

Circuit Design with VHDL

Volnei A. Pedroni

MIT Press
Cambridge, Massachusetts
London, England

Contents

Preface	xi
I CIRCUIT DESIGN	1
1 Introduction	3
1.1 About VHDL	3
1.2 Design Flow	3
1.3 EDA Tools	4
1.4 Translation of VHDL Code into a Circuit	5
1.5 Design Examples	8
2 Code Structure	13
2.1 Fundamental VHDL Units	13
2.2 LIBRARY Declarations	13
2.3 ENTITY	15
2.4 ARCHITECTURE	17
2.5 Introductory Examples	17
2.6 Problems	22
3 Data Types	25
3.1 Pre-Defined Data Types	25
3.2 User-Defined Data Types	28
3.3 Subtypes	29
3.4 Arrays	30
3.5 Port Array	33
3.6 Records	35
3.7 Signed and Unsigned Data Types	35
3.8 Data Conversion	37
3.9 Summary	38
3.10 Additional Examples	38
3.11 Problems	43
4 Operators and Attributes	47
4.1 Operators	47
4.2 Attributes	50
4.3 User-Defined Attributes	52
4.4 Operator Overloading	53

4.5	GENERIC	54
4.6	Examples	55
4.7	Summary	60
4.8	Problems	61
5	Concurrent Code	65
5.1	Concurrent versus Sequential	65
5.2	Using Operators	67
5.3	WHEN (Simple and Selected)	69
5.4	GENERATE	78
5.5	BLOCK	81
5.6	Problems	84
6	Sequential Code	91
6.1	PROCESS	91
6.2	Signals and Variables	93
6.3	IF	94
6.4	WAIT	97
6.5	CASE	100
6.6	LOOP	105
6.7	CASE versus IF	112
6.8	CASE versus WHEN	113
6.9	Bad Clocking	114
6.10	Using Sequential Code to Design Combinational Circuits	118
6.11	Problems	121
7	Signals and Variables	129
7.1	CONSTANT	129
7.2	SIGNAL	130
7.3	VARIABLE	131
7.4	SIGNAL versus VARIABLE	133
7.5	Number of Registers	140
7.6	Problems	151
8	State Machines	159
8.1	Introduction	159
8.2	Design Style #1	160
8.3	Design Style #2 (Stored Output)	168

8.4	Encoding Style: From Binary to OneHot	181
8.5	Problems	183
9	Additional Circuit Designs	187
9.1	Barrel Shifter	187
9.2	Signed and Unsigned Comparators	191
9.3	Carry Ripple and Carry Look Ahead Adders	194
9.4	Fixed-Point Division	198
9.5	Vending-Machine Controller	202
9.6	Serial Data Receiver	208
9.7	Parallel-to-Serial Converter	211
9.8	Playing with a Seven-Segment Display	212
9.9	Signal Generators	217
9.10	Memory Design	220
9.11	Problems	225
II	SYSTEM DESIGN	231
10	Packages and Components	233
10.1	Introduction	233
10.2	PACKAGE	234
10.3	COMPONENT	236
10.4	PORT MAP	244
10.5	GENERIC MAP	244
10.6	Problems	251
11	Functions and Procedures	253
11.1	FUNCTION	253
11.2	Function Location	256
11.3	PROCEDURE	265
11.4	Procedure Location	266
11.5	FUNCTION versus PROCEDURE Summary	270
11.6	ASSERT	270
11.7	Problems	271
12	Additional System Designs	275
12.1	Serial-Parallel Multiplier	275
12.2	Parallel Multiplier	279

12.3	Multiply-Accumulate Circuits	285
12.4	Digital Filters	289
12.5	Neural Networks	294
12.6	Problems	301
Appendix A: Programmable Logic Devices		305
Appendix B: Xilinx ISE + ModelSim Tutorial		317
Appendix C: Altera MaxPlus II + Advanced Synthesis Software Tutorial		329
Appendix D: Altera Quartus II Tutorial		343
Appendix E: VHDL Reserved Words		355
	Bibliography	357
	Index	359

Index

- + Addition operator, 49, 54, 60, 195
- => Assignment operator for OTHERS, 47, 60
- <= Assignment operator for SIGNAL, 47, 60
- := Assignment operator for VARIABLE, CONSTANT, or GENERIC, 47, 60
- / Division operator, 49, 60
- = Equal-to operator, 49, 60
- ** Exponentiation operator, 49, 60
- > Greater-than operator, 49, 60
- >= Greater-than-or-equal-to operator, 49, 60
- < Less-than operator, 49, 60
- <= Less-than-or-equal-to operator, 49, 60
- * Multiplication operator, 49, 60
- /= Not-equal-to operator, 49, 60
- Subtraction operator, 49, 60

- ABS operator, 49, 60
- Absolute value operator, 49, 60
- Accumulate (multiply-and-), 285–288, 290, 292, 295, 299
- Actel FPGAs, 315
- ACTIVE attribute, 52, 61
- Adder circuits, 5–7, 42–43, 106–109, 194–198
 - Basic adder, 42–43
 - Carry-look-ahead adder, 196–198
 - Carry-ripple adder, 106–109, 194–196
 - Full-adder, 5–7, 279, 282–283
 - MAC (multiply-and-accumulate), 285–288, 290, 292, 295, 299
 - Overflow, 286
- Addition operator, 49, 54, 60, 195
- Advanced Synthesis Software, 329, 336–338
- AFTER clause, 28
- Altera
 - Advanced Synthesis Software, 329, 336–338
 - CPLDs, 312
 - MaxPlus II software, 5, 20, 329–342
 - Quartus II software, 4–5, 20, 343–353
- ALU, 75–78, 247–251
- AND operator, 36–37, 41–42, 48, 60
- Antifuse, 306, 313, 315
- Application Specific Integrated Circuit. *See* ASIC
- ARCHITECTURE
 - Description, 17
 - Introductory examples, 6, 17–22
 - Usage, 13–14
- Arithmetic logic unit. *See* ALU
- Arithmetic operators, 48–49, 60
- ARRAY, 30–33, 39
- Array indexes, 51
- ASIC, 3, 4, 6, 211, 293
- ASSERT statement, 270–271
- Assignment operators, 47–48, 60
- ATTRIBUTE statement, 52–53
- Attributes, 50–53, 61
 - Data attributes, 51, 61
 - Signal attributes, 52, 61
 - Summary, 61
 - User-defined attributes, 52–53

- Barrel/vector shifter circuits, 80–81, 109–111, 187–190
- Base type, 29–30
- BEGIN keyword
 - with ARCHITECTURE, 17
 - with BLOCK, 81–83
 - with FUNCTION, 253
 - with PROCEDURE, 265
 - with PROCESS, 91
- Behavioral description, 91
- Binary versus one-hot and two-hot encoding, 181–182
- Binary-to-Gray-code converter, 87
- BIT. *See* Data types
- BIT_VECTOR. *See* Data types
- BIT versus BIT_VECTOR, 41–42
- BLOCK statement
 - Guarded, 83–84
 - Simple, 81–83
- BODY. *See* PACKAGE BODY
- BOOLEAN. *See* Data types
- Buffer circuit, 73
- BUFFER mode, 16, 37–138, 140, 143, 145, 152, 154, 157

- Carry-look-ahead adder circuit. *See* Adder circuits
- Carry-ripple adder circuits. *See* Adder circuits
- CASE statement, 91, 100–104
- CASE versus IF, 112–113
- CASE versus WHEN, 113–114
- Combinational versus sequential circuits, 65–66
- Comparator circuits, 191–194
- Comparison operators, 49, 60
- Complex programmable logic devices. *See* CPLDs
- COMPONENT, 234, 236–244
- Concatenation operators, 50, 60
- Concurrent code, 65–84
- Concurrent statements
 - BLOCK, 65, 81–84
 - GENERATE, 65, 78–81, 195, 197–198
 - WHEN, 65, 69–78
 - WHEN versus CASE, 113–114
- Concurrent versus sequential code, 65–67
- CONFIGURATION statement, 72
- CONSTANT, 31, 47, 129–131, 174–176, 220–221, 234–235, 270
- Controller circuit for traffic light, 174–178, 186
- Controller circuit for vending machine, 202–208, 226–227
- Conv_integer function, 25, 37, 43, 255, 258–259

- Conv_signed function, 25, 37
- Conv_std_logic_vector function, 25, 38
- Conv_unsigned function, 25, 37
- Conversion functions. *See* Data conversion functions
- Count ones circuits, 130–133
- Counter circuits, 94–96, 99–100, 102–104, 144–146, 155, 164–166, 272
- CPLDs, 3–4, 305–306, 311–313, 317, 338–340
- Data attributes, 51, 61
- Data conversion functions, 25, 37–38
- Data objects. *See* Objects
- Data types, 25–43
 - ARRAY, 30–33, 39
 - BIT, 16–17, 21–22, 25–26, 39, 48–49, 54–55
 - BIT_VECTOR, 25–26, 28–30, 39, 48–49, 54–55
 - BIT versus BIT_VECTOR, 41–42
 - BOOLEAN, 25, 27–28, 30, 39, 204, 235–237, 254–257
 - Enumerated data types, 28–29, 39, 51, 53, 61, 70, 101, 160, 162, 164, 204
 - INTEGER, 25, 27–28, 30, 35, 37–39, 48–49
 - NATURAL, 27, 29–30, 34–35, 262, 300
 - Physical data types, 27
 - Port array, 33–34
 - REAL, 25, 27, 39
 - RECORD, 35, 39
 - SIGNED, 25, 27, 30, 35–39, 42–43, 48, 191, 285, 291, 297
 - STD_LOGIC, 25–27, 39
 - STD_LOGIC_VECTOR, 25–27, 39
 - STD_ULOGIC, 25–27, 39
 - STD_ULOGIC_VECTOR, 26, 39
 - SUBTYPE, 29–30, 39, 80, 226
 - UNSIGNED, 25, 27, 30, 35–39, 48, 191, 263–265
 - User-defined, 28–29, 34, 39, 299
- Decoder circuits, 55–57, 62–63, 156
- Delay circuit, 152
- DFF, 18–22, 83–84, 92–93, 99, 101–102, 125–127, 137–138, 142–143, 152–155, 157–158, 254–255
- Digital filter circuit, 289–294, 303
- Divider circuit, fixed-point, 198–202
- Division operator, 49, 60
- Don't care, 26
- D-type flip-flop. *See* DFF
- EDIF, 329, 336, 337
- EEPROM, 306, 309, 312–313
- ELSE
 - with WHEN, 69
 - with IF, 94
- ELSIF, 94
- Encoder circuit, 73–75
- END keyword
 - with ARCHITECTURE, 17
 - with BLOCK, 81–83
 - with CASE, 100
 - with FUNCTION, 253
 - with GENERATE, 78–79
 - with IF, 94
 - with LOOP, 105
 - with PROCEDURE, 265
 - with PROCESS, 91
- ENTITY
 - Description, 15–17
 - Introductory examples, 6, 17–22
 - Usage, 13–14
- Enum_encoding attribute, 53
- Enumerated data types. *See* Data types
- EPROM, 308–309
- Equal-to operator, 49, 60
- Error message. *See* ASSERT statement
- EVENT attribute, 52, 61
- Event counter circuit, 121
- Exclusive-NOR operator, 48, 60
- Exclusive-OR operator, 48, 60
- EXIT statement, 105–106, 111–112
- Exponentiation operator, 49, 60
- Field Programmable Gate Arrays. *See* FPGAs
- Finite State Machine. *See* FSM
- FIR filter. *See* Digital filter circuit
- Flip-flop. *See* DFF
- FOR statement
 - with GENERATE, 78–81
 - with LOOP, 105–112
 - with WAIT, 98–99
- FPGAs, 3–4, 305–306, 311–315, 317, 338–339
- Frequency divider circuit, 122, 138–140
- FSM, 159–182, 202–208, 213–218
- Full-adder. *See* Adder circuits
- FUNCTION, 253–265
 - Arithmetic shift function, 261–262
 - Convert-to-integer function, 37, 43, 255, 258–259
 - Definition and syntax, 253–254
 - Function location, 256–258
 - Multiplication function, 263–265
 - Multiplier function, 263–265
 - Overloaded “+” operator function, 260–261
 - Positive_edge function, 254–258
- FUNCTION versus PROCEDURE, 270
- GAL devices, 305–306, 309–312
- GENERATE statement, 65, 78–81, 195, 197–198
- GENERIC MAP, 244–247

- GENERIC statement, 54–60, 97, 108–109, 117, 191–195, 201, 221, 223, 225
- Gray code, 87
- Greater-than operator, 49, 60
- Greater-than-or-equal-to operator, 49, 60
- GUARDED BLOCK, 83–84

- Hexadecimal, 28, 43
- HIGH attribute, 51, 61
- High-impedance, 26, 73

- IEEE library, 13–15, 25–27, 35–38
- IEEE standards for VHDL, 25
- IF statement, 91, 94–97
- IF versus CASE, 112–113
- IF-GENERATE, 78–79
- IN mode, 16–17
- Inferred registers, number of, 140–151
- INOUT mode, 16, 225
- INTEGER. *See* Data types
- Intensity encoder circuit, 124
- IS keyword
 - with ARCHITECTURE, 17
 - with ATTRIBUTE, 53
 - with CASE, 100
 - with COMPONENT, 237
 - with ENTITY, 16
 - with FUNCTION, 253
 - with PACKAGE, 234
 - with PACKAGE BODY, 234
 - with PROCEDURE, 265
 - with SUBTYPE, 29–30
 - with TYPE, 28–29
- ISE software, 4–5, 20, 317–327

- Keypad debouncer/encoder circuit, 184–186

- LAST_ACTIVE attribute, 52
- LAST_EVENT attribute, 52
- LAST_VALUE attribute, 52
- Latch, 83–84, 119, 121 (*see also* DFF)
- Leading zeros counter circuit, 111–112
- LEFT attribute, 51, 61
- LEFTOF attribute, 51, 61
- LENGTH attribute, 51, 61
- Less-than operator, 49, 60
- Less-than-or-equal-to operator, 49, 60
- Library
 - Declaration, 13–15
 - IEEE library, 13, 15, 25–27, 35–38
 - Introductory examples, 18–22
 - Standard library, 13, 15, 25
 - Std_logic_1164 package, 13, 15, 25–27
 - Std_logic_1164 package, 25–27
 - Std_logic_arith package, 15, 25, 27, 35–38, 42–43, 191, 263, 285, 287, 291, 296, 299
 - Std_logic_signed package, 15, 25, 27, 36, 38, 48
 - Std_logic_unsigned package, 15, 25, 27, 36, 38, 48
 - Work library, 13, 15
- Logic systems
 - Binary (std library), 25
 - STD_LOGIC, 25–27
 - STD_ULOGIC, 25–27
- Logical operators, 48, 60
- LOOP statement, 91, 105–112
- LOW attribute, 51, 61

- MAC circuits, 285–288, 290, 292, 295, 299
- MAP
 - GENERIC MAP, 244–247
 - PORT MAP, 237, 241, 244–245, 251, 277–279, 281–284
- MaxPlus II software, 5, 20, 329–342
- Min_max procedure, 267–270
- MOD operator, 49, 60
- Mode
 - BUFFER, 16, 137–138, 140, 143, 145, 152, 154, 157
 - IN, 16–22
 - INOUT, 16, 225, 266
 - OUT, 16–22
- ModelSim software, 5, 20, 317, 325–326
- Modulus operator, 49, 60
- Multiplexer circuits, 68, 70–72, 85, 134–137
- Multiplication operator, 49, 60
- Multiplier circuits, 263–265, 275–285
- Multiply-and-accumulate circuit. *See* MAC circuits
- Multivalued logic systems
 - STD_LOGIC, 25–27
 - STD_ULOGIC, 25–27
- MUX. *See* Multiplexer circuits

- NAND operator, 48, 60
- NATURAL. *See* Data types
- Neural networks, 294–301, 303
- NEXT statement, 105–106
- NOR operator, 48, 60
- NOT operator, 48, 60
- Not-equal-to operator, 49, 60
- NULL statement, 101–102, 104, 113–114
- Number of registers inferred, 140–151
- Numeric data types. *See* Data types

- Objects
 - CONSTANT, 31, 47, 129–131, 174–176, 220–221, 234–235, 270

- Objects (cont.)
 - SIGNAL, 19, 21–22, 129–132
 - SIGNAL versus VARIABLE, 133–140
 - VARIABLE, 129–133
- ON keyword, 98–99
- One-hot encoding, 181–182
- Operator overloading, 53–54, 260–261
- Operators
 - Arithmetic, 48–49, 60
 - Assignment, 47–48, 60
 - Comparison, 49, 60
 - Concatenation, 50, 60
 - Logical, 48, 60
 - Shift, 49–50, 60
 - Summary, 60
- OR operator, 48, 60
- OTHERS clause, 40, 47–48, 69–73, 94, 101–102, 112–114
- OUT mode, 16–22
- Overloading, 53–54, 260–261
- PACKAGE
 - Description, 13–14, 93, 133, 233–236, 256, 266–270
 - Examples, 34–35, 235–236, 239, 241–244, 257–260, 263–264, 268–270, 275–284, 287, 299
- PACKAGE BODY, 234–236, 256–260, 269–270, 287
- PAL devices, 305–308, 339
- PALCE devices, 305
- Parity detector circuit, 57–59, 123
- Parity generator circuit, 59–60
- Physical data types, 27
- PLA devices, 305–306, 308–309, 312
- Playing with a seven-segment display, 212–216, 228
- PLDs
 - CPLD, 3–4, 305–306, 311–313, 317, 338–340
 - FPGA, 3–4, 305–306, 311–315, 317, 338–339
 - GAL, 305–306, 309–312
 - PAL, 305–308, 339
 - PLA, 305–306, 308–309, 312
- PORT
 - Introductory examples, 5–6, 18–22
 - Modes. *See* Mode
- Port array, 33–35
- PORT MAP, 237, 241, 244, 251, 277–279, 281–284
- Pre-defined data attributes, 51, 61
- Pre-defined data types, 25–28, 39
- Pre-defined operators, 47–50, 60
- Pre-defined signal attributes, 52, 61
- Priority encoder circuit, 85, 121–122
- PROCEDURE, 13–14, 91, 94, 105, 113, 130–133, 233–235, 265–270
- PROCEDURE versus FUNCTION, 270
- PROCESS
 - Description, 91–94
 - Introductory examples, 18–22, 56–60, 92–120
 - Programmable array logic. *See* PAL
 - Programmable logic array. *See* PLA
 - Programmable logic devices. *See* PLDs
- Quartus II software, 4–5, 20, 343–353
- QUIET attribute, 52
- RAM circuits, 116–118, 221–225
- Random access memory. *See* RAM circuits
- RANGE attribute, 28–30, 34–35, 37, 51, 61
- Read-only memory. *See* ROM circuits
- REAL. *See* Data types
- RECORD, 35, 39
- Registers. *See* DFF
- Registers inferred, number of, 140–151
- Relational operators. *See* Operators
- REM operator, 49, 60
- Remainder operator, 49, 60
- REPORT statement, 271
- Reserved words, 355
- Resolved data type, 26–27, 39
- REVERSE_RANGE attribute, 51, 61
- RIGHT attribute, 51, 61, 262
- ROL operator, 50, 60
- ROM circuits, 44, 220–221
- ROR operator, 50, 60
- Rotate left logic operator, 50, 60
- Rotate right logic operator, 50, 60
- RTL, 4
- SELECT statement, 67, 69–73, 113
- Sequential code, 65, 91–121
- Sequential statements
 - CASE, 91, 100–104, 112–114
 - IF, 91, 94–97, 112–113
 - LOOP, 91, 105–112
 - WAIT, 91, 97–100
- Serial data receiver circuit, 208–211, 227
- Seven-segment display, 212–216, 228
- Shift left arithmetic operator, 50, 60
- Shift left logic operator, 49–50, 60
- Shift operators, 49–50, 60.
- Shift register circuits, 96–97, 121, 146–151
- Shift right arithmetic operator, 50, 60
- Shift right logic operator, 49–50, 60
- SIGNAL, 19, 21–22, 129–132
- Signal attributes, 52, 61
- Signal generator circuits, 178–181, 183–184, 186, 217–220
- SIGNAL versus VARIABLE, 133–140
- SIGNED. *See* Data types

- SLA operator, 50, 60
- SLL operator, 49–50, 60
- Speed monitor circuit, 228–229
- SRA operator, 50, 60
- SRL operator, 49–50, 60
- STD_LOGIC, 25–27, 39
- STD_LOGIC_VECTOR, 25–27, 39
- STD_ULOGIC, 25–27, 39
- STD_ULOGIC_VECTOR, 26–27, 39
- Stop-watch circuit, 252
- String detector circuit, 172–174
- Subtype, 29–30, 39

- Timer circuits, 123
- Traffic light controller circuit, 174–178, 186
- Tri-state buffer circuit, 73
- Two-hot encoding, 181–182
- Types. *See* Data types

- UNAFFECTED statement, 69–70, 101, 113–114
- Unresolved data type, 27, 39
- UNSIGNED. *See* Data types
- UNTIL, 52, 98–100
- USE clause, 13, 15
- User-defined attributes, 52–53
- User-defined data types, 28–29, 34, 39, 299

- VARIABLE, 129–133
- VARIABLE versus SIGNAL, 133–140
- Vending machine controller circuit, 202–208, 226–227
- VHDL acronym, 3
- VHDL reserved words, 355

- WAIT statement, 91, 97–100
- WHEN statement, 65, 69–78
- WHEN versus CASE, 113–114
- WHILE statement, 105–106
- WITH, 67, 69–73, 113
- Work library, 13, 15

- Xilinx
 - CPLDs, 312
 - FPGAs, 314
 - ISE software, 4–5, 20, 317–327
- XNOR operator, 48, 60
- XOR operator, 48, 60

- “Z” logic state, 26–27, 39, 73