

Mastering FPGA Chip Design

For Speed, Area, Power,
and Reliability



Academy Pro Title by
Kevin Hubbard

Mastering FPGA Chip Design

For Speed, Area, Power, and Reliability



Kevin Hubbard

● This is an Elektor Publication. Elektor is the media brand of
Elektor International Media B.V.
PO Box 11, NL-6114-ZG Susteren, The Netherlands
Phone: +31 46 4389444

● All rights reserved. No part of this book may be reproduced in any material form, including photocopying, or storing in any medium by electronic means and whether or not transiently or incidentally to some other use of this publication, without the written permission of the copyright holder except in accordance with the provisions of the Copyright Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency Ltd., 90 Tottenham Court Road, London, England W1P 9HE. Applications for the copyright holder's permission to reproduce any part of the publication should be addressed to the publishers.

● Declaration

The authors and publisher have used their best efforts in ensuring the correctness of the information contained in this book. They do not assume, and hereby disclaim, any liability to any party for any loss or damage caused by errors or omissions in this book, whether such errors or omissions result from negligence, accident or any other cause.

● **ISBN 978-3-89576-687-9** Print
ISBN 978-3-89576-686-2 eBook

● © Copyright 2024 Elektor International Media
www.elektor.com
Layout: Kevin Hubbard
Printers: Ipskamp, Enschede, The Netherlands

Elektor is the world's leading source of essential technical information and electronics products for pro engineers, electronics designers, and the companies seeking to engage them. Each day, our international team develops and delivers high-quality content - via a variety of media channels (including magazines, video, digital media, and social media) in several languages - relating to electronics design and DIY electronics. **www.elektormagazine.com**

Table of Contents

Preface	1
1. Digital Logic Design	3
1.1. What is Digital Logic Design?	3
1.2. CMOS Transistor Theory 101	3
1.3. D-Type Flip-Flop	8
1.4. Chapter Quiz	10
2. History of the Computer Chip	11
2.1. First Integrated Circuits	11
2.2. Early CMOS Integrated Circuits	11
2.3. First Integrated Circuit Microprocessor	12
2.4. Full Custom Computer Chip	12
2.5. Gate Array	13
2.6. Field Programmable Gate Array (FPGA)	15
2.7. Chapter Quiz	18
3. EDA Tool Flow	19
3.1. What is EDA?	19
3.2. ASIC versus FPGA EDA Design Flow	19
3.3. EDA Definitions	22
3.4. Chapter Quiz	26
4. Structural Verilog (+VHDL)	27
4.1. Schematic Design	27
4.2. Hardware Description Languages (HDLs)	28
4.3. Chapter Quiz	34
5. RTL Verilog (+VHDL)	35
5.1. Register Transfer Level (RTL)	35
5.2. Verilog versus VHDL	38
5.3. Vivado GUI Schematic Viewer	39
5.4. Chapter Quiz	40
6. FPGA Look Up Tables	41
6.1. Chapter Quiz	46
7. Verilog (+VHDL) Design Hierarchy	47
7.1. Design Hierarchy	47
7.2. ChipVault IDE	54
7.3. Chapter Quiz	58
8. Finite State Machines	59

8.1. Hardware State Machines	59
8.2. Software State Machines	66
8.3. Chapter Quiz	68
9. IO Banks	69
9.1. Single-Ended Signaling	71
9.2. Differential Signaling	73
9.3. SERDES Transceivers	79
9.4. Optical Fiber	81
9.5. Chapter Quiz	82
10. RAMs and ROMs	83
10.1. Block RAM (BRAM)	84
10.2. Using the RAMB36 in an Artix7 FPGA.	86
10.3. Block ROM	92
10.4. Chapter Quiz	94
11. FIFOs	95
11.1. Chapter Quiz	100
12. Clocking	101
12.1. Chapter Quiz	110
13. Timing Closure	111
13.1. FSM Timing Closure	118
13.2. Attack Timing Arcs from both ends.	120
13.3. Register Retiming	122
13.4. Chapter Quiz	124
14. Placement and Floorplanning	125
14.1. Placement	126
14.2. Floorplanning	128
14.3. Chapter Quiz	134
15. Digital Logic Simulation	135
15.1. My First Digital Logic Designs	135
15.2. What is an RTL Simulator?	136
15.3. ModelSim	138
15.4. VivadoSim	141
15.5. IcarusVerilog	146
15.6. Verilator	148
15.7. Chapter Quiz	152
16. Test Benches	153
16.1. Definition	153

16.2. Stimulus only	153
16.3. Test Vector Stimulus and Result Capture	158
16.4. Stimulus with Output Analysis	163
16.5. Chapter Quiz	166
17. Python Software	167
17.1. Software for Hardware Design	167
17.2. Hello World	168
17.3. Python Strings	168
17.4. File Filter	171
17.5. Cheats	173
17.6. Chapter Quiz	178
18. Digital Arithmetic	179
18.1. Unsigned Binary	179
18.2. Signed Magnitude	180
18.3. Two's Complement	181
18.4. Fixed-Point	182
18.5. Floating-Point	184
18.6. Overflow Detection and/or Saturation	185
18.7. Rounding	186
18.8. Pitfalls of working with two's complement	188
18.9. Chapter Quiz	190
19. Example Design - Blinky	191
19.1. Chapter Quiz	198
20. Design for Speed	199
20.1. Pipelined Design	199
20.2. Simpler is Faster	199
20.3. Don't Go Overboard on Pipelining	200
20.4. Serialize Buses	200
20.5. Reduce the number of inputs to FSM states	200
20.6. Stay below 90% resource utilization	201
20.7. Partitioning into Fast and Slow Clock Domains	201
20.8. Be Cognizant of Physical RAM Implementations	201
20.9. Use Clock Prescalers for Large Time Counters	202
20.10. Posted-Write Buses	203
20.11. Don't Accidentally Force a Difficult Placement	203
20.12. Chapter Quiz	204
21. Example Design - UART Interfacing	205

21.1. Why UART Serial Communications	205
21.2. RS-232	206
21.3. Lesson-1 : Simple Loopback Design	208
21.4. Lesson-2 : UART RX Design	210
21.5. Lesson-3 : UART TX Design	214
21.6. Chapter Quiz	218
22. Design for Area	219
22.1. Even Free Gates Come at a High Price	219
22.2. Keep Track of Build Utilization Over Time	220
22.3. Use SRLs whenever possible	220
22.4. Minimize Saturation and Rounding Operations	221
22.5. Minimize RAM size	222
22.6. Minimize Register Widths	222
22.7. Minimize Bus Address Decoding	223
22.8. Shadow RAMs for Register Readback with Reset Clear Counters	223
22.9. Minimize Number of Clock Trees	226
23. Example Design - Mesa Bus	229
23.1. Mesa Bus Protocol	229
23.2. Local Bus over Mesa Bus	230
23.3. Building an FPGA with Mesa Bus	230
23.4. bd_server.py device driver to Mesa Bus	232
23.5. bd_shell.py Command Line Interface shell to Mesa Bus	235
23.6. mesa_bus_ex.py Example Python Client	236
23.7. Chapter Quiz	238
24. Design for Power	239
24.1. CV ² F	239
24.2. Reduce Capacitance	239
24.3. Clock Frequency	240
24.4. Regional Clock Gating	240
24.5. Global Clock Gating	241
24.6. Powering Down	241
24.7. Parking Counters	241
24.8. Parking Data Buses	241
24.9. Reset	241
24.10. Chapter Quiz	242
25. Example Design - SUMP3 Logic Analyzer	243
25.1. Traditional ILAs	244

25.2. SUMPn-RLE Project History	244
25.3. SUMP3-RLE Features	246
25.4. SUMP3-RLE Example Hardware Design	248
25.5. SUMP3-RLE Software Tutorial	254
25.6. Chapter Quiz	272
26. Design for Reliability	273
26.1. Toss Bad Designs out the Window	273
26.2. Fixed Input / Output Timing	274
26.3. Clock Jitter	274
26.4. Power Rails	274
26.5. Adding Margin	275
26.6. Make Timing Closure	275
26.7. Resets	275
26.8. FSMs	277
26.9. Diagnostic Buses	277
26.10. Learn from Your Mistakes and Don't Repeat Them	278
26.11. Register Packing	278
26.12. Thread Safety	279
26.13. Stuck Flops	280
26.14. Clock Domain Crossing	281
26.15. Check for invalid input parameters	282
26.16. Detect the Unexpected	283
26.17. Simulate the Unexpected	283
26.18. Simulate the Timing Sweep	283
26.19. Faster than Fastest	283
26.20. Two's Complement	284
26.21. Overflow Detection and Saturation, Interrupts	284
26.22. Simulate the Data Sweep	285
26.23. CRC32 Checking	285
26.24. Power-On Self-Test (POST)	285
26.25. Chapter Quiz	286
27. Example Design - Video Graphics Controller	287
27.1. Cathode Ray Tube (CRT)	287
27.2. VGA (Video Graphics Array)	289
27.3. Digilent BASYS3 board	290
27.4. vga_timing.v	291
27.5. Video "Hello World" Test Pattern	294

27.6. Video Color Test Pattern	297
27.7. Video Frame Buffer	299
27.8. Video Text Buffer	302
27.9. Video Sprite Controller	307
27.10. Chapter Quiz	314
28. Design for Style	315
28.1. Long Hand Verilog	316
28.2. Net Naming Conventions	317
28.3. Source Code Width	317
28.4. Source Code Indentation	317
28.5. Avoid Magic Numbers	318
28.6. Comments	318
28.7. ASCII Art	319
28.8. Refactoring	319
28.9. Chapter Quiz	320
Appendix A: Verilog vs VHDL	321
Operator Comparison	321
RTL Source Comparison	323
Appendix B: Write Your Own Book Using AsciiDoc	329
Index	331