

RAPID PROTOTYPING OF DIGITAL SYSTEMS SOPC EDITION

Table of Contents

1	<i>Tutorial I: The 15 Minute Design</i>	2
1.1	Design Entry using the Graphic Editor	9
1.2	Compiling the Design	16
1.3	Simulation of the Design	17
1.4	Testing Your Design on an FPGA Board	18
1.5	Downloading Your Design to the DE1 Board	19
1.6	Downloading Your Design to the DE2 Board	22
1.7	Downloading Your Design to the UP3 Board	25
1.8	Downloading Your Design to the UP2 or UP1 Board	27
1.9	The 10 Minute VHDL Entry Tutorial	29
1.10	Compiling the VHDL Design	32
1.11	The 10 Minute Verilog Entry Tutorial	34
1.12	Compiling the Verilog Design	36
1.13	Timing Analysis	38
1.14	The Floorplan Editor	39
1.15	Symbols and Hierarchy	40
1.16	Functional Simulation	41
1.17	Laboratory Exercises	42
2	<i>FPGA Development Board Hardware and I/O Features</i>	46
2.1	FPGA and External Hardware Features	47
2.2	The FPGA Board's Memory Features	48
2.3	The FPGA Board's I/O Features	49
2.4	Obtaining an FPGA Development Board and Cables	53
3	<i>Programmable Logic Technology</i>	56
3.1	CPLDs and FPGAs	59
3.2	Altera MAX 7000S Architecture – A Product Term CPLD Device	60

3.3	Altera Cyclone Architecture – A Look-Up Table FPGA Device	62
3.4	Xilinx 4000 Architecture – A Look-Up Table FPGA Device	65
3.5	Computer Aided Design Tools for Programmable Logic	67
3.6	Next Generation FPGA CAD tools	68
3.7	Applications of FPGAs	69
3.8	Features of New Generation FPGAs	69
3.9	For additional information	70
3.10	Laboratory Exercises	71
4	<i>Tutorial II: Sequential Design and Hierarchy</i>	74
4.1	Install the Tutorial Files and FPGAcure Library for your board	74
4.2	Open the tutor2 Schematic	75
4.3	Browse the Hierarchy	76
4.4	Using Buses in a Schematic	78
4.5	Testing the Pushbutton Counter and Displays	79
4.6	Testing the Initial Design on the Board	80
4.7	Fixing the Switch Contact Bounce Problem	81
4.8	Testing the Modified Design on the FPGA Board	82
4.9	Laboratory Exercises	83
5	<i>FPGAcure Library Functions</i>	88
5.1	FPGAcure LCD_Display: LCD Panel Character Display	90
5.2	FPGAcure DEC_7SEG: Hex to Seven-segment Decoder	92
5.3	FPGAcure Debounce: Pushbutton Debounce	94
5.4	FPGAcure OnePulse: Pushbutton Single Pulse	95
5.5	FPGAcure Clk_Div: Clock Divider	96
5.6	FPGAcure VGA_Sync: VGA Video Sync Generation	97
5.7	FPGAcure Char_ROM: Character Generation ROM	99
5.8	FPGAcure Keyboard: Read Keyboard Scan Code	100
5.9	FPGAcure Mouse: Mouse Cursor	102
5.10	For additional information	103
6	<i>Using VHDL for Synthesis of Digital Hardware</i>	106
6.1	VHDL Data Types	106
6.2	VHDL Operators	107
6.3	VHDL Based Synthesis of Digital Hardware	108
6.4	VHDL Synthesis Models of Gate Networks	108

6.5	VHDL Synthesis Model of a Seven-segment LED Decoder	109
6.6	VHDL Synthesis Model of a Multiplexer	111
6.7	VHDL Synthesis Model of Tri-State Output	112
6.8	VHDL Synthesis Models of Flip-flops and Registers	112
6.9	Accidental Synthesis of Inferred Latches	114
6.10	VHDL Synthesis Model of a Counter	114
6.11	VHDL Synthesis Model of a State Machine	115
6.12	VHDL Synthesis Model of an ALU with an Adder/Subtractor and a Shifter	117
6.13	VHDL Synthesis of Multiply and Divide Hardware	118
6.14	VHDL Synthesis Models for Memory	119
6.15	Hierarchy in VHDL Synthesis Models	123
6.16	Using a Testbench for Verification	125
6.17	For additional information	126
6.18	Laboratory Exercises	126
7	<i>Using Verilog for Synthesis of Digital Hardware</i>	130
7.1	Verilog Data Types	130
7.2	Verilog Based Synthesis of Digital Hardware	130
7.3	Verilog Operators	131
7.4	Verilog Synthesis Models of Gate Networks	132
7.5	Verilog Synthesis Model of a Seven-segment LED Decoder	132
7.6	Verilog Synthesis Model of a Multiplexer	133
7.7	Verilog Synthesis Model of Tri-State Output	134
7.8	Verilog Synthesis Models of Flip-flops and Registers	135
7.9	Accidental Synthesis of Inferred Latches	136
7.10	Verilog Synthesis Model of a Counter	136
7.11	Verilog Synthesis Model of a State Machine	137
7.12	Verilog Synthesis Model of an ALU with an Adder/Subtractor and a Shifter	138
7.13	Verilog Synthesis of Multiply and Divide Hardware	139
7.14	Verilog Synthesis Models for Memory	140
7.15	Hierarchy in Verilog Synthesis Models	143
7.16	For additional information	144
7.17	Laboratory Exercises	144
8	<i>State Machine Design: The Electric Train Controller</i>	148
8.1	The Train Control Problem	148

8.2	Train Direction Outputs (DA1-DA0, and DB1-DB0)	149
8.3	Switch Direction Outputs (SW1, SW2, and SW3)	150
8.4	Train Sensor Input Signals (S1, S2, S3, S4, and S5)	150
8.5	An Example Controller Design	151
8.6	VHDL Based Example Controller Design	154
8.7	Verilog Based Example Controller Design	157
8.8	Automatically Generating a State Diagram of a Design	160
8.9	Simulation Vector file for State Machine Simulation	161
8.10	Running the Train Control Simulation	162
8.11	Running the Video Train System (After Successful Simulation)	162
8.12	A Hardware Implementation of the Train System Layout	164
8.13	Laboratory Exercises	166
9	<i>A Simple Computer Design: The μP 3</i>	170
9.1	Computer Programs and Instructions	171
9.2	The Processor Fetch, Decode and Execute Cycle	172
9.3	VHDL Model of the μ P 3	179
9.4	Verilog Model of the μ P 3	182
9.5	Automatically Generating a State Diagram of the μ P3	186
9.6	Simulation of the μ P3 Computer	187
9.7	Laboratory Exercises	188
10	<i>VGA Video Display Generation using FPGAs</i>	192
10.1	Video Display Technology	192
10.2	Video Refresh	192
10.3	Using an FPGA for VGA Video Signal Generation	195
10.4	A VHDL Sync Generation Example: FPGAcore VGA_SYNC	196
10.5	Final Output Register for Video Signals	198
10.6	Required Pin Assignments for Video Output	198
10.7	Video Examples	199
10.8	A Character Based Video Design	200
10.9	Character Selection and Fonts	200
10.10	VHDL Character Display Design Examples	203
10.11	A Graphics Memory Design Example	206
10.12	Video Data Compression	207
10.13	Video Color Mixing using Dithering	207

10.14	VHDL Graphics Display Design Example	208
10.15	Higher Video Resolution and Faster Refresh Rates	209
10.16	Laboratory Exercises	210
11	<i>Interfacing to the PS/2 Keyboard and Mouse</i>	214
11.1	PS/2 Port Connections	214
11.2	Keyboard Scan Codes	215
11.3	Make and Break Codes	215
11.4	The PS/2 Serial Data Transmission Protocol	216
11.5	Scan Code Set 2 for the PS/2 Keyboard	218
11.6	The Keyboard FPGACore	220
11.7	A Design Example Using the Keyboard FPGACore	223
11.8	Interfacing to the PS/2 Mouse	224
11.9	The Mouse FPGACore	226
11.10	Mouse Initialization	226
11.11	Mouse Data Packet Processing	227
11.12	An Example Design Using the Mouse FPGACore	228
11.13	For Additional Information	229
11.14	Laboratory Exercises	229
12	<i>Legacy Digital I/O Interfacing Standards</i>	232
12.1	Parallel I/O Interface	232
12.2	RS-232C Serial I/O Interface	233
12.3	SPI Bus Interface	235
12.4	I ² C Bus Interface	237
12.5	For Additional Information	239
12.6	Laboratory Exercises	239
13	<i>FPGA Robotics Projects</i>	242
13.1	The FPGA-bot Design	242
13.2	FPGA-bot Servo Drive Motors	242
13.3	Modifying the Servos to make Drive Motors	243
13.4	VHDL Servo Driver Code for the FPGA-bot	244
13.5	Low-cost Sensors for an FPGA Robot Project	246
13.6	Assembly of the FPGA-bot Body	259
13.7	I/O Connections to the board's Expansion Headers	266
13.8	Robot Projects Based on R/C Toys, Models, and Robot Kits	267

13.9	For Additional Information	275
13.10	Laboratory Exercises	277
14	<i>A RISC Design: Synthesis of the MIPS Processor Core</i>	284
14.1	The MIPS Instruction Set and Processor	284
14.2	Using VHDL to Synthesize the MIPS Processor Core	287
14.3	The Top-Level Module	288
14.4	The Control Unit	291
14.5	The Instruction Fetch Stage	293
14.6	The Decode Stage	296
14.7	The Execute Stage	298
14.8	The Data Memory Stage	300
14.9	Simulation of the MIPS Design	301
14.10	MIPS Hardware Implementation on the FPGA Board	302
14.11	For Additional Information	303
14.12	Laboratory Exercises	304
15	<i>Introducing System-on-a-Programmable-Chip</i>	310
15.1	Processor Cores	310
15.2	SOPC Design Flow	311
15.3	Initializing Memory	313
15.4	SOPC Design versus Traditional Design Modalities	315
15.5	An Example SOPC Design	316
15.6	Hardware/Software Design Alternatives	317
15.7	For additional information	317
15.8	Laboratory Exercises	318
16	<i>Tutorial III: Nios II Processor Software Development</i>	322
16.1	Install the DE board files	322
16.2	Starting a Nios II Software Project	322
16.3	The Nios II IDE Software	324
16.4	Generating the Nios II System Library	325
16.5	Software Design with Nios II Peripherals	326
16.6	Starting Software Design – main()	329
16.7	Downloading the Nios II Hardware and Software Projects	330
16.8	Executing the Software	331
16.9	Starting Software Design for a Peripheral Test Program	331

16.10	Handling Interrupts	334
16.11	Accessing Parallel I/O Peripherals	335
16.12	Communicating with the LCD Display (<i>DE2 only</i>)	336
16.13	Testing SRAM	339
16.14	Testing Flash Memory	340
16.15	Testing SDRAM	341
16.16	Downloading the Nios II Hardware and Software Projects	346
16.17	Executing the Software	347
16.18	For additional information	347
16.19	Laboratory Exercises	348
17	<i>Tutorial IV: Nios II Processor Hardware Design</i>	352
17.1	Install the DE board files	352
17.2	Creating a New Project	352
17.3	Starting SOPC Builder	353
17.4	Adding a Nios II Processor	355
17.5	Adding UART Peripherals	358
17.6	Adding an Interval Timer Peripheral	359
17.7	Adding Parallel I/O Components	360
17.8	Adding an SRAM Memory Controller	361
17.9	Adding an SDRAM Memory Controller	362
17.10	Adding the LCD Module (<i>DE2 Board Only</i>)	362
17.11	Adding an External Bus	363
17.12	Adding Components to the External Bus	364
17.13	Global Processor Settings	364
17.14	Finalizing the Nios II Processor	365
17.15	Add the Processor Symbol to the Top-Level Schematic	366
17.16	Create a Phase-Locked Loop Component	367
17.17	Complete the Top-Level Schematic	368
17.18	Design Compilation	368
17.19	Testing the Nios II Project	369
17.20	For additional information	370
17.21	Laboratory Exercises	370
18	<i>Operating System Support for SOPC Design</i>	374
18.1	Nios II OS Support	376

18.2	eCos	377
18.3	μ C/OS-II	378
18.4	μ Clinux	379
18.5	Implementing the μ Clinux on the DE Board	380
18.6	Hardware Design for μ Clinux Support	380
18.7	Configuring the DE Board	382
18.8	Exploring μ Clinux on the DE Board	385
18.9	PS/2 Device Support in μ Clinux	386
18.10	Video Display in μ Clinux	386
18.11	USB Devices in μ Clinux (<i>DE2 Board Only</i>)	387
18.12	Network Communication in μ Clinux (<i>DE2 Board Only</i>)	387
18.13	For additional information	388
18.14	Laboratory Exercises	388
<i>Appendix A: Generation of Pseudo Random Binary Sequences</i>		<i>391</i>
<i>Appendix B: Quartus II Design and Data File Extensions</i>		<i>393</i>
<i>Appendix C: Common FPGA Pin Assignments</i>		<i>394</i>
<i>Appendix D: ASCII Character Code</i>		<i>396</i>
<i>Appendix E: Common I/O Connector Pin Assignments</i>		<i>397</i>
<i>Glossary</i>		<i>399</i>
<i>Index</i>		<i>407</i>
<i>About the Accompanying DVD</i>		<i>411</i>



<http://www.springer.com/978-0-387-72670-0>

Rapid Prototyping of Digital Systems

SOPC Edition

Hamblen, J.O.; Hall, T.S.; Furman, M.D.

2008, XVII, 411 p. With online files/update., Softcover

ISBN: 978-0-387-72670-0