

Contents

Part I VERIFICATION OF A MODERN PROCESSOR

1	VERIFICATION OF A MODERN PROCESSOR	3
1.1	The Birth of the Microprocessor	3
1.2	Verification Throughout the Processor Life-cycle	6
1.3	Verification of a Modern Processor: a Case Study	8
1.4	Looking Ahead	9
1.5	Summary	11
	References	12
2	THE VERIFICATION UNIVERSE	13
2.1	Pre-silicon Verification	14
2.1.1	From specification to microarchitectural description	14
2.1.2	Verification through logic simulation	15
2.1.3	Formal verification	19
2.1.4	Logic optimization and equivalence verification	24
2.1.5	Emulation and beyond	25
2.2	Post-silicon Validation	26
2.2.1	Structural testing	28
2.2.2	Functional post-silicon validation	33
2.3	Runtime Verification	35
2.4	Summary	38
	References	39

Part II FUNCTIONAL POST-SILICON VERIFICATION

3	POST-SILICON VALIDATION OF PROCESSOR CORES	45
3.1	Traditional post-silicon validation in industry	45
3.2	The Reversi Test Generation System	52
3.3	Reversible and Non-reversible Instructions	54
3.3.1	Arithmetic and logic instructions	54

3.3.2	Load/store instructions	57
3.3.3	Branch instructions	58
3.3.4	Control register manipulation	60
3.3.5	Floating point instructions	62
3.3.6	Limitations	62
3.3.7	Reversi Generator	63
3.4	Example	65
3.5	Experimental Framework	66
3.5.1	Performance Evaluation	68
3.5.2	Design Error Coverage	70
3.6	Summary	72
	References	73
4	POST-SILICON VERIFICATION OF MULTI-CORE PROCESSORS	75
4.1	Overview of Multi-core Processor Architectures	76
4.2	The Challenge of Multi-core Processor Verification	79
4.3	Cache Coherence Verification Using String Matching	83
4.4	Verification of Memory Consistency Through Constraint Graph Analysis	89
4.5	Summary	92
	References	92
5	CONSISTENCY VERIFICATION USING DATA COLORING	95
5.1	Introduction	96
5.2	Dacota Overview	97
5.3	Activity Logging	99
5.3.1	Access vector	100
5.3.2	Core Activity Log	101
5.3.3	Activity Logging Example	103
5.4	Policy Validation Algorithm	105
5.4.1	Access Log Aggregation	105
5.4.2	Graph Construction	106
5.4.3	Consistency Graph Analysis	110
5.4.4	Error Detection Examples	111
5.4.5	Checking Algorithm Requirements	113
5.5	Strengths and Limitations	114
5.5.1	Debugging with Dacota	114
5.5.2	Design Considerations	115
5.6	Experimental Evaluation of Dacota	115
5.6.1	Design Error Coverage	116
5.6.2	Performance Evaluation	116
5.6.3	Area Evaluation	123
5.7	Summary	124
	References	125

Part III RUNTIME VERIFICATION FOR MODERN MICROPROCESSORS

6	RUNTIME VERIFICATION WITH PATCHING AND HARDWARE CHECKERS	129
6.1	Analysis of Escaped Errors in Commercial Processors	129
6.2	Classification of Runtime Verification Solutions	132
6.3	DIVA: Dynamic Verification of Microprocessors	135
6.3.1	Checker core operation	136
6.3.2	DIVA in action	137
6.3.3	Benefits and limitations	139
6.4	Runtime Verification of Simple Cores with Argus	141
6.5	Hardware Patching Approaches for Runtime Verification	144
6.6	Conclusions	148
	References	149
7	HARDWARE PATCHING WITH FIELD-REPAIRABLE CONTROL LOGIC	151
7.1	Introduction	151
7.2	Field-Repairable Control Logic Overview	153
7.2.1	Pattern Generation	154
7.2.2	Matching Flawed Configurations	155
7.2.3	Pattern Compression Algorithm	157
7.2.4	Processor Recovery	161
7.2.5	Example	161
7.3	Design Flow	163
7.3.1	Overview of the Design Framework	163
7.3.2	Verification Methodology	165
7.3.3	Control Signal Selection	166
7.3.4	Automatic Signal Selection	167
7.3.5	Performance-Critical Execution	168
7.4	Trusted Hardware Design with Semantic Guardians	168
7.4.1	Combining Semantic Guardians and Hardware Patching	172
7.5	Experimental Evaluation	173
7.5.1	Experimental Framework	173
7.5.2	Design Defects	176
7.5.3	Specificity of the Matcher	177
7.5.4	State Matcher Area and Timing Overheads	180
7.5.5	Performance Impact of Degraded Mode	181
7.5.6	Semantic Guardian Framework Analysis	182
7.6	Summary	187
	References	188

8	RUNTIME VERIFICATION IN MULTI-CORES	189
8.1	Dynamic Verification of Memory Consistency	189
8.2	Caspar: A Multi-core Patching Solution	193
8.3	Caspar’s Design	194
8.3.1	Detection and Coverage	196
8.3.2	Recovery and Bypass	198
8.3.3	Checkpointing	198
8.4	Post-silicon Debugging with Caspar	199
8.5	Experimental Evaluation	201
8.5.1	Error Resiliency Analysis	201
8.5.2	Checkpointing Overhead	202
8.5.3	Caspar Recovery Performance	203
8.5.4	Area Overhead	205
8.6	Summary	205
	References	206
9	ENSURING CORRECTNESS IN FUTURE MICROPROCESSORS	207
9.1	Advances and Trends in Processor Validation	207
9.2	A Proactive Approach to Verification	208
	References	213
	Index	223



<http://www.springer.com/978-1-4419-8033-5>

Post-Silicon and Runtime Verification for Modern
Processors

Wagner, I.; Bertacco, V.

2011, XVII, 224 p., Hardcover

ISBN: 978-1-4419-8033-5