Metric driven Framework for Processor Verification

Asheesh Shah, A. Mazyad, and A.K.Ramani, *Member, IEEE.*

Abstract— **Processor verification is a time consuming task and with processor complexity increasing by the day, managing the complete verification process successfully has become a major challenge. Besides, a small bug in the final product may ruin all the efforts and could prove as a critical setback. This problem has resulted in verification methodologies, like formal verification, gaining considerable importance over the years. Yet, integration of formal verification with existing methodologies like simulation and other verification modules is still not very clearly established and remains vendor specific. Then there are other issues that make the whole process very complex. This paper looks into the various aspects of verification methodologies presenting key ideas. We present a framework that can enhance verification process along with metrics will serve to increase overall efficiency.**

Index Terms **– verification, processor, metric, framework**

I. Introduction

The sophistication of recent processor architectures requires major logic verification effort both in terms of time and manpower. This has become a major bottleneck in the overall time to market the final product. Verifying the processor requires thorough test plans, efficient simulation technology and a proper execution plan. Further, verification challenges are created due to cache coherency, memory management and other subtle architecture design and features which can be vendor specific. Beside verification of the design, it is also necessary to test the performance of the newly designed chip [5]. All these tasks require large man hours and millions of investment.

To keep with the pace of processor complexity and performance targets, it is necessary to have an efficient verification strategy and execution plan [6]. Already a lot of work has been carried out in the domain of verification; still there are enough challenges to meet. This paper starts with a brief summary of the various verification technologies and methodologies in vogue. Based on industrial verification strategies [1], [2], [3] and our own experience with Leon [32], we describe the contours of our framework. We believe that a good strategy and a framework [35], [36], [37] holds the key to further the cause of the verification process, and can help to increase efficiency. The paper ends with conclusion and future work to be carried.

II. Present status

Verification technologies offer two broad methodologies for processor verification. Although they are well known, we will briefly summarize them as a precursor to our work. To a large extent simulation serves as the main driving engine of all verification flows. Simulation-based verification essentially consists of Register Transfer Level (RTL) code describing the design along with a test bench which emulates the real environment the design sits in.

Besides simulation, formal and semi formal methods are quite popular and act as a strong complementary methodology to the simulation linked verification process. No doubt they have been gaining grounds over the years [27]. Formal methods are effective to identify unknown bugs but are time consuming and slow. The use of formal tools for verification has been well researched and documented [1], [2], [3], [7], [8], [9], [12], [16], [21]. However they are constrained by the complexity of the design and hence could not be applied across the core design. The practical limitation of Functional Formal Verification (FFV) stems from the fact that formal algorithms tend to require exponential resources with respect to design size [10]. Therefore formal methods will always be restricted to specialized design components [11]. A good description on the strategies to deploy formal tools can be found in [7], [8]. Formal verification with model checking technology increases the controllability of the design. Once a design is instrumented with assertions, formal verification can verify areas of concern, known as hot spots. Model checking analyzes the RTL structure of a design and characterizes its internal nature, and it targets corner-case behaviors directly. Each assertion violation discovered by model checking is reported along with its counter-example. This uncovers functional errors that would have been missed using traditional verification methodologies. FFV tools are an important part of overall process. Both formal and semi

Manuscript submitted on December 31, 2009 . This work is supported in part by the CCES Kharj, King Saud University, Riyadh Saud Arabia.

Asheesh Shah is with College of Computer Engineering and Sciences, at King Saud University Kharj , Saudi Arabia. Asheesh Shah was earlier at IIT Delhi, India.(email: asheesh.shah@gmail.com).

Abulaziz Mazyad is the Dean of CCES, KSU Kharj Saudi Arabia. He can be contacted at Mazyad@ksu.edu.sa

Ashwani Ramani is professor and head of Department of Computer Science, Devi Ahilya Vishwavidhyalaya Indore, India.

formal tools are important parts of any commercial driven verification process [10].

A. *Early work*

Daniel Lewin et al in 'Methodology for Processor Implementation Verification' [22] describe the verification process in four distinct steps. Since then lot of research and work has been carried out targeting processor verification [4], [23]. These works cover a broad range of aspects covering processor verification methodology [24] and framework [25].

B. *Challenges in Processor Verification*

Although Processor verification has been going on for years, but with increasing of processor complexities, the challenges remain for a fast and efficient methodology. Most of the challenges are well known and have been well documented [36]. Besides simulation, verification also depends on certain confidence measures that can help in deciding the success of the process and project management techniques. These confidence measures are in the form of test and metrics. Metric driven verification thus has gained significant importance over the years. [38] However there remains need and scope to integrate metrics within a framework for superior results.

C. *Tools and techniques*

A successful processor verification strategy cannot be complete without taking into consideration test generation methodologies, data collection for coverage and analysis, and typical management issues.. Picking the right tool is one of the main tasks of the whole verification procedure. The challenges here are manifold. One of them is to integrate formal verification tool to simulation and others verification modules [18], [19]. The choice can be best addressed by certain criterion like time required in setting up the environment to its final results, expertise, risk- reward ratio, size, coverage etc. Coverage data collection and analysis is another area of concern which is well acknowledged and needs to be addressed with right earnestness. There are also a host of test generation methodologies [17] ranging from direct, random and constraint driven technologies [20]. A proper log of events and bugs reported will be helpful in the overall and future verification strategy and needs to be shared among the various teams. A detailed bug report, review and analysis report [1] will be useful in reducing the time and man power effort besides in improving the future design work. There are many other vital tools that will be required but are beyond the scope of this paper. Some specialized methods are reported for functional verification in [3], [15], and would require relevant tool support. Co-simulation is also widely used to reduce time in many applications [15]. Most of the tools that are used are in house especially for large and commercial companies like IBM, Intel and some others [1], [2], [3], [13], [23]. However a host of tools are also available from vendors [28], [29], [30], [31]. One can also gain from the work of universities and research [26] and can use them with due diligence.

III. METRICS

Metric driven verification is an important part and strategic approach towards a successful verification process. It has been adopted successfully by industry and commercialized by EDA vendors [30]. Due to the high risk in any potential bug detection, coverage analysis and useful metrics will always be in demand. In our framework we have ensured that proper metrics are incorporated to increase confidence level of various verification teams. The framework creates software functions and other important parameters which can be used to measure the depth of verification.

IV. FRAMEWORK

One of the main challenges that lie before the processor verification team is to come up with a pre-RTL, fast functional verification. Attempts are being made to raise the level of abstraction at the system level [7] along with growing interest in ESL [33]. Another area that can reduce the verification time considerably is by looking the compliance problem [25]. The notion of framework for a successful verification plan and strategy stems from these pre-RTL attempts which can help in considerable time savings. However there is ample space to expand the scope of framework to cover pre RTL and post RTL verification process [3]. VMM based verification approach offers a systematic and modular approach to increase the efficiency of processor verification in the post RTL phase. Our framework is based on this approach and utilizes the VMM library extensively. As a first step to build the framework, it is essential to follow the guidelines prescribed by VMM methodology for compliance. To achieve the desired results we have automated and built a tool which can be used to support VMM environment. This calls for creation of monitors, protocols, interface, Assertion library, coverage gathering mechanisms, scripts, and a clean directory to start the operation. It also requires considerable effort in the integration of all the components and to make sure that the VMM test benches work effectively. Besides, a number of other components are also developed for increasing the efficiency and making the framework user friendly.

The main engine of our framework is the generation of metric which is used to understand the depth of the verification process. This metric looks into ever line of code for code coverage and passes the information directly to test bench generation module. The VMM acts as a base to the framework which also includes strategy [35] and planning modules which helps to increase the efficiency of overall verification process.

V. Results

A framework for verifying processor architecture was created based on Verification Methodology Manual (VMM). In order to check the framework we have used small processor components to verify the design logic. The framework accepts these components as Design under Test (DUT) and constructs the required environment for VMM based approach. Our framework derives its strength from the modularity and uniformity adopted by VMM which helps significantly to reduce the verification time and effort. Thus it can be used at an early stage of processor verification to detect and find bugs which can increase the overall efficiency of the verification effort. The integration of metric and strategy with the basic verification process helps to utilize time efficiently. We have extensively verified processor components available under GNU [32]. As a case study we worked upon the ALU unit of general purpose processor architecture. The time saved in verifying the ALU was significantly lower with the framework that we created. The framework is still in the development stage therefore full results are still not reported.

VI. CONCLUSION

This paper is based upon our project to build a strong verification plan, framework and strategy encompassing and integrating various tools, techniques and methodologies in place. As the abstraction level is increased further there will be a greater need for compliance checking. A majority of verification bugs arises from wrong specifications, communication problems and ambiguity. While coverage data and analysis in the post RTL scenario is a continuous area of development. There are a slew of tools and techniques both in house and external to large industry houses engaged in processor verification. Therefore to reduce the time to market the focus has to be on the integration of these tools and earlier reporting of failures. Our work for a framework is directed towards such considerations which can be helpful for reducing the overall verification timeframe.

The scale and scope of the work prevents us from further elaboration on the subject in a limited space. Nevertheless we have presented key ideas to enhance verification process. The present status in the verification area reinforces the need of an efficient framework which can exploit some existing tools and techniques by superior integration and the creation of an environment and integration.

VII. FUTURE WORK

Some of the ideas related to processor verification has been discussed at length. There is no dearth of tools and techniques in this field. However the challenges in the face of growing processor complexity will remain. The next step from here is to further enhance the framework based upon the ideas that we have discussed. We are in the process of enlarging the scope of our framework to include large and complex processors. Therefore this may be considered as the first part of the whole project. Future work comprises of making the process fully automated and integrating other verification techniques into the framework.

Acknowledgment

The authors would like to place on record the support of King Saud University and the Dean College of Computer Engineering and Sciences for granting permission to carry out the project and provide the necessary facilities.

References

[1] David Burns, “Pre-Silicon Validation of Hyper-Threading Technology,” Intel Technology Journal Q1, 2002. Vol. 6 Issue 1.

[2] J.M.Ludden, et al, “Functional Verification of the POWER4 Microprocessor And POWER4 Microprocessor And POWER4 Microprocessor Systems.” IBM J.RES. & DEV. VOL 46 NO 1 January 2002

[3] D.W.Victor, “Functional Verification of the POWER5 Microprocessor and POWER5 Multiprocessor Systems,” IBM J. RES. & DEV. VOL 49 NO. 4/5 July/ Setember 2005.

[4] Yves Mathys, Andre Chatelain, “Verification Strategy for Integration 3G Baseband SoC,” DAC 2003, June 2003, Anaheim California, USA.

[5] Ronak Singhal , et al, “Performance Analysis and Validation of the Intel® Pentium® 4 Processor on 90nm Technology.” Intel Technology Journal Volume 8, Issue 1, 2004.

[6] Mohamed Salem, Harry Foter, “Planning for Functional Closure,” International Design and Test Workshop 2007.

[7] Robert Beers, “ Pre-RTL Formal Verification: An Intel Experience, “DAC 2008, June 8- 13 2008, Anaheim, California, USA. 2008, Anaheim California, USA.

[8] Raj S. Mitra, “Strategies for Mainstream Usage of Formal Verification,” DAC 2008, June 8- 13, 2008, Anaheim California, USA.

[9] Eyal Segev, Sharon Goldshlager, Hillel Miller, Oren Shua, Olga Sher, Shlomo Greenberg, “Evaluating and Computing Simulation Verification vs Formal Verification Approach on Block Level Design,” ICECS 2004.

[10] R.M.Gott, J.R.Baumgartner, P.Roessler, S.I.Joe, “Functional Formal Verification on Designs of Pseries Microprocessors and Communication Subsystems. IBM J. RES. & DEV. VOL. 49 NO. 4/5 July/ September 2005.

[11] Rajesh Gupta, Shishpal Rawat, Brian Bailey, “PANEL -Formal Verification – Prove it for Pitch It,” DAC 2003, June 2- 6, 2003, Anaheim, California, USA.

[12] Gary York, Robert Mueller-Thuns, Jagat Patel, Derek Beatty, “An Integrated Environment for HDL Verification,” IEEE 1995, pp. 9-18

[13] B.Wile, et al, “Functional Verification of the CMOS S/390 Parallel Enterprise Server G4 System,” IBM J. RES. & DEV. VOL. 41 NO. 4/5 July/September 1997.

[14] D.G.Bair, “Functinal Verification of the z990 Superscalar Multibook Microprocessor Complex.” IBM J. RES. & DEV. VOL. 48, NO.3/4 July 2004.

[15] S.P.Goldman, L.M.Mohr, D.R.Smith, “Usng Microcode in the Functional Verification of an I/O Chip,” IBM J. RES & DEV. VOL. 49, NO. 4/5 July/September 2005

[16] 16. W.A.Hunt Jr., J.Sawada, “Verifying the FM9801 Microarchitecture,” IEEE Micro, May-June 1999.

[17] Aharon Aharon, et al, “Test Pogram Generation for Functional Verification of PowerPC Processors in IBM,” IBM J. RES. & DEV.

[18] Asgeir Th Eiriksson, “ Integrating Formal Verification Methods with A Coventional Project Design Flow,” DAC 1996.

[19] Scott Hazelhurst, Osnat Weissberg, Gila Kamhi, Limor Fix, “A Hybrid Approach Getting Deep into the Design,” DAC 2002, June 10- 14 2002, New Orleans USA.

[20] Mike G Bartley, Darren Galpin, Tim Blackmore, "A Comaprison Of Three Verification Techniques: Dircted Testing, Pseudo-Random Testing and Property Checking," DAC 2002, June 10- 14 2002, New Orleans, USA.

[21] Daniel Kroening, Sanjit A. Sinha," Formal Verification at Higher Levels of Abstraction," DAC 2007.

[22] D. Levin "A Methodology for Processor Implementation Verification", First international conference on formal methods in computer-aided design, 1996.

[23] S. Taylor, M. Quinn, D. Brown, N. Dohm, S. Hildebrandt, J. Huggins, C. Ramey,"Functional Verification of a Multiple-issue, Out-of-Order, Superscalar Alpha Processor – The DEC Alpha 21264 Microprocessor," *35th ACM/IEEE Design Automation Conference*, pages 638–643, June 1998.

[24] D. Geist, G. Biran, T. Arons, M. Slavkin, Y. Nustov, M. Farkas, K. Holtz,"A Methodology For the Verification of a System on Chip,". *36th ACM/IEEE Design Automation Conference*, pages 574–579, June 1999.

[25] Allon Adir, Sigal Asaf, Laurent Fournier, Itai Jaeger, Ofer Peled, "A Framework for the Validation of Processor Architecture Compliance", 44th Design automation conference.

[26] K.L.McMillan, "Symbolic Model Checking: An Approach to the State Explosion Problem," Ph.D. thesis, technical rep. CMU-CS-92-131, School of computer science, Carneige Mellon University. PA, USA. May 1992.

[27] T.Schlipf, T.Buechner, R.Fritz, M.Helms, J.Koehl, "Formal Verification made Easy," IBM J. RES. & DEV. VOL. 41 NO. 4/5 July/September 1997.

[28] http://www.denali.com/purespec; Denali is a EDA vendor

[29] http://www.mentor.coml Mentor Graphics is a leading EDA vendor

[30] http://www.cadence.com; Cadence is a leading EDA vendor

[31] http://www.synopsy.com; Synopsys is a leading EDA vendor

[32] http://www.gaisler.com; is the web site for GNU project 'Leon'.

[33] Hiroyuki yagi, et al " ESL Hand-off: Fact or EDA Fiction?, " DAC 2008, June 8- 13 2008, Anaheim, California USA.

[34] Introduction to Design Verification with VMM: A Quickstart Guide

.[35] Shah Asheesh, Mazyad Abdulaziz, Ramani Ashwani, " A Strategy and Framework for Processor verification", VDAT 2009 Bangalore India.

[36] Shah Asheesh, Mazyad Abdulaziz, Ramani Ashwani, "Formalizing Certain Tasks for Processor Architecture Verification", AFRICON 2009.

[37] Shah Asheesh, Mazyad Abdulaziz, Ramani Ashwani, "A Verifiable Strategy for Processor Verification", DSD2009, Euromicro, Patras, Greece.

[38] Carter B.Hamilton, Hemmady Shankar, "Metric Driven Design Verification", Springer

ISBN: 978-988-18210-4-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)