

Daniel D. Gajski • Samar Abdi
Andreas Gerstlauer • Gunar Schirner

Embedded System Design

Modeling, Synthesis and Verification

 Springer

Acknowledgments

This book was in the making for many years: from concepts to methodologies to experiments. Many generations of researchers at the Center for Embedded Systems at UCI participated in finding and proving what works and what does not. We would like to thank the members of the first generation that established basic principles of embedded systems: Frank Vahid, Sanjiv Narayan, Jie Gong and Smita Bakshi. We would also like to acknowledge the second generation that brought us SpecC and System on Chip Environment: Jianwen Zhu, Rainer Doemer, Lukai Cai, Haobo Yu, Sequin Zhao, Dongwan Shin, and Jerry Peng. And the third generation that made Embedded System Environment available: Lochi Yu, Hansu Cho, Yongyun Hwang, Ines Viskic. In addition, we would like to acknowledge the NISC team: Mehrdad Reshadi, Bitu Gorjiara and Jelena Trajkovic for their high-level synthesis contributions and Pramod Chandraria for his work on design drivers.

We would also like to thank Quoc-Viet Dang, who helped us with book formatting, figure creation, generation, and without whom this book would not be possible. We also want to thank our editors Matt Nelson and Brian Thill who made the sentences readable and ideas flow without interruptions. We also want to thank Simone Lacina from grafikdesign-lacina.de for an excellent and artistic cover.

However, the highest credits go to Grace Wu and Melanie Kilian for making our center work flawlessly while we were working and thinking about the book.

Last but not the least, we would like to thank Carl Harris from Springer for encouragement and asking at every conference in the last 5 years the same question: "When is the Orange book coming?"

Contents

Preface	v
Acknowledgments	xi
List of Figures	xix
List of Tables	xxv
1. INTRODUCTION	1
1.1 System-Design Challenges	1
1.2 Abstraction Levels	3
1.2.1 Y-Chart	3
1.2.2 Processor-Level Behavioral Model	5
1.2.3 Processor-level structural model	7
1.2.4 Processor-level synthesis	10
1.2.5 System-Level Behavioral Model	13
1.2.6 System Structural Model	14
1.2.7 System Synthesis	14
1.3 System Design Methodology	18
1.3.1 Missing semantics	20
1.3.2 Model Algebra	21
1.4 System-Level Models	23
1.5 Platform Design	27
1.6 System Design Tools	29
1.7 Summary	32
2. SYSTEM DESIGN METHODOLOGIES	35
2.1 Bottom-up Methodology	35
2.2 Top-down Methodology	37
2.3 Meet-in-the-middle Methodology	38

2.4	Platform Methodology	40
2.5	FPGA Methodology	43
2.6	System-level Synthesis	44
2.7	Processor Synthesis	45
2.8	Summary	47
3.	MODELING	49
3.1	Models of Computation	50
3.1.1	Process-Based Models	52
3.1.2	State-Based Models	58
3.2	System Design Languages	65
3.2.1	Netlists and Schematics	66
3.2.2	Hardware-Description Languages	66
3.2.3	System-Level Design Languages	68
3.3	System Modeling	68
3.3.1	Design Process	69
3.3.2	Abstraction Levels	71
3.4	Processor Modeling	72
3.4.1	Application Layer	73
3.4.2	Operating System Layer	75
3.4.3	Hardware Abstraction Layer	78
3.4.4	Hardware Layer	80
3.5	Communication Modeling	83
3.5.1	Application Layer	84
3.5.2	Presentation Layer	88
3.5.3	Session Layer	90
3.5.4	Network Layer	92
3.5.5	Transport Layer	93
3.5.6	Link Layer	94
3.5.7	Stream Layer	98
3.5.8	Media Access Layer	99
3.5.9	Protocol and Physical Layers	100
3.6	System Models	102
3.6.1	Specification Model	103
3.6.2	Network TLM	104
3.6.3	Protocol TLM	106
3.6.4	Bus Cycle-Accurate Model (BCAM)	107
3.6.5	Cycle-Accurate Model (CAM)	108

3.7 Summary	109
4. SYSTEM SYNTHESIS	113
4.1 System Design Trends	114
4.2 TLM Based Design	117
4.3 Automatic TLM Generation	120
4.3.1 Application Modeling	122
4.3.2 Platform Definition	123
4.3.3 Application to Platform Mapping	124
4.3.4 TLM Based Performance Estimation	126
4.3.5 TLM Semantics	130
4.4 Automatic Mapping	132
4.4.1 GSM Encoder Application	134
4.4.2 Application Profiling	135
4.4.3 Load Balancing Algorithm	138
4.4.4 Longest Processing Time Algorithm	142
4.5 Platform Synthesis	146
4.5.1 Component data models	147
4.5.2 Platform Generation Algorithm	148
4.5.3 Cycle Accurate Model Generation	151
4.5.4 Summary	152
5. SOFTWARE SYNTHESIS	155
5.1 Preliminaries	156
5.1.1 Target Languages for Embedded Systems	157
5.1.2 RTOS	159
5.2 Software Synthesis Overview	162
5.2.1 Example Input TLM	164
5.2.2 Target Architecture	166
5.3 Code Generation	167
5.4 Multi-Task Synthesis	173
5.4.1 RTOS-based Multi-Tasking	173
5.4.2 Interrupt-based Multi-Tasking	176
5.5 Internal Communication	181
5.6 External Communication	182
5.6.1 Data Formatting	183
5.6.2 Packetization	185
5.6.3 Synchronization	186
5.6.4 Media Access Control	191

5.7 Startup Code	193
5.8 Binary Image Generation	194
5.9 Execution	195
5.10 Summary	196
6. HARDWARE SYNTHESIS	199
6.1 RTL Architecture	201
6.2 Input Models	204
6.2.1 C-code specification	204
6.2.2 Control-Data Flow Graph specification	205
6.2.3 Finite State Machine with Data specification	207
6.2.4 RTL specification	208
6.2.5 HDL specification	209
6.3 Estimation and Optimization	211
6.4 Register Sharing	216
6.5 Functional Unit Sharing	220
6.6 Connection Sharing	224
6.7 Register Merging	227
6.8 Chaining and Multi-Cycling	229
6.9 Functional-Unit Pipelining	232
6.10 Datapath Pipelining	235
6.11 Control and Datapath Pipelining	237
6.12 Scheduling	240
6.12.1 RC scheduling	243
6.12.2 TC scheduling	244
6.13 Interface Synthesis	248
6.14 Summary	253
7. VERIFICATION	255
7.1 Simulation Based Methods	257
7.1.1 Stimulus Optimization	260
7.1.2 Monitor Optimization	262
7.1.3 SpeedUp Techniques	263
7.1.4 Modeling Techniques	264
7.2 Formal Verification Methods	265
7.2.1 Logic Equivalence Checking	266
7.2.2 FSM Equivalence Checking	268

7.2.3	Model Checking	270
7.2.4	Theorem Proving	273
7.2.5	Drawbacks of Formal Verification	275
7.2.6	Improvements to Formal Verification Methods	275
7.2.7	Semi-formal Methods: Symbolic Simulation	276
7.3	Comparative Analysis of Verification Methods	276
7.4	System Level Verification	278
7.4.1	Formal Modeling	280
7.4.2	Model Algebra	282
7.4.3	Verification by Correct Refinement	283
7.5	Summary	285
8.	EMBEDDED DESIGN PRACTICE	287
8.1	System Level Design Tools	287
8.1.1	Academic Tools	289
8.1.2	Commercial Tools	296
8.1.3	Outlook	299
8.2	Embedded Software Design Tools	300
8.2.1	Academic Tools	301
8.2.2	Commercial Tools	303
8.2.3	Outlook	305
8.3	Hardware Design Tools	306
8.3.1	Academic Tools	308
8.3.2	Commercial Tools	314
8.3.3	Outlook	319
8.4	Case Study	319
8.4.1	Embedded System Environment	320
8.4.2	Design Driver: MP3 Decoder	324
8.4.3	Results	327
8.5	Summary	333
	References	335
	Index	349