Middle-East Journal of Scientific Research 15 (11): 1570-1574, 2013
ISSN 1990-9233
© IDOSI Publications, 2013
DOI: 10.5829/idosi.mejsr.2013.15.11.11284

Post Silicon Functional Validation from an Industrial Perspective

[1]P. Manikandan, [2]Juhee Mala and [3]S. Balamurugan

[1,3]School of Electrical Engineering VIT University, Vellore, Tamil Nadu, India
[2]ST Microelectronics, Greater Noida, India

Abstract: Post silicon validation is the final process in semiconductor chip manufacturing. Functional Validation (FV) is one among many methods used in post silicon validation. The aim of this process is to validate the functionality of the silicon prototype with respect to the functional specification given for the particular SoC (System on Chip) design. Thus by making sure the output product is free. This will lead to the mass manufacturing of the SoC Chips. In this paper we tried to explain how this post silicon functional validation is performed in the industry level. We also discussed about the various challenges, the industry is facing in terms of post silicon validation and the future research opportunities in this field.

Key words: SoC validation · Functional Validation · Research opportunities · Test · Silicon chip industry

INTRODUCTION

Post silicon validation is the process of finding out the bugs inside the silicon prototype of the designed SoC. The SoC chip manufacturing process has several steps. Figure 1 explains all the processes and flow. It starts with forming the functional specifications for the design. Market survey and customer feedbacks will help in forming these specifications. After completing the specifications the rest of the process can be directed into two paths. One is for implementing the specifications. Second is for testing the implemented designs.

The testing can be divided into two categories pre silicon and post silicon. Pre silicon validation deals with simulating the RTL code that was written to implement the specification and correcting the errors. After completing pre silicon validation the bug free implementation is manufactured as a silicon chip which is called a prototype or first silicon. Finding bugs in that prototype is termed as Post silicon validation. No matter how accurate our pre silicon validation, bug free silicon prototype can never be achieved owing to the fact of interaction between various components in an SoC will influence the functionality of the chip and the limitations of the Pre-Silicon subsystem in terms of clocks/power/analog components and other dynamic components of the SoC design.

Pre silicon validation is much slower compared to post silicon. The frequency will normally be in few hertz range. But the visibility of the circuit components and accessibility to those components will be very high in pre silicon validation. Post silicon validation operates in the actual device frequency which could be in the Giga hertz range and the visibility and accessibility of circuit components will be much lesser compared to pre silicon validation [1].

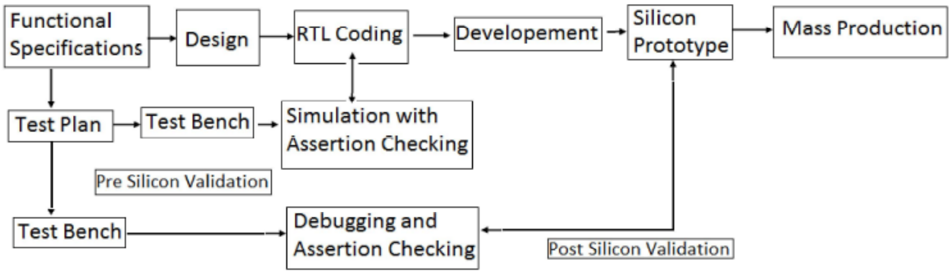

Fig. 1: Complete SoC design cycle

Corresponding Author: P. Manikandan, School of Electrical Engineering VIT University, Vellore, Tamil Nadu, India.

Post silicon validation consists of several disciplines [1], [2] Functional Validation (FV), Compatibility Validation (CV) and Electrical Validation (EV). Functional validation is the basic validation of the functionality of the SoC prototype. Compatibility validation deals with validating the prototype for compatible operation with previous version components and it also deals with the number of operating system that the product supports. It also comprises the compatible performance in various operating conditions for the product for example temperature variations. Electrical Validation is validating the electrical properties of the SoC components and IO pins.

Xiao Liu and Qiang Xu [3] are presenting an automated trace signal selection strategy that is able to dramatically enhance the visibility in post-silicon validation. They define the gate-level restorability's in a theoretically-precise manner and they propose an algorithm that can give the functionality details on the targeted IC product when decoding the state of the incorporated logical section of the IC.

Most commonly used method for doing post silicon validation is to form test benches and writing test cases. Onur Guzey, Li-C. Wang, Jayanta Bhadra [4] are explaining various constraints involves with.

forming test benches and a method to make the test bench development process automated. To develop high-quality test-benches, it can be helpful for an engineer to know, from the chip boundary, how to control signals in the unit under test. One may treat this as an Automatic Test Pattern Generation (ATPG) problem and desires a powerful sequential ATPG to solve the problem. This generally is hard in practice because of the complexity and accessibility issues mentioned above. First, the complexity of a design can be too high for an ATPG tool to run efficiently. Second, we may not have the ATPG-friendly models on certain units for the ATPG tool to run. They are giving a data mining based methodology for increasing signal controllability in functional test-benches.

Kanad Basu, Prabhat Mishra and Priyadarsan Patra [5] are proposing an efficient algorithm to select a profitable combination of trace and scan signals to maximize the overall signal restoration performance. Post silicon validation majorly uses two ways one is trace based method and the second one is scan chain based method. They are trying to Combine trace (nonscan) and scan signals to facilitate debug by enhancing signal reconstruction within trace buffer size and bandwidth constraints.

The post silicon validation process remains a specialized work in an industrial perspective due to the fact of huge capital involvement and time consumed by this process. Also this process is very essential for producing the bug free output product chip. Because of these factors a huge research is going on in this field to make the post silicon validation process more economical and less time consuming.

In this paper we presented the detailed process of post silicon functional validation and the various challenges faced by the industry in terms of post silicon validation. We discussed an initiative for automating the post silicon validation process which can be taken as a big topic for research and development.

Functional Validation: Functional Validation is performed on a customized board for a particular Silicon we are interested in testing. The board is customized in a way to support additional probing, measurement and other tools that are used for the validation purpose. One such important tool is debugger. Debugger is the tool used for accessing the internal circuit components and memory registers inside the silicon prototype. Unlike pre-silicon validation the accessibility for the inner circuit component is much smaller in post silicon due to the hardware complexity.

The validation starts with preparing a test plan. Test plan preparation starts well before the arrival of first silicon prototypes. As soon as the specifications are derived the test plan work kicks off. Test plan consists of several test cases where each test case is aimed to test the single aspect of the targeted part of the silicon such as a particular IP in an SoC.

Preparing this test plan requires clear understanding of what should be tested and how should be tested. The number of test cases required for testing a particular component or an IP is entirely depends on the validation engineer. The ultimate aim of the validation engineer who is making the test plan is to hit all the corner cases of the target. At the same time it is important to know when the testing gets over and the targeted part is bug free. The experience of the engineer will play a major role in the above aspect.

After preparing the test plan for each part or IP of the SoC, the test cases are written in a language which the compiler tool chain supports or the engineer is familiar with [6]. Systems C, C, System Verilog is few languages that are used for this purpose. For writing this test case there is no any particular algorithm fixed because of the uniqueness of each test case. But the process happening inside the test case can grouped under common steps.

Step 1: Configuring the Silicon to the targeted part or IP.

Step 2: Configuring the targeted IP according to the required test case.

Step 3: Configuring other IP's that are influencing the operation of the target IP (optional, depends on the test case).

Step 4: Triggering the events by giving input signals or writing to the control registers of the required operation.

Step 5: Data monitoring and recording. Raise the error flag if any mismatch occurs or stop the execution.

Step 6: Exit if no bug occur or repeat step. 1 to stop. 6 if any bug occurs.

Step 7: In case of consistent occurring of same bug, localize the bug source and report back the bug details.

Step 8: Try to rectify the bugs. If it's not rectifiable then report it back to the design team.

We can take an example and explain the above algorithm. Take an IP called DMA which is Direct Memory Access. It is a very common IP used in most of the SoC's. The purpose of this IP is to control the data transfer process without any interference from the processor so that the processor can do some other work when the data transferring is taken care by the DMA. By doing so the efficiency of the processor is increased. To test this IP first step is to prepare the test plan. The test plan can be like the list of test cases that should be run on the silicon to test the various activities of the DMA. The required components for doing the functional validation. The test cases can be like to test the basic data transfer movement. DMA can have multiple channels so each channel has to be verified for the reliable data movement. When it has multiple channels obviously there should be some mechanism for doing arbitration, say Priority wise. So there can be a test case for checking this Priority arbitration. Also DMA can be influenced by external interrupt pins. So this aspect also should be included in the test plan.

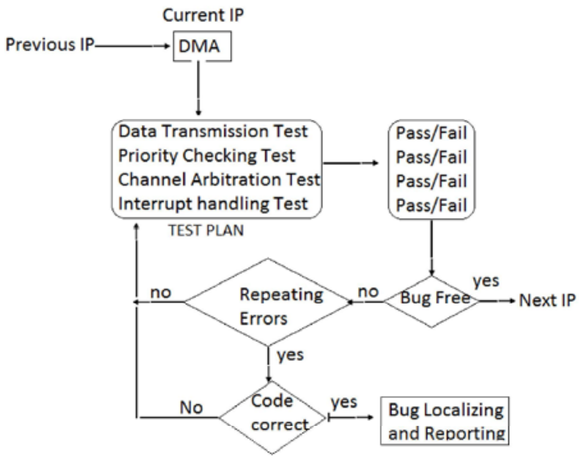

Fig. 2: Functional validation flow of DMA IP

After forming the test plan we should write the test cases. For this the above algorithm can be followed. Starting with, assigning values to the registers to configure the DMA active, followed by assigning source values and addresses of source and destination for required data transaction. Arbitrating the channel for the transaction is done next and the data transferred through the arbitrated channel to the particular address mentioned as destination should be the ultimate task. Like this all the other test cases can be written and run on the silicon to validate the IP.

Few IP's will influence most of the other IP's in the SoC. For example Clock generation and Interrupt Controller will influence the performance of most of the other IP's. These IP's interactions will be required in the test plans of all IP's. Few IP's will influence only particular IP's not all. For example CRC (Cyclic Redundancy Checksum) IP will influence CANSubsytem and Ethernet IP's only. These IP's will contribute to the test plans of the corresponding IP influenced by them. The above explained example is illustrated in figure 2.

After localizing the bugs the validation team will try to resolve it. The bug can be formed because of the electrical characteristics of the external pins or it can be of not accessing the registers for the particular IP module. These bugs can be classified into four categories 1) Minor Bug- can be fixed by document modification, 2) S/W workaround is possible, 3) Can be fixed by Metal Mask, 4) Complete Re-spin. If the bug is not fixable by the validation team then it is reported back to the design team. Changing the production method for overcoming post silicon bugs is an economically unviable task. So customer needs and feedback play a major role in dealing with post silicon bugs. Sometimes tolerable bugs are omitted owing to the fact of large money involvement. But the causes of the bugs are noted so that the next product will not suffer from the same bug [7, 8].

Challenges in Post Silicon Validation: Today's high demand for reduced size and increased functionality needs millions of transistor's to be incorporated in a single silicon chip. The possibilities for interactions between the various IP's inside a chip is getting increasing day by day. Silicon chip functionality is highly dependent on the interaction between the various IP's inside the sac and with the external interrupt pins. So no matter how accurately we are doing pre silicon validation bug free first silicon cannot be achieved. Also the re-spinning process for rectifying the post silicon bugs involves a huge capital investment. At the same time the output product should be bug free. Then only the industry can make success in the market. Which makes this post silicon validation as the most challenging and important processes of product development.

Post silicon validation remains an art. Only few experienced engineers can do this validation. The validation engineer who is doing post silicon validation should have good creative thinking other than intelligence. Because he/she has to imagine all the possible cases the product will face in the application perspective. And make those cases in the test code and run that in the silicon. Also to find out what causes the error in the process and to locate the reason behind those errors it needs some creativity. He/she also have to know the possible and cost effective way for correcting the errors. All these can be given in a single statement, the post silicon validation engineer should have the ability to hit all the corner cases for the product. This makes the whole post silicon validation process more challenging.

Research Opportunities: Post Silicon Validation is a very big field for research. Especially for automating this process many manufacturers are spending huge investments in R&D.Owing to the facts of everyday increasing number of IP's inside a silicon chip and the unpredictability of combination of bugs that can occur in a silicon prototype makes, post silicon validation process in automated mode as a very difficult task. Lot of researches are going on this concept. Few implementations are also available in industries. But those automated implementations will work only for a specific family of silicon chips having common functionalities [9, 10]. Changing the process for other family chips will again become a huge cost consuming task. Inserting recorders inside a chip for recording validation data [2] will result in increasing the area and the inherent errors will also pose the risk on the usefulness of the design. Bridging the gap between pre silicon validation and post silicon validation is also one topic of great interest for the researchers over the years [11, 12].

Generalizing the process of automated post silicon validation for all the silicon SoC chips is almost impossible because of the varying properties of the SoC's both specification sides and electrical characteristic side. But making a common platform for automating all chip vendors and all families of SoC's post silicon validation can be definitely possible. We are not proposing any programming language here. But it should be a complete setup having everything to complete the validation process. Say for example, a debugger, CRO, database for recording the values, an IDE (Integrated Development Environment) to write the test cases and to create hexadecimal files. The board which we use can be varied according to the chip vendor or the chip specification of the targeted SoC. This setup is already used in many industries but that whole setup differs from vendor to vendor. We are proposing that for making the post silicon validation process automated, the factor of a common platform for all vendors and all chip specifications should be considered. So that the huge process of Post silicon validation can be covered with a single umbrella holding backwards and all the vendors can put whatever they want into that open space and take whatever they need. This will lead to a clear understanding of the actual differences between the various SoC's functionalities and by that unified approach and broader visibility for post silicon validation can be created. The ultimate aim is to reduce the number of post silicon bugs and save millions of dollars for the semiconductor industry.

But this is a very far vision. We are still in the bottom step of this very big ladder. For achieving this we need boards and debuggers to support multiple bus structures for communicating between them. Also when it comes to common platform different vendors will be having their own architectural specifications and own communication protocols, so the platform should be capable of supporting all the bus protocols or protocol conversions should be made possible between the devices used in the platform. The idea is to make a system which can be connected to the specific board designed for the particular SoC by the particular vendor. If this is achieved then the post silicon validation process can be automated using a common system. By doing this, the aim of accomplishing common automation platform can be accomplished and the advances said in the above paragraph will be achieved. The combined effect of EDA industries and academic institutions can lead to a feasible solution for all challenges we will face in building this automation process for post silicon validation [1]. But once it's done it will save a huge amount of capital both in terms of production cost and time to market for the semiconductor industry.

CONCLUSION

Post silicon functional validation is a very challenging and time consuming task from an industrial perspective. The required skill set for doing this validation makes the task even complicated. Exploring all the aspects of this post silicon validation in a detailed manner will lead to clear goals for research and development in this field. By doing so an economical and less time consuming post silicon validation process can be achieved and the product can reach the market in a faster manner and millions of dollars can be saved in the semiconductor manufacturing industry.

ACKNOWLEDGEMENT

The authors thank Deep Mehta, Ruchika Saxena and Hishm Ashraf of STMicroelectronics for their immense contribution to this paper.

REFERENCES

1. Keshava, J., N. Hakim and C. Prudvi, 2010. Post-silicon validation challenges: how EDA and academia can help. In Proceedings of the 47th Design Automation Conference, pp: 3-7.
2. Mitra, S., S.A. Seshia and N. Nicolici, 2010. Post-silicon validation opportunities, challenges and recent advances. InDesign Automation Conference (DAC), 2010 47th ACM/IEEE, pp: 12-17.
3. Liu, X. and Q. Xu, 2009. Trace signal selection for visibility enhancement in post-silicon validation. InProceedings of the Conference on Design, Automation and Test in Europe, pp: 1338-1343.
4. Guzey, O., L. Wang and J. Bhadra, 2007. Enhancing signal controllability in functional test-benches through automatic constraint extraction. InTest Conference, 2007. ITC 2007. IEEE International, pp: 1-10.
5. Basu, K., P. Mishra and P. Patra, 2011. Efficient combination of trace and scan signals for post silicon validation and debug. InTest Conference (ITC), 2011 IEEE International, pp: 1-8.
6. Daoud, E. A. and N. Nicolici, 2009. Real-time lossless compression for silicon debug.Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 28(9): 1387-1400.
7. Gao, M., P. Lisherness and K.T.T. Cheng, 2011. Post-silicon bug detection for variation induced electrical bugs. InProceedings of the 16th Asia and South Pacific Design Automation Conference, pp: 273-278.
8. Lin, D., T. Hong, F. Fallah, N. Hakim and S. Mitra, 2012. Quick detection of difficult bugs for effective post-silicon validation. InDesign Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE1, pp: 561-566.
9. Ko, H.F. and N. Nicolici, 2008. On automated trigger event generation in post-silicon validation. InProceedings of the conference on Design, automation and test in Europe, pp: 256-259.
10. Adir, A., M. Golubev, S. Landa, A. Nahir, G. Shurek, V. Sokhin and A. Ziv, 2011. Threadmill: A post-silicon exerciser for multi-threaded processors. InDesign Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE, pp: 860-865.
11. Nahir, A., A. Ziv, R. Galivanche, A. Hu, M. Abramovici, A. Camilleri and S. Kapoor, 2010. Bridging pre-silicon verification and post-silicon validation. InProceedings of the 47th Design Automation Conference, pp: 94-95.
12. Singerman, E., Y. Abarbanel and S. Baartmans, 2011. Transaction based pre-to-post silicon validation. InDesign Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE, pp: 564-568.