

---

# Contents

---

List of Abbreviations	xvii
Preface	xix
Aid Sheet of Verilog HDL Design and Simulation	xxvii
<b>I Fundamentals of IC Design and Simulation with Verilog HDL</b>	<b>1</b>
<b>1 Introduction to IC Design</b>	<b>3</b>
1.1 History and Road Map of IC Design	3
1.1.1 A brief history of IC design	3
1.1.2 Today's IC industry and technology	4
1.1.3 Beyond Moore's law	6
1.1.4 Preview	8
1.2 Introduction to IC Industry	8
1.2.1 What Are FPGA and ASIC?	8
1.2.2 Brief introduction to ASIC and FPGA industry	9
1.2.3 EDA tools and vendors	10
1.2.4 Hardware design solutions	11
1.3 FPGA and ASIC Design	15
1.3.1 FPGA and ASIC implementations	15
1.3.2 Comparison of FPGA and ASIC	17
1.3.3 ASIC development: MPW vs. Fullset	19
1.3.4 FPGA development and applications	20
1.4 Fundamentals of Verilog HDL Design	20
1.4.1 Multi-level description with Verilog HDL	20
1.4.2 Fundamentals of combinational circuits	24
1.4.3 Fundamentals of sequential circuits	27
1.5 Attributes of Verilog HDL Design	30
1.5.1 Key attributes of hardware description	30
1.5.2 Concurrency	31
1.5.3 Timing diagram	32
1.5.4 Signal values	35

<b>2</b>	<b>IC Design Flow</b>	<b>38</b>
2.1	Design Specification	38
2.2	RTL Design in IP Level	40
2.2.1	IP-level design	40
2.2.2	Hardware design considerations	41
2.2.3	Code editors	43
2.3	RTL Verification in IP Level	43
2.3.1	Simulation testbench with Verilog HDL	44
2.3.2	Verification methods and methodologies	45
2.3.3	EDA tools for simulation	47
2.4	SoC Integration	48
2.5	SoC Verification	49
2.6	FPGA Netlist	50
2.6.1	FPGA development and assessment	50
2.6.2	MOF and timing constraints	50
2.6.3	EDA tools for FPGA demonstration	52
2.7	FPGA Verification	52
2.7.1	Importance of FPGA verification	52
2.7.2	Challenges of FPGA verification	53
2.8	ASIC Synthesis	54
2.8.1	ASIC synthesis and script file	54
2.8.2	Gate-level netlist and synthesis report	55
2.8.3	EDA tools for ASIC synthesis	57
2.9	Pre-simulation and static timing analysis	57
2.9.1	Pre-simulation vs. static timing analysis	57
2.9.2	Pre-simulation example	58
2.9.3	Static timing analysis and timing failures	59
2.10	Layout	60
2.10.1	What is layout?	60
2.10.2	Pad-limited and core-limited chip size	62
2.11	Post-Simulation and Post-STA	62
2.11.1	Post-simulation	63
2.11.2	Post-STA	63
2.12	DRC and LVS	64
2.13	Tape-Out	65
2.13.1	Chip tape-out procedure	65
2.13.2	MPW/shuttle and full-set	67
2.14	Chip Prob and Design for Test	68
2.14.1	Chip prob and tester	68
2.14.2	DFT and ATPG	68
2.15	Packaging and Testing	69
2.16	An Example of IC Design Flow	70
2.16.1	Front-end phase	70
2.16.2	Back-end phase	72
2.17	Basic Qualifications Needed For IC Design and Simulation	73

2.17.1	RTL design and simulation . . . . .	73
2.17.2	Functional verification (simulation) and FPGA verification . . . . .	74
2.17.3	Industry design rules and design standards . . . . .	75
2.17.4	Book introduction . . . . .	75
<b>3</b>	<b>Introduction to Verilog HDL</b>	<b>77</b>
3.1	Background of Verilog HDL . . . . .	77
3.2	An overview of Verilog HDL . . . . .	78
3.2.1	Starting with a Verilog design example . . . . .	78
3.2.2	Basic Verilog design syntax . . . . .	80
3.2.3	Recommended coding format . . . . .	80
3.3	Verilog Data Types . . . . .	82
3.3.1	Verilog constant . . . . .	82
3.3.2	Verilog data types: wire and reg . . . . .	83
3.3.3	Hardware results for declared data types . . . . .	83
3.3.4	Mismatched data width . . . . .	85
3.4	Basic Verilog operators . . . . .	86
3.4.1	Equality and inequality . . . . .	86
3.4.2	Bit-wise logic operators . . . . .	88
3.4.3	Reduction on multi-bit signals . . . . .	89
3.4.4	Relational operators . . . . .	93
3.4.5	Concatenation . . . . .	94
3.4.6	Logical shift . . . . .	97
3.4.7	Arithmetic operators . . . . .	98
3.5	Behavioral Verilog Blocks: initial, assign, and always . . . . .	98
3.5.1	Synthesizable block – assign . . . . .	99
3.5.2	Synthesizable block – always . . . . .	102
3.5.3	Unsynthesizable block – initial . . . . .	108
	Exercises . . . . .	108
	PBL 1: AND-OR Combinational Circuit . . . . .	111
	PBL 2: AND-MUX Combinational Circuit . . . . .	111
	PBL 3: AND-OR-Tri Combinational Circuit . . . . .	112
<b>4</b>	<b>RTL Design with Verilog HDL</b>	<b>114</b>
4.1	Design Statements: if-else, case, for/while/repeat/forever loop . . . . .	114
4.1.1	if-else statement . . . . .	114
4.1.2	case statement . . . . .	119
4.1.3	Loop statements: for, while, repeat, and forever . . . . .	123
4.2	Blocking and Non-Blocking . . . . .	127
4.2.1	Examples of blocking designs . . . . .	128
4.2.2	Examples of non-blocking designs . . . . .	130
4.3	Asynchronous and Synchronous Reset . . . . .	131
4.3.1	Verilog design examples of asynchronous and synchronous reset . . . . .	131

4.3.2	Simulation and synthesis of asynchronous and synchronous reset . . . . .	133
4.4	Hierarchical Design and Instantiation . . . . .	134
4.4.1	Verilog design example of hierarchical design . . . . .	134
4.4.2	Verilog testbench example of instantiation . . . . .	137
4.5	RTL Design Rules with Verilog HDL . . . . .	138
	Exercises . . . . .	140
	PBL 4: AND-OR-Latch Sequential Circuit . . . . .	143
	PBL 5: AND-OR-REG Sequential Circuit . . . . .	144
<b>5</b>	<b>Design Simulation with Verilog HDL</b>	<b>146</b>
5.1	System Tasks . . . . .	146
5.1.1	\$display, \$monitor, and \$fwrite . . . . .	146
5.1.2	\$time . . . . .	148
5.1.3	\$finish and \$stop . . . . .	148
5.1.4	\$dumpfile and \$dumpvars . . . . .	149
5.1.5	\$readmemh and \$readmemb . . . . .	149
5.1.6	\$random and \$urandom . . . . .	150
5.1.7	A testbench example using system tasks . . . . .	151
5.2	Compiler Directives . . . . .	155
5.2.1	`define vs. parameter . . . . .	155
5.2.2	`ifdef-`else/`elsif-`endif . . . . .	156
5.2.3	`include . . . . .	158
5.2.4	`timescale . . . . .	159
5.3	Functions and Tasks . . . . .	160
5.3.1	Functions . . . . .	161
5.3.2	Tasks . . . . .	165
5.4	Verilog Delay Control . . . . .	169
5.4.1	Delay expression . . . . .	170
5.4.2	Event expression . . . . .	170
5.5	Automated Simulation Environment and Verilog Testbench . . . . .	171
5.5.1	Structured project directory . . . . .	172
5.5.2	Automated simulation testbench utilizing BFM and monitor . . . . .	175
5.5.3	Verilog design on automated simulation testbench . . . . .	176
5.6	Guidelines for RTL Simulation and Verification . . . . .	181
	PBL 6: Four-Bit Full-Adder Design and Simulation . . . . .	183
	PBL 7: Combinational Circuit Design and Simulation . . . . .	184
	PBL 8: Sequential Circuit Design and Simulation . . . . .	184
	PBL 9: Sequential Circuit Instantiation and Simulation . . . . .	185
<b>6</b>	<b>Synthesis: Matching Verilog HDL with Basic Combinational and Sequential Circuits</b>	<b>186</b>
6.1	Introduction to Synthesis . . . . .	186
6.1.1	What is synthesis? . . . . .	186

6.1.2	Mismatches between simulation and synthesis . . . . .	187
6.1.3	Synthesizable Verilog HDL . . . . .	188
6.2	Synthesis of Combinational Logic . . . . .	190
6.2.1	Fundamental combinational logic . . . . .	190
6.2.2	Uni-directional and bi-directional buses . . . . .	192
6.2.3	Multiplexer . . . . .	192
6.3	Synthesis of sequential latches . . . . .	195
6.3.1	Intentional latches design with Verilog . . . . .	195
6.3.2	Accidental latches with Verilog . . . . .	197
6.4	Synthesis of Sequential Registers . . . . .	201
6.4.1	Single-bit register with asynchronous and synchronous reset . . . . .	201
6.4.2	Multi-bit register with asynchronous and synchronous reset . . . . .	202
6.4.3	Shift register . . . . .	203
6.5	Synthesis of Counter and Timer . . . . .	205
6.5.1	Counter 0–f . . . . .	205
6.5.2	Timer 0–f . . . . .	207
	Exercises . . . . .	211
	PBL 10: Four-Bit Full-Adder with Tri-State Buffer . . . . .	213
	PBL 11: Four-Bit Full-Adder with Latch . . . . .	214
	PBL 12: Four-Bit Full-Adder with Register . . . . .	214
	PBL 13: Counter 0–9 . . . . .	214
	PBL 14: Timer 0–9 . . . . .	216
	PBL 15: Timing Interface . . . . .	217
	PBL 16: Timing Controller . . . . .	218
<b>7</b>	<b>FSM Design</b> . . . . .	<b>221</b>
7.1	Modeling FSM . . . . .	221
7.1.1	Introduction to FSM . . . . .	221
7.1.2	FSM design template with Verilog HDL . . . . .	222
7.2	Mealy and Moore FSMs with Verilog HDL . . . . .	223
7.2.1	Introduction to Mealy and Moore FSMs . . . . .	223
7.2.2	Mealy machine . . . . .	224
7.2.3	Moore machine . . . . .	230
7.3	Design Example: Sequence Detector . . . . .	233
7.3.1	Introduction to sequence detector . . . . .	234
7.3.2	Mealy FSM: 0101 sequence detector . . . . .	235
7.3.3	Moore FSM: 0101 sequence detector . . . . .	239
7.4	Design Example: Odd/Even Number of 1s Checker . . . . .	244
7.4.1	Introduction to odd/even number of 1s checker . . . . .	244
7.4.2	State graph of odd/even 1s checker . . . . .	244
7.4.3	State and transition tables of odd/even 1s checker. . . . .	245
7.4.4	Verilog HDL design of odd/even 1s checker. . . . .	246

7.4.5	Simulation and synthesis analysis of odd/even 1s checker. . . . .	247
7.5	Design Example: Data Package Receiver . . . . .	248
7.5.1	Introduction to data package receiver . . . . .	248
7.5.2	State graph of data package receiver . . . . .	249
7.5.3	State and transition tables of data package receiver . . . . .	250
7.5.4	Verilog HDL design of data package receiver . . . . .	251
7.5.5	Simulation and synthesis analysis of data package receiver . . . . .	254
	Exercises . . . . .	255
	PBL 17: Sequence Detector . . . . .	256
<b>8</b>	<b>FSM-Datapath Design and Bus Communication</b>	<b>258</b>
8.1	FSM-Datapath and Bus Communication . . . . .	258
8.1.1	FSMD construction . . . . .	258
8.1.2	Bus communications . . . . .	259
8.2	Bus Communication Mechanisms . . . . .	260
8.2.1	Ready-valid protocol . . . . .	260
8.2.2	Enable-finish handshaking . . . . .	262
8.2.3	Request-grant arbitration. . . . .	263
8.3	Design Example: I2C Write . . . . .	264
8.3.1	I2C bus protocol . . . . .	264
8.3.2	An FSMD design example: I2C write operations . . . . .	269
8.3.3	Datapath design . . . . .	270
8.3.4	FSM design . . . . .	272
8.3.5	I2C Master design with verilog HDL . . . . .	273
8.4	Design Example: MSBUS Communication . . . . .	277
8.4.1	MSBUS Protocol . . . . .	277
8.4.2	An FSMD design example: MSBUS master and slave . . . . .	281
8.4.3	MSBUS master design with Verilog HDL . . . . .	283
8.4.4	MSBUS slave design with Verilog HDL . . . . .	287
8.4.5	MSBUS master-slave circuit analysis . . . . .	292
	Exercises . . . . .	293
	PBL 18: FP Add-Mul-Sub Design with FSMD . . . . .	295
	PBL 19: FP Add-Mul-Sub Design with Pipeline FP Operators . . . . .	298
	PBL 20: Pipeline FP Add-Mul-Sub Design . . . . .	298
	PBL 21: Parallel and Pipeline FP Add-Mul-Sub Design . . . . .	300
<b>II</b>	<b>Advanced IC Design and Integration</b>	<b>303</b>
<b>9</b>	<b>Numerical Hardware Design and Integration</b>	<b>305</b>
9.1	Design Template for Numerical Hardware . . . . .	305
9.2	Register Files . . . . .	307
9.2.1	RTL design on register files . . . . .	307
9.2.2	Single-port register file . . . . .	308

9.2.3	Dual-port register file . . . . .	313
9.2.4	Ping-pong buffer . . . . .	316
9.3	Design Example: FP Matrix-Matrix Adder . . . . .	319
9.3.1	Design structure of FP matrix-matrix adder . . . . .	319
9.3.2	Timing diagram of FP matrix-matrix adder . . . . .	321
9.3.3	Verilog code for FP matrix-matrix adder . . . . .	323
9.4	Design Example: FP AXPY Calculation . . . . .	326
9.4.1	Design structure of FP AXPY calculation . . . . .	326
9.4.2	Timing diagram of FP AXPY calculation . . . . .	327
9.4.3	Verilog code for FP AXPY calculation . . . . .	329
9.5	Design Example: Basic Design on FP DDOT . . . . .	332
9.5.1	Design structure of basic design on FP DDOT . . . . .	333
9.5.2	Timing diagram of basic design on FP DDOT calculation . . . . .	334
9.5.3	Verilog code for basic design on FP DDOT . . . . .	335
PBL 22:	Design Simulation on Single-Port Register File . . . . .	339
PBL 23:	Design Simulation on Dual-Port Register File . . . . .	339
PBL 24:	FP Matrix-Matrix Subtractor . . . . .	339
PBL 25:	FP Multiplication-Addition Circuit . . . . .	340
PBL 26:	FP Matrix-Matrix Multiplier . . . . .	341
PBL 27:	Power of FP Matrix . . . . .	341
<b>10</b>	<b>Streaming and Iterative Design on Numerical Hardware</b>	<b>343</b>
10.1	Streaming and Iterative Design and Integration . . . . .	343
10.1.1	Streaming and iterative designs . . . . .	343
10.1.2	Multi-controller design with pipelined and parallel framework . . . . .	344
10.2	Streaming Design on DDOT . . . . .	346
10.2.1	Design structure of streaming design on DDOT . . . . .	346
10.2.2	Timing diagram of streaming design on DDOT . . . . .	347
10.2.3	Timing controller of streaming design on DDOT . . . . .	348
10.2.4	Verilog code for streaming design on DDOT . . . . .	352
10.3	Iterative design with streaming width four . . . . .	354
10.3.1	Iterative design structure with streaming width four . . . . .	354
10.3.2	Timing diagram of iterative design with streaming width four . . . . .	355
10.3.3	Timing controller of iterative design with streaming width four . . . . .	356
10.3.4	Verilog code for iterative design with streaming width four . . . . .	358
10.4	Iterative Design with Streaming Width Two . . . . .	362
10.4.1	Iterative design structure with streaming width two . . . . .	362
10.4.2	Timing diagram of iterative design with streaming width two . . . . .	363

10.4.3	Timing controller of iterative design with streaming width two . . . . .	364
10.4.4	Verilog code for iterative design with streaming width two . . . . .	366
PBL 28:	Iterative Design on DDOT with Streaming Width One . . . . .	369
PBL 29:	Streaming Design on $DFT_2$ . . . . .	371
PBL 30:	Streaming Design on $DFT_4$ . . . . .	373
<b>11</b>	<b>Timing Constraints and High-Speed RTL Design</b>	<b>375</b>
11.1	Critical Path and True/False Path . . . . .	375
11.1.1	True/false path for AND/NAND and OR/NOR gates . . . . .	376
11.1.2	True/false path for combinational circuits. . . . .	376
11.1.3	Propagation/contamination delay and critical path. . . . .	379
11.2	Timing Information . . . . .	380
11.2.1	Timing constraints . . . . .	380
11.2.2	Propagation/contamination delay examples . . . . .	382
11.3	Maximum Delay Constraints . . . . .	384
11.3.1	Maximum delay constraint and setup time violation . . . . .	384
11.3.2	Maximum operational frequency . . . . .	385
11.3.3	Setup time slack . . . . .	385
11.3.4	Design Example #1: binary counter MOF . . . . .	386
11.3.5	Design example #2: sequential circuit MOF . . . . .	387
11.3.6	Design example #3: pipeline design MOF and latency . . . . .	389
11.4	Minimum Delay Constraints . . . . .	391
11.4.1	Minimum delay constraint and hold time violation . . . . .	391
11.4.2	Design example #1: hold timing analysis of binary counter . . . . .	392
11.4.3	Design example #2: hold timing constraint of sequential circuit . . . . .	392
11.5	Clock Skew . . . . .	393
11.5.1	Maximum delay constraints with clock skew . . . . .	393
11.5.2	Minimum delay constraints with clock skew . . . . .	394
11.5.3	Design Example #1: clock skew analysis of binary counter . . . . .	394
11.5.4	Design Example #2: clock skew analysis of sequential circuit . . . . .	395
11.6	Clock Domain Crossing . . . . .	395
11.6.1	Request-grant handshaking . . . . .	396
11.6.2	Design specification of request-grant handshaking crossing asynchronous clock domains . . . . .	397
11.6.3	Design on request-grant handshaking . . . . .	399
11.6.4	Synchronizer design . . . . .	401
11.6.5	System integration for request-grant handshaking and CDC . . . . .	403
	Exercises . . . . .	404



PBL 31: MOF Assessment of FP Numerical Hardware . . . . .	405
PBL 32: Request-Grant Handshaking and CDC . . . . .	406
PBL 33: Synchronous FIFO . . . . .	406
PBL 34: High-Speed Design on FP Matrix-Matrix Adder . . . . .	408
PBL 35: High-Speed Design on FP AXPY Calculation . . . . .	409
PBL 36: High-Speed Design on FP DDOT . . . . .	409
PBL 37: High-Speed Design on FP Matrix-Matrix Multiplier . . . . .	409
PBL 38: High-Speed Streaming Design on DDOT . . . . .	410
PBL 39: High-Speed Iterative Design with Streaming Width Four . . . . .	410
PBL 40: High-Speed Streaming Design on DFT2 . . . . .	410
PBL 41: High-Speed Streaming Design on DFT4 . . . . .	411
<b>12 SoC Design and Integration</b> . . . . .	<b>412</b>
12.1 SoC Bus Architecture . . . . .	412
12.1.1 AMBA AXI bus architecture . . . . .	413
12.1.2 AMBA AXI channels . . . . .	414
12.1.3 AMBA AXI bus valid-ready handshaking . . . . .	414
12.1.4 AMBA AXI write and read operations . . . . .	416
12.1.5 AMBA AXI burst length and burst size . . . . .	419
12.1.6 AMBA AXI burst types . . . . .	419
12.1.7 AMBA AXI address alignment . . . . .	423
12.2 Direct Memory Access (DMA) . . . . .	424
12.2.1 DMA design architecture . . . . .	425
12.2.2 DMA arbiter design . . . . .	426
12.2.3 DMA Command and address mapping . . . . .	428
12.3 RGB-to-Grayscale Converter . . . . .	431
12.3.1 Floating-point design . . . . .	431
12.3.2 Floating-to-integer conversion and approximate designs . . . . .	433
12.3.3 Hardware utilization vs. speed estimation . . . . .	437
12.4 Neural Networks . . . . .	438
12.4.1 Single-layer perceptron neural network . . . . .	438
12.4.2 Streaming design on SLP neural network . . . . .	439
12.4.3 Iterative design on SLP neural network . . . . .	442
12.4.4 Hardware utilization vs. latency estimation . . . . .	448
Exercises . . . . .	449
PBL 42: DMA Arbiter Design . . . . .	452
PBL 43: FP RGB-to-Grayscale Converter . . . . .	453
PBL 44: High-Speed Design on FP RGB-to-Grayscale Converter . . . . .	453
PBL 45: Integer Designs on RGB-to-Grayscale Converter . . . . .	453
PBL 46: Streaming Design on Sigmoid Neuron . . . . .	455
PBL 47: High-Speed Streaming Design on Sigmoid Neuron . . . . .	455
PBL 48: Iterative Design on Sigmoid Neuron with Streaming Width Eight . . . . .	456
PBL 49: Iterative Design on Sigmoid Neuron with Streaming Width Four . . . . .	457

PBL 50: High-Speed Iterative Designs on Sigmoid Neuron with Streaming Width Eight and Four . . . . .	457
<b>Bibliography</b>	<b>459</b>
<b>Index</b>	<b>461</b>

T & F --- Not for Distribution

---

## *List of Abbreviations*

---

**ABV** Assertion-Based Verification  
**AHB** Advanced High-Performance Bus  
**AMBA** Advanced Microcontroller Bus Architecture  
**APR** Automated Placement and Routing  
**ATPG** Auto Test Pattern Generation  
**ASIC** Application-Specific Integrated Circuit  
**AXI** Advanced eXtensible Interface  
**BFM** Bus Functional Model  
**CDC** Clock Domain Crossing  
**CDV** Coverage-Driven Verification  
**CLB** Configurable Logic Block  
**CMOS** Complementary Metal-Oxide-Semiconductor  
**CRV** Constrained-Random Verification  
**DFT** Design for Test  
**DRC** Design Rule Check  
**DUT** Design-Under-Test  
**DMA** Direct Memory Access  
**ECO** Engineering Change Order  
**EDA** Electronic Design Automation  
**FPGA** Field-Programmable Gate Array  
**GDS** Graphic Data System  
**HDL** Hardware Description Language  
**HCL** Hardware Construction Language

<b>HLS</b>	High Level Synthesis
<b>IC</b>	Integrated Circuit
<b>IDM</b>	Integrated Device Manufacturer
<b>IO</b>	Input and Output
<b>IP</b>	Intellectual Property
<b>LVS</b>	Layout Versus Schematic
<b>LUT</b>	Look-Up Table
<b>LSB/MSB</b>	Least-/Most-Significant Bit
<b>MPW</b>	Multi Project Wafer, or Shuttle
<b>MOF</b>	Maximum Operational Frequency
<b>MOSFET</b>	Metal-Oxide-Semiconductor Field-Effect Transistors
<b>NRE</b>	Non-Recurring Engineering
<b>OOP</b>	Object-Oriented Programming
<b>Pre/Post-Sim</b>	Pre-/Post-Layout Simulation
<b>Post-STA</b>	Post-Layout Static Timing Analysis
<b>PCB</b>	Printed Circuit Board
<b>RAM</b>	Random Access Memory
<b>ROM</b>	Read-Only Memory
<b>RTL</b>	Register-Transfer Level
<b>SDF</b>	Standard Delay Format
<b>SPF</b>	Standard Parasitic Format
<b>STA</b>	Static Timing Analysis
<b>UVM</b>	Universal Verification Methodology
<b>VMM</b>	Verification Methodology Manual
<b>WNS</b>	Worst Negative Slack