A Designer's Guide to Asynchronous VLSI

PETER A. BEEREL
University of Southern California/Timeless Design Automation

RECEP O. OZDAG
Fulcrum Microsystems Inc.

MARCOS FERRETTI
PST Electrônica S. A. (Positron)

CAMBRIDGE
UNIVERSITY PRESS

Contents

Acknowledgments — *page* xi

1 **Introduction** — 1

- 1.1 Synchronous design basics — 2
- 1.2 Challenges in synchronous design — 4
- 1.3 Asynchronous design basics — 5
- 1.4 Asynchronous design flows — 6
- 1.5 Potential advantages of asynchronous design — 7
- 1.6 Challenges in asynchronous design — 10
- 1.7 Organization of the book — 11

2 **Channel-based asynchronous design** — 16

- 2.1 Asynchronous channels — 16
- 2.2 Sequencing and concurrency — 24
- 2.3 Asynchronous memories and holding state — 30
- 2.4 Arbiters — 33
- 2.5 Design examples — 36
- 2.6 Exercises — 40

3 **Modeling channel-based designs** — 43

- 3.1 Communicating sequential processes — 44
- 3.2 Using asynchronous-specific languages — 46
- 3.3 Using software programming languages — 47
- 3.4 Using existing hardware design languages — 47
- 3.5 Modeling channel communication in Verilog — 48
- 3.6 Implementing VerilogCSP macros — 55
- 3.7 Debugging in VerilogCSP — 58
- 3.8 Summary of VerilogCSP macros — 61
- 3.9 Exercises — 62

4 **Pipeline performance** 66

4.1 Block metrics 67
4.2 Linear pipelines 69
4.3 Pipeline loops 73
4.4 Forks and joins 79
4.5 More complex pipelines 81
4.6 Exercises 82

5 **Performance analysis and optimization** 84

5.1 Petri nets 84
5.2 Modeling pipelines using channel nets 88
5.3 Performance analysis 90
5.4 Performance optimization 96
5.5 Advanced topic: stochastic performance analysis 100
5.6 Exercises 102

6 **Deadlock** 106

6.1 Deadlock caused by incorrect circuit design 107
6.2 Deadlock caused by architectural token mismatch 108
6.3 Deadlock caused by arbitration 110

7 **A taxonomy of design styles** 116

7.1 Delay models 116
7.2 Timing constraints 118
7.3 Input–output mode versus fundamental mode 119
7.4 Logic styles 119
7.5 Datapath design 123
7.6 Design flows: an overview of approaches 129
7.7 Exercises 132

8 **Synthesis-based controller design** 136

8.1 Fundamental-mode Huffman circuits 136
8.2 STG-based design 146
8.3 Exercises 149

9 **Micropipeline design** 152

9.1 Two-phase micropipelines 152
9.2 Four-phase micropipelines 159
9.3 True-four-phase pipelines 162
9.4 Delay line design 164

9.5 Other micropipeline techniques 168
9.6 Exercises 169

10 Syntax-directed translation 172

10.1 Tangram 173
10.2 Handshake components 174
10.3 Translation algorithm 176
10.4 Control component implementation 177
10.5 Datapath component implementations 178
10.6 Peephole optimizations 187
10.7 Self-initialization 188
10.8 Testability 189
10.9 Design examples 192
10.10 Summary 196
10.11 Exercises 197

11 Quasi-delay-insensitive pipeline templates 200

11.1 Weak-conditioned half buffer 200
11.2 Precharged half buffer 204
11.3 Precharged full buffer 216
11.4 Why input-completion sensing? 217
11.5 Reduced-stack precharged half buffer (RSPCHB) 220
11.6 Reduced-stack precharged full buffer (RSPCFB) 229
11.7 Quantitative comparisons 232
11.8 Token insertion 232
11.9 Arbiter 236
11.10 Exercises 238

12 Timed pipeline templates 240

12.1 Williams' PS0 pipeline 240
12.2 Lookahead pipelines overview 242
12.3 Dual-rail lookahead pipelines 242
12.4 Single-rail lookahead pipelines 247
12.5 High-capacity pipelines (single-rail) 250
12.6 Designing non-linear pipeline structures 253
12.7 Lookahead pipelines (single-rail) 255
12.8 Lookahead pipelines (dual-rail) 257
12.9 High-capacity pipelines (single-rail) 259
12.10 Conditionals 262
12.11 Loops 263
12.12 Simulation results 264
12.13 Summary 266

13 Single-track pipeline templates 267

13.1 Introduction 267
13.2 GasP bundled data 269
13.3 Pulsed logic 270
13.4 Single-track full-buffer template 271
13.5 STFB pipeline stages 275
13.6 STFB standard-cell implementation 283
13.7 Back-end design flow and library development 290
13.8 The evaluation and demonstration chip 290
13.9 Conclusions and open questions 299
13.10 Exercises 300

14 Asynchronous crossbar 304

14.1 Fulcrum's Nexus asynchronous crossbar 305
14.2 Clock domain converter 309

15 Design example: the Fano algorithm 313

15.1 The Fano algorithm 313
15.2 The asynchronous Fano algorithm 321
15.3 An asynchronous semi-custom physical design flow 329

Index 336