

---

# Contents

<b>Preface</b> .....	VII
<b>Preface to Second Edition</b> .....	XI
<b>1. Introduction</b> .....	1
1.1 Overview of Digital Signal Processing (DSP) .....	1
1.2 FPGA Technology .....	3
1.2.1 Classification by Granularity .....	3
1.2.2 Classification by Technology .....	5
1.2.3 Benchmark for FPLs .....	6
1.3 DSP Technology Requirements .....	10
1.3.1 FPGA and Programmable Signal Processors .....	11
1.4 Design Implementation .....	12
1.4.1 FPGA Structure .....	16
1.4.2 The Altera EPF10K70RC240-4 .....	19
1.4.3 Case Study: Frequency Synthesizer .....	22
Exercises .....	27
<b>2. Computer Arithmetic</b> .....	31
2.1 Introduction .....	31
2.2 Number Representation .....	32
2.2.1 Fixed-Point Numbers .....	32
2.2.2 Unconventional Fixed-Point Numbers .....	35
2.2.3 Floating-Point Numbers .....	47
2.3 Binary Adders .....	50
2.3.1 Pipelined Adders .....	52
2.3.2 Modulo Adders .....	57
2.4 Binary Multipliers .....	58
2.4.1 Multiplier Blocks .....	62
2.5 Binary Dividers .....	63
2.5.1 Linear Convergence Division Algorithms .....	66
2.5.2 Fast Divider Design .....	71
2.5.3 Array Divider .....	76
2.6 Floating-point Arithmetic Implementation .....	76

2.6.1	Fixed-point to Floating-point Format Conversion . . . . .	77
2.6.2	Floating-point to Fixed-point Format Conversion . . . . .	79
2.6.3	Floating-point Multiplication . . . . .	80
2.6.4	Floating-point Addition . . . . .	81
2.6.5	Floating-point Division . . . . .	83
2.6.6	Floating-point Reciprocal . . . . .	85
2.6.7	Floating-point Synthesis Results . . . . .	86
2.7	Multiply-Accumulator (MAC) and Sum of Product (SOP) . . .	87
2.7.1	Distributed Arithmetic Fundamentals . . . . .	88
2.7.2	Signed DA Systems . . . . .	91
2.7.3	Modified DA Solutions . . . . .	92
2.8	Computation of Special Functions Using CORDIC . . . . .	94
2.8.1	CORDIC Architectures . . . . .	98
	Exercises . . . . .	103
<b>3.</b>	<b>Finite Impulse Response (FIR) Digital Filters . . . . .</b>	<b>109</b>
3.1	Digital Filters . . . . .	109
3.2	FIR Theory . . . . .	110
3.2.1	FIR Filter with Transposed Structure . . . . .	111
3.2.2	Symmetry in FIR Filters . . . . .	114
3.2.3	Linear-phase FIR Filters . . . . .	115
3.3	Designing FIR Filters . . . . .	116
3.3.1	Direct Window Design Method . . . . .	117
3.3.2	Equiripple Design Method . . . . .	119
3.4	Constant Coefficient FIR Design . . . . .	121
3.4.1	Direct FIR Design . . . . .	122
3.4.2	FIR Filter with Transposed Structure . . . . .	126
3.4.3	FIR Filter Using Distributed Arithmetic . . . . .	128
	Exercises . . . . .	143
<b>4.</b>	<b>Infinite Impulse Response (IIR) Digital Filters . . . . .</b>	<b>147</b>
4.1	IIR Theory . . . . .	150
4.2	IIR Coefficient Computation . . . . .	153
4.2.1	Summary of Important IIR Design Attributes . . . . .	155
4.3	IIR Filter Implementation . . . . .	156
4.3.1	Finite Wordlength Effects . . . . .	160
4.3.2	Optimization of the Filter Gain Factor . . . . .	161
4.4	Fast IIR Filter . . . . .	162
4.4.1	Time-domain Interleaving . . . . .	163
4.4.2	Clustered and Scattered Look-Ahead Pipelining . . . . .	165
4.4.3	IIR Decimator Design . . . . .	168
4.4.4	Parallel Processing . . . . .	168
4.4.5	IIR Design Using RNS . . . . .	171
	Exercises . . . . .	172

<b>5. Multirate Signal Processing</b>	175
5.1 Decimation and Interpolation	175
5.1.1 Noble Identities	176
5.1.2 Sampling Rate Conversion by Rational Factor	178
5.2 Polyphase Decomposition	179
5.2.1 Recursive IIR Decimator	183
5.2.2 Fast-running FIR Filter	184
5.3 Hogenauer CIC Filters	187
5.3.1 Single-Stage CIC Case Study	187
5.3.2 Multistage CIC Filter Theory	189
5.3.3 Amplitude and Aliasing Distortion	194
5.3.4 Hogenauer Pruning Theory	196
5.3.5 CIC RNS Design	201
5.4 Multistage Decimator	203
5.4.1 Multistage Decimator Design Using Goodman–Carey Half-band Filters	204
5.5 Frequency-Sampling Filters as Bandpass Decimators	206
5.6 Filter Banks	210
5.6.1 Uniform DFT Filter Bank	210
5.6.2 Two-channel Filter Banks	215
5.7 Wavelets	230
5.7.1 The Discrete Wavelet Transformation	233
Exercises	237
<b>6. Fourier Transforms</b>	241
6.1 The Discrete Fourier Transform Algorithms	242
6.1.1 Fourier Transform Approximations Using the DFT	242
6.1.2 Properties of the DFT	244
6.1.3 The Goertzel Algorithm	247
6.1.4 The Bluestein Chirp- $z$ Transform	248
6.1.5 The Rader Algorithm	251
6.1.6 The Winograd DFT Algorithm	257
6.2 The Fast Fourier Transform (FFT) Algorithms	259
6.2.1 The Cooley–Tukey FFT Algorithm	260
6.2.2 The Good–Thomas FFT Algorithm	270
6.2.3 The Winograd FFT Algorithm	273
6.2.4 Comparison of DFT and FFT Algorithms	276
6.3 Fourier-related Transforms	278
6.3.1 Computing the DCT Using the DFT	280
6.3.2 Fast Direct DCT Implementation	281
Exercises	283

<b>7. Advanced Topics</b>	289
7.1 Rectangular and Number Theoretic Transforms (NTTs)	289
7.1.1 Arithmetic Modulo $2^b \pm 1$	291
7.1.2 Efficient Convolutions Using NTTs	293
7.1.3 Fast Convolution Using NTTs	294
7.1.4 Multidimensional Index Maps and the Agarwal–Burrus NTT	297
7.1.5 Computing the DFT Matrix with NTTs	299
7.1.6 Index Maps for NTTs	301
7.1.7 Using Rectangular Transforms to Compute the DFT	305
7.2 Error Control and Cryptography	306
7.2.1 Basic Concepts from Coding Theory	307
7.2.2 Block Codes	312
7.2.3 Convolutional Codes	316
7.2.4 Cryptography Algorithms for FPGAs	324
7.3 Modulation and Demodulation	341
7.3.1 Basic Modulation Concepts	341
7.3.2 Incoherent Demodulation	345
7.3.3 Coherent Demodulation	351
Exercises	360
<b>8. Adaptive Filters</b>	365
8.1 Application of Adaptive Filter	366
8.1.1 Interference Cancellation	366
8.1.2 Prediction	367
8.1.3 Inverse Modeling	367
8.1.4 Identification	368
8.2 Optimum Estimation Techniques	368
8.2.1 The Optimum Wiener Estimation	370
8.3 The Widrow–Hoff Least Mean Square Algorithms	374
8.3.1 Learning Curves	381
8.3.2 Normalized LMS (NLMS)	383
8.4 Transform Domain LMS Algorithms	386
8.4.1 Fast-convolution Techniques	386
8.4.2 Using Orthogonal Transforms	387
8.5 Implementation of the LMS Algorithm	391
8.5.1 Quantization Effects	391
8.5.2 FPGA Design of the LMS Algorithm	392
8.5.3 Pipelined LMS Filters	395
8.5.4 Transposed Form LMS Filter	398
8.5.5 Design of DLMS Algorithms	399
8.5.6 LMS Designs using SIGNUM Function	402
8.6 Recursive Least Square Algorithms	405
8.6.1 RLS with Finite Memory	409
8.6.2 Fast RLS Kalman Implementation	411

8.6.3 The Fast a Posteriori Kalman RLS Algorithm . . . . .	416
8.7 Comparison of LMS and RLS Parameters . . . . .	417
Exercises . . . . .	419
<b>References</b> . . . . .	423
<b>A. Verilog Source Code</b> . . . . .	435
<b>B. VHDL and Verilog Coding</b> . . . . .	487
B.1 List of Examples . . . . .	489
B.2 Library of Parameterized Modules (LPM) . . . . .	490
B.2.1 The Parameterized Flip-flop Megafunction (lpm_ff) . . .	490
B.2.2 The Parameterized Adder/Subtractor Megafunction (lpm_add_sub) . . . . .	494
B.2.3 The Parameterized Multiplier Megafunction (lpm_mult) . . . . .	499
B.2.4 The Parameterized ROM Megafunction (lpm_rom) . . .	503
B.2.5 The Parameterized Divider Megafunction (lpm_divide) . . . . .	506
<b>C. Glossary</b> . . . . .	509
<b>D. CD-ROM File: “1readme.ps”</b> . . . . .	515
<b>Index</b> . . . . .	523