

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	IRC 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

List of Figures

1.1	Y-Chart	3
1.2	FSMD model	5
1.3	CDFG model	6
1.4	Instruction-set flow chart	8
1.5	Processor structural model	9
1.6	Processor synthesis	11
1.7	System behavioral model	13
1.8	System structural model	15
1.9	System synthesis	16
1.10	Evolution of design flow over the past 50 years	17
1.11	Missing semantics	20
1.12	Model equivalence	22
1.13	SER Methodology	23
1.14	System TLM	25
1.15	System CAM	26
1.16	Platform architecture	28
1.17	General system environment	29
1.18	System tools	31
2.1	Bottom-up methodology	36
2.2	Top-down methodology	37
2.3	Meet-in-the-middle methodology (option 1)	39
2.4	Meet-in-the-middle methodology (option 2)	40
2.5	Platform methodology	41
2.6	System methodology	42
2.7	FPGA methodology	43

2.8	System-level synthesis	44
2.9	Processor synthesis	46
3.1	Kahn Process Network (KPN) example	54
3.2	Synchronous Data Flow (SDF) example	56
3.3	Finite State Machine with Data (FSMD) example	60
3.4	Hierarchical, Concurrent Finite State Machine (HCFSM) example	61
3.5	Process State Machine (PSM) example	64
3.6	System design and modeling flow	69
3.7	Model granularities	71
3.8	Processor modeling layers	73
3.9	Application layer	74
3.10	Operating system layer	75
3.11	Operating system modeling	76
3.12	Task scheduling	77
3.13	Hardware abstraction layer	79
3.14	Interrupt scheduling	80
3.15	Hardware layer	81
3.16	Application layer synchronization	86
3.17	Application layer storage	87
3.18	Application layer channels	88
3.19	Presentation layer	89
3.20	Session layer	91
3.21	Network layer	92
3.22	Communication elements	93
3.23	Link layer	95
3.24	Link layer synchronization	96
3.24	Link layer synchronization (con't)	97
3.25	Media access layer	99
3.26	Protocol layer	100
3.27	Physical layer	101
3.28	System models	102
3.29	Specification model	104
3.30	Network TLM	105
3.31	Protocol TLM	106
3.32	Bus Cycle-Accurate Model (BCAM)	107
3.33	Cycle-Accurate Model (CAM)	108

3.34	Modeling results	110
4.1	A traditional board-based system design process.	114
4.2	A virtual platform based development environment.	115
4.3	A model based development flow of the future.	116
4.4	TLM based design flow.	117
4.5	Modeling layers for TLM.	118
4.6	System synthesis flow with given platform and mapping.	120
4.7	A simple application expressed in PSM model of computation.	122
4.8	A multicore platform specification.	123
4.9	Mapping from application model to platform.	124
4.10	Computation timing estimation.	125
4.11	Communication timing estimation.	128
4.12	Synchronization Modeling with Flags and Events.	128
4.13	Automatically Generated TLM from system specification.	131
4.14	System synthesis with fixed platform.	133
4.15	Application example: GSM Encoder	134
4.16	Application profiling steps.	135
4.17	Profiled statistics of GSM encoder.	137
4.18	Abstraction of profiled statistics into an application graph.	138
4.19	Creation of platform graph.	139
4.20	Flowchart of load balancing algorithm for mapping generation.	140
4.21	Platform graph with communication costs.	142
4.22	LPT cost function computation.	143
4.23	Flowchart of LPT algorithm for mapping generation.	145
4.24	System synthesis from application and constraints.	146
4.25	Flowchart of a greedy algorithm for platform generation.	149
4.26	Illustration of platform generation on a GSM Encoder example.	150
4.27	Cycle accurate model generation from TLM.	152
5.1	Synthesis overview	155
5.2	Software synthesis flow	163
5.3	Input system TLM example	164
5.4	Generic target architecture	166
5.5	Task specification	169
5.6	Software execution stack for RTOS-based multi-tasking	173
5.7	Multi-task example model	175
5.8	Software execution stack for interrupt-based multi-tasking	177

5.9	Interrupt-based multi-tasking example	178
5.10	Internal communication	181
5.11	External communication	183
5.12	Marshalling example	184
5.13	Packetization	185
5.14	Chain for interrupt-based synchronization	187
5.15	Events in interrupt-based synchronization	188
5.16	Polling-based synchronization	190
5.17	Events in polling-based synchronization	190
5.18	Transferring a packet using bus primitives	191
5.19	Binary image generation	195
5.20	ISS-based Virtual platform	196
6.1	HW synthesis design flow	199
6.2	High-level block diagram	201
6.3	RTL diagram with FSM controller	202
6.4	RTL diagram with programmable controller	203
6.5	CDFG for Ones counter	206
6.6	FSMD specification	207
6.7	RTL Specification	208
6.8	Square-root algorithm (SRA)	212
6.9	Gain in register sharing	217
6.10	General partitioning algorithm	218
6.11	Variable merging for SRA example	219
6.12	SRA datapath with register sharing	220
6.13	Gain in functional unit sharing	221
6.14	Functional unit merging for SRA	222
6.15	SRA design after register and unit merging	224
6.16	SRA Datapath with labeled connections	225
6.17	Connection merging for SRA	227
6.18	SRA Datapath after connection merging	227
6.19	Register merging	228
6.20	Datapath schematic after register merging	229
6.21	Modified FSMD models for SRA algorithm	230
6.22	Datapath with chained functional units	231
6.23	SRA datapath with chained and multi-cycle functional units	232
6.24	Functional unit pipelining	234

6.25	Datapath pipelining	236
6.26	Control and datapath pipelining	239
6.27	C and CDFG	241
6.28	ASAP, ALAP, and RC schedules for SRA	243
6.29	RC algorithm	245
6.30	TC algorithm	245
6.31	ASAP, ALAP, and RC schedules for SRA	246
6.32	Distribution graphs for TC scheduling of the SRA example	247
6.33	HW Synthesis timing constraints	249
6.34	FSMD for MAC driver	250
6.35	Custom HW component with bus interface	251
6.36	A typical bus protocol	252
6.37	Transducer structure	253
7.1	A typical simulation environment	257
7.2	A test case that covers only part of the design.	261
7.3	Coverage analysis results in a more useful test case.	262
7.4	Graphical visualization of the design helps debugging.	263
7.5	A typical emulation setup.	263
7.6	Logic equivalence checking by matching of cones.	266
7.7	DeMorgan's law illustrated by ROBDD equivalence.	267
7.8	Equivalence checking of sequential design using product FSMs.	269
7.9	Product FSM for with a reachable error state.	270
7.10	A typical model checking scenario.	270
7.11	A computation tree derived from a state transition diagram.	271
7.12	Various temporal properties shown on the computation tree.	272
7.13	Proof generation process using a theorem prover.	273
7.14	Associativity of parallel behavior composition.	273
7.15	Basic laws for a theory of system models.	274
7.16	Symbolic simulation of Boolean circuits.	277
7.17	System level models.	279
7.18	A simple hierarchical specification model.	280
7.19	Behavior partitioning and the equivalence of models.	280
7.20	Equivalence of models resulting from channel mapping.	281
7.21	Model refinement using functionality preserving transformations	284
8.1	Metropolis framework	289
8.2	SystemCoDesigner tool flow	290

8.3	Daedalus tool flow	292
8.4	PeaCE tool flow	293
8.5	SCE tool flow	295
8.6	NISC technology tools	310
8.7	The SPARK Synthesis Methodology	311
8.8	xPilot Synthesis System	313
8.9	ESE tool flow	320
8.10	System level design with ESE front end	321
8.11	SW-HW synthesis with ESE back end	323
8.12	MP3 decoder application model	324
8.13	MP3 decoder platform SW+4	326
8.14	Execution speed and accuracy trade-offs for embedded system models	328
8.15	MP3 manual design quality	329
8.16	Automatically generated MP3 design quality	330
8.17	Development productivity gains from model automation	331
8.18	Validation productivity gain from using TLM vs. CAM	332

List of Tables

3.1	Processor models	82
3.2	Communication layers	84
4.1	A sample capacity table of platform components.	147
6.1	Input logic table	209
6.2	Output logic table	209
6.3	Variable usage	213
6.4	Operation usage	214
6.5	SRA connectivity	215
6.6	Connection usage table	226
7.1	A comparison of various verification schemes.	278