Table of Contents

Preface v

Part 1: Evolvable Hardware and GA

Chapter 1. Automated design synthesis and partitioning for adaptive reconfigurable hardware **3**
Ranga Vemuri, Sriram Govindarajan, Iyad Ouaiss, Meenakshi Kaul, Vinoo Srinivasan, Shankar Radhakrishnan, Sujatha Sundaraman, Satish Ganesan, Awartika Pandey, and Preetham Lakshmikathan

1. Introduction 3
2. Adaptive reconfigurable hardware systems 5
    2.1. Reconfigurable computer applications 7
    2.2. Classification of reconfigurable computers 8
3. Design automation for reconfigurable computers 11
    3.1. Design specification model 12
    3.2. Temporal partitioning 13
    3.3. Spatial partitioning 13
    3.4. Interconnection synthesis 14
    3.5. Design synthesis 15
4. The SPARCS system: an overview 18
5. Design specification model in SPARCS 18
    5.1. The unified specification model 19
    5.2. The behavior blocks intermediate format 20
6. Temporal partitioning 20
    6.1. Temporal partitioning for complete reconfiguration 21
    6.2. Temporal partitioning for partial reconfiguration 25
7. Spatial partitioning 27
    7.1. Issues in spatial partitioning 29
    7.2. Spatial partitioning in SPARCS 29
    7.3. Architectural constraint evaluator 31
    7.4. Partitioning engine 32
8. Interconnection synthesis 33
    8.1. Related research 34
    8.2. Interconnection synthesis is SPARCS 34
9. Design synthesis 35

9.1. Interaction between HLS and partitioning 36
9.2. Arbiter synthesis in SPARCS 39
9.3. Integrating logic synthesis and HLS 42
10. Conclusions 44
References 44

Chapter 2. High-performance hardware design and implementation of genetic algorithms 53
Barry Shackleford, Etsuko Okushi, Mitsuhiro Yasuda, Hisao Koizumi, Katsuhiko Seo, Takahashi Iwamoto, and Hiroto Yasuura

1. Introduction 53
2. Modifying the genetic algorithm for efficient hardware implementation 55
2.1. Algorithm explanation 55
2.2. Contrast with original GA 57
2.3. Validity of survival-based genetic algorithm 58
3. Pipelined architecture for survival-based genetic algorithm 60
3.1. Pipeline description 60
3.2. Hardware implementation 63
4. Fitness function examples 68
4.1 Set-coverage problem example 68
4.2. Logic circuit synthesis example 71
5. Prototype and experiment 82
5.1. Prototype 82
5.2. Experiment 83
6. Conclusions 84
References 85

Part 2: Fuzzy Logic Hardware Implementations

Chapter 3. Hardware implementation of intelligent systems 91
Marco Russo and Luigi Caponetto

1. Introduction 91
2. Specifications 93
2.1. Basics 93
2.2. Hardware 95
2.3. Performance 97
2.4. Classification of fuzzy architectures 98
2.5. Classification of artificial neural networks 100
3. Hardware building blocks for computational intelligence 103
3.1 Digital design 103
3.2. Analog design 110

4. Conclusions 115
References 116

Chapter 4. High performance fuzzy processors **121**
Gian Carlo Cardarilli, Roberto Lojacono, Marco Re

1. Introduction 121
2. Hardware implementations classification 123
3. Fuzzification block 127
4. Inference block 132
5. Defuzzification block 133
6. General architecture and performance 141
7. The radar target tracking application 143
8. Conclusions 145
References 145

Chapter 5. A digital fuzzy processor for fuzzy-rule-based systems.. **147**
Davide Falchieri, Alessandro Gabrielli, Enzo Gandolfi

1. Introduction 147
2. Background: fuzzy hardware and software 149
   2.1. Fuzzy hardware 149
   2.2. Fuzzy rule generators: neural and genetic 151
3. A family of fast fuzzy processors 153
   3.1. The 4-input fuzzy processor 153
   3.2. The two-input fuzzy processor 154
4. Genetic rule generator 156
5. Ten-input fuzzy processor performance 158
6. Processor architecture 159
7. Layout Design 161
8. Conclusions 162
References 163

Part 3: Neural Networks Hardware Implementations

Chapter 6. Optimum multiuser detection for CDMA systems using the mean field annealing neural network........ **167**
Po-Rong Chang, Bor-Chin Wang, and Tan-Hsu Tan

1. Introduction 167
2. Mapping the optimum CDMA multiuser detection problem onto a mean-field annealing network 170

3. Equations of motion for Hopfield-like mean field annealing networks 177
4. Numerical implementation for mean-field annealing machines .. 180
5. Programmable analog VLSI implementation of mean-field annealing machines 183
6. Computer simulation results 186
7. Conclusions 190
References 191

Chapter 7. Analog VLSI hardware implementation of a supervised learning algorithm **193**
Gian Marco Bo, Daniele Caviglia, Hussein Chiblé, and Maurizio Valle

1. Introduction 193
2. Analog VLSI for neural computation 194
   2.1. Hardware implementation of neural networks 194
   2.2. Analog on-chip learning implementation 196
3. Neural network and learning algorithm 198
   3.1. The feed-forward network 198
   3.2. The learning algorithm 200
   3.3. Improvements in the learning convergence speed 202
4. Analog on-chip learning architecture 203
   4.1. The synaptic module 205
   4.2. The neuron module 207
5. Analog CMOS implementation 208
   5.1. SLANP circuit implementation 208
   5.2. The physical design and implementation 209
6. Experimental results 210
   6.1. XOR2 experiment 211
   6.2. Classification experiment 211
   6.3. Discussion 213
7. Conclusions 214
References 216

Chapter 8. pRAM: the probabilistic RAM neural processor **219**
Trevor G. Clarkson

1. Introduction 219
2. pRAM development 221
3. On-chip training 224
4. Interface to a host computer 227
5. Applications 229
   5.1. Using neurons in thermal image processing systems 230
   5.2. Neural net object identification 231
   5.3. Character recognition and the use of noise in training 233

5.4. Medical and biometric image analysis 234
5.5. Satellite on-board processing 235
5.6. ATM network control 237
6. Conclusions 239
References 239

Part 4: Algorithms for Parallel Machines

Chapter 9. Parallel subgraph matching on a hierarchical interconnection network 245
Stuart Campbell, Mohan Kumar, Horst Bunke

1. Introduction 245
2. Subgraph isomorphisms 246
2.1. Applications 246
2.2. Graph definitions and terms 247
2.3. Solving the subgraph isomorphism problem 251
3. Hierarchical cliques: a hierarchical interconnection network 254
4. Algorithm design and implementation 256
4.1. The off-line algorithm 257
4.2. The on-line algorithm 263
4.3. Parallelism within the PN algorithm 267
4.4. Parallelising the PN algorithm on the HiC 268
5. Complexity 269
6. Conclusions 273
References 274

About the editors 277
Index of terms 281