HARDWARE/SOFTWARE SYSTEM-ON-CHIP CO-VERIFICATION PLATFORM BASED ON LOGIC-BASED ENVIRONMENT FOR APPLICATION PROGRAMMING INTERFACING

TEO HONG YAP

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

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2015

Dedicated, in thankful appreciation for support, encouragement and understanding to my beloved mother, father and sister....

ACKNOWLEDGEMENT

First and foremost, thank God for giving me the strength to complete this project. My sincere thanks to my supervisor, Assoc. Prof. Dr. Muhammad Nadzir bin Marsono for his generous guidance, advice and motivation throughout this study. His critical advices and comments help me to develop the study project are highly appreciated. I would like to sincerely thank Universiti Teknologi Malaysia (UTM) for providing the facilities and equipment for this research. I would also like to take this golden opportunity to express my appreciation to my family, especially my parents, for supporting, inspiriting, and encouraging me during the study and completion of this thesis. Lastly, I would like to express my gratitude to those who support me to get through all the difficulties during the completion of the study.

Teo Hong Yap

ABSTRACT

System-on-chip (SoC) is a single-chip that integrates hardware and software components. Hardware/software co-design and co-verification are crucial steps to ensure functional correctness of SoC design. Hardware/software co-verification technique is needed to test and decide ways to partition software and hardware components for an optimized system. Recently, field-programmable gate array (FPGA) prototyping has been proposed as a method that provides a rapid prototyping platform of SoC development and verification. SoC FPGA prototyping involves multiple cross-platform asynchronous clock domains that make SoC verification process becomes more challenging. This project implements an asynchronous first-in-first-out (FIFO) based data transfer between two hardware components which are operating in different clock domains. This implementation operates in actual FPGA and makes use of Logic-based Environment for Application Programming (LEAP) infrastructure such as communication mechanism to allow communication between hardware and software models or components. A study related to execution time characterization is done to understand the effects of hardware/software tasks partitioning on hardware/software communication, hardware execution and software execution time. Resource analysis is done on asynchronous FIFO implementation and it shows a logarithmic relationship between the logic elements and FIFO entries. An approximately linear relationship between two-way average latency and data size are shown by passing data from FPGA to host and return back the data from host to FPGA. MPEG-2 Audio Layer III (MP3) decoder case study shows with an optimum hardware/software partitioning, the co-verification platform is able to achieve a communication time of approximately 30 million cycles with 99.99 percent of the time spent originated from hardware/software communication. This result clearly shows that bidirectional communication between hardware and software plays a significant role in affecting the total communication time spent for particular application which has tasks running in both hardware and software.

ABSTRAK

Sistem-atas-Cip (SoC) adalah litar bersepadu (IC) yang mengintegrasikan semua komponen komputer atau litar elektronik ke dalam satu mikrocip. Pembangunan dan pengesahan SoC telah menjadi langkah-langkah yang penting untuk memastikan ketepatan fungsi reka bentuk SoC. Satu teknik pengesahan diperlukan bagi menguji sistem yang merangkumi tugas perkakasan dan perisian. Teknik ini juga membolehkan partition komponen perisian dan perkakasan dijalankan untuk mendapat satu sistem yang optimum. Baru-baru ini, tatasusunan get boleh atur cara medan (FPGA) telah dicadangkan sebagai satu kaedah bagi menyediakan satu platform untuk pembangunan dan pengesahan SoC. Prototaip FPGA seperti reka bentuk SoC yang melibatkan beberapa sistem jam tak segerak telah menjadikan proses pengesahan SoC semakin mencabar. Projek ini bertujuan untuk mereka bentuk logik masuk dulu keluar dulu (FIFO) tak segerak untuk membolehkan pemindahan data antara dua komponen perkakasan yang beroperasi dengan system jam yang berbeza. Reka bentuk logik ini harus dapat beroperasi dalam FPGA dan menggunakan infrastruktur Logic-based Environment for Application Programming (LEAP) seperti mekanisme komunikasi untuk membolehkan komunikasi antara tugas perkakasan dan perisian. Kajian yang berkaitan dengan masa perlakuan telah dilakukan untuk memahami kesan pembahagian tugas perkakasan/perisian atas masa komunikasi, masa perlakuan perkakasan dan masa perlakuan perisian. Sumber analisis bagi pelaksanaan FIFO tak segerak menunjukkan hubungan logaritma antara unsur logik dan FIFO saiz alamat. Kajian dua hala hantaran data menunjukan hubungan linear antara purata dua cara kependaman komunikasi dan saiz data. Kajian kes penyahkod MPEG-2 Audio Layer III (MP3) menunjukkan pembahagian perkakasan/perisian yang optimum boleh mencapai masa komunikasi 30 juta kitaran di mana 99.99 peratus daripada masa komunikasi itu adalah berasal daripada komunikasi bentuk perkakasan/perisian. Keputusan ini jelas menunjukkan bahawa komunikasi dua hala antara perkakasan dan perisian memainkan peranan penting yang memberi kesan kepada jumlah masa komunikasi yang digunakan untuk aplikasi tertentu yang mempunyai tugas berjalan dalam kedua-dua perkakasan dan perisian.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AWB	–	Asim Architect's Workbench
CPU	–	Central Processing Unit
FIFO	–	First-In-First-Out
FPGA	–	Field-Programmable Gate Array
GPS	–	Global Positioning System
IPC	–	Inter-Process Communication
ISS	–	Instruction Set Simulator
LEAP	–	Logic-based Environment for Application Programming
MP3	–	MPEG-2 Audio Layer III
RTL	–	Register Transfer Level
SoC	–	System-on-Chip

LIST OF SYMBOLS

n_{Comm}	–	Communication ratio
n_{HSComm}	–	Number of hardware-software communication
n_{HHComm}	–	Number of hardware-hardware communication

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Problem Background

System-on-Chip (SoC) is a single integrated circuit (IC) that integrates or packages all components of a computer or electronic circuits into a single microchip. SoC is evolving and causing more system components being integrated on single chip device for each generation. Figure 1.1 shows the basic architecture example of a SoC.

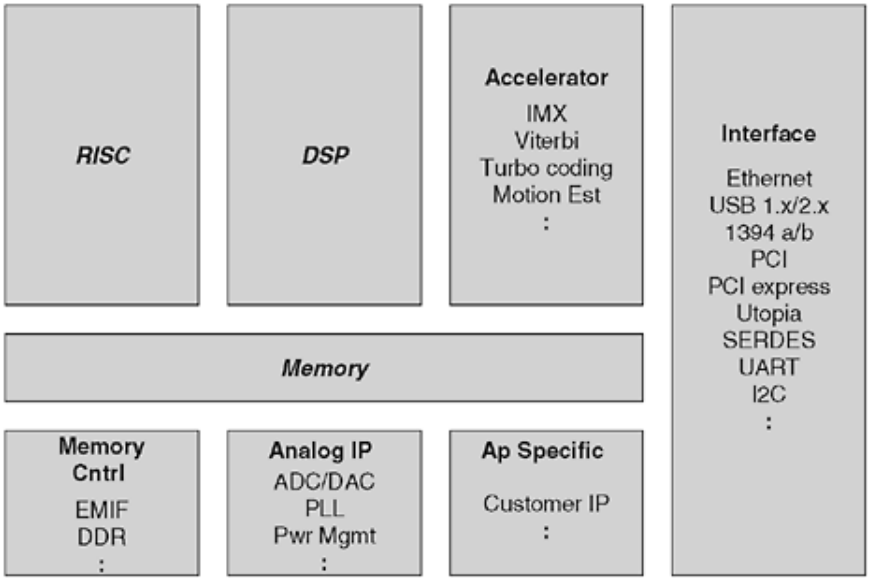

Figure 1.1: SoC basic architecture.

Existence of FPGA prototyping has become a method which provides a platform for SoC development and verification. FPGA prototyping provides a fast and accurate SoC model for early embedded software development in pre-silicon stage.

Virtual platform provides designer with a set of common functionalities by abstracting physical devices into an abstraction layer. Hybrid virtual platform splits module functionality between FPGA and software [8]. Figure 1.2 shows a hybrid virtual platform which consists of abstraction layer made up of hardware and software modules with user application prototyping. Concept of virtual platform and prototyping allows concurrent developed software design to be simulated on a model of virtual platform [1]. System modeling can be done at early stages before the actual product being produced.

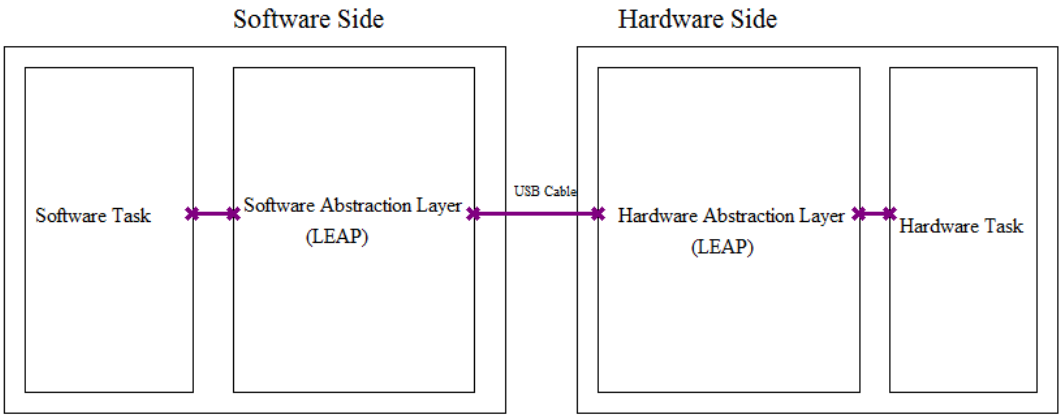

Figure 1.2: Hybrid virtual platform which consists of abstraction layer made up of hardware and software modules with user application prototyping [9].

Hardware/software co-verification technique is needed to test and decide the way to partition software and hardware components in order to produce an optimized system [1, 9, 10]. Co-verification is able to reflect concurrency and synchronization of sub-systems within the main system by observing output signals correctness while concurrently executing hardware and software components with appropriate inputs. This technique co-verifies correct interactions between each of SoC sub-system and is not limited to only verify the correctness of individual sub-system [1].

Virtual and FPGA-based prototyping requires an interface between hardware and software. It has been identified by co-design community that interface synthesis between hardware and software components of system-on-chip (SoC) is becoming one of the important research fields [1,9]. Hardware/software co-design and co-verification methodologies allow faster time-to-market frame as it makes development, integration and verification of sub-system become easier than before.

1.2 Problem Statement

The benefit of hybrid virtual platform to provide a solution which allowed functionality of system-on-chip to be fully modelled through virtual and FPGA prototyping has become widely used in the industry. It is a known issue that high latency of hardware/software communication has caused system modeling becomes more challenging. Inaccurate data sampling by prototyping system due to high latency communication between hardware/software may result in inaccurate system modeling. The latency of hardware/software communication greatly depends on architecture being used while designing the hardware/software interface.

The FPGA prototyping such as system-on-chip design may involve platform crossing that involves employment of multiple asynchronous clock domains has make the design verification process becomes more challenging. It is important to implement a clock synchronization module which allows different functional components in different clock domains to synchronize with each other. Thus, asynchronous FIFO has been identified as an approach to achieve sub-system synchronization.

Application can be partitioned into multiple small tasks which run in either hardware or software. It is important to understand the effects of different hardware/software task partitioning and to identify factors that contribute to total application execution time.

1.3 Objective

The objective of this project is to implement a hybrid virtual platform which enables prototyping of SoC and early software development in pre-silicon stage. The interface between hardware and software task should allow verification of correct interactions between each of the hardware and software components. Thus, LEAP [3, 4] has been identified as interface between hardware and software task through primitive fixed-width bi-directional JTAG communication channel.

In this project, a clock synchronization module is proposed to enable verification to be done for system having sub-system components operating in different clock domains. The implementations of both are verified through actual data passing in real time. The latency and resources usage of the implementations is then analyzed.

A study related to execution time characterization is done to understand the effects of hardware/software tasks partitioning on communication time, hardware execution time and software execution time. This characterization is done through actual data passing through LEAP interface which being implemented for a specific case study.

1.4 Project Scope

In this project, LEAP interface is implemented using Asim Architect's Workbench [2] and Bluespec System Verilog. This project also covers the implementation of clock synchronization module using Bluespec System Verilog. Both implementations are targeted for Altera DE2 platform. The behavior or functionality of the implementation is pre-verified though BlueSim and Altera Modelsim. Both implementations are finally verified through actual data passing in real time. Data analysis is done for each implementation. MPEG-2 Audio Layer III Decoder is used as a case study for execution time characterization which is done for different combination of hardware and software tasks obtained using hardware/software application mapping algorithm. Figure 3.1 shows the project flow been used in this project.

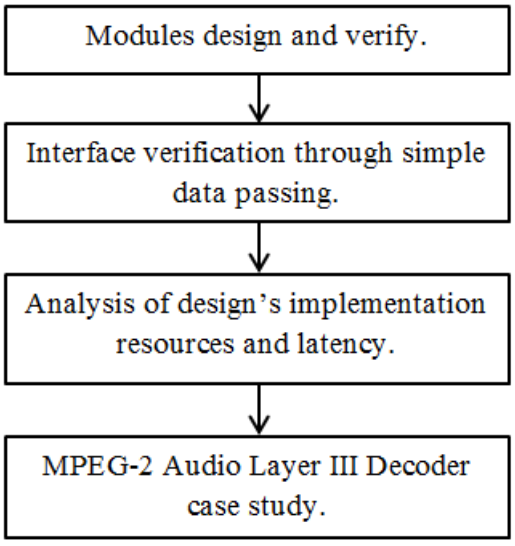

Figure 1.3: Project flow and procedures.

1.5 Report Organization

Figure 1.4 shows the project report organization. This report is written in six chapters. The remaining chapters are organized as follows.

1. Chapter 2 describes previous works related to this project. Background information related to this project is also discussed in Chapter 2.
2. The project flow and methodology is presented in Chapter 3. It includes the flow being taken while conducting this research. The tools being used are also discussed in Chapter 3. General approach and method used while designing the clock synchronization module is also presented in Chapter 3.
3. The actual interface implementation that is being implemented in real hardware design is discussed in Chapter 4. Design verification is also presented in Chapter 4.
4. The interpretation and analysis of the results from implementation are discussed in Chapter 5. Detail analysis of the design is also presented.
5. Chapter 6 concludes the report. Direction of future work is also proposed.

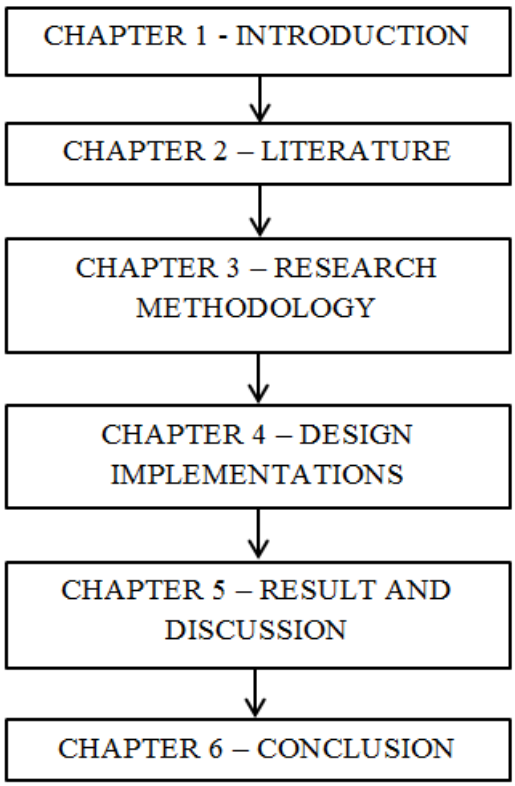

Figure 1.4: Project report organization.

REFERENCES

1. Teich, J. (2012). Hardware/software codesign: The past, the present, and predicting the future. Proceedings of the IEEE, 100(Special Centennial Issue), 1411-1430.
2. Emer, J., Beckmann, C., & Pellauer, M. (2007, June). AWB: The Asim Architect's Workbench. In 3rd Annual Workshop on Modeling, Benchmarking, and Simulation (MoBS 2007).
3. Fleming, K., Yang, H. J., Adler, M., & Emer, J. (2014, September). The LEAP FPGA operating system. In Field Programmable Logic and Applications (FPL), 2014 24th International Conference on (pp. 1-8). IEEE.
4. Parashar, A., Adler, M., Fleming, K., Pellauer, M., & Emer, J. (2010, December). LEAP: A virtual platform architecture for FPGAs. In The First Workshop on the Intersections of Computer Architecture and Reconfigurable Logic (CARL 2010).
5. Oprescu, F. A. (2002). U.S. Patent No. 6,359,479. Washington, DC: U.S. Patent and Trademark Office.
6. Zewei, L., & Mei, X. (2008, May). An effective AS-FIFO design for multiple asynchronous clock data transmission. In Communications, Circuits and Systems, 2008. ICCCAS 2008. International Conference on (pp. 960-963). IEEE.
7. Ono, T., & Greenstreet, M. (2009, May). A modular synchronizing FIFO for NoCs. In Proceedings of the 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip (pp. 224-233). IEEE Computer Society.
8. Synopsys, Inc "Synopsys Hybrid Prototyping Solution." , 2012.
9. Rao R. Tummala: System on Package: Miniaturization of the Entire System. Introduction to System-on-Chip (SOC), Chapter (McGraw-Hill Professional, 2008), AccessEngineering.
10. Ernst, R. (1998). Codesign of embedded systems: Status and

trends. Design & Test of Computers, IEEE, 15(2), 45-54.

11. Schaumont, P. (2012). A practical introduction to hardware/software codesign. Springer Science & Business Media.

12. Reis, R., Lubaszewski, M., & Jess, J. (Eds.). (2006). Design of Systems on a Chip: Design and Test. Springer.

13. Richards, M. A. (1994). The Rapid Prototyping of Application Specific Signal Processor (RASSP) Program: Overview and Accomplishments (No. 942550). SAE Technical Paper.

14. Mark Balch: Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture. Logic Design and Finite State Machines, Chapter (McGraw-Hill Professional, 2003), AccessEngineering

15. Gangadharan, S., & Churiwala, S. (2013). Constraining Designs for Synthesis and Timing Analysis. Springer..

16. Churiwala, S., & Garg, S. (2011). Principles of VLSI RTL Design: A Practical Guide. Springer Science & Business Media.

17. Altera "DE2 Development and Education Board, user manual" Version 1.5, 2012.

18. Parashar, A., Adler, M., Pellauer, M., & Emer, J. (2008, June). Hybrid cpu/fpga performance models. In 3rd Workshop on Architectural Research Prototyping (WARP 2008).

19. Le Beux, S., Bois, G., Nicolescu, G., Bouchebaba, Y., Langevin, M., and Paulin, P. "Combining mapping and partitioning exploration for nocbased embedded systems," Journal of Systems Architecture, vol. 56, no. 7, pp. 223–232, 2010.

20. Bouchhima, A., Gerin, P., & Pétrot, F. (2009, January). Automatic instrumentation of embedded software for high level hardware/software co-simulation. In Design Automation Conference, 2009. ASP-DAC 2009. Asia and South Pacific (pp. 546-551). IEEE.

21. You, M. K., & Song, G. Y. (2009, January). Case study: Co-simulation and co-emulation environments based on SystemC & SystemVerilog. In TENCON 2009-2009 IEEE Region 10 Conference (pp. 1-4). IEEE.

22. Cimatti, A., & Bernardo, M. (2006). Formal Methods for Hardware Verification. 6th International School on Formal Methods for the

Design of Computer, Communication, and Software Systems, SFM, 22-27.

23. Yang, H., Yi, Y., & Ha, S. (2009, July). A timed HW/SW coemulation technique for fast yet accurate system verification. In Systems, Architectures, Modeling, and Simulation, 2009. SAMOS'09. International Symposium on (pp. 74-81). IEEE.

24. Kim, Y. I., Aim, K. Y., Shim, H., Yang, W., Kwon, Y. S., Ki, A., & Kyung, C. M. (2005, April). Automatic generation of software/hardware co-emulation interface for transaction-level communication. In VLSI Design, Automation and Test, 2005.(VLSI-TSA-DAT). 2005 IEEE VLSI-TSA International Symposium on (pp. 196-199). IEEE.