QSYS HPS INTERCONNECT VERIFICATION METHODOLOGY FOR SOC FPGA

By

LOH TAT JEN

A Dissertation submitted for partial fulfilment of the requirement for the degree of Master of Science

July 2013

Acknowledgements

First and foremost, I would like to express utmost gratitude to my supervisor, Assc. Prof. Dr. Mohd Rizal Arshad, for giving me opportunity to work under him. With his valuable inputs and constructive feedbacks helped me throughout the research and also the writing of the thesis.

Besides, I would like to express my appreciation to MyBrain15 scholarship program and my company Altera for the financial assistance to complete the Master program. Furthermore, without Altera and its facilities, the whole research methodology and work would not be feasibly completed for the Master program.

Also, I would to thank Qsys HPS verification team and other colleagues and friends for their direct or indirect help during the research. And I would like to record a special thanks to my Altera manager, Mr. See Theam Thye for the continuous support during the development of the project.

Last but not the least, I would like to thank my family especially my wife, How Whee Yan, for her continuous encouragement and understanding throughout the entire Master program.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Meaning
AHB	Advanced High-performance Bus
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
ASIC	Application Specific Integrated Circuit
AXI	Advanced Extensible Interface
Avalon-MM	Avalon Memory-Mapped
Avalon-ST	Avalon Streaming
BFM	Bus Functional Model
CAN	Controlled Area Network
CTI	Cross Trigger Interface
DDR	Double Data Rate
DMA	Direct Memory Access
EMAC	Ethernet Media Access Controller
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
HPS	Hard Processor System
HSSI	High Speed Serial Interface
I2C	Inter-Integrated Circuit
IP	Intellectual Property
LIN	Local Interconnect Network
NoC	Network-on-Chip
PCI	Peripheral Component Interconnect
PIO	Parallel Input Output
PLI	Programming Language Interface
QoS	Quality of Service
RTL	Register Transfer Level
SoC	System-on-Chip
SPI	Serial Parallel Interface
TCL	Tool Command Language
UART	Universal Asynchronous Receiver Transceiver
USB	Universal Serial Bus
VIP	Verification Intellectual Property
VMM	Verification Methodology Manual
XML	Extensible Markup Language

METODOLOGI PENGESAHAN SALING-SAMBUNG HPS QSYS UNTUK SOC FPGA

Abstrak

FPGA yang mengandungi unit pemprosesan terbenam adalah aliran masa depan bagi aplikasi-aplikasi berprestasi tinggi dan berkuasa rendah. Saling-sambung HPS Qsys, telah direka untuk menyambungkan FPGA dengan sistem pemprosesan terbenam (HPS) melalui satu klik tetikus. Walaupun, model berfungsi bas (BFM) sering digunakan bagi metodologi pengesahan untuk saling-sambung Qsys, HPS melibatkan protocol-protokol antaramuka yang berbeza. Tugas untuk merekabentuk dan mengesahkan BFM akan mengambil masa yang panjang. Oleh itu, metodologi pengesahan yang baru telah dicadangkan untuk saling-sambung HPS Qsys di dalam projek penyelidikan ini. Bagi kaedah pengesahan yang dicadangkan, saling-sambung HPS Qsys akan digabungkan ke dalam bangku ujian pengesahan HPS RTL melalui sejenis rekabentuk suis pin. Selain itu, rekabentuk Qsys juga digabungkan ke dalam simulasi ujian HPS RTL. Lima antaramuka Qsys, iaitu *UART, SPI, FPGA-CTI, FPGA interrupt* dan *boot-from-FPGA* telah berjaya disahkan melalui metodologi pengesahan yang dicadangkan. Berbanding dengan ujian pengesahan HPS RTL, masa simulasi yang lebih pendek telah diperhatikan semasa menguji fungsi yang sama dalam cadangan kaedah pengesahan.

QSYS HPS INTERCONNECT VERIFICATION METHODOLOGY FOR SOC FPGA

Abstract

Field programmable gate array (FPGA) with embedded processor is the future trend for the high performance and low power applications. Qsys HPS interconnect is designed to provide seamless connection between FPGA and the embedded hard processor system (HPS) through a click of mouse. Although bus functional model (BFM) is extensively used in existing Qsys non-HPS interconnect verification methodology, HPS consists of many different interface protocols. The task of develop and validate the BFMs become the bottleneck in verification. A new Qsys HPS interconnect verification methodology has been proposed in this research project. In the proposed verification methodology, the Qsys HPS interconnect will be integrated into HPS RTL verification test bench through a pin switch architecture. Besides, the Qsys design is also integrated into HPS RTL simulation test flow. Five different Qsys interface designs, namely UART, SPI, FPGA-CTI, FPGA interrupt and boot-from-FPGA have been successfully verified using the proposed verification methodology. Shorter simulation time has been observed while testing same function in the proposed verification methodology as compared to HPS RTL verification test.

CHAPTER 1

INTRODUCTION

1.1 Research Background

Soft-core microprocessors such as Altera's Nios™ II and Xilinx's MicroBlaze™ have been widely embedded in Field Programmable Gate Array (FPGA) logics. However, the lack of application specific integrated circuit (ASIC) silicon efficiency has made soft-core microprocessor not suitable for low power applications and high performance applications. Moreover, it has been estimated that more than 50% of future FPGA designs will have embedded processor (Garibay, 2012). In response to these demands, a Hard Processor System (HPS) has been embedded in Altera Cyclone™ V System-on-Chip (SoC) FPGA to achieve ASIC silicon efficiency while maintaining FPGA programmable flexibility.

The following major components in the HPS have been shown in Figure 1.1 (Altera Corporation - A, 2012):

1. Microprocessor unit (MPU) subsystem, which has ARM Cortex™-A9 MPCore processor and L2 cache.
2. SDRAM controller subsystem which supports double data rate 2 (DDR2), double data rate 3 (DDR3) and low-power double data rate 2 (LPDDR2).
3. Flash memory controller
4. On-chip memories

5. Support peripherals including clock manager, reset manager, system manager, scan managers, FPGA manager, DMA controller, timers and watchdog timers.
6. Interface peripheral including Ethernet Media Access Controller (EMAC), Universal Serial Bus (USB), Universal Asynchronous Receiver Transmitter (UART), Inter Integrated Circuit (I2C), Controller Area Network (CAN), Serial Parallel Interface (SPI) and General Purpose Input Output (GPIO).

In order to allow masters in the FPGA to communicate with slaves in the HPS and vice versa, the HPS-FPGA Bridge is built with a high performance AXI bus with a configurable data width of 32, 64 and 128 bits (Altera Corporation - A, 2012). 3 AXI bridges, namely HPS-to-FPGA Bridge, FPGA-to-HPS Bridge and a fixed 32bits data width Lightweight HPS-to-FPGA Bridge are shown in Figure 1.2

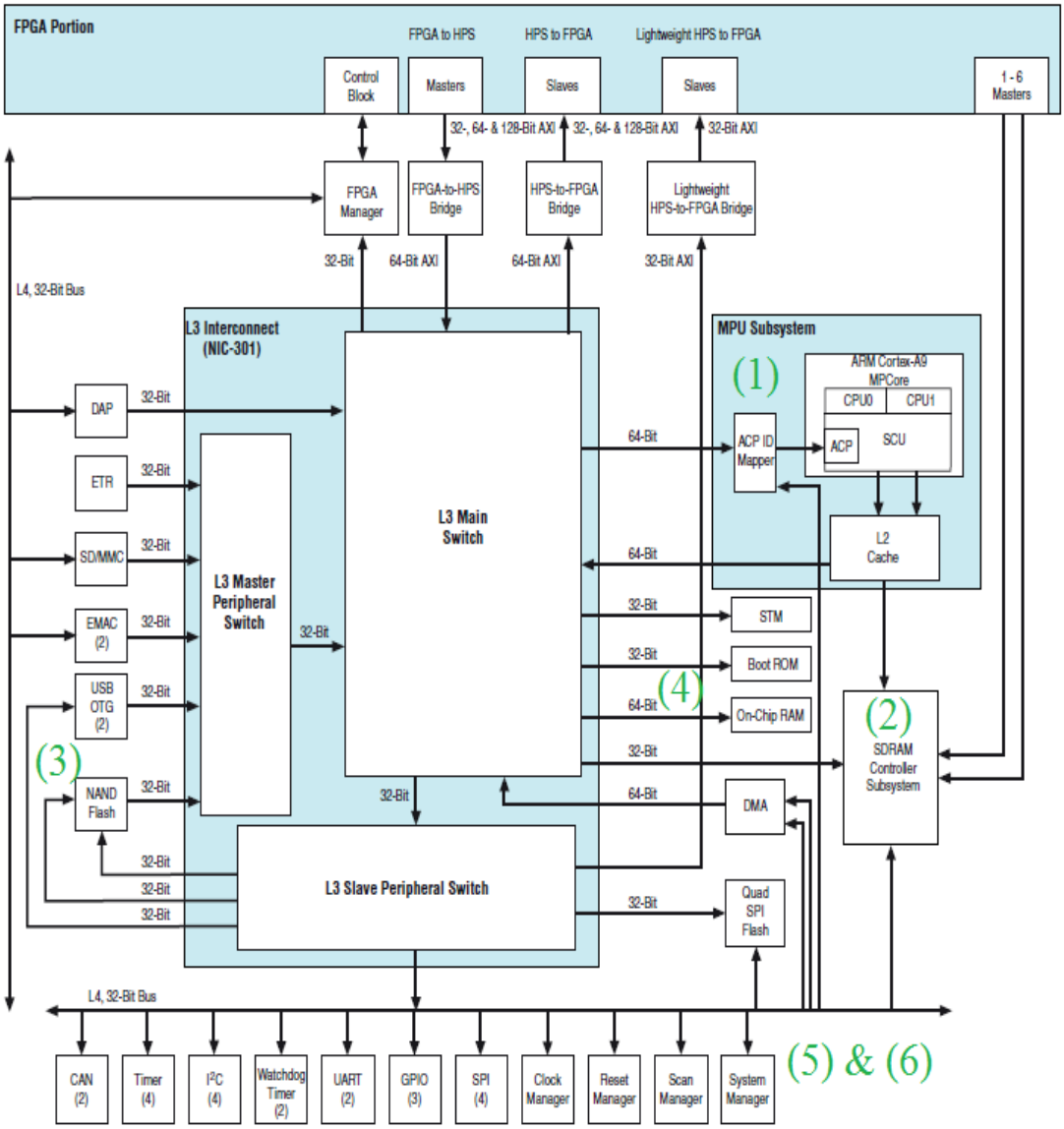

Figure 1.1: HPS components block diagram (Altera Corporation - A, 2012)

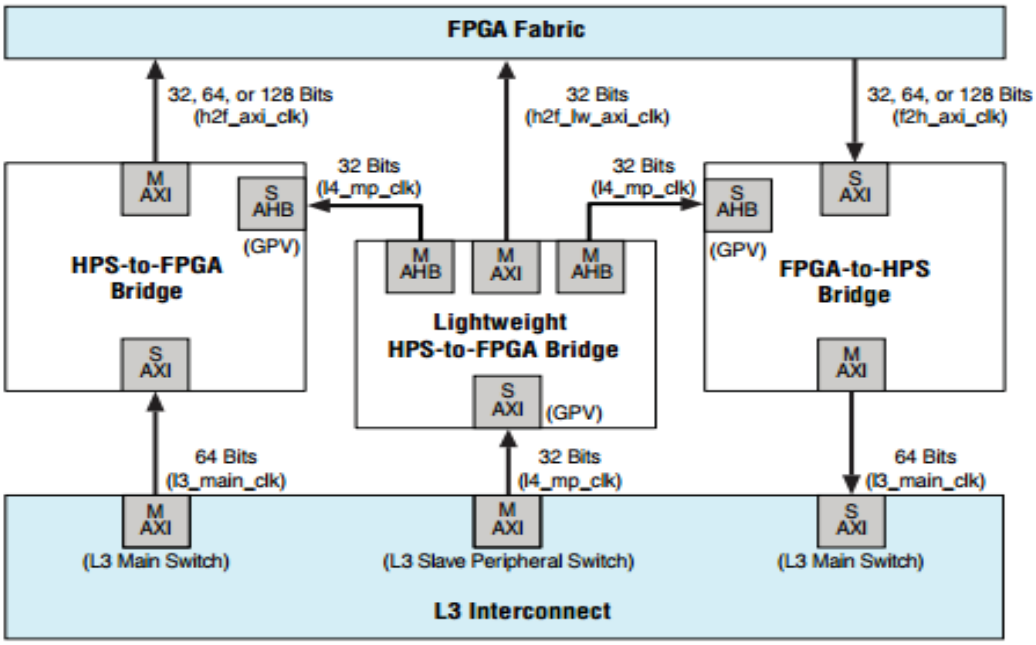

Figure 1.2: HPS-FPGA AXI bridges (Altera Corporation - A, 2012)

More and more complex design will be produced by FPGA designers in very tight time-to-market window. In order to achieve that, the design flow has to be simplified. As such, system level design abstraction is preferred over intellectual property (IP) level design abstraction, whereby FPGA designers can develop higher performance design without extensive knowledge of on-chip interconnects networks. (Garibay, 2012) Design productivity improvement with higher level of design abstraction is illustrated in Figure 1.3. In realizing this, Altera Qsys system integration tool is designed to capture system level hardware designs at higher level of abstraction. The definition and integration of the customized HDL components which includes Altera or 3rd party IP cores, verification IP and other custom design modules will be automated through Qsys (Altera Corporation - B, 2012).

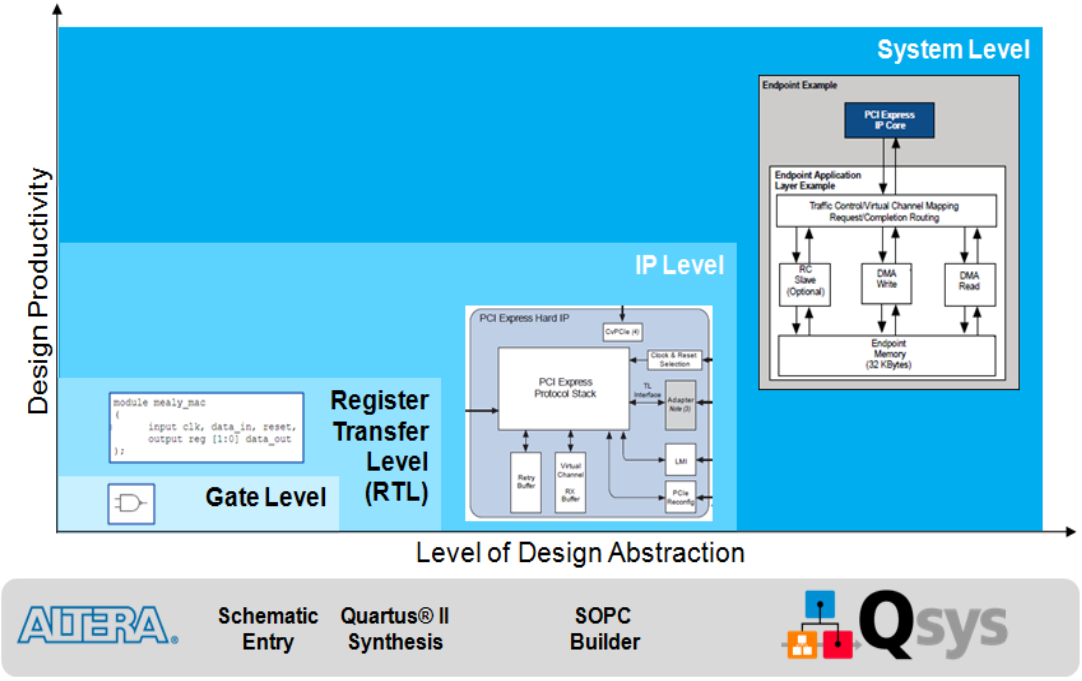

Figure 1.3: Level of design abstraction versus Design productivity (Garibay, 2012)

Network on chip (NoC) architecture has been designed in Qsys to implement system transactions. Flexible network interconnect and various packet formats are the key features of Qsys NoC architecture. Traditional interconnect is handled by individual component of the system. High development cost for redesigning every individual component in the system will be incurred whenever the interface protocol changes. On the other hand, the information is encapsulated at each layer of protocol stack in NoC architecture. Therefore only the particular layer that supports new feature would get impacted whenever there is a new change to the protocol (Altera Corporation, 2011).

The task of integrating all subsystems and IP functions has been made even more complex because of bigger scale of the design. Traditionally, IP-level

abstraction is used by designers to produce a FPGA design prior to Qsys. In IP-level abstraction, the components of the system are manually connected through hardware description language (HDL) coding. Longer time is needed to design with this approach and it is also prone to human error. On the other hand, the instantiation and parameterization of various system components can be easily done through Qsys GUI or a scripted system description TCL and then the connectivity between components will be handled by Qsys automatically.

Various HPS interconnects has been designed by Qsys for FPGA designers to connect HPS to other FPGA components. However, verification of Qsys HPS interconnect must be done before enabling HPS interconnect in Qsys and allowing FPGA designers to use it. This is important to make sure the HPS-FPGA connectivity generated by Qsys is accurate and verified.

1.2 Problem Statement

Bus functional model is used in the existing Qsys non-HPS interconnect verification such as Avalon interfaces and advanced extensible interface (AXI). The BFM is a non-synthesizable software model of actual IP design. It is used to drive and sample transaction according to IP specifications as well as responding to received transactions. HPS and FPGA IP design components are the two main components for Qsys HPS interconnect. Extensive knowledge in the ARM processor architecture and various third party IPs are required for BFM development for HPS and its various peripherals. Besides, large amount of verification work is also needed

to verify the BFM (Mitchell, 2001). Due to these reasons, a new verification methodology will be researched and developed for Qsys HPS interconnect.

1.3 Objectives

The main goal of this Master project is to develop a verification methodology for Qsys HPS interconnect in SoC FPGA.

In order to achieve the goal, these objectives have been set:

1. To research and develop verification methodology for Qsys HPS interconnect.
2. To implement Qsys HPS interconnect verification methodology during HPS hardware RTL development.
3. To demonstrate Qsys HPS interconnect verification test case executing on the new methodology.

1.4 Scope of Work

The whole Qsys software verification consists of HPS design compilation in Quartus™ and interconnectivity testing between HPS and FPGA. The interconnectivity testing between HPS and FPGA will be covered as the scope of this thesis, which includes the following:

1. To evaluate current hardware and software verification methodologies and determine the best implementation for Qsys HPS interconnect verification.

2. To analyze current HPS hardware RTL verification test bench and determine the required modification to enhance it for Qsys HPS interconnect verification.
3. To create test cases to verify the implementation of the methodology.

The context of this Master project is limited to Qsys HPS interconnect verification in SoC FPGA, made the finding only applicable to Qsys HPS interconnect verification in SoC FPGA, thus it cannot generalize to other interconnect verification on SoC FPGA.

1.5 Thesis Organization

Chapter 1 focuses on the introductory background discussion and followed by the problem statement. Besides, the project aim and scope are also mentioned. The remainder of the thesis is organized as follows.

Chapter 2 discusses relevant literature review on past and existing verification methodologies. This includes understanding of interconnect in VLSI design and interconnect verification. Besides that, the Qsys interconnect is also reviewed in this chapter.

Chapter 3 outlines the existing Qsys non-HPS interconnect verification methodology and explains the development of Qsys HPS interconnect verification methodology. The implementation steps of the integration of Qsys HPS interconnect in HPS RTL verification are also detailed in this chapter.

Chapter 4 presents the 5 interfaces verification test results using the proposed Qsys HPS interconnect verification methodology. Besides, the verification methodologies comparison among Qsys HPS interconnect, Qsys non-HPS interconnect as well as HPS RTL will be also discussed in this chapter.

The last chapter, Chapter 5 concludes project findings and the contribution of the research. Lastly proposals for further work are also discussed.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the beginning of the chapter, the reasons to integrate multicore CPU into a FPGA to form SoC FPGA will be reviewed. Subsequently, types of Network-on-Chip (NoC) for SoC integration especially Qsys will be discussed. Besides, various verification methodologies will be reviewed in the following sub-chapters. Finally, all of the reviewed verification methodologies will be summarized at the end of the chapter.

2.2 Evolution of FPGA to SoC FPGA

The electronic systems are implemented in 3 ways, namely software centric system, hardware centric system and alternative hybrid system. In software centric systems, a microprocessor/microcontroller is used to run the software. Although bug fixing and requirements change are easier and faster to modify the system, it is unsuitable for high performance and low power applications. More importantly, product differentiation would be a problem if the software is copied by competitor and run on the same hardware (Biran, 2012).

Meanwhile, the hardware design and software which run on the hardware are developed concurrently in hardware centric design, in which it is also called

application-specific ICs (ASICs). Highest system speeds and greatest energy efficiency are always associated with ASIC product, but highly skilled IC design team and advanced process nodes are required to design a good ASIC product. Besides, changing the hardware for bug fixing or requirements change are extremely expensive (Biran, 2012).

The third is the alternative hybrid system is called field programmable gate array (FPGA). Although FPGA programming is not as easy as software programming, task is completed faster and less energy consumed in FPGA than in software. Besides, shorter time is required for bug fixing or requirements change in FPGA and it is far more economical way than ASICs. Speed and power efficiency are the two main FPGA disadvantages if compared to ASIC (Kuon & Rose, 2007).

Enabled by the advancing silicon process technology, multi-core CPU subsystem has been included in FPGA and become system-on-chip (SoC) FPGA; multiple applications hardware together with CPU is added into ASSPs/Microcontroller to become SoC ASSPs; while ASICs become more and more expensive and limited to very high volume applications. Between SoC FPGA and SoC ASSP, the product differentiation from software down to the hardware level can be provided by SoC FPGA only through its programmable logic fabric (Biran, 2012).

2.2 Network-on-Chip (NoC)

As semiconductor process technology enter submicron, the effect of interconnect has become a significant factor than the transistor. And back in 1998, Samani has already highlighted that interconnect as one of the key challenges for SoC development due to the lack of accurate interconnect modeling (Samani, 1998). Most of the SoC implement rack based system interconnects whereby a bunch of parallel signal connections are tied together to formed a 'bus', hence also called system bus as shown in Figure 2.1. The bus type of interconnects are no longer scaled well in deep submicron process technology, especially timing closure (Furber, 2005). Besides, as SoC grows larger and sophisticated, the functionality of the each components and the communication between the components are harder to be fully verified. To overcome this, an organized methodology is required when designing SoC (Sheets, et al., 2001).

There are several work have been done on addressing the problem of SoC interconnect. The evaluation of the mesh based and butterfly fat tree (BFT) interconnect topologies was done in multi-processor SoC platform (Pande, Grecu, Jones, Ivanov, & Saleh, 2004). Meanwhile another on-chip interconnect system called chip level arbitration and switching system (CLASS) has been designed by Freescale Semiconductor. Multi-masters are connected with multi-slaves, therefore allowing multiple parallel master-slave access (Goren & Netanel, 2006). However, CLASS can be only applied to the SoC design developed by Freescale Semiconductor because it is a proprietary standard.