Specification and Design of Embedded Systems

Daniel D. Gajski
Frank Vahid
Sanjiv Narayan
Jie Gong

University of California at Irvine

P T R Prentice Hall
Englewood Cliffs, New Jersey 07632

Contents

Preface **xiii**

Acknowledgements **xvii**

1 Introduction **1**

1.1 Design Representation 2

1.2 Levels of abstraction 4

1.3 Current design methodologies 6

1.4 System-level methodology 10

1.5 System specification and design 13

2 Models and Architectures **15**

2.1 Introduction 15

2.2 Model taxonomy 19

2.3 State-oriented models 20

2.3.1 Finite-state machine 20

2.3.2 Petri net 24

2.3.3 Hierarchical concurrent finite-state machine 27

2.4 Activity-oriented models 28

2.4.1 Dataflow graph 28

2.4.2 Flowcharts 31

2.5 Structure-oriented models 32

2.5.1 Component-connectivity diagram 32

2.6 Data-oriented models 34

2.6.1 Entity-relationship diagram 34

2.6.2 Jackson's diagram 35

2.7 Heterogeneous models 36

2.7.1 Control/data flow graph 36

2.7.2 Structure chart 39

2.7.3 Programming language paradigm 40

2.7.4 Object-oriented model 42

2.7.5 Program-state machine 43

2.7.6 Queueing model 45

2.8 Architecture taxonomy 47

2.9 Application-specific architectures 47

2.9.1 Controller architecture 47

2.9.2 Datapath architecture 48

2.9.3 Finite-state machine with datapath 50

2.10 Processors 51

2.10.1 Complex instruction set computer 51

2.10.2 Reduced instruction set computer 53

2.10.3 Vector machine 55

2.10.4 Very long instruction word computer 56

2.11 Parallel processors 58

2.12 Conclusion 60

2.13 Exercises 61

3 Specification Languages **63**

3.1 Introduction 63

3.2 Characteristics of conceptual models 65

3.2.1 Concurrency 65

3.2.2 State transitions 69

3.2.3 Hierarchy 70
3.2.4 Programming constructs 74
3.2.5 Behavioral completion 75
3.2.6 Communication 77
3.2.7 Synchronization 80
3.2.8 Exception handling 83
3.2.9 Non-determinism 83
3.2.10 Timing 84
3.3 Specification requirements for embedded systems 86
3.4 Survey of specification languages 88
3.4.1 VHDL 88
3.4.2 Verilog 92
3.4.3 HardwareC 93
3.4.4 CSP 96
3.4.5 Statecharts 97
3.4.6 SDL 100
3.4.7 Silage 101
3.4.8 Esterel 103
3.5 SpecCharts 104
3.5.1 Language description 104
3.5.2 Embedded system specification in SpecCharts 108
3.5.3 Equivalent graphical version 110
3.5.4 Possible language extensions 111
3.6 Conclusion and future directions 113
3.7 Exercises 114

4 A Specification Example **117**
4.1 Introduction 117
4.2 Telephone answering machine 118
4.3 Specification capture with SpecCharts 121

4.4 Sample testbench 129
4.5 Advantages of executable specification 132
4.6 Strengths of the PSM model 133
4.6.1 Hierarchy 134
4.6.2 State transitions 134
4.6.3 Programming constructs 134
4.6.4 Concurrency 135
4.6.5 Exception handling 135
4.6.6 Completion 135
4.6.7 Equivalence of state decomposition and code 136
4.7 Experiments 136
4.7.1 Specification capture 136
4.7.2 Specification comprehension 137
4.7.3 Specification quantification 138
4.7.4 Design quality 140
4.8 Conclusion 140
4.9 Exercises 142

5 Translation to VHDL 145

5.1 Introduction 145
5.2 State-transitions 149
5.3 Message-passing communication 152
5.3.1 Blocking message passing 152
5.3.2 Non-blocking message passing 154
5.4 Concurrency 156
5.4.1 Dataflow 156
5.4.2 Fork 159
5.5 Exception handling 161
5.6 Program-state machines to tasks 164
5.6.1 Overview 164

5.6.2 Algorithm 165
5.6.3 Time-shift 167
5.6.4 Synthesis 168
5.7 Conclusion and future directions 168
5.8 Exercises 169

6 System Partitioning 171

6.1 Introduction 171
6.2 Structural versus functional partitioning 172
6.2.1 Structural partitioning 172
6.2.2 Functional partitioning 175
6.3 Partitioning issues 176
6.3.1 Specification abstraction-level 177
6.3.2 Granularity 179
6.3.3 System-component allocation 180
6.3.4 Metrics and estimations 180
6.3.5 Objective functions and closeness functions 182
6.3.6 Partitioning algorithms 183
6.3.7 Output 184
6.3.8 Flow of control and designer interaction 185
6.3.9 Typical system configuration 186
6.4 Basic partitioning algorithms 186
6.4.1 Random mapping 187
6.4.2 Hierarchical clustering 187
6.4.3 Multi-stage clustering 190
6.4.4 Group migration 191
6.4.5 Ratio cut 194
6.4.6 Simulated annealing 196
6.4.7 Genetic evolution 198
6.4.8 Integer linear programming 200

6.5 Functional partitioning for hardware 201
6.5.1 Yorktown Silicon Compiler 201
6.5.2 BUD 205
6.5.3 Aparty 209
6.5.4 Other techniques 213
6.6 Hardware/software partitioning algorithms 214
6.6.1 Greedy algorithms 214
6.6.2 Hill-climbing algorithms 216
6.6.3 A binary constraint-search algorithm 217
6.7 Functional partitioning for systems 219
6.7.1 Vulcan 219
6.7.2 Cosyma 221
6.7.3 SpecSyn 222
6.7.4 Other techniques 225
6.8 Exploring tradeoffs 225
6.9 Conclusion and future directions 227
6.10 Exercises 228

7 Design Quality Estimation **233**
7.1 Introduction 233
7.1.1 Accuracy versus speed 235
7.1.2 Fidelity of estimation 236
7.2 Quality metrics 238
7.2.1 Hardware cost metrics 238
7.2.2 Software cost metrics 239
7.2.3 Performance metrics 240
7.2.4 Other metrics 246
7.3 Hardware estimation 249
7.3.1 Hardware estimation model 249
7.3.2 Clock cycle estimation 251

7.3.3 Control step estimation 260
7.3.4 Execution time estimation 268
7.3.5 Communication rate estimation 272
7.3.6 Area estimation 274
7.3.7 Pin estimation 288
7.4 Software estimation 290
7.4.1 Software estimation model 290
7.4.2 Program execution time 295
7.4.3 Program memory size 296
7.4.4 Data memory size 297
7.5 Estimation techniques in system-level tools 298
7.5.1 BUD 298
7.5.2 Aparty 300
7.5.3 Vulcan 301
7.5.4 SpecSyn 302
7.6 Conclusion and future directions 304
7.7 Exercises 305

8 Specification Refinement 309

8.1 Introduction 309
8.2 Refining variable groupings 310
8.2.1 Variable folding 310
8.2.2 Memory address translation 312
8.3 Channel refinement 313
8.3.1 Characterizing channels and buses 314
8.3.2 Problem definition 315
8.3.3 Bus generation 315
8.3.4 Protocol generation 326
8.4 Resolving access conflicts 330
8.4.1 Arbitration models 330

8.4.2 Arbitration schemes 332
8.4.3 Arbiter generation 333
8.5 Refining incompatible interfaces 335
8.5.1 Problem definition 337
8.5.2 Specifying communication protocols 338
8.5.3 Interface process generation 342
8.5.4 Other approaches for protocol compatibility 350
8.6 Refining hardware/software interfaces 354
8.6.1 Target architecture 356
8.6.2 Variable distribution 357
8.6.3 Interface generation 361
8.6.4 Data access refinement 363
8.6.5 Control access refinement 366
8.7 Conclusion and future directions 369
8.8 Exercises 370

9 System-Design Methodology 373

9.1 Introduction 373
9.2 Basic concepts 374
9.3 An example design methodology 375
9.3.1 Current practice 379
9.3.2 System-level methodology 381
9.4 A generic synthesis system 384
9.4.1 System synthesis 387
9.4.2 ASIC synthesis 389
9.4.3 Logic and sequential synthesis 393
9.4.4 Physical design 395
9.4.5 Software synthesis 395
9.4.6 System database 397
9.5 Conceptualization environment for system design 397

9.6 Conclusion and future directions 402
9.7 Exercises 403

A Answering machine in English **405**

B Answering machine in SpecCharts **409**

Bibliography **424**

Glossary **440**

Index **445**