Contents

Part I Basic Concepts

1 The Nature of Hardware and Software 3
1.1 Introducing Hardware/Software Codesign 3
1.1.1 Hardware 3
1.1.2 Software 5
1.1.3 Hardware and Software 7
1.1.4 Defining Hardware/Software Codesign 11
1.2 The Quest for Energy Efficiency 13
1.2.1 Performance 13
1.2.2 Energy Efficiency 13
1.3 The Driving Factors in Hardware/Software Codesign 15
1.4 The Hardware-Software Codesign Space 17
1.4.1 The Platform Design Space 17
1.4.2 Application Mapping 18
1.5 The Dualism of Hardware Design and Software Design 20
1.6 Modeling Abstraction Level 21
1.7 Concurrency and Parallelism 24
1.8 Summary 26
1.9 Further Reading 27
1.10 Problems 27

2 Data Flow Modeling and Transformation 31
2.1 Introducing Data Flow Graphs 31
2.1.1 Tokens, Actors, and Queues 35
2.1.2 Firing Rates, Firing Rules, and Schedules 37
2.1.3 Synchronous Data Flow (SDF) Graphs 38
2.1.4 SDF Graphs are Determinate 38
2.2 Analyzing Synchronous Data Flow Graphs 39
2.2.1 Deriving Periodic Admissible Sequential Schedules 40
2.2.2 Example: Deriving a PASS for the PAM-4 System 43

2.3 Control Flow Modeling and the Limitations of Data Flow Models 44
2.3.1 Emulating Control Flow with SDF Semantics 45
2.3.2 Extending SDF Semantics 45
2.4 Adding Time and Resources 46
2.4.1 Real-Time Constraints and Input/Output Sample Rate ... 47
2.4.2 Data Flow Resource Model 47
2.4.3 Limits on Throughput 48
2.5 Transformations 50
2.5.1 Multirate Expansion 51
2.5.2 Retiming 52
2.5.3 Pipelining 53
2.5.4 Unfolding 54
2.6 Data Flow Modeling Summary 55
2.7 Further Reading 56
2.8 Problems 57

3 Data Flow Implementation in Software and Hardware 61
3.1 Software Implementation of Data Flow 61
3.1.1 Converting Queues and Actors into Software 61
3.1.2 Software Implementation with a Dynamic Scheduler 66
3.1.3 Example: Four-Point Fast Fourier Transform as an SDF System 68
3.1.4 Sequential Targets with Static Schedule 74
3.2 Hardware Implementation of Data Flow 76
3.2.1 Single-Rate SDF Graphs into Hardware 77
3.2.2 Pipelining 79
3.3 Hardware/Software Implementation of Data Flow 82
3.4 Summary 86
3.5 Further Reading 86
3.6 Problems 87

4 Analysis of Control Flow and Data Flow 89
4.1 Data and Control Edges of a C Program 89
4.2 Implementing Data and Control Edges 92
4.3 Construction of the Control Flow Graph 93
4.4 Construction of the Data Flow Graph 95
4.5 Application: Translating C to Hardware 99
4.5.1 Designing the Datapath 99
4.5.2 Designing the Controller 100
4.6 Single-Assignment Programs 103
4.7 Summary 106
4.8 Further Reading 106
4.9 Problems 107

Part II The Design Space of Custom Architectures

5 Finite State Machine with Datapath 113
5.1 Cycle-Based Bit-Parallel Hardware 113
5.1.1 Wires and Registers 113
5.1.2 Precision and Sign 116
5.1.3 Hardware Mapping of Expressions 118
5.2 Hardware Modules 120
5.3 Finite State Machines 122
5.4 Finite State Machines with Datapath 125
5.4.1 Modeling 125
5.4.2 The FSMD Model As Two Stacked FSM 129
5.4.3 An FSMD Is Not Unique 130
5.4.4 Implementation 132
5.5 FSMD Design Example: A Median Processor 133
5.5.1 Design Specification: Calculating the Median 133
5.5.2 Mapping the Median in Hardware 135
5.5.3 Sequentializing the Data Input 136
5.5.4 Fully Sequentialized Computation 137
5.6 Proper FSMD 141
5.7 Language Mapping for FSMD by Example 143
5.7.1 GCD in GEZEL 143
5.7.2 GCD in Verilog 144
5.7.3 GCD in VHDL 146
5.7.4 GCD in SystemC 148
5.8 Summary 150
5.9 Further Reading 151
5.10 Problems 151

6 Microprogrammed Architectures 157
6.1 Limitations of Finite State Machines 157
6.1.1 State Explosion 157
6.1.2 Exception Handling 158
6.1.3 Runtime Flexibility 159
6.2 Microprogrammed Control 159
6.3 Micro-instruction Encoding 161
6.3.1 Jump Field 161
6.3.2 Command Field 162
6.4 The Micro-programmed Datapath 164
6.4.1 Datapath Architecture 165
6.4.2 Writing Micro-programs 166
6.5 Implementing a Micro-programmed Machine 168
6.5.1 Micro-instruction Word Definition 168
6.6 Micro-program Interpreters 175

6.7 Micro-program Pipelining 180
6.7.1 Micro-instruction Register 181
6.7.2 Datapath Condition-Code Register 182
6.7.3 Pipelined Next-Address Logic 182
6.8 Microprogramming with Microcontrollers 183
6.8.1 System Architecture 183
6.8.2 Example: Bresenham Line Drawing 184
6.9 Summary 189
6.10 Further Reading 190
6.11 Problems 190

7 General-Purpose Embedded Cores 193
7.1 Processors 193
7.1.1 The Toolchain of a Typical Micro-processor 194
7.1.2 From C to Assembly Instructions 194
7.2 The RISC Pipeline 199
7.2.1 Control Hazards 201
7.2.2 Data Hazards 202
7.2.3 Structural Hazards 204
7.3 Program Organization 205
7.3.1 Data Types 205
7.3.2 Variables in the Memory Hierarchy 206
7.3.3 Function Calls 209
7.3.4 Program Layout 212
7.4 Compiler Tools 213
7.4.1 Examining Size 215
7.4.2 Examining Sections 216
7.4.3 Examining Assembly Code 218
7.5 Low-Level Program Analysis 219
7.6 Processor Simulation 222
7.6.1 Instruction-Set Simulation 222
7.6.2 Analysis Based on Execution of Object Code 224
7.6.3 Simulation at Low Abstraction Level 228
7.7 Summary 229
7.8 Further Reading 230
7.9 Problems 230

8 System on Chip 237
8.1 The System-on-Chip Concept 237
8.1.1 The Cast of Players 238
8.1.2 SoC Interfaces for Custom Hardware 239
8.2 Four Design Principles in SoC Architecture 240
8.2.1 Heterogeneous and Distributed Data Processing 241
8.2.2 Heterogeneous and Distributed Communications 242
8.2.3 Heterogeneous and Distributed Storage 243
8.2.4 Hierarchical Control 247

8.3 Example: Portable Multimedia System 247
8.4 SoC Modeling in GEZEL 250
8.4.1 An SoC with a StrongARM Core 251
8.4.2 Ping-Pong Buffer with an 8051 254
8.4.3 UART on the AVR ATMega128 258
8.5 Summary 262
8.6 Further Reading 263
8.7 Problems 263

Part III Hardware/Software Interfaces

9 Principles of Hardware/Software Communication 269
9.1 Connecting Hardware and Software 269
9.2 Synchronization Schemes 270
9.2.1 Synchronization Concepts 271
9.2.2 Semaphore 273
9.2.3 One-Way and Two-Way Handshake 275
9.2.4 Blocking and Non-blocking Data-Transfer 277
9.3 Communication-Constrained Versus Computation-Constrained 279
9.4 Tight and Loose Coupling 281
9.5 Summary 283
9.6 Further Reading 284
9.7 Problems 284

10 On-Chip Busses 287
10.1 On-Chip Bus Systems 287
10.1.1 A Few Existing On-Chip Bus Standards 287
10.1.2 Elements in a Shared Bus 288
10.1.3 Elements in a Point-to-Point Bus 290
10.1.4 Physical Implementation of On-Chip Busses 290
10.1.5 Bus Naming Convention 291
10.1.6 Bus Timing Diagram 292
10.1.7 Definition of the Generic Bus 293
10.2 Bus Transfers 293
10.2.1 Simple Read and Write Transfers 294
10.2.2 Transfer Sizing and Endianess 296
10.2.3 Improved Bus Transfers 298
10.3 Multi-master Bus Systems 302
10.3.1 Bus Priority 304
10.3.2 Bus Locking 305
10.4 Bus Topologies 307
10.4.1 Bus Switches 308
10.4.2 Network On Chip 309
10.5 Summary 311

10.6 Further Reading 312
10.7 Problems 312

11 Microprocessor Interfaces 317
11.1 Memory-Mapped Interfaces 317
11.1.1 The Memory-Mapped Register 317
11.1.2 Mailboxes 320
11.1.3 First-In First-Out Queues 321
11.1.4 Slave and Master Handshakes 322
11.1.5 Shared Memory 323
11.1.6 GEZEL Modeling of Memory-Mapped Interfaces 324
11.2 Coprocessor Interfaces 328
11.2.1 The Fast Simplex Link 330
11.2.2 The LEON-3 Floating Point Coprocessor Interface 332
11.3 Custom-Instruction Interfaces 334
11.3.1 ASIP Design Flow 335
11.3.2 Example: Endianness Byte-Ordering Processor 336
11.3.3 Example: The Nios-II Custom-Instruction Interface 341
11.3.4 Finding Good ASIP Instructions 343
11.4 Summary 347
11.5 Further Reading 348
11.6 Problems 349

12 Hardware Interfaces 353
12.1 The Coprocessor Hardware Interface 353
12.1.1 Functions of the Coprocessor Hardware Interface 353
12.1.2 Layout of the Coprocessor Hardware Interface 354
12.2 Data Design 355
12.2.1 Flexible Addressing Mechanisms 356
12.2.2 Multiplexing and Masking 356
12.3 Control Design 358
12.3.1 Hierarchical Control 359
12.3.2 Control of Internal Pipelining 361
12.4 Programmer's Model = Control Design + Data Design 365
12.4.1 Address Map 366
12.4.2 Instruction Set 366
12.5 Summary 367
12.6 Further Reading 368
12.7 Problems 368

Part IV Applications

13 Trivium Crypto-Coprocessor 375
13.1 The Trivium Stream Cipher Algorithm 375
13.1.1 Stream Ciphers 375
13.1.2 Trivium 377

13.1.3 Hardware Mapping of Trivium 378
13.1.4 A Hardware Testbench for Trivium 382
13.2 Trivium for 8-bit Platforms 383
13.2.1 Overall Design of the 8051 Coprocessor 384
13.2.2 Hardware Platform of the 8051 Coprocessor 385
13.2.3 Software Driver for 8051 389
13.3 Trivium for 32-bit Platforms 393
13.3.1 Hardware Platform Using Memory-Mapped Interfaces 393
13.3.2 Software Driver Using Memory-Mapped Interfaces 397
13.3.3 Hardware Platform Using a Custom-Instruction Interface 399
13.3.4 Software Driver for a Custom-Instruction Interface 403
13.4 Summary 405
13.5 Further Reading 406
13.6 Problems 406

14 AES Co-processor 409
14.1 AES Encryption and Decryption 409
14.2 Memory-Mapped AES Encryption Coprocessor 409
14.2.1 Hardware Interface Operation 411
14.2.2 Programmer's Model 412
14.2.3 Software Driver Design 414
14.2.4 Hardware Interface Design 416
14.2.5 System Performance Evaluation 418
14.3 AES Encryption/Decryption with Custom Instructions 420
14.3.1 AES T-box Reference Implementation 421
14.3.2 AES T-box Custom Instruction Design 424
14.3.3 AES T-box Custom Instruction in GEZEL 426
14.3.4 AES T-box Software Integration and Performance 431
14.4 Summary 433
14.5 Further Reading 433
14.6 Problems 433

15 CORDIC Co-processor 435
15.1 The Coordinate Rotation Digital Computer Algorithm 435
15.1.1 The Algorithm 435
15.1.2 Reference Implementation in C 437
15.2 A Hardware Coprocessor for CORDIC 439
15.2.1 A CORDIC Kernel in Hardware 439
15.2.2 A Hardware Interface for Fast-Simplex-Link Coprocessors 442
15.3 An FPGA Prototype of the CORDIC Coprocessor 446
15.4 Handling Large Amounts of Rotations 448
15.5 Summary 453

15.6 Further Reading 454
15.7 Problems 454

A Hands-on Experiments in GEZEL 457
A.1 Overview of the GEZEL Tools 457
A.2 Installing the GEZEL Tools 458
A.2.1 Installation on a Ubuntu System 458
A.2.2 Installation of Cross-Compiler Tools 460
A.2.3 Compiling GEZEL from Source Code on a 32-bit System 460
A.2.4 Compiling GEZEL from Source Code on a 64-bit System 463
A.3 Running the Examples 464
A.3.1 Examples from FSMD Chapter 464
A.3.2 Examples from Microprogrammed Architectures Chapter 465
A.3.3 Examples from System on Chip Chapter 466
A.3.4 Examples from Microprocessor Interfaces Chapter 468
A.3.5 Examples from Trivium Chapter 469
A.3.6 Examples from AES Chapter 471
A.3.7 Examples from CORDIC Chapter 472

References 473

Index 477

http://www.springer.com/978-1-4614-3736-9

A Practical Introduction to Hardware/Software Codesign

Schaumont, P.
2013, XXII, 482 p., Hardcover
ISBN: 978-1-4614-3736-9