SYSTEM VERILOG RTL MODELING WITH EMBEDDED ASSERTIONS

CHOW CHEE SIANG

UNIVERSITI TEKNOLOGI MALAYSIA

SYSTEM VERILOG RTL MODELING WITH EMBEDDED ASSERTIONS

CHOW CHEE SIANG

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical - Computer & Microelectronic System)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

MAY 2011

Specially dedicated to my beloved parents, wife, supervisor, lectures, fellow friends and those who have guided and inspired me throughout my journey of education.

ACKNOWLEDGEMENT

This project will not able to complete without the help and guidance from others. Thus, I would like to take this opportunity to acknowledge the people below.

First and most importantly, I would like to express my gratitude to my project supervisor Dr. Muhammad Nasir. He guided me along the journey and always gives constructive suggestions and opinions. With his help, I do not lose in term of the project scope and objectives. I would say, throughout the whole year of working under him, I really gain a lot of knowledge not only in the technical area but also prepare myself to be a better person.

On top of that, I would like to thank my manager and Intel Penang Design Centre for supporting and sponsoring me to sign up this part time course. Thanks to my manager for being so thoughtful and always allows me to take examination leave to complete my project.

On a personal level, my deepest thank go to my parents, wife, friends and peers for their mental support throughout my academic years.

ABSTRACT

This project has a final goal of developing a new methodology of pre-silicon and post-silicon validation which helps in better IP delivery to SOC system. Hardware Description Language, System Verilog is adopted in doing RTL modeling and System Verilog Assertions are used in verifications. Both design and validation components are combined into single module with pre-defined compiler directive. The conversion of non-synthesizable System Verilog assertions into synthesizable format enables designers to integrate some built in checkers into own design for pre-silicon and post silicon validation. By having synthesizable assertions in the design, the validation cycle can be shorten because some of the testing can be carried out using FPGA. The testing on FPGA can run much faster than simulation which has dependency on simulator tool. Every System Verilog assertion is being modeled as a real hardware component and embedded into design block. The project provides the methodology and examples of how to synthesize System Verilog Assertions using component cascading method to represent temporal expressions used in non-synthesizable assertions.

ABSTRAK

Projek ini bertujuan untuk membangun satu metodologi baru untuk pra silicon era dan lewat silikon era yang mana membantu dalam penyampaian IP lebih baik kepada sistem SOC. System Verilog, bahasa baru untuk menyampai RTL peragakan dan System Verilog Assertions digunakan dalam penentusahan. Kedua-dua komponen reka bentuk dan pengesahan disatukan menjadi modul tunggal dengan arahan penyusun ditakrifkan. Transformasi dari System Verilog Assertions yang tidak boleh di-sintesis ke bentuk yang boleh di-sintesis membolehkan pereka-pereka menyepadukan beberapa jenis bentuk kepada reka bentuk sendiri. Kitaran pengesahan boleh dipendekkan kerana ujian-ujian RTL boleh dilaksanakan dengan menggunakan FPGA. Ujian pada FPGA dapat berjalan lebih cepat lagi daripada simulasi yang banyak bergantung kepada alat pensimulasi. Setiap System Verilog Assertion boleh dibentukkan dalam sintesi format dan ditanam dalam blok reka bentuk. Projek ini menyediakan kaedah dan contoh-contoh bagaimana untuk mensintesiskan System Verilog Assertions menggunakan komponen melata kaedah mewakili ungkapan-ungkapan menggunakan dalam non-sintesis ungkapan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SVA	–	System Verilog Assertions
SV	–	System Verilog
RTL	–	Register Transfer Level
IP	–	Intellectual Property
SOC	–	System On Chip
HDL	–	Hardware Description Language

LIST OF SYMBOLS

D	– Number of clock cycle delay
!	– Boolean NOT function
D1	– First delay in clock cycles
D2	– Second delay in clock cycles

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

This project report proposes the use of System Verilog in Register Transfer Level (RTL) modeling with embedded verifications components which are also in System Verilog format. The overall idea is to accelerate the design and verification of Intellectual Property Core (IP core) in a standard, easy and consistent way. In this chapter, the near end challenges and future of RTL logic system are discussed. This chapter also covers the background, research objectives and motivations, research work scope, brief of methodology and how the subsequent chapters are being organized.

1.7 Problem Statement and Motivations

An HDL does a good job of spanning design concepts (called the register transfer level, RTL) down to a few primitives that are used in great numbers to implement a design (called the gate level). However, years over years, the design complexity and size has grown exponentially. The System Verilog extensions to the Verilog HDL address two major engineering needs, i.e. to efficiently model designs and verify these complex designs.

Some motivations to why there are getting more designers use System Verilog in the RTL design. There is comparable amount of code reduction, thus it eliminates most of the coding errors and improve the readability of the code. System

Verilog has the capability to represent complex functionality in concise, easier to read, easier to reuse RTL code. System Verilog also provides the capability to add white-box assertions to synthesizable code, without the convoluted synthesis "translate_off" and "translate_on" pragmas. System Verilog is a single language used throughout the design flow, including synthesizable RTL models, test programs and bus functional models. These advantages can lead to improved time to market through clearer specifications, and earlier detection of design bugs. One of the biggest design issues today is the occurrence of tape-out delays or re-spins caused by finding bugs late in the design cycle. Under System on Chip (SOC) concept, the time to develop particular RTL design or an IP, including design and verifications, is getting shorter to align the development duration to shorter "time to market" period. So, in the very tight schedule to deliver an IP product, IP development team needs to complete both design and verifications with good quality. The proposed methodology, switching the HDL to System Verilog and embeds the assertions into the synthesizable RTL code will greatly help the IP development team to achieve the important milestone and allow better IP integration process in SOC (System on Chip) design. In short, System Verilog includes features that address current design challenge through improved specification of design, conciseness of expression and unifications of design and verifications.

Generally, all of assertions are non-synthesizable and ignored by synthesis tool. By migrating assertions to be synthesizable RTL code, it greatly helps the post silicon debug, as the embedded checks are served as built in RTL checkers provide valuable insights to the cause of failure. Moreover, the synthesizable RTL assertions tend to have better simulation performance against non-synthesizable in pre-silicon validation environment, as it eliminates the threads which happen in non-synthesizable assertions which simulator needs to keep track of. One more advantage to gain is, designers are the candidates who understand own design the most, should be involved in writing checks and assertions to ensure the design always does the right thing, so the verification engineers can focus on higher level (functional or system level) checking instead of focusing on gate or small logic block. Using System Verilog as verification language brings us a lot of advantages, such as, System Verilog as IEEE standard since 2005 ensures a wide embracing and support

by multiple vendors of EDA tools and verification IPs, as well as interoperability between different tools and vendors. The risks and costs of adopting a new verification language are reduced because System Verilog is an extension of the popular Verilog language, the adoption process by engineers is easier and straightforward. System Verilog enables engineers to adopt a modular approach for integrating new modules into any existing code. Being an integrated part of the simulation engine, System Verilog eliminates the need for external verification tools and interfaces, and thus ensures optimal performance (running at least x2 faster than with any other verification languages).

System Verilog indeed offers many interesting new constructs, both for system engineers, RTL designers and verifications purposes. The excitements grow when there are more and more emerging tools supporting System Verilog. The perspective of using System Verilog for RTL design is to move to higher level of abstraction. A gain in design productivity could be hoped for, but an issue such as design for verifications should certainly also be addressed when considering new RTL modeling styles. In addition, focus has been put on the elements in the language believed to improve the robustness, readability and reusability of the RTL while escaping a few known pitfalls in Verilog.

1.8 Research Objectives

There are few objectives need to be achieved in this research,

1. To explore RTL modeling using System Verilog
2. To provide macro for standard and synthesizable System Verilog RTL code generation
3. To explore RTL modeling with System Verilog assertions embedded
4. To provide macro for standard and synthesizable System Verilog assertions to be added into the design

1.9 Research Scope

This research is divided into two parts and based on available hardware, software resources, limited time frame and expertise, this research project is narrowed down to the following scope of work:

1. The RTL design will be using System Verilog.
2. The algorithm to translate System Verilog assertions from non-synthesizable to synthesizable RTL code will be outlined and developed.
3. The final RTL design block will consists of both logic design and assertions for verifications.
4. The synthesizable System Verilog assertions are developed in component manner, and how design can make use of it by cascading several components.
5. High level macro is developed to assist in RTL code generation.
6. This project is limited to design, synthesis, simulate and verify the design correctness with simulation software (Altera Quartus II)
7. Several test cases will be selected to demonstrate the flow of using high level macro and how to embed assertions in the design.

1.10 Research Contributions

1. A standard algorithm to convert non-synthesizable System Verilog assertions to synthesizable RTL code. Any complex assertions can be modeled easily based on the algorithm.
2. An approach of using synthesizable assertions in RTL code, which is well suited in SOC design.
3. Synthesizable System Verilog assertions in components
4. Detail examples on how SVA components help to model complex temporal expressions

1.11 Research Methodology

This research work starts off with literature review regarding System Verilog in RTL modeling and assertions. A lot of readings and analysis have been done to understand the new language better. This research involves mostly the effort to figure out the proper and standard algorithm to convert non-synthesizable assertions to synthesizable RTL code, how to use System Verilog in RTL modeling efficiently and how the cascading synthesizable System Verilog assertions components help to model complicated assertion systematically. The remaining effort will be spent to develop the macro to assist designers and verification engineers in System Verilog coding, also including the effort spent in doing test case design and verifications using new proposed flow. In order to make the research successful, planning of the work, procedures and methodology is essential. The methodology will be discussed in detail in the coming chapter to demonstrate how the objectives being achieved.

1.12 Report Organization

Chapter I is the introduction to this new idea. The chapter gives an overall picture and brief summary of what will be discussed in this research. Chapter II discusses understanding of materials and information regarding System Verilog RTL modeling and assertions to enable the new methodology to run. Chapter III talks about the methodology and approach used to achieve the objectives, how the synthesizable assertions algorithm is developed includes detail descriptions. Chapter IV consists of software development work carried out in this project. Chapter V discusses some design examples using System Verilog and synthesizable assertions. Chapter VI summarizes the work has been done in this project and suggests some future works and enhancements.

REFERENCES

1. Peter Jensen, Wolfgang Ecker, Thomas Kruse, *System Verilog Interface Based Design*, Infineon Techonologies.

2. Stuart Sutherland, *A Proposal for a Standard Synthesizable Subset for System Verilog 2005, What the IEEE Failed to Define*, Sutherland HDL, Inc, Portland Oregon, 2006

3. Don Mills, *System Verilog Assertions are for Design Engineers too*, Sutherland HDL, Inc, Portlahnd Oregon, 2006

4. Stuart Sutherland, *Modeling with System Verilog in a Synopsys Synthesis Design Flow*, Sutherland HDL, Inc, Portland Oregon, 2006

5. Micheal Pellauer, Mieszko Lis, Donald Blatus, *Synthesis of Synchronous Assertions with Guarded Atomic Actions*, Massachisetts Institute of Technology, BlueSpec Inc, 2005

6. ALon Gluska, Lior Libis, *Shortening the Verification Cycle with Synthesizable Abstract Models*, Intel MMG, MATAM, Haifa, Israel, 2009

7. Mark Litterick, *Using System Verilog Assertions for Functional Coverage*, Verilab, 2005

8. *System Verilog 3.1 language reference manual*, Accellera

9. Steve Haynal, Timothy Kam, Micheal Kishinevsky, Emily Shriver, Xinning Wang, *A System Verilog Rewriting System for RTL Abstraction with Pentium Case Study*, Strategic CAD Labs, Intel Corporation, 2008

10. Shu-Hsuan Chou, Che-Neng Wen. Yan-Ling Luie, Tien-Fu Chen, *VeriC: A Semi-Hardware Description Language to Bridge the Gap Between ESL Design and RTL Models*, Dept. Of CSIE, National Chung Cheng University, Chia-Yi Taiwan, 2009

11. Ravi Surepeddi, *System Verilog for Quality of Results*, Magma Design Automation Inc, 2008

12. Peter Jensen, Wolfgang Ecker, Thomas Kruse, System Verilog: Interface Based Design, Syosil Consulting, Infineon Technology, 2004

13. Clifford E. Cummings, *System Verilog Ports & Data Types for Simple, Efficient and Enhanced HDL Modeling*, Sunburst Design, Inc, 2002