Index

Page numbers in *italics* indicate figures, tables and text boxes.

0, 22. *See also* LOW, OFF
1, 22. *See also* HIGH, ON
32-bit datapath, 461
32-bit microprocessor, 454
4004 microprocessor chip, 458, *459*
74xx series logic, 583–587
  parts,
    2:1 mux (74157), *586*
    3:8 decoder (74138), *586*
    4:1 mux (74153), *586*
    AND (7408), *585*
    AND3 (7411), *585*
    AND4 (7421), *585*
    counter (74161, 74163), *586*
    FLOP (7474), 583, *585*
    NAND (7400), *585*
    NOR (7402), *585*
    NOT (7404), 583
    OR (7432), *585*
    register (74377), *586*
    tristate buffer (74244), *586*
    XOR (7486), *585*
80386 microprocessor chip, 459, *460*
80486 microprocessor chip, 460, *461*
`#define`, 627–628
`#include`, 628–629. *See also* Standard libraries

A

Abstraction, 4–5
  digital. *See* Digital abstraction
Accumulator, 353
Acquisition time. *See* Sampling time
Active low, 74–75
A/D conversion, 531–533
  registers in, 532
ADCs. *See* Analog/digital converters
`add`, 297
Adders, 239–246
  carry-lookahead, 241
  carry propagate, 240
  full, *56*, 240
  half, 240
  HDL for, 184, 200
  prefix, 243
  ripple-carry, 240
`addi`, 304
`addiu`, 345
`addu`, 345
Addition, 14–15, 17–18, 235, 239–246, *297*. *See also* Adders
  binary, 14–15
  floating point, 258–259
  MIPS instructions, 344–345, 622
  signed binary, 15–17
Address. *See also* Memory
  physical, 497–501
  translation, 497–500
  virtual, 497. *See also* Virtual memory
Addressing modes
  MIPS, 333–335
    base, 333
    immediate, 333
    PC-relative, 333–334
    pseudo-direct, 334–335
    register-only, 333
  x86, 349
Advanced Micro Devices (AMD), 296, 375, 457, 460
Advanced microarchitecture, 444–458
  branch prediction. *See* Branch prediction
  deep pipelines. *See* Deep pipelines
  heterogeneous multiprocessors. *See* Heterogeneous multiprocessors
  homogeneous multiprocessors. *See* Homogeneous 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
  superscalar processor. *See* Superscalar processor
Altera FPGA, 274–279
ALU. *See* Arithmetic/logical unit
ALU decoder, 382–384
  HDL for, 432
ALUControl, 378, 384
ALUOp, 382–384
ALUResult, 378
ALUSrc, 384
ALUSrcA, 397
ALUSrcB, 397
AMAT. *See* Average memory access time
AMD. *See* Advanced Micro Devices
Amdahl, Gene, 480

Amdahl's Law, 480
American Standard Code for Information Interchange (ASCII), 322, *323*, 630, 649–650
Analog I/O, 531–537
    A/D conversion, 532–533
    D/A conversion, 533–537
    Pulse-width modulation (PWM), 536–537
Analog-to-digital converters (ADCs), 531–533
Analytical engine, 7, *8*
AND gate, 20–22, 179
    chips (7408, 7411, 7421), *585*
    truth table, *20*, *22*
    using CMOS transistors, 32–33
and, 311
andi, 311–312
AND-OR (AO) gate, 46
Anode, 27
Application-specific integrated circuits (ASICs), 591
Architectural state, 310, 371–372
Architecture, 295–356, 619–622
    MIPS
        addressing modes, 333–335
        assembly language, 296–304, 619-622
        instructions, 619–622
        machine language, 305–310
        operands, 298–304
    x86, 347–356
Arguments, 326, 332–333, 637
    pass by reference, 644
    pass by value, 644
Arithmetic
    C operators, 633–635
    circuits, 239–254
    HDL operators, 185
    MIPS instructions, 310–314
    packed, 454
Arithmetic/logical unit (ALU), 248–250, 378
    implementation of, *249*
    in MIPS processor, 382–385. *See also* ALUControl, ALUOp
Arrays, 320–324, 645–651
    accessing, 320–322, 645
    as input argument, 646–647
    bytes and characters, 322–324, 649–651
    comparison or assignment of, 650
    declaration, 645
    indexing, 320–322, 645–649
    initialization, 645–646
    multi-dimension, 648–649
ASCII. *See* American Standard Code for Information Interchange
ASICs. *See* Application-specific integrated circuits
Assembler, 338–339, 666
Assembler directives, 338
Assembler temporary register ($at), 342
Assembly language, MIPS, 295–356, 619–622. *See also* MIPS instructions
    instructions, 296–304, 619–622
    logical instructions, 311–312
    operands, 298–304
    translating high-level code into, 300
    translating machine language to, 309
    translating to machine language, 306–307
Assembly language, x86. *See* x86 instructions
Associativity
    in Boolean algebra, 62, *63*
    in caches, 481, 486–488
Astable circuits, 119
Asymmetric multiprocessors. *See* Heterogeneous multiprocessors
Asynchronous circuits, 120–123
Asynchronous resettable flip-flops
        definition, 116
    HDL, 194–196
Asynchronous serial link, *522*. *See also* Universal Asynchronous Receiver Transmitter (UART)
AT Attachment (ATA), 562
Average memory access time (AMAT), 479, 492

B

Babbage, Charles, *7*
Base address, 301–302, 307, 320–322, 324
Base addressing, 333
Baud rate register (BRG), 518
BCD. *See* Binary coded decimal
Behavioral modeling, 173–174
Benchmarks, 375
beq, 314–315
Biased exponent, 257
Big-endian memory, 302–303
Big-endian order, 178
Binary addition, 14–15. *See also* Adders, Addition
Binary coded decimal (BCD), 258
Binary encoding, 125–126, 129–131
    for divide-by-3 counter, 129–131
    for traffic light FSM, 125–126
Binary numbers
    signed, 15–19
    unsigned, 9–11
Binary to decimal conversion, 10, *10–11*
Binary to hexadecimal conversion, 12
Bipolar junction transistors, 26
Bipolar motor drive, *555*
Bipolar signaling, 524
Bipolar stepper motor, 554, *554–555*
    AIRPAX LB82773-M1, 554, *555*
    direct drive current, *556*
Bistable element, 109
Bit, 8
    dirty, 494
    least significant, 13, *14*
    most significant, 13, *14*
    sign, 16
    use, 490
    valid, 484
Bit cells, 264–269
    DRAM, 266–267
    ROM, 268–269
    SRAM, 267
Bitline, 264
Bit swizzling, 188
Bitwise operators, 177–179
Block, 481
Block offset, 488–489
Block size (*b*), 481, 488–489
Blocking and nonblocking assignments, 199–200, 205–209
BlueSMiRF silver module, 548, *548*
Bluetooth wireless communication, 547–548
    BlueSMiRF silver module, 548
    classes, *547*
    PIC32 to PC link, *548*
bne, 314–315
Boole, George, *8*
Boolean algebra, 60–66
    axioms, 61

  - equation simplification, 65–66
  - theorems, 61–64
- Boolean equations, 58–60
  - product-of-sums (POS) form, 60
  - sum-of-products (SOP) form, 58–60
- Boolean logic, 8. *See also* Boolean algebra, Logic gates
- Boolean theorems, 61–64
  - associativity, 63
  - combining, 62
  - complements, 62
  - consensus, 62, 64
  - covering, 62
  - De Morgan's, 63–64
  - distributivity, 63
  - idempotency, 62
  - identity, 62
  - involution, 62
  - null element, 62
- *Branch*, 384
- Branch equal (`beq`)
  - machine code for, 334
  - processor implementations of, 381–382, 395–396, 401–402
- Branch hazards. *See* Control hazards
- Branch misprediction penalty, 421–422
- Branch prediction, 446–447
- Branch target address (BTA), 333–334, 381
- Branch target buffer, 446
- Branching
  - conditional, 314–315
  - unconditional (jump), 315–316
- Breadboards, 600–601
- BTA. *See* Branch target address
- Bubble, 20, 63, 419
  - pushing, 63–64, 71–73
- Buffers, 20
  - lack of, 117
  - tristate, 74–75
- Bugs, 175
  - in C code, 667–671
- Bus, 56
  - tristate, 75
- Bypassing, 416. *See also* Forwarding
- Byte, 13–14, 322–324. *See also* Character
  - least significant, 13–14
  - most significant, 13–14
- Byte-addressable memory, 301–303
  - big-endian, 302–303
  - little-endian, 302–303
- Byte offset, 483

C

- C programming, 623–671
  - common mistakes. *See* Common mistakes
  - compilation. *See* Compilation
  - conditional statements. *See* Conditional statements
  - control-flow statements. *See* Control-flow statements
  - data types. *See* Data types
  - function calls. *See* Function calls
  - loops. *See* Loops
  - operators. *See* Operators
  - running, 626
  - simple program, 625–626
  - standard libraries. *See* Standard libraries
  - variables. *See* Variables
- Caches, 480–495
  - address fields,
    - block offset, 488–489
    - byte offset, 483
    - set bits, 483
    - tag, 483
  - advanced design, 491–495
  - evolution of, in MIPS, 495
  - multiple level, 492
  - nonblocking, 566
  - organizations, 490
    - direct mapped, 482–486
    - fully associative, 487–488
    - multiway set associative, 486–487
  - parameters
    - block, 481
    - block size, 481, 488–489
    - capacity (C), 480–481
    - degree of associativity (N), 486
    - number of sets (S), 481
  - performance of
    - hit, 478–480
    - hit rate, 478–480
    - miss, 478–480, 493
      - capacity, 493
      - compulsory, 493
      - conflict, 486, 493
      - penalty, 488
    - miss rate, 478–480
      - reducing, 493–494
    - miss rate *vs.* cache parameters, *493–494*
  - replacement policy, 490–491
  - status bits
    - dirty bit (D), 494
    - use bit (U), 490
    - valid bit (V), 484
  - write policy, 494–495
    - write-back, 494–495
    - write-through, 494–495
- CAD. *See* Computer-aided design
- Callee-saved registers, 329
- Canonical form. *See* Sum-of-products, Product-of-sums
- Capacitors, 28
- Capacity, of cache, 480–481
- Capacity miss, 493
- Carry propagate adder (CPA). *See* Ripple-carry adder, Carry-lookahed adder, and Prefix adder
- Carry-lookahead adder (CLA), 241–243, *242*
- Case statement, in HDL, 201–203. *See also* Switch/case statement
- Casez, case?, in HDL, 205
- Cathode, 27
- Cathode ray tube (CRT), 541–542. *See also* VGA monitor
  - horizontal blanking interval, 542
  - vertical blanking interval, 542
- Cause register, 343–344, 441
- Character LCDs, 538–541
- Characters (`char`), 322–324, 630, 649
  - arrays, 322–324. *See also* Strings
  - C type, 649
- Chips, 28
  - multiprocessors, 456
- Chopper constant current drive, 556
- Circuits
  - 74xx series. *See* 74xx series logic
  - application-specific integrated (ASICs), 591
  - astable, 119
  - asynchronous, 120, 122–123
  - combinational. *See* Combinational logic
  - definition of, 55
  - delay, 88–92
  - multiple-output, 68
  - priority, 68
  - sequential. *See* Sequential logic

Circuits (*Continued*)
  synchronous, 122–123
  synchronous sequential, 120–123, *122*
  synthesized, *176*, *179*, *181*
  timing, 88–95
  with two-stage pipeline, *160*
  without glitch, *95*
CISC. *See* Complex Instruction Set Computer
CLBs. *See* Configurable logic blocks
Clock cycles per instruction (CPI), 444, 446
Clock period, 142, 376
Clock skew, 148–151
Clustered multiprocessing, 456
CMOS. *See* Complementary Metal-Oxide-Semiconductor Logic
Combinational composition, 56
Combinational logic, 174
  design, 55–106
    Boolean algebra, 60–66
    Boolean equations, 58–60
    building blocks, 83–88, 239–254
    delays, 88–92
    don't cares, 81–82
    Karnaugh maps (K-maps), 75–83
    multilevel, 66–73
    precedence, 58
    timing, 88–95
    two-level, 69
    X's (contention). *See* Contention
    X's (don't cares). *See* Don't cares (X)
    Z's (floating). *See* Floating (Z)
  HDLs and. *See* Hardware description languages
  truth tables with don't cares, 69, 81–82, 205
Combining theorem, 62
Command line arguments, 666–667
Comments
  in C, 627
  in MIPS assembly, 297
  in SystemVerilog, 180
  in VHDL, 180
Common mistakes in C, 667–671
Comparators, 246–248
Comparison
  in hardware. *See* Comparators, ALU
  in MIPS assembly, 319–320, 345
  using ALU, 250
Compilation, in C, 626–627, 665–666
Compiler, 338–339
  for C, 626–627, 665–666
Complementary Metal-Oxide-Semiconductor Logic (CMOS), 26–34
Complements theorem, 62
Complex instruction set computer (CISC), 298, 347
Complexity management, 4–7
  digital abstraction, 4–5
  discipline, 5–6
  hierarchy, 6–7
  modularity, 6–7
  regularity, 6–7
Compulsory miss, 493
Computer Architecture (Hennessy & Patterson), 444
Computer-aided design (CAD), 71, 129
Concurrent signal assignment statement, 179, 183–184, 193, 200–206
Condition codes. *See* Status flags
Conditional assignment, 181–182
Conditional branches, 314–315
Conditional operator, 181–182
Conditional signal assignments, 181–182
Conditional statements
  in C, 639–640
    if, 639–640
    if/else, 639
    switch/case, 639–640
  in HDL, 194, 201–205
    case, 201–203
    casez, case?, 205
    if, if/else, 202–205
  in MIPS assembly, 316–317
    if, 316–317
    if/else, 317
    switch/case, 317
Configurable logic blocks (CLBs), 274, 589. *See also* Logic elements
Conflict miss, 493
Consensus theorem, 62, 64
Constants
  in C, 627–628
  in MIPS assembly, 304, 313. *See also* Immediates
Contamination delay, 88–92. *See also* Short path
Contention (X), 73–74
Context switching, 455
Continuous assignment statements, 179, 193, 200, 206
Control hazards, 415, 421–424
Control signals, 91, 249
Control unit. *See also* ALU decoder, Main decoder
  of multicycle MIPS processor, 396–408
  of pipelined MIPS processor, 413–414
  of single-cycle MIPS processor, 382–387
Control-flow statements
  conditional statements. *See* Conditional statements
  loops. *See* Loops
Coprocessor 0 registers, 441. *See also* Cause and EPC
Core Duo microprocessor chip, 464
Core i7 microprocessor chip, *465*
Cores, 456
Counters, 260
  divide-by-3, *130*
Covering theorem, 62
CPA. *See* Carry propagate adder (CPA)
CPI. *See* Clock cycles per instruction, Cycles per instruction
Critical path, 89–92, 388
Cross-coupled inverters, 109, *110*
  bistable operation of, *110*
CRT. *See* Cathode ray tube
Cycle time. *See* Clock period
Cycles per instruction (CPI), 375
Cyclic paths, 120
Cyclone IV FPGA, 274–279

D

D flip-flops. *See* flip-flops
D latch. *See* Latch
D/A conversion, 533–537
DACs. *See* Digital-to-analog converters
Data Acquisition Systems (DAQs), 562–563
  myDAQ, 563
Data hazards, 415–421
Data memory, 373
  HDL for, 439
Data segment, 340
Data sheets, 591–596

- Data types, 643–657
  - arrays. *See* Arrays
  - characters. *See* Character (`char`)
  - dynamic memory allocation. *See* Dynamic memory allocation (`malloc` and `free`)
  - linked list. *See* Linked list
  - pointers. *See* Pointers
  - strings. *See* Strings (`str`)
  - structures. *See* Structures (`struct`)
  - typedef, 653–654
- Datapath
  - multicycle MIPS processor, 390–396
  - pipelined MIPS processor, 412–413
  - single-cycle MIPS processor, 376–382
- DC. *See* Direct current
- DC motors, 548–552, *549*
  - H-bridge, 549, *550*
  - shaft encoder, 549–552
- DC transfer characteristics, 24–26. *See also* Noise margins
- DDR3. *See* Double-data rate memory
- DE-9 cable, *524*
- De Morgan's theorem, 63
- Decimal numbers, 9
- Decimal to binary conversion, 11
- Decimal to hexadecimal conversion, 13
- Decode stage, 409–411
- Decoders
  - definition of, 86–87
  - HDL for
    - behavioral, 202–203
    - parameterized, 219
  - logic using, 87–88
  - Seven-segment. *See* Seven-segment display decoder
- Deep pipelines, 444–445
- Delay generation using counters, 528–529
- `Delaymicros` function, 528
- Delays, logic gates. *See* Propagation delay
  - in HDL (simulation only), 188–189
- `DeleteUser` function, 655
- De Morgan, Augustus, *63*
- De Morgan's theorem, 63–64
- Dennard, Robert, *266*
- Destination register (`rd` or `rt`), 378–379, 385, 393
- Device driver, 507–508, 526
- Device under test (DUT), 220
- Dice, 28
- Dielectric, 28
- Digital abstraction, 4–5, 7–9, 22–26
- Digital circuits. *See* Logic
- Digital signal processor (DSP), 457
- Digital system implementation, 583–617
  - 74xx series logic. *See* 74xx series logic
  - application-specific integrated circuits (ASICs), 591
  - assembly of, 599–602
  - breadboards, 600–601
  - data sheets, 591–596
  - economics, 615–617
  - logic families, 597–599
  - packaging, 599–602
  - printed circuit boards, 601–602
  - programmable logic, 584–591
- Digital-to-analog converters (DACs), 531
- DIMM. *See* Dual inline memory module
- Diodes, 27–28
  - p-n junction, *28*
- DIPs. *See* Dual-inline packages
- Direct current (DC) transfer characteristics, 24, *25*
- Direct mapped cache, 482–486, *484*
- Direct voltage drive, 554
- Dirty bit (*D*), 494
- Discipline
  - dynamic, 142–151. *See also* Timing analysis
  - static, 142–151. *See also* Noise margins
- Discrete-valued variables, 7
- Distributivity theorem, 63
- `div`, 314
- Divide-by-3 counter
  - design of, 129–131
  - HDL for, 210–211
- Divider, 253–254
- Division
  - circuits, 253–254
  - MIPS instruction, 314
  - MIPS signed and unsigned instructions, 345
- `divu`, 345
- Don't care (X), 69, 81–83, 205
- Dopant atoms, 27
- Double, C type, 630–631
- Double-data rate memory (DDR), 266, 561
- Double-precision formats, 257–258
- Do/while loops, in C, 641–642
- DRAM. *See* Dynamic random access memory
- Dual inline memory module (DIMM), 561
- Dual-inline packages (DIPs), *28*, 583, 599
- Dynamic branch predictors, 446
- Dynamic data segment, 337
- Dynamic discipline, 142–151. *See also* Timing analysis
- Dynamic memory allocation (`malloc`, `free`), 654–655
  - in MIPS memory map, 337
- Dynamic power, 34
- Dynamic random access memory (DRAM), 266, *267*, 475–478, 561

E

- Economics, 615
- Edge-triggered flip-flop. *See* flip-flop
- EEPROM. *See* Electrically erasable programmable read only memory
- EFLAGS register, 350
- Electrically erasable programmable read only memory (EEPROM), 269
- Embedded I/O (input/output) systems, 508–558
  - analog I/O, 531–537
    - A/D conversion, 532–533
    - D/A conversion, 533–536
  - digital I/O, 513–515
  - general-purpose I/O (GPIO), 513–515
  - interrupts, 529–531
  - LCDs. *See* Liquid Crystal Displays
  - microcontroller peripherals, 537–558
  - motors. *See* Motors
  - PIC32 microcontroller, 509–513
  - serial I/O, 515–527. *See also* Serial I/O
  - timers, 527–529
  - VGA monitor. *See* VGA monitor, 493
- Enabled flip-flops, 115–116
- Enabled registers, 196–197. *See also* flip-flops
- `EPC`. *See* Exception program counter
- EPROM. *See* Erasable programmable read only memory

- Equality comparator, 247
- Equation minimization
  - using Boolean algebra, 65–66
  - using Karnaugh maps. *See* Karnaugh maps
- Erasable programmable read only memory (EPROM), 269, 588
- Ethernet, 561
- Exception program counter (`EPC`), 343–344
- Exceptions, 343–344, 440–443
  - `Cause`. *See* Cause register
  - cause codes, 344
  - `EPC`. *See* Exception program counter
  - handler, 343
  - processor support for, 440–443
    - circuits, 441–442
    - controller, 442–443
- Executable file, 340
- Execution time, 375
- `exit`, 663
- Extended instruction pointer (EIP), 348

F

- Factorial function call, 330–331
  - stack during, *331*
- Factoring state machines, 134–136
- FDIV. *See* Floating-point division
- Field programmable gate arrays (FPGAs), 274–279, 457, 520, 543, 564, 589–591
  - driving VGA cable, *543*
  - in SPI interface, 519–521
- File manipulation, in C, 660–662
- Finite state machines (FSMs), 123–141, 209–213
  - deriving from circuit, 137–140
  - divide-by-3 FSM, 129–131, 210–211
  - factoring, 134–136, *136*
  - in HDL, 209–213
  - LE configuration for, 277–278
  - Mealy FSM, 132–134
  - Moore FSM, 132–134
  - multicycle control, 396–408, *405*, *408*
  - snail/pattern recognizer FSM, 132–134, 212–213
  - state encodings, 129–131. *See also* Binary encoding, One-cold encoding, One-hot encoding
  - state transition diagram, 124, *125*
  - traffic light FSM, 123–129
- Fixed-point numbers, 255–256
- Flags, 250
- Flash memory, 269. *See also* Solid state drives
- Flip-flops, 114–118, 193–197. *See also* Registers
  - back-to-back, *145*, 152–157, *197*. *See also* Synchronizer
  - comparison with latches, 118
  - enabled, 115–116
  - HDL for, 436. *See also* Registers
  - metastable state of. *See* Metastability
  - register, 114–115
  - resettable, 116
  - scannable, 262–263
  - shift register, 261–263
  - transistor count, 114, 117
  - transistor-level, 116–117
- Float, C type, 628–631
  - print formats of, 658–659
- Floating output node, 117
- Floating point division (FDIV) bug, 175
- Floating (Z), 74–75
  - in HDLs, 186–188
- Floating-gate transistor, 269. *See also* Flash memory
- Floating-point coprocessors, 457
- Floating-point division (FDIV), *259*
- Floating-point instructions, MIPS, 346–347
- Floating-point numbers, 256–257
  - addition, 258–259
  - formats, single- and double-precision, 256–258
  - in programming. *See* Float and Double
  - rounding, 258
  - special cases
    - infinity, 257
    - NaN, 257
- Floating-point unit (FPU), *259*, 461
- For loops, 319–320, 322, 642
- Format conversion (`atoi`, `atol`, `atof`), 663–664
- Forwarding, 416–418. *See also* Hazards
- FPGAs. *See* Field programmable gate arrays
- FPU. *See* Floating-point unit
- Frequency shift keying (FSK), 548
  - and GFSK waveforms, *548*
- Front porch, 542
- FSK. *See* Frequency Shift Keying
- FSMs. *See* Finite state machines
- Full adder, 56, 182, 184, 200, 240
  - using always/process statement, 200
- Fully associative cache, 487–488
- Funct field, 305, 621–622
- Function calls, 325–333, 637–638
  - arguments, 325–326, 637
  - leaf, 330
  - naming conventions, 638
  - nonleaf, 330
  - preserved and non-preserved registers, 329–332
  - prototypes, 638
  - recursive, 330–332
  - return, 325–326, 637
  - stack, use of, 327–333. *See also* Stack
  - with no inputs or outputs, 325, 637
- Fuse-programmable ROM, 269

G

- Gated time accumulation, 529
- Gates
  - AND, 20, 22, *128*
  - buffer, 20
  - multiple-input, 21–22
  - NAND, 21, 31
  - NOR, 21–22, 111, *128*
  - NOT, 20
  - OR, 21
  - transistor-level. *See* Transistors
  - XNOR, 21
  - XOR, 21
- General-purpose I/O (GPIO), 513
  - PIC32 ports (pins) of, *515*
  - switches and LEDs example, 513–514
- Generate signal, 241, 243
- Genwaves function, 535
- Glitches, 92–95
- Global data segment, 336–337
- Global pointer (`$gp`), 337
- GPIO. *See* General-purpose I/O
- Graphics accelerators, 464
- Graphics processing unit (GPU), 457

Gray codes, 76
Gray, Frank, *76*
Ground (GND), 22
  symbol for, 31

H

Half adder, 240, *240*
Hard disk, 478–479. *See also* Hard drive
Hard drive, 478–479, 496. *See also* Hard disk, Solid state drive, and Virtual memory
Hardware description languages (HDLs). *See also* SystemVerilog, VHDL
  capacity, 493
  combinational logic, 174, 198
    bitwise operators, 177–179
    blocking and nonblocking assignments, 205–209
    case statements, 201–202
    conditional assignment, 181–182
    delays, 188–189
  data types, 213–217
  history of, 174–175
  if statements, 202–205
    internal variables, 182–184
    numbers, 185
    operators and precedence, 184–185
    reduction operators, 180–181
  modules, 173–174
  parameterized modules, 217–220
  processor building blocks, 434–437
  sequential logic, 193–198, 209–213
  simulation and synthesis, 175–177
  single-cycle MIPS processor, 429–440
  structural modeling, 190–193
  testbench, 220–224, 437–438
Hardware handshaking, 523
Hardware reduction, 70–71. *See also* Equation minimization
Hazard unit, 416–427
Hazards. *See also* Hazard unit
  control hazards, 415, 421–424
  data hazards, 416–421
  read after write (RAW), 415, 451
  solving
    control hazards, 421–424
    forwarding, 416–418
    stalls, 418–421
  write after read (WAR), 451
  write after write (WAW), 451–452
H-bridge control, *550*
HDL. *See* Hardware description language, SystemVerilog, and VHDL
Heap, 337
Heterogeneous multiprocessors, 456–458
Hexadecimal numbers, 11–13
Hexadecimal to binary and decimal conversion, 11, *12*
Hierarchy, 6
HIGH, 22. *See also* 1, ON
High-level programming languages, 296, 624
  compiling, assembling, and loading, 336–341
  translating into assembly, 300
High-performance microprocessors, 444
Hit, 478
Hit rate, 478–480
Hold time constraint, 142–148
  with clock skew, 149–151
Hold time violations, 145, *146*, 147–148, 150–151
Homogeneous multiprocessors, *456*
Hopper, Grace, *337*

I

IA-64, *354*
IA-32 architecture. *See* x86
ICs. *See* Integrated circuits
Idempotency theorem, 62
Identity theorem, 62
Idioms, 177
If statements
  in C, 639
  in HDL, 202–205
  in MIPS assembly, 316–317
If/else statements, 317, 649
  in C, 639–640
  in HDL, 202–205
  in MIPS assembly, 317
IM. *See* Instruction memory
Immediate addressing, 333
Immediates, 304, 313. *See also* Constants
  32-bit, 313
  immediate field, 307–308
  logical operations with, *311*
Implicit leading one, 256
Information, amount of, 8
Initializing
  arrays in C, 645–646
  variables in C, 633
InitTimer1Interrupt function, 530
Input/output elements (IOEs), 274
Input/Output (I/O) systems, 506–569
  device driver, 507–508, 526
  embedded I/O systems. *See* Embedded I/O systems
  I/O registers, 507–508
  memory-mapped I/O, 507–508
  personal computer I/O systems. *See* Personal computer I/O systems
Institute of Electrical and Electronics Engineers (IEEE), *257*
Instruction encoding, x86, 352–354, *353*
Instruction formats, MIPS
  F-type, 346
  I-type, 307–308
  J-type, 308
  R-type, 305–306
Instruction formats, x86, 352–354
Instruction level parallelism (ILP), 452, 455
Instruction memory (IM), 373, 411
  MIPS, *440*
Instruction register (IR), 391, 398
Instruction set, 295, 371–372. *See also* Architecture
Instructions, MIPS, 295–347, 619–622
  arithmetic, 299–300, 304, 314, 344–345, *620–622*
  branching, 314–316
  floating-point, 346–347, *622*
  for accessing memory. *See* Loads, Stores
  F-type, 346
  I-type, 307–308
  J-type, 308
  logical, 308, 310–313
  multiplication and division, 314, 345
  R-type, 305–306, *621–622*
  set less than, 319–320, 345
  signed and unsigned, 344–345
Instructions, x86, 347–355
Instructions per cycle (IPC), 375
Integer Execution Unit (IEU), 461
Integrated circuits (ICs), 599
Intel. *See* x86
Intel x86. *See* x86

- Interrupt service routine (ISR), *529*. *See also* Exceptions
- Interrupts, 343, *529–531*
  - PIC32, *529–531*
- Invalid logic level, 186
- Inverters, 20, *119*, *178*. *See also* NOT gate
  - cross-coupled, 109, *110*
  - in HDL, 178, 199
- *An Investigation of the Laws of Thought* (Boole), *8*
- Involution theorem, 62
- I/O. *See* Input/output systems
- IOEs. *See* Input/output elements
- *IorD*, 393, 397
- IPC. *See* Instructions per cycle
- IR. *See* Instruction register
- *IRWrite*, 391, 397
- ISR. *See* Interrupt service routine
- I-type instructions, 307–308

J

- `j`, 315–316
- `jal`, 325
- Java, 322. *See also* Language
- `jr`, 315–316, 325
- JTA. *See* Jump target address
- J-type instructions, 308
- Jump, MIPS instruction, 315–316
- Jump, processor implementation, 386–387, 404–408
- Jump target address (JTA), 334–335, 386

K

- Karnaugh maps (K-maps), 75–84, 93–95, 126
  - logic minimization using, 77–83
  - prime implicants, 65, 77–81, 94–95
  - seven-segment display decoder, 79–81
    - with “don’t cares,” 81–82
  - without glitches, *95*
- Karnaugh, Maurice, *75*
- Kilobit (Kb/Kbit), 14
- Kilobyte (KB), 14
- K-maps. *See* Karnaugh maps

L

- LAB. *See* Logic array block
- Land grid array, *558*
- Language. *See also* Instructions
  - assembly, 296–304
  - machine, 305–310
  - mnemonic, 297
  - translating assembly to machine, 306
- Last-in-first-out (LIFO) queue, 327. *See also* Stack
- Latches, 111–113
  - comparison with flip-flops, 109, 118
  - D, 113, *120*
  - SR, 111–113, *112*
  - transistor-level, 116–117
- Latency, 157–160, 409–411, 418
- Lattice, silicon, 27
- `lb`, load byte. *See* Loads
- `lbu`, load byte unsigned. *See* Loads
- LCDs. *See* Liquid crystal displays
- Leaf function, 330
- Leakage current, 34
- Least recently used (LRU) replacement, 490–491
  - two-way associative cache with, 490–491, *491*
- Least significant bit (lsb), 13, *14*
- Least significant byte (LSB), 13, *14*, 302
- LEs. *See* Logic elements
- Level-sensitive latch. *See* D latch
- `lh`, load half. *See* Loads
- `lhu`, load half unsigned. *See* Loads
- LIFO. *See* Last-in-first-out queue
- Line options, compiler and command, 665–667
- Linked list, 655–656
- Linker, 340–341
- Liquid crystal displays (LCDs), 538–541
- Literal, 58, 96
- Little-endian memory, 302–303, *302*
- Little-endian bus order in HDL, 178
- Loads, 345
  - base addressing of, 333
  - load byte (`lb` or `lbu`), 304, 323–324, 345
  - load half (`lh` or `lhu`), 345
  - load word (`lw`), 301–304
- Local variables, 332–333
- Locality, 476
- Logic
  - bubble pushing, 71–73
  - combinational. *See* Combinational logic
  - families, 597–599
  - gates. *See* Gates
  - hardware reduction. *See* Equation simplification and Hardware reduction
  - multilevel. *See* Multilevel combinational logic
  - programmable, 584–591
  - sequential. *See* Sequential logic
  - transistor-level. *See* Transistors
  - two-level, 69
- Logic array block (LAB), 275
- Logic arrays, 272–280. *See also* Programmable logic arrays and Field programmable gate arrays
  - transistor-level implementation, 279–280
- Logic elements (LEs), 274–279
  - of Cyclone IV, 276
  - functions built using, 277–278
- Logic families, 25, 597–599
  - compatibility of, 26
  - logic levels of, *25*
  - specifications, *597*, *599*
- Logic gates, 19–22, *179*, 584
  - AND. *See* AND gate
  - AND-OR (AO) gate, 46
  - multiple-input gates, 21–22
  - NAND. *See* NAND gate
  - NOR. *See* NOR gate
  - OR. *See* OR gate
  - OR-AND-INVERT (OAI) gate, 46
  - with delays in HDL, *189*
  - XNOR. *See* XNOR gate
  - XOR. *See* XOR gate
- Logic levels, 22–23
- Logic simulation, 175–176
- Logic synthesis, 176–177, *176*
- Logical instructions, 311–312

- Logical shifter, 250
- Lookup tables (LUTs), 270, 275
- Loops, 317–319, 641–642
  - in C,
    - do/while, 641–642
    - for, 642
    - while, 641
  - in MIPS assembly,
    - for, 319–320
    - while, 318–319
- LOW, 22. *See also* 0, OFF
- Low Voltage CMOS Logic (LVCMOS), 25
- Low Voltage TTL Logic (LVTTL), 25
- LRU. *See* Least recently used replacement
- LSB. *See* Least significant byte
- lsb. *See* Least significant bit
- `lui`, load upper immediate, 313
- LUTs. *See* Lookup tables
- LVCMOS. *See* Low Voltage CMOS Logic
- LVTTL. *See* Low Voltage TTL Logic
- `lw`, load word. *See* Loads

M

- Machine code, assembly and, *437*
- Machine language, 305–310
  - formats, 305–308
    - F-type, 346
    - I-type, 307–308, *307*
    - J-type, 308, *308*
    - R-type, 305–306, *305*
  - interpreting, 308–309
  - stored program, 309–310, *310*
  - translating assembly language to, 306
- Magnitude comparator, 247
- Main decoder, 382–387
  - HDL for, 432
- `main` function in C, 625
- Main memory, 478
- Malloc function, 654
- Mantissa, 258–259
- Mapping, 482
- Master latch, 114
- Master-slave flip-flop, 114
- Masuoka, Fujio, *269*
- math.h, C library, 664–665
- Max-delay constraint. *See* Setup time constraint
- Maxterms, 58
- MCM. *See* Multichip module
- Mealy machines, 123, *123*, 132
  - state transition and output table, *134*
  - state transition diagrams, *133*
  - timing diagrams for, *135*
- Mean time between failure (MTBF), 153–154
- Medium-scale integration (MSI) chips, 584
- Memory. *See also* Memory arrays
  - addressing modes, *349*
  - area and delay, 266–267
  - arrays. *See* Memory arrays average memory access time, 479
  - big-endian, 178, 302–303
  - byte-addressable, 301–303
  - HDL for, 270–272
  - hierarchy, *478*
  - little-endian, 178, 302–303
  - logic using, 270–272
  - main, 478
  - operands in, 301–304
  - physical, 497
  - ports, 265
  - protection, 503. *See also* Virtual memory
  - types, 265–270
    - DDR, 267
    - DRAM, 266
    - flash, 269–270
    - register file, 267–268
    - ROM, 268–270
    - SRAM, 266
  - virtual, 478. *See also* Virtual memory
- Memory arrays, 263–272. *See also* Memory
  - bit cell, 264–269
  - HDL for, 270–272
  - logic using, 270–272
  - organization, 263–265
- Memory hierarchy, 478–479
- Memory interface, 475–476
- Memory map
  - MIPS, 336–337, 341, 507
  - PIC32, 509–510
- Memory Performance. *See* Average Memory Access Time
- Memory protection, 503
- Memory systems, 475
  - MIPS, 495
  - performance analysis, 479–480
  - x86, 564–568
- Memory-mapped I/O
  - address decoder, 507
  - communicating with I/O devices, 507–508
  - hardware, 508
- *Mem Write*, 379, *397*
- *MemtoReg*, 380, *397*
- Metal-oxide-semiconductor field effect transistors (MOSFETs), 26
  - switch models of, *30*
- Metastability, 151–157
  - metastable state, 110, 151
  - resolution time, 151–152, 154–157
  - synchronizers, 152–154
- `mfc0`. *See* Move from coprocessor 0
- Microarchitecture, 351–466. *See also* Architecture
  - advanced. *See* Advanced microarchitecture
  - architectural state. *See* Architectural state
  - description of, 371–374
  - design process, 372–374
  - HDL representation, 429–440
  - multicycle processor. *See* Multicycle MIPS processor
  - performance analysis, 374–376. *See also* Performance analysis
  - pipelined processor. *See* Pipelined MIPS processor
  - single-cycle processor. *See* Single-cycle MIPS processor
  - x86, 458–465
    - evolution of, 458
- Microchip ICD3, *513*
- Microchip In Circuit Debugger 3 (ICD3), 513
- Microcontroller, 508
  - PIC32 (PIC32MX675F512H), 509–513, *510*
    - 64-pin TQFP package in, *511*
    - operational schematic of, *512*
    - to PC serial link, *526*
    - pinout of, *511*
    - virtual memory map of, *510*
- Microcontroller peripherals, 537–558
  - Bluetooth wireless communication, 547–548
  - character LCD, 538–541
    - control, 540–541
    - parallel interface, *539*

Microcontroller peripherals (*Continued*)
motor control, 548–549
VGA monitor, 541–547
Microcontroller units (MCUs), 508
Micro-ops, 461
Microprocessors, 3, *13*, 295
architectural state of, 310
designers, 444
high-performance, 444
Millions of instructions per second, 409
Min-delay constraint. *See* Hold time constraint
Minterms, 58
MIPS. *See also* Architecture and Microarchitecture
architecture, 296, 509
floating-point instructions, *346*, 346–347
instruction set, 385
microarchitectures
multicycle. *See* Multicycle MIPS processor
pipelined. *See* Pipelined MIPS processor
single-cycle. *See* Single-cycle MIPS processor
microprocessor, 441, 452, 455
data memory, 373
instruction memory, 373
program counter, 373
register file, 373
state elements of, *373*
processor control, 344
register set, *300*
vs. x86 architecture, *348*
MIPS instructions, 295–356, *219–222*
branching. *See* Branching
formats
F-type, *622*
I-type, *307*, 307–308
J-type, 308, *308*
R-type, 305–307
multiplication and division, 314, 345
opcodes, *620–621*
R-type funct fields, *621–622*
MIPS processors. *See* MIPS multi-cycle processor, MIPS pipelined processor, and MIPS single-cycle processor
HDL for. *See* MIPS single-cycle HDL
MIPS registers,
co-processor 0 registers, 344, 441–443
program counter, 310, 372–373
register file, 372–373
register set, 298–300
MIPS single-cycle HDL, 429–440
building blocks, 434–437
controller, 429
datapath, 429
testbench, 437–440
Miss, 478–480, 493
capacity, 493
compulsory, 493
conflict, 486, 493
Miss penalty, 488
Miss rate, 478–480
and access times, *480*
Misses
cache, 478
capacity, 493
compulsory, 493
conflict, 493
page fault, 497
Modularity, 6
Modules, in HDL
behavioral and structural, 173–174
parameterized modules, 217–220
Moore, Gordon, *30*
Moore machines, 123, 132
state transition and output table, *134*
state transition diagrams, *133*
timing diagrams for, *135*
Moore's law, *30*
MOS transistors. *See* Metal-oxide-semiconductor field effect transistors
MOSFET. *See* Metal-oxide-semiconductor field effect transistors
Most significant bit (msb), 13, *14*
Most significant byte (MSB), 13, *14*, 302
Motors
DC, *548–552*
H-bridge, *550*
servo, 549, *552–554*
stepper, 548, *554–558*
Move from coprocessor 0 (`mfc0`), 344, 441–443. *See also* Exceptions
MPSSE. *See* Multi-Protocol Synchronous Serial Engine
MSB. *See* Most significant byte
msb. *See* Most significant bit
MSI chips. *See* Medium-scale integration
MTBF. *See* Mean time between failure
`mul`, multiply, 32-bit result, 314
`mult`, multiply, 64-bit result, 314
Multichip module (MCM), *566*
Multicycle MIPS processor, 389–408
control, 396–404
datapath, 390–396
performance, 405–408
Multilevel combinational logic, 69–73. *See also* Logic
Multilevel page tables, 504–506
Multiple-output circuit, 68–69
Multiplexers, 83–86
definition of, 83–84
HDL for
behavioral model of, 181–183
parameterized N-bit, 218–219
structural model of, 190–193
logic using, 84–86
symbol and truth table, *83*
Multiplicand, 252
Multiplication, 314, 345. *See also* Multiplier
MIPS instruction, 314
signed and unsigned instructions, 345
Multiplier, 252–253
schematic, 252
HDL for, 253
Multiprocessors
chip, 456
heterogeneous, 456–458
homogeneous, 456
Multi-Protocol Synchronous Serial Engine (MPSSE), *563*, *563*
Multithreaded processor, 455
Multithreading, 455
`multu`, 345
Mux. *See* Multiplexers
myDAQ, *563*

N

NAND (7400), *585*
NAND gate, 21, *21*, *31*
CMOS, 31–32, *31–32*
Nested if/else statement, 640
Nibbles, 13–14
nMOS transistors, 28–31, *29–30*
Noise margins, 23–26, *23*
calculating, 23–24
Nonarchitectural state, 372
Nonblocking and blocking assignments, 199–200, 205–209

- Nonleaf function, 330
- Nonpreserved registers, 329, *330*
- `nop`, 342
- `nor`, 311
- NOR gate, 21–22, 111, *128*, *585*
  - chip (7402), *585*
  - CMOS, 32
  - pseudo-nMOS logic, *33*
  - truth table, *22*
- Not a number (NaN), 257
- NOT gate, 20
  - chip (7404), *585*
  - CMOS, 31
- Noyce, Robert, *26*
- Null element theorem, 62
- Number conversion
  - binary to decimal, *10–11*
  - binary to hexadecimal, 12
  - decimal to binary, 11, 13
  - decimal to hexadecimal, 13
  - hexadecimal to binary and decimal, 11, *12*
  - taking the two's complement, 16
- Number systems, 9–19
  - binary, 9–11, *10–11*
  - comparison of, 18–19, *19*
  - decimal, *9*
  - estimating powers of two, 14
  - fixed-point, 255, *255–256*
  - floating-point, 256–259
    - addition, 258–259, *259*
    - special cases, 257
  - hexadecimal, 11–13, *12*
  - negative and positive, 15
  - signed, 15
  - unsigned, 9–11

O

- OFF, 23. *See also* 0, LOW
- Offset, 391, *392*
- ON, 23. *See also* 1, HIGH
- One-bit dynamic branch predictor, 446–447
- One-cold encoding, 130
- One-hot encoding, 129–131
- One-time programmable (OTP), 584
- Opcode, 305, *620–621*
- Operands
  - MIPS, 298–304
    - immediates (constants), 304, 313
    - memory, 301–304
    - registers, 298–300
  - x86, 348–350, *349*
- Operation code. *See* Opcode
- Operators
  - in C, 633–636
  - in HDL, 177–185
    - bitwise, 177–181
    - precedence, 185
    - reduction, 180–181
    - table of, 185
    - ternary, 181–182
- `or`, 311
- OR-AND-INVERT (OAI) gate, 46
- OR gate, 21
- `ori`, 311–312
- OTP. *See* One-time programmable
- Out-of-order execution, *453*
- Out-of-order processor, 450–452
- Overflow
  - handling exception for, 343–345, 440–443
  - with addition, 15
- Oxide, 28

P

- Packages, chips, 599–600
- Packed arithmetic, 454
- Page fault, 497
- Page number, 498
- Page offset, 498
- Page table, 498, 500–501
  - number, 504
  - offset, 504
- Pages, 497
- Paging, 504
- Parallel I/O, 515
- Parallelism, 157–160
- Parity gate. *See* XOR
- Partial products, 252
- Pass by reference, 644
- Pass by value, 644
- Pass gate. *See* Transmission gates
- PC. *See* Program counter or Personal computer
- PCB. *See* Printed circuit board
- PCI. *See* Peripheral Component Interconnect
- PCI express (PCIe), 560
- PC-relative addressing, 333–334
- *PCSrc*, 395, *396–397*, 397
- *PCWrite*, 393, 397
- Pentium processors, 460, *462*
  - Pentium 4, 375, *463*, 463–464
  - Pentium II, 461
  - Pentium III, 375, 461, *462*
  - Pentium M, 464
  - Pentium Pro, 461
- Perfect induction, proving theorems using, 64–65
- Performance Analysis, 374–376. *See also* Average Memory Access Time
  - multi-cycle MIPS processor, 405–407
  - pipelined MIPS processor, 426–428
  - processor comparison, 428
  - single-cycle MIPS processor, 388–389
- Periodic interrupts, 530–531
- Peripheral bus clock (PBCLK), *512*
- Peripheral Component Interconnect (PCI), 560
- Peripherals devices. *See* Input/output systems
- Personal computer (PC). *See* x86
- Personal computer (PC) I/O systems, 558–564
  - data acquisition systems, 562–563
  - DDR3 memory, 561
  - networking, 561–562
  - PCI, 560
  - SATA, 562
  - USB, 559–560, 563–564
- Phase locked loop (PLL), 544
- Physical address extension, 567
- Physical memory, 497
- Physical page number (PPN), 499
- Physical pages, 497
- PIC32 microcontroller (PIC32MX675F512H), 509–513. *See also* Embedded I/O systems
- Pipelined MIPS processor, 409–428
  - abstract view of, 411
  - control, 413–414
  - datapath, 412–413
  - description, 409–412
  - hazards, 414–426. *See also* Hazards
  - performance, 426–428
  - throughput, 411
- Pipelining, 158–160. *See also* Pipelined MIPS processor

PLAs. *See* Programmable logic arrays
Plastic leaded chip carriers (PLCCs), 599
Platters, 496
PLCCs. *See* Plastic leaded chip carriers
PLDs. *See* Programmable logic devices
PLL. *See* Phase locked loop
pMOS transistors, 28–31, *29*
Pointers, 643–645, 647, 650, 652, 654
POS. *See* Product-of-sums form
Positive edge-triggered flip-flop, 114
Power consumption, 34–35
Power processor element (PPE), 457
PPN. *See* Physical page number
Prefix adders, 243–245, *244*
Prefix tree, 245
Preserved registers, 329–330, *330*
Prime implicants, 65, 77
Printed circuit boards (PCBs), 601–602
`printf`, 657–659
Priority
  circuit, 68–69
  encoder, 102–103, 105
Procedure calls. *See* Function calls
Processor-memory gap, *477*
Processor performance comparison
  multicycle MIPS processor, 407–408
  pipelined MIPS processor, 428
  single-cycle processor, 388–389
Product-of-sums (POS) form, 60
Program counter (PC), 310, 333, 373, *379*
Programmable logic arrays (PLAs), 67, 272–274, 588–589
  transistor-level implementation, 280
Programmable logic devices (PLDs), 588
Programmable read only memories (PROMs), 268, *270*, 584–588
Programming
  arrays. *See* Arrays
  branching. *See* Branching
  conditional statements, 316–317
  constants. *See* Constants, Immediates
  function calls. *See* Functions
  in C. *See* C programming
  in MIPS, 310–333
  instructions, 619–622
  logical instructions, 311–312
  loops. *See* Loops
  multiplication and division, 314
  shift instructions, 312–313, *312*
PROMs. *See* Programmable read only memories
Propagate signal, 241
Propagation delay, 88–92. *See also* Critical path
Pseudo-direct addressing, 334–335
Pseudo instructions, 342–343
Pseudo-nMOS logic, 33–34, *33*
  NOR gate, *33*
  ROMs and PLAs, 279–280
Pulse-Width Modulation (PWM), 536–537
  analog output with, *537*
  duty cycle, 536
  signal, *536*
PWM. *See* Pulse-Width Modulation

Q

Quiescent supply current, 34

R

Race conditions, 119–120, *120*
`rand`, 662–663
Random access memory (RAM), 265–267, *271*
Read after write (RAW) hazards, 415, 451. *See also* Hazards
Read only memory (ROM), 265, *268–269*, 268–270
  transistor-level implementation, 279–280
ReadData, 378
Read/write head, 496
Receiver gate, 22
Recursive function calls, 330–332
Reduced instruction set computer (RISC), 298
Reduction operators, 180–181
RegDst, 381, 384, 397
Register file (RF)
  HDL for, 435
  in pipelined MIPS processor (write on falling edge), 412
  MIPS register descriptions, 299–300
  schematic, 267–268
  use in MIPS processor, 373
Register renaming, 452–454
Register set, 299–300. *See also* Register file
Register-only addressing, 333
Registers. *See* Flip-flops, MIPS registers, and x86 registers
Regularity, 6
RegWrite, 378, 384, 397, 413, *414*
Replacement policies, 504
Reserved segment, 337
Resettable flip-flops, 116
Resettable registers, 194–196
Resolution time, 151–152
  derivation of, 154–157
  See also Metastability
RF. *See* Register file
Ring oscillator, 119, *119*
Ripple-carry adder, *240*, 240–241, 243
Rising edge, 88
ROM. *See* Read only memory
Rotations per minute (RPM), 549
Rotators, 250–252
Rounding modes, 258
RPM. *See* Rotations per minute
RS-232, *523–524*
R-type instructions, 305–306

S

Sampling, 141
Sampling rate, 531
Sampling time, *532*
SATA. *See* Serial ATA
`sb`, store byte. *See* Stores
Scalar processor, *447*
Scan chains, 261–263
`scanf`, 660
Scannable flip-flop, 262–263
Schematics, rules of drawing, 31, 67
SCK. *See* Serial Clock
SDI. *See* Serial Data In
SDO. *See* Serial Data Out
SDRAM. *See* Synchronous dynamic random access memory
Segment descriptor, 353
Segmentation, 354
Selected signal assignment statements, 182

Semiconductors, 27
  industry, sales, 3
Sequencing overhead, 143–144, 149, 160, 428
Sequential building blocks. *See* Sequential logic
Sequential logic, 109–161, 260–263
  counters, 260
  finite state machines. *See* Finite state machine
  flip-flops, 114–118. *Also see* Registers
  latches, 111–113
    D, 113
    SR, 111–113
  registers. *See* Registers
  shift registers, 261–263
  timing of. *See* Timing Analysis
Serial ATA (SATA), *562*
Serial Clock (SCK), *516*
Serial communication, with PC, *525–527*
Serial Data In (SDI), *516*
Serial Data Out (SDO), *516*
Serial I/O, *515–527*
  SPI. *See* Serial peripheral interface
  UART. *See* Universal Asynchronous Receiver Transmitter
Serial Peripheral Interface (SPI), *515–521*
  connection between PIC32 and FPGA, *519*
  ports
    Serial Clock (SCK), *516*
    Serial Data In (SDI), *516*
    Serial Data Out (SDO), *516*
  register fields in, *517*
  slave circuitry and timing, *520*
  waveforms, *516*
Servo motor, *549*, *552–554*
Set bits, 483
set if less than immediate (`slti`), 345
set if less than immediate unsigned (`sltiu`), 345
set if less than (`slt`)
  circuit, 250
  in MIPS assembly, 319–320
set if less than unsigned (`sltu`), 345
Setup time constraint, 142, 145–147
  with clock skew, 148–150
Seven-segment display decoder, 79–82
  HDL for, 201–202
  with don’t cares, 82–83

SFRs. *See* Special function registers
`sh`, store half. *See* Stores
Shaft encoder, *552*, *552*
Shift instructions, 312–313, *312*
Shift registers, 261–263
Shifters, 250–252
Short path, 89–92
Sign bit, *16*
Sign extension, 18, 308
  HDL for, 436
Signed and unsigned instructions, 344–345
Signed binary numbers, 15–19
Signed multiplier, 217
Sign/magnitude numbers, 15–16, 255
Silicon dioxide (SiO2), 28
Silicon lattice, 27
SIMD. *See* Single instruction multiple data
Simple programmable logic devices (SPLDs), 274
Simulation waveforms, *176*
  with delays, *189*
Single-cycle MIPS processor, 376–389
  control, 382–385
  datapath, 376–382
  example operation, 384–385
  HDL of, 429–440
  performance, 389
Single instruction multiple data (SIMD), *447*, 454, 463
Single-precision formats, 257–258. *See also* Floating-point numbers
Skew. *See* Clock skew
Slash notation, *56*
Slave latch, 114. *See also* D flip-flop
`sll`, 312
`sllv`, 313
SLT. *See* set if less than
`slt`, set if less than, 319–320
`slti`, 345
`sltiu`, 345
`sltu`, 345
Small-scale integration (SSI) chips, *584*
Solid state drive (SSD), 478–479. *See also* Flash memory and Hard drive
SOP. *See* Sum-of-products form
Spatial locality, 476, 488–490
Spatial parallelism, 157–158
Special function registers (SFRs), *509*
SPECINT2000, 406
SPI. *See* Serial Peripheral Interface
Spinstepper function, *557*

SPIxCON, *516*
Squashing, 452
SR latches, 111–113, *112*
SRAM. *See* Static random access memory
`srand`, *662–663*
`srl`, 312
`srlv`, 313
SSI chips. *See* Small-scale integration
Stack, 327–333. *See also* Function calls
  during recursive function call, *331*
  preserved registers, 329–330
  stack frame, 328, *332*
  stack pointer (`$sp`), 327
  storing additional arguments on, 332–333
  storing local variables on, 332–333
Stalls, 418–421. *See also* Hazards
Standard libraries, *657–665*
  math, *664–665*
  stdio, *657–662*
    file manipulation, *660–662*
    `printf`, *657–659*
    `scanf`, *660*
  stdlib, *662–664*
    `exit`, *663*
    format conversion (`atoi`, `atol`, `atof`), *663–664*
    `rand`, `srand`, *662–663*
  string, *665*
State encodings, FSM, 129–131, *134*. *See also* Binary encoding, One-cold encoding, One-hot encoding
State machine circuit. *See* Finite state machines
State variables, 109
Static branch prediction, 446
Static discipline, 24–26
Static power, 34
Static random access memory (SRAM), 266, *267*
Status flags, 350
stdio.h, C library, *657–662*. *See also* Standard libraries
stdlib.h, C library, *662–664*. *See also* Standard libraries
Stepper motors, *549*, *554–556*
  bipolar stepper motor, *554–555*
    half-step drive, *554*
    two-phase-on drive, *554*
    wave drive, *554*
Stored program, 309–310

- Stores
  - store byte (`sb` or `sbu`), 302–304, 323–324
  - store half (`sh` or `shu`), 345
  - store word (`sw`), 302–304
- string.h, C library, 665
- Strings, 324, 650–651. *See also* Characters (`char`)
- Structural modeling, 173–174, 190–193
- Structures (`struct`), 651–653
- `sub`, 297
- Substrate, 28–29
- Subtraction, 17, 246, 297
  - signed and unsigned instructions, 344–345
- Subtractor, 246–247
- `subu`, 345
- Sum-of-products (SOP) form, 58–60
- Superscalar processor, 447–449
- Supply voltage, 22. *See also* V_{DD}
- `sw`, store word, 302–304. *See also* Stores
- Swap space, 504
- Switch/case statements
  - in C, 639–640
  - in HDL. *See* Case statement
  - in MIPS assembly, 317
- Symbol table, *339*
- Symmetric multiprocessing (SMP). *See* Homogeneous multiprocessors
- Synchronizers, 152–154, *152–153*
- Synchronous circuits, 122–123
- Synchronous dynamic random access memory (SDRAM), 267
  - DDR, 267
- Synchronous logic, design, 119–123
- Synchronous resettable flip-flops, 116
- Synchronous sequential circuits, 120–123, *122*. *See also* Finite state machines
  - timing specification. *See* Timing analysis
- Synergistic processor elements (SPEs), 457
- Synergistic Processor Unit (SPU) ISA, *458*
- SystemVerilog, 173–225. *See also* Hardware description languages
  - accessing parts of busses, 188, 192
  - bad synchronizer with blocking assignments, 209
  - bit swizzling, 188
  - blocking and nonblocking assignment, 199–200, 205–208
  - case statements, 201–202, 205
  - combinational logic using, 177–193, 198–208, 217–220
  - comments, 180
  - conditional assignment, 181–182
  - data types, 213–217
  - decoders, 202–203, 219
  - delays (in simulation), 189
  - divide-by-3 FSM, 210–211
  - finite state machines (FSMs), 209–213
    - Mealy FSM, 213
    - Moore FSM, 210, 212
  - full adder, 184
    - using always/process, 200
    - using nonblocking assignments, 208
  - history of, 175
  - if statements, 202–205
  - internal signals, 182–184
  - inverters, 178, 199
  - latches, 198
  - logic gates, 177–179
  - multiplexers, 181–183, 190–193, 218–219
  - multiplier, 217
  - numbers, 185–186
  - operators, *185*
  - parameterized modules, 217–220
    - N:2^N decoder, 219
    - N-bit multiplexers, 218–219
    - N-input AND gate, 220
  - priority circuit, 204
    - using don't cares, 205
  - reduction operators, 180–181
  - registers, 193–197
    - enabled, 196
    - resettable, 194–196
  - sequential logic using, 193–198, 209–213
  - seven-segment display decoder, 201
  - simulation and synthesis, 175–177
  - structural models, 190–193
  - synchronizer, 197
  - testbench, 220–224, 437–438
    - self-checking, 222
    - simple, 221
    - with test vector file, 223–224
  - tristate buffer, 187
  - truth tables with undefined and floating inputs, 187, *188*
  - z's and x's, 186–188, 205

T

- Tag, 483
- Taking the two's complement, 16–17
- Temporal locality, 476, 481–482, 485, 490
- Temporal parallelism, 158–159
- Temporary registers, 299, 329–330
- Ternary operators, 181, 635
- Testbenches, HDLs, 220–224
  - for MIPS processor, 437–438
  - simple, 220–221
  - self-checking, 221–222
  - with testvectors, 222–224
- Text segment, 336, 340
- Thin Quad Flat Pack (TQFP), 510
- Thin small outline package (TSOP), 599
- Thread level parallelism (TLP), 455
- Threshold voltage, 29
- Throughput, 157–160, 374–375, 409–411, 455
- Timers, 527–529
  - delay generation using, 528–529
- Timing
  - of combinational logic, 88–95
    - delay. *See* Propagation delay, Contamination delay
    - glitches. *See* Glitches
  - of sequential logic, 141–157
    - analysis. *See* Timing analysis
    - clock skew. *See* Clock skew
    - dynamic discipline, 141–142
    - metastability. *See* Metastability
    - resolution time. *See* Resolution time
    - system timing. *See* Timing analysis
- Timing analysis, 141–151
  - calculating cycle time. *See* Setup time constraint
  - hold time constraint. *See* Hold time constraint
  - max-delay constraint. *See* Setup time constraint
  - min-delay constraint. *See* Hold time constraint
  - multi-cycle processor, 407–408
  - pipelined processor, 428
  - setup time constraint. *See* Setup time constraint
  - single-cycle processor, 388–389
  - with clock skew. *See* clock skew

- TLB. *See* Translation lookaside buffer
- Trace cache, 463
- Transistors, 26–34
  - bipolar, 26
  - CMOS, 26–33
  - gates made from, 31–34
  - latches and flip-flops, 116–117
  - MOSFETs, 26
  - nMOS, 28–34, *29–33*
  - pMOS, 28–34, *29–33*
    - pseudo-nMOS, 33–34
    - ROMs and PLAs, 279–280
    - transmission gate, 33
- Transistor-Transistor Logic (TTL), 25–26, 597–598
- Translating and starting a program, 337–342, *338*
- Translation lookaside buffer (TLB), 502–503
- Transmission Control Protocol and Internet Protocol (TCP/IP), *561*
- Transmission gates, 33
- Transmission lines, 602–615
  - characteristic impedance (Z_0), 612–613
    - derivation of, 612–613
  - matched termination, 604–606
  - mismatched termination, 607–610
  - open termination, 606–607
  - reflection coefficient (k_r), 613–614
    - derivation of, 613–614
  - series and parallel terminations, 610–612
  - short termination, 607
  - when to use, 610
- Transparent latch. *See* D latch
- Traps, 343
- Tristate buffer, 74–75, *187*
  - HDL for, 186–187
  - multiplexer built using, 84–85, 91–93
- Truth tables, 20
  - ALU decoder, 383, *384*
  - multiplexer, 83
  - seven-segment display decoder, 79
  - SR latch, 111, *112*
  - with don't cares, 69, 81–83, 205
  - with undefined and floating inputs, *187–188*
- TSOP. *See* Thin small outline package
- TTL. *See* Transistor-Transistor Logic
- Two-bit dynamic branch predictor, 447
- Two-cycle latency of `lw`, 418
- Two-level logic, 69
- Two's complement numbers, 16–18
- `typedef`, 653–654

U

- UART. *See* Universal Asynchronous Receiver Transmitter
- Unconditional branches, 315–316
- Undefined instruction exception, 343–344, 440–443
- Unicode, 322
- Unit under test (UUT), 220
- Unity gain points, 24
- Universal Asynchronous Receiver Transmitter (UART), *521–527*
  - hardware handshaking, 523
  - STA register, 524
- Universal Serial Bus (USB), *270*, 523, *559–560*
  - USB 1.0, *560*
  - USB 2.0, *560*
  - USB 3.0, *560*
- Unsigned multiplier, *217*
- Unsigned numbers, 18
- USB. *See* Universal Serial Bus
- USB links, *563–564*
  - FTDI, *563*
    - UM232H module, *564*
- Use bit (*U*), 490

V

- Valid bit (*V*), 484
- Vanity Fair (Carroll), *76*
- Variables in C, 629–633
  - global and local, 631–632
  - initializing, 633
  - primitive data types, 630–631
- Variable-shift instruction, 313
- V_{CC}, *23*. *See also* Supply voltage, V_{DD}
- V_{DD}, 22, *23*. *See also* Supply voltage
- Vector processor, *447*
- Verilog. *See* SystemVerilog
- Very High Speed Integrated Circuits (VHSIC), 175. *See also* VHDL
- VGA. *See* VGA monitor
- VGA (Video Graphics Array) monitor, 541–547
  - connector pinout, *543*
  - driver for, *544–547*
- VHDL. *See* VHSIC Hardware Description Language
- VHSIC. *See* Very High Speed Integrated Circuits
- VHSIC Hardware Description Language (VHDL), 173–175
  - accessing parts of busses, 188, 192
  - bad synchronizer with blocking assignments, 209
  - bit swizzling, 188
  - blocking and nonblocking assignment, 199–200, 205–208
  - case statements, 201–202, 205
  - combinational logic using, 177–193, 198–208, 217–220
  - comments, 180
  - conditional assignment, 181–182
  - data types, 213–217
  - decoders, 202–203, 219
  - delays (in simulation), 189
  - divide-by-3 FSM, 210–211
  - finite state machines (FSMs), 209–213
    - Mealy FSM, 213
    - Moore FSM, 210, 212
  - full adder, 184
    - using always/process, 200
    - using nonblocking assignments, 208
  - history of, 175
  - if statements, 202
  - internal signals, 182–184
  - inverters, 178, 199
  - latches, 198
  - logic gates, 177–179
  - multiplexer, 181–183, 190–193, 218–219
  - multiplier, 217
  - numbers, 185–186
  - operators, *185*
  - parameterized modules, 217–220
    - N:2^N decoder, 219
    - N-bit multiplexers, 218, *219*
    - N-input AND gate, 220, *220*
  - priority circuit, 204
    - reduction operators, 180–181
    - using don't cares, 205

- VHSIC Hardware Description Language (VHDL) (*Continued*)
  - reduction operators, 180–181
  - registers, 193–197
    - enabled, 196
    - resettable, 194–196
  - sequential logic using, 193–198, 209–213
  - seven-segment display decoder, 201
  - simulation and synthesis, 175–177
  - structural models, 190–193
  - synchronizer, 197
  - testbench, 220–224, 437–438
    - self-checking, 222
    - simple, 221
    - with test vector file, 223–224
  - tristate buffer, 187
  - truth tables with undefined and floating inputs, 187, *188*
  - z's and x's, 186–188, 205
- Video Graphics Array (VGA). *See* VGA monitor
- Virtual address, 497
  - space, 503
- Virtual memory, 478, 496–506
  - address translation, 497–500
  - cache terms comparison, 497
  - memory protection, 503
  - multilevel page tables, 504–506
  - page fault, 497
  - page number, 498
  - page offset, 498
  - page table, 500–501
  - pages, 497
  - replacement policies, 504
  - translation lookaside buffer (TLB), 502–503
  - write policy, 494–495
  - x86, 567. *See also* x86
- Virtual page number (VPN), 499
- Virtual pages, 497
- V_{SS}, *23*

W

- Wafers, 28
- Wait states, 564
- Wall, Larry, *20*
- WAR. *See* Write after read
- WAW. *See* Write after write
- Weak pull-up, 33
- Weird number, 18
- While loops, 318–319, 641
- White space, 180
- Whitmore, Georgiana, *7*
- Wi-Fi, 561
- Wire, 67
- Wireless communication, Bluetooth, 547–548
- Word-addressable memory, *301*, 302
- Wordline, 264
- Write after read (WAR) hazard, 451–453. *See also* Hazards
- Write after write (WAW) hazard, 451
- Write policy, 494–495
  - write-back, 494–495
  - write-through, 494–495

X

- X. *See* Contention, Don't care
- x86
  - architecture, 347–355
    - branch conditions, *352*
    - instruction encoding, 352–354, *353*
    - instructions, 350–352, *351*
    - memory addressing modes, *349*
    - operands, 348–350
    - registers, 348
    - status flags, 350
    - *vs.* MIPS, *348*
  - cache systems, 564–567
  - memory system, evolution of, *565*
  - microarchitecture, 458–465
    - evolution of, 458–459
  - programmed I/O, 567–568
  - registers, 348
  - virtual memory, 567
    - protected mode, 567
    - real mode, 567
- Xilinx FPGA, 274–276
- XNOR gate, 21–22
- XOR gate, 21
- `xor`, 311
- `xori`, 311–312

Z

- Z. *See* Floating
- Zero extension, 250, 308, 311–312, 345