System Under Test
SV	System Validation
VLSI	Very Large Scale Integration

PENDEKATAN PEMODELAN MODEL BUS FUNGSI PASCA-SILIKON MENGGUNAKAN PENGATURCARAAN C BAGI SKIM PERKONGSIAN DAN PENGGUNAAN SEMULA

ABSTRAK

Pengesahan pra-silikon dan pasca-silikon merupakan salah satu kesesakan dalam proses pembangunan litar bersepadu di mana penambaikan dalam bidang ini boleh mengurangkan masa pemprosesan produk. Jurang wujud antara pra- dan pasca-silikon di mana kedua-dua bidang melakukan proses yang serupa, tetapi dalam persekitaran yang berbeza. Salah satu cara untuk merapatkan jurang ini ialah dengan menggunakan skim pengesahan yang sama yang boleh digunakan dalam kedua-dua pra dan pasca silikon. Disertasi ini membincangkan kaedah untuk memodelkan pasca silikon Model Bus Fungsi dengan mengguna bahasa pengaturcaraan peringkat tinggi (kod C) yang mampu berkongsi kandungan dan penggunaan semula antara pengesahan pra dan pasca silikon. Dua kaedah pasca silicon dicuba dan yang terbaik dipilih. Model prototaip yang dihasilkan diuji dalam persekitaran pengesahan pasca-silikon untuk membuktikan kesahihan penggunaan kaedah yang dicadang. Kandungan program dalam kod C dicipta dan disintesis ke dalam FPGA, yang bertindak sebagai penjanaan rangsangan dan data untuk proses pengesahan. Permodelan peranti BFM dengan protokol bas Audio diperiksa dengan memeriksa isyarat surih sumulasi, untuk memastikan ia berfungsi dengan betul. Dengan model ini, pra- dan pasca-pengesahan silikon boleh berkongsi skim ujian yang sama, menjimatkan usaha dan masa membangunkan Model Bus Fungsi.

POST-SILICON GENERAL BUS FUNCTION MODEL MODELLING APPROACH USING C CODE FOR TEST SCHEME REUSE AND SHARING

ABSTRACT

Short product life cycle of a semiconductor IC is crucial for a company to gain better market share. But current validation process has become one of the bottleneck in semiconductor IC development process, where enhancement in this area can reduce the product lead time to market. There is a gap between pre- and post-silicon where both area is doing similar things but in different environment. A way to bridge the gap is by using the same test scheme which is applicable for both environment. This paper discuss and come out with an approach on how to model post-silicon BFM using high level programming language (C code) that is capable of content sharing and reuse between both pre- and post-silicon validation. Two methods to model post-silicon BFM are being approached and the more suitable one is being selected. The comparison is done is aspect of meeting strict timing requirement in sending signal. A prototype model of the selected model, which is second model is developed and tested in post-silicon validation environment to prove validity of the design. Test content in C code is created and synthesize into FPGA, act as stimulus generator for validation. The BFM device bit-accuracy behavior of the modelled Audio bus protocol is checked with waveform and signal trace. With this working model for developing BFM, pre- and post-silicon validation can share the same test scheme, saving efforts and time of developing test devices.

CHAPTER ONE

INTRODUCTION

1.1 Introduction

At this advance electronic technology decade, the design of IC (integrated circuit) and SoC (system on chip) is getting more and more complex. The total number of transistor packed in a silicon chip had double every eighteen months since 1965 as predicted by Gordon E. Moore, co-founder of Intel Corporation; giving us more functionality to be embed in a smaller foot-print. With the VLSI technology we have today, a fully functional and powerful computer can be in the size of a book which is also known as tablet in contrast with those bulky computers decades ago. Designing of such complex IC and SoC is very difficult and requires collaboration of experts from different fields. Apart from designing, a product must also undergo a series of functionality validation and quality control before delivering to the market to make sure a good product is being produced. Design and development of IC is prone to errors mainly due to human errors and compatibility issue upon integration of all parts together. Hence proper validation process is crucial to ensure consumers in getting a stable product.

Verification and validation such as pre-silicon and post-silicon is only a strategy to demonstrate the quality and attributes of a product, which stands differently compared with testing. Verification and validation is to check the design correctness by referring to design specifications while testing is to check for manufacturing correctness(Hakim et al. 2010). Testing is done in a totally different way, in a later stage

compared to verification and validation which took place in the front-end design process. All these process is imported in order to deliver a healthy product to customers.

Validation of IC and SoC is done in several timeframe of the whole development process. Validation is started as soon as possible as this process consumes a lot of time which is up to 70% of the overall design process, further lengthen the time-to-market of the product(Kakoee et al. 2008). This is where pre-silicon validation exist to validate the design functionality in simulation environment right after the chip design is created in HDL (hardware description language). This process is impossible to be done manually and hence engineers create test programs to verify the design. The design in HDL is simulated in a powerful computer and test codes which is also known as testbench to exercise all the functions of the design as stated in the product specification sheet. Engineers create testbench to check the functionality of the design and informs when there are error in the checking. Whenever an error is found in the design, they will root-cause the error and feed back to the designers to rectify the errors.

After validating the design in pre-silicon stage, the next is post-silicon validation where verification is carried out on real hardware, which is the prototype chips fabricated with ASIC process. Validation in this stage is close to real world environment versus simulation which is done in pre-silicon. The characteristic of the design is more deterministic in real hardware as the design is implemented with electrical signals and devices in a chip. The topology of carrying out validation at this stage is same as what is done in pre-silicon, just in a different methodology due to environment of validation is different. The algorithm of the test codes will be the same but implemented in different way and environment.

Post-silicon validation uses some testing tools and equipment to interact with the prototype chip, verifying the functionality. From audio IP (Intellectual Property) perspective, the validation of the IC design is done by using test tools connected to the output peripheral as an output device to collect data from the design.

The key element in validation of IC and SoC design is the test codes (testbench) and hence these codes must be perfectly free from errors and capable to fully exercise the design. Testbench itself cannot create error else this will create difficulties in debugging an error in the design. With a fail testbench, engineers cannot be sure the design is wrong or the testbench and need to put extra effort in verification. The best way of preventing such error is to use a single test algorithm that is known stable, instead of creating different test codes in different stages.

1.2 Problem Statement

The biggest challenge in the semiconductor industry is not only creating product with the greatest technology but also compromise of delivering product with the latest technology as soon as possible to be the first in the market and gain greater market share. What make this a challenge is the complexity and technology advance of semiconductors IC nowadays, according to Moore's Law prediction of growth in transistor count in an IC. With more transistors available in an IC, which means more functionality can be packed in a smaller size IC. This is why mobile devices now is getting smaller in size but functionality wise is much greater compared to last time. With more functions to be integrated into a single IC, meaning there will be huge integration of different design modules both analog IP and digital IP. All these different designs are done by different group of people and there maybe problems when integrating them all together such as design modules incompatible with each other,

signal integrity issue and design failure. Hence proper verification and validation of the product need to be done carefully in different stages to make sure healthy products are delivered to consumers. Briefly going through the process in developing an IC above, a conclusion can be made which is a more complex design will require extra effort, resources and time in the development(Chen et al. 2013).

A lots of strategies had been introduced by the semiconductor industry to shorten a product lead-time to market such that many advance EDA (Electronic Design Automation) tools are introduced. Companies such as Synopsys, Cadence and Mentor Graphics are among the leaders in EDA tools industry, contributed a lot in the growth of semiconductor industry. These tools help to automate most of the IC semiconductor development process, not only speed-up the design-life-cycle but also enhance and increase the efficiency. The challenges that EDA tools faced as semiconductor IC design complexity increase includes: reuse of hierarchical design for SOC integration, verification and test, cost-driven design optimization, reliable implementation platforms and tools for networked system environments(Ruchir Puri et al. 2010). There is still a very big opportunity of improvements for design and validations tools to par with the advancement of semiconductor technology.

Apart from designing process, verification and validation is also a crucial process to produce a good product. This process usually requires many effort especially when the size and complexity of modern hardware system which further burden the task(Wile et al. 2005). Hence verifying such design involves tens or hundreds of person and the compute power of thousand workstation(Adir et al. 2011). Lead time to market of an IC product is usually gated by verification and validation which requires equal or larger effort compared to the design process, becoming one of the major bottlenecks in design and verification flow(Puhar & Zemva 2008). The ultimate challenge in pre-

silicon and post-silicon is to deliver highest quality of a product in the most efficient way(Keshava et al. 2010).

The pressure and commitment of post silicon validation is significant when all these challenges come together, forcing post silicon validation engineers to continuously enhance the validation process in a faster, cheaper and better way with different methodologies(Keshava et al. 2010). From the older days without computers where engineers hand-craft circuit and schematics on papers till today's complex IC design done in computers, this industry is always evolving. Even though semiconductor IC technology is getting better but the cost of development and production is not increasing. This phenomenon is results of engineers coming out with better design and better process, having break-through in every generations of IC design.

When a design is to be validated in different stages and environment, different methods and tools are needed to suit the requirements. Take functional test as example, the methodology of validating the design is same in pre-silicon and post-silicon environment, but two different traffic generator or stimulus generator is needed. Hence there will be double effort to develop the traffic generator for only one purpose. Developing test content for complex semiconductor IC design is challenging and requires a lot of effort where else replicating this effort for every project and model amplifies the validation cost. Besides of increasing cost, extra effort is required to verify a bug in different model with a completely different and new testing tools(Torres et al. 2009).

Customers will expect a quality product despite the reducing product development cycle and the industry must come out with better methodology in the development process in order to survive in the highly competitive market. A new

methodology to bridge the gap between pre-silicon verification and post silicon validation is needed, saving resources and reduce lead-time. One of the potential approach is the verification and validation infrastructure reuse which is able to reduce verification cycle and reduce overall time to market for product delivery(Srivastava et al. 2012a).

The goal of this research is to propose a general model and method of developing functional test stimuli reuse scheme for both pre-silicon verification and post-silicon validation, solving the problems states above. When the test scheme can be shared between pre- and post-silicon, engineers from these two environment can share resources on developing content, saving resource and time especially for post-silicon validation.

1.3 Objectives

The objectives of the research project are as follows:

- Path searching for the appropriate model and method of developing BFM for resource reuse between pre-silicon verification and post-silicon validation for developing testing devices.
- To create a model of developing post-silicon BFM to ease the efforts of content reuse from pre-silicon to post-silicon.
- To develop a prototype BFM test device in FPGA for functional validation in post-silicon environment.

1.4 Project Scope

The scopes of work consist of:

- Research based on system validation in post-silicon environment.
- Audio bus protocol (I2S) is chosen as a medium to implement on proposed methodology.
- The research will focus on BFM only but not the whole validation framework, testbench or OVM (overall validation methodology)
- Create basic architecture of the software framework in high level programming language (C codes) for the test model.
- Prototype will be developed on a FPGA kit to implement the proposed methodologies.
- Prototype BFM device developed is not tested in real validation platform.
- The prototype BFM device functionality is verified through simulation and logic analyzer waveform capture.
- The prototype BFM device functionality and performance in timing specific is measured through signal and trace capture.
- Performance of the proposed methodology is not measured in signal integrity and power.
- Logic synthesis and design implementation performance is not taken in a factor of analysis.

1.5 Thesis Outline

The thesis is mainly consists of five chapters:

Chapter one defines the research background information and problems to be solved at the end of this research. The objectives and scope of this research is stated in this chapter as well making clear statements on this research. This chapter is mainly about the overview of this whole research.

Chapter two will be the literature review which give a more detail background information about this research. There are several sub topic in this chapter to discuss in detail of certain background knowledge that is crucial about this research such as semiconductor industry, pre-silicon, post-silicon and validation approach.

Chapter three is about methodologies to carry out this research. Methodologies include tools and technique to use them, how to complete this research from very beginning and methods to examine the results. There are two methodologies proposed in this research and one will be chosen as the most appropriate to meet objectives stated in chapter one.

Chapter four will show the results collected and analyzed after carrying out this research according to methodologies in chapter three. The performance of two proposed methodologies will be compared and the most applicable will be chosen. Discussion and analysis continues to get a suitable method and working model for developing post-silicon BFM which is also friendly to pre-silicon environment.

Finally, chapter five gives the conclusion about the overall research. There will be some discussion about the advantages and disadvantages of the selected methodology and recommendations for future work on this project are stated as well.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter will discuss the areas of research related to this project which includes overview of verification and validation in Application Specific Integrated Circuit (ASIC) design, implementations of current methodologies in verification and validation for both pre-silicon and post-silicon, .

ASIC design is getting harder for complex design nowadays and prone to errors or bugs especially after integrating the whole design from different IP. Hence, proper verification and validation of the design is crucial to deliver a healthy product to consumers. Verification started as soon as the design is created in hardware description language (HDL) in pre-silicon stage using simulation environment. This process continues to post-silicon stage where validation is done on real silicon after fabrication. Most of the errors are identified in pre-silicon stage, where approximately 90% of bugs are found here. Then post-silicon will continue the effort to identify the rest of bugs which is not found in pre-silicon.

This chapter will also discuss in more detail about verification and validation methodologies that the industry is practicing. The most common way of doing verification in pre-silicon is creating testbench to exercise the DUT (Design Under Test). Testbench is a test code written in high level language which is not able to be synthesized into RTL (Register Transfer Level) and only able to run in simulation

environment. Behavioral Function Model (BFM) in testbench will be discussed further especially how this can be implemented into post-silicon validation.

The next part to be discussed is FPGA (Field-Programmable Gate Array) emulation background knowledge and how it is implemented as a testing device in post-silicon validation. A brief overview about FPGA emulation is how a design can be emulated as a real device using FPGA integrated circuit.

Last topic to be discussed is example of audio data transfer bus protocol. This research will use Inter-IC Sound (I2S) bus protocol to demonstrate the implementation of proposed methodology. I2S is one of the bus protocol widely used for transferring data between host and audio device. This protocol support lower bandwidth data transfer which is normally used in mobile devices which does not require multiple channels of high-definition audio quality.

2.2 Semiconductor Integrated Circuit Validation

Verification and validation is a part of the development of a new product which play the role to ensure fault-free product delivered to customers. This applies to the development of IC which incorporate complex designs. The main function of verification and validation is to make sure the product is designed as defined in the design specifications. For IC development, this process will check the functionality of the design through a series of test to exercise the logics.

Reliability and compatibility of an IC especially advance micro-processor and SOC (System-on-chip) must be maintained at a standard as any issue can be costly to users. For OEM (original equipment manufacturer) customers, a faulty product can cost huge losses not only in terms of money but do include product time-to-market and

market share. For end users which do not have technical expertise will also face a lots of difficulties as they cannot get immediate access to technical support, end up not utilizing what they paid for. An IC that undergo intensive and comprehensive validation program can help to reduce all such losses at the same time increase the term of life of the product(Intel Corp 2003).

Many IC manufacturers is investing a lot in the product quality control where verification and validation is part of it. At 65 nm technology semiconductor IC development process, validation effort consumed more than half of total design effort as measured in cost and this problem become more serious as the industry is moving to smaller processing technology(Nahir et al. 2010). The whole process of validation mainly consist of three stages: pre-silicon verification, post-silicon validation and volume testing as shown in Figure 2.1. Pre-silicon verification is started in the design phase and go concurrently. While post-silicon is only started when the design is fabricated into a real silicon, which most-likely will be the prototype IC. These two stages' role is to look for design faulty and let the designers to correct their design. Then the last stage of is volume testing that is totally different from the previous two stages, plays the role to check for manufacturing defects and yield issue.

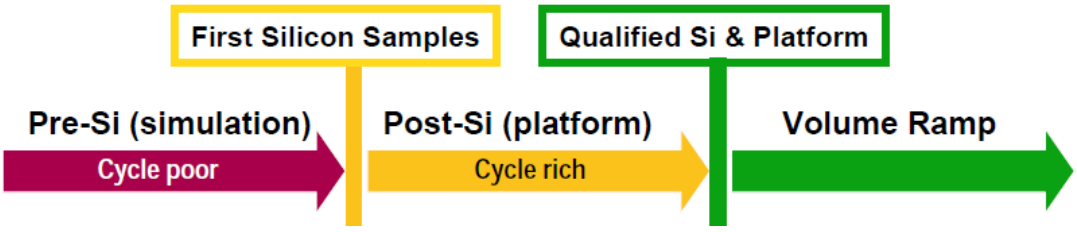

Figure 2.1: Validation and verification process in IC development(Hakim et al. 2010)

2.2.1 Pre-Silicon Verification

To shorten the product development lead-time to market, verification and validation of the design of semiconductor integrated circuit is started as early in pre-silicon stage where there is no real hardware yet. The verification is carried out in virtual environment where the design is simulated in powerful machines or FPGA emulation system. These are the two main verification strategy in pre-silicon.

Simulation is the earliest stage of verification where the semiconductor integrated circuit design is in register transfer level (RTL) and gate level, haven't go through all those back-end process such as netlist synthesis and physical layout design. The design in HDL is simulated in powerful processing power machines to achieve the real behavior of the product. Not much effort required to simulate a design from HDL language especially with the aid of nowadays advance EDA (Electronic Design Automation) tools. Designers can get to see the how their design behave as compared to the design specification and make corrections in this stage. The biggest advantage of verification in simulation environment is the efficiency in making correction and flexibility in monitoring internal signals. When designers found some bugs in the design throughout some verification process they can immediately make the corrections required and rerun simulation again to see the effect. Pre-silicon simulation is very close to the design process and hence very efficient. Another advantage of simulation is the internal signal visibility and controllability of the IC design(Ruan et al. 2011). Designers have the flexibility to monitor any internal signals for debugging purpose since the design is only simulated in machine for verification and strongly increase the debugging capability(Nahir et al. 2010).

Emulation is a way of hardware acceleration which can speed up the execution of pre-silicon verification by loading the design into a large FPGA system such as shown in Figure 2.2. The goal of emulation is to create a system that mimics the real silicon behavior and this the closest approach to a real silicon. Emulation system combines a number of FPGAs together to be able to support large and complex design. Verification methodology is similar to simulation but with less control and visibility in emulation system. Setting up an emulation system for a full chip integration of a silicon design requires huge effort that not many companies are willing to invest in.

Figure 2.2: Xsigo emulation system(Stuart 2011)

FPGA (Field Programming Gate Array) emulation is a cost and time saving option of system modelling and verification method compared with ASIC (Application Specific Integrated Circuit) prototyping. Using FPGA in the industry is getting common and popular due to the high performance of nowadays FPGA, flexibility and low cost. The greatest advantage of FPGA over ASIC is the flexibility of reconfiguring a hardware logic design and short time to run prototyping. ASIC design is not reconfigurable after being manufactured and the whole process requires a lots of steps before coming to fabrication(Rosinger 2004).

Comparing with simulation verification, FPGA emulation does not require continuous connection with a workstation during execution and emulation speed can achieve 1 MCPS or more(Kim n.d.). The whole design can be loaded into a hardware

for acceleration, imitate the functions of a real device. However, the biggest drawback of FPGA emulation is it only supports synthesizable HDL code where resources needed to transform those non-synthesizable testbench into synthesizable one. This will requires enormous effort to load a design into emulation system.

They way of doing verification in pre-silicon simulation is writing test flow in testbench and use BFM as an interface connecting the DUT (design under test) to drive stimulus out and sample input signal from DUT to verify functional result (Lahti & Wilson 1999). The key components for pre-silicon verification is testbench, BFM and DUT. Testbench is basically test program codes written in high-level programming language, usually system Verilog which instruct the whole test flow from setup to verify. BFM main role is to imitate the "bus" or I/O signal characteristic of a design, allowing testbench to interact with the DUT. The overview functional diagram is shown in Figure 2.3.

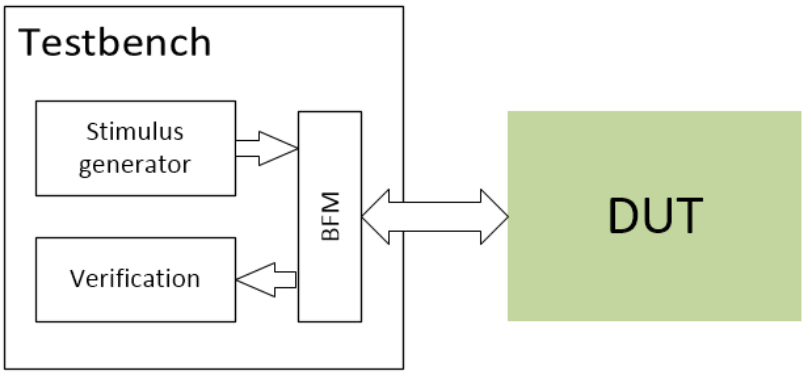

Figure 2.3: Overview of pre-silicon verification architecture

2.2.2 Post-Silicon validation

The term post-silicon means post design process in ASIC development where real IC unit is fabricated. Test and verification of the design done as this stage is called post-silicon validation and also the last step in the development process. Prototype IC unit are fabricated for testing in real world environment, to ensure the stability and functionality of the design.

Verification and validation of semiconductor IC design is always focused on pre-silicon stage, where a lot of research are done for pre-silicon analysis. As the complexity of IC design is getting larger, post-silicon is becoming more prominent to catch bugs not found in pre-silicon verification(Ray & Hunt 2009). Because of the complexity, some logical bugs may escape from pre-silicon verification and remain in the silicon which are found after the design is fabricated(Fujita 2011). Validation in this stage is very important as to cover some test case which is not able to perform in pre-silicon environment. An example of test which pre-silicon cannot cover in simulation environment is electrical and power issue. There are some simulation software to simulate the electrical and power performance but this can only serve as a high level overview for the designers. Besides, post-silicon validation run on real device with real-time clock speed, allowing the exploration of more advance design states which pre-silicon verification is not capable of(Ray & Hunt 2009).

There are two main categories of validation in post-silicon: the functional and logic validation done by compatibility verification and system validation, analog signal validation done by analog validation, circuit marginality and speed validation(Tommy et al. 2007). This research will only focus on system validation where the methodologies discuss later is only applicable for functional validation. The main role

of system validation is to find all logic/functional bug in the shortest time, before the product is delivered for further process, since this environment is friendlier for low level debugging.

Since system validation need to find all functional bugs and must overcome challenges such as short validation time-frame and validation efficiency, a well-defined environment and methodology must be established. Validations vectors that could assist to overcome the above challenges include: an aggressive software environment to stress the design, a host computer that support software environment, automation tools, controllability and debug capabilities tools, efficient test execution and debug methodology.

The greatest advantage of post-silicon validation compared to pre-silicon verification is the speed of running test cycles, hence widen the test coverage dramatically. Running a test cycle in pre-silicon simulation environment needs at least an hour and several minutes in emulation depending on the design complexity. In contrast with post-silicon environment, a test cycle requires a mere of few seconds which is thousand time faster. Table 2.1 clearly shows a comparison between validation approaches of different environments with their performance in throughput. With such a speed, post-silicon validation is able to cover all the design architectural scenarios, including internal and external events(Tommy et al. 2007). The methodology used on post-silicon validation, leveraging the high processing speed in real silicon is running random instruction test. Random instruction test can have a wide spectrum of events and cases as the silicon is exercise with a huge amount of discrete tests to cover all the possible architecture and micro-architectural scenarios. This test is initiated by software program which setup all the hardware for concurrency, prepare randomize stimulus, execution by sending and receiving data stimulus and lastly verify the result. All these

test program are written based on hardware specification or features and must go through reviews with experts across architectural, design and validation teams, to prevent any possible architectural and micro-architectural corner case left-out not tested(Bojan et al. 2007).

Table 1: Simulation, emulation and silicon approach for validation throughput(Foutris et al. 2011)

Validation approach	**Throughput (instructions/sec)**
System simulation	$\sim 10^3$
RTL simulation	$10^1 \sim 10^3$
Emulation	$\sim 10^5$
FPGA prototyping	$\sim 10^6$
Silicon	$10^7 \sim 10^9$

Validation system in post-silicon environment consist of a host computer as work station, SV (system validation) test-card as BFM and SUT which is the test unit. Host computer works as the main controlling unit in this validation system, playing the role from setting test to verification. SV test card is the device used to interact with SUT by transmitting and receiving data. The data is transmitted in the desired bus protocol suit with SUT. Then SUT is the new design unit which is under testing and validation. Figure 2.4 will give a clear picture of how the 3 devices interact in system validation.

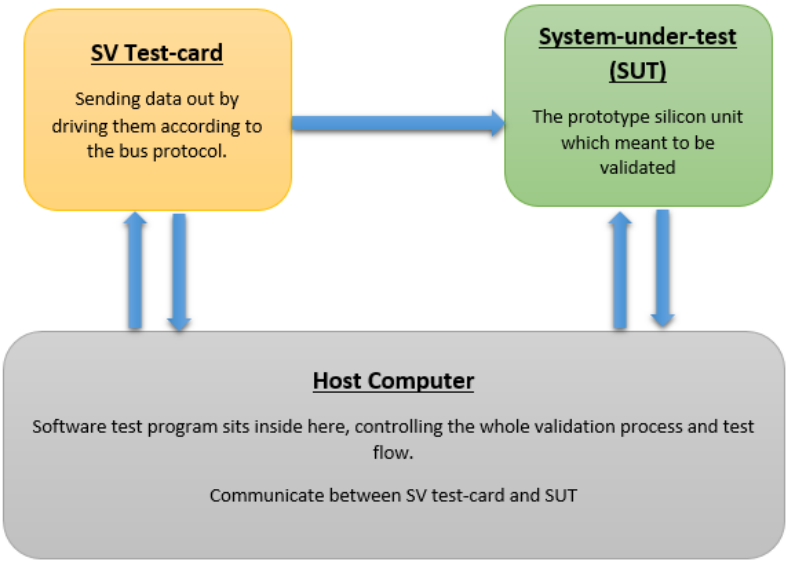

Figure 2.4: Block diagram of System Validation platform

The test flow in post-silicon can be simplified into three main steps: setup test, execute test and verify result as shown in Figure 2.5. The first step is setting everything needed for running a test including hardware and test environment. Take serial communication as an example, software need to setup the SV test-card for communication protocol and data to send for transmission, at the same time setting up the receiving end which is the system under test with the communication protocol and total amount of data to be receive. After everything has been setup properly, software will initiate execution by triggering start at the transmitting end. Data transactions happen in this stage and software will poll for completion from the receiving end, making sure the correct amount of data is sent. The last step will be verifying the result by comparing data setup from the transmission end and data received from the receiving end. This test sequence is repeated on every cycle with different data stimulus and hardware setting permutations.

Figure 2.5: Block diagram of a basic test cycle

A problem for post-silicon validation is the dependency on platform stability, electrical and signal integrity issues which is harder for engineers to debug(Nahir et al. 2010). Engineers is hard to narrow down to root cause the problem as unstable platform might be causing noise and interface the result. Furthermore, engineer might have spent effort debugging an issue but at the end root cause to be platform or hardware problem, instead of design bug(Bose et al. 2009). Hence a stable and reusable test scheme and hardware is required for all the projects and products, avoid debugging noise and save resource and time in developing test content.

Effective post-silicon validation for modern architectures must maximize test scheme content sharing and reuse to save time, resources, and budget developing testing devices, at the same time not limiting bug detection capabilities. Silicon validation and debug require a labor-intensive engineering effort of several months and have become most time-consuming with advance fabrication technology(Abramovici 2008). One of the easiest and common way to speed up the validation process is to increase validation resources, which is not practical as this will increase the cost of the product(Keshava et al. 2010). Companies are always trying to reduce the cost of product in order to gain an advantage over competitors. Creative and innovative solutions is needed for every aspect of IC development for winning the market. A good example of a related work with new methodology is able to reduce effort and time around 50% in setting up emulation for verification(Srivastava et al. 2012b).

2.2.3 Bridging between Pre-Silicon and Post-silicon Validation strategy

Pre-silicon verification and post-silicon validation is holding the same role of ensuring IC design is healthy and bug-free upon delivering to customers. Both are equally important in the design process as there are certain condition which is only able to be done in one method. Pre-silicon verification is limited to scope and volume coverage as compared to post-silicon due to slow simulation and emulation speed, but hold the greatest advantage of signal visibility and controllability. Post-silicon validation can have large and wide coverage as compared to pre-silicon but loses the visibility and controllability of internal signals(Nahir et al. 2010).

Table 2.1: Platform tradeoffs of different environment

	Simulation	**Emulation**	**Silicon**
Performance	Slow	Faster	Ideal speed
Control & visibility	Total model control and visibility	Control and visibility with some penalty	Very limited
Cost	Inexpensive	Expensive emulation system	Very expensive ASIC process

One of the common method to bridge between these two environments to enhance the validation process is leverage the advantages from each side to counter the disadvantage. An example is to run wide coverage test in post-silicon validation and

capture the failure to be reproduced in pre-silicon environment which has greater debugging capability(Nahir et al. 2010).

Some of the current effort to bridge the gap between pre-silicon and post-silicon includes various forms of doing software acceleration in hardware platform such as FPGA or emulation. Although these methods are still slower than real silicon unit, but they are significantly faster than simulation. Another challenge of this approach is the required of huge amount of unique engineering work from the design team and cost of setting up these system is relatively high(Bently 2010).

Reusability over project life-cycle is achieved by having the same test scheme running on different environment, allowing comparison of test results. The reuse of verification code and methodology is a major factor providing significant reduction of the overall verification costs(O et al. 2000).

2.3 Testbench for Pre-Silicon Verification

Testbench is a set of testing written in programming codes generating input to a design, then optionally observe the response in simulation to verify the correctness of IC design(Bergeron 2007). The property of testbench is to mimic the environment in which the design will reside, verify the IC design meets the design specification or not. Testbench is programmed to exercise the IC design with meaningful scenarios to detect failures so that bugs can be identified and corrected before a design is shipped to customer. A testbench can be as simple as a file with clock and input data or a more complicated file that includes error checking, file input and output, and conditional testing.

Testbench in simulation environment provides the stimulus to exercise the IC design. The stimulus generator will first generate input test pattern at a high level abstraction which is pre-defined by verification engineer. A driver is a type of BFM (Bus Function Model) that converts the test pattern from stimulus generator into actual input in the form of low level like bits, driving them into the DUT (Design Under Test). At the same time, a same set of test pattern is stored in a scoreboard which is a memory or storage until the output comes out from DUT for verification. DUT will process the input data and create output signal which will send to another BFM named as Receiver. The Receiver BFM collects DUT output and sends the data to Scoreboard.

As more and more design entry moves to higher level languages such as C/C++ and System C, it's possible to write testbench in C to verify the functionality of these high level models. However, for design implementation, an RTL description is still required. If this RTL is coded from scratch, most of the advantages of creating a high-level description is not worth anymore. Although most of the testbench nowadays is in System Verilog which is very friendly to OVM (Overall Verification Methodology), but using other high level languages is also applicable with some workarounds needed.

Usually the testbench is used with OVM technology for creating a reliable stimulus generator. OVM is a framework and software architecture with many functions that is able to be reused across projects. Engineers need to create testbench that can work together with OVM framework and run pre-silicon verification. The testbench can be viewed as a small core in the OVM framework, calling functions available to run the whole verification flow. Testbench is usually written in System Verilog but nowadays the industry is trying to adapt high level language for writing testbench. Using high level programming language such as C code, C++ and SystemC will create behavioral testbench and OVM wrapper is needed to wrap around the codes

for interactions between the testbench and OVM framework. Figure 2.6 shows how behavioral testbench interact with OVM framework by using an OVM wrapper.

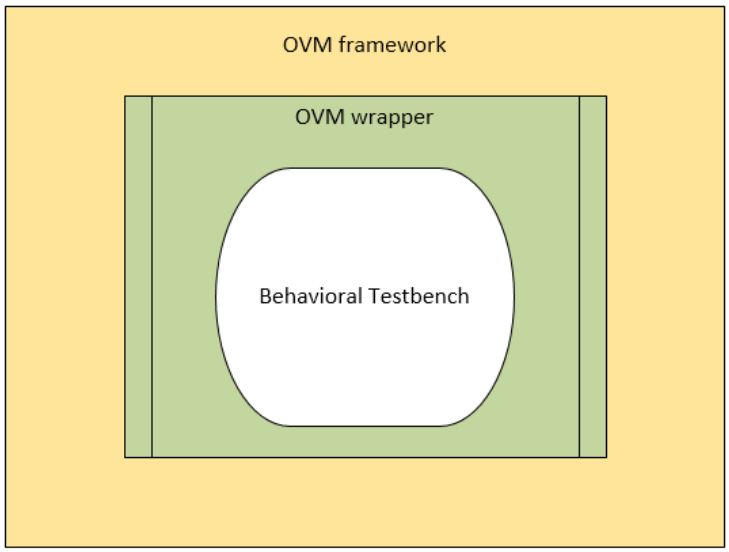

Figure 2.6: Illustration of how behavioral testbench work with OVM framework

2.4 Bus Function Model (BFM)

BFM is also known as transaction verification model which is able to simulate bus transaction or reflect the I/O (input and output) level behavior of an IP IC design. This model is usually used in pre-silicon design verification which is done in simulation environment(Torres et al. 2009); function as a device that interacts with DUT by both driving and sampling signals(Pesavento & Privett 1999). The IC design is in hardware description language coded according to the design specification requires some external stimulus for verification. To test and verify the design effectively, the test is done by simulating the real environment and condition where the IC design will reside. Some tasks are created to exercise the IC design and BFM is the model that communicates with it, send and receive data in the form of complex waveforms and protocols.

BFM is very important in functional and system validation because of the capability to replace and imitate real peripheral or device. Connecting a real audio device such as microphone, speaker or codec as data traffic generator is not applicable for functional validation. A deterministic traffic generator such as BFM is a must for validating and errors in data transmission since comparing data sent and receive at different ends is the verifying method. The function of BFM is illustrated in Figure 2.7.

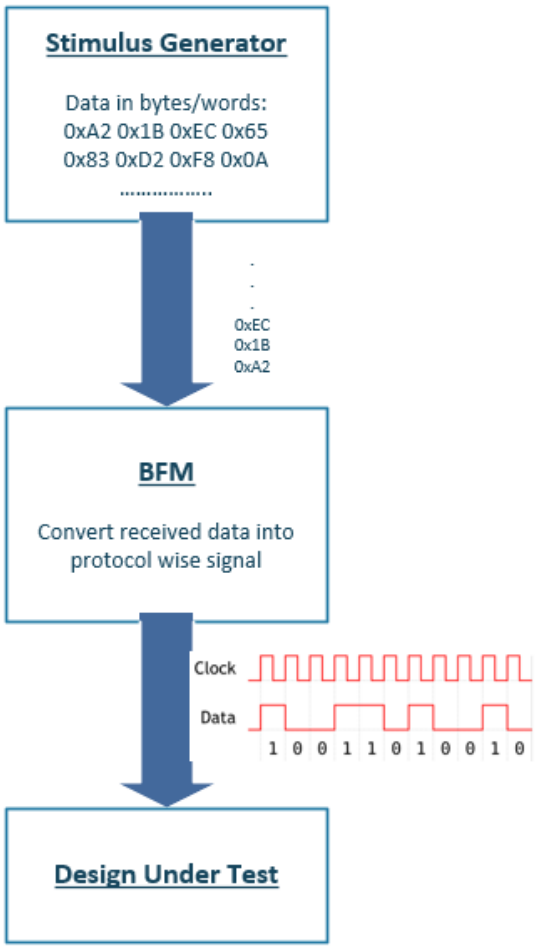

Figure 2.7: Illustration of BFM function

For post-silicon validation approach, external traffic has to be generated from an external device, where remote instructions execute to mimic BFM behavior. The test environment and method differences cause test scheme to diverge across projects and