

Contents

Part I Evolvable Hardware Practice Martin A. Trefzer and Andy M. Tyrrell

1	Evolution, Development and Evolvable Hardware	3
	Andy M. Tyrrell and Martin A. Trefzer	
1.1	Introduction	3
1.2	Chromosomes and Genes	3
1.3	Genotype and Phenotype	4
1.4	Biological Living Organism Cells	4
	1.4.1 Stages of Development	6
	1.4.2 Characteristics of Biological Development	8
1.5	Mathematical Abstractions of Development	9
	1.5.1 Gene Regulatory Network Abstractions	9
	1.5.2 L-Systems	10
1.6	Developmental Systems in Artificial Systems	12
1.7	Brief Introduction to Evolvable Hardware	14
	1.7.1 Extrinsic Evolvable Hardware	17
	1.7.2 Intrinsic Evolvable Hardware	19
	1.7.3 Models and Modelling	19
1.8	Current Achievements and Challenges	23
	References	23
2	Devices and Architectures for Evolutionary Hardware	27
	Martin A. Trefzer and Andy M. Tyrrell	
2.1	Introduction	27
2.2	Definition of Terms	28
	2.2.1 Architecture, Interconnect and Fabric Structure	29
	2.2.2 Granularity	29
	2.2.3 Mapping Designs	31
2.3	Digital Commercial Architectures	34
	2.3.1 Xilinx vs. Altera	34

2.3.2	Lattice	36
2.3.3	QuickLogic ArcticLink and PolarPro	38
2.3.4	Microsemi (Actel)	39
2.3.5	Elixent (Panasonic) D-Fabrix	40
2.3.6	PACT XPP-III	40
2.3.7	QuickSilver Adapt2400	41
2.3.8	Coherent Logix HyperX	42
2.3.9	Adapteva Parallella	42
2.3.10	Discussion: Digital Commercial Architectures	43
2.4	Digital, Coarse-Grained Research & Development Architectures	44
2.4.1	KressArray	48
2.4.2	RaPiD	48
2.4.3	Colt	48
2.4.4	MATRIX	50
2.4.5	Pleiades	50
2.4.6	REMARC	50
2.4.7	RAA, CHESS	51
2.4.8	MorphoSys	51
2.4.9	PipeRench	51
2.4.10	MONTIUM	52
2.4.11	DReAM	53
2.4.12	Discussion: Coarse-Grained Digital Architectures	53
2.5	Digital, Medium-Grained Research & Development Architectures	54
2.5.1	FIPSOC	54
2.5.2	Sun FleetZERO (TTA)	54
2.5.3	H-Tree SW	55
2.5.4	RAW	56
2.5.5	Discussion: Medium-Grained Digital Architectures	56
2.6	Digital, Fine-Grained Research & Development Architectures	57
2.6.1	PADDI, PADDI-2	57
2.6.2	MOVE32INT (TTA)	58
2.6.3	Garp	58
2.6.4	CellMatrix (PIG)	58
2.6.5	POEtic	59
2.6.6	RISA	59
2.6.7	UbiChip (PERPLEXUS)	60
2.6.8	SABRE (Unitronics)	61
2.6.9	PAnDA	62
2.6.10	Discussion: Fine-Grained Digital Architectures	62
2.7	Discussion: Digital Commercial vs. Digital Research	63
2.8	Analogue Commercial Architectures	66
2.8.1	Zetex TRAC 020LH	68
2.8.2	Lattice ispPAC-10, ispPAC-20, ispPAC-30 and ispPAC-80/81	69
2.8.3	Anadigm AN221E04	69

2.8.4	Programmable Delay Lines	72
2.8.5	Discussion: Analogue Commercial	72
2.9	Analogue Research & Development Architectures	73
2.9.1	FPAA (Lee)	73
2.9.2	FIPSOC	74
2.9.3	Intermediate Frequency Filter	74
2.9.4	Anti-fuse FPAA	75
2.9.5	FPTA, FPTA-2 (HD)	75
2.9.6	FPTA0, FPTA1 and FPTA2 (JPL)	75
2.9.7	HAGEN	77
2.9.8	Hex-FPAA	77
2.9.9	HICANN	78
2.9.10	Discussion: Analogue Research & Development	78
2.10	Analogue Research vs. Analogue Commercial	79
2.11	Final Thoughts on Future Platforms	80
	References	82
3	Representations and Algorithms	89
	Andy M. Tyrrell and Martin A. Trefzer	
3.1	Introduction	89
3.2	Basic Requirements	90
3.2.1	Netlists	91
3.2.2	Bitstreams	94
3.3	Evolutionary Algorithms	94
3.3.1	Genetic Algorithms (GAs)	98
3.3.2	Artificial Genotypes	98
3.4	Selection Schemes	100
3.4.1	Simple Roulette Wheel Selection	100
3.4.2	Tournament Selection	101
3.4.3	Elitism	101
3.5	Reproduction	102
3.5.1	Crossover	102
3.5.2	Single-Point and N-Point Crossover	102
3.5.3	Uniform Crossover	103
3.5.4	Mutation	104
3.6	Schema Theory	105
3.7	Evolutionary Strategies (ES)	106
3.8	Genetic Programming (GP)	107
3.9	Cartesian Genetic Programming (CGP)	111
3.10	Development	112
3.10.1	Benefits of Multicellular Development	113
3.11	Summary	115
	References	116

4	Measurement and Fitness Function	119
	Martin A. Trefzer and Andy M. Tyrrell	
4.1	Introduction	119
4.2	Hardware Measurement	121
4.2.1	Analogue and Digital Signals.....	121
4.2.2	Mismatch and Noise	122
4.2.3	Parasitic Effects	124
4.2.4	Configuration Circuitry.....	124
4.2.5	Bandwidth, Distortion and Accuracy	126
4.3	Test and Verification	128
4.3.1	Input Pattern Order Problem (IPOP)	128
4.3.2	Implications of Transient Effects	129
4.3.3	Effect of Physical Input Location	130
4.3.4	Influence of the Fitness-Measuring Method	130
4.4	Validation of Evolved Circuits	131
4.4.1	Random Input Test Patterns	131
4.4.2	Measuring at Different Hardware Locations	132
4.4.3	Testing at Different Frequencies and Supply Voltages ...	133
4.5	Fitness Functions for Analogue Hardware	133
4.5.1	Random Input Pattern Order	134
4.5.2	Fitness Functions with Physical Meaning.....	134
4.5.3	Absolute vs. Relative Error	135
4.6	Hierarchical Fitness Functions for Digital Hardware	135
4.6.1	Bitwise Fitness Calculation	137
4.6.2	Bitwise Fitness Modified for Hardware (BMH).....	137
4.6.3	Hierarchical If-And-Only-If (HIFF)	139
4.6.4	Hierarchical Bit-String Sampling (HBS)	140
4.7	Multi-objective Optimisation for Real-World Applications	140
4.7.1	Non-dominated Sorting	141
4.7.2	Crowding Distance	142
4.7.3	Multi-objective Evolutionary Loop	143
4.7.4	Selection Schemes.....	143
4.8	Successes of Evolutionary Computation When Fighting Noise and Mismatch	144
4.9	Final Thoughts on Measurement and Fitness.....	145
	References	146

Part II Evolvable Hardware Applications

5	Overcoming Variability Through Transistor Reconfiguration: Evolvable Hardware on the PAnDA Architecture	153
	James A. Walker	
5.1	Introduction	153
5.2	Limitations of Current Field-Programmable Gate Array (FPGA) Architectures	156

5.3	Current Field-Programmable Transistor Array (FPTA) Architectures	157
5.4	Combining Inspiration from Field-Programmable Transistor Arrays (FPTAs) and Variation-Aware Design	158
5.5	PAnDA Architecture	159
5.5.1	Configurable Transistor	160
5.5.2	Combinational Configurable Analogue Block (CCAB)	164
5.6	Case Study: ISCAS C17 Benchmark	167
5.7	Final Thoughts	171
	References	171
6	Functional Equivalence Checking for Evolution of Complex Digital Circuits	175
	Lukáš Sekanina and Zdeněk Vašíček	
6.1	Introduction	175
6.2	Functional Equivalence Checking	177
6.2.1	SAT Problem	177
6.2.2	SAT-Based Functional Equivalence Checking	178
6.2.3	Creating a Circuit to Be Verified from the Parent and Offspring	178
6.2.4	Converting the Circuit to a Logic Formula in CNF	179
6.2.5	Solving the Logic Formula Using a SAT Solver	181
6.2.6	Further Optimisations of Functional Equivalence Checking	181
6.3	Embedding Functional Equivalence Checking into CGP	182
6.3.1	Cartesian Genetic Programming	183
6.3.2	SAT Solver in the Fitness Function	184
6.4	Experimental Results	184
6.4.1	Speedup Against Standard CGP	185
6.4.2	Benchmark Circuits	186
6.5	Final Thoughts	187
	References	188
7	Fault Tolerant Applications	191
	Andy M. Tyrrell	
7.1	Brief Introduction to Fault Tolerance	191
7.2	Bio-inspired Fault Tolerance	195
7.3	A Place for Evolved Fault Tolerance?	196
7.3.1	Method 1: Evolving Fault Tolerance	196
7.3.2	Method 2: Fault Tolerance Using Natural Redundancy	196
7.3.3	Method 3: Fault Evolution	197
7.4	Explicit Fault Tolerance Using Evolvable Hardware	197
7.5	Evolved Fault Tolerance	198
7.6	Results and Discussion	198
7.7	Implicit Fault Tolerance Using Evolvable Hardware	201
7.8	Final Thoughts	206

References 206

8 Principles and Applications of Polymorphic Circuits 209
 Lukáš Sekanina

8.1 Introduction 209

8.2 Polymorphic Components 210

8.2.1 Polymorphic Gates and Their Properties 210

8.2.2 Reconfigurable Polymorphic Chip 212

8.3 Synthesis and Optimisation of Polymorphic Circuits 213

8.3.1 Polymorphic Multiplexing 214

8.3.2 Evolution of Polymorphic Circuits 216

8.4 Applications of Polymorphic Electronics 216

8.4.1 Signal-Processing Circuits 217

8.4.2 Dependability and Testing 218

8.4.3 Other Applications 221

8.5 Final Thoughts 221

References 223

9 A Developmental Image Compression Technique Using Gene Regulatory Networks 225
 Martin A. Trefzer

9.1 Introduction 225

9.2 Standard Image Compression Techniques 227

9.2.1 JPEG and JPEG 2000 Image Compression Standards 227

9.2.2 Vector Quantisation (VQ) 228

9.3 Non-standard Image Compression Techniques 228

9.3.1 Fractal Image Compression 229

9.3.2 Evolutionary Computation for Image Compression 229

9.4 Artificial Developmental System 230

9.4.1 Representation and Gene Regulation 231

9.4.2 Cell Signalling and Growth 233

9.4.3 Cell Specialisation and Structuring 237

9.5 Image Compression Using GRN 237

9.6 Evolutionary Optimisation of the Model 240

9.7 Multi-pass Encoding 241

9.8 Comparison of GCI with JPEG Using the Lena Image 242

9.9 Dynamics of the GRN 245

9.10 Testing GCI on Different Images 246

9.11 Final Thoughts 248

References 251

10 Medical Applications of Evolvable Hardware 253
 Michael A. Lones and Stephen L. Smith

10.1 Introduction 253

10.1.1 Processing Signals from Medical Devices 254

10.1.2 Hardware-Assisted Evolution 254

- 10.1.3 Evolving Medical Devices 256
- 10.1.4 Embedded Evolutionary Algorithms 256
- 10.2 Neurodegenerative Disease Diagnosis and Monitoring 256
 - 10.2.1 Classifying Assessment Data 258
 - 10.2.2 Classification in Hardware 261
 - 10.2.3 Long-Term Monitoring 263
- 10.3 Final Thoughts 267
- References 269

11 Metamorphic Systems: A Schema for Adaptive Autonomous Systems 273

Garrison W. Greenwood and Andy M. Tyrrell

- 11.1 Introduction 273
- 11.2 Metamorphic Systems 274
- 11.3 Event-Driven Architecture 276
 - 11.3.1 The Concept of an Event 276
 - 11.3.2 Time-Driven Versus Event-Driven Systems 277
 - 11.3.3 Event Processing 278
- 11.4 Example Problem 279
 - 11.4.1 DC Motor Dynamics 279
 - 11.4.2 Motor Control 280
- 11.5 A Metamorphic System Approach 286
 - 11.5.1 Achieving Autonomous Behaviour Adaption 286
 - 11.5.2 Assessment Module 288
 - 11.5.3 Detection Mechanism 291
 - 11.5.4 Implementation Ideas 292
- 11.6 Final Thoughts 294
- References 295

12 Hierarchical Networks-on-Chip Architecture for Neuromorphic Hardware 297

Snaider Carrillo, Jim Harkin and Liam McDaid

- 12.1 Introduction 297
 - 12.1.1 Spiking Neural Networks 298
 - 12.1.2 Networks-on-Chip 298
 - 12.1.3 Chapter Outline 300
- 12.2 Related Work 300
- 12.3 Overview of the H-NoC Architecture 302
 - 12.3.1 Neuron Facility: Bottom Level of H-NoC 304
 - 12.3.2 Configuration Bank Registers 306
 - 12.3.3 Spike Compression Technique 309
 - 12.3.4 Tile Facility: Intermediate Level of H-NoC 311
 - 12.3.5 Cluster Facility: Top Level of H-NoC 312
 - 12.3.6 Adaptive-Routing Scheme 315
- 12.4 Example Scenarios and Analysis 317
 - 12.4.1 Feed-Forward Network 317

12.4.2	Recurrent Network	321
12.4.3	Experimental Results and Large-Scale Analysis	325
12.5	Final Thoughts	326
	References	328
13	Evolvable Robot Hardware	331
	Alan F. T. Winfield and Jon Timmis	
13.1	Introduction	331
13.2	A Brief Introduction to Evolutionary Robotics	332
13.2.1	The Evolutionary Robotics Process	332
13.2.2	Evolution of Robot Bodies	334
13.2.3	Evolution of Robot Controllers	336
13.3	Evolvable Robot Hardware, Challenges and Directions	337
13.3.1	An Engineering Approach	338
13.3.2	A Bio-inspired (Modular or Multicellular) Approach	341
13.3.3	Self-healing	343
13.3.4	Integrating the Evolutionary Processes	344
13.3.5	Final Thoughts	346
	References	347
14	Developmental Evolvable Hardware	349
	Pauline C. Haddow	
14.1	Introduction	349
14.2	Brief Survey of Developmental Evolvable Hardware	351
14.3	On-chip Rule-Based Development for Digital Design	352
14.3.1	Cellular Model	353
14.3.2	Virtual sBlock Architecture	356
14.3.3	Implementation of the sBlock Architecture	357
14.3.4	On-chip Development Architecture	358
14.3.5	Counting Example	360
14.4	Gene Switching for Adaptive Intrinsic Analogue Evolvable Hardware	362
14.4.1	Gene-Switching Model	362
14.4.2	Hardware System	367
14.4.3	Adaptive Example	369
14.5	Final Thoughts	370
	References	371
A	Evolvable Hardware Practice	373
A.1	Lab 1: Evolvable Logic on FPGAs	373
A.1.1	Prerequisites for Lab 1	374
A.1.2	Program and Run a Test Bit File (Atlys_Test.bit)	374
A.1.3	The Evolvable Hardware Architecture	378
A.1.4	Evolution of Full Adders	381
A.1.5	Opening EDK and the Evolution System	383
A.1.6	Start Evolving Full Adders	385

- A.1.7 Finding Solutions 386
- A.1.8 Evolution of a 2-bit Adder 387
- A.1.9 Things to Try And Investigate 388
- A.1.10 Function Reference 389
- A.1.11 Bit Manipulation in C 392
- A.2 Lab 1: Evolvable Logic on FPGAs — Solutions 392
 - A.2.1 2-bit Adder Output Calculation 392
 - A.2.2 Elitism 394
 - A.2.3 Crossover 394
 - A.2.4 Solution Source File 396
 - A.2.5 Example Solution 396
- A.3 Lab 2: Fault Tolerance Through Evolvable Logic on FPGAs 398
 - A.3.1 Prerequisites for Lab 2 398
 - A.3.2 Parity Generator Specification 398
 - A.3.3 Fault Tolerance Evolvable Hardware System 399
 - A.3.4 Creating the Fault Tolerant System on FPGAs 404
 - A.3.5 Fault Injection 406
 - A.3.6 Things to Try 407
 - A.3.7 Function Reference 407



<http://www.springer.com/978-3-662-44615-7>

Evolvable Hardware

From Practice to Application

Trefzer, M.A.; Tyrrell, A.M.

2015, XXVII, 411 p. 187 illus., 78 illus. in color.,

Hardcover

ISBN: 978-3-662-44615-7