## Contents

**Preface**  xv  
**List of Abbreviations**  xxi  

1  Introduction to Field Programmable Gate Arrays  1  
1.1 Introduction  1  
1.2 Field Programmable Gate Arrays  2  
1.2.1 Rise of Heterogeneous Computing Platforms  4  
1.2.2 Programmability and DSP  5  
1.3 Influence of Programmability  6  
1.4 Challenges of FPGAs  8  
Bibliography  9  

2  DSP Basics  11  
2.1 Introduction  11  
2.2 Definition of DSP Systems  12  
2.2.1 Sampling  14  
2.2.2 Sampling Rate  14  
2.3 DSP Transformations  16  
2.3.1 Discrete Fourier Transform  16  
2.3.2 Fast Fourier Transform  17  
2.3.3 Discrete Cosine Transform  18  
2.3.4 Wavelet Transform  19  
2.4 Filters  20  
2.4.1 Finite Impulse Response Filter  20  
2.4.2 Infinite Impulse Response Filter  24  
2.4.3 Wave Digital Filters  25  
2.5 Adaptive Filtering  29  
2.5.1 Applications of Adaptive Filters  30  
2.5.2 Adaptive Algorithms  30  
2.5.3 LMS Algorithm  31  
2.5.4 RLS Algorithm  32  
2.5.5 Squared Givens Rotations  36  
2.6 Final Comments  38  
Bibliography  38
Contents

3 Arithmetic Basics 41
  3.1 Introduction 41
  3.2 Number Representations 42
    3.2.1 Signed Magnitude 43
    3.2.2 One’s Complement 43
    3.2.3 Two’s Complement 44
    3.2.4 Binary Coded Decimal 44
    3.2.5 Fixed-Point Representation 45
    3.2.6 Floating-Point Representation 46
  3.3 Arithmetic Operations 47
    3.3.1 Adders 47
    3.3.2 Adders and Subtracters 49
    3.3.3 Adder Final Remarks 51
    3.3.4 Multipliers 51
  3.4 Alternative Number Representations 55
    3.4.1 Signed Digit Number Representation 55
    3.4.2 Logarithmic Number Systems 56
    3.4.3 Residue Number Systems 57
    3.4.4 CORDIC 58
  3.5 Division 59
    3.5.1 Recurrence Division 59
    3.5.2 Division by Functional Iteration 60
  3.6 Square Root 60
    3.6.1 Digit Recurrence Square Root 61
    3.6.2 Square Root by Functional Iteration 61
    3.6.3 Initial Approximation Techniques 62
  3.7 Fixed-Point versus Floating-Point 65
  3.8 Conclusions 66
  Bibliography 67

4 Technology Review 70
  4.1 Introduction 70
  4.2 Implications of Technology Scaling 71
  4.3 Architecture and Programmability 72
  4.4 DSP Functionality Characteristics 74
    4.4.1 Computational Complexity 74
    4.4.2 Parallelism 75
    4.4.3 Data Independence 75
    4.4.4 Arithmetic Requirements 76
    4.4.5 Processor Classification 76
  4.5 Microprocessors 76
    4.5.1 ARM Microprocessor Architecture Family 78
    4.5.2 Parallella Computer 80
  4.6 DSP Processors 82
    4.6.1 Evolutions in DSP Microprocessors 83
    4.6.2 TMS320C6678 Multicore DSP 85
4.7 Graphical Processing Units 86
4.7.1 GPU Architecture 87
4.8 System-on-Chip Solutions 88
4.8.1 Systolic Arrays 89
4.9 Heterogeneous Computing Platforms 91
4.10 Conclusions 92
Bibliography 92

5 Current FPGA Technologies 94
5.1 Introduction 94
5.2 Toward FPGAs 95
5.2.1 Early FPGA Architectures 97
5.3 Altera Stratix® V and 10 FPGA Family 98
5.3.1 ALMs 99
5.3.2 Memory Organization 100
5.3.3 DSP Processing Blocks 101
5.3.4 Clocks and Interconnect 103
5.3.5 Stratix® 10 innovations 103
5.4 Xilinx Ultrascale™/Virtex-7 FPGA Families 103
5.4.1 Configurable Logic Block 104
5.4.2 Memory 105
5.4.3 Digital Signal Processing 106
5.5 Xilinx Zynq FPGA Family 107
5.6 Lattice iCE40isp FPGA Family 108
5.6.1 Programmable Logic Blocks 109
5.6.2 Memory 110
5.6.3 Digital Signal Processing 110
5.7 MicroSemi RTG4 FPGA Family 111
5.7.1 Programmable Logic Blocks 111
5.7.2 Memory 111
5.7.3 Mathblocks for DSP 112
5.8 Design Strategies for FPGA-based DSP Systems 112
5.8.1 DSP Processing Elements 112
5.8.2 Memory Organization 113
5.8.3 Other FPGA-Based Design Guidelines 113
5.9 Conclusions 114
Bibliography 114

6 Detailed FPGA Implementation Techniques 116
6.1 Introduction 116
6.2 FPGA Functionality 117
6.2.1 LUT Functionality 117
6.2.2 DSP Processing Elements 120
6.2.3 Memory Availability 121
6.3 Mapping to LUT-Based FPGA Technology 123
6.3.1 Reductions in Inputs/Outputs 123
6.3.2 Controller Design 125
6.4 Fixed-Coefficient DSP 125
  6.4.1 Fixed-Coefficient FIR Filtering 126
  6.4.2 DSP Transforms 127
  6.4.3 Fixed-Coefficient FPGA Techniques 130
6.5 Distributed Arithmetic 130
  6.5.1 DA Expansion 130
  6.5.2 DA Applied to FPGA 132
6.6 Reduced-Coefficient Multiplier 133
  6.6.1 DCT Example 134
  6.6.2 RCM Design Procedure 134
  6.6.3 FPGA Multiplier Summary 137
6.7 Conclusions 137

Bibliography 138

7 Synthesis Tools for FPGAs 140
  7.1 Introduction 140
  7.2 High-Level Synthesis 141
  7.2.1 HLS from C-Based Languages 143
  7.3 Xilinx Vivado 143
  7.4 Control Logic Extraction Phase Example 144
  7.5 Altera SDK for OpenCL 145
  7.6 Other HLS Tools 147
  7.6.1 Catapult 147
  7.6.2 Impulse-C 147
  7.6.3 GAUT 148
  7.6.4 CAL 148
  7.6.5 LegUp 150
  7.7 Conclusions 150

Bibliography 150

8 Architecture Derivation for FPGA-based DSP Systems 152
  8.1 Introduction 152
  8.2 DSP Algorithm Characteristics 153
  8.2.1 Further Characterization 154
  8.3 DSP Algorithm Representations 157
  8.3.1 SFG Descriptions 158
  8.3.2 DFG Descriptions 158
  8.4 Pipelining DSP Systems 160
  8.4.1 Retiming 160
  8.4.2 Cut-Set Theorem 163
  8.4.3 Application of Delay Scaling 164
  8.4.4 Calculation of Pipelining Period 167
  8.4.5 Longest Path Matrix Algorithm 167
  8.5 Parallel Operation 170
  8.5.1 Unfolding 173
  8.5.2 Folding 174
  8.6 Conclusions 178

Bibliography 179
# Contents

## 9 Complex DSP Core Design for FPGA  180

9.1 Introduction  180  
9.2 Motivation for Design for Reuse  181  
9.3 Intellectual Property Cores  182  
9.4 Evolution of IP Cores  184  
9.4.1 Arithmetic Libraries  185  
9.4.2 Complex DSP Functions  187  
9.4.3 Future of IP Cores  187  
9.5 Parameterizable (Soft) IP Cores  187  
9.5.1 Identifying Design Components Suitable for Development as IP  189  
9.5.2 Identifying Parameters for IP Cores  190  
9.5.3 Development of Parameterizable Features  193  
9.5.4 Parameterizable Control Circuitry  194  
9.5.5 Application to Simple FIR Filter  194  
9.6 IP Core Integration  195  
9.6.1 Design Issues  196  
9.7 Current FPGA-based IP Cores  197  
9.8 Watermarking IP  198  
9.9 Summary  198  

## Bibliography  199  

## 10 Advanced Model-Based FPGA Accelerator Design  200

10.1 Introduction  200  
10.2 Dataflow Modeling of DSP Systems  201  
10.2.1 Process Networks  201  
10.2.2 Synchronous Dataflow  202  
10.2.3 Cyclo-static Dataflow  203  
10.2.4 Multidimensional Synchronous Dataflow  204  
10.3 Architectural Synthesis of Custom Circuit Accelerators from DFGs  204  
10.4 Model-Based Development of Multi-Channel Dataflow Accelerators  205  
10.4.1 Multidimensional Arrayed Dataflow  207  
10.4.2 Block and Interleaved Processing in MADF  209  
10.4.3 MADF Accelerators  209  
10.4.4 Pipelined FE Derivation for MADF Accelerators  210  
10.4.5 WBC Configuration  213  
10.4.6 Design Example: Normalized Lattice Filter  214  
10.4.7 Design Example: Fixed Beamformer System  216  
10.5 Model-Based Development for Memory-Intensive Accelerators  219  
10.5.1 Synchronous Dataflow Representation of FSME  219  
10.5.2 Cyclo-static Representation of FSME  221  
10.6 Summary  223  

## References  223  

## 11 Adaptive Beamformer Example  225

11.1 Introduction to Adaptive Beamforming  226  
11.2 Generic Design Process  226  
11.2.1 Adaptive Beamforming Specification  229  
11.2.2 Algorithm Development  230
# Contents

11.3 Algorithm to Architecture 231  
11.3.1 Dependence Graph 232  
11.3.2 Signal Flow Graph 233  
11.4 Efficient Architecture Design 235  
11.4.1 Scheduling the QR Operations 239  
11.5 Generic QR Architecture 240  
11.5.1 Processor Array 242  
11.6 Retiming the Generic Architecture 246  
11.6.1 Retiming QR Architectures 250  
11.7 Parameterizable QR Architecture 253  
11.7.1 Choice of Architecture 254  
11.7.2 Parameterizable Control 256  
11.7.3 Linear Architecture 256  
11.7.4 Sparse Linear Architecture 258  
11.7.5 Rectangular Architecture 262  
11.7.6 Sparse Rectangular Architecture 264  
11.7.7 Generic QR Cells 264  
11.8 Generic Control 266  
11.8.1 Generic Input Control for Linear and Sparse Linear Arrays 266  
11.8.2 Generic Input Control for Rectangular and Sparse Rectangular Arrays 267  
11.8.3 Effect of Latency on the Control Seeds 268  
11.9 Beamformer Design Example 269  
11.10 Summary 271  

References 271

12 FPGA Solutions for Big Data Applications 273  
12.1 Introduction 273  
12.2 Big Data 274  
12.3 Big Data Analytics 275  
12.3.1 Inductive Learning 276  
12.3.2 Data Mining Algorithms 277  
12.3.3 Classification 277  
12.3.4 Regression 278  
12.3.5 Clustering 279  
12.3.6 The Right Approach 279  
12.4 Acceleration 280  
12.4.1 Scaling Up or Scaling Out 280  
12.4.2 FPGA-based System Developments 281  
12.4.3 FPGA Implementations 281  
12.4.4 Heston Model Acceleration Using FPGA 282  
12.5 $k$-Means Clustering FPGA Implementation 283  
12.5.1 Computational Complexity Analysis of $k$-Means Algorithm 285  
12.6 FPGA-Based Soft Processors 286  
12.6.1 IPPro FPGA-Based Processor 287  
12.7 System Hardware 290  
12.7.1 Distance Calculation Block 291  
12.7.2 Comparison Block 292
Contents

12.7.3 Averaging 292
12.7.4 Optimizations 292
12.8 Conclusions 293

Bibliography 293

13 Low-Power FPGA Implementation 296
13.1 Introduction 296
13.2 Sources of Power Consumption 297
13.2.1 Dynamic Power Consumption 297
13.2.2 Static power consumption 299
13.3 FPGA Power Consumption 300
13.3.1 Clock Tree Isolation 302
13.4 Power Consumption Reduction Techniques 302
13.5 Dynamic Voltage Scaling in FPGAs 303
13.6 Reduction in Switched Capacitance 305
13.6.1 Data Reordering 305
13.6.2 Pipelining 306
13.6.3 Locality 311
13.6.4 Data Mapping 313
13.7 Final Comments 316

Bibliography 317

14 Conclusions 319
14.1 Introduction 319
14.2 Evolution in FPGA Design Approaches 320
14.3 Big Data and the Shift toward Computing 320
14.4 Programming Flow for FPGAs 321
14.5 Support for Floating-Point Arithmetic 322
14.6 Memory Architectures 322

Bibliography 323

Index 325