

Index

0, 23. *See also* LOW, OFF
1, 23. *See also* HIGH, ON
74xx series logic, 515–516
 parts,
 2:1 Mux (74157), 518
 3:8 Decoder (74138), 518
 4:1 Mux (74153), 518
 AND (7408), 517
 AND3 (7411), 517
 AND4 (7421), 517
 Counter (74161, 74163), 518
 FLOP (7474), 515–517
 NAND (7400), 517
 NOR (7402), 517
 NOT (7404), 515
 OR (7432), 517
 XOR (7486), 517
 Register (74377), 518
 Tristate buffer (74244), 518
 schematics of, 517–518

A

add, 291
Adders, 233–240
 carry-lookahead. *See* Carry-lookahead adder
 carry-propagate (CPA). *See* Carry-propagate adder
 prefix. *See* Prefix adder
 ripple-carry. *See* Ripple-carry adder
add immediate (addi), 327, 378
add immediate unsigned (addiu), 552
add unsigned (addu), 554
Addition, 14–15, 233–240, 291

 floating-point, 252–255
 overflow. *See* Overflow
 two's complement, 15, 240
 underflow. *See* Underflow
 unsigned binary, 15
Address, 485–490
 physical, 485
 translation, 486–489
 virtual, 485–490
 word alignment, 298
Addressing modes, 327–329
 base, 327
 immediate, 327
 MIPS, 327–329
 PC-relative, 327–328
 pseudo-direct, 328–329
 register-only, 327
Advanced microarchitecture, 435–447
 branch prediction. *See* Branch prediction
 deep pipelines. *See* Deep pipelines
 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) units
 superscalar processor. *See* Superscalar processor
 vector processor. *See* Single instruction multiple data (SIMD) units
Alignment. *See* Word alignment
ALU. *See* Arithmetic/logical unit

ALU decoder, 374–376
ALU decoder truth table, 376
ALUControl, 370
ALUOp, 374–375
ALUOut, 383–385
ALUResult, 370–371
AMAT. *See* Average memory access time
Amdahl, Gene, 468
Amdahl's Law, 468
Anodes, 27–28
and immediate (andi), 306, 346–347
AND gate, 20–22, 32–33
Application-specific integrated circuits (ASICs), 523
Architectural state, 363–364
Architecture. *See* Instruction Set Architecture
Arithmetic, 233–249, 305–308, 515.
 See also Adders, Addition, Comparator, Divider, Multiplier
 adders. *See* Adders
 addition. *See* Addition
 ALU. *See* Arithmetic/logical unit
 circuits, 233–248
 comparators. *See* Comparators
 divider. *See* Divider
 division. *See* Division
 fixed-point, 249
 floating-point, 252–255
 logical instructions, 305–308
 multiplier. *See* Multiplier
 packed, 445
 rotators. *See* Rotators
 shifters. *See* Shifters
 signed and unsigned, 338–339
 subtraction. *See* Subtraction

- Arithmetic (*Continued*)
 - subtractor. *See* Subtractor
 - underflow. *See* Addition, Underflow
 - Arithmetic/logical unit (ALU), 242–244, 515
 - 32-bit, 244
 - adder. *See* Adders
 - ALUOp, 374–376
 - ALUResult, 370–371
 - ALUOut, 383–385
 - comparator. *See* Comparators
 - control, 242
 - subtractor. *See* Subtractor
 - Arrays, 314–318
 - accessing, 315–317
 - bytes and characters, 316–318
 - FPGA, *See* Field programmable gate array
 - logic. *See* Logic arrays
 - memory. *See* Memory
 - RAM, 265
 - ROM, 266
 - ASCII (American Standard Code for Information Interchange)
 - codes, 317
 - table of, 317
 - Assembler directives, 333
 - ASICs. *See* Application-specific integrated circuits
 - Assembly language, MIPS. *See* MIPS assembly language
 - Associativity, 58–59
 - Average memory access time (AMAT), 467–468
 - Asynchronous circuits, 116–117
 - Asynchronous inputs, 144
 - Asynchronous resettable registers
 - HDL for, 192
 - Axioms. *See* Boolean axioms
- B**
- Babbage, Charles, 7–8, 26
 - Base address, 295–296
 - Base register, 302, 343, 347
 - Base 2 number representations. *See* Binary numbers
 - Base 8 number representations. *See* Octal numbers
 - Base 16 number representations. *See* Hexadecimal numbers
 - Block,
 - digital building. *See* Digital building blocks
 - in code,
 - else block, 311
 - if block, 310–311
 - Base addressing, 327
 - Baudot, Jean-Maurice-Emile, 317
 - Behavioral modeling, 171–185
 - Benchmarks, 367
 - SPEC2000, 398–399
 - Biased numbers, 41. *See also* Floating point
 - Big-Endian, 172, 296–297
 - Binary numbers, 9–11. *See also*
 - Arithmetic
 - ASCII, 317
 - binary coded decimal, 252
 - conversion. *See* Number conversion
 - fixed point, 249–250
 - floating-point. *See* Floating-point numbers
 - signed, 15–19
 - sign/magnitude. *See* Sign/magnitude numbers
 - two's complement. *See* Two's complement numbers
 - unsigned, 9–15
 - Binary to decimal conversion, 10–11
 - Binary to hexadecimal conversion, 12
 - Bit, 9–11
 - dirty, 482–483
 - least significant, 13–14
 - most significant, 13–14
 - sign, 16
 - use, 478–479
 - valid, 472
 - Bit cells, 258
 - Bit swizzling, 182
 - Bitwise operators, 171–174
 - Boole, George, 8
 - Boolean algebra, 56–62
 - axioms, 57
 - equation simplification, 61–62
 - theorems, 57–60
 - Boolean axioms, 57
 - Boolean equations, 54–56
 - product-of-sums (POS) canonical form, 56
 - sum-of-products (SOP) canonical form, 54–55
 - terminology, 54
 - Boolean theorems, 57–60
 - DeMorgan's, 59
 - complement, 58, 59
 - consensus, 60
 - covering, 58
 - combining, 58
 - Branching, 308–310
 - calculating address, 309
 - conditional, 308
 - prediction, 437–438
 - target address (BTA), 327–328
 - unconditional (jump), 309
 - Branch equal (beq), 308–309
 - Branch not equal (bne), 308–309
 - Branch/control hazards, 407, 413–416. *See also* Hazards
 - Branch prediction, 437–438
 - Breadboards, 532–533
 - Bubble, 59
 - pushing, 67–69
 - Buffer, 20
 - tristate, 70–71
 - “Bugs,” 169
 - Bus, 52
 - tristate, 71
 - Bypassing, 408
 - Bytes, 13–14
 - Byte-addressable memory, 295–297
 - Byte order, 296–297
 - Big-Endian, 296–297
 - Little-Endian, 296–297
- C**
- C programming language, 290
 - overflows, 339
 - strings, 318
 - Caches, 468–484. *See also* Memory
 - accessing, 472–473, 476–477, 479
 - advanced design, 479–483
 - associativity, 469, 474–475, 478–479
 - block size, 476–477
 - blocks, 469–470, 476
 - capacity, 468–470
 - data placement, 469–470
 - data replacement, 478–479

- definition, 468
 - direct mapped, 470–474
 - dirty bit, 482–483
 - entry fields, 472–473
 - evolution of MIPS, 483
 - fully associative, 475–476
 - hit, 466
 - hit rate, 467
 - IA-32 systems, 499–500
 - level 2 (L2), 480
 - mapping, 470–478
 - miss, 466
 - capacity, 481
 - compulsory, 481
 - conflict, 481
 - miss rate, 467
 - miss rate versus cache parameters, 481–482
 - miss penalty, 476
 - multiway set associative, 474–475
 - nonblocking, 500
 - organizations, 478
 - performance, 467
 - set associative, 470, 474–475
 - tag, 471–472
 - use bit, 478–479
 - valid bit, 472–473
 - write policy, 482–483
 - CAD. *See* Computer-aided design
 - Canonical form, 53
 - Capacitors, 28
 - Capacity miss, 481
 - Carry-lookahead adder, 235–237
 - Carry propagate adder (CPA), 274
 - Cathodes, 27–28
 - Cause register, 337–338
 - Chips, 28, 449
 - 74xx series logic. *See* 74xx series logic
 - Circuits, 82–83, 87–88
 - 74xx series logic. *See* 74xx series logic
 - application-specific integrated (ASIC), 523
 - arithmetic. *See* Arithmetic
 - asynchronous, 116–117
 - bistable device, 147
 - combinational, 53
 - definition, 51
 - delays, 73–75, 84–87
 - calculating, 86–87
 - dynamic, 273
 - multiple-output, 64
 - pipelining, 152
 - priority, 65, 202, 203
 - synchronous sequential, 114–116
 - synthesized, 186–190, 193–195, 199, 200
 - timing, 84–91
 - timing analysis, 138
 - types, 51–54
 - without glitch, 91
 - CISC. *See* Complex instruction set computers
 - CLBs. *See* Configurable logic blocks
 - Clock cycle. *See* Clock period
 - Clock period, 135–139
 - Clock rate. *See* Clock period
 - Clock cycles per instruction (CPI), 367–368
 - Clustered computers, 447
 - Clock skew, 140–143
 - CMOS. *See* Complementary Metal-Oxide-Semiconductor Logic
 - Code, 303
 - Code size, 334
 - Combinational composition, 52–53
 - Combinational logic design, 51–100
 - Boolean algebra, 56–62
 - Boolean equations, 54–56
 - building blocks, 79–84
 - delays, 85–87
 - don't cares, 65
 - HDLs and. *See* Hardware description languages
 - Karnaugh maps, 71–79
 - logic, 62–65
 - multilevel, 65–69
 - overview, 51–54
 - precedence, 54
 - timing, 84–91
 - two-level, 65–66
 - X's (contention). *See* Contention
 - X's (don't cares). *See* Don't cares
 - Z's (floating). *See* Floating
 - Comparators, 240–241
 - equality, 241
 - magnitude, 241, 242
 - Compiler, 331–333
 - Complementary Metal-Oxide-Semiconductor Logic (CMOS), 25
 - bubble pushing, 69
 - logic gates, 31–33
 - NAND gate, 32
 - NOR gate, 32–33
 - NOT gate, 31
 - transistors, 26–34
 - Complex instruction set computers (CISC), 292, 341
 - Compulsory miss, 481
 - Computer-aided design (CAD), 167
 - Computer Organization and Design (Patterson and Hennessy), 290, 363
 - Complexity management, 4–6
 - abstraction, 4–5
 - discipline, 5–6
 - hierarchy, 6
 - modularity, 6
 - regularity, 6
 - Conditional assignment, 175–176
 - Conditional branches, 308
 - Condition codes, 344
 - Conflict misses, 481
 - Conditional statements, 310–311
 - Constants, 298. *See also* Immediates
 - Contamination delay, 84–88
 - Contention (X), 69–70
 - Context switch, 446
 - Control signals, 79, 242–243
 - Control unit, 364, 366, 374–406
 - multicycle MIPS processor FSM, 381–395
 - pipelined MIPS processor, 405–406
 - single-cycle MIPS processor, 374–377
 - Configurable logic blocks (CLBs), 268–272
 - Control hazards. *See* Branch/control hazards, Pipelining
 - Coprocessor 0, 338
 - Counters, 254
 - Covalent bond, 27
 - CPA. *See* Carry propagate adder
 - CPI. *See* Clock cycles per instruction
 - Critical path, 85–89
 - Cyclic paths, 114
 - Cycle time. *See* Clock Period
- D**
- Data hazards. *See* Hazards
 - Data memory, 365
 - Data sheets, 523–528
 - Datapath, 364. *See also* MIPS micro-processors

- Datapath (*Continued*)
 - elements, 364
 - multicycle MIPS processor, 382–388
 - pipelined MIPS processor, 404
 - single-cycle MIPS processor, 368–374
 - Data segments. *See* Memory map
 - Data types. *See* Hardware description languages
 - DC. *See* Direct current
 - Decimal numbers, 9
 - conversion to binary and hexadecimal. *See* Number conversion
 - scientific notation, 249–250
 - Decimal to Binary conversion, 11
 - Decimal to hexadecimal conversion, 13
 - Decoders
 - implementation, 83
 - logic, 83
 - parameterized, 212
 - Deep pipelines, 435–436
 - Delay, 182
 - DeMorgan, Augustus, 56
 - DeMorgan's theorem, 59, 60
 - Dennard, Robert, 260
 - Destination register, 301, 305–306, 370–371, 377–378
 - Device under test (DUT), 214–218. *See also* Unit under test
 - Device driver, 496, 498
 - Dice, 28
 - Digital abstraction, 4–5, 8–9, 22–26
 - DC transfer characteristics, 23–24
 - logic levels, 22–23
 - noise margins, 23
 - supply voltage, 22
 - Digital design
 - abstraction, 7–9
 - discipline, 5–6
 - hierarchy, 6
 - modularity, 6
 - regularity, 6
 - Digital system implementation, 515–548
 - 74xx series logic. *See* 74xx series logic
 - application-specific integrated circuits (ASICs), 523
 - data sheets, 523–528
 - economics, 546–548
 - logic families, 529–531
 - overview, 515
 - packaging and assembly, 531–534
 - breadboards, 532
 - packages, 531–532
 - printed circuit boards, 533–534
 - programmable logic, 516–523
 - transmission lines. *See* Transmission lines
 - Diodes, 27–28
 - DIP. *See* Dual-inline package
 - Direct current (DC), 23, 24
 - transfer characteristics, 23–24, 25
 - Direct mapped cache, 470–474
 - Dirty bit, 482–483
 - Discipline, 5–6
 - Disk. *See* Hard disk
 - divide (*div*), 308
 - divide unsigned (*divu*), 339
 - Divider, 247–248
 - Division, 247–248
 - floating-point, 253
 - instructions, 308
 - Divisor, 247
 - Don't care (X), 65
 - Dopant atoms, 27
 - Double. *See* Double-precision floating-point numbers
 - Double-precision floating point numbers, 251–252
 - DRAM. *See* Dynamic random access memory
 - Driver. *See* Device driver
 - Dual-inline package (DIP), 28, 531
 - DUT. *See* Device under test
 - Dynamic discipline, 134
 - Dynamic data segment, 331
 - Dynamic random access memory (DRAM), 257, 260, 463
- E**
- Edison, Thomas, 169
 - Edge-triggered digital systems, 108, 112
 - Equations
 - simplification, 61–62
 - Electrically erasable programmable read only memory (EEPROM), 263
 - Enabled registers, 193
 - HDL for, 193
 - EPC. *See* Exception Program Counter register
 - Erasable programmable read only memory (EPROM), 263
 - Exceptions, 337–339
 - Exception handler, 337–338
 - Exception program counter (EPC), 337–338
 - Exclusive or. *See* XOR
 - Executable file, 334
 - Execution time, 367
 - Exponent, 250–253
- F**
- Failures, 144–146
 - FDIV bug, 253
 - FET. *See* Field effect transistors
 - Field programmable gate array (FPGA), 268–272, 521–523
 - Field effect transistors, 26
 - FIFO. *See* First-in-first-out queue. *See also* Queue
 - Finite state machines (FSMs), 117–133
 - design example, 117–123
 - divide-by-3, 207–208
 - factoring, 129–132
 - HDLs and, 206–213
 - Mealy machines. *See* Mealy machines
 - Moore machines. *See* Moore machines
 - Moore versus Mealy machines, 126–129
 - state encodings, 123–126
 - state transition diagram, 118–119
 - First-in-first-out (FIFO) queue, 508
 - Fixed-point numbers, 249–250
 - Flash memory, 263–264
 - Flip-flops, 103–112, 257. *See also* Registers
 - comparison with latches, 106, 112
 - D, 108
 - enabled, 109–110
 - register, 108–109
 - resettable, 110, 427
 - asynchronous, 192
 - synchronous, 192
 - transistor count, 108
 - transistor-level, 110–111
 - Floating (*Z*), 69–71
 - Floating-point numbers, 250–253

- addition, 252–255. *See also* Addition
 - converting binary or decimal to. *See* Number conversions
 - division, 253. *See also* Division
 - double-precision, 251–252
 - FDIV bug. *See* FDIV bug
 - floating-point unit (FPU), 253
 - instructions, 340–341
 - rounding, 252
 - single-precision, 251–252
 - special cases
 - infinity, 251
 - not a number (NaN), 251
 - Forwarding, 408–409
 - Fractional numbers, 274
 - FPGA. *See* Field programmable gate array
 - Frequency, 135. *See also* Clock period
 - FSM. *See* Finite state machines
 - Full adder, 52, 178. *See also* Adder, Addition
 - HDL using always/process, 197
 - using nonblocking assignments, 204
 - Fully associative cache, 475–476
 - Funct field, 299–300
 - Functional specification, 51
 - Functions. *See* Procedure calls
 - Fuse, 263
- G**
- Gates, 19–22
 - AND, 20–22, 32–33
 - NAND, 21, 32
 - NOR, 21, 32–33
 - OR, 21
 - transistor-level implementation, 262
 - XOR, 21
 - XNOR, 21
 - Gedanken-Experiments on Sequential Machines (Moore), 111
 - Generate signal, 235, 237
 - Glitches, 75, 88–91
 - Global pointer (\$gp), 294, 331. *See also* Static data segment
 - Gray, Frank, 65
 - Gray codes, 65, 72
 - Gulliver's Travels (Swift), 297
- H**
- Half word, 339
 - Half adder, 233–234
 - Hard disk, 484
 - Hard drive. *See* Hard disk
 - Hardware reduction, 66–67
 - Hardware description languages (HDLs), 167–230
 - assignment statements, 177
 - behavioral modeling, 171–184
 - combinational logic, 171–185, 195–206
 - bit swizzling, 182
 - bitwise operators, 171–174
 - blocking and nonblocking assignments, 201–206
 - case statements, 198–199
 - conditional assignment, 175–176
 - delays, 182–183
 - if statements, 199–201
 - internal variables, 176–178
 - numbers, 179
 - precedence, 178–179
 - reduction operators, 174
 - synthesis tools. *See* Synthesis Tools
 - Verilog, 201
 - VHDL libraries and types, 183–185
 - Z's and X's, 179–182
 - finite state machines, 206–213
 - generate statement, 213
 - generic building blocks, 426
 - invalid logic level, 181
 - language origins, 168–169
 - modules, 167–168
 - origins of, 168–169
 - overview, 167
 - parameterized modules, 211–213
 - representation, 421–431
 - sequential logic, 190–195, 205
 - enabled registers, 193
 - latches, 195
 - multiple registers, 194–195
 - registers, 190–191. *See also* Registers
 - resettable registers, 191–193
 - simulation and synthesis, 169–171
 - single-cycle processor, 422
 - structural modeling, 185–189
 - testbenches, 214–218, 428–431
- Hazards**, 75, 406–418. *See also* Glitches
 - control hazards, 413–416
 - data hazards 408–413
 - solving, 408–416
 - forwarding, 408–410
 - stalls, 410–413
 - WAR. *See* Write after read
 - WAW. *See* Write after write
- Hazard unit, 408, 411, 419
- Heap, 331
- HDLs. *See* Hardware description languages
- Hennessy, John, 290, 364
- Hexadecimal numbers, 11–13
 - to binary conversion table, 12
- Hierarchy, 6, 189
- HIGH, 23. *See also* 1, ON
- High-level programming languages, 290–294
 - translating into assembly, 290–291
 - compiling, linking, and launching, 330–331
- Hit, 466
- High impedance. *See* Floating, Z
- High Z. *See* Floating, High impedance, Z
- Hold time, 133–154
- I**
- I-type instruction, 301–302
- IA-32 microprocessor, 290, 341–349, 447–453
 - branch conditions, 346
 - cache systems, 499–500
 - encoding, 346–348
 - evolution, 448, 500
 - instructions, 344, 345
 - memory and input/output (I/O) systems, 499–502
 - operands, 342–344
 - programmed I/O, 502
 - registers, 342
 - status flags, 344
 - virtual memory, 501
- IEEE 754 floating-point standard, 251
- Idempotency, 58
- Idioms, 171
- IEEE, 169

- If statements, 199–201, 310
 - If/else statements, 311
 - ILP. *See* Instruction-level parallelism
 - Immediates, 298
 - Immediate addressing, 327
 - Information, amount of, 8
 - IorD, 385
 - I/O (input/output), 337, 494–502
 - communicating with, 494–495
 - device driver, 496, 498
 - devices, 494–496. *See also* Peripheral devices
 - memory interface, 494, 502
 - memory-mapped I/O, 494–499
 - Inputs, asynchronous, 144–145
 - Instruction encoding. *See* Machine Language
 - Instruction register (IR), 383, 390
 - Instruction set architecture (ISA), 289–361. *See also* MIPS instruction set architecture
 - Input/output blocks (IOBs), 268
 - Input terminals, 51
 - Institute of Electrical and Electronics Engineers, 250
 - Instruction decode, 401–402
 - Instruction encoding. *See* Instruction format
 - Instruction format
 - F-type, 340
 - I-type, 301–302
 - J-type, 302
 - R-type, 299–300
 - Instruction-level parallelism (ILP), 443, 446
 - Instruction memory, 365
 - Instruction set. *See* Instruction set architecture
 - Instructions. *See also* Language
 - arithmetic/logical, 304–308
 - floating-point, 340–341
 - IA-32, 344–346
 - I-type, 301–302
 - J-type, 302
 - loads. *See* Loads
 - multiplication and division, 308
 - pseudoinstructions, 336–337
 - R-type, 299–300
 - set less than, 339
 - shift, 306–307
 - signed and unsigned, 338–339
 - Intel, 30, 111, 290, 348, 367
 - Inverter. *See* NOT gate
 - Integrated circuits (ICs), 26, 137, 515, 532
 - costs, 137, 169
 - manufacturing process, 515
 - Intel. *See* IA-32 microprocessors
 - Interrupts. *See* Exceptions
 - An Investigation of the Laws of Thought (Boole), 8
 - Involution, 58
 - IOBs. *See* Input/output blocks
 - I-type instructions, 301–302
- J**
- Java, 316. *See also* Language
 - JTA. *See* Jump target address
 - J-type instructions, 302
 - Jump, 309–310. *See also* Branch, unconditional, Programming
 - Jump target address (JTA), 329
- K**
- K-maps. *See* Karnaugh maps
 - Karnaugh, Maurice, 64, 71
 - Karnaugh maps (K-maps), 71–79
 - logic minimization using, 73–76
 - prime implicants, 61, 74
 - seven-segment display decoder, 75–77
 - with “don’t cares,” 78
 - without glitches, 91
 - Kilby, Jack, 26
 - Kilobyte, 14
 - K-maps. *See* Karnaugh maps
- L**
- Labels, 308–309
 - Language. *See also* Instructions
 - assembly, 290–299
 - high-level, 290–294
 - machine, 299–304
 - mnemonic, 291
 - translating assembly to machine, 300
 - Last-in-first-out (LIFO) queue, 321. *See also* Stack, Queue
 - Latches, 103–112
 - comparison with flip-flops, 106, 112
 - D, 107
 - SR, 105–107
 - transistor-level, 110–111
 - Latency, 149
 - Lattice, 27
 - Leaf procedures, 324–325
 - Least recently used (LRU) replacement, 478–479
 - Least significant bit (lsb), 13
 - Least significant byte (LSB), 296
 - LIFO. *See* Last-in-first-out queue
 - Literal, 54, 61, 167
 - Little-Endian, 296–297
 - Load, 255
 - byte (l b), 317
 - byte unsigned (l bu), 317
 - half (l h), 339
 - immediate (l i), 336
 - upper immediate (l ui), 308
 - word (l w), 295–296
 - Loading, 335
 - Locality, 464
 - Local variables, 326–327
 - Logic, 62–65. *See also* Multilevel combinational logic; Sequential logic design
 - bubble pushing. *See* Bubble pushing
 - combinational. *See* Combinational logic
 - families, 529–531
 - gates. *See* Logic gates
 - hardware reduction. *See* Hardware reduction, Equation simplification
 - multilevel. *See* Multilevel combinational logic
 - programmable, 516–523
 - sequential. *See* Sequential logic
 - synthesis, 170–171
 - two-level, 65–66
 - using memory arrays, 264. *See also* Logic arrays
 - Logic arrays, 266–274
 - field programmable gate array, 268–272
 - programmable logic array, 266–268
 - transistor-level implementations, 273–274
 - Logic families, 25, 529–531

- compatibility, 26
 - specifications, 529, 531
 - Logic gates, 19–22, 173
 - buffer, 20
 - delays, 183
 - multiple-input gates, 21–22
 - two-input gates, 21
 - types
 - AND. *See* AND gate
 - AOI (and-or-invert).
 - See* And-or-invert gate
 - NAND. *See* NAND gate
 - NOR. *See* NOR gate
 - NOT. *See* NOT gate
 - OAI (or-and-invert).
 - See* Or-and-invert gate
 - OR. *See* OR gate
 - XOR. *See* XOR gate
 - XNOR. *See* XNOR gate
 - Logic levels, 22–23
 - Logical operations, 304–308
 - Lookup tables (LUTs), 268
 - Loops, 311–314
 - for, 313
 - while, 312–313
 - LOW, 23. *See also* 0, OFF
 - Low Voltage CMOS Logic (LVCMOS), 25
 - Low Voltage TTL Logic (LVTTTL), 25
 - LRU. *See* Least recently used replacement
 - LSB. *See* Least significant byte
 - LUTs. *See* Lookup tables
 - LVCMOS. *See* Low Voltage CMOS Logic
 - LVTTTL. *See* Low Voltage TTL Logic
- M**
- Machine language, 299–304
 - function fields, 299
 - interpreting code, 302–303
 - I-type instructions, 301–302
 - J-type instructions, 302
 - opcodes, 299–303
 - R-type instructions, 299–300
 - stored programs, 303–304
 - translating from assembly language, 300–302
 - translating into assembly language, 303
 - Main decoder, 374–379
 - Main memory, 466–469
 - Mapping, 470
 - Mantissa, 250, 252–253. *See also* Floating-point numbers
 - Masuoka, Fujio, 263
 - MCM. *See* Multichip module
 - Mealy, George H., 111
 - Mealy machines, 126–129, 130, 210
 - combined state transition and output table, 127
 - state transition diagram, 118–119
 - timing diagram, 131
 - Mean time between failures (MTBF), 146
 - Memory, 51, 295–298
 - access, 298
 - average memory access time (AMAT), 467
 - cache. *See* Caches
 - DRAM. *See* Dynamic random access memory
 - hierarchy, 466
 - interface, 464
 - main, 466–469
 - map, 330–331
 - dynamic data segment, 331
 - global data segment, 330–331
 - reserved segment, 331
 - text segment, 330
 - nonvolatile, 259–260
 - performance, 465
 - physical, 466, 485–486
 - protection, 491. *See also* Virtual memory
 - RAM, 259
 - ROM, 259
 - separate data and instruction, 430–431
 - shared, 71
 - stack. *See* Stack
 - types
 - flip-flops, 105–112
 - latches, 105–112
 - DRAM, 257
 - registers, 108–109
 - register file, 261–262
 - SRAM, 257
 - virtual, 466. *See also* Virtual memory
 - volatile, 259–261
 - word-addressable, 29. *See* Word-addressable memory
 - Memory arrays, 257–266
 - area, 261
 - bit cells, 258
 - delay, 261
 - DRAM. *See* Dynamic random access memory
 - HDL code for, 264–266
 - logic implementation using, 264. *See also* Logic arrays
 - organization, 258
 - overview, 257–260
 - ports, 259
 - register files built using, 261–262
 - types, 259–260
 - DRAM. *See* Dynamic random access memory
 - ROM. *See* Read only memory
 - SRAM. *See* Static random access memory
 - Memory-mapped I/O (input/output), 494–498
 - address decoder, 495–496
 - communicating with I/O devices, 495–496
 - hardware, 495
 - speech synthesizer device driver, 498
 - speech synthesizer hardware, 496–497
 - SP0256, 496
 - Memory protection, 491
 - Memory systems, 463–512
 - caches. *See* Caches
 - IA-32, 499–502
 - MIPS, 470–478
 - overview, 463–467
 - performance analysis, 467–468
 - virtual memory. *See* Virtual memory
 - Mercedes Benz, 268
 - Metal-oxide-semiconductor field effect transistors (MOSFETs), 26–31. *See also* CMOS, nMOS, pMOS, transistors
 - Metastability, 143–144
 - MTBF. *See* Mean time between failures
 - metastable state, 143
 - probability of failure, 145
 - resolution time, 144
 - synchronizers, 144–146
 - A Method of Synthesizing Sequential Circuits (Mealy), 111

- Microarchitecture, 290, 363–461
 - advanced. *See* Advanced microarchitecture
 - architectural state. *See* Architectural State. *See also* Architecture
 - design process, 364–366
 - exception handling, 431–434
 - HDL representation, 421–431.
 - IA-32. *See* IA-32 microprocessor instruction set. *See* Instruction set
 - MIPS. *See* MIPS microprocessor overview, 363–366
 - performance analysis, 366–368
 - types,
 - advanced. *See* Advanced microarchitecture
 - multicycle. *See* Multicycle MIPS processor
 - pipelined. *See* Pipelined MIPS processor
 - single-cycle. *See* Single-cycle MIPS processor
- Microprocessors, 3, 13
 - advanced. *See* Advanced microarchitecture
 - chips. *See* Chips
 - clock frequencies, 124
 - IA-32. *See* IA-32 microprocessor instructions. *See* MIPS instructions, IA-32 instructions
 - MIPS. *See* MIPS microprocessor
- Microsoft Windows, 501
- Minterms, 54
- Maxterms, 54
- MIPS (Millions of instructions per second). *See* Millions of instructions per second
- MIPS architecture. *See* MIPS instruction set architecture (ISA)
- MIPS assembly language. *See also* MIPS instruction set architecture
 - addressing modes, 327–329
 - assembler directives, 333
 - instructions, 290–292
 - logical instructions, 304–308
 - mnemonic, 291
 - operands, 292–298
 - procedure calls, 319–327
 - table of instructions, 336
 - translating machine language to, 303
 - translating to machine language, 300
- MIPS instruction set architecture (ISA)
 - addressing modes, 327–329
 - assembly language, 290–299
 - compiling, assembling, and loading, 330–335
 - exceptions, 337–338
 - floating-point instructions, 340–341
 - IA-32 instructions, 344–346
 - machine language, 299–304
 - MIPS instructions, 290–292
 - overview, 289–290
 - programming, 304–327
 - pseudoinstructions, 336–337
 - signed and unsigned instructions, 338–339
 - SPARC, 364
 - translating and starting a program. *See* Translating and starting a program
- MIPS instructions, 551–554
 - formats
 - F-type, 340
 - I-type, 301–302
 - J-type, 302
 - R-type, 299–300
 - tables of, 552–554
 - opcodes, 552
 - R-type funct fields, 553–554
 - types,
 - arithmetic, 304–308
 - branching. *See* Branching
 - division, 308
 - floating-point, 340–341
 - logical, 304–308
 - multiplication, 308
 - pseudoinstructions, 336–337
- MIPS microprocessor, 364
 - ALU, 242–244
 - multicycle. *See* Multicycle MIPS processor
 - pipelined. *See* Pipelined MIPS processor
 - single-cycle. *See* Single-cycle MIPS processor
- MIPS processor. *See* MIPS microprocessor
- MIPS registers, 293–294, 308
 - nonpreserved, 322–324
 - preserved, 322–324
 - table of, 294
- MIPS single-cycle HDL implementation, 421–431
 - building blocks, 426–428
 - controller, 423
 - datapath, 425
 - testbench, 429
 - top-level module, 430
- Misses, 466, 481
 - AMAT. *See* Average memory access time
 - cache, 466
 - capacity, 481
 - compulsory, 481
 - conflict, 481
 - page fault, 485
 - Miss penalty, 476
 - Miss rate, 467
- Mnemonic, 291
- Modeling, structural. *See* Structural modeling
- Modeling, behavioral. *See* Behavioral modeling
- Modularity, 6, 168
- Modules, in HDL 167–168. *See also* Hardware description languages
 - behavioral, 168, 171
 - parameterized, 211–213
 - structural, 168
- Moore, Edward F., 111
- Moore, Gordon, 30
- Moore machine, 126–129, 130, 208, 209
 - output table, 128
 - state transition diagram, 127
 - state transition table, 128
 - timing diagram, 131
- Moore's law, 30
- MOSFETs. *See* Metal-oxide-semiconductor field effect transistors
- Most significant bit (msb), 13
- Most significant byte (MSB), 296
- Move from hi (mfhi), 308
- Move from lo (mflo), 308
- Move from coprocessor 0 (mfc0), 338
- msb. *See* Most significant bit
- MSB. *See* Most significant byte
- MTBF. *See* Mean time before failure
- Multichip module (MCM), 499
- Multicycle MIPS processor, 366, 381–400
 - control, 388–394
 - control FSM, 394–395
 - datapath, 382–388
 - performance analysis, 397–400
- Multilevel combinational logic, 65–69. *See also* Logic
- Multilevel page table, 493
- Multiplexers, 79–82, 175, 176, 428

instance, 188
 logic, 80–82
 parameterized, 211. *See also*
 Hardware description
 languages
 symbol and truth table, 79
 timing, 87–88
 with type conversion, 185
 wide, 80
 Multiplicand, 246
 Multiplication, 246–247. *See also*
 Arithmetic, Multiplier
 architecture, 339
 instructions, 308
 Multiplier, 246–247
 multiply (mult), 308, 339
 multiply unsigned (multu), 339
 Multiprocessors, 447
 chip, 448
 Multithreading, 446
 Mux. *See* Multiplexers

N

NaN. *See* Not a number
 Negation, 340. *See also* Taking the
 two's complement
 Not a number (NaN), 251
 NAND gate, 21, 32
 nor, 179
 NOR gate, 21, 32–33
 NOT gate, 24, 172. *See also* Inverter
 in HDL using always/process, 196
 Nested procedure calls, 324–326
 Netlist, 170
 Next state, 115
 Nibbles, 13–14
 nMOS, 28–31
 nMOS transistors, 28–31
 No operation. *See* nop
 Noise margins, 23, 24
 nop, 336
 Noyce, Robert, 26
 Number conversion, 9–19 *See also*
 Number systems, Binary numbers
 binary to decimal, 10–11
 binary to hexadecimal, 12
 decimal to binary, 11
 decimal to hexadecimal, 13
 taking the two's complement, 15, 240
 Number systems, 9–19, 249–253

addition. *See* Addition
 binary numbers. *See* Binary numbers
 comparison of, 18–19
 conversion of. *See* Number
 conversions
 decimal numbers. *See* Decimal
 numbers
 estimating powers of two, 14
 fixed-point, 249–250. *See* Fixed-
 point numbers
 floating-point, 250–253. *See*
 Floating-point numbers
 in hardware description languages,
 179
 hexadecimal numbers. *See*
 Hexadecimal numbers
 negative and positive, 15–19
 rounding, 252. *See also* Floating-
 point numbers
 sign bit, 16
 signed, 15–19. *See also* Signed
 binary numbers
 sign/magnitude numbers. *See*
 Sign/magnitude numbers
 two's complement numbers. *See*
 Two's complement
 numbers
 unsigned,

O

OAI gate. *See* Or-and-invert gate
 Object files, 331–334
 Octal numbers, 180
 OFF, 23. *See also* 0, LOW
 Offset, 295–296
 ON, 23. *See also* 1, HIGH, Asserted
 One-cold, 124
 One-hot, 82
 Opcode, 299
 Operands, 292–298
 Operators
 bitwise, 171–174
 or immediate (ori), 308
 OR gate, 21
 Or-and-invert (OAI) gate, 43
 precedence table of, 179
 reduction, 174
 Out-of-order execution, 443
 Out-of-order processor, 441–443
 Output devices, 466

Output terminals, 51
 Overflow, 15
 detecting, 15
 with addition, 15

P

Page, 485
 size, 485
 Page fault, 485
 Page offset, 486–488
 Page table, 486
 Parity, 22
 Parallelism, 149–153
 pipelining. *See* Pipelining, Pipelined
 MIPS processor
 SIMD. *See* Single instruction
 multiple data unit
 spatial and temporal, 151
 vector processor. *See* Vector processor
 Patterson, David, 290, 364
 PCBs. *See* Printed circuit boards
 PC-relative addressing, 327–328.
 See also Addressing modes
 PCSrc, 281, 387–390
 PCWrite, 385
 Pentium processors, 449–452. *See also*
 Intel, IA-32
 Pentium 4, 452
 Pentium II, 450, 499
 Pentium III, 450, 451, 499
 Pentium M, 453
 Pentium Pro, 450, 499, 501
 Perfect induction, 60
 Performance, 366–368
 Peripheral devices. *See* I/O devices
 Perl programming language, 20
 Physical memory, 466, 485–486
 Physical pages, 486–494
 Physical page number, 486
 Pipelined MIPS processor, 151, 366,
 401–421. *See also* MIPS,
 Architecture, Microarchitecture
 control, 405–406
 datapath, 404
 forwarding, 408–410
 hazards. *See* Hazards
 performance analysis, 418–421
 processor performance
 comparison, 420
 stalls, 410–413. *See also* Hazards
 timing, 402

Pipelining hazards. *See* Hazards
 PLAs. *See* Programmable logic arrays
 Plastic leaded chip carriers (PLCCs), 531–532
 PLCCs. *See* Plastic leaded chip carriers
 PLDs. *See* Programmable logic devices
 pMOS, 29–30
 pMOS transistors, 28–31
 Pointer, 321
 global, 331
 stack, 321
 Pop, 345–346. *See also* Stack
 Ports, 259
 POS. *See* Product of sums form
 Power consumption, 34–35
 Prediction. *See* Branch prediction
 Prefix adder, 237–239
 Preserved registers, 322–324
 Prime implicants, 61, 74
 Printed circuit boards (PCBs), 533–534
 Procedure calls, 319–327
 arguments and variables, 326–327
 nested, 324–326
 preserved versus nonpreserved registers, 323–324
 returns, 319–320
 return values, 320
 stack frame, 322
 stack usage, 321–322
 Processors. *See* Microprocessors
 Product-of-sums (POS) canonical form, 56
 Program counter (PC), 365
 Programmable logic arrays (PLAs), 63, 266–268, 520–521
 Programmable logic devices (PLDs), 268
 Programmable read only memories (PROMs), 516–520. *See also* Read only memories
 Programming, 304–327
 arithmetic/logical instructions. *See* Arithmetic, 304–308
 arrays. *See* Arrays
 branching. *See* Branching
 conditional statements. *See* Conditional statements
 constants, 307–308
 immediates, 298
 loops. *See* Loops
 procedure calls. *See* Procedure calls
 shift instructions, 306–307

 translating and starting a program, 331–335
 Programming languages, 290–294
 Propagation delay, 84
 Protection, memory. *See* Memory protection
 Proving Boolean theorems. *See* Perfect induction
 PROMs. *See* Programmable read only memories
 Pseudo-direct addressing, 328–329. *See also* Addressing modes
 Pseudo-nMOS logic, 33–34. *See also* Transistors
 Pseudoinstructions, 336–337
 Push, 67–69. *See also* Stack

Q

Queue, 321
 FIFO. *See* First-in-first-out queue
 LIFO. *See* Last-in-first-out queue
 Q output. *See* Sequential logic design

R

R-type instructions, 299–300
 RAM. *See* Random access memory
 Random access memory (RAM), 257, 262–264. *See also* Memory arrays
 synthesized, 265
 Read only memory, 199, 257, 262–264. *See also* Memory arrays
 EPROM. *See* Erasable programmable read only memory
 EEPROM. *See* Electrically erasable programmable read only memory
 flash memory. *See* Flash memory
 PROM. *See* Programmable read only memory
 transistor-level implementation, 273
 Read/write head, 484
 Recursive procedure calls, 324–325. *See also* Procedure calls
 Reduced instruction set computer (RISC), 292, 364

Reduction operators, 174. *See also* Hardware description languages, Verilog
 RegDst, 373–374
 Register-only addressing, 327. *See also* Addressing modes
 Register renaming, 443–445. *See also* Advanced microarchitecture
 Register(s), 190–191, 261–262, 292–294. *See also* Flip-flops, Register file
 arguments (\$a0–\$a3)
 assembler temporary (\$at)
 enabled. *See* Enabled registers
 file, 261
 global pointer (\$gp), 331
 multiple, 194–195
 program counter (PC), 365
 preserved and nonpreserved, 322–324
 renaming, 443–445
 resettable. *See* Resettable registers
 asynchronous, 192
 synchronous, 192
 return address (\$ra), 319–320
 Register set, 294. *See also* Register file, MIPS registers, IA-32 registers
 Regularity, 6, 188
 RegWrite, 371
 Replacement policies, 492. *See also* Caches, Virtual memory
 Resettable registers, 191–193
 asynchronous. *See* Asynchronous resettable registers
 synchronous. *See* Synchronous resettable registers
 Return address (\$ra), 319–320
 Ripple-carry adder, 234
 RISC. *See* Reduced instruction set computer
 ROM, *See* Read Only Memory
 Rotators, 244–246
 Rounding, 252

S

Scalar processor, 438
 Scan chains, 255. *See also* Shift registers
 Schematic, 62–65
 Scientific notation, 249–250

- Seek time, 484
 - Segments, memory, 330–331
 - Semiconductor industry, sales, 3
 - Semiconductors, 27. *See also* Transistors
 - CMOS. *See* Complementary metal oxide silicon
 - diodes. *See* Diodes
 - transistors. *See* Transistors
 - MOSFET. *See* Metal oxide silicon field effect transistors
 - nMOS. *See* nMOS transistors
 - pMOS. *See* pMOS transistors
 - pseudo nMOS. *See* Pseudo nMOS
 - Sensitivity list, 190, 191, 196
 - Sequential building blocks. *See* Sequential logic
 - Sequential logic, 103–165, 190–195, 254–257
 - enabled registers, 193
 - latches, 103–112, 195
 - multiple registers, 194–195
 - overview, 103
 - registers, 190–191
 - shift registers, 255–257
 - synchronous, 113–117
 - timing of, 133–149. *See also* Timing
 - set if less than (`slt`), 313
 - set if less than immediate (`slti`), 339
 - set if less than immediate unsigned (`sltiu`), 339
 - set if less than unsigned (`sltu`), 339
 - Setup time, 133, 135–136
 - Seven-segment display decoder, 75–77
 - with “don’t cares,” 78
 - Shared memory, 71
 - Shift amount (`shamt`), 245
 - shift left logical (`sll`), 306
 - shift left logical variable (`sllv`), 306
 - shift right arithmetic (`sra`), 306
 - shift right arithmetic variable (`srav`), 306
 - shift right logical (`srl`), 306
 - shift right logical variable (`srlv`), 306
 - Shifters, 244–246
 - arithmetic, 244
 - logical, 244
 - Shift instructions, 306–307
 - Shift registers, 255–257
 - Short path, 86
 - Sign/magnitude numbers, 15–16
 - Sign bit, 16
 - Sign extension, 18
 - Significand. *See* Mantissa
 - Silicon (Si), 27
 - Silicon dioxide (SO₂), 28
 - Silicon Valley, 26
 - Simplicity, 291–292
 - SIMD. *See* Single instruction multiple data units
 - Single-cycle MIPS processor, 366, 368–381. *See also* MIPS micro-processor, MIPS architecture, MIPS microarchitecture
 - control, 374–377
 - ALU decoder truth table. *See* ALU decoder
 - Main decoder truth table. *See* Main decoder
 - datapath, 368–374
 - HDL representation, 422
 - operation, 376–377 performance analysis, 380–381
 - timing, 402
 - Single instruction multiple data (SIMD) units, 438, 445
 - Slash notation, 53
 - SPARC architecture, 364
 - SRAM. *See* Static random access memory
 - SP0256, 496
 - Spatial locality, 464
 - Speech synthesis, 496–498
 - device driver, 498
 - SP0256, 496
 - Stack, 321–322. *See also* Memory map, Procedure calls, Queue
 - dynamic data segment, 331
 - frame, 322
 - LIFO. *See* Last-in-first-out queue
 - pointer, 321
 - Stalls, 410–413. *See also* Hazards
 - Static discipline, 24–26
 - Static random access memory (SRAM), 257, 260
 - Status flags, 344
 - Stored program concept, 303–304
 - Stores
 - store byte (`sb`), 318
 - store half (`sh`), 553
 - store word (`sw`), 296
 - Strings, 318
 - Structural modeling, 185–189
 - subtract (`sub`), 291
 - subtract unsigned (`subu`), 339
 - Subtraction, 17–18, 240, 291
 - Subtractor, 240
 - Sum-of-products (SOP) canonical form, 54–55
 - Sun Microsystems, 364
 - Superscalar processor, 438–440
 - Supply voltage, 22
 - Swap space, 492
 - Swift, Jonathan, 297
 - Switch/case statements, 311
 - Symbol table, 333
 - Synchronizers, 144–146
 - asynchronous inputs, 146
 - MTBF. *See* Mean time before failure
 - probability of failure. *See* Probability of failure
 - Synchronous resettable registers, 192
 - HDL for, 192
 - Synchronous sequential logic, 113–117
 - problematic circuits, 113–114
 - Synthesis, 78–79, 186, 187, 188, 189, 190, 193, 194, 195, 199, 200
 - Synthesis Tools, 195
- ## T
- Taking the two’s complement, 16
 - Temporal locality, 464
 - Testbenches, 171, 214–218, 428–431
 - self-checking, 215
 - with test vector file, 216
 - Text segment, 330
 - Theorems. *See* Boolean Theorems
 - Thin small outline package (TSOP), 531
 - Threshold voltage, 29
 - Throughput, 149
 - Timing
 - analysis, 137–138
 - delay, 86–87
 - glitches, 88–91
 - of sequential logic, 133–149
 - clock skew, 140–143
 - dynamic discipline, 134
 - hold time. *See* Hold time
 - hold time constraint, 136–137. *See* Hold time constraint, Hold time violation
 - hold time violation. *See* Hold time violation, Hold time constraint, 139

Timing (*Continued*)

- metastability, 143–144
 - setup time. *See* Setup time
 - setup time constraint, 135–136
 - setup time violations
- resolution time, 146–149. *See also* Metastability
- synchronizers, 144–146
- system timing, 135–140
- specification, 51
- TLB. *See* Translation lookaside buffer
- Token, 149
- Transistors, 23, 28–31, 34
 - CMOS. *See* Complement metal oxide silicon
 - nMOS. *See* nMOS
 - pMOS. *See* pMOS
- Transistor-Transistor Logic (TTL), 25
- Translation lookaside buffer (TLB), 490
- Translating and starting a program, 331–336
- Transmission gates, 33. *See also* Transistors
- Transmission lines, 534–546
 - reflection coefficient, 544–545
 - Z0, 543–544
 - matched termination, 536–538
 - mismatched termination, 539–541
 - open termination, 538–539
 - proper terminations, 542
 - short termination, 539
 - when to use, 41–542
- Tristate buffer, 70–71
- Truth tables, 55, 56, 60, 61, 177
 - ALU decoder, 376
 - “don’t care,” 77
 - main decoder, 376, 379
 - multiplexer, 79
 - seven-segment display decoder, 76
 - SR latch, 106
 - with undefined and floating inputs, 181
- TSOP. *See* Thin small outline package
- TTL. *See* Transistor-Transistor Logic
- Two-level logic, 65–66
- Two’s complement numbers, 16–18. *See also* Binary numbers

U

- Unconditional branches, 308. *See also* Jumps
- Unicode, 316. *See also* ASCII
- Unit under test (UUT), 201
- Unity gain points, 24
- Unsigned numbers, 18
- Use bit, 478–479
- UUT. *See* Unit under test

V

- Valid bit, 472. *See also* Caches, Virtual memory
- Vanity Fair (Carroll), 65
- V_{CC}, 23
- V_{DD}, 23
- Vector processors, 438. *See also* Advanced microarchitecture
- Verilog, 167, 169, 172, 173, 174, 175, 176, 178, 180, 181, 201, 203, 205
 - 3:8 decoder, 199
 - accessing parts of busses, 189
 - adder, 426
 - ALU decoder, 424
 - AND, 168
 - architecture body, 168
 - assign statement, 168, 197, 178
 - asynchronous reset, 192
 - bad synchronizer with blocking assignments, 206
 - bit swizzling, 182
 - blocking assignment, 202
 - case sensitivity, 174
 - casez, 201
 - combinational logic, 168, 202
 - comments, 174
 - comparators, 242
 - continuous assignment statement, 173
 - controller, 423
 - counter, 254
 - datapath, 425
 - default, 198
 - divide-by-3 finite state machine, 207, 208
 - D latch, 195
 - eight-input AND, 174
 - entity declaration, 168
 - full adder, 178
 - using always/process, 197
 - using nonblocking assignments, 204
 - IEEE_STD_LOGIC_1164, 168, 183
 - IEEE_STD_LOGIC_SIGNED, 183
 - IEEE_STD_LOGIC_UNSIGNED, 183
 - inverters, 172
 - using always/process, 196
 - left shift, 427
 - library use clause, 168
 - logic gates, 173
 - with delays, 183
 - main decoder, 424
 - MIPS testbench, 429
 - MIPS top-level module, 430
 - multiplexers, 175, 176, 428
 - multiplier, 247
 - nonblocking assignment, 191, 202
 - NOT, 168
 - numbers, 180
 - operator precedence, 179
 - OR, 168
 - parameterized
 - N:2^N decoder, 212
 - N-bit multiplexer, 211
 - N-input AND gate, 213
 - pattern recognizer
 - Mealy FSM, 210
 - Moore FSM, 209
 - priority circuit, 201
 - RAM, 265
 - reg, 191, 194
 - register, 191
 - register file, 42
 - resettable flip-flop, 427
 - resettable enabled register, 193
 - resettable register, 192
 - ROM, 266
 - self-checking testbench, 215
 - seven-segment display decoder, 198
 - shift register, 256
 - sign extension, 427
 - single-cycle MIPS processor, 422
 - STD_LOGIC, 168, 183
 - statement, 173
 - structural models

- 2:1 multiplexer, 188
 - 4:1 multiplexer, 187
 - subtractor, 241
 - synchronizer, 194
 - synchronous reset, 192
 - testbench, 214
 - with test vector file, 216
 - tristate buffer, 180
 - truth tables with undefined and floating inputs, 181
 - type declaration, 168
 - wires, 178
 - VHDL libraries and types, 167, 169, 172, 173, 174, 175, 176, 178, 180, 181, 183–185, 203, 205
 - 3:8 decoder, 199
 - accessing parts of busses, 189
 - adder, 426
 - architecture, 187
 - asynchronous reset, 192
 - bad synchronizer with blocking assignments, 206
 - bit swizzling, 182
 - boolean, 183
 - case sensitivity, 174
 - clk'event, 191
 - combinational logic, 168
 - comments, 174
 - comparators, 242
 - concurrent signal assignment, 173, 178
 - controller, 423
 - CONV_STD_LOGIC_VECTOR, 212
 - counter, 254
 - datapath, 425
 - decoder, 424
 - divide-by-3 finite state machine, 207, 208
 - D latch, 195
 - eight-input AND, 174
 - expression, 173
 - full adder, 178
 - using always/process, 197
 - using nonblocking assignments, 204
 - generic statement, 211
 - inverters, 172
 - using always/process, 196
 - left shift, 427
 - logic gates, 173
 - with delays, 183
 - main decoder, 424
 - MIPS testbench, 429
 - MIPS top-level module, 430
 - multiplexers, 175, 176, 428
 - multiplier, 247
 - numbers, 180
 - operand, 173
 - operator precedence, 179
 - others, 198
 - parameterized
 - N-bit multiplexer, 211
 - N-input AND gate, 213
 - N:2^N decoder, 212
 - pattern recognizer
 - Mealy FSM, 210
 - Moore FSM, 209
 - priority circuit, 201
 - process, 191
 - RAM, 265
 - register, 191
 - register file, 426
 - resettable enabled register, 193
 - resettable flip-flop, 427
 - resettable register, 192
 - RISING_EDGE, 191
 - ROM, 266
 - selected signal assignment statements, 176
 - self-checking testbench, 215
 - seven-segment display decoder, 198
 - shift register, 256
 - sign extension, 427
 - signals, 178, 194
 - simulation waveforms with delays, 170, 183
 - single-cycle MIPS processor, 422
 - STD_LOGIC_ARITH, 212
 - structural models
 - 2:1 multiplexer, 188
 - 4:1 multiplexer, 187
 - subtractor, 241
 - synchronizer, 194
 - synchronous reset, 192
 - testbench, 214
 - with test vector file, 216
 - tristate buffer, 180
 - truth tables with undefined and floating inputs, 181
 - VHSIC, 169
 - Virtual address, 485–490
 - Virtual memory, 484–494
 - address translation, 486–488
 - IA-32, 501
 - memory protection, 491
 - pages, 485
 - page faults, 485
 - page offset, 486–488
 - page table, 488–489
 - multilevel page tables, 492–494
 - replacement policies, 492
 - translation lookaside buffer (TLB), 490. *See* Translation lookaside buffer
 - write policies, 482–483
 - Virtual pages, 485
 - Virtual page number, 487
 - Volatile memory, 259. *See also* DRAM, SRAM, Flip-flops
 - Voltage, threshold, 29
 - V_{ss}, 23
- ## W
- Wafers, 28
 - Wall, Larry, 20
 - WAR. *See* Write after read
 - WAW. *See* Write after write
 - While loop, 312–313
 - White space, 174
 - Whitmore, Georgiana, 7
 - Wire, 63
 - Word-addressable memory, 295
 - Write policies, 482–483
 - Write after read (WAR) hazard, 442. *See also* Hazards
 - Write after write (WAW) hazard, 442–443. *See also* Hazards
- ## X
- XOR gate, 21
 - XNOR gate, 21
 - Xilinx FPGA, 268
 - X. *See* Contention, Don't care.,
- ## Z
- Z., *See* Floating
 - Zero extension, 302