Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

A Two-Dimensional Extensible Bus Technology and Protocol for VLSI Processor Core

A thesis presented in partial fulfilment of the requirements for the degree of

Master of Engineering

In

Computer and Electronic Engineering

at School of Engineering and Advanced Technology,

Massey University, Albany

New Zealand

By

Loke Chun Eng

2011

Main Advisor:

Dr. S.M. Rezaul Hasan

Abstract

Intellectual property (IP) core design modularity and reuse in Very-Large-Scale-Integration (VLSI) silicon have been the key focus areas in design productivity improvement in order to shorten product development lead time as well as minimize design error on new product [11]. The System-On-Chip (SoC) design approach has been adopted in microprocessor design flow with many functional blocks reuse in silicon. SoC has the advantage of cost efficiency and higher fabrication yield. The fundamental building block of SoC is the interconnection of intellectual property (IP) core through a shared bus to establish an on-chip communication. As IP core integration is severely constraint by silicon wafer sizes (cost per die), the right level of integration is never an easy decision. System-in-Package (SiP) addresses this drawback with package level IP core integration. However, SiP has the drawback of lower fabrication yield which results in higher manufacturing cost [6]. In order to address these issues, a new level of integration has been suggested in order to reduce the drawbacks of SiP and SoC approaches. This new integration methodology is also known as System-in-System (SiS) which emulates SoC and SiP at the system level.

The thesis contains a detailed treatment on the processor architecture and SoC used. The design methodologies have been discussed too.

The thesis also contains treatment on the verification methodologies and technologies that are used in design validation.

Research includes the design of two dimensional XBUS system for external IP core integration on SoC. The thesis proposed a system level bus for IP integration through the XBUS. As there are multiple ways of integrating IP core at the system level, the XBUS is limited to two channels (hence two dimensional) in order to simplify implementation complexities.

Based on experimental results, the proposed method can be introduced as a very promising method for the design of SoC and various other high-performance computer systems.

Acknowledgement

First and foremost, I would like to offer my deepest gratitude to the supervisor of this research: Dr. S.M. Rezaul Hasan, who, with his guidance allowed for the completion of this dissertation. Without his help and support throughout the research it would have been impossible to complete.

As usual, the unconditional support of my family and loved ones is something always appreciated; as such, I would like to acknowledge my mother and father; sister and friends. Their support, both direct and indirect, provided a bastion of confidence during times of difficulty.

For those who I have gained knowledge from indirectly, your work has provided a rich source of information that has furthered my own abilities, and I thank you.

Lastly, I would like to thank the staff and lecturers of Massey University's School of Engineering and Advanced Technology at Albany for the interest shown in the project and their freely given advice.

Working towards the Master's Degree in Massey University was the most important, amazing and astonishing experience in my life. This research and training has completely changed the way of my thinking toward problem solving.

Table of contents

Abstract	2
Acknowledgement	3
Table of contents	4
List of illustrations	6
Chapter 1: Introduction	10
1.1 Problem Description	10
1.2 Motivation	11
1.3 Extensible bus (XBUS)	13
1.4 The thesis contribution	15
Chapter 2: Literature Review	17
2.1 Global Bus I Architecture	17
2.2 Global Bus II Architecture	18
2.3 Bi-FiFo Bus Architecture	18
2.4 Crossbar Switch Bus Architecture	19
2.5 IBM CoreConnect Bus Architecture	20
2.6 The development of DTP-XBUS-2 as SoC-SiP Hybrid	22
2.7 Conclusion	25
Chapter 3: System Environment and Organization	26
3.1 System Architecture – The Big Picture	26
3.2 Instruction-Level Parallelism (ILP), Thread-Level Parallelism (TLP) and System-Level Parallelism (SLP)	의 26
3.3 DTP-XBUS-2 System Overview	29
3.4 Processor Local Interconnect Bus Standard and Implementation	33
3.5 The Data Transfer Protocol (DTP) Memory Architecture	39
3.6 The Two-Dimensional Extensible Bus (XBUS-2) Architecture	43
3.7 SPARC V9 and the Data Transfer Protocol (DTP)	49
3.8 Ultra-High-Bandwidth Data Transfer Operation	55
3.9 Power-On Framework	58
3.10: Conclusion	60
Chapter 4: Verification Concepts	61
4.1 Minimal Verification Requirements	61

4.2 Test Methods		51
4.3 Verification Technolog	gies6	57
4.4 Verification Methodol	ogies7	'0
4.5 Verification Environme	ent7	'1
4.6 Conclusion		'5
Chapter 5: DTP-XBUS-2 Verif	ication7	'6
5.1 Memory System Verifi	ication7	'6
5.2 Interfacing with the M	lemory7	'8
5.3 DTP-XBUS-2 Functiona	al Verification 8	30
5.4 SPARC V9 Functional \	/erification 8	32
5.5 SystemC Wrapper and	Reference Model 8	38
5.6 Programming Languag	ge Interface 8	39
5.7 Verilog Wrapper and S	SPARC V9 Core)1
5.8 Verification Environme	ent9)2
5.9 Main Test Bench for D	TP-XBUS-2)3
5.10 System Verification C	Component (SVC)9)7
5.11 Conclusion)0
Chapter 6: Experimental Res	ults 10)1
6.1 Introduction)1
6.2 DTP-XBUS-2 Power-Or	n Test Results 10)1
6.3 DTP-XBUS-2 Complete	e Verification 10)4
6.4 DTP-XBUS-2 SoC Perfo	rmance Analysis 10)9
6.5 Conclusion		.2
Chapter 7: Conclusion and F	uture work 11	.3
7.1 Conclusion		.3
7.2 Future work		.3
Abbreviations	A 11	.5
Hardware Implementation	В11	.7
Clock Strip Analysis	C 13	35
Linker Script	D14	13
Startup Script	E14	15
ISS Program	F14	16
Bibliography		19

List of illustrations

Figure 1.1: A complete System-on-the-chip
Figure 1.2: Typical IC design flow 12
Figure 1.3: Conventional System Level Bus
Figure 1.4: DTP-XBUS-2
Figure 2.1: Global Bus I Architecture
Figure 2.2: Global Bus II Architecture
Figure 2.3: Bi-FiFo Bus Architecture
Figure 2.4: Crossbar Switch Bus Architecture
Figure 2.5: IBM CoreConnect Bus 20
Figure 2.6: Electric field distribution of second order mode in SiP. (a) Long Period Coplanar
Electromagnetic Bandgap Power Planes (LPC-EBG) (b) LPC-EBG with multi via ground
surface perturbation lattice (MV-GSPL)
Figure 2.7: Differential rates of system IC upgrades
Figure 2.8: SiP system interconnect routing architecture
Figure 2.9: Radiative electric field of common-mode current varying with the distance
arranged strips, clock frequency f=500MHz22
Figure 2.10: Spectral density of radiative electric field of common-mode current varying
from fc to 10fc, fc=100MHz. The distance from a clock strip to other strip is $\lambda/1623$
Figure 2.11: Clock strip analysis and S-Parameters. Refer Appendix C 23
Figure 2.12: Clock strip analysis for package connector and S-Parameters. Refer Appendix C
Figure 3.1: Thread-Level Parallelism (TLP). The figure shows the starts of Strand 1, Strand 2,
Strand 3 and Strand 4 arbitrarily and sequentially at t_1,t_2,t_3 and t_4 respectively after time t_0
on a single TLP processing core
Figure 3.2: Instruction-Level Parallelism (ILP). The figure shows the starts of Strand 1, Strand
2, Strand 3 and Strand 4 arbitrarily and synchronously at t_1 after time t_0 on a single ILP
processing core
Figure 3.3: System- Level Parallelism (SLP). The figure shows the starts of Strand 1, Strand 2,
Strand 3 and Strand 4 arbitrarily and synchronously at t_1,t_2,t_3 and t_4 respectively after time
$t_{0} \text{on} \text{multiple TLP processing cores}.$
Figure 3.4: DTP-XBUS-2 System Overview. PCX and CPX are the Processor-to-Cache-
Crossbar and Cache-Crossbar-to-Processor interfaces respectively. Fast Simplex Link (FSL) is
used as a uni-directional point-to-point high-speed communication. Local Memory Bus
(LMB) is used as the interface to on-chip Block RAM (BRAM). Processor Local Bus (PLB) is
used as the interface that interconnects multiple IP cores
Figure 3.5: Cache Organization
Figure 3.6: Local Bus Interconnect Implementation with XBUS-2

Figure 3.7: Central Bus core	35
Figure 3.8: The initiation of Address Cycle arbitrarily at time t_1 after t_0 . For this cycle, the	е
Request Phase, Transfer Phase and Address Acknowledgment Phase take $t_2 - t_1$, $t_3 - t_2$, and $t_3 - t_2$, $t_3 - t_3 - t_3$.	and
t ₄ – t ₃ , time intervals respectively	36
Figure 3.9: The initiation of Data Cycle arbitrarily at t_1 after t_0 . For this cycle, the Transfe	er
Phase and Data Acknowledgment Phase take $t_2 - t_1$ and $t_3 - t_2$ time intervals respectivel	y. 36
Figure 3.10: Master Request Schematic	38
Figure 3.11: M_Request of three Master devices.	38
Figure 3.12: Schematic representation of DTMP transfer	40
Figure 3.13: Memory addressing modes with DTMP	41
Figure 3.14: Memory Organization for DTMP	42
Figure 3.15: Byte write control circuit	43
Figure 3.16: Example of a Bus-based Communication Architecture	44
Figure 3.17: Tristate Buffer based Bidirectional Signals	45
Figure 3.18: XBUS-2 Architecture	46
Figure 3.19: XBUS-2 Data Frame	47
Figure 3.20: Snapshot of XBUS-2 CRC Generation Circuit.	48
Figure 3.21: Core Block Diagram	49
Figure 3.22: Integer Pipelining Operation	49
Figure 3.23: Floating Pipeline stages	50
Figure 3.24: Instruction Fetch Unit	51
Figure 3.25: Execution Unit	53
Figure 3.26: Load Store Unit	54
Figure 3.27: On-chip System Monitor	56
Figure 3.28: Frame Extension for collision detection prior to frame bursting.	57
Figure 3.29: Frame Burst	58
Figure 3.30: Framework packages	58
Figure 3.31: Memory Initialization Sequence (Hex)	58
Figure 3.32: Linker script	59
Figure 4.1: Functional Test	62
Figure 4.2: Structural Test (Overview)	64
Figure 4.3: Scan chain in structural test	65
Figure 4.4: Structural Tester minimum requirements	66
Figure 4.5: Verification Environment	72
Figure 4.6: Interface Verification Component	73
Figure 4.7: Module/System Verification Component	74
Figure 5.1: Test Generation	76
Figure 5.2: Built-in Self Test (BIST)	77
Figure 5.3: Algorithmic Built-in-Self-Test (AGBIST)	77
Figure 5.4: Memory Test bench	79
Figure 5.5: Memory partition	81
Figure 5.6: XBUS-2/Sub-bus Test bench	82
Figure 5.7: simICS	85
Figure 5.8: Generic	87
Figure 5.9: PLI functions	90

Figure 5.10: SPARC V9 Golden Model	91
Figure 5.11: Verification Environment	92
Figure 5.12: Verification Components	93
Figure 5.13: Interface Verification Component	95
Figure 5.14: Module Monitor	98
Figure 6.1: Experimental Setup	101
Figure 6.2: SPARC V9 expected instruction fetch waveform	103
Figure 6.3: Memory Test Results	103
Figure 6.4: OPB Boot-loader	104
Figure 6.5: Single frame transfer.	104
Figure 6.6: Verification Coverage	105
Figure 6.7: System setup	106
Figure 6.8: Truecolor composite	107
Figure 6.9: First attempt enhancement	107
Figure 6.10: Histogram Accumulation Class examination	108
Figure 6.11 Accumulation Class Sampling	108
Figure 6.12: Truecolor composite enhancement with a contrast stretch	109
Figure 6.13: Single Core DTP-XBUS-2 SoC setup	109
Figure 6.14: Dual Core DTP-XBUS-2 SoC	110
Figure 6.15: DTP-XBUS-2 SoC with External GPU IP core.	110
Figure 6.16: Performance Analysis	111
Figure B.1: DTP-XBUS-2 Top-level illustration	126
Figure B.2: Synthesized DTP-XBUS-2.	127
Figure B.3: DTP-XBUS-2 Data Transmitter implementation	127
Figure B.4: Synthesized DTP-XBUS-2 Data Transmitter implementation	128
Figure B.5: DTP-XBUS-2 Receiver implementation	128
Figure B.6: Synthesized DTP-XBUS-2 receiver.	129
Figure B.7: DTP-XBUS-2 CRC	129
Figure B.8: Synthesized DTP-XBUS-2 CRC	129
Figure B.9: DTP-XBUS-2 Transmit control	130
Figure B.10: Synthesized DTP-XBUS-2 Transmit Control	130
Figure B.11: DTP-XBUS-2 Receive control	131
Figure B.12: Synthesized DTP-XBUS-2 Receive Control	131
Figure B.13: DTP-XBUS-2 CRC Checker	132
Figure B.14: Synthesized DTP-XBUS-2 CRC Checker	132
Figure B.15: DTP-XBUS-2 Data Interface	133
Figure B.16: Synthesized DTP-XBUS-2 Data Interface	133
Figure B.17: DTP-XBUS-2 implemented in ML505 Virtex-5 FPGA	134
Figure C.1: Air-box setup	135
Figure C.2: Copper Net179 setup	136
Figure C.3: Copper Net178 setup	136
Figure C.4: Copper plane 2 Setup	137
Figure C.5: Copper plane 1 setup	137
Figure C.6: FR4 Epoxy setup	138
Figure C.7: Vacuum box setup	139

39
10
10
ļ 1
ļ 1
12