A Methodology to Decrease Product Development Time While Increasing Product Quality

John M Acken
OSU-Tulsa
acken@okstate.edu

Abstract

The technology-based markets place an increasing pressure to reduce the product development time, where the time is measured from the beginning of gathering requirements through the delivery of volume production. Concurrently, the markets are demanding an increase in the quality of shipped products. The time required to raise production quality is in conflict with the urgency to deliver a first product. This paper describes a methodology that combines Rapid System Prototyping (RSP) with Built-In Self-Testing (BIST) signature analysis to meet these conflicting requirements. Rapid System Prototyping is used to evaluate ideas, demonstrate feasibility, and refine requirements in order to ship a quick first product. The goal of production testing is to deliver a high quality product to the customer. This paper describes how to add BIST to RSP thereby creating a methodology that adds quality and shortens product delivery. This paper presents an example of a system developed with this new methodology. We demonstrate that BIST signature analysis can test the prototypes, do design verification, and provide for quality final product manufacturing test.

1. Introduction

In general, the earlier in the product development process that a design issue can be analyzed successfully the better. For example, a system level methodology to analyze power requirements as early as the software implementations of algorithms has been proposed [1]. As the authors noted, early analysis does not remove the need for addressing the design issue later in the product development process. However, it does allow for more efficient solutions and methods. The march of technology, as demonstrated by Moore's Law, robotics, and communications, pushes the market place to continuously develop faster, better, and cheaper products. The technology-based markets are pressuring suppliers to reduce product development time, where the time is measured from the beginning of gathering requirements to the delivery of volume production. However, the markets are pressing for an increase in the quality of shipped products. The time required to raise production quality is in conflict with the urgency to deliver a first product. One barrier to fast product development is the gaps between the various tools used for various stages. Efforts continue on smoothing the process by creating an environment that seamlessly links disparate tools together [2]. Similarly, an approach to a seamless environment was created for developing, evaluating, and integrating existing tools and new tools [3]. One effort to recapture early development effort is through reuse [4]. The proposed methodology in this paper utilizes reuse in the sense of reusing the signature analysis that at first is for design checking, in the middle stages is used for data checking, between stages it is used for design verification and in the final product it is used for production testing. Thus a "concept" or "idea" is reused in the form of a polynomial and the signatures.

2. BIST signature analysis and production testing

As the number of circuits that can be integrated into one piece of silicon exceeds several million, a portion of those can be devoted to testing the device. The capability of a circuit (chip, board, or system) to test itself is known as Built-In Self-Test (BIST). A fully self-testing circuit generates it own inputs and measures its own outputs to determine whether it is working [6]. Figure 1 shows the general block diagram for a self-testing device, including a block connected as a signature analyzer [7, 8, 9]. The shaded area includes the part added for self-test. Notice that self test can be software, hardware,

or a mix. In general, to utilize self testing there must be a test mode where the inputs are selected from the test block. However, when only signature analysis is implemented the test module can be passive and always monitoring the device under test. The remainder resulting from polynomial division in Galois Field 2 is an ideal hashing function. Polynomial division can be implemented in software or hardware. These hashes can be compared between prototypes and versions to add confidence in the design verification efforts. These signatures are supplementary and are not intended to replace, nor can it replace, other design verification efforts.

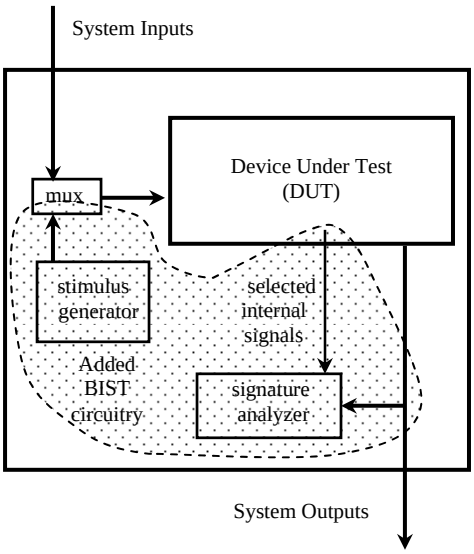

Figure 1. Built-In Self-Test.

In hardware the implementation is with a Linear Feedback Shift Register (LFSR), which uses the remainder left in the register after completion of the test as the retained statistic for comparison with the good remainder. It is an extension of the well-known Cyclic Redundancy Check (CRC) code and it is easily modified for use with multiple output circuits. The remainder is usually called a signature and the technique is called signature analysis, a name coined by Hewlett Packard and first used in Frohwerk (1977), [20]. Any data such as the test response results from a circuit or the variable values in a software routine can be compressed into a code word by an LFSR. This code word, the remainder from the division process, is called the signature of the input data stream. The LFSR itself is called the signature analyzer. Signature analysis using selective feedback of various stages of a shift register fed by the data stream being created by a circuit is a powerful technique for coping with a large volume of built in test response data. Signature analysis is now a well-accepted technique in industry for compressing the Device Under Test's (DUT) output responses in Built-In Self Test designs [10, 11]. An LFSR used as a signature analyzer maps the output vector space into a signature. Since the mapping is not one-to-one, it is possible for the signature of a faulty circuit to match the signature of a fault-free circuit. This is called aliasing [11, 12, 13]. The length of the LFSR is chosen to meet a minimum acceptable level of aliasing [14, 15, 16, 17].

3. Incorporating BIST signature analysis into design methodology

Traditional product development was labeled the waterfall model because as each stage was completed the results were passed to the next stage without any return, much as water flows down a waterfall. There have been many improvements upon this model including the spiral model of continuous improvement and concurrent engineering where the tasks of each design stage are considered during the previous design stage. The general time frame for the stages of product development is shown in Figure 2. Early in the product development times line there are many informal uncertainties, such as actual requirements, customer understanding, and technical feasibility. During the design stages many decisions are made based upon the discovery by and the understanding of the previous stages. The later stages have more formal information, and both methodologies and tools exist to link design, test, and manufacture. Albeit the job is not complete and the rapid pace of technology frequently leaves the existing tools wanting.

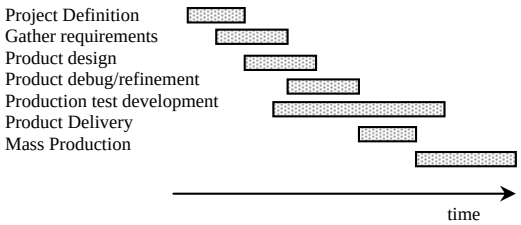

Figure 2. Product Development timeline.

The environment of the informal stages, the formal stages, and even the changes due to time tend to prevent linking the information available at the various stages. This includes complete disparate data representations as well as the fundamental nature of vague, unconstrained, conflicting, and even incorrect concepts in the early stages versus the very constrained and specific information. A hashed signature allows a unifying theme for information checking among very disparate stages. As Figure 3 demonstrates, each stage can do its analysis within the appropriate implementation framework, and the resulting signatures can be compared across all the borders.

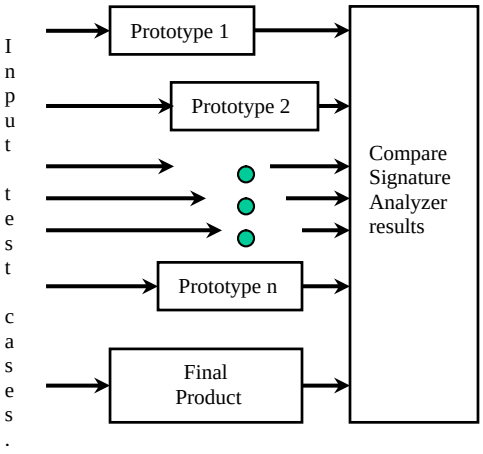

Figure 3. Using BIST signature analysis to check and verify different prototypes.

Although Figure 3 shows the pieces as prototypes, the blocks to be analyzed can include various models, such as Register Transfer Language (RTL) or Verilog logic circuits. The signatures can be formally compared via tool links or via data file comparisons for incompatible models and prototypes.

4. Rapid system prototyping

The process of gathering product requirements and defining a product is the least formalized part of the product development process. As product requirements are being gathered, the feasibility of new solutions must be measured. Rapid prototyping refers to the capability of creating a prototype with significantly less time than it takes to produce an implementation for operational use. With increased pressure to shorten time-to-market logic designers find FPGA solutions attractive [18]. As requirements are gathered and prototypes are created and examined, not only are strengths and weaknesses of alternative designs identified, but omissions and discrepancies in the requirements can be found [19]. One requirement is that the prototype (be it software, hardware, or a mix) be flexible for modeling and support analysis, which is usually some kind of simulation. Tools as well as general languages are used to support the prototyping effort [20]. Because Rapid System Prototyping ultimately leads to a product, work is growing to relate RSP to the rest of the design flow and traditional CAD tools [21]. Because many prototypes are in software for hardware systems, they are frequently thrown away after actual design begins. The most apparent disadvantage of throwaway prototypes is spending implementation effort on code that will not contribute directly to the final product.

5. Example system

Determining which individuals are authorized access to places, information, or transactions requires some level of identification. There are three components to identification: what one has, such as a credit card or car key; what one knows, such as a password or PIN number; and what one is, such as eye color or fingerprints. The latter are implemented with physical measurements called biometrics. Early in the product development process the security decisions must be made. The following example shows the data driven decisions in the first stage, followed by the feasibility experiments in the second stage and a final product design implemented in the last stage.

A secure access device allows or denies access based upon measurements and entered data. There are trade-offs between how accurate an identification measurement is and how time consuming or expensive it is. Also, when dealing with human beings there are psychological effects such as invasiveness, personal privacy, and discomfort. There are also formal metrics for the accuracy and effectiveness of a system. The simplest measure is percent of correct data. By that we mean that access is correctly granted or denied based upon the data. There are two types of correct data. The first is the data indicates access should be granted, and indeed the access should be granted. The other correct response of the system is to indicate

access denial and in fact the individual should be denied access. The incorrect decisions are also of two types. The False Match Rate(FMR) is percentage of results when access is granted for an individual that should be denied. The False Non-Match Rate (FNMR) is the percentage of individuals or trials when access is denied and it should have been granted. For a very secure facility one wants a very low FMR even at the cost of a high FNMR. That is keep out those bad guys even at a high costs. For a relatively open environments, one can sacrifice a high FMR, that is allowing unauthorized access, in order to achieve a low FNMR, that is to avoid turning away someone who deserves access.

The example project is to enhance the security of a bank ATM by adding biometrics. The basic security of an account number (what one has, i.e. the ATM card) and the PIN (what one knows) is augmented by some biometrics (what one is). The first prototype evaluated several biometrics including height, weight, and relative finger length. The first prototype also needed to evaluate different matching criteria for the measurements. The criteria included absolute difference of the measured data to the records (that is the database data), the ratio of the measured data to the records, and the root-mean-square between the measured data and the database values. This prototype was implemented as a FORTRAN program. The surprising result was that the simple absolute difference gave the best match rate results. The signatures for the test cases were analyzed and were recorded for subsequent stages.

The second prototype was implemented as an embedded microprocessor system. This was simulated using the TEst eXecute And Simulate system for an MC68HC11A8 [23] The goal of the second prototype was to demonstrated feasibility, evaluated database representations, and to identify circuit bottlenecks. A side benefit of the signature analysis was that when the test cases were run, errors in database entry were identified, and diagnosed for correction. The final implementation was in Verilog.

6. Conclusion

Product development is must address the need for increased quality. This takes time, and this time is being reduced as technology and customers drive for shorter and shorted product development times. This paper proposes a product development methodology that increases product quality and shortens development time. These conflicting constraints are satisfied by a methodology merging BIST and RSP. The methodology was applied to a secure access system. The example system demonstrated the success of the methodology for rapid development, design verification, and even database entry error checking.

References

[1] Dr. Stanley J. Krolickoski, Dr. Wolfgang Nebel and Dr. Laila Kabout, “Low-Power Analysis using ORINOCO®,” Electronic Design Processes Workshop, April 2004.

[2] Dwight Hill, “Using Tcl/CCI to Turn EDA Cousins into Sisters,” Electronic Design Processes Workshop, April 2004.

[3] Jose Augusto Domingues Fernandes Lima, “The UEDK: A VLSI CAD/EDA Learning-by-example Platform,” Electronic Design Processes Workshop, April 2001.

[4] Aparna Dey and Jim Moudy, “Cost Savings via Reuse,” Electronic Design Processes Workshop, April 2002.

[5] Juan-Antonio Carballo and Raminderpal Singh, “Impact of Signal Integrity on system-On-Chip Design Methodologies,” Electronic Design Processes Workshop, April 2004.

[6] E. J. McCluskey and F. Buelow, “ IC Quality and test transparency”, International test conference, Proc. IEEE Sept 1988, pp. 295- 301

[7] R. A. Frohwerk, "Signature Analysis: A New Digital Field Service Method," Hewlett-Packard Journal, pp 2-8, May 1977.

[8] J. E. Smith, "Measures of the Effectiveness of Fault Signature Analysis," IEEE Trans. Computer, vol. C-29, no. 6, pp. 510-514, June 1980.

[9] D. K. Bhavsar and B. Krishnamurthy, "Can We Eliminate Fault Escape in Self Testing by Polynomial Division," 1984 ITC, pp. 134-139

[10] D. K. Pradhan and S. Gupta, "A New Framework for Designing and Analyzing BIST Techniques and Zero Aliasing Compression," IEEE Transactions on Computers, pp. 743-763.

[11] Dhanendra Jani, Ben Cohen, and John M. Acken, “CAD Tools for BIST Insertion and Analysis,” IEEE Custom Integrated Circuits Conference, May 1992.

[12] T. W. Williams, A. Daehn, M. Gruetzner, and C. W. Starke, "Aliasing Errors in Signature Analysis Registers,"IEEE Design and Test of Computers, pp. 39-45, April 1987.

[13] S. K. Gupta and D. K. Pradhan, "A New Framework for Designing and Analyzing BIST Techniques: Computation of Exact Aliasing

Probability," Proc. Of the 1988 Intl. Test Conf., pp. 329-342.
[14] Dhanendra Jani and John M. Acken, “Test Synthesis for Microprocessors,” ITC Test Synthesis Seminar, October 1994.
[15] Paul H. Bardell, William H. McAnney, Jacob Savir, “Built-In Test for VLSI: Pseudorandom techniques,” John Wiley & Sons, New York, 1987.
[16] E. J. McCluskey, "Built-In Self Test Techniques," IEEE Design and Test of Computers, pp. 21-28, April 1985.
[17] M. Abramovici, M. A. Breuer, and A. D. Friedman, “Digital System Testing and Testable design,” Computer Science Press.
[18] Ryan J. Fong, Scott J. Harper, Peter M. Athanas, “A Versatile Framework for FPGA Field Updates: An Application of Partial Self-Reconfiguration,” 14th IEEE International Workshop on Rapid System Prototyping, pp117-124, June 2003
[19] Luqi, M. Shing, J. Puett, V. Berzins, Z. Guan, Y. Qiao, L. Zhang, N. Chaki, X. Liang, W. Ray, M. Brown, D. Flood “Comparative Rapid Prototyping, A Case Study,” 14th IEEE International Workshop on Rapid System Prototyping, June 2003.
[20] J. Bret Michael, Man-Tak Shing, Michael H. Miklaski, Joel D. Babbitt, “Modeling and Simulation of System-of-Systems Timing Constraints with UML-RT and OMNeT++,” 15th IEEE International Workshop on Rapid System Prototyping, pp. 202-209, June 2004.
[21] Apostolos Dollas, Kyprianos Papademetriou, Euripides Sotiriades, Dimitrios Theodoropoulos, Iosif Koidis, George, “ A Case Study on Rapid Prototyping of Hardware Systems: The Effect of CAD Tool Capabilities, Design Flows, and Design Styles,” 15th IEEE International Workshop on Rapid System Prototyping, pp. 180-186, June 2004
[22] Nahreen Maherukh, “A New Design Methodology for High Quality Design,” Master’s Thesis, Oklahoma State University, June 2004.
[23] Valvano, Jonathan W., “Embedded Microcomputer Systems Real Time Interfacing,” Brooks/Cole, 2000.