Scientific Research Publishing

Optics and Photonics Journal, 2016, 6, 261-268
http://www.scirp.org/journal/opj
ISSN Online: 2160-889X
ISSN Print: 2160-8881

Unified Coverage Methodology for SoC Post-Silicon Validation

Semih Aslan[1], G. Karuna Ranganathapura Chandrai[1], Vittal Siddaiah[2]

[1]Ingram School of Engineering, Electrical Engineering, Texas State University, San Marcos, TX, USA
[2]Intel Corporation, Austin, TX, USA
Email: aslan@txstate.edu, k_r251@txstate.edu, vittal.siddaiah@intel.com

How to cite this paper: Aslan, S., Chandrai, G.K.R. and Siddaiah, V. (2016) Unified Coverage Methodology for SoC Post-Silicon Validation. *Optics and Photonics Journal*, **6**, 261-268.
http://dx.doi.org/10.4236/opj.2016.610026

Received: September 5, 2016
Accepted: October 15, 2016
Published: October 18, 2016

Abstract

The System-on-Chip's increased complexity and shortened design cycle calls for innovation in design and validation. A high quality System-on-Chip creates distinction and position in the market, and validation is the key to a quality product. Validation consumes >60% of the product cycle. Therefore, validation should be carried out efficiently. Validation must be quantified to aid in determining its quality. Pre-silicon uses various coverage metrics for quantifying the validation. The available on-chip coverage logic limits the use of pre-silicon-like coverage metrics in post-silicon. Although on-chip coverage logic increases observability, it does not contribute to the functional logic; hence, they are controlled and limited. Discounting the need for the on-chip coverage logic, the question to be answered is whether or not these pre-silicon coverage metrics applicable to post-silicon. We discuss the reasons for limited applicability of pre-silicon coverage metrics in post-silicon. This paper presents a unified SoC post-silicon coverage methodology centered on functional coverage metrics.

Keywords

SoC, IoT, Coverage Metrics, Post-Silicon

1. Introduction

Technology devices consumed around the globe are the key drivers of the semiconductor industry. Figure 1 depicts the key components of this industry. A System-on-Chip (SoC) contains the intelligence of products, including mobile phones, Internet of Things (IoT) based devices, laptops, personal computers, tablets and servers. The ever-shrinking market window has shortened the SoC design cycle. This reduction calls for innovative design automation solutions. The SoC industry relies heavily on electronic design automation (EDA) tools to automate the product development and va-

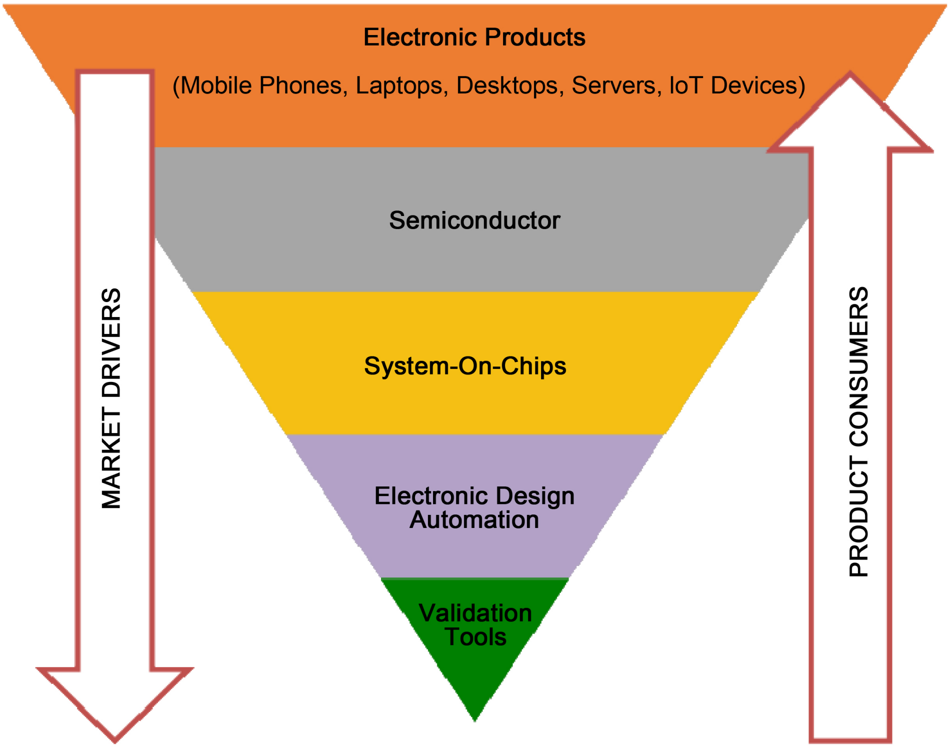


Figure 1. Interconnected market propellers.

lidation steps. Achieving desired results with lowered cost and time determines the efficiency of these EDA tools. Efficient tools can significantly reduce the design cycle.

SoC's approach of integrating an entire system on a single piece of silicon has increased the complexity of design and validation [1] [2]. Validation of the SoC is carried out in both pre-silicon and post-silicon phases. Pre-silicon addresses the validation of the hardware logic in its definition. This step, also called white-box testing, allows for high visibility into the design for debugging. Although debugging a failure is easier in this environment, limited simulation speed restricts the number of tests. A manufactured chip's functional validation is carried out in the post-silicon phase. The post-silicon validation also known as black-box testing has limited visibility in the design. Hence, debugging a failure in this environment is laborious and time-consuming [1].

The goal of validation is to deliver a bug-free product by exercising every cell of the design. Validation must be quantified to achieve that purpose. Coverage metrics quantify the validation. Capturing coverage data are more viable options in pre-silicon as compared to post-silicon. The design must include additional coverage monitoring hardware to enable coverage analysis in post-silicon. On-chip coverage logic increases observability, but does not contribute to the functional logic; hence, it is controlled and limited [3]. Qualitative post-silicon validation can significantly reduce costs and enable shorter time-to-market.

Efforts pursued in improving post-silicon validation can be categorized into two main approaches. First, to improve the quality of validation by enhancing bug-hunting methods, such as the hybrid quick error detection technique [4] and by improving the on-chip debug capabilities [5]-[7]. Second, to reduce the on-chip logic required for coverage monitors [3] [8]-[10]. However, much work must be carried out in defining the coverage metrics specific for post-silicon.

Pre-silicon uses traditional coverage metrics [9]. Not all of these traditional coverage metrics apply to post-silicon. Statement coverage, branch coverage, and path coverage (details in Section 2) measure the extent of RTL code execution. Post-silicon contains logic equivalent to RTL code. Hence, pre-silicon-like coverage metrics will not yield the right measure of coverage in post-silicon. As post-silicon validation mainly focuses on functionality, functionality-based coverage metrics will add value to coverage analysis. In this paper, we review the existing pre-silicon coverage metrics and provide reasons for their limited applicability in post-silicon. We present functionality-based coverage metrics as a viable option for post-silicon coverage analysis.

This paper contains six sections. Following the introduction, Section 2 describes the need for coverage. Section 3 outlines the pre-silicon coverage metrics. Section 4 explains the limitations in using pre-silicon coverage metrics in post-silicon and articulates the proposed use of functionality-based coverage metrics. Section 5 details the proposed unified methodology for functionality-based coverage metrics usage in post-silicon.

2. Coverage

Design and validation consume a majority of the SoC development cycle. The design uses 40% of the time while validation utilizes 60% of the product cycle. As validation consumes a higher portion of the project costs, innovative solutions to reduce the validation cycle are in demand. To control or improve a process, it must be measurable. Hence, quantifying the validation is crucial to shortening the design cycle. How much validation is good enough? Millions of test cases exercise the SoCs before its release into the market. Are these tests necessary? If we can cut down on a portion of these tests, would we compromise the quality of the design? The validation methodologies followed in today's design are working towards answering such questions. Coverage analysis provides key metrics to quantify validation effectiveness [3] [8] [11].

3. Coverage Metrics in Pre-Silicon

Pre-silicon validation is carried out in several phases, including unit-level testing, interconnection tests, and full-chip testing [2]. Unit-level testing focuses on functional validation of individual modules. Basic coverage metrics including statement coverage and branch coverage provide the test coverage data. Thus, individually validated blocks are subjected to interconnection testing along with their neighboring blocks. Here, the path coverage will quantify inter-block communication. Further, the entire design undergoes the final step of full-chip validation. The preference for full-chip validation is functionality-based coverage metrics versus individual coverage metrics like statement and branch.

Coverage Metrics

Pre-silicon uses varieties of coverage metrics [12]. We will briefly review the prominent ones in this section.

Code coverage: Code coverage quantifies the lines of RTL code exercised [3] [12]. The main code coverage types include statement coverage, branch coverage, and path coverage.

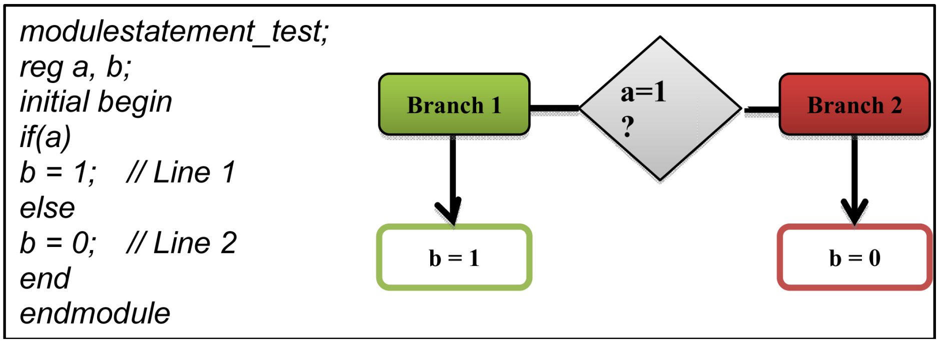
- Statement Coverage: This metric indicates if a line of code has been executed at least once by the tests. For example, a test case which forces the signal value of a = 1 in Figure 2 yields statement coverage of 50%.
- Branch Coverage: This metric indicates if all possible branches of a conditional statement including if-else-if, case, and for-loop, are executed [12]. Execution of "True branch" alone results in 50% branch coverage, as shown in Figure 2.
- Path Coverage: A path exists when two or more branch statements occur. Path coverage accounts for the execution of all possible combinational paths existing between two or more branches [12]. For example, if the code has two branch statements, each with two possible branches, they result in four paths as shown in Figure 3. Execution of all four paths in Figure 3 will result in 100% path coverage.

Coverage metrics evolved to address basic metrics' limited applicability to the functionality. Hence, functionality-based coverage metrics, including Finite State Machine (FSM) coverage, toggle coverage, and combinatorial logic coverage were introduced.

- FSM coverage: This accounts for exercising each state and all transitions of the given state machine.
- Toggle coverage: Toggle coverage accounts for a signal's transition in the order 1-0-1 or 0-1-0. In order for toggle coverage to be 100%, each signal must have completed the mentioned transition.
- Combinatorial coverage: This coverage metric assesses the combinations of input values applied to a combinatorial logic. For example, a two-input NAND gate's combinatorial coverage reaches 100% when all possible four combinations of the input are applied.

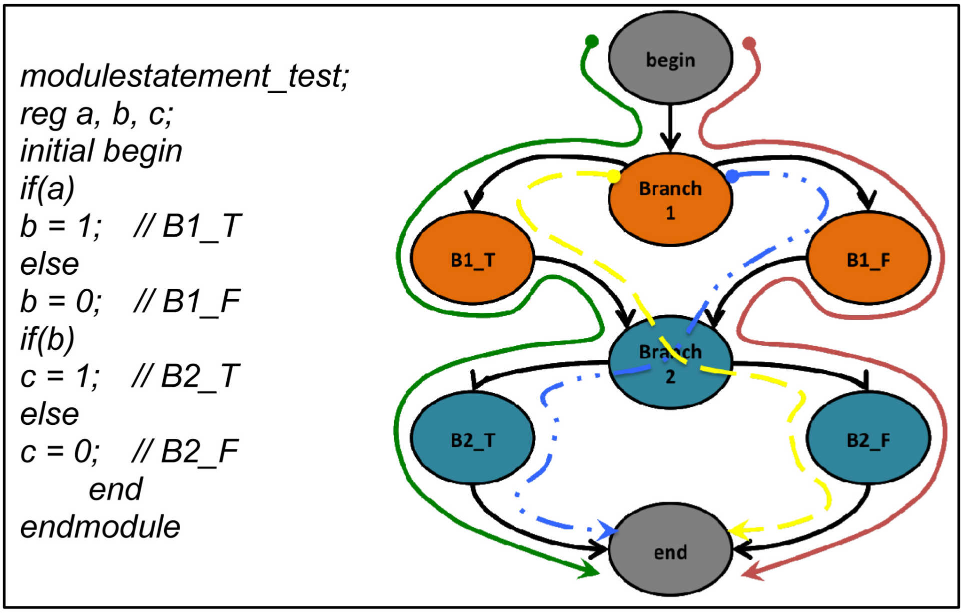
4. Coverage Metrics in Post-Silicon

Standardized coverage metrics is an emerging research area in post-silicon validation.



```
modulestatement_test;
reg a, b;
initial begin
if(a)
b = 1;    // Line 1
else
b = 0;    // Line 2
end
endmodule
```

Figure 2. Example of statement and branch coverage.



```
modulestatement_test;
reg a, b, c;
initial begin
if(a)
b = 1;   // B1_T
else
b = 0;   // B1_F
if(b)
c = 1;   // B2_T
else
c = 0;   // B2_F
      end
endmodule
```

Figure 3. Example of path coverage.

The use of pre-silicon-like code coverage metrics poses the following challenges in post-silicon:

- Code to logic: Silicon houses the hardware logic created from the RTL code. Hence, code-based coverage metrics, including statement, branch and path coverage will not be applicable. These metrics can only be explored as a proof-of-concept in post-silicon to demonstrate the possibility of capturing coverage based on such metrics, but would not add value for quantifying the validation.
- Logic and cost overhead: Additional on-chip logic is required to capture the coverage information and to transfer this information outside the silicon for analysis. Design and validation of this extra on-chip coverage logic will increase the design time and cost.

For facilitating efficient post-silicon validation, coverage metrics must quantify the functionality. The key challenge in this approach is to derive generalized coverage metrics applicable to functionally differing units. Generalization of functionality is proposed based on the data-flow among the interconnected blocks in a SoC. The data flow-based coverage metric measures the data streams between nodes. All tests aimed at cross-functional validation force data flow. For example, with direct-memory-access (DMA), data are transferred from the memory to a device connected to the universal serial bus (USB). Therefore, this metric generalizes the functionality to the flow of data among different functional blocks on-chip. Therefore, validation effectiveness can be quantified using such a metric in post-silicon.

5. Unified Coverage Methodology

The proposed unified coverage methodology centers on data-flow coverage analysis. Figure 4 captures the implementation of this methodology. This solution consists of

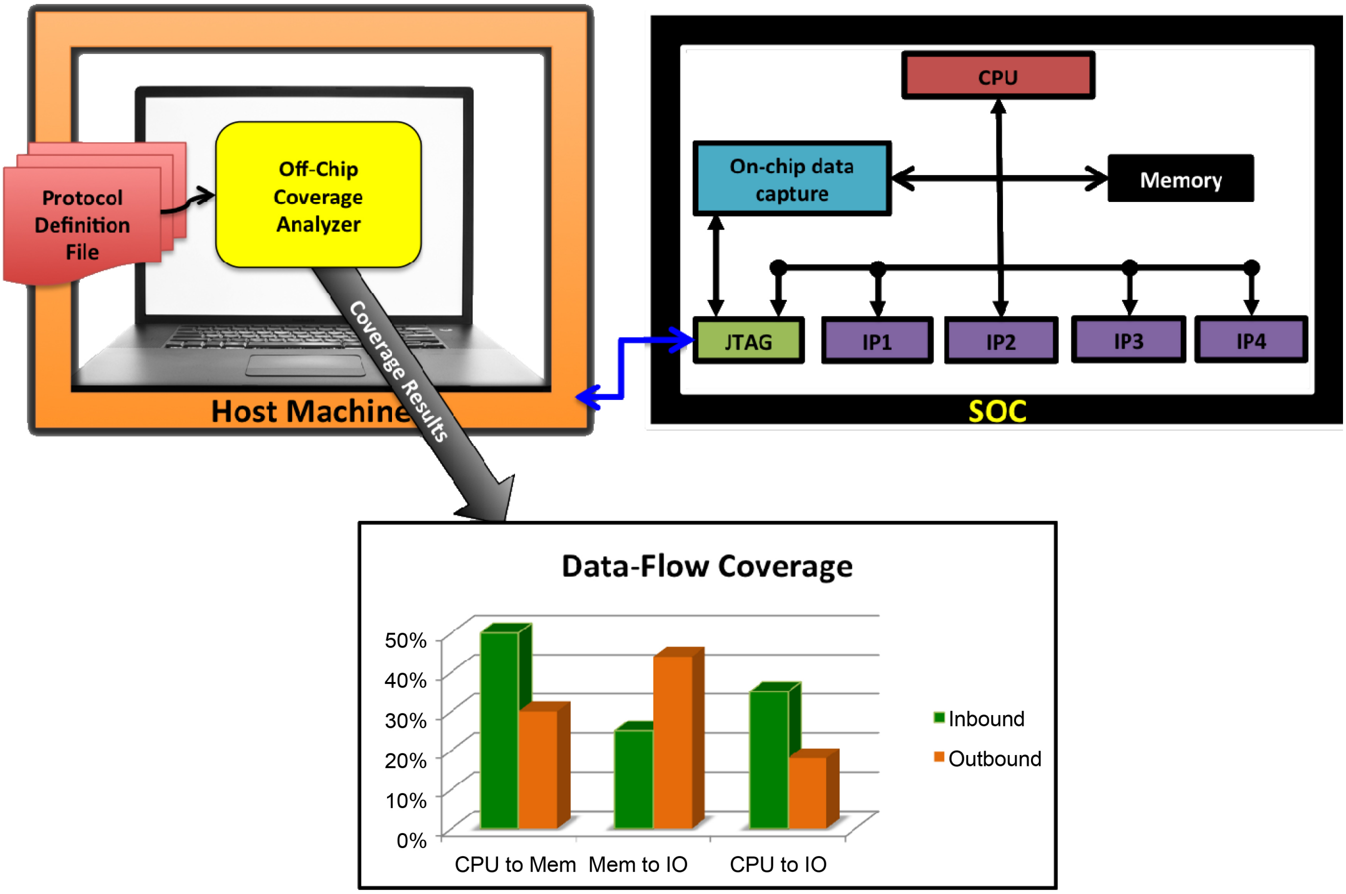


Figure 4. Unified coverage methodology.

two main components-the On-chip Data Capturing (ODC) unit and the Off-chip Coverage Analyzer (OCA) software. The ODC designed as an intellectual property (IP) connects to the bus of interest and captures the transactions. The ODC communicates the captured data to OCA software through JTAG port. The software processes the received data by applying the rules defined in the protocol definition file. Thus, processed data represent the coverage achieved. Application of coverage targets as an input to the OCA will result in accurate coverage calculations. Intuitive GUI presents details of the coverage results with a click on the displayed graph. The details will articulate the different types of transactions generated by the tests and also will highlight the missing data-flows. The validation engineer can then fine-tune the tests to achieve the missing coverage.

The proposed methodology is easily prototyped on an FPGA using existing technologies. Xilinx^{TM} ChipscopePro^{TM} on-chip Integrated Logic Analyzer (ILA) core can perform the function of ODC [13]. The ILA core is a customizable logic analyzer core that can be used to monitor any internal signal of the design. OCA software must be developed to communicate with the ODC through JTAG. OCA software can work for different protocols with a change in the protocol definition file. However, the ODC must be designed to operate with differing bus protocols. Many SoCs use standardized bus protocols. Hence, a predictable set of bus interfaces is implementable in the ODC.

6. Conclusion

Coverage analysis is widely used in the pre-silicon environment to quantify validation. Such analysis is a must for post-silicon, but pre-silicon coverage metrics are not readily applicable. In line with the proposed data-flow-based coverage metric, further research must be conducted to innovate coverage metrics relevant to post-silicon. Such research will enable a reduction in post-silicon validation cycles and enhance validation effectiveness. Coverage for post-silicon has to be given the utmost importance, as the shortened market window has forced reusability. Due to a shorter market window, spinning new SoCs with multiple new features is risky. Successive SoCs differ in only a few features, and this calls for a focus on ECOs validation rather than end-to-end validation. Quantifying validation through coverage metrics will contribute to re-focusing validation efforts on ECOs, thereby reducing the cost and shortening the SoC development cycle. The proposed methodology of functionality-based coverage will support quantification of the post-silicon validation effectiveness. This idea will be researched further for efficient ODC and OCA implementation.

References

[1] Goodenough, J. and Aitken, R. (2010) Post-Silicon Is Too Late Avoiding the $50 Million Paperweight Starts with Validated Designs. 2010 47*th ACM/IEEE Design Automation Conference* (*DAC*), 8-11. http://dx.doi.org/10.1145/1837274.1837279

[2] Keshava, J., Hakim, N. and Prudvi, C. (2010) Post-Silicon Validation Challenges: How EDA and Academia Can Help. 2010 47*th ACM/IEEE Design Automation Conference* (*DAC*), 3-7. http://dx.doi.org/10.1145/1837274.1837278

[3] Balston, K., Karimibiuki, M., Hu, A.J., Ivanov, A. and Wilton, S.J.E. (2013) Post-Silicon Code Coverage for Multiprocessor System-on-Chip Designs. *IEEE Transactions on Computers*, **62**, 242-246. http://dx.doi.org/10.1109/TC.2012.163

[4] Campbell, K.A., Lin, D., Mitra, S. and Chen, D. (2015) Hybrid Quick Error Detection (H-QED): Accelerator Validation and Debug Using High-Level Synthesis Principles. 2015 52*nd ACM/EDAC/IEEE Design Automation Conference* (*DAC*), 1-6. http://dx.doi.org/10.1145/2744769.2753768

[5] Nicolici, N. (2012) On-Chip Stimuli Generation for Post-Silicon Validation. 2012 *IEEE International High Level Design Validation and Test Workshop* (*HLDVT*), 9-10 November 2012, 108-109. http://dx.doi.org/10.1109/hldvt.2012.6418251

[6] Lisherness, P. and Cheng, K.T. (2009) An Instrumented Observability Coverage Method for System Validation. *IEEE International High Level Design Validation and Test Workshop*, 2009. *HLDVT* 2009, 4-6 November 2009, 88-93. http://dx.doi.org/10.1109/hldvt.2009.5340171

[7] Abramovici, M. (2008) In-System Silicon Validation and Debug. *IEEE Design & Test of Computers*, **25**, 216-223. http://dx.doi.org/10.1109/MDT.2008.77

[8] Bojan, T., Arreola, M.A., Shlomo, E. and Shachar, T. (2007) Functional Coverage Measurements and Results in Post-Silicon Validation of Core^{TM}2 Duo Family. *High Level Design Validation and Test Workshop*, 2007. *HLVDT* 2007. *IEEE International*, 7-9 Novem-

ber 2007, 145-150. http://dx.doi.org/10.1109/hldvt.2007.4392804

[9] Karimibiuki, M., Balston, K., Hu, A.J. and Ivanov, A. (2011) Post-Silicon Code Coverage Evaluation with Reduced Area Overhead for Functional Verification of SoC. *High Level Design Validation and Test Workshop* (*HLDVT*), 2011 *IEEE International*, 9-11 November 2011, 92-97. http://dx.doi.org/10.1109/hldvt.2011.6113982

[10] Gallardo, R.O., Huy, A.J., Ivanov, A. and Mirian, M.S. (2015) Reducing Post-Silicon Coverage Monitoring Overhead with Emulation and Bayesian Feature Selection. 2015 *IEEE/ACM International Conference on Computer-Aided Design* (*ICCAD*), 2-6 November 2015, 816-823. http://dx.doi.org/10.1109/ICCAD.2015.7372655

[11] Mitra, S., Seshia, S.A. and Nicolici, N. (2010) Post-Silicon Validation Opportunities, Challenges and Recent Advances. 2010 47*th ACM/IEEE Design Automation Conference* (*DAC*), 12-17. http://dx.doi.org/10.1145/1837274.1837280

[12] (2004) Functional Verification Coverage Measurement and Analysis. Springer, Boston.

[13] Xilinx (2016) ChipScope Pro Software and Cores.