INDEX

Symbols
& (AND, Verilog), 42
= or := (blocking assignment), 22, 214
# (delay, Verilog), 76, 128
%f (formatter), 233
<, > (comparison), 122
{} (concatenation, Verilog), 101
& (concatenation, VHDL), 101
<= (non-blocking assignment), 54, 214
| (OR, Verilog), 42
<< (shift left, Verilog), 225
>> (shift right, Verilog), 226
? (ternary operator, Verilog), 93, 178
^ (XOR, Verilog), 42

A
Actel, 3, 5
addition, 215–219, 232
Alchitry Cu, 257
almost empty (AE), 118, 145
almost full (AF), 118, 145
Altera, 3, 16, 69
ALU (arithmetic logic unit), 197
always block, 53, 57
one vs. two, 149–157
AMD, 3, 16, 46, 64, 69, 87, 144, 188–189, 197, 246, 256
analog-to-digital converter (ADC), 5, 191, 195, 240
AND gate, 32, 40–42, 54, 58, 71
and keyword, 42
application-specific integrated circuits (ASICs), 7–8, 89
arbiter, 127, 228
architecture keyword, 22
Arduino, 5, 16, 257
arithmetic logic unit (ALU), 197
artificial intelligence (AI), 4
ASCII, 97–98
assertions, 84
assert keyword, 85–86, 128, 157
assign keyword, 22
assignment operators
  blocking (=, :=), 22, 214
  non-blocking (<=), 54, 214
Atmel. See Microchip Technology

B
bandwidth, 6, 249
bank (pins), 242
bidirectional pin, 240
binary, 33, 162
Binary_To_7Segment module, 161, 169
Bitcoin, 8
black box, 68
block RAM, 115–116, 186, 188, 191–194
creation, 193
error detection and correction, 193
features and limitations, 192
initializing, 128, 193
instantiation template, 186
precalculated table, 227
size, 192
Boolean algebra, 31–32, 36–40
on FPGA, 39
order of operations, 36
symbols, 36
bouncing of switch, 75
BRAM. See block RAM
buffers, 238–239, 246, 253
bugs, 67, 89, 117

C
career tips, 259–267
case statement, 152–153, 164, 172
clock data recovery (CDR), 251–253
clocks, 47
  constraints, 131
counting cycles, 77, 80, 96, 102, 108, 182
creation via PLL, 199–204
crossing domains, 141–146
  reference, 200
  skew, 249–251
$clog2()$ function, 80, 114
coding style, 78. See also naming convention
combinational logic, 57–59, 122
concatenation, 101, 107
constraints, 54–55, 131, 145–146
Coordinate Rotation Digital Computer (CORDIC), 229
cores, 185
core voltage, 32
counter
  LFSR, 102
  signed, 206
  traditional, 107
  traditional vs. LFSR, 110
  wraparound, 208
cyclic redundancy check (CRC), 35

D
datasheets, 63
data types, 206–209
  converting, 210–211
data valid (DV) signal, 112
DC balance, 251–253
debounce filter, 76
debouncing of switch, 75–84, 161, 169
decimals, 33, 230
delay, 50, 61, 73, 96. See also propagation delay
De Morgan’s law, 40
demultiplexer (demux), 92, 94–95, 106
depth, 112, 114
development board, 14, 255–259
device under test (DUT), 70
D flip-flop (DFF), 46, 64
Diamond Programmer, 14–15, 20
  installation, 18
  programming, 26–28
differential signaling, 243–245
Digilent, 16, 257
digital signal processing (DSP), 194.
  See also DSP block
$display()$ function, 128
division, 225–228
done pulse, 103, 105, 107
double-flopping, 141–142
double rate data (DDR), 245
drive strength, 242–243
DSP block, 194, 229
  analog vs. digital signals, 194–196
  arithmetic logic unit (ALU), 197
  creation, 198–199
  features, 197–198
  multiplier, 197
  pre-adder, 197
dual-port RAM, 111
$dumpfile()$ function, 73
duty cycle, 47

E
EDA Playground, 69–70, 72–74, 83, 211
edge detection, 51, 54, 179–180
electromagnetic interference (EMI), 244, 253
endmodule keyword, 22
entity keyword, 22
erenumeration, 152, 171
EPWave, 74
Ethernet, 245, 247
events, state machine, 148–149

F
fab (ASIC foundry), 7, 89
falling edge, 47
fiber optics, 253
file_open() function, 128
files, working with, 128
filter, 196
finance, 4
finite state machine (FSM). See state machine
first in, first out (FIFO), 116–117, 144–145
  AE (almost empty), 118, 145
  AF (almost full), 118, 145
crossing clock domains, 144
  implementation, 119–122
input and output, 117–119
interface, 117
fixed-point numbers, 230–236
flip-flops, 45–46
  behavior of, 48–52, 57
clock enable (EN), 46, 48–49, 61
clock input (>), 46, 48–51
creation in Verilog or VHDL, 54
data input (D), 46, 48–51
data output (Q), 46, 48–51
double-flopping, 141–142
edge detection, 51, 54
instantiation template, 189
JK and T flip-flops, 51
physical component, 63
register, 48, 53
reset, 61–63, 87, 152
  synchronous vs. asynchronous resets, 62
use in RAM, 115
floating-point numbers, 230
floating (isolated) electrical ground, 245
$fopen()$ function, 128
for loops, 128–131
FPGAs (field programmable gate arrays), 1–2
  applications, 4, 89
  vs. ASICs, 7–9
  history, 2–3
  languages, 9–11
  vs. microcontrollers, 5–8
  picking a family and package, 126
full adder, 229
full-duplex communication, 240, 247
fwrite() function, 128
H
half adder, 229
half-duplex communication, 240
hard IP, 3, 185
hard processors, 3
hardware debugging, 67
hardware description language (HDL), 9
hertz (Hz), 47
hexadecimal, 98
high impedance (aka hi-Z or tri-state), 239–240
high-speed data, 247, 251, 253
hold time ($t_h$), 133–135
I
I2C (inter-integrated circuit), 240
iCE40 (FPGA family), xxii–xxiii, 14–16, 63
iCEcube2, 14–15, 20, 55–56, 190
  building, 25
  creating a project, 22–24
  installation, 16–18
iCEstick, 256–257
ieee library, 22
if statement, 81
inference, 186
infrared (IR) cameras, 4
initial block, 73, 213
input, 239. See also GPIO instantiation, 186–189
integrated circuit (IC), 1
Intel, 3, 16, 64
intellectual property (IP), 3
interview tips, 265–267
input/output (I/O). See GPIO isolated (floating) electrical ground, 245
K
Karnaugh maps, 40
L
latches, 59–61, 124, 155
Lattice Diamond, 16, 190, 198–199

G
gain, applying to a signal, 196
genral purpose input/output. See GPIO
genrics, 78, 109, 114, 169
Go Board, 23, 47, 102, 256
GPIO, 238–239
  differential signaling, 243–245
drive strength, 242–243
operating voltage, 242
output enable (OE), 239
single-ended, 242–243
slew rate, 243, 245

guard condition, 160
GUI approach, 190–191
Lattice Semiconductor, 3, 14, 257
least significant bit, 98
LEDs, 15
  blinking, 51–56
  blinking selectively, 101–111
  lighting with logic gate, 40–42
  memory game, 158–183
  seven-segment display, 161
  showing pattern, 178–179
  wiring to switch, 19
linear feedback shift register (LFSR), 99–101
  applications, 100, 107
  code, 106
  counter, 102
  pseudorandom pattern generation, 171, 177
localparam keyword, 152
logic analyzer, 87–88
logic cell, 64
logic gates, 32–36, 38
logic minimization, 40, 124
look-up table (LUT), 38–40, 54
  physical component, 63
  shortcomings, 45
low-pass filter (LPF), 196
low-power double data rate (LPDDR), 116, 118, 148, 246
LVCMOS25, 242
LVCMOS33, 242, 245
LVDS (low-voltage differential signaling), 244
Manchester code, 250–252
math
  precalculating results, 227
  rules, 236
memory blocks, 192. See also RAM
metastability, 133–134, 141–142, 145, 191, 250
Microchip Technology, 3, 5
  microcontroller, 2, 5–7
  offloading math operations, 229
Microsemi. See Microchip Technology
Microsoft, 20
minimum clock period (\(t_{\text{clk(min)}}\)), 136
ModelSim, 69
module keyword, 21
most significant bit, 98–99, 207, 215–217
multiplexer (mux), 92–94
multiplication, 221–225, 234
multiplier, 197
multiply–accumulate (MAC) operation, 194, 197
naming convention, 22, 62, 78, 96, 121, 158, 213, 232
NAND (not and) gate, 35–36
negotiating a job offer, 267
nonrecurring engineering (NRE) cost, 7
non-synthesizable code, 127
NOR (not or) gate, 36
NOT gate, 34, 103
now keyword, 76, 128
numbers, 206, 208–211
  negative, 206
  representing in FPGA, 208
  signed vs. unsigned, 206–208
numeric_std package, 206
one-time programmable (OTP) FPGAs, 89
open keyword, 105, 182
operating voltage, 242
optimization, 124
OR gate, 33, 42
or keyword, 42
output enable (OE), 239
output keyword, 21–22, 239
overloading functions, 213–214
parallel communication, 248–250
parallel thinking, 2, 54
parameters, 78, 109, 114, 169
path slack, 139
.pcf file, 20, 24
PCI, 249
period (of clock), 47–48, 55, 80, 135
phase-locked loop (PLL), 142, 185, 199–204, 252
  creation, 202
  inputs, 200
  locked signal, 202
  operation, 200
  phase of a signal, 201
physical constraints file, 20, 24
pipelining, 136–140
place and route, 20, 131
  constraints, 24, 55, 131, 145, 183, 245
  mapping, 24
  pin report, 56
  timing errors, 56, 131–141, 145
  timing report, 56, 138
Pmod (peripheral module)
  connector, 16
positive edge, 54
pre-adder, 197
primitives, 144, 185–186, 190–191, 247
printed circuit board (PCB), 14
printing to console, 128
process block, 53, 57–58, 73, 213
  one vs. two, 149–155
Programmable Array Logic (PAL), 39
  projects
    blinking an LED, 51–57
    creating a memory game, 158–183
    debouncing a switch, 75–84
    lighting an LED with logic gates, 40–42
    selectively blinking an LED, 101–111
    wiring switches to LEDs, 19–28
propagation delay ($t_p$), 135–136, 249
protocol, 240, 245
pulse, 103, 107
  stretching, 144
push-button switch, 15, 101
  debouncing, 75, 161, 169
  edge detection, 180
  selector, 102
  wiring to LED, 19

R
  radar, 4
  radiation, 4, 89
  radix, 230
RAM (random-access memory), 111–116
  depth, 112, 114
  dual-port, 111
  single-port, 111
  width, 112, 114
range keyword, 80
real data type, 233
reg keyword, 53, 213
register, 48
replicated logic, 128
report keyword, 128
resetting a flip-flop. See flip-flops
resize() function, 217, 219
resource utilization. See synthesis
resume tips, 260–265
rising edge, 47–49, 54, 133, 152
routing, 5

S
  sampling (analog to digital), 195
  schematic, 25, 131
  .sdc file, 55
  selector inputs, 92–93
  self-checking testbenches, 84–86
  sensitivity list, 53–54, 58
  sequential logic, 57–58, 61
SerDes (serializer/deserializer), 247–250, 252–253
  8B/10B, 253
  clock data recovery (CDR), 251, 253
  DC balance, 251
  encoding scheme, 250, 253
  self-clocking signals, 250
  speed, 247, 250
  transceiver, 247
serial communication, 248–250
serial thinking, 2
set_io keyword, 24
set/reset pin, 61–63
setup time ($t_{su}$), 133–136
seven-segment display, 15, 159, 161–165
shift_left() function, 224–226

Q
  Q notation, 231
Quartus, 16, 18
Quine–McCluskey algorithm, 40
shift register, 50, 95–101, 129, 224–225
converting between serial and parallel, 97
creating delay, 96
divide by two, 225
multiply by two, 224
signals, 21–22
address, 114
analog vs. digital, 194–196
asynchronous, 141–146
clock, 47
data valid, 112, 117
declaring, 21
differential vs. single-ended, 243–245
dynamic sizing, 209
gain, applying, 196
initial condition, 86–87
input and output, 117–119
mapping to pin, 24–25
monitoring, 74–75
self-clocking, 250–252
synchronous vs. asynchronous, 248
toggling, 101–111
sign bit, 207
signed data type, 206–207, 210
signed() function, 211, 221
sign extension, 216–219
Simon (game), 158. See also testbench
simulation, 68–75
tools, 69–70
single-ended signaling, 243
single-port RAM, 111
slew rate, 243
state machine, 147–149, 152, 155, 157–160, 184
best practices, 157–158
diagram, 148–149, 158–159
events, 148–149
guard condition, 160
implementation, 149–155
initial state, 149, 152, 160
memory game project, 158–183
states, 148
transitions, 148
turnstile example, 148–152
std_logic_1164 package, 22
std_logic_arith package, 206
std_logic data type, 22
std_logic_vector data type, 206, 210
subtraction, 219–221, 232–234
switches. See push-button switch
synchronous logic, 57
syntax errors, 125
synthesis, 20, 124–127
constraints, 54–55
inference, 186
logic minimization, 40
notes, 124
pruning, 105, 208
report, 42, 55, 60, 84, 124, 183
syntax errors, 125
translate directives, 127
utilization, 42, 84, 110, 183, 194, 204
errors, 125–127
warnings, 124
synthesizable code, 77, 87, 127–130
system on a chip (SoC), 229
SystemVerilog, 70, 86, 89, 152

T

t_{clk(min)} (minimum clock period), 136
telecommunications, 4
ternary operator, 93, 179, 241
testbench, 70–72
creating, 81–83
clock creation, 82
math operations, 211–228
running, 74–75
self-checking, 84–86
speeding up, 83
writing, 71–73
t_h (hold time), 133
$time, 76, 128
time, measuring, 76–77
timing. See place and route
toggle a signal, 101–111
to_integer() function, 210–211
to_signed() function, 211
to_unsigned() function, 211
t_p (propagation delay), 135–136, 249
transition, 148

tri-state, 239
truncation, 233–234
truth tables, 32–39, 41
   AND, 33
   multiple gates, 37
   NAND, 35
   NOT, 34
   OR, 34
   three-input, 37
   XOR, 35
   $t_{pu}$ (setup time), 133–136
TTI (transistor–transistor logic), 242
Turing, Alan, 33
TWI (two-wire interface), 240
two-dimensional (2D) array, 115, 175
two’s complement, 207–208

**U**
unit under test (UUT), 70–75, 83
universal asynchronous receiver-transmitter (UART), 97–99
unsigned data type, 210
unsigned() function, 211
USB requirements, 15
utilization errors, 125–127. See also synthesis: utilization

**V**
variable keyword, 213
verification, 8, 88–89
Verilog
   background, 9–11
   enumeration support, 152
   weak typing, 10

VHDL
   2008 version, 109
   attributes, 211, 219
   background, 9–11
   data type conversions, 210–211
   strong typing, 10, 178, 182, 210, 213, 217
   verbosity, 22
Visual Studio Code (VS Code), 20
Vivado, 16, 18
evoltage, 46. See also GPIO

**W**
wait keyword, 73, 76, 128
waveforms, 74–75, 83–84
when keyword, 61
width, 112, 114
wraparound, 208
write() function, 128

**X**
Xilinx, 2–3, 16, 69
XNOR (exclusive not or) gate, 36, 99–100, 107
XOR (exclusive or) gate, 35–36, 39, 42, 99, 250–251
xor keyword, 42

**Z**
Z (high impedance), 239