

Table of Contents

- 1 Introduction** **1**
- 1.1 Motivations 2
- 1.2 Contributions 2
- 1.3 Thesis Organization 3

- 2 Background** **5**
- 2.1 Introduction to FPGAs 5
 - 2.1.1 What are FPGAs? 5
 - 2.1.2 Applications of FPGAs 6
 - 2.1.3 A Generic Architecture Model of FPGAs 6
 - 2.1.4 Programming Technologies and Reprogrammability 9
 - 2.1.5 FPGA Design Flow 9
 - 2.1.6 Verification Techniques for FPGA Designs 13
 - 2.1.7 FPGA-Specific Hardware Debugging Techniques 15
- 2.2 Reconfigurable Computing 20
 - 2.2.1 What is Reconfigurable Computing? 21
 - 2.2.2 Reconfigurable Computing Architectures 24
 - 2.2.3 Classifications of Reconfigurable Computing Architectures 25
- 2.3 Hardware Reconfiguration Techniques 28
 - 2.3.1 Hardware Reconfiguration 28
 - 2.3.2 Full and Partial Reconfiguration 29
 - 2.3.3 Static and Dynamic Reconfiguration 30
 - 2.3.4 Rate of Reconfiguration Activities 37
 - 2.3.5 Single- and Multi-Context Reconfiguration 38
 - 2.3.6 Time of Configuration Generation 39
 - 2.3.7 Time of Configuration Scheduling 40
 - 2.3.8 Sequential and Parallel Reconfiguration 41
- 2.4 Run-Time Reconfigurable Design Issues 42
 - 2.4.1 RTR Applications of FPGAs 43
 - 2.4.2 Modeling and Simulation Techniques for RTR Designs 44
 - 2.4.3 Online Scheduling and Placement of Tasks 45
 - 2.4.4 Run-Time Area Defragmentation 47

2.4.5	RTR-Supporting Design Frameworks	48
2.5	Chapter Summary	48
3	A Java Framework for Modeling and Simulation of RTR Designs	51
3.1	Motivations	51
3.2	RTR Hardware Modeling and Simulation	52
3.3	Java-based Co-Simulation and Emulation Environment	54
3.3.1	Rationale	55
3.3.2	Simulation Features	56
3.3.3	Simulation Semantics	60
3.3.4	Dynamic Task Creation and Removal	61
3.3.5	Dynamic Loading and Unloading of Modules	62
3.3.6	Interfacing with Hardware for Co-Emulation	63
3.3.7	Interfacing with Other Software Components	64
3.4	Examples of Hardware Modeling using JCOSEE	64
3.4.1	D Flip-Flop with Synchronous Clear and Clock Enable	64
3.4.2	Inter-Task Communication: Producer and Consumer	67
3.4.3	Communication Model Refinement	70
3.5	Chapter Summary	73
4	A Method for FPGA-based Co-Emulation Modeling	75
4.1	Hardware Infrastructure for Co-Emulation	75
4.1.1	Motivations and Related Work	75
4.1.2	JTAG-based Communication Link	77
4.1.3	Hardware Infrastructure	78
4.1.4	Synchronization Levels	83
4.1.5	Analysis of FPGA Resource Usage	84
4.1.6	Performance Analysis of JTAG Communication Interface	87
4.1.7	On-Chip Monitoring for Performance Analysis	90
4.2	Java API for Co-Emulation Modeling	91
4.2.1	Cycle-Accurate Co-Emulation Modeling	92
4.2.2	Transaction-Based Co-Emulation Modeling	96
4.3	Performance Evaluation of Two Co-Emulation Modeling Approaches	100
4.3.1	IP Cores and Testbenches	101
4.3.2	Experimental Results and Analysis	104
4.4	Distribution of IP Cores for Evaluation	106
4.4.1	Motivations and Related Work	106
4.4.2	Bitstream-level IP Delivery	107
4.4.3	Application Scenario	108
4.4.4	Security Considerations	108
4.5	Chapter Summary	109

5	Platform Development for RTR Co-Emulation	111
5.1	Considerations for Developing RTR Co-Emulation Platforms	111
5.1.1	Xilinx Modular Design Flow for Partial Reconfiguration	112
5.1.2	Use of Bus Macros for Inter-Slot Connections	113
5.1.3	RTR Models and Implementation Styles for Co-Emulation	117
5.1.4	FPGA Pin Assignments of External I/O Peripherals	119
5.1.5	Device Reconfiguration Handling	120
5.1.6	Architecture Models of FPGA-based RTR Systems	122
5.2	JTAG-based Hardware Platform	127
5.2.1	Motivation	127
5.2.2	Slot-based FPGA Floorplanning	128
5.3	PCI-based Hardware Platform with RTR Support	129
5.3.1	The Hardware Architecture Model	129
5.3.2	The Proof-of-Concept FPGA Platform	131
5.3.3	PCI-based Communication	132
5.3.4	Partial Bitstream Download via PCI-ICAP	133
5.3.5	The HIF-TB Communication	134
5.3.6	The DUT-TB Communication	135
5.3.7	Software Support	135
5.3.8	Synthesis Results	136
5.3.9	Performance Evaluation	137
5.4	Chapter Summary	137
6	Case Studies	139
6.1	Runtime-Loadable IP Cores for Co-Emulation	139
6.1.1	Cycle-accurate co-emulation using a Virtex-II chip	140
6.1.2	Transaction-based co-emulation using a Virtex-E chip	141
6.2	Run-time Reconfigurable Coprocessor	143
6.3	Chapter Summary	146
7	Conclusion and Future Work	147
7.1	Conclusion	147
7.2	Future Work	148