

Practical FPGA Programming in C

David Pellerin
Scott Thibault



Prentice Hall Professional Technical Reference

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Contents

Foreword by Clive "Max" Maxfield	xiii
Why is this book of interest to the hardware folks?	xiii
And what about the software guys and gals?	xv
So what's the catch?	xvi
Preface	xvii
C Language for FPGA-Based Hardware Design?	xviii
Compelling Platforms for Software Acceleration	xix
The Power to Experiment	xxi
How This Book Is Organized	xxi
Where This Book Came From	xxii
Acknowledgments	xxiii
CHAPTER 1	
The FPGA as a Computing Platform	1
1.1 A Quick Introduction to FPGAs	2
1.2 FPGA-Based Programmable Hardware Platforms	4
1.3 Increasing Performance While Lowering Costs	6
1.4 The Role of Tools	8
1.5 The FPGA as an Embedded Software Platform	10
1.6 The Importance of a Programming Abstraction	12
1.7 When Is C Language Appropriate for FPGA Design?	13
1.8 How to Use This Book	15

CHAPTER 2**A Brief History of Programmable Platforms 17**

2.1	The Origins of Programmable Logic	18
2.2	Reprogrammability, HDLs, and the Rise of the FPGA	23
2.3	Systems on a Programmable Chip	25
2.4	FPGAs for Parallel Computing	27
2.5	Summary	29

CHAPTER 3**A Programming Model for FPGA-Based Applications 31**

3.1	Parallel Processing Models	32
3.2	FPGAs as Parallel Computing Machines	35
3.3	Programming for Parallelism	38
3.4	Communicating Process Programming Models	39
3.5	The Impulse C Programming Model	41
3.6	Summary	43

CHAPTER 4**An Introduction to Impulse C 45**

4.1	The Motivation Behind Impulse C	47
4.2	The Impulse C Programming Model	48
4.3	A Minimal Impulse C Program	50
4.4	Processes, Streams, Signals, and Memory	58
4.5	Impulse C Signed and Unsigned Datatypes	59
4.6	Understanding Processes	59
4.7	Understanding Streams	63
4.8	Using Output Streams	66
4.9	Using Input Streams	67
4.10	Avoiding Stream Deadlocks	69
4.11	Creating and Using Signals	73
4.12	Understanding Registers	74
4.13	Using Shared Memories	76
4.14	Memory and Stream Performance Considerations	81
4.15	Summary	86

CHAPTER 5**Describing a FIR Filter 87**

5.1	Design Overview	87
5.2	The FIR Filter Hardware Process	88
5.3	The Software Test Bench	90
5.4	Desktop Simulation	97
5.5	Application Monitoring	98
5.6	Summary	100

CHAPTER 6**Generating FPGA Hardware 103**

6.1	The Hardware Generation Flow	104
6.2	Understanding the Generated Structure	108
6.3	Stream and Signal Interfaces	112
6.4	Using HDL Simulation to Understand Stream Protocols	116
6.5	Debugging the Generated Hardware	119
6.6	Hardware Generation Notes	125
6.7	Making Efficient Use of the Optimizers	127
6.8	Language Constraints for Hardware Processes	129
6.9	Summary	131

CHAPTER 7**Increasing Statement-Level Parallelism 133**

7.1	A Model of FPGA Computation	133
7.2	C Language Semantics and Parallelism	135
7.3	Exploiting Instruction-Level Parallelism	135
7.4	Limiting Instruction Stages	139
7.5	Unrolling Loops	141
7.6	Pipelining Explained	142
7.7	Summary	145

CHAPTER 8**Porting a Legacy Application to Impulse C 147**

8.1	The Triple-DES Algorithm	148
8.2	Converting the Algorithm to a Streaming Model	150
8.3	Performing Software Simulation	155
8.4	Compiling to Hardware	156
8.5	Preliminary Hardware Analysis	159
8.6	Summary	160

CHAPTER 9**Creating an Embedded Test Bench 163**

9.1	A Mixed Hardware and Software Approach	164
9.2	The Embedded Processor as a Test Generator	165
9.3	The Role of Hardware Simulators	168
9.4	Testing the Triple-DES Algorithm in Hardware	168
9.5	Software Stream Macro Interfaces	174
9.6	Building the Test System	175
9.7	Summary	194

CHAPTER 10**Optimizing C for FPGA Performance 195**

10.1 Rethinking an Algorithm for Performance	196
10.2 Refinement 1: Reducing Size by Introducing a Loop	199
10.3 Refinement 2: Array Splitting	199
10.4 Refinement 3: Improving Streaming Performance	201
10.5 Refinement 4: Loop Unrolling	203
10.6 Refinement 5: Pipelining the Main Loop	204
10.7 Summary	208

CHAPTER 11**Describing System-Level Parallelism 209**

11.1 Design Overview	210
11.2 Performing Desktop Simulation	213
11.3 Refinement 1: Creating Parallel 8-Bit Filters	214
11.4 Refinement 2: Creating a System-Level Pipeline	219
11.5 Moving the Application to Hardware	231
11.6 Summary	256

CHAPTER 12**Combining Impulse C with an Embedded Operating System 257**

12.1 The uClinux Operating System	257
12.2 A uClinux Demonstration Project	259
12.3 Summary	277

CHAPTER 13**Mandelbrot Image Generation 279**

13.1 Design Overview	280
13.2 Expressing the Algorithm in C	282
13.3 Creating a Fixed-Point Equivalent	285
13.4 Creating a Streaming Version	286
13.5 Parallelizing the Algorithm	290
13.6 Future Refinements	297
13.7 Summary	299

CHAPTER 14**The Future of FPGA Computing 301**

14.1 The FPGA as a High-Performance Computer	302
14.2 The Future of FPGA Computing	305
14.3 Summary	307

APPENDIX A**Getting the Most Out of Embedded FPGA Processors 309**

- A.1 FPGA Embedded Processor Overview 310
- A.2 Peripherals and Memory Controllers 312
- A.3 Increasing Processor Performance 313
- A.4 Optimization Techniques That Are Not FPGA-Specific 314
- A.5 FPGA-Specific Optimization Techniques 319
- A.6 Summary 322

APPENDIX B**Creating a Custom Stream Interface 325**

- B.1 Application Overview 326
- B.2 The DS92LV16 Serial Link for Data Streaming 327
- B.3 Stream Interface State Machine Description 329
- B.4 Data Transmission 331
- B.5 Summary 332

APPENDIX C**Impulse C Function Reference 341****APPENDIX D****Triple-DES Source Listings 375****APPENDIX E****Image Filter Listings 405****APPENDIX F****Selected References 417****Index 419**