

Contents

Part I Introduction

1	Introduction	3
1.1	Overview	3
1.2	System Chip Trends	4
1.2.1	Process Technologies	4
1.2.2	SOC Design Process	6
1.2.3	SOC Verification and Validation	10
1.3	Post-Silicon SOC Debug	14
1.4	Problem Statement	20
1.5	Proposed Approach and Book Organization	20
1.6	Book Contributions	21
1.7	Summary	22
	References	23

Part II The Complexity of Debugging System Chips

2	Post-silicon Debugging of a Single Building Block	27
2.1	Behavior of a Single Building Block	27
2.1.1	Formal Definitions	27
2.1.2	Execution Behavior	30
2.2	Complicating Factors for Debugging	31
2.2.1	Limited Observability and Controllability	31
2.2.2	Undefined Substate and Outputs	32
2.2.3	State Comparison	33
2.2.4	Transient Errors	34
2.3	Summary	35
	References	35
3	Post-silicon Debugging of Multiple Building Blocks	37
3.1	Communication Between Two Building Blocks	37
3.1.1	Overview	37
3.1.2	Synchronous Communication	41

3.1.3	Asynchronous Communication	43
3.1.4	SOC Communication Protocols	47
3.1.5	Variable Communication Duration	50
3.2	Resource Sharing Between Building Blocks	52
3.3	Complicating Factors for Debugging	55
3.3.1	Non-determinism at the Clock-Cycle, Handshake, and Transaction Level	56
3.3.2	Uncertain Errors	58
3.3.3	No Instantaneous, Distributed, Global Actions	60
3.4	Summary	62
	References	63

Part III The CSAR Debug Approach

4	CSAR Debug Overview	67
4.1	Introduction	67
4.1.1	Communication-Centric Debug	68
4.1.2	Scan-Based Debug	70
4.1.3	Abstraction-Based Debug	73
4.1.4	Run/Stop-Based Debug	75
4.2	CSAR Debug Analysis	78
4.3	CSAR Debug Infrastructure Requirements	81
4.3.1	Communication-Centric Debug	81
4.3.2	Scan-Based Debug	84
4.3.3	Abstraction-Based Debug	84
4.3.4	Run/Stop-Based Debug	85
4.3.5	Debug Requirements Overview	85
4.4	Summary	86
	References	86
5	On-Chip Debug Architecture	87
5.1	Overview	87
5.2	Debug Control and Status Interconnect	90
5.2.1	Test Access Port and Associated Controller	91
5.2.2	Test Control Block	94
5.2.3	Global Mode Control	96
5.2.4	Test Point Register	96
5.2.5	TAP-DTL Bridge	101
5.3	Monitoring the Communication	102
5.3.1	Overview of a DTL Monitor	102
5.3.2	DTL Front End	103
5.3.3	Data Matcher	104
5.3.4	Event Encoder	106
5.3.5	Event Sequencer	106

5.4	Controlling the Communication	108
5.4.1	Overview of a DTL Protocol Specific Instrument	108
5.4.2	DTL PSI Mask	109
5.4.3	DTL PSI Test Point Register	112
5.4.4	DTL PSI Event Generator	114
5.4.5	Example DTL PSI Timing Diagram	114
5.5	Event Distribution Interconnect	116
5.5.1	Overview	116
5.5.2	EDI Node	117
5.5.3	EDI Clock Domain Crossing Module	118
5.5.4	TAP-EDI Bridge	119
5.6	Debug Wrapper	120
5.7	Test Wrapper	121
5.7.1	Overview	121
5.7.2	Memory Test Wrapper	121
5.7.3	Primary Input/Output Unit	123
5.7.4	Scan Input Multiplexer	125
5.7.5	Local Clock Gate	126
5.7.6	Protocol-Specific Controller	127
5.8	Clock and Reset Control	128
5.8.1	Overview	128
5.8.2	CRGU Test Control Block	131
5.8.3	Clock Control Slices	132
5.8.4	Reset Generation Unit	135
5.9	Summary	136
	References	137
6	Design-for-Debug Flow	139
6.1	Overview	139
6.2	DfD Tool Architecture	140
6.2.1	Overview	140
6.2.2	Tool Architecture	142
6.2.3	CSAR Configuration Classes	143
6.2.4	CSAR Software Classes	145
6.2.5	External Libraries	146
6.3	Module Implementation or Generation	147
6.4	Debug Wrapper Generation	148
6.4.1	Tool Overview	148
6.4.2	Debug Wrapper Configuration	148
6.4.3	Debug Wrapper Generation Process	151
6.4.4	Executing the Debug Wrapper Generation Process	154
6.5	Other Tools in the DfD Flow	154
6.6	Summary	155
	References	155

7	Off-Chip Debugger Software	157
7.1	Overview	157
7.1.1	CSARDE Requirements	157
7.1.2	Software Architecture	158
7.1.3	CSARDE Design Concepts	159
7.2	The SOC Manager	161
7.2.1	Overview	161
7.2.2	SOC Environment Abstraction	162
7.2.3	SOC Customization Support	164
7.2.4	SOC Mode Model	165
7.2.5	SOC State Access	167
7.3	The Abstraction Manager	168
7.3.1	Overview	168
7.3.2	Structural Abstraction	170
7.3.3	Data Abstraction	174
7.3.4	Behavioral Abstraction	176
7.3.5	Temporal Abstraction	177
7.4	The Scripting Engine	183
7.5	The User Interfaces	184
7.6	Summary	187
	References	187

Part IV Case Studies

8	Case Studies	191
8.1	CSAR DfD in Industrial SOCs	191
8.1.1	Co-Processor Array SOC	193
8.1.2	PNX8525 and CODEC SOCs	194
8.1.3	Xetal-II SOC	197
8.1.4	En-II SOC	198
8.2	CSAR SOC Overview	200
8.2.1	CSAR SOC Application	200
8.2.2	CSAR SOC Hardware Architecture	201
8.2.3	CSAR SOC Clock Domains	203
8.3	Application of the CSAR DfD Flow	205
8.3.1	Overview	205
8.3.2	DfD and Tool Configuration Effort	206
8.3.3	DfD Flow Execution Time	206
8.3.4	CSARDE Configuration	207
8.4	Evaluating the Complicating Factors for Debugging	209
8.4.1	Silicon Area Cost	209
8.4.2	CSAR SOC Observability and Controllability	210
8.4.3	Undefined Substate and State Comparison	216
8.4.4	Transient Errors	218

8.4.5	Non-determinism at Clock-Cycle, Handshake, and Transaction Levels	218
8.4.6	Uncertain Errors	223
8.4.7	No Instantaneous, Distributed, Global Actions	224
8.5	Use Cases	225
8.5.1	Overview	225
8.5.2	Debugging a Permanent, Certain Error	225
8.5.3	Debugging a Transient, Certain Error	226
8.5.4	Debugging a Transient, Uncertain Error	230
8.6	Summary	231
	References	231

Part V Related Work, Conclusion, and Future Work

9	Related Work	235
9.1	Internal Observability and Controllability	235
9.1.1	Intrinsic Physical and Optical Observability	235
9.1.2	Intrinsic Functional Observability and Controllability	236
9.1.3	DfD for Internal Observability and Controllability	237
9.2	Execution Control	242
9.2.1	Deterministic Architectures	242
9.2.2	Deterministic Replay	243
9.2.3	DfD for Execution Control	244
9.3	Debug Standardization	245
9.4	Debug Tool Support	246
9.4.1	DfD Tool Support	246
9.4.2	Debug Application Programmer's Interfaces	246
9.4.3	Debugger Tools	247
9.5	Debug Algorithms	249
	References	250
10	Conclusion and Future Work	257
10.1	Conclusions	257
10.2	Future Work	259
	References	261

Appendix A	Design-for-Debug Flow (Continued)	263
A.1	Test Wrapper Generation	263
A.1.1	Tool Overview	263
A.1.2	Test Wrapper Configuration	263
A.1.3	Test Wrapper Generation Process	265
A.1.4	Executing the Test Wrapper Generation Process	267
A.2	Top-Level Integration	268
A.2.1	Tool Overview	268
A.2.2	Top-Level Configuration	269

- A.2.3 Top-Level Integration Process 270
 - A.2.4 Executing the Top-Level Integration Process 272
 - A.3 Chip-Level Integration 273
 - A.3.1 Tool Overview 273
 - A.3.2 Chip-Level Configuration 273
 - A.3.3 Chip-Level Integration Process 275
 - A.3.4 Executing the Chip-Level Integration Process 277
 - A.4 Boundary-Scan Level Integration 277
 - A.4.1 Tool Overview 277
 - A.4.2 Boundary-Scan-Level Configuration 277
 - A.4.3 Boundary-Scan-Level Integration Process 280
 - A.4.4 Executing the Boundary-Scan-Level Integration Process . . . 281
 - A.5 DfD Configuration Files 282
 - References 284
- Appendix B CSAR SOC 285**
 - B.1 NoC Specification 285
 - B.2 Producer Code 287
 - B.3 Consumer Code 288
 - B.4 Module Implementation Parameters 291
- Appendix C CSARDE Grammars and Scripts 293**
 - C.1 CSARDE Tool Control Language Grammar 293
 - C.2 CSARDE Event Sequencer Grammar 295
 - C.3 CSARDE Scripts 295
 - References 303
- Glossary 305**
- Index 307**

Debugging Systems-on-Chip
Communication-centric and Abstraction-based
Techniques

Vermeulen, B.; Goossens, K.

2014, XV, 311 p. 127 illus., 7 illus. in color., Hardcover

ISBN: 978-3-319-06241-9