Index

Note: Page numbers followed by "*f*," "*t*," and "*b*" refer to figures, tables, and boxes, respectively; page numbers preceded by "e" refer to online material.

0, 5–6, 20. *See also* LOW, FALSE
1, 5–6, 20. *See also* HIGH, TRUE
32-bit datapath, 394
32-bit instructions, 332
64-bit architecture, 373
74xx series logic, 543.e1–2
  parts
    2:1 mux (74157), 543.e4*f*
    3:8 decoder (74138), 543.e4*f*
    4:1 mux (74153), 543.e4*f*
    AND (7408), 543.e3*f*
    AND3 (7411), 543.e3*f*
    AND4 (7421), 543.e3*f*
    counter (74161, 74163), 543.e4*f*
    FLOP (7474), 543.e3*f*
    NAND (7400), 543.e3*f*
    NOR (7402), 543.e3*f*
    NOT (7404), 543.e3*f*
    OR (7432), 543.e3*f*
    register (74377), 543.e4*f*
    tristate buffer (74244), 543.e4*f*
    XOR (7486), 543.e3*f*
`#define`, 545.e5–6
`#include`, 545.e6–7. *See also* Standard libraries

A

ABI. *See* Application Binary Interface (ABI)
Abstraction, 2–3
  digital. *See* Digital abstraction
Accumulator, 372
Active low, 72–73
A/D conversion, 542.e37–39
Ad hoc testing, 464
ADCs. *See* Analog-to-digital converters (ADCs)
`add` instruction, 301, 444, 472–473
Adders, 237–244
  carry propagate, 238
  carry-lookahead, 239–241
  full, 238
  half, 238
  HDL for, 181–182, 198*b*, 461*b*
  prefix, 241–243
  ripple-carry, 238–239
Addition, 12*f*, 13*b*, 16*b*, 237–244, 301*b*. *See also* Adders
  binary, 12–13
  floating point, 261
  signed binary, 13–17
Address. *See also* Memory
  physical, 521, 521*f*
  translation, 523–524
  virtual, 521. *See also* Virtual memory
Addressing modes, RISC-V, 341–342
  base, 341
  immediate, 341
  PC-relative, 341–342
  register, 341
Advanced Micro Devices (AMD), 300
Advanced microarchitecture, 468–482
  branch prediction. *See* Branch prediction
  deep pipelines. *See* Deep pipelines
  heterogeneous multiprocessors. *See* Heterogeneous multiprocessors
  homogeneous multiprocessors. *See* Homogeneous multiprocessors
  micro-operations. *See* Microoperations
  multiprocessors. *See* Multiprocessors
  multithreading. *See* Multithreading
  out-of-order processor. *See* Out-of-order processor
  register renaming. *See* Register renaming
  single instruction multiple data. *See* Single instruction multiple data (SIMD)
  superscalar processor. *See* Superscalar processor
AHB. *See* Advanced High-performance Bus (AHB)
AHB-Lite bus, 485
Altera FPGA, 278
ALU. *See* Arithmetic/logical unit (ALU)
ALU Decoder, 408–409, 409*t*
ALUControl, 401–402, 457–458
ALUOp, 408
ALUResult, 418
ALUSrc, 404
AMAT. *See* Average memory access time (AMAT)
AMBA. *See* Advanced Microcontroller Bus Architecture (AMBA)
AMD. *See* Advanced Micro Devices (AMD)

- AMD64, 373
- Amdahl, Gene, 504
- Amdahl's Law, 505
- American Standard Code for Information Interchange (ASCII), 545.e8–9, 545.e27–28
- Analog I/O, 542.e31–39
  - A/D conversion, 542.e37–39
  - D/A conversion, 542.e31–34
  - pulse-width modulation (PWM), 542.e34–37
- Analog-to-digital converters (ADCs), 542.e31, 542.e37–39
- Analytical engine, 5
- `and` instruction, 309
- AND gate, 18–19, 177*b*
  - chips (7408, 7411, 7421), 543.e3*f*
  - truth table, 18*f*, 20*f*
  - using CMOS transistors, 31
- AND-OR (AO) gate, 45
- Anode, 25–26
- Antidependence, 475
- Application-specific integrated circuits (ASICs), 543.e9
- Architectural state, 344, 369
  - for RISC-V, 393–394
- Architecture, 299
  - assembly language, 300–308
    - instructions, 301–302
    - operands, 302–308
  - compiling, assembling, and loading, 344–355
    - assembling, 350–353
    - compilation, 348–350
    - linking, 353–355
    - loading, 355
    - memory map, 344–346
  - evolution of RISC-V architecture, 363–366
    - comparison of RISC-V and ARM architectures, 365–366
    - comparison of RISC-V and MIPS architectures, 365
    - RISC-V base instruction sets and extensions, 364–365
  - machine language, 332–344
    - addressing modes, 341–342
    - immediate encodings, 340–341
    - I-type instruction format, 334–336
    - interpreting, 342–343
    - R-type instruction format, 332–334
    - S/B-type instruction format, 336–338
    - stored program, 343–344
    - U/J-type instruction format, 338–340
  - odds and ends, 355–363
    - compressed instructions, 362–363
    - endianness, 355–356
    - exceptions, 356–360
    - floating-point instructions, 361–362
    - signed and unsigned instructions, 360–361
  - programming, 308–332
    - arrays, 317–320
    - branching, 311–313
    - conditional statements, 313–315
    - function calls, 320–330
    - getting loopy, 315–317
    - logical and arithmetic instructions, 308–311
    - memory, 306–308
    - pseudoinstructions, 330–332
  - x86 architecture, 366–374
    - big picture, 373–374
    - instruction encoding, 371–372
    - instructions, 369
    - operands, 367–368
    - peculiarities, 372–373
    - registers, 366–367
    - status flags, 369
- Arguments, 320, 545.e24
  - pass by reference, 545.e22
  - pass by value, 545.e22
- Arithmetic
  - C operators, 545.e11–15
  - circuits, 237–256
  - HDL operators, 183*b*
  - RISC-V instructions, 309
- Arithmetic/logical unit (ALU), 247–251
  - implementation of, 247–248
  - in processor, 401–402, 409*b*
- Arrays, 317–319, 545.e23–27
  - accessing, 316*b,* 545.e23
  - bytes and characters, 317–320
  - comparison or assignment of, 545.e27–28
  - declaration, 319*b,* 545.e23
  - indexing, 317, 545.e23–27
  - initialization, 545.e23*b*–e24*b,* 545.e24*b,* 545.e25
  - as input argument, 545.e24*b*–25*b,* 545.e24
  - multi-dimensional, 545.e25–26
- ASCII. *See* American Standard Code for Information Interchange (ASCII)
- ASICs. *See* Application-specific integrated circuits (ASICs)
- Assembler, 344, 350–351
- Assembler directives, 346–348
- Assembling, 344–355
- Assembly language, RISC-V, 300–332
  - instructions, 301–302
  - operands, 302–308
  - translating high-level code to, 303–332
  - translating machine language to, 342*b*–343*b*
- Assembly language, x86. *See* x86
- Associativity
  - in Boolean algebra, 60, 61*t*
  - in caches, 505–511
- Astable circuits, 117*b*
- Asymmetric multiprocessors. *See* Heterogeneous multiprocessors
- Asynchronous circuits, 120–121
- Asynchronous resettable flip-flops
  - definition, 114
  - HDL, 193*b*
- Asynchronous serial link, 542.e24. *See also* Universal Asynchronous Receiver Transmitter (UART)
- Average memory access time (AMAT), 503–505

B

- B-type instruction format. *See* S/B-type instruction format
- Babbage, Charles, 5
- Base addressing, 341
- Baud rate, 542.e24–25
- BCD. *See* Binary coded decimal (BCD)
- Behavioral modeling, 188
- Benchmarks, 397
- `beq` instruction. *See* Branch if equal (`beq`)
- `bge` instruction. *See* Branch if greater than or equal (`bge`)
- Biased exponent, 258–259
- Big-endian memory, 355–356, 356*f*
- Big-endian order in HDL, 176
- Binary addition, 12–13. *See also* Adders; Addition
- Binary coded decimal (BCD), 260*b*
- Binary encoding, 127–128
  - for divide-by-3 counter, 129*t*
  - for traffic light FSM, 133

Binary numbers
  signed, 13–17
  unsigned, 7–9
Binary to decimal conversion, 8*b*, 9*f*
Binary to hexadecimal conversion, 10*b*
Bipolar junction transistors, 24
Bipolar motor drive, 542.e61*f*
Bipolar signaling, 542.e25
Bipolar stepper motor, 542.e59*f*, 542.e60–63
  AIRPAX LB82773-M1, 542.e62*f*
  direct drive current, 542.e62*f*
Bistable element, 107–108
Bit, 6
  dirty, 519
  least significant, 11, 12*f*
  most significant, 11, 12*f*
  sign, 13–14
  use, 515–516
  valid, 509
Bit cells, 267
  DRAM, 268–269
  ROM, 268
  SRAM, 269
Bit swizzling, 186
Bitline, 267
Bitwise operators, 175–176
Block, 505
Block offset, 513–514
Block size (b), 505–506
Blocking and nonblocking assignments, 197, 203–207
`blt` instruction. *See* Branch if less than (`blt`)
BlueSMiRF silver module, 542.e53–54
Bluetooth wireless communication, 542.e53–54
  BlueSMiRF silver module, 542.e53–54
  classes, 542.e53*t*
`bne` instruction. *See* Branch if not equal (`bne`)
Boole, George, 6
Boolean algebra, 58–64
  axioms, 59
  equation simplification, 63–64
  theorems, 59–62
Boolean equations, 56–58
  product-of-sums (POS) form, 58
  sum-of-products (SOP) form, 56–58
Boolean logic, 6. *See also* Boolean algebra; Logic gates
Boolean theorems, 59–62
  associativity, 60
  combining, 60
  commutativity, 60
  complements, 60
  consensus, 60
  covering, 60
  De Morgan's, 61
  distributivity, 60
  idempotency, 60
  identity, 59
  involution, 60
  null element, 59–60
Branch if equal (`beq`), 311
Branch if greater than or equal (`bge`), 311, 316*b*
Branch if less than (`blt`), 360
Branch if not equal (`bne`), 311–312
Branch instructions, 311
Branch misprediction penalty, 451, 470
Branch prediction, 470–472
Branch target address (BTA), 337, 430–431, 432*f*
Branch target buffer, 470–471
Branching, 311–313
  conditional, 311–312
  unconditional, *see* Jump
Breadboards, 543.e20
BTA. *See* Branch target address (BTA)
Bubble, 18, 61–62
  pushing, 61–62, 67–68
Bubble, in pipeline, 448
Buffers, 18
  lack of, 115
  tristate, 72, 73
Bugs, 173–174
  in C code, 545.e44–48
Bus, 54
  tristate, 73*f*
Bypassing, 446. *See also* Forwarding
Byte, 11–12, 317–320. *See also* Characters
  least significant, 11
  most significant, 11
Byte offset, 508
Byte-addressable memory, 306–307
  big-endian, 355
  little-endian, 355

C

C programming, 545.e1–48
  common mistakes. *See* Common mistakes in C
  conditional statements. *See* Conditional statements
  control-flow statements. *See* Control flow statements
  data types. *See* Data types
  executing a program, 545.e3
  function calls. *See* Function calls
  loops. *See* Loops
  operators. *See* Operators
  simple program, 545.e3
  standard libraries. *See* Standard libraries
  variables. *See* Variables in C
Caches, 505–520
  address fields
    block offset, 513
    byte offset, 508
    set bits, 508
    tag, 508
  advanced design, 516–520
  multiple level, 516–518
  organizations, 515–516
    direct mapped, 507–511
    fully associative, 512
    multi-way set associative, 511–512
  parameters
    block size (*b*), 513–514
    capacity (*C*), 505
    degree of associativity (*N*), 505
    number of blocks (*B*), 505, 512–514
    number of sets (*S*), 505
  performance of
    capacity, 518
    compulsory, 518
    conflict, 512, 518
    hit, 502
    hit rate, 503–505
    miss, 502, 518
    miss rate, 503–505
    miss rate *vs*. cache parameters, 518–519
    penalty, 513
    reducing, 518–519
  replacement policy, 527–529
  status bits
    dirty bit (D), 528
    use bit (U), 528–529
    valid bit (V), 525
  write policy, 519–520
    write-back, 519, 519*b*
    write-through, 519, 519*b*

CAD. *See* Computer-aided design (CAD)
Callee, 320
Callee save rule, 326
Callee-saved registers, 325
Caller save rule, 326
Caller-saved registers, 325
Canonical form. *See* Sum-of-products (SOP) form; Product-of-sums (POS) form
Capacitors, 26
Capacity, of cache, 505
Capacity miss, 518
Carry propagate adder (CPA). *See* Carry-lookahead adder (CLA); Prefix adders; Ripple-carry adder
Carry-lookahead adder (CLA), 239–241, 240*f*
`case` statement, in HDL, 199–200. *See also* Switch/case statement
`casez, case?`, in HDL, 203*b*
Cathode, 25–26
Cathode ray tube (CRT), 542.e47–48. *See also* VGA (Video Graphics Array) monitor
  horizontal blanking interval, 542.e47–48
  vertical blanking interval, 542.e47–48
Character LCDs, 542.e44–47
Characters (`char`), 317–319, 542.e44–47
  arrays. *See also* Strings
  C type, 545.e27
Chips, 26
  multiprocessors, 479–482
Chopper constant current drive, 542.e62
Circuits
  74xx series. *See* 74xx series logic
  application-specific integrated (ASICs), 543.e9
  astable, 117*b*
  asynchronous, 120–121
  combinational. *See* Combinational logic
  definition of, 53–54
  delay, 86–90
  glitches in, 90–93
  multiple-output, 66*b*
  priority, 66
  sequential. *See* Sequential logic
  synchronous, 120–121
  synchronous sequential, 121
  synthesized, 174*f*, 176*f*, 177*f*
  timing, 86–93, 139–155
CISC. *See* Complex instruction set computer
CLA. *See* Carry-lookahead adder (CLA)
CLBs. *See* Configurable logic blocks (CLBs)
Clock cycles per instruction (CPI). *See* Cycles per instruction (CPI)
Clock period, 140, 398
Clock skew, 146–149, 468–469
Clustered multiprocessors, 481–482
CMOS. *See* Complementary Metal-Oxide-Semiconductor gates (CMOS)
Combinational composition, 54–55
Combinational logic, 172*b*
  design, 53–56
    Boolean algebra, 58–64
    Boolean equations, 56–58
    building blocks, 81–86
    delays, 86–90
    don't cares, 79–80
    Karnaugh maps (K-maps), 73–81
    multilevel, 67–71
    precedence, 56
    timing, 86–93
    two-level, 67–68
    X (contention). *See* Contention (X)
    X (don't cares). *See* Don't care (X)
    Z (floating). *See* Floating (Z)
  HDLs. *See* Hardware description languages (HDLs)
Combining theorem, 60
Command line arguments, 545.e44
Comments
  in RISC-V assembly, 302
  in C, 302, 545.e5
  in SystemVerilog, 178
  in VHDL, 178
Common mistakes in C, 545.e44–48
Comparators, 245–246
Comparison
  in hardware. *See* Arithmetic/logical unit (ALU); Comparators
  processor performance, 397–398
  using ALU, 249
Compiler, in C, 348–350, 545.e4, 545.e43
Complementary Metal-Oxide-Semiconductor gates (CMOS), 23–32
Complements theorem, 60
Complex instruction set computer (CISC) architectures, 302, 366, 469–470
Complexity management, 2–5
  abstraction, 2–3
  discipline, 3–4
  hierarchy, 4
  modularity, 4
  regularity, 4
Compressed instructions, 362–363
Compulsory miss, 518
Computer-aided design (CAD), 127, 171, 246
Concurrent signal assignment statement, 177, 182
Condition flags, 369
Condition mnemonics, 332
Conditional assignment, 179–180
Conditional branches, 311, 312*b*
Conditional operator, 179
Conditional signal assignments, 179
Conditional statements, 313–315
  in RISC-V assembly
    `if`, 313
    `if/else`, 313–314
    `switch/case`, 314–315
  in C, 545.e17–19
    `if`, 545.e17
    `if/else`, 545.e17
    `switch/case`, 545.e18
  in HDL, 199–200
    `case`, 199–200
    `casez`, `case?`, 203
    `if`, `if/else`, 200–203
Configurable logic blocks (CLBs), 167, 278. *See also* Logic elements (LEs)
Conflict miss, 518
Consensus theorem, 60, 62*b*
Constants
  in RISC-V assembly, 302–308. *See also* Immediates
  in C, 545.e5–6
Contamination delay, 86–90. *See also* Short path
Contention (x), 67*b*, 71–72
Context switching, 478
Continuous assignment statements, 177, 191
Control hazard, 446, 450–453
Control signals, 81, 399, 406*f*, 407–410, 422–435

Control unit, 394. *See also* ALU Decoder; Main Decoder; Instr Decoder
- of multicycle RISC-V processor, 422–432
- of pipelined RISC-V processor, 439–456
- of single-cycle RISC-V processor, 398–415

Control-flow statements
- conditional statements. *See* Conditional statements
- loops. *See* Loops

CoreMark, 397

Counters, 261–263
- divide-by-3, 129*t*

Covering theorem, 60

CPA. *See* Carry propagate adder (CPA)

CPI. *See* Cycles per instruction (CPI)

Critical path, 87–88, 412–414, 414*f*

Cross-coupled inverters, 108, 269
- bistable operation of, 108*f*

Cycle time. *See* Clock period

Cycles per instruction (CPI), 398, 413, 437, 454, 468

Cyclic paths, 118

Cyclone IV FPGA, 278–283

D

D flip-flops. *See* Flip-flops

D latch. *See* Latches

D/A conversion, 542.e31–34

DACs. *See* Digital-to-analog converters (DACs)

Data hazard, 446–450

Data memory, 395, 467*b*

Data segment, 345–346

Data types, 545.e21–35
- arrays. *See* Arrays
- characters. *See* Characters (`char`)
- dynamic memory allocation. *See* Dynamic memory allocation (`malloc`, `free`)
- linked list. *See* Linked list
- pointers. *See* Pointers
- strings. *See* Strings
- structures. *See* Structures (`struct`)
- `typedef`, 545.e31

Datapath
- multicycle processor, 416–438
  - I-type ALU instructions, 433–434
  - `jal` instruction, 433–434
  - `lw` instruction, 417–419
  - R-type instructions, 421
  - `sw` instruction, 420–421
- pipelined processor, 441–443
- single-cycle processor, 399–407
  - I-type ALU instructions, 410–411
  - `jal` instruction, 411–412
  - `lw` instruction, 400–402
  - R-type instructions, 404–405
  - `sw` instruction, 403–404

Datasheets, 543.e9–13

DC motors, 542.e55–59
- H-bridge, 542.e55–58, 542.e56*t*
- shaft encoder, 542.e58*f*, 542.e54–55, 542.e58–59

DC transfer characteristics, 22, 23*f*, 45*f*. *See also* Direct current (DC) transfer characteristics; Noise margins

DDR. *See* Double-data rate memory (DDR)

De Morgan, Augustus, 61*b*

De Morgan's theorem, 61

DE-9 cable, 542.e26*f*

Decimal numbers, 7

Decimal to binary conversion, 9*b*

Decimal to hexadecimal conversion, 11*b*

Decode stage, 439

Decoders
- definition of, 84–86
- HDL for
  - behavioral, 200*b*
  - parameterized, 217*b*
- logic using, 85–86
- seven-segment. *See* Seven-segment display decoder

Deep pipelines, 468–469

Delay, logic gates. *See also* Propagation delay, Contamination Delay
- in HDL (simulation only), 186–187

`DeleteUser` function, 545.e33

Dennard, Robert, 268*b*

Destination register (**rd**), 333

Device driver, 542.e3, 542.e10–14

Device under test (DUT), 218

Dhrystone, 397

Dice, 26

Dielectric, 27

Digital abstraction, 5–7, 20–24

Digital circuits. *See* Logic

Digital signal processors (DSPs), 254, 257*b*

Digital system implementation, 543.e1–37
- 74xx series logic. *See* 74xx series logic
- application-specific integrated circuits (ASICs), 543.e9
- assembly of, 543.e19–23
- breadboards, 543.e20
- datasheets, 543.e9–13
- economics, 543.e35–37
- logic families, 543.e15–16
- packaging, 543.e19–23
- printed circuit boards, 543.e20–23
- programmable logic, 543.e2–9

Digital-to-analog converters (DACs), 542.e31–34

DIMM. *See* Dual inline memory module (DIMM)

Diodes, 25–26
- p-n junction, 26*f*

DIPs. *See* Dual-inline packages (DIPs)

Direct current (DC) transfer characteristics, 22

Direct mapped cache, 506–511

Direct voltage drive, 542.e63

Dirty bit (D), 519, 528

Discipline
- dynamic, 140. *See also* Timing analysis
- static, 140. *See also* Noise margins

Discrete-valued variables, 5

Distributivity theorem, 60

Divide-by-3 counter
- design of, 127, 128*f*, 128*t*, 129*f*
- HDL for, 208*b*

Divider, 255–256

Division
- circuits, 254–256

Do/while loops, in C, 545.e19

Don't care (X), 66–67, 79–80, 203

Dopant atoms, 25

Double, C type, 545.e8*t*

Double-precision formats, 259–260

DRAM. *See* Dynamic random access memory (DRAM)

Dual-inline packages (DIPs), 26*b*, 543.e1

DUT. *See* Device under test (DUT)

Dynamic branch predictors, 470–471

Dynamic data segment, 346

Dynamic discipline, 140. *See also* Timing analysis

Dynamic memory allocation (malloc, free), 545.e40
Dynamic power, 32
Dynamic random access memory (DRAM), 266, 268–269, 268*f*

E

Economics, 543.e35–37
Edge-triggered flip-flop. *See* Flip-flops
EEPROM. *See* Electrically erasable programmable read only memory (EEPROM)
EFLAGS register, 369
EIP. *See* Extended instruction pointer (EIP)
Electrically erasable programmable read only memory (EEPROM), 272
Embedded I/O (input/output) systems, 542.e1–43
  A/D conversion, 542.e37–39
  analog I/O, 542.e31–39
  D/A conversion, 542.e31–34
  digital I/O, 542.e5–10
  general-purpose I/O (GPIO), 542.e5–10
  interrupts, 542.e39–43
  LCDs. *See* Liquid Crystal Displays (LCDs)
  microcontroller peripherals, 542.e43
  motors. *See* Motors
  serial I/O, 542.e14–29
  timers, 542.e29–31
  VGA monitor. *See* VGA (Video Graphics Array) monitor
Enabled flip-flops, 113–114
Enabled registers, 194. *See also* Flip-flops
Endianness, 355–356
EPROM. *See* Erasable programmable read only memory (EPROM)
Equality comparator, 245
Equation minimization
  using Boolean algebra, 63*b*
  using Karnaugh maps. *See* Karnaugh maps (K-maps)
Erasable programmable read only memory (EPROM), 543.e2
Ethernet, 73, 542.e15
Exceptions, 356–360
  exception-related instructions, 358
  execution modes and privilege levels, 356–357
  handler, 357–358
Execution time, 397
exit, 545.e41
Extended instruction pointer (EIP), 367

F

factorial function call, 327–329
  stack during, 329
Factoring state machines, 132–135
False, 6, 18, 56–58, 58*f*, 109–114
Fast Fourier Transform (FFT), 257*b*
FDIV. *See* Floating-point division (FDIV)
FE310-G002 system-on-chip, 542.e5
FFT. *See* Fast Fourier Transform (FFT)
Field programmable gate arrays (FPGAs), 542.e23*f*, 542.e49*f*, 276, 543.e7–9
  driving VGA cable, 542.e49*f*
  in SPI interface, 542.e19, 542.e19–20
File manipulation, in C, 545.e38–39
Finite state machines (FSMs), 121–139, 207–209, 423
  complete multicycle control, 434*f*
  deriving from circuit, 135*b*–138*b*
  divide-by-3 FSM, 127*b*–128*b*, 208*b*
  factoring, 132–135
  in HDL, 207–209
  LE configuration for, 280*b*–281*b*
  Mealy FSM, 130–132
  Moore FSM, 130–132
  snail/pattern recognizer FSM, 130*b*–131*b*, 210*b*
  state encodings, 127–128. *See also* Binary encoding; One-cold encoding; One-hot encoding
  state transition diagram, 122*f*, 123*f*
  traffic light FSM, 121–127
Fixed-point numbers, 256–257
Flags, 248
Flash memory, 272. *See also* Solid state drive (SSD)
Flip-flops, 112, 112–113, 113–114, 114, 114–116, 191. *See also* Registers
  back-to-back, 111, 143–146, 195. *See also* Synchronizers
  comparison with latches, 116–117
  enabled, 113–114
  HDL for, 451. *See also* Registers
  metastable state of. *See* Metastability
  register, 112–113
  resettable, 114
  scannable, 265
  shift register, 169
  transistor count, 112*b*
  transistor-level, 114–116
Float, C type, 545.e6
  print formats of, 545.e36*t*
Floating (Z), 72–73
Floating output node, 115
Floating-gate transistor, 272. *See also* Flash memory
Floating-point division (FDIV) bug, 174, 261*b*
Floating-point instructions, 361–362
Floating-point numbers, 257–261
  addition, 261
  formats, single-and double-precision, quad-precision, 259–260
  in programming. *See* Double, C type; Float, C type
  rounding, 260–261
  special cases
    infinity, 259
    NaN, 259
Floating-point unit (FPU), 261*b*
For loops, 315–316, 545.e20
Format conversion (atoi, atol, atof), 545.e41
Forwarding, 446–448. *See also* Hazards
FPGAs. *See* Field programmable gate arrays (FPGAs)
FPU. *See* Floating-point unit (FPU)
Frequency shift keying (FSK), 542.e53–54
  and GFSK waveforms, 542.e53*f*
Front porch, 542.e48
FSK. *See* Frequency shift keying (FSK)
FSMs. *See* Finite state machines (FSMs)
Full adder, 54, 180–181, 182*b*, 198*b*, 238
  using always/process statement, 238
Fully associative cache, 512
funct3 field, 333*b*, 334
funct7 field, 332*b*, 333

Function calls, 320–330, 545.e15–16
additional arguments and local variables, 330
arguments, 330
leaf, 326
multiple registers, loading and storing, 365–366
naming conventions, 545.e16
with no inputs or outputs, 320–321, 545.e15
nonleaf, 326–327
preserved registers, 324–326
prototypes, 545.e16
recursive, 327–329
return, 320–321, 545.e15
stack, use of, 322–324. *See also* Stack
Fuse-programmable ROM, 271

G

Gates
AND, 18–19, 31*f*
buffer, 18
multiple-input, 19–20
NAND, 18, 29–30, 30*f*
NOR, 19, 30*f*, 109–110
NOT, 18, 29*f*
OR, 19
transistor-level. *See* Transistors
XNOR, 19*b*
XOR, 19
General-purpose I/O (GPIO), 542.e5–10
switches and LEDs example, 542.e13*b*–14*b*
Generate signal, 239–241
GenWaves function, 542.e33–34
Glitches, 90–93
Global data segment, 345
GND. *See* Ground (GND)
GPIO. *See* General-purpose I/O (GPIO)
GPUs. *See* Graphics processing units (GPUs)
Graphics accelerators, 480–481
Graphics processing units (GPUs), 472
Gray, Frank, 74*b*
Gray codes, 74*b*
Ground (GND), 20
symbol for, 29

H

Half adder, 238
Hard disk, 502–504. *See also* Hard drive
Hard drive, 502, 520. *See also* Hard disk; Solid state drive (SSD); Virtual memory
Hardware description languages (HDLs), 171–223. *See also* SystemVerilog; VHDL (VHSIC Hardware Description Language)
2:1 multiplexer, 463*b*
adder, 461*b*
combinational logic, 172*b*, 175–188
bitwise operators, 175–176
blocking and nonblocking assignments, 203–207
case statements, 199–200
conditional assignment, 179–180
delays, 186–187
data memory, 467*b*
data types, 211–215
flip-flop, 191–192, 192*b*
history of, 172–173
If statements, 200–203
internal variables, 180–182
numbers, 183
operators and precedence, 182–183, 183*b*
reduction operators, 178
immediate extension, 430–431
instruction memory, 466*b*–467*b*
modules, 171–172
parameterized modules, 215–218
processor building blocks, 461–464
register file, 461*b*
resettable flip-flop, 192–194, 193*b*, 194*f*, 462*b*
resettable flip-flop with enable, 194–195, 194*b*, 195*f*, 463*b*
sequential logic, 191–196, 207
simulation and synthesis, 173–175
single-cycle RISC-V processor, 456–467
structural modeling, 188–190
testbench, 218–222, 464–468
top-level module, 466*b*
Hardware handshaking, 542.e25
Hardware multithreaded processor, 479
Hardware reduction, 68–69. *See also* Equation minimization
Hazard unit, 446–448
Hazards. *See also* Hazard unit
control hazards, 446, 450–452
data hazards, 446–448
pipelined processor, 443–454
read after write (RAW), 444, 475
solving
control hazards, 450–452
forwarding, 446–448
stalls, 448–450
write after read (WAR), 475
write after write (WAW), 476
H-bridge control, 542.e56*f*
HDL. *See* Hardware description languages (HDLs); SystemVerilog; VHDL
Heap, 346
Heterogeneous multiprocessors, 480–481
Hexadecimal numbers, 9–11
Hexadecimal to binary and decimal conversion, 9*b*
Hierarchy, 4
HIGH, 6
High-level programming languages, 308, 545.e5
compiling, assembling, and loading, 344–355
translating into assembly, 304*b*
High-performance microprocessors, 468
Hit, 502
Hit rate, 503–505
Hold time constraint, 142–143
with clock skew, 146–149
Hold time violations, 143, 143–144, 145*b*–146*b*, 148*b*–149*b*
Homogeneous multiprocessors, 479–480
Hopper, Grace, 346*b*

I

I-type instruction format, 334–336
I/O. *See* Input/output (I/O) systems
IA-32 architecture. *See* x86
IA-64, 370*t*
ICs. *See* Integrated circuits (ICs)
Idempotency theorem, 60
Identity theorem, 59
Idioms, 175

IEEE. *See* Institute of Electrical and Electronics Engineers (IEEE)
If statements
  in C, 545.e17
  in HDL, 200–203
  in RISC-V assembly, 313, 313*b*
If/else statements, 308
  in C, 545.e17
  in HDL, 200–203
  in RISC-V assembly, 314*b*
ILP. *See* Instruction level parallelism (ILP)
IM. *See* Instruction memory
Immediate addressing, 341
Immediate encodings, 340–341
Immediate extension, 406*b*, 412*b*
Immediate field (**imm**), 334–336
Immediate instructions (I-type instructions), *see* I-type instruction format
Immediates, 304–306, 365–366. *See also* Constants
Implicit leading one, 258
Information, amount of, 6
Initializing
  arrays in C, 545.e23*b*, 545.e24*b*, 545.e24*b*
  variables in C, 545.e11
Input/Output (I/O) systems, 542.e1–43
  device driver, 542.e3, 542.e10–14
  embedded I/O systems. *See* Embedded I/O (input/output) systems
  I/O registers, 542.e3
  memory-mapped I/O, 542.e1–3
  personal computer I/O systems. *See* Personal computer (PC) I/O systems
Input/output elements (IOEs), 278
Institute of Electrical and Electronics Engineers (IEEE), 173–174, 258
Instr Decoder, 423
Instruction Decoder. *See* Instr Decoder
Instruction encoding, x86, 369, 370*t*
Instruction formats, RISC-V, 332–340
  I-type, 334–336
  R-type, 332–334
  S/B-type, 336–338
  U/J-type, 338–340
Instruction formats, x86, 369
Instruction level parallelism (ILP), 476
Instruction memory, 395, 440–441, 466*b*–467*b*
Instruction register (IR), 416–417, 423–424
Instruction set, 299. *See also* Architecture
Instructions, RISC-V, 299
  branch instructions, 311
  branching, 311–313
  logical, 309
  multiply instructions, 310–311
  shift instructions, 309–310
Instructions, x86, 366–374
Instructions per cycle (IPC), 398, 472
Integrated circuits (ICs), 543.e19
Intel. *See* x86
Intel processors, 366
Intel x86. *See* x86
Interrupts, 356, 542.e39–43
Invalid logic level, 184
Inverters, 18, 117, 176*b*. *See also* NOT gate
  cross-coupled, 107–108, 108*f*
  in HDL, 176*b*, 196–197
Investigation of the Laws of Thought, An (Boole), 6
Involution theorem, 60
IOEs. *See* Input/output elements (IOEs)
IPC. *See* Instructions per cycle (IPC)
IR. *See* Instruction register (IR)
IRWrite, 416–417, 423–424

J

`j` instruction, 312, 313*b*
J-type instruction format. *See* U/J-type instruction format
`jal` instruction, 320–321
`jalr` instruction, 321
Java, 308. *See also* Language
`jr` instruction, 320–321
Jump instruction. *See* `j` instruction

K

K-maps. *See* Karnaugh maps (K-maps)
Karnaugh, Maurice, 73–74
Karnaugh maps (K-maps), 73–81, 92
  with "don't cares", 79–80
  logic minimization using, 73–81
  prime implicants, 75–79
  seven-segment display decoder, 77*b*–79*b*
KB. *See* Kilobyte (KB)
Kib. *See* Kibibit (Kib/Kibit)
KiB. *See* Kibibyte (KiB)
Kibibit (Kib/Kibit), 12
Kibibyte (KiB), 12
Kilobit (Kb/Kbit), 12
Kilobyte (KB), 12

L

LAB. *See* Logic array block (LAB)
Language. *See* Instructions
  assembly, 300–308
  machine, 332–344
  mnemonic, 301
Last-in-first-out (LIFO) queue, 322. *See also* Stack
Latches, 109–111
  comparison with flip-flops, 107–108, 116–117
  D, 111–112
  SR, 109–111
  transistor-level, 114–116
Latency, 155, 439, 448
Lattice, silicon, 25*f*, 31
LCDs. *See* Liquid crystal displays (LCDs)
Leaf function, 326
Leakage current, 33–34
Least recently used (LRU) replacement, 515
  two-way associative cache with, 515–516, 515*f*
Least significant bit (lsb), 11, 12*f*
Least significant byte (LSB), 11, 12*f*, 306–307
LEDs. *See* Light-emitting diodes (LEDs)
LEs. *See* Logic elements (LEs)
Level-sensitive latch. *See* Latches
LIFO. *See* Last-in-first-out (LIFO) queue
Light-emitting diodes (LEDs), 542.e7, 543.e18–19
Line options, compiler and command, 348–350, 545.e43–44
Linked list, 545.e33
Linker, 344, 353

Linking, 344
Linux, 545.e4
Liquid crystal displays (LCDs), 542.e43–44
Literal, 56, 94
  loading, 344–355
Little-endian bus order in HDL, 176
Little-endian memory addressing, 356
Load word (`lw`), 417–419
Loading literals, 345
Loads. *See* Load word (`lw`)
Local variables, 330
Locality, 500
Logic
  bubble pushing, 69–71
  combinational. *See* Combinational logic
  families, 23, 543.e10, 543.e15–16
  gates. *See* Gates
  hardware reduction, 68–69
  multilevel. *See* Multilevel combinational logic
  programmable, 543.e2–9
  sequential. *See* Sequential logic
  transistor-level. *See* Transistors
  two-level, 67–68
Logic array block (LAB), 279
Logic arrays, 275–284. *See also* Field programmable gate arrays (FPGAs)
  transistor-level implementation, 283–284
Logic elements (LEs), 278
  of Cyclone IV, 278*f*, 279
  functions built using, 280*b*–281*b*
Logic families, 23, 543.e10, 543.e15–16
  compatibility of, 24*t*
  logic levels of, 23
  specifications, 543.e17*t*
Logic gates, 17–20, 543.e1
  AND. *See* AND gate
  AND-OR (AO) gate, 45
  with delays in HDL, 187*b*
  multiple-input gates, 19–20
  NAND. *See* NAND gate
  NOR. *See* NOR gate
  NOT. *See* NOT gate
  OR. *See* OR gate
  OR-AND-INVERT (OAI) gate, 45
  XNOR. *See* XNOR gate
  XOR. *See* XOR gate
Logic levels, 20–21
Logic simulation, 173–174
Logic synthesis, 174–175
Logical instructions, 309
Logical shifter, 252
Lookup tables (LUTs), 82–83, 273
Loops, 315–317, 545.e19–20
  in C
    for, 545.e20
    do/while, 545.e19
    while, 545.e19
  in RISC-V assembly
    for, 315–316
    while, 315
Lovelace, Ada, 343*b*
LOW, 6
Low Voltage CMOS Logic (LVCMOS), 23
Low Voltage TTL Logic (LVTTL), 23
lsb. *See* Least significant bit (lsb)
LSB. *See* Least significant byte (LSB)
LUTs. *See* Lookup tables (LUTs)
LVCMOS. *See* Low Voltage CMOS Logic (LVCMOS)
LVTTL. *See* Low Voltage TTL Logic (LVTTL)
`lw` instruction, 417–419

M

MAC. *See* Multiply-accumulate (MAC)
Machine code. *See* Machine language
Machine language, 332–344
  addressing modes, 341–342
  interpreting, 342–343
  I-type instructions, 334–336
  R-type instructions, 332–334
  S/B-type instructions, 336–338
  stored program, 343–344, 343*f*
  translating to assembly language, 342*b*–343*b*
  U/J-type instructions, 338–340
Magnitude comparator, 245
Main Decoder, 408, 411*t*
Main FSM, 422–432, 436*f*
`main` function in C, 545.e3
Main memory, 501–503
`malloc` function, 545.e40
Mantissa, 258
Masuoka, Fujio, 272*b*
`math.h`, C library, 545.e35*t*, 545.e42
Max-delay constraint. *See* Setup time constraint
Maxterms, 56
Mb. *See* Megabit (Mb/Mbit), Mebibit (Mib/Mibit)
Mbit. *See* Megabit (Mb/Mbit), Mebibit (Mib/Mibit)
MCUs. *See* Microcontroller units (MCUs)
Mealy machines, 121, 121*f*, 130–132
  state transition and output table, 132*t*
  state transition diagrams, 131*f*
  timing diagrams for, 133*f*
Mean time between failure (MTBF), 151
Mebibit (Mib/Mibit), 12
Medium-scale integration (MSI) chips, 543.e2
Megabit (Mb/Mbit), 12. *See* also Mebibit (Mib/Mibit)
Memory, 317. *See also* Memory arrays
  access time, 503–505
  addressing modes, 306–308, 368*t*
  area and delay, 269–270
  big-endian, 355
  byte-addressable, 306-307, 307*b*, 355
  bytes and characters, 317–320
  HDL for, 313*b*, 317*f*, 466*b*–467*b*
  hierarchy, 502*f*
  little-endian, 356
  logic using, 272–273
  main, 502
  operands in, 306, 310–311
  physical, 521
  ports, 267
  protection, 527. *See also* Virtual memory
  types, 267–268
    DDR, 269–270
    DRAM, 269–270
    flash, 272
    register file, 270–271
    ROM, 271–272
    SRAM, 269–270
  virtual, 503. *See also* Virtual memory
Memory address computation, 307–308, 401–402, 425*f*
  data flow during, 402*f*, 425*f*
Memory arrays, 265–275. *See also* Memory
  bit cell, 267
  HDL for, 273*b*, 274*b*, 275*f*, 466*b*–467*b*
  logic using, 272–273
  organization, 267

- Memory hierarchy, 503
- Memory instructions, 306–308, 317–320, 334–336, 400–402
  - encodings, 334–335
- Memory interface, 499
- Memory map, RISC-V, 344–346, 542.e2*f*
- Memory performance. *See* Average Memory Access Time (AMAT)
- Memory protection, 527
- Memory systems, 499
  - performance analysis, 503–505
- Memory-mapped I/O, 542.e1–3, 542.e7
  - address decoder, 542.e1–3, 542.e2
  - communicating with I/O devices, 542.e2*b*–3*b*
  - hardware, 542.e2*f*, 542.e3
- *MemWrite*, 403, 404*f*
- Metal-oxide-semiconductor field effect transistors (MOSFETs), 24
  - switch models of, 28*f*
- Metastability, 149–150
  - metastable state, 108, 149
  - resolution time, 149–150, 152–155
  - synchronizers, 150–152
- Mib. *See* Mebibit (Mib/Mibit)
- Mibit. *See* Mebibit (Mib/Mibit)
- Micro-operations (micro-ops), 469–470
- Microarchitecture, 300, 393, 396. *See also* Architecture
  - advanced. *See* Advanced microarchitecture
  - architectural state. *See* Architectural state
  - description of, 393–394
  - design process, 394–396
  - evolution of RISC-V, 482–485
  - HDL representation, 456–468
    - generic building blocks, 461–464
    - single-cycle processor, 457
    - testbench, 464–468
  - multicycle processor. *See* Multicycle RISC-V processor
  - performance analysis, 397–398. *See also* Performance analysis
  - pipelined processor. *See* Pipelined RISC-V processor
  - real-world perspective, 482–485
  - single-cycle processor. *See* Single-cycle RISC-V processor
- Microcontroller, 542.e3, 542.e31
- Microcontroller peripherals, 542.e43
  - Bluetooth wireless communication, 542.e53–54
  - character LCD, 542.e44–47
    - control, 542.e47*b*
    - parallel interface, 542.e44, 542.e45*f*
  - motor control, 542.e54–64
  - VGA monitor, 542.e45–52
- Microcontroller units (MCUs), 542.e3
- Micro-operations, 469–470
  - designers, 468
  - high-performance, 468
- Microprocessors, 1, 11*b*, 299, 393
  - architectural state of, 395
- Millions of instructions per second, 439
- Min-delay constraint. *See* Hold time constraint
- Minterms, 56
- Miss, 502–505, 518
  - capacity, 518
  - compulsory, 518
  - conflict, 511
- Miss penalty, 513
- Miss rate, 503–505
  - and access times, 504*t*
- Misses
  - cache, 502
    - capacity, 518
    - compulsory, 518
    - conflict, 518
  - page fault, 521–522
- Mnemonics, 301, 301*b*
- `ModR/M` byte, 371–372
- Modularity, 4
- Modules, in HDL
  - behavioral and structural, 171–172
  - parameterized modules, 215–218
- Moore, Gordon, 28*b*
- Moore machines, 121–132
  - state transition and output table, 131*t*
  - state transition diagrams, 131*f*
  - timing diagrams for, 133*f*
- Moore's law, 28*b*
- MOS transistors. *See* Metal-oxide-semiconductor field effect transistors (MOSFETs)
- MOSFET. *See* Metal-oxide-semiconductor field effect transistors (MOSFETs)
- MOSFETs. *See* Metal-oxide-semiconductor field effect transistors (MOSFETs)
- Most significant bit (msb), 11, 12*f*
- Most significant byte (MSB), 11, 12*f*, 306–307
- Motors
  - DC, 542.e54–59
  - H-bridge, 542.e55–58
  - servo, 542.e54, 542.e58–60
  - stepper, 542.e54, 542.e60–64
- msb. *See* Most significant bit (msb)
- MSB. *See* Most significant byte (MSB)
- MSI chips. *See* Medium-scale integration (MSI) chips
- MTBF. *See* Mean time between failure (MTBF)
- Multicycle processor, 415–438
  - control, 422–432, 436*f*
  - datapath, 416–422
    - `beq` instruction, 421–422
    - I-type ALU instructions,
    - `lw` instruction, 417–419
    - R-type instructions, 421–422
    - `sw` instruction, 420–421
  - instructions, 410–412
  - performance, 435–438
- Multicycle microarchitectures, 396
- Multilevel combinational logic, 67–71. *See also* Logic
- Multilevel page tables, 529–530
- Multiple-output circuit, 66*b*
- Multiplexers, 81–84
  - definition of, 81
  - HDL for
    - behavioral model of, 179*b*
    - parameterized *N*-bit, 216*b*
    - structural model of, 188*b*
  - logic using, 82–84
  - symbol and truth table, 81*f*
- Multiplicand, 253
- Multiplication. *See* Multiplier
- Multiplier, 253
  - HDL for, 254
- Multiply instructions, 310–311
- Multiprocessors, 479–482
  - chip, 479
  - heterogeneous, 479
  - homogeneous, 479
- Multithreaded processor, 479
- Multithreading, 478–479
- Mux. *See* Multiplexers

N

NaN. *See* Not a number (NaN)
NAND (7400), 543.e3*f*
NAND gate, 19
  CMOS, 29–30
Nested if/else statement, 314–315, 545.e18
Nibbles, 11–12
nMOS transistors, 26–29, 27*b*
Noise margins, 21–22, 21*f*
  calculating, 21*b*–22*b*
Nonarchitectural state, 393–394, 396, 416
Nonblocking and blocking assignments, 197, 203–207
Nonleaf function calls, 326–327
Nonpreserved registers, 324, 327–329
`nop`, 331–332, 444–445
NOR gate, 19–20, 61, 543.e3*f*
  chip (7402), 543.e3*f*
  CMOS, 30*b*
  pseudo-nMOS logic, 31
  truth table, 20*f*
Not a number (NaN), 259
NOT gate, 18
  chip (7404), 543.e3*f*
  CMOS, 29
Noyce, Robert, 24*b*
Null element theorem, 59–60
Number conversion
  binary to decimal, 8*b*
  binary to hexadecimal, 9–11
  decimal to binary, 9*b*, 11*b*
  decimal to hexadecimal, 11*b*
  hexadecimal to binary and decimal, 9*b*, 10*b*
  taking the two's complement, 14–16
Number systems, 7–17
  binary, 7–17
  comparison of, 16–17, 17*t*
  estimating powers of two, 12*b*
  fixed-point, 256–257, 256*f*
  floating-point, 257–261
    addition, 260*f*, 261
    special cases, 259
  hexadecimal, 9–11, 10*t*
  negative and positive, 13–14
  sign/magnitude, 13–14
  signed, 13–17
  two's complement, 14–16
  unsigned, 7–13

O

Odds and ends, 355–363
  endianness, 355–356
  exceptions, 356–360
  nop, 331–332
OFF, 24, 28
Offset, 307, 400–401, 418
ON, 24, 28
One-bit dynamic branch predictor, 471
One-cold encoding, 128
One-hot encoding, 127–128
One-time programmable (OTP), 543.e2
op field, 333
Opcode. *See* **op** field
Operands
  RISC-V, 302
    constants/immediates, 304–306
    memory, 306–308
    register set, 304
    registers, 303–304
  x86, 367–369
Operation code. *See* op field
Operators
  in C, 316
  in HDL, 176*b*
    bitwise, 175–176
    precedence, 183*b*
    reduction, 178
    table of, 183*b*
    ternary, 179*b*
OR gate, 19
OR-AND-INVERT (OAI) gate, 194
`or` instruction, 309
OTP. *See* One-time programmable (OTP)
Out-of-order execution, 477*f*
Out-of-order processor, 473–476
Output dependence, 476
Overflow
  with addition, 13
  detection, 248–249
Oxide, 26–27

P

Packages, chips, 543.e19–20
Page fault, 521–522
Page number, 523
Page offset, 523
Page table, 522–523
Pages, 521–522
Paging, 528
Parallel I/O, 542.e14–15
Parallelism, 155–158
Parity gate. *See* XOR gate
Partial products, 253
Pass by reference, 545.e22
Pass by value, 545.e22
Pass gate. *See* Transmission gates
PC. *See* Program counter (PC)
PC Logic, 415
PC-relative addressing, 341–342
PCBs. *See* Printed circuit boards (PCBs)
PCI. *See* Peripheral Component Interconnect (PCI)
PCSrc, 406–407, 409, 411
PCWrite, 419
Perfect induction, proving theorems using, 62*b*
Performance analysis, 397–398. *See also* Average Memory Access Time (AMAT)
  multicycle processor, 436–437
  pipelined processor, 439–456
  processor comparison, 438*b*
  single-cycle processor, 412–414
Peripherals, 542
Peripherals devices. *See* Input/output (I/O) systems
Phase locked loop (PLL), 542.e50–53
Physical memory, 503
Physical page number (PPN), 523
Physical pages, 522
Pipelined processor, 439–456
  abstract view of, 441*f*
  control unit, 443, 453
  datapath, 441–443
  description, 439–456
  hazards, 443–453
  performance analysis, 454–456
  throughput, 439–440
Pipelined microarchitecture. *See* Pipelined processor
Pipelining, 156–157. *See also* Pipelined processor
PLAs. *See* Programmable logic arrays (PLAs)
Plastic leaded chip carriers (PLCCs), 543.e19–20
Platters, 520–521

PLCCs. *See* Plastic leaded chip carriers (PLCCs)
PLDs. *See* Programmable logic devices (PLDs)
PLL. *See* Phase locked loop (PLL)
pMOS transistors, 26–29, 27*f*
Pointers, 542.e9, 545.e21–22, 545.e33
POS. *See* Product-of-sums (POS) form
Positive edge-triggered flip-flop, 112
Power consumption, 32–34
PPN. *See* Physical page number (PPN)
Prefix adders, 241–243
Prefix tree, 243
Preserved registers, 324–326, 325*t*
Prime implicants, 63–81
Printed circuit boards (PCBs), 543.e20–23
`printf`, 545.e36–37
Priority
  circuit, 66–67
  encoder, 100–101, 103
Procedure calls. *See* Function calls
Processor performance comparison, 438*b*, 455*b*
  multicycle processor, 436–437
  pipelined processor, 454–456
  single-cycle processor, 415*b*
Processor-memory gap, 502
Product-of-sums (POS) form, 58
Program counter (PC), 308, 343–344, 402
Programmable logic arrays (PLAs), 275–276, 543.e2, 543.e6–7
  transistor-level implementation, 283–284
Programmable logic devices (PLDs), 543.e6–7
Programmable read only memories (PROMs), 271, 543.e2–6
Programming
  arrays. *See* Arrays
  branching. *See* Branching
  in C. *See* C programming
  conditional statements, 313–315
  constants. *See* Constants; Immediates
  function calls. *See* Function calls
  getting loopy. *See* Loops
  logical and arithmetic instructions, 309
  loops. *See* Loops
  memory, 317–320
  in RISC-V, 308
  shift instructions, 309–310, 333–335, 335*f*
PROMs. *See* Programmable read only memories (PROMs)
Propagate signal, 239
Propagation delay, 86–90. *See also* Critical path
Pseudo-nMOS logic, 31–32, 31*f*
  NOR gate, 31*f*
  ROMs and PLAs, 283–284
Pseudoinstructions, 330–332, 358*b*
Pulse-Width Modulation (PWM), 542
  analog output with, 542.e31
  duty cycle, 542.e35
  signal, 542.e35*f*
PWM. *See* Pulse-Width Modulation (PWM)

Q

Quiescent supply current, 33–34

R

R-type instruction, 35–37
Race conditions, 117*b*–118*b,* 118*f*
RAM. *See* Random access memory (RAM)
`rand`, 545.e40–41
Random access memory (RAM), 267–269, 272–273
RAW hazard. *See* Read after write (RAW) hazard
rd field. *See* Destination register (**rd**)
Read after write (RAW) hazard, 444, 475. *See also* Hazards
Read only memory (ROM), 267, 269–271
  transistor-level implementation, 284
Read/write head, 520–521
ReadData bus, 402, 403*f*
Receiver gate, 20–21
Recursive function calls, 327–329
Reduced instruction set computer (RISC) architecture, 302, 469–470
Reduction operators, 178
Register file (RF)
  HDL for, 461–464
  in pipelined RISC-V processor (write on falling edge), 441–442
  RISC-V register descriptions, 303
  schematic, 270–271
Register renaming, 476–478
Register set, 304. *See also* Register file (RF)
R-type (register-type) instruction, 332–334
Registers. *See* RISC-V registers; Flip-flops; x86 registers
  preserved and nonpreserved, 324–326
Regularity, 4
RegWrite, 402
Replacement policies, 527–529
Resettable flip-flops, 114
Resettable registers, 192–193
Resolution time, 149–150. *See also* Metastability
  derivation of, 152–155
Return value, 320
RF. *See* Register file (RF)
Ring oscillator, 313*b,* 335*f*
Ripple-carry adder, 238–239, 241*b*
RISC architecture. *See* Reduced instruction set computer (RISC) architecture
RISC-V architecture, evolution of, 363–366
  comparison of RISC-V and ARM architectures, 365–366
  comparison of RISC-V and MIPS architectures, 365
RISC-V Base Instruction Sets and Extensions, 364–365
RISC-V instructions, 299
  addressing modes, 341–342
  branch instructions, 311–313
  immediate encodings, 340–341
  interpreting, 342–343
  logical instructions, 309
  multiply instructions, 310–311
  shift instructions, 309–310
  formats
    I-type instructions, 334–336
    R-type instructions, 332–334
    S/B-type instructions, 336–338
    U/J-type instructions, 338–340
  instruction set, 302
  memory instructions, 306–308
  miscellaneous instructions, 355–356
  pseudoinstructions, 330–332
  stored program, 343–344
RISC-V microprocessor, 393–467
  data memory, 395
  instruction memory, 395

multicycle, 396, 415–438
pipelined, 396, 439–456
program counter, 394
register file, 395
single-cycle, 396, 398–415
state elements of, 393–394
RISC-V processors. *See* RISC-V microprocessors
RISC-V registers, 303–304
program counter, 341, 343–344, 394
register file, 394–395
register set, 304–305
RISC-V single-cycle HDL, 456–467
building blocks, 461–463
controller, 458*b*–460*b*
datapath, 460*b*–461*b*
testbench, 464–468
Rising edge, 86
ROM. *See* Read only memory (ROM)
Rotations per minute (RPM), 542.e55
Rotators, 251–253
Rounding modes, 260–261
RPM. *See* Rotations per minute (RPM)
rs1 field. *See* Source register 1 (**rs1**)
rs2 field. *See* Source register 2 (**rs2**)
RS-232, 542.e25
RV32IMAC instruction set, 484–486

S

S-type instruction format. *See* S/B-type instruction format
S/B-type instruction format, 336–338
Sampling, 139
Sampling rate, 542.e31
SATA. *See* Serial ATA (SATA)
Scalar processor, 472–473
Scan chains, 265
`scanf`, 545.e37
Scannable flip-flop, 265
Schematics, rules of drawing, 29, 65
SCK. *See* Serial Clock (SCK)
SDI. *See* Serial Data In (SDI)
SDO. *See* Serial Data Out (SDO)
SDRAM. *See* Synchronous dynamic random access memory (SDRAM)
Segment descriptor, 372
Segmentation, 373
Selected signal assignment statements, 180*b*
Semiconductors, 25
industry, sales, 1
Sequencing overhead, 141–142, 147–149, 157–158, 455–456
Sequential building blocks. *See* Sequential logic
Sequential logic, 107, 261–265
counters, 261–263
finite state machines. *See* Finite state machines (FSMs)
flip-flops, 112–113. *See also* Registers
latches, 107–117
D, 111–112
SR, 109–111
registers. *See* Registers
shift registers, 263–265
timing of. *See* Timing analysis
Serial Clock (SCK), 542.e15
Serial communication, with PC, 542.–e27
Serial Data In (SDI), 542.e15
Serial Data Out (SDO), 542.e15
Serial I/O, 542.e14–29
SPI. *See* Serial peripheral interface (SPI)
UART. *See* Universal Asynchronous Receiver Transmitter (UART)
Serial Peripheral Interface (SPI), 542.e4, 542.e15–22
connection between FE310 controller and FPGA, 542.e19–20
ports
Serial Clock (SCK), 542.e15
Serial Data In (SDI), 542.e15
Serial Data Out (SDO), 542.e15
waveforms, 542.e15
Servo motor, 542.e60*b,* 542.e54, 542.e58–60
Set bits, 508
Setup time constraint, 141–142
with clock skew, 146–149
Seven-segment display decoder, 77*b*–79*b*
with don't cares, 80*b*
HDL for, 199*b*
Shaft encoder, 542.e58*f,* 542.e54, 542.e58–59
Shift instructions, 309–310, 310*f*
Shift registers, 263–265
Shifters, 251–253
Short path, 87–90
Sign bit, 13–14
Sign extension, 16
Sign/magnitude numbers, 13–14, 256–257
Signed binary numbers, 13–17
Signed instructions, 360–361
Signed multiplier, 215
Silicon dioxide (SiO_2), 26–27
Silicon lattice, 25
SIMD. *See* Single instruction multiple data (SIMD)
`simple` function, 320–321
Simple programmable logic devices (SPLDs), 277
Simulation waveforms, 174
with delays, 187*f*
Single instruction multiple data (SIMD), 473
Single-cycle processor, 398–415, 457
control, 407–410
controller, 458*b*
datapath, 399–407
`addi` instruction, 410–411
`beq` instruction, 421–422
`jal` instruction, 411–412
`lw` instruction, 400–402
R-type instructions, 404–405
`sw` instruction, 403–404
decoders, 459*b*
HDL for, 457–467
instructions, 410–412
performance, 412–415
Single-cycle microarchitecture. *See* Single-cycle processor
Single-pole double-throw switches (SPDT switches), 543.e17–18
Single-pole single-throw switches (SPST switches), 543.e17–18
Single-precision formats, 259–260. *See also* Floating-point numbers
SiO_2. *See* Silicon dioxide (SiO_2)
Skew. *See* Clock skew
Slash notation, 54*f*
Small-scale integration (SSI) chips, 543.e1
SMP. *See* Symmetric multiprocessing (SMP)
Solid state drive (SSD), 502. *See also* Flash memory; Hard drive
SOP. *See* Sum-of-products (SOP) form
Source register 1 (**rs1**), 332*f*, 333
Source register 2 (**rs2**), 332*f*, 333
`sp`. *See* Stack pointer (`sp`)
SparkFun's RED-V RedBoard, 542
Spatial locality, 500, 514*b*
Spatial parallelism, 155–156

SPDT switches. *See* Single-pole double-throw switches (SPDT switches)
SPEC, 397
SPECINT2000, 438
SPI. *See* Serial Peripheral Interface (SPI)
SPLDs. *See* Simple programmable logic devices (SPLDs)
SPST switches. *See* Single-pole single-throw switches (SPST switches)
Squashing, 476
SR latches, 109–111, 110*f*
SRAM. *See* Static random access memory (SRAM)
`srand`, 545.e40–41
SSD. *See* Solid state drive (SSD)
SSI chips. *See* Small-scale integration (SSI) chips
Stack, 322–324. *See also* Function calls
  preserved registers, 324–326
  during recursive function call, 327–329
  stack frame, 324–325
  stack pointer (SP), 329
  storing additional arguments on, 330
  storing local variables on, 330
Stalls, 448–450. *See also* Hazards
Standard libraries, 545.e35–43
  file manipulation, 545.e38–39
    `printf`, 545.e36–37
    `scanf`, 545.e37
  math, 545.e42
  stdio, 545.e35–40
  stdlib, 545.e40–42
    `exit`, 545.e41
    format conversion (`atoi`, `atol`, `atof`), 545.e41
    `rand`, `srand`, 545.e40–41
  string, 545.e42–43
State encodings, FSM, 127–128, 132–135. *See also* Binary encoding; One-cold encoding; One-hot encoding
State machine circuit. *See* Finite state machines (FSMs)
State variables, 107
Static branch prediction, 470–472
Static discipline, 22–24
Static power, 32
Static random access memory (SRAM), 268–269
Status flags, 369. *See also* Condition flags
`stdio.h`, C library, 545.e35. *See also* Standard libraries
`stdlib.h`, C library, 545.e40. *See also* Standard libraries
Stepper motors, 542.e54, 542.e60–64
  bipolar stepper motor, 542.e60–64,
    half-step drive, 542.e60–61
    two-phase-on drive, 542.e60–61
    wave drive, 542.e62–63
Store/branch instructions (S/B-type instructions). *See* S/B-type instruction format
Stored program, 343–344
`string.h`, C library, 545.e42–43
Strings, 319, 545.e27–28. *See also* Characters (`char`)
`struct`. *See* Structures (`struct`)
Structural modeling, 171–172, 188–190
Structures (`struct`), 545.e29–30
`sub` instruction, 301
Substrate, 26–27
Subtraction, 15, 244–245
Subtractor, 244–245
Sum-of-products (SOP) form, 56–58
Superscalar processor, 472–473
Supply voltage, 56–58. *See also* VDD
`sw` instruction, 307–308
Swap space, 528
Switch/case statements
  in C, 545.e18
  in HDL. *See* case statement, in HDL
  in RISC-V assembly, 314–315
Switches, 543.e17–18
Symbol table, 351–353
Symmetric multiprocessing (SMP), 479–480. *See also* Homogeneous multiprocessors
Synchronizers, 150–152
Synchronous circuits, 120–121
Synchronous dynamic random access memory (SDRAM), 269–270
  DDR, 269–270
Synchronous logic, design, 117–121
Synchronous resettable flip-flops, 114
Synchronous sequential circuits, 118–120, 120*b*. *See also* Finite state machines (FSMs)
  timing specification. *See* Timing analysis
SystemVerilog, 171–223. *See also* Hardware description languages (HDLs)
  accessing parts of busses, 186*b*, 190*b*
  bad synchronizer with blocking assignments, 207*b*
  bit swizzling, 186
  blocking and nonblocking assignment, 197, 203–207
  case statements, 199–200, 203*b*
  combinational logic using, 175–188, 196–207, 215–218
  comments, 178
  conditional assignment, 179–180
  data types, 211–215
  decoders, 200*b*, 217
  delays (in simulation only), 187*b*
  divide-by-3 FSM, 208*b*
  finite state machines (FSMs), 207–209
    Mealy FSM, 211*b*
    Moore FSM, 208–210
  full adder, 182*b*
  history of, 173*b*
  if statements, 200–203
  internal signals, 180–182
  inverters, 176*b*, 197*b*
  latches, 196
  logic gates, 175–188
  multiplexers, 179*b*, 188*b*, 216*b*
  multiplier, 215
  numbers, 183
  operators, 182–183
  parameterized modules, 215–218
    N-bit multiplexers, 216*b*
    N-input AND gate, 218*b*
    N:2^N decoder, 217*b*
  priority circuit, 202*b*
    using don’t cares, 203*b*
  reduction operators, 178
  registers, 191
    enabled, 194
    resettable, 192–193
  sequential logic using, 191–196, 207–209
  seven-segment display decoder, 77*b*–79*b*
  simulation and synthesis, 173–175
  structural models, 188–190
  synchronizer, 195*b*
  testbench, 218–222
    self-checking, 220*b*
    simple, 219*b*
    with test vector file, 221*b*–222*b*
  tristate buffer, 185*b*
  truth tables with undefined and floating inputs, 185*b*, 186*t*
  using always/process, 198*b*
  using nonblocking assignments, 206
  z’s and x’s, 184–186, 203*b*

T

Tag, 508
Taking the two's complement, 15
Temporal locality, 500, 505, 509
Temporal parallelism, 155–156
Temporary registers, 303*b*
Ternary operators, 179*b*, 545.e13
Testbench, 464–468
Testbenches, HDLs, 218
  self-checking, 219
  simple, 219
  with testvectors, 221*b*–222*b*
Text Segment, 345, 355
Thin small outline package (TSOP), 543.e19–20
Thread level parallelism (TLP), 478
Threshold voltage, 27–28
Throughput, 155, 396, 439, 479
Timers, 542.e29–31
Timing
  of combinational logic, 86–87
    delay. *See* Contamination delay
    glitches. *See* Glitches
  of sequential logic, 139–155
    analysis. *See* Timing analysis
    clock skew. *See* Clock skew
    dynamic discipline, 139
    metastability. *See* Metastability
    resolution time. *See* Resolution time
    system timing. *See* Timing analysis
Timing analysis, 143*b*–145*b*
  calculating cycle time. *See* Setup time constraint
  with clock skew. *See* Clock skew
  hold time constraint. *See* Hold time constraint
  max-delay constraint. *See* Setup time constraint
  min-delay constraint. *See* Hold time constraint
  multicycle processor, 437*b*
  pipelined processor, 452
  setup time constraint. *See* Setup time constraint
  single-cycle processor, 415*b*
TLB. *See* Translation lookaside buffer (TLB)
TLP. *See* Thread level parallelism (TLP)
Transistor-Transistor Logic (TTL), 23, 543.e15
Transistors, 24–32
  bipolar, 24
  CMOS, 24–32
  gates made from, 29, 29–31, 30, 30, 31
  latches and flip-flops, 114–115
  MOSFETs, 24
  nMOS, 26–29
  pMOS, 26–29
    pseudo-nMOS, 31–32
    ROMs and PLAs, 283
    transmission gate, 31
Translating and starting a program, 344*f*
Translation lookaside buffer (TLB), 525–526, 526–527
Transmission gates, 31
Transmission lines, 543.e23–35
  characteristic impedance (Z_0), 543.e24, 543.e33
    derivation of, 543.e33–34
  matched termination, 543.e24–26
  mismatched termination, 543.e27–29
  open termination, 543.e26
  reflection coefficient (k_r), 543.e33–34
    derivation of, 543.e33–34
  series and parallel terminations, 543.e30–32
  short termination, 543.e27
  when to use, 543.e30
Transparent latch. *See* Latches: D
Traps, 356
Tristate buffer, 72, 185*b*
  HDL for, 184
  multiplexer built using, 82, 89*b*–90*b*
True, 6, 18, 56, 68, 72, 109–110, 114, 127–128, 178, 180, 203
Truth tables, 17–18
  ALU decoder, 408–409, 412*t*
  with don't cares, 66–67, 79–80, 203
  multiplexer, 81–84
  seven-segment display decoder, 77*b*–79*b*
  SR latch, 132–135
  with undefined and floating inputs, 185*b*
TSOP. *See* Thin small outline package (TSOP)
TTL. *See* Transistor-Transistor Logic (TTL)
Two-bit dynamic branch predictor, 471
Two-cycle latency of LDR, 448
Two-level logic, 67–68
Two's complement numbers, 14–16
`typedef`, 459*b*, 545.e31

U

U. *See* Use bit (*U*)
U-type instruction format. *See* U/J-type instruction format
U/J-type instruction format, 338–340
UART. *See* Universal Asynchronous Receiver Transmitter (UART)
Unconditional branches, 311
Undefined instruction exception, 356
Unicode, 317*b*
Unit under test (UUT), 218*b*
Unity gain points, 22
Universal Asynchronous Receiver Transmitter (UART), 542.e23–29
  hardware handshaking, 542.e25
Universal Serial Bus (USB), 275, 542.e15, 542.e24
Unsigned instructions, 360–361
Unsigned multiplier, 215*b*, 360
Unsigned numbers, 16
Upper immediate/jump instructions (U/J-Type instructions), 338–340
USB. *See* Universal Serial Bus (USB)
Use bit (*U*), 515–516
UUT. *See* Unit under test (UUT)

V

V. *See* Valid bit (*V*)
Valid bit (*V*), 509
Variables in C, 545.e7–11
  global and local, 545.e9–10
  initializing, 545.e11
  primitive data types, 545.e8–9
V_{CC}, 21*b*. *See also* Supply voltage; *VDD*
V_{DD}, 20–21, 30*b*. *See also* Supply voltage
Vector processor, 472*b*–473*b*
Verilog. *See* SystemVerilog
Very High Speed Integrated Circuits (VHSIC), 173*b*. *See also* VHDL (VHSIC Hardware Description Language)

VGA (Video Graphics Array) monitor, 542.e47–53
  connector pinout, 542.e49*f*
  driver for, 542.e50*b*–53*b*
VHDL, 171–223
  accessing parts of busses, 190*b*
  bad synchronizer with blocking assignments, 207*b*
  bit swizzling, 186*b*
  blocking and nonblocking assignment, 197, 203–207
  `case` statements, 199–200, 217
  combinational logic using, 175, 196–207, 220–222
  comments, 178
  conditional assignment, 179–180
  data types, 211–215
  decoders, 200
  delays (in simulation), 186–187
  divide-by-3 FSM, 208*b*
  finite state machines (FSMs), 393
    Mealy FSM, 211*b*
    Moore FSM, 209, 210*b*
  full adder, 182*b*
    using always/process, 198*b*
    using nonblocking assignments, 208
  history of, 173*b*
  if statements, 200–203
  internal signals, 182*b*
  inverters, 176, 196–197
  latches, 196
  logic gates, 177*b*
  multiplexer, 179*b*, 188–189, 216*b*
  multiplier, 215*b*
  N-bit multiplexers, 216*b*
  N-input AND gate, 218*b*
  N:2^N decoder, 217
  numbers, 183
  operators, 182–183
  parameterized modules, 215
  priority circuit, 203*b*
  reduction operators, 178
  using don't cares, 203*b*
  reduction operators, 178
  registers, 191
    enabled, 194
    resettable, 192–193
  sequential logic using, 191–196, 207
  seven-segment display decoder, 222
  simulation and synthesis, 173–175
  structural models, 188–190
  synchronizer, 195
  testbench, 218–222
    self-checking, 220*b*
    simple, 219
    with test vector file, 221*b*–222*b*
  tristate buffer, 185*b*
  truth tables with undefined and floating inputs, 185*b*, 203
  z's and x's, 184–186
VHSIC. *See* VHDL
VHSIC Hardware Description Language. *See* VHDL
Video Graphics Array (VGA). *See* VGA (Video Graphics Array) monitor
Virtual address, 521
  space, 527
Virtual memory, 503, 520–530
  address translation, 521–522
  cache terms comparison, 522
  memory protection, 527
  multilevel page tables, 529–530
  page fault, 521–522
  page number, 523
  page offset, 523
  page table, 524–525
  pages, 521–522
  replacement policies, 529–530
  translation lookaside buffer (TLB), 525–526
  write policy, 519–520
Virtual page number (VPN), 524
Virtual pages, 521–522
VPN. *See* Virtual page number (VPN)
V_{SS}, 21*b*

W

Wafers, 26
Wall, Larry, 18*b*
WAR hazard. *See* Write after read (WAR) hazard
WAW hazard. *See* Write after write (WAW) hazard
Weak pull-up, 31
Weird number, 16
While loops, 315, 545.e19
White space, 178
Whitmore, Georgiana, 5*b*
Wire, 65
Wireless communication, Bluetooth, 542.e53–54
Wordline, 267
Write after read (WAR) hazard, 475. *See also* Hazards
Write after write (WAW) hazard, 476
Write policy, 519–520
  write-back, 519
  write-through, 519

X

X. *See* Contention (x); Don't care (X)
x86
  architecture, 366–374
    big picture, 373–374
    branch conditions, 371*t*
    instruction encoding, 371–372
    instructions, 369
    memory addressing modes, 368*t*
    operands, 367–368
    peculiarities, 372–373
    registers, 366–367
    status flags, 369
Xilinx FPGA, 278
XNOR gate, 19*b*
XOR gate, 19

Z

Z. *See* Floating (Z)