Verification Method for VLSI Design

Saikot Das Joy

Dept. of Electronic Engineering
Hochschule Hamm-Lippstadt
Hamm, Germany

***Abstract*—Before implementing any hardware, it is expected be checked to see how effective and functional it will be in the manufacturing process. There is also a need for testing after development. In this paper, we will look at different verification methods for VLSI(Very Large Scale Integration) design, which is an important part of VLSI design. This paper focuses primarily on functional verification and emulated verification methods to check their feasibility. The functions of several tools were then discussed. The testing and validation procedure was then thoroughly described with flow diagrams. We concluded by outlining some difficulties we encounter while using VLSI design techniques.**

***Index Terms*—Verification,Design,VLSI,Formal,Simulation, Testing,Validation**

I. Introduction

VLSI (very large-scale integration) is very difficult to understand for the engineer because of its complex system, model, and increasing number of circuits. There are many models for designing integrated complex circuits. But each model should be checked for its proper functionality before implementation. After the manufacturing process, it should also be tested [1]. The process that verifies the correctness and completeness of a design to fulfill a specification before manufacturing is called the verification method in VLSI design. These methods confirm the quality of the design as well as the quality of the manufactured device. We need verification at every step of the manufacturing process. If we find any errors, then the process should be returned to its initial state [2]. The verification method gives a higher level of abstraction. It uses its own language and consumes almost 70–80 percent of overall manufacturing time. Due to the complex systems, the integrated circuit design is difficult. A design verification strategy can be developed to achieve our desired goal. It will also be cost-effective if the design or model verification is done. Simulation can be done to check the design, and testing can be done to check for any hardware damage after manufacturing. In this paper, the methods of simulation-based verification and formal-based verification are described. It is a significant step forward in IC fabrication technology. Formal methods are important for reliable design verification. We can create a model that will be subjected to automated verification to ensure a reliable and fully functional design with no errors or hardware damage. The modeling must be identical to the logical representation or built model.Critical path analysis can also be determined through the simulation software as well as the determination of the correctness of the requested design. To ensure the correct operation, we need hardware verification, which can be called the backbone of the verification methods.

II. VLSI Realization Process

Product realization is the process of achieving a clear picture of the final product, which is typically represented by drawings, statements of work, functional product specifications, or their equivalents(ISO9001:2015). Setting a baseline without a product realization plan will be challenging. The quality plan adheres to quality goals and rules. Criteria for accepting goods and records are required in order to ensure this [3], [4].The realization process of VLSI starts with the customer's needs. When we do the verification, the design must be checked according to the customer's need. Product specification that describes the expected input-output are written. The formal

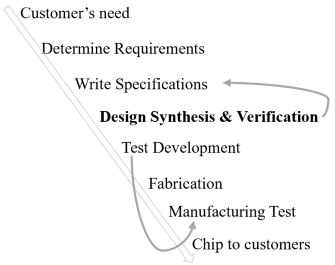

Fig. 1. VLSI Realization Process

diagram (As shown in Fig 1) of the realization process demonstrates that the most significant step in this process is design synthesis and verification. Let us understand the following-

• Design Synthesis: Let us assume, we have some given I/O function. Design synthesis is a procedure that should be developed based on this requirement using known materials and processes to manufacture a device.

• Verification: When manufactured a device, verification is a predictive analysis to ensure that the synthesized design will give us the requires I/O.

• Test: Testing is the manufacturing step that ensures that the physical device has no manufacturing defect which is designed from the synthesized design.

III. Role of verification of VLSI

Being a high-reliability process, VLSI is used widely in the modern era. We have already discussed the motive of

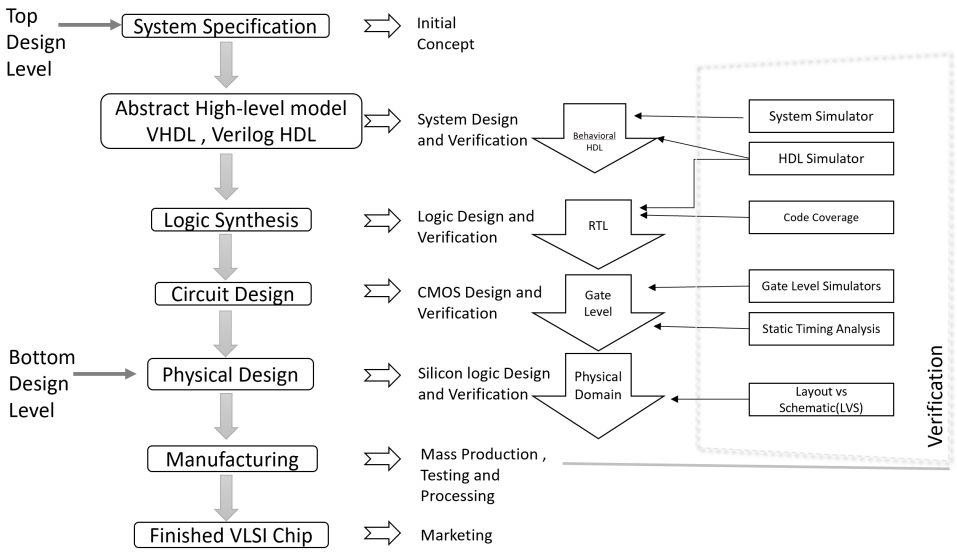

Fig. 2. Design Flow and Verification

our work. We will now extend our work in the light of VLSI design. Verification is the process where it confirms the standard of the product and implies the best way to check the failure of the instrument. One common definition of verification can be found in [ANSI/ASQC 1978] is “the act of reviewing, inspecting, testing, checking, auditing, or otherwise establishing and documenting whether or not items, processes, services or documents conform to specified requirements.” The conjugate process, design verification, and testing is a time consuming, it takes almost 80 % of the overall time during the manufacturing process. There is various factor that works on the basis of verifying and testing. But at the initial step, Verification is required. One can come back to the initial state after fixing any mistake in any level of work. The common two-type VLSI design is used in different articles which are found very effective [5], [6], [7].

1. Functional verification: In a typical design ICs design, functional verification conforms to the specifications. Due to the rapid growth of complexity and size design, functional verification plays an important issue. According to the current report, almost 70 % of the effort is consumed by functional verification and the number may raise constantly. The bug of any computer code can make complexity similar to the small bug of functional verification increasing the company’s loss [8].

There is three approaches for Functional Simulation :

(1) Black Box Approach

(2) White Box Approach

(3) Gray Box Approach

In the following table in Fig 3, This three types of Approaches are explained.

Black Box
unable to look into the design
To be carried out without internal implementation knowledge
exclusively through available interfaces; no access to internal state
White Box
intimate understanding and controls over a design's internals
By using this method, implementation-specific features can be ensured to behave properly.
using a pure white box approach at the system level: Modules are considered as black boxes while the system itself is treated like a white box when.
Gray Box
Black box test written with Full knowledge of internal details
Mostly written to increase code average

Fig. 3. Black Box, White Box & Gray Box Approaches

2. Static Timing Analysis (STA): It is the path of design timing that validates the timing performance. STA is commonly used to determine signal delay and timing performance violations. It determined the full circuit behavior with the best vector outputs [9].

There are various methods of describing VLSI design verification. The commonly used methods are the functional method, performance verification, power-aware method, clocking, CDC verification, DFT verification, FPGA/emulation, HW co-verification, etc. Among them, the functional verification method and emulation are the most widely used methods. The functional method has two sub-units called ”simulated” and ”formal-based” verification.

We will go over these methods in depth in this paper. An overall VLSI design verification procedure with VLSI design flow is given on Fig 2 .

IV. Simulation and Formal Based Verification

To imply the design flow intensively, we should understand the process of different methods and from this, one can pick up the best methods for design. There are various methods used for verification testing. These methods are based on the function of the work processing and confirm how they work very inexpensively and quickly.

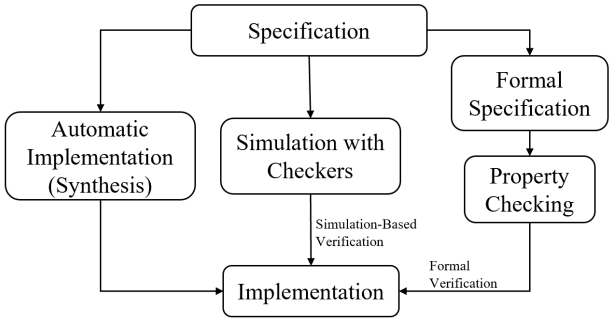

Fig. 4. Comparing design flow between Simulation and Formal Based Verification

Compared with others methods, we should choose the method which is low-cost and best for the company.The evaluation between simulation verification and formal verification in light of functional verification is shown in Figure 4. That depicts how these two processes are related and, after completing some steps, how they meet in the final implementation. Simulation-based verification is a widely used process that is sometimes called an error-free method.

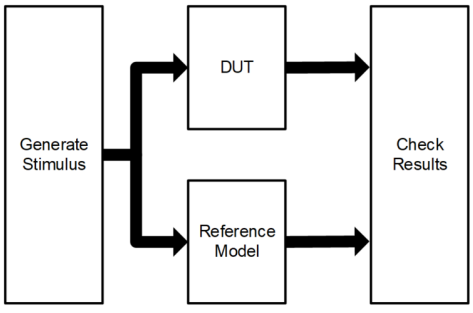

Fig. 5. Simulation based verification flow

The function of the SBV is to generate a new vector and derive the output (As shown in Fig 5). In this sense, sometimes this method is called the input-driven method. Confirming a point in the input space is systematically related to simulating a vector. This view proposes input space sampling as a tool for simulation-based data verification. There is a probability that a mistake will go undetected if all points are not sampled. The function of SBV says that only one output point is checked at a time via simulation-based verification, which sometimes makes this process facile to use. as example, we use EDA type tools for simulation based verification [10], [11].

Formal verification refers to a group of methods that use static analysis based on mathematical transformations (RTL design) to determine whether hardware or software behavior is legitimate, as opposed to dynamic verification methods like simulation (As shown in Fig 6). By using various techniques, verification minimizes the number of vectors required for the system's operation to an acceptable level. One of the tools used to generate this work is formal verification. Formal verification is very fast, so no potential state is required to demonstrate that the logic holds up; the provided information demonstrates how to set the properties in all circumstances. However, its effectiveness is greatly influenced by how it is utilized and the kind of logic it is applied to.Users have been urged to embrace formal verification in more concentrated methods owing to safety risks and communication issues like a stalemate in on-chip networks [12], [13].

For instance, formal verification tools, such as the security software created by Jasper Design Automation, are more effective than simulation at checking for logical sneak paths that could compromise security. When focused formal verification is utilized, the vendor will combine several mathematical procedures using scripts created specifically for the situation. Compared to the more versatile items created, these and others can be offered as tools that are simpler to use.

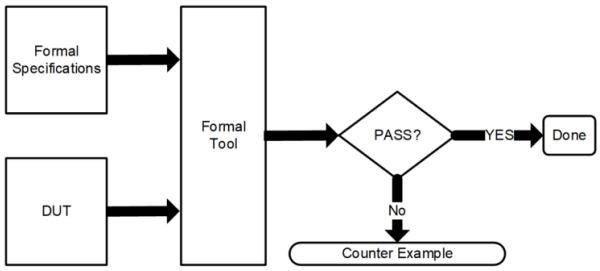

Fig. 6. Formal verification method flow

In short for the design flow, simulation-based verification can be applied for generating the vector output.Its an error free design. At a time, it checks one output point. It has an advantage of excellent controllability and disadvantage of extremely slow. Formal verification is limited to the small design. At a time, it checks a group of output points. In Fig 7, we see that the simulation-based method works where data points need to be verified, but formal verification verifies the properties of the prototype for VLSI circuits. The formal verification techniques are:

1. Deductive Verification: It proves the system's correctness.
2. Model Checking: It proves the correctness of the concurrent system.
3. Equivalence Checking: It checks if two circuits are equiv-

alent.

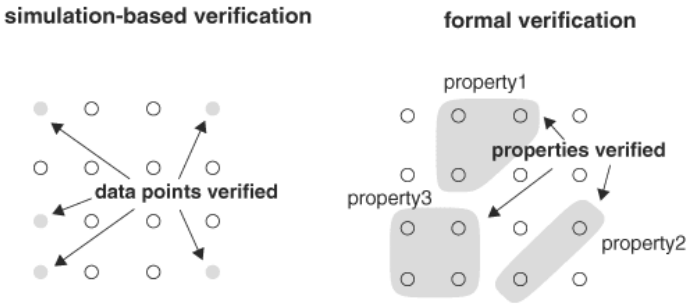

Fig. 7. showing the properties of simulation-based verification and Formal verification

V. Emulated Verification

Emulation is the method of performing the functional verification of the hardware and software of the design by automatically mapping the design's RTL representation to its internal programmable gate array using a specialized computer. Sometimes it's called an emulator. The motivation behind creating emulation verification is to get effective manufacturing and get counting in design. Since it provides information about synthesis, power, timing, area, and other constraints that are not covered by simulation verification, this process improves our level of detail and understanding of the chip. For a better understanding of the functionality, we usually emulate the design into an FPGA [14], [15].

VI. Tools for VLSI VERIFICATION

Tools for verification are efficient. A variety of tools are available for verification. Tools are dependent on how the device that initiates the various verification processes is implemented. A software-like tool called electronic design automation (EDA) aids in the creation of electronic circuits. This software's primary focus is large-scale circuit design, and various businesses employ these tools in subpar ways. VLSI design tools assist various businesses in creating digital versions of their chips and other tools, as well as a method of shrinking chip size and speed so that we can use them on our laptops and smartphones.There are some well-known companies that offer free VLSI verification tools. Sometimes we need licenses to use these tools. In the synthesis of circuit design, implementation, and simulation of complex designs, among other things, tools are helpful [16], [17]. Some companies and their tools are listed in Table I

TABLE I
Different types of tools for VLSI

Companies	Famous Tools
Synopsys	primetime
Cadence	Innovus
Mentors grapics	Calibre
Ansys In.	Power Artist,VeloErf,Ansoft HFSS
keysight Technologies Inc	Pathwave
Electric	Electric VLSI Design System
Alliance	Coriolis VLSI CAD tools
Zuken Ltd.	CR-5000, CR-8000, Visula
Lauterbach GmbH	TRACE32
Aldec Inc.	Riviera-PRO,Active-HDL
AGNISYS Inc.	IDesignSpec,IVerifySpec

Tools with work function

Primetime	- Static timing analysis tool - Ensuring design integrity - Accurate signoff analysis - Power and variation aware analysis
Innovus	- Design, stimulate and synthesis - Design rule check with verification facilities - lay out and parasitic capacitance verification - available in working
Mentor Graphics	- Creates advanced IC's - Optimal balance power facilities - validation process active in all levels - have the Si life cycle method for ensuring safety -Deliver Coverage, and high resolution
Power Artist	-generate CAD and CAM - Design rule checking process fast and efficient data sharing - Use in multiphysics simulation
Pathwave	- Electromagnetic simulation -Yielding with optimization -To make nonlinear design -visualization of the physical layout
CR-500	- used in design for Ic's complex speed boards -improve in signal integrity -electromagnetic compatibility

VII. VLSI testing

VLSI testing is done in various ways by different people and has an impact on the size of the chip, among other parameters. When we introduce new chips, they should be tested to confirm their correctness and verify the integrity of the process.How do they verify and get the test result efficiently is shown in Fig 8.

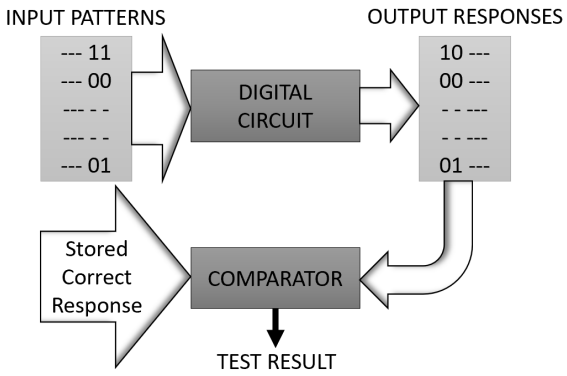

Fig. 8. VLSI testing flow

Testing is one of the most effective methods of ensuring chip quality, and we can improve the process for the upcoming VLSI version.Successful verification testing is the result of finding chips that might be good, and additionally, we can exclude the item that is faulty. The manufacturing process approaches successful verification, and that indicates the large-scale production rate for a company. Independent testing is used to perform incoming inception for the user. The test is done by applying the new inputs to the circuits. If we get the matching result, we will call it a good response. That good response is the main factor for testing [18], [19]. There are four types of VLSI testing processes [20]. (As shown in the fig 9).

1. Characterization: It is a design debug verification process before being sent to production that verifies all specifications are correct. SEM and AI are some common tools used in testing that depicts the limits of operating values. We measure the data statistically to ensure the testing by taking average data or getting pass/fail results indicates the good testing format.

2. Production: Production tests, which are performed on every fabricated chip but are less thorough than characterization tests, are still required to maintain quality standards by determining whether the device meets specifications

3. Burn-In: the production reliability of chips should be checked by the Burn in process. We put the chips through a combination of production tests, high temperatures, and over-voltage power supplies during burn-in.

4. Incoming inception: This testing's primary objective is to avoid installing a defective component in a system assembly, where diagnosing the problem could cost much more than performing an initial inspection.

Aside from that, all potential defects that occur during the manufacturing process are subject to "defect-based testing." If the chip has all of the functionality we need, then we can call it a good chip.

• Some good chips being rejected – yield faults
• Some bad chips being shifted – defect level

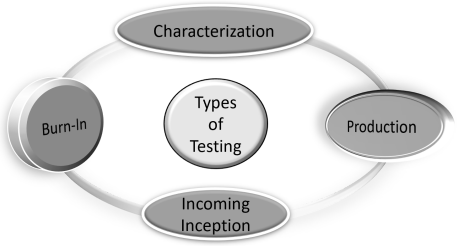

Fig. 9. Types of VLSI Testing

A. Verification and Validation

The process of validating of VLSI (the finished product) involves determining whether it truly satisfies the needs and expectations of the customer. This process is an output type even though it is only for the needs of the customer. Although it may never be relevant to the internal design of a circuit, it aids in the technical difficulties of the development process [21]. The procedure will start once the verification process is over. Validation is the main focus of all dynamics testing procedures. It includes some testing to help find the bug that the verification method did not pick up on. It employs various testing techniques for this, including functional testing, system testing, integration testing, etc. the testing flow of validation is shown in the Fig 10.

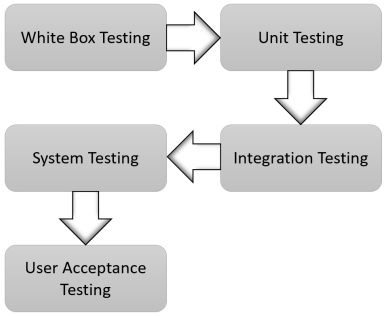

Fig. 10. validation process

VIII. Design Verification Model and Challenge

If we have specification those are created from the customers requirement and we want to implement the system or circuit, this implementation should respect the specification.That means there should be equivalence between the specification and implementation. There are couple of synthesis steps. It has to go from specification to implementation. Most of the time is consumed by design verification.The methods we described in this paper have some drawbacks, and to overcome them, we may face some unthinkable challenges.The main issue with these verification methods is reducing verification time. Each device has unique ease characteristics that relate to the functionality, accessibility, performance potential, security, and other aspects of the device. By utilizing approaches that are configurable and scalable, we can resolve this issue. The

tool's constrained scalability is yet another significant obstacle. In Table listed in Fig 11 we see yearwise evoultion of VLSI design.

Year	1995	1998	2001	2004	2007	2010
Feature size (μm)	0.35	0.25	0.18	0.13	0.10	0.07
	5M	14M	26M	50M	210M	430M
DRAM bits/chip	64M	256M	1G	4G	16G	64G
Number of I/O	900	1350	2000	2600	3600	4800
Chip-to-board speed	150	200	250	300	375	475
Maximum number of wiring levels (logic)	4-5	5	5-6	6	6-7	7-8
On-chip speed (MHz)	300	450	600	800	1000	1100

Fig. 11. Evoulution of VLSI

The formal verification method essentially deals with these issues. Things have only gotten worse as a result of stricter configuration requirements and the ensuing increase in design complexity.We should make proper use of them and pay attention to the initial design to address this. The power management challenge is a critical challenge that comes into contact on all VLSI verification types. Despite the significant challenges, the issue can be resolved in a different way that simplifies our VLSI design, and if we look at the evaluation in the chips category, we can be sure that we will soon overcome one such obstacle.

IX. Conclusion

It is challenging to design integrated circuits because of complicated systems.A design verification strategy can be developed to achieve our desired goal. It will also be cost-effective if the design or model verification is done.For developing integrated complicated circuits, there are numerous models available.Prior to implementation, each model should be examined for adequate functioning.It needs to be tested as well after the production process. In addition to describing various approaches for the VLSI design in this work, we also discussed strategies for accessibility, the tools required for design, and a justification for validation and testing.

References

[1] Fujita, Masahiro, Indradeep Ghosh, and Mukul Prasad. Verification techniques for system-level design. Morgan Kaufmann, 2010.

[2] Shostak, R. E. (1983). Verification of VLSI designs. In Third Caltech Conference on Very Large Scale Integration (pp. 185-206). Springer, Berlin, Heidelberg.

[3] Arya, N., & Singh, A. P. (2017). Comparative analysis of time and physical redundancy techniques for fault detection. Indonesian Journal of Electrical Engineering and Computer Science, 6(1), 66-71.

[4] Abdullah, A. C., & Ooi, C. Y. (2013). Study on Test compaction in high-Level automatic test pattern generation (ATPG) platform.

[5] Petrov, A. B., & Tarasov, I. E. (2019, October). About the features of the verification of VLSI class "System on a chip" for complex information systems. In Journal of Physics: Conference Series (Vol. 1333, No. 2, p. 022011). IOP Publishing.

[6] Loupis, M. I., & Tziallas, G. D. (1990). A knowledge-based framework for VLSI design. Microprocessing and microprogramming, 28(1-5), 327-331.

[7] Dahiya, P., Saini, J. S., & Kumar, S. (2007). Evolutionary Trends in VLSI Design. In Proc. of International Conference on Intelligent Systems & Networks (IISN-2007).

[8] Wen, H. P. C., Wang, L. C., & Cheng, K. T. T. (2009). Functional verification. Electronic Design Automation, 513-573.

[9] Liu, B. (2009, October). On VLSI statistical timing analysis and optimization. In 2009 IEEE 8th International Conference on ASIC (pp. 718-721). IEEE.

[10] Panda, S. K., Roy, A., Chakrabarti, P. P., & Kumar, R. (2008). Simulation-based verification using Temporally Attributed Boolean Logic. ACM Transactions on Design Automation of Electronic Systems (TODAES), 13(4), 1-52.

[11] Kaergel, M., Olbrich, M., & Barke, E. (2014, September). Simulation based verification with range based signal representations for mixed-signal systems. In Proceedings of the 27th Symposium on Integrated Circuits and Systems Design (pp. 1-7).

[12] Karmakar, R. (2022). Formal verification techniques: A comparative analysis for critical system design. In International Conference on Intelligent Systems Design and Applications (pp. 93-102). Springer, Cham.

[13] Abate, A. (2017, September). Formal verification of complex systems: Model-based and data-driven methods. In Proceedings of the 15th ACM-IEEE International Conference on Formal Methods and Models for System Design (pp. 91-93).

[14] Druml, N., Menghin, M., Steger, C., Weiss, R., Genser, A., Bock, H., & Haid, J. (2013, February). Emulation-Based Test and Verification of a Design's Functional, Performance, Power, and Supply Voltage Behavior. In 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (pp. 328-335). IEEE.

[15] Koczor, A., Matoga, Ł., Penkala, P., & Pawlak, A. (2016, April). Verification approach based on emulation technology. In 2016 IEEE 19th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS) (pp. 1-6). IEEE.

[16] Bouyssounouse, B., & Sifakis, J. (2005). Tools for verification and validation. In Embedded Systems Design (pp. 72-84). Springer, Berlin, Heidelberg.

[17] Garcia, P. J., Pruteanu, A., van Driel, W. D., van Kooten, W., & Linnartz, J. P. M. G. (2016). Tools and methods for testing and verification. In Runtime reconfiguration in networked embedded systems: design and testing practices (pp. 133-136). Springer.

[18] Wang, L. T., Chang, Y. W., & Cheng, K. T. T. (Eds.). (2009). Electronic design automation: synthesis, verification, and test. Morgan Kaufmann.

[19] Wen, X. (2011, July). VLSI testing and test power. In 2011 International Green Computing Conference and Workshops (pp. 1-6). IEEE.

[20] Bushnell, M. L., & Agrawal, V. D. (2002). VLSI Testing Process and Test Equipment. Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits, 17-34.

[21] Preece, A. (2001). Evaluating verification and validation methods in knowledge engineering. In Industrial Knowledge Management (pp. 91-104). Springer, London.