

CD information is listed by chapter and section number followed by page ranges (3.10:6–9). Page references preceded by a single letter refer to appendixes.

- 1-bit ALU, C-26–29
 - adder, C-27
 - CarryOut, C-28
 - illustrated, C-29
 - logical unit for AND/OR, C-27
 - for most significant bit, C-33
 - performing AND, OR, and addition, C-31, C-33
 - See also* Arithmetic logic unit (ALU)
- 32-bit ALU, C-29–38
 - from 31 copies of 1-bit ALU, C-34
 - with 32 1-bit ALUs, C-30
 - defining in Verilog, C-35–38
 - illustrated, C-36
 - ripple carry adder, C-29
 - tailoring to MIPS, C-31–35
 - See also* Arithmetic logic unit (ALU)
- 32-bit immediate operands, 128–29
- 7090/7094 hardware, CD3.10:6

A

- Absolute references, 142
- Abstractions
 - defined, 20
 - hardware/software interface, 20–21
 - principle, 21
- Accumulator architectures, CD2.20:1
- Accumulators, CD2.20:1
- Acronyms, 8

- Addition, 224–29
 - binary, 224–25
 - floating-point, 250–54, 259, B-73–74
 - instructions, B-51
 - operands, 225
 - significands, 250
 - speed, 229
 - See also* Arithmetic
- Address-control lines, D-26
- Addresses
 - 32-bit immediates, 128–36
 - base, 83
 - byte, 84
 - defined, 82
 - memory, 91
 - virtual, 493–95, 514
- Addressing
 - 32-bit immediates, 128–36
 - base, 133
 - displacement, 133
 - intermediate, 132, 133
 - in jumps and branches, 129–32
 - MIPS modes, 132–33
 - PC-relative, 130, 133
 - pseudodirect, 133
 - register, 132, 133
 - x86 modes, 168, 170
- Addressing modes, B-45–47
 - desktop architectures, E-6
 - embedded architectures, E-6
- Address select logic, D-24, D-25
- Address space, 492, 496
 - extending, 545
 - flat, 545
 - ID (ASID), 510
 - inadequate, CD5.13:5
 - shared, 639–40
 - single physical, 638
 - unmapped, 514
 - virtual, 510
- Address translation
 - AMD Opteron X4, 540
 - defined, 493
 - fast, 502–4
 - Intel Nehalem, 540
 - TLB for, 502–4
- Add unsigned instruction, 226
- Advanced Technology Attachment (ATA) disks, 577, 613, 614
- AGP, A-9
- Algol-60, CD2.20:6–7
- Aliasing, 508
- Alignment restriction, 84
- All-pairs N-body algorithm, A-65
- Alpha architecture
 - bit count instructions, E-29
 - defined, 527
 - floating-point instructions, E-28
 - instructions, E-27–29
 - no divide, E-28
 - PAL code, E-28
 - unaligned load-store, E-28
 - VAX floating-point formats, E-29
- ALU control, 316–18
 - bits, 317
 - logic, D-6
 - mapping to gates, D-4–7
 - truth tables, D-5
 - See also* Arithmetic logic unit (ALU)
- ALU control block, 320
 - defined, D-4
 - generating ALU control bits, D-6
- ALUOp, 316, D-6
 - bits, 317, 318
 - control signal, 320
- AMD64, 167, CD2.20:5

- Amdahl's law, 477, 635
 - corollary, 52
 - defined, 51
 - fallacy, 684
 - AMD Opteron X4 (Barcelona), 20, 44–50, 300
 - address translation, 540
 - architectural registers, 404
 - base versus fully optimized performance, 683
 - caches, 541
 - characteristics, 677
 - CPI, miss rates, and DRAM accesses, 542
 - defined, 677
 - illustrated, 676
 - LBMHD performance, 682
 - memory hierarchies, 540–43
 - microarchitecture, 404, 405
 - miss penalty reduction techniques, 541–43
 - pipeline, 404–6
 - pipeline illustration, 406
 - roofline model, 678
 - shared L3 cache, 543
 - SPEC CPU benchmark, 48–49
 - SPEC power benchmark, 49–50
 - SpMV performance, 681
 - TLB hardware, 540
 - American Standard Code for Information Interchange. *See* ASCII
 - AND gates, C-12, D-7
 - AND operation, 103–4, B-52, C-6
 - Annual failure rate (AFR), 573, 613
 - Antidependence, 397
 - Antifuse, C-78
 - Apple computer, CD1.10:6–7
 - Application binary interface (ABI), 21
 - Application programming interfaces (APIs)
 - defined, A-4
 - graphics, A-14
 - Architectural registers, 404
 - Arithmetic, 222–83
 - addition, 224–29
 - division, 236–42
 - floating point, 242–70
 - for multimedia, 227–28
 - multiplication, 230–36
 - subtraction, 224–29
 - Arithmetic instructions
 - desktop RISC, E-11
 - embedded RISC, E-14
 - logical, 308
 - MIPS, B-51–57
 - operands, 80
 - See also* Instructions
 - Arithmetic intensity, 668
 - Arithmetic logic unit (ALU)
 - 1-bit, C-26–29
 - 32-bit, C-29–38
 - before forwarding, 368
 - branch datapath, 312
 - hardware, 226
 - memory-reference instruction use, 301
 - for register values, 308
 - R-format operations, 310
 - signed-immediate input, 371
 - See also* ALU control; Control units
 - ARM instructions, 161–65
 - 12-bit immediate field, 164
 - addressing modes, 161–63
 - block loads and stores, 165
 - brief history, CD2.20:4
 - calculations, 161–63
 - compare and conditional branch, 163–64
 - condition field, 383
 - data transfer, 162
 - features, 164–65
 - formats, 164
 - logical, 165
 - MIPS similarities, 162
 - register-register, 162
 - unique, E-36–37
 - ARPANET, CD6.14:7
 - Arrays
 - logic elements, C-18–19
 - multiple dimension, 266
 - pointers versus, 157–61
 - procedures for setting to zero, 158
 - ASCII
 - binary numbers versus, 123
 - character representation, 122
 - defined, 122
 - symbols, 126
 - Assembler directives, B-5
 - Assemblers, 140–42, B-10–17
 - conditional code assembly, B-17
 - defined, 11, B-4
 - function, 141, B-10
 - macros, B-4, B-15–17
 - microcode, D-30
 - number acceptance, 141
 - object file, 141–42
 - pseudoinstructions, B-17
 - relocation information, B-13, B-14
 - speed, B-13
 - symbol table, B-12
 - Assembly language
 - defined, 11, 139
 - drawbacks, B-9–10
 - floating-point, 260
 - high-level languages versus, B-12
 - illustrated, 12
 - MIPS, 78, 98–99, B-45–80
 - production of, B-8–9
 - programs, 139
 - translating into machine language, 98–99
 - when to use, B-7–9
 - Asserted signals, 305, C-4
 - Associativity
 - in caches, 482–83
 - degree, increasing, 481, 518
 - floating-point addition, testing, 270–71
 - increasing, 486–87
 - set, tag size versus, 486–87
 - Asynchronous interconnect, 583
 - Atomic compare and swap, 139
 - Atomic exchange, 137
 - Atomic fetch-and-increment, 139
 - Atomic memory operation, A-21
 - Attribute interpolation, A-43–44
 - Availability, 573
 - Average memory access time (AMAT), 478
 - calculating, 478–79
 - defined, 478
- ## B
- Backpatching, B-13
 - Backplane bus, 582
 - Backups, 615–16
 - Bandwidth
 - bisection, 661
 - external to DRAM, 474
 - I/O, 618

- L2 cache, 675
 - memory, 471, 472
 - network, 661
 - Barrier synchronization, A-18
 - defined, A-20
 - for thread communication, A-34
 - Base addressing, 83, 133
 - Base registers, 83
 - Basic block, 108–9
 - Benchmarks
 - defined, 48
 - I/O, 596–98
 - Linpack, 664, CD3.10:3
 - multicores, 657–84
 - multiprocessor, 664–66
 - NAS parallel, 666
 - parallel, 665
 - PARSEC suite, 666
 - SPEC CPU, 48–49
 - SPEC power, 49–50
 - SPECrate, 664
 - SPLASH/SPLASH 2, 664–66
 - Stream, 675
 - Biased notation, 94, 247
 - Big-endian byte order, 84, B-43
 - Binary digits. *See* Bits
 - Binary numbers
 - ASCII versus, 123
 - conversion to decimal numbers, 90
 - conversion to hexadecimal numbers, 96
 - defined, 87
 - Bisection bandwidth, 661
 - Bit error rate (BER), CD6.11:9
 - Bit-interleaved parity, 602
 - Bit maps, 17
 - defined, 16, 87
 - goal, 17
 - storing, 17
 - Bits
 - ALUOp, 317, 318
 - defined, 11
 - dirty, 501
 - done, 588
 - error, 588
 - guard, 266–67
 - patterns, 269
 - reference, 499
 - rounding, 268
 - sign, 90
 - state, D-8
 - sticky, 268
 - valid, 458
 - Blocking assignment, C-24
 - Block-interleaved parity, 602–3
 - Blocks
 - combinational, C-4
 - defined, 454
 - finding, 519–20
 - flexible placement, 479–84
 - least recently used (LRU), 485
 - loads/stores, 165
 - locating in cache, 484–85
 - miss rate and, 465
 - multiword, mapping addresses to, 463–64
 - placement locations, 518–19
 - replacement strategies, 481
 - replacement selection, 485
 - replacement strategies, 520–21
 - spatial locality exploitation, 464
 - state, C-4
 - valid data, 458
 - Boolean algebra, C-6
 - Bounds check shortcut, 110
 - Branch datapath
 - ALU, 312
 - operations, 311
 - Branch delay slots
 - defined, 381
 - scheduling, 382
 - Branch equal, 377
 - Branches
 - addressing in, 129–32
 - compiler creation, 107
 - condition, 313
 - decision, moving up, 377
 - delayed, 111, 313, 343, 377–79, 381, 382
 - ending, 108
 - execution in ID stage, 378
 - pipelined, 378
 - target address, 378
 - unconditional, 106
 - See also* Conditional branches
 - Branch hazards. *See* Control hazards
 - Branch history tables. *See* Branch prediction, buffers
 - Branch instructions, B-59–63
 - jump instruction versus, 328
 - list of, B-60–63
 - pipeline impact, 376
 - Branch not taken
 - assumption, 377
 - defined, 311
 - Branch-on-equal instruction, 326
 - Branch prediction
 - buffers, 380, 381
 - as control hazard solution, 342
 - defined, 341
 - dynamic, 341, 342, 380–83
 - static, 393
 - Branch predictors
 - accuracy, 381
 - correlation, 383
 - information from, 382
 - tournament, 383
 - Branch taken
 - cost reduction, 377
 - defined, 311
 - Branch target
 - addresses, 310
 - buffers, 383
 - Bubbles, 374
 - Bubble Sort, 156
 - Bus-based coherent multiprocessors, CD7.14:6
 - Buses, 584, 585
 - backplane, 582
 - defined, C-19
 - processor-memory, 582
 - synchronous, 583
 - Bytes
 - addressing, 84
 - order, 84, B-43
- ## C
- Cache-aware instructions, 547
 - Cache coherence, 534–38
 - coherence, 534
 - consistency, 535
 - enforcement schemes, 536
 - implementation techniques, CD5.9:10–11
 - migration, 536
 - problem, 534, 535, 538
 - protocol example, CD5.9:11–15
 - protocols, 536
 - replication, 536
 - snooping protocol, 536–537–538
 - snoopy, CD5.9:16
 - state diagram, CD5.9:15

- Cache coherency protocol, CD5.9:11–15
 - finite-state transition diagram, CD5.9:12, CD5.9:14
 - functioning, CD5.9:12
 - mechanism, CD5.9:13
 - state diagram, CD5.9:15
 - states, CD5.9:11–12
 - write-back cache, CD5.9:12
- Cache controllers, 538
 - cache coherency protocol, CD5.9:11–15
 - coherent cache implementation techniques, CD5.9:10–11
 - implementing, CD5.9:1–16
 - snoopy cache coherence, CD5.9:16
 - SystemVerilog, CD5.9:1–9
- Cache hits, 508
- Cache misses
 - block replacement on, 520–21
 - capacity, 523
 - compulsory, 523
 - conflict, 523
 - defined, 465
 - direct-mapped cache, 482
 - fully associative cache, 483
 - handling, 465–66
 - memory-stall clock cycles, 475
 - reducing with flexible block placement, 479–84
 - set-associative cache, 482–83
 - steps, 466
 - in write-through cache, 467
- Cache performance, 475–92
 - calculating, 477
 - hit time and, 478
 - impact on processor performance, 476–77
- Caches, 457–75
 - accessing, 459–65
 - associativity in, 482–83
 - bits in, 463
 - bits needed for, 460
 - contents illustration, 461
 - defined, 20, 457
 - direct-mapped, 457, 459, 463, 479
 - disk controller, 578
 - empty, 460
 - flushing, 595
 - FSM for controlling, 529–39
 - fully associative, 479
 - GPU, A-38
 - inconsistent, 466
 - index, 460
 - Intrinsity FastMATH example, 468–70
 - locating blocks in, 484–85
 - locations, 458
 - memory system design, 471–74
 - multilevel, 475, 487–91
 - nonblocking, 541
 - physically addressed, 508
 - physically indexed, 507
 - physically tagged, 507
 - primary, 488, 489, 492
 - secondary, 488, 489, 492
 - set-associative, 479
 - simulating, 543–44
 - size, 462
 - split, 470
 - summary, 474–75
 - tag field, 460
 - tags, CD5.9:10, CD5.9:11
 - virtually addressed, 508
 - virtually indexed, 508
 - virtually tagged, 508
 - virtual memory and TLB integration, 504–8
 - write-back, 467, 468, 521, 522
 - writes, 466–68
 - write-through, 467, 468, 521, 522
 - See also* Blocks
- Callee, 113, 116
- Callee-saved register, B-23
- Caller, 113
- Caller-saved register, B-23
- Capabilities, CD5.13:7
- Capacity misses, 523
- Carry lookahead, C-38–47
 - 4-bit ALUs using, C-45
 - adder, C-39
 - fast, with first level of abstraction, C-39–40
 - fast, with “infinite” hardware, C-38–39
 - fast, with second level of abstraction, C-40–46
 - plumbing analogy, C-42, C-43
 - ripple carry speed versus, C-46
 - summary, C-46–47
- Carry save adders, 235
- Cause register, 590
 - defined, 386
 - fields, B-34, B-35
 - illustrated, 591
- CDC 6600, CD1.10:6, CD4.15:2
- Central processor unit (CPU)
 - classic performance equation, 35–37
 - coprocessor 0, B-33–34
 - defined, 19
 - execution time, 30, 31, 32
 - performance, 30–32
 - system, time, 30
 - time, 475
 - time measurements, 31
 - user, time, 30
 - See also* Processors
- Cg pixel shader program, A-15–17
- Channel controllers, 593
- Characters
 - ASCII representation, 122
 - in Java, 126–27
- Chips. *See* Integrated circuits (ICs)
- C++ language, CD2.15:26, CD2.20:7
- C language
 - assignment, compiling into MIPS, 79–80
 - compiling, 161, CD2.15:1–2
 - compiling assignment with registers, 81–82
 - compiling while loops in, 107–8
 - sort algorithms, 157
 - translation hierarchy, 140
 - translation to MIPS assembly language, 79
 - variables, 118
- Classes
 - defined, CD2.15:14
 - packages, CD2.15:20
- Clock cycles
 - defined, 31
 - memory-stall, 475, 476
 - number of registers and, 81
 - worst-case delay and, 330
- Clock cycles per instruction (CPI), 33–34, 341
 - one level of caching, 488
 - two levels of caching, 488
- Clocking methodology, 305–7, C-48
 - defined, 305
 - edge-triggered, 305, 306, C-48, C-73
 - level-sensitive, C-74, C-75–76
 - for predictability, 305

- Clock rate
 - defined, 31
 - frequency switched as
 - function of, 40
 - power and, 39
- Clocks, C-48–50
 - edge, C-48, C-50
 - in edge-triggered design, C-73
 - skew, C-74
 - specification, C-57
 - synchronous system, C-48–49
- Clusters, CD7.14:7–8
 - defined, 632, 641, CD7.14:7
 - drawbacks, 642
 - isolation, 644
 - organization, 631
 - overhead in division of memory, 642
 - scientific computing on, CD7.14:7
- Cm*, CD7.14:3–4
- C.mmp, CD7.14:3
- Coarse-grained multithreading, 645–46
- Cobol, CD2.20:6
- Code generation, CD2.15:12
- Code motion, CD2.15:6
- Combinational blocks, C-4
- Combinational control units, D-4–8
- Combinational elements, 304
- Combinational logic, 306, C-3, C-9–20
 - arrays, C-18–19
 - decoders, C-9
 - defined, C-5
 - don't cares, C-17–18
 - multiplexors, C-10
 - ROMs, C-14–16
 - two-level, C-11–14
 - Verilog, C-23–26
- Commands, to I/O devices, 588–89
- Commercial computer development, CD1.10:3–9
- Commit units
 - buffer, 399
 - defined, 399
 - in update control, 402
- Common case fast, 177
- Common subexpression elimination, CD2.15:5
- Communication, 24–25
 - overhead, reducing, 43
 - thread, A-34
- Compact code, CD2.20:3
- Compact disks (CDs), 23, 24
- Comparison instructions, B-57–59
 - floating-point, B-74–75
 - list of, B-57–59
- Comparisons, 108–9
 - constant operands in, 109
 - signed versus unsigned, 110
- Compilers, 139
 - branch creation, 107
 - brief history, CD2.20:8
 - conservative, CD2.15:5–6
 - defined, 11
 - front end, CD2.15:2
 - function, 13, 139, B-5–6
 - high-level optimizations, CD2.15:3–4
 - ILP exploitation, CD4.15:4–5
 - Just In Time (JIT), 148
 - machine language production, B-8–9, B-10
 - optimization, 160, CD2.20:8
 - speculation, 392–93
 - structure, CD2.15:1
- Compiling
 - C assignment statements, 79–80
 - C language, 107–8, 161, CD2.15:1–2
 - floating-point programs, 262–65
 - if-then-else, 106
 - in Java, CD2.15:18–19
 - procedures, 114, 117–18
 - recursive procedures, 117–18
 - while loops, 107–8
- Compressed sparse row (CSR) matrix, A-55, A-56
- Compulsory misses, 523
- Computers
 - application classes, 5–7
 - applications, 4
 - arithmetic for, 222–83
 - characteristics, CD1.10:12
 - commercial development, CD1.10:3–9
 - component organization, 14
 - components, 14, 223, 569
 - design measure, 55
 - desktop, 5, 15
 - embedded, 5–7, B-7
 - first, CD1.10:1–3
 - in information revolution, 4
 - instruction representation, 94–101
 - laptop, 18
 - performance measurement, CD1.10:9
 - principles, 100
 - rack mount, 606
 - servers, 5
- Compute Unified Device Architecture. *See* CUDA programming environment
- Conditional branches
 - ARM, 163
 - changing program counter with, 383
 - compiling if-then-else into, 106
 - defined, 105
 - desktop RISC, E-16
 - embedded RISC, E-16
 - implementation, 112
 - in loops, 130
 - PA-RISC, E-34, E-35
 - PC-relative addressing, 130
 - RISC, E-10–16
 - SPARC, E-10–12
- Conditional move instructions, 383
- Condition field, 383
- Conflict misses, 523
- Constant-manipulating instructions, B-57
- Constant memory, A-40
- Constant operands, 86–87
 - in comparisons, 109
 - frequent occurrence, 87
- Content Addressable Memory (CAM), 485
- Context switch, 510
- Control
 - ALU, 316–18
 - challenge, 384
 - finishing, 327
 - forwarding, 366
 - FSM, D-8–21
 - implementation, optimizing, D-27–28
 - for jump instruction, 329
 - mapping to hardware, D-2–32
 - memory, D-26
 - organizing, to reduce logic, D-31–32
 - pipelined, 359–63
- Control flow graphs, CD2.15:8–9
 - defined, CD2.15:8
 - illustrated examples, CD2.15:8, CD2.15:9

- Control functions
 - ALU, mapping to gates, D-4–7
 - defining, 321
 - PLA, implementation, D-7, D-20–21
 - ROM, encoding, D-18–19
 - for single-cycle implementation, 327
- Control hazards, 339–43, 375–84
 - branch delay reduction, 377–79
 - branch not taken assumption, 377
 - branch prediction as solution, 342
 - defined, 339, 376
 - delayed decision approach, 343
 - dynamic branch prediction, 380–83
 - logic implementation in Verilog, CD4.12:7–9
 - pipeline stalls as solution, 340
 - pipeline summary, 383–84
 - simplicity, 376
 - solutions, 340
 - static multiple-issue processors and, 394
- Control lines
 - asserted, 323
 - in datapath, 320
 - execution/address calculation, 361
 - final three stages, 361
 - instruction decode/register file read, 361
 - instruction fetch, 361
 - memory access, 362
 - setting of, 321, 323
 - values, 360
 - write-back, 362
- Control signals
 - ALUOp, 320
 - defined, 306
 - effect of, 321
 - multi-bit, 322
 - pipelined datapaths with, 359
 - truth tables, D-14
- Control units, 303
 - address select logic, D-24, D-25
 - combinational, implementing, D-4–8
 - with explicit counter, D-23
 - illustrated, 322
 - logic equations, D-11
 - main, designing, 318–26
 - as microcode, D-28
 - MIPS, D-10
 - next-state outputs, D-10, D-12–13
 - output, 316–17, D-10
 - See also* Arithmetic logic unit (ALU)
- Conversion instructions, B-75–76
- Cooperative thread arrays (CTAs), A-30
- Coprocessors
 - coprocessor 0, B-33–34
 - defined, 266
 - move instructions, B-71–72
- Copy back. *See* Write-back
- Core MIPS instruction set, 282
 - abstract view, 302
 - desktop RISC, E-9–11
 - implementation, 300–303
 - implementation illustration, 304
 - overview, 301–3
 - subset, 300–301
 - See also* MIPS
- Cores
 - defined, 41
 - number per chip, 42
- Correcting code, 602
- Correlation predictor, 383
- Cosmic Cube, CD7.14:6
- Count register, B-34
- Cray computers, CD3.10:4, CD3.10:5
- Critical word first, 465
- Crossbar networks, 662
- CTSS (Compatible Time-Sharing System), CD5.13:8
- CUDA programming environment, 659, A-5, CDA.11:5
 - barrier synchronization, A-18, A-34
 - defined, A-5
 - development, A-17, A-18
 - hierarchy of thread groups, A-18
 - kernels, A-19, A-24
 - key abstractions, A-18
 - paradigm, A-19–23
 - parallel plus-scan template, A-61
 - per-block shared memory, A-58
 - plus-reduction implementation, A-63
 - programs, A-6, A-24
 - scalable parallel programming with, A-17–23
 - SDK, 172
 - shared memories, A-18
 - threads, A-36
- D**
- Databases
 - brief history, CD6.14:4
 - Integrated Data Store (IDS), CD6.14:4
 - relational, CD6.14:5
- Datcenters, 5
- Data flow analysis, CD2.15:8
- Data hazards, 336–39, 363–75
 - defined, 336
 - forwarding, 336, 363–75
 - load-use, 338, 377
 - stalls and, 371–74
 - See also* Hazards
- Data layout directives, B-14
- Data-level parallelism, 649
- Data movement instructions, B-70–73
- Data parallel problem decomposition, A-17, A-18
- Datapath elements
 - defined, 307
 - sharing, 313
- Datapaths
 - branch, 311, 312
 - building, 307–16
 - control signal truth tables, D-14
 - control unit, 322
 - defined, 19
 - design, 307
 - exception handling, 387
 - for fetching instructions, 309
 - for hazard resolution via forwarding, 370
 - for jump instruction, 329
 - for memory instructions, 314
 - for MIPS architecture, 315
 - in operation for branch-on-equal instruction, 326
 - in operation for load instruction, 325
 - in operation for R-type instruction, 324
 - operation of, 321–26
 - pipelined, 344–58
 - for R-type instructions, 314, 323
 - single, creating, 313–16
 - single-cycle, 345
 - static two-issue, 395
- Data race, 137
- Data rate, 596
- Data segment, B-13
- Data selectors, 303

- Data structure compression, 680
- Data transfer instructions
 - defined, 82
 - load, 83
 - offset, 83
 - store, 85
 - See also* Instructions
- Deasserted signals, 305, C-4
- Debugging information, B-13
- DEC disk drive, CD6.14:3
- Decimal numbers
 - binary number conversion to, 90
 - defined, 87
- Decision-making instructions, 105–12
- Decoders, C-9
 - defined, C-9
 - two-level, C-65
- Decoding machine language, 134
- DEC PDP-8, CD1.10:5
- Deep Web, CD6.14:8
- Delayed branches, 111
 - as control hazard solution, 343
 - defined, 313
 - embedded RISCs and, E-23
 - for five-stage pipelines, 382
 - reducing, 377–79
 - scheduling limitations, 381
 - See also* Branches
- Delayed decision, 343
- DeMorgan's theorems, C-11
- Denormalized numbers, 270
- Dependencies
 - bubble insertion and, 374
 - detection, 365
 - name, 397
 - between pipeline registers, 367
 - between pipeline registers and ALU inputs, 366
 - sequence, 363
- Design
 - compromises and, 177
 - datapath, 307
 - digital, 406–7
 - I/O system, 598–99
 - logic, 303–7, C-1–79
 - main control unit, 318–26
 - memory hierarchy, challenges, 525
 - pipelining instruction sets, 335
- Desktop and server RISCs
 - addressing modes, E-6
 - architecture summary, E-4
 - arithmetic/logical instructions, E-11
 - conditional branches, E-16
 - constant extension summary, E-9
 - control instructions, E-11
 - conventions equivalent to MIPS core, E-12
 - data transfer instructions, E-10
 - features added to, E-45
 - floating-point instructions, E-12
 - instruction formats, E-7
 - multimedia extensions, E-16–18
 - multimedia support, E-18
 - types of, E-3
 - See also* Reduced instruction set computer (RISC) architectures
- Desktop computers
 - defined, 5
 - illustrated, 15
- D flip-flops, C-51, C-53
- Dicing, 46
- Dies, 46
- Digital design pipeline, 406–7
- Digital signal-processing (DSP)
 - extensions, E-19
- Digital video disks (DVDs), 23, 24
- DIMMs (dual inline memory modules), CD5.13:4
- Direct3D, A-13
- Direct-mapped caches
 - address portions, 484
 - choice of, 520
 - defined, 457, 479
 - illustrated, 459
 - memory block location, 480
 - misses, 482
 - single comparator, 485
 - total number of bits, 463
 - See also* Caches
- Direct memory access (DMA)
 - defined, 592
 - multiple devices, 593
 - setup, 593
 - transfers, 593, 595
- Dirty bit, 501
- Dirty pages, 501
- Disk controllers
 - caches, 578
 - defined, 576
 - time, 576
- Disk read time, 577
- Disk storage, 575–79
 - characteristics, 579
 - densities, 577
 - history, CD6.14:1–4
 - interfaces, 577–78
 - as nonvolatile, 575
 - rotational latency, 576
 - sectors, 575
 - seek time, 575
 - tracks, 575
 - transfer time, 576
- Displacement addressing, 133
- Divide algorithm, 239
- Dividend, 237
- Division, 236–42
 - algorithm, 238
 - dividend, 237
 - divisor, 237
 - faster, 241
 - floating-point, 259, B-76
 - hardware, 237–39
 - hardware, improved version, 240
 - instructions, B-52–53
 - in MIPS, 241–42
 - operands, 237
 - quotient, 237
 - remainder, 237
 - signed, 239–41
 - SRT, 241
 - See also* Arithmetic
- Divisor, 237
- D latches, C-51, C-52
- Done bit, 588
- Don't cares, C-17–18
 - example, C-17–18
 - term, 318
- Double Data Rate RAMs (DDRAMs), 473, C-65
- Double precision
 - defined, 245
 - FMA, A-45–46
 - GPU, A-45–46, A-74
 - representation, 249
 - See also* Single precision
- Double words, 168
- Dynamically linked libraries (DLLs), 145–46
 - defined, 146
 - lazy procedure linkage version, 146, 147

- Dynamic branch prediction, 380–83
 - branch prediction buffer, 380
 - defined, 380
 - loops and, 380
 - See also* Control hazards
 - Dynamic hardware predictors, 341
 - Dynamic multiple-issue processors, 392, 397–400
 - pipeline scheduling, 398–400
 - superscalar, 397
 - See also* Multiple issue
 - Dynamic pipeline scheduling, 399–400
 - commit unit, 399
 - concept, 400
 - defined, 398
 - hardware-based speculation, 400
 - primary units, 399
 - reorder buffer, 399
 - reservation station, 399
 - Dynamic random access memory (DRAM), 453, 471, C-63–65
 - bandwidth external to, 474
 - cost, 23
 - defined, 18–19, C-63
 - DIMM, CD5.13:4
 - Double Date Rate (DDR), 473
 - early board, CD5.13:4
 - GPU, A-37–38
 - growth of capacity, 27
 - history, CD5.13:3–4
 - pass transistor, C-63
 - SIMM, CD5.13:4, CD5.13:5
 - single-transistor, C-64
 - size, 474
 - speed, 23
 - synchronous (SDRAM), 473, C-60, C-65
 - two-level decoder, C-65
- E**
- Early restart, 465
 - Edge-triggered clocking methodology, 305, 306, C-48, C-73
 - advantage, C-49
 - clocks, C-73
 - defined, C-48
 - drawbacks, C-74
 - illustrated, C-50
 - rising edge/falling edge, C-48
 - EDSAC (Electronic Delay Storage Automatic Calculator), CD1.10:2, CD5.13:1–2
 - Eispack, CD3.10:3
 - Electrically erasable programmable read-only memory (EEPROM), 581
 - Elements
 - combinational, 304
 - datapath, 307, 313
 - memory, C-50–58
 - state, 305, 306, 308, C-48, C-50
 - Embedded computers
 - application requirements, 7
 - defined, B-7
 - design, 6
 - growth, CD1.10:11–12
 - Embedded Microprocessor Benchmark Consortium (EEMBC), CD1.10:11–12
 - Embedded RISCs
 - addressing modes, E-6
 - architecture summary, E-4
 - arithmetic/logical instructions, E-14
 - conditional branches, E-16
 - constant extension summary, E-9
 - control instructions, E-15
 - data transfer instructions, E-13
 - delayed branch and, E-23
 - DSP extensions, E-19
 - general purpose registers, E-5
 - instruction conventions, E-15
 - instruction formats, E-8
 - multiply-accumulate approaches, E-19
 - types of, E-4
 - See also* Reduced instruction set computer (RISC) architectures
 - Encoding
 - defined, D-31
 - floating-point instruction, 261
 - MIPS instruction, 98, 135, B-49
 - ROM control function, D-18–19
 - ROM logic function, C-15
 - x86 instruction, 171–72
 - ENIAC (Electronic Numerical Integrator and Calculator), CD1.10:1, CD1.10:2, CD1.10:3, CD5.13:1
 - EPIC, CD4.15:4
 - Error bit, 588
 - Error correction, C-65–67
 - Error detection, 602, C-66
 - Ethernet, 24, 25, CD6.14:8
 - defined, CD6.11:5
 - multiple, CD6.11:6
 - success, CD6.11:5
 - Exception enable, 512
 - Exception handlers, B-36–38
 - defined, B-35
 - return from, B-38
 - Exception program counters (EPCs), 385
 - address capture, 390
 - copying, 227
 - defined, 227, 386
 - in restart determination, 385
 - transferring, 229
 - Exceptions, 384–91, B-35–36
 - association, 390
 - datapath with controls for handling, 387
 - defined, 227, 385
 - detecting, 385
 - event types and, 385
 - imprecise, 390
 - instructions, B-80
 - interrupts versus, 384–85
 - in MIPS architecture, 385–86
 - overflow, 387
 - PC, 509, 511
 - pipelined computer example, 388
 - in pipelined implementation, 386–91
 - precise, 390
 - reasons for, 385–86
 - result due to overflow in add instruction, 389
 - saving/restoring stage on, 515
 - Exclusive OR (XOR) instructions, B-57
 - Executable files, B-4
 - defined, 142
 - linker production, B-19
 - Execute/address calculation
 - control line, 361
 - load instruction, 350
 - store instruction, 352
 - Execute or address calculation stage, 350, 352
 - Execution time
 - CPU, 30, 31, 32
 - pipelining and, 344
 - as valid performance measure, 54
 - Explicit counters, D-23, D-26

- Exponents, 244–45
- EX stage
 - load instructions, 350
 - overflow exception detection, 387
 - store instructions, 353
- External labels, B-10
- F**
- Facilities, B-14–17
- Failures
 - disk, rates, 613–14
 - mean time between (MTBF), 573
 - mean time to (MTTF), 573, 574, 613, 630
 - reasons for, 574
 - synchronizer, C-77
- Fallacies
 - add immediate unsigned, 276
 - Amdahl's law, 684
 - assembly language for performance, 174–75
 - commercial binary compatibility importance, 175
 - defined, 51
 - disk failure rates, 613–14
 - GPUs, A-72–74, A-75
 - low utilization uses little power, 52
 - MTTF, 613
 - peak performance, 684–85
 - pipelining, 407
 - powerful instructions mean higher performance, 174
 - right shift, 275–76
 - See also* Pitfalls
- False sharing, 537
- Fast carry
 - with first level of abstraction, C-39–40
 - with “infinite” hardware, C-38–39
 - with second level of abstraction, C-40–46
- Fast Fourier Transforms (FFT), A-53
- Fiber Distributed Data Interface (FDDI), CD6.14:8
- Fibre Channel Arbitrated Loop (FC-AL), CD6.11:11
- Field programmable devices (FPDs), C-78
- Field programmable gate arrays (FPGAs), C-78
- Fields
 - Cause register, B-34, B-35
 - defined, 95
 - format, D-31
 - MIPS, 96–97
 - names, 97
 - Status register, B-34, B-35
- Filebench, 597
- Files, register, 308, 314, C-50, C-54–56
- File server benchmark (SPECFS), 597
- Fine-grained multithreading, 645, 647
- Finite-state machines (FSMs), 529–34, C-67–72
 - control, D-8–22
 - controllers, 532
 - defined, 531, C-67
 - implementation, 531, C-70
 - Mealy, 532
 - Moore, 532
 - for multicycle control, D-9
 - next-state function, 531, C-67
 - output function, C-67, C-69
 - for simple cache controller, 533
 - state assignment, C-70
 - state register implementation, C-71
 - style of, 532
 - synchronous, C-67
 - SystemVerilog, CD5.9:6–9
 - traffic light example, C-68–70
- Fixed-function graphics pipelines, CDA.11:1
- Flash-based removable memory cards, 23
- Flash memory, 580–82
 - brief history, CD6.14:4
 - characteristics, 23, 580
 - defined, 22, 580
 - as EEPROM, 581
 - NAND, CD6.14:4
 - NOR, 581, CD6.14:4
 - wear leveling, 581
- Flat address space, 545
- Flip-flops
 - defined, C-51
 - D flip-flops, C-51, C-53
- Floating point, 242–70
 - assembly language, 260
 - backward step, CD3.10:3–4
 - binary to decimal conversion, 249
 - branch, 259
 - challenges, 280
 - defined, 244
 - diversity versus portability, CD3.10:2–3
 - division, 259
 - first dispute, CD3.10:1–2
 - form, 245
 - fused multiply add, 268
 - guard digits, 266–67
 - history, CD3.10:1–10
 - IEEE 754 standard, 246, 247
 - immediate calculations, 266
 - instruction encoding, 261
 - machine language, 260
 - MIPS instruction frequency for, 282
 - MIPS instructions, 259–61
 - operands, 260
 - operands variation in x86, 274
 - overflow, 245
 - packed format, 274
 - precision, 271
 - procedure with two-dimensional matrices, 263–65
 - programs, compiling, 262–65
 - registers, 265
 - representation, 244–50
 - rounding, 266–67
 - sign and magnitude, 245
 - SSE2 architecture, 274–75
 - subtraction, 259
 - underflow, 245
 - units, 267
 - in x86, 272–74
- Floating-point addition, 250–54
 - arithmetic unit block diagram, 254
 - associativity, testing, 270–71
 - binary, 251, 253
 - illustrated, 252
 - instructions, 259, B-73–74
 - steps, 250–51
- Floating-point arithmetic (GPUs), A-41–46
 - basic, A-42
 - double precision, A-45–46, A-74
 - performance, A-44
 - specialized, A-42–44
 - supported formats, A-42
 - texture operations, A-44
- Floating-point instructions, B-73–80
 - absolute value, B-73
 - addition, B-73–74
 - comparison, B-74–75

- Floating-point instructions (*continued*)
 - conversion, B-75–76
 - desktop RISC, E-12
 - division, B-76
 - load, B-76–77
 - move, B-77–78
 - multiplication, B-78
 - negation, B-78–79
 - SPARC, E-31
 - square root, B-79
 - store, B-79
 - subtraction, B-79–80
 - truncation, B-80
 - Floating-point multiplication, 255–59
 - binary, 256–57
 - illustrated, 258
 - instructions, 259
 - significands, 255
 - steps, 255–56
 - Floating vectors, CD3.10:2
 - Flow-sensitive information, CD2.15:14
 - Flushing instructions, 377, 378
 - defined, 377
 - exceptions and, 390
 - For loops, 157
 - inner, CD2.15:25
 - SIMD and, CD7.14:2
 - Formal parameters, B-16
 - Format fields, D-31
 - Fortran, CD2.20:6
 - Forwarding, 363–75
 - ALU before, 368
 - control, 366
 - datapath for hazard resolution, 370
 - defined, 336
 - functioning, 364–65
 - graphical representation, 337
 - illustrations, CD4.12:25–30
 - multiple results and, 339
 - multiplexors, 370
 - pipeline registers before, 368
 - with two instructions, 336–37
 - Verilog implementation, CD4.12:3–5
 - Forward references, B-11
 - Fractions, 244, 245, 246
 - Frame buffer, 17
 - Frame pointers, 119
 - Front end, CD2.15:2
 - Fully associative caches
 - block replacement strategies, 521
 - choice of, 520
 - defined, 479
 - memory block location, 480
 - misses, 483
 - See also* Caches
 - Fully connected networks, 661, 662
 - Function code, 97
 - Fused-multiply-add (FMA) operation, 268, A-45–46
- G**
- Game consoles, A-9
 - Gates, C-3, C-8
 - AND, C-12, D-7
 - defined, C-8
 - delays, C-46
 - mapping ALU control function to, D-4–7
 - NAND, C-8
 - NOR, C-8, C-50
 - Gateways, CD6.11:6
 - General Purpose GPUs (GPGPUs), 656, A-5, CDA.11:3
 - General-purpose registers
 - architectures, CD2.20:2–3
 - embedded RISCs, E-5
 - Generate
 - defined, C-40
 - example, C-44
 - super, C-41
 - Gigabytes, 23
 - Global common subexpression
 - elimination, CD2.15:5
 - Global memory, A-21, A-39
 - Global miss rates, 489
 - Global optimization, CD2.15:4–6
 - code, CD2.15:6
 - defined, CD2.15:4
 - implementing, CD2.15:7–10
 - Global pointers, 118
 - GPU computing
 - defined, A-5
 - visual applications, A-6–7
 - See also* Graphics processing units (GPUs)
 - GPU system architectures, A-7–12
 - graphics logical pipeline, A-10
 - heterogeneous, A-7–9
 - implications for, A-24
 - interfaces and drivers, A-9
 - unified, A-10–12
 - Graph coloring, CD2.15:11
 - Graphics displays
 - computer hardware support, 17
 - LCD, 16
 - Graphics logical pipeline, A-10
 - Graphics processing units (GPUs), 654–60
 - as accelerators, 654
 - attribute interpolation, A-43–44
 - computing, CDA.11:4
 - defined, 44, 634, A-3
 - driver software, 655
 - evolution, A-5, CDA.11:2
 - fallacies and pitfalls, A-72–75
 - floating-point arithmetic, A-17, A-41–46, A-74
 - future trends, CDA.11:5
 - GeForce 8-series generation, A-5
 - general computation, A-73–74
 - General Purpose (GPGPUs), 656, A-5, CDA.11:3
 - graphics mode, A-6
 - graphics trends, A-4
 - history, A-3–4
 - logical graphics pipeline, A-13–14
 - main memory, 655
 - mapping applications to, A-55–72
 - memory, 656
 - multilevel caches and, 655
 - N-body applications, A-65–72
 - NVIDIA architecture, 656–59
 - parallelism, 655, A-76
 - parallel memory system, A-36–41
 - performance doubling, A-4
 - perspective, 659–60
 - programmable real-time, CDA.11:2–3
 - programming, A-12–24
 - programming interfaces to, 654, A-17
 - real-time graphics, A-13
 - scalable, CDA.11:4–5
 - summary, A-76
 - See also* GPU computing
 - Graphics shader programs, A-14–15
 - Gresham's Law, 283, CD3.10:1
 - Grids, A-19
 - Guard digits
 - defined, 266
 - rounding with, 267

H

- Half precision, A-42
 - Halfwords, 126
 - Handlers
 - defined, 513
 - TLB miss, 514
 - Handshaking protocol, 584
 - Hard disks
 - access times, 23
 - defined, 22
 - diameters, 23
 - illustrated, 22
 - read-write head, 22
 - Hardware
 - as hierarchical layer, 10
 - language of, 11–13
 - operations, 77–80
 - supporting procedures in, 112–22
 - synthesis, C-21
 - translating microprograms to, D-28–32
 - virtualizable, 527
 - Hardware-based speculation, 400
 - Hardware description languages
 - defined, C-20
 - using, C-20–26
 - VHDL, C-20–21
 - See also* Verilog
 - Hardware multithreading, 645–48
 - coarse-grained, 645–46
 - defined, 645
 - fine-grained, 645, 647
 - options, 646
 - simultaneous, 646–48
 - Harvard architecture, CD1.10:3
 - Hazard detection units, 372
 - functions, 373
 - pipeline connections for, 373
 - Hazards, 335–43
 - control, 339–43, 375–84
 - data, 336–39, 363–75
 - defined, 335
 - forwarding and, 371
 - structural, 335–36, 352
 - See also* Pipelining
 - Heap
 - allocating space on, 120–22
 - defined, 120
 - Heterogeneous systems, A-4–5
 - architecture, A-7–9
 - defined, A-3
 - Hexadecimal numbers, 95–96
 - binary number conversion to, 96
 - defined, 95
 - High-level languages, 11–13, B-6
 - benefits, 13
 - computer architectures, CD2.20:4
 - defined, 12
 - importance, 12
 - High-level optimizations, CD2.15:3–4
 - Hit rate, 454
 - Hit time
 - cache performance and, 478
 - defined, 455
 - Hit under miss, 541
 - Hold time, C-54
 - Horizontal microcode, D-32
 - Hot-swapping, 605
 - Hubs, CD6.11:6, CD6.11:7
 - Hybrid hard disks, 581
- I**
- IBM 360/85, CD5.13:6
 - IBM 370, CD6.14:2
 - IBM 701, CD1.10:4
 - IBM 7030, CD4.15:1
 - IBM ALOG, CD3.10:6
 - IBM Blue Genie, CD7.14:8–9
 - IBM Cell QS20
 - base versus fully optimized performance, 683
 - characteristics, 677
 - defined, 679
 - illustrated, 676
 - LBMHD performance, 682
 - roofline model, 678
 - SpMV performance, 681
 - IBM Personal Computer, CD1.10:7, CD2.20:5
 - IBM System/360 computers, CD1.10:5, CD3.10:4, CD3.10:5, CD5.13:5
 - IBM z/VM, CD5.13:7
 - ID stage
 - branch execution in, 378
 - load instructions, 349
 - store instruction in, 349
 - IEEE 754 floating-point standard, 246, 247, CD3.10:7–9
 - first chips, CD3.10:7–9
 - in GPU arithmetic, A-42–43
 - implementation, CD3.10:9
 - rounding modes, 268
 - today, CD3.10:9
 - See also* Floating point
 - IEEE 802.11, CD6.11:8–10
 - with base stations, CD6.11:9
 - cellular telephony versus, CD6.11:10
 - defined, CD6.11:8
 - Wired Equivalent privacy, CD6.11:10
 - IEEE 802.3, CD6.14:8
 - I-format, 97
 - If statements, 130
 - If-then-else, 106
 - Immediate instructions, 86
 - Imprecise interrupts, 390, CD4.15:3
 - Index-out-of-bounds check, 110
 - Induction variable elimination, CD2.15:6
 - Inheritance, CD2.15:14
 - In-order commit, 400
 - Input devices, 15
 - Inputs, 318
 - Instances, CD2.15:14
 - Instruction count, 35, 36
 - Instruction decode/register file read stage
 - control line, 361
 - load instruction, 348
 - store instruction, 352
 - Instruction execution illustrations, CD4.12:16–30
 - clock cycles 1 and 2, CD4.12:20
 - clock cycles 3 and 4, CD4.12:21
 - clock cycles 5 and 6, CD4.12:22
 - clock cycles 7 and 8, CD4.12:23
 - clock cycle 9, CD4.12:24
 - examples, CD4.12:19–24
 - forwarding, CD4.12:25, CD4.12:26–27
 - no hazard, CD4.12:16–19
 - pipelines with stalls and forwarding, CD4.12:25, CD4.12:28–30
 - Instruction fetch stage
 - control line, 361
 - load instruction, 348
 - store instruction, 352
 - Instruction formats
 - ARM, 164
 - defined, 95
 - desktop/server RISC architectures, E-7
 - embedded RISC architectures, E-8
 - I-type, 97
 - J-type, 129

- Instruction formats (*continued*)
 - jump instruction, 328
 - MIPS, 164
 - R-type, 97, 319
 - x86, 173
- Instruction latency, 408
- Instruction-level parallelism (ILP)
 - compiler exploitation, CD4.15:4–5
 - defined, 41, 391
 - exploitation, increasing, 402
 - See also* Parallelism
- Instruction mix, 37, CD1.10:9
- Instructions, 74–221
 - add immediate, 86
 - addition, 226, B-51
 - Alpha, E-27–29
 - arithmetic-logical, 308, B-51–57
 - ARM, 161–65, E-36–37
 - assembly, 80
 - basic block, 108–9
 - branch, B-59–63
 - cache-aware, 547
 - comparison, B-57–59
 - conditional branch, 105
 - conditional move, 383
 - constant-manipulating, B-57
 - conversion, B-75–76
 - core, 282
 - data movement, B-70–73
 - data transfer, 82
 - decision-making, 105–12
 - defined, 11, 76
 - desktop RISC conventions, E-12
 - division, B-52–53
 - as electronic signals, 94
 - embedded RISC conventions, E-15
 - encoding, 98
 - exception and interrupt, B-80
 - exclusive OR, B-57
 - fetching, 309
 - fields, 95
 - floating-point, 259–61, B-73–80
 - floating-point (x86), 273
 - flushing, 377, 378, 390
 - immediate, 86
 - introduction to, 76–77
 - I/O, 589
 - jump, 111, 113, B-63–64
 - left-to-right flow, 346
 - load, 83, B-66–68
 - load linked, 138
 - logical operations, 102–5
 - M32R, E-40
 - memory access, A-33–34
 - memory-reference, 301
 - MIPS-16, E-40–42
 - MIPS-64, E-25–27
 - multiplication, 235, B-53–54
 - negation, B-54
 - nop, 373
 - PA-RISC, E-34–36
 - performance, 33–34
 - pipeline sequence, 372
 - PowerPC, E-12–13, E-32–34
 - PTX, A-31, A-32
 - remainder, B-55
 - representation in computer, 94–101
 - restartable, 513
 - resuming, 516
 - R-type, 308–9
 - shift, B-55–56
 - SPARC, E-29–32
 - store, 85, B-68–70
 - store conditional, 138–39
 - subtraction, 226, B-56–57
 - SuperH, E-39–40
 - thread, A-30–31
 - Thumb, E-38
 - trap, B-64–66
 - vector, 652
 - as words, 76
 - x86, 165–74
 - See also* Arithmetic instructions; MIPS; Operands
- Instruction set architecture
 - ARM, 161–65
 - branch address calculation, 310
 - defined, 21, 54
 - history, 179
 - maintaining, 54
 - protection and, 528–29
 - thread, A-31–34
 - virtual machine support, 527–28
- Instruction sets
 - ARM, 383
 - design for pipelining, 335
 - MIPS, 77, 178, 279
 - MIPS-32, 281
 - NVIDIA GeForce 8800, A-49
 - Pseudo MIPS, 281
 - x86 growth, 176
- Instructions per clock cycle (IPC), 391
- Integrated circuits (ICs)
 - cost, 46
 - defined, 26
 - manufacturing process, 45
 - very large-scale (VLSIs), 26
 - See also specific chips*
- Integrated Data Store (IDS), CD6.14:4
- Intel IA-64 architecture, CD4.15:4
- Intel Nehalem
 - address translation for, 540
 - caches, 541
 - die processor photo, 539
 - memory hierarchies, 540–43
 - miss penalty reduction techniques, 541–43
 - TLB hardware for, 540
- Intel Paragon, CD7.14:7
- Intel Threading Building Blocks, A-60
- Intel Xeon e5345
 - base versus fully optimized performance, 683
 - characteristics, 677
 - defined, 677
 - illustrated, 677
 - LBMHD performance, 682
 - roofline model, 678
 - SpmV performance, 681
- Interference graphs, CD2.15:11
- Interleaving, 472, 474
- Intermediate addressing, 132, 133
- Internetworking, CD6.11:1–3
- Interprocedural analysis, CD2.15:13
- Interrupt-driven I/O, 589
- Interrupt enable, 512
- Interrupt handlers, B-33
- Interrupt priority levels (IPLs), 590–92
 - defined, 591
 - higher, 592
- Interrupts
 - defined, 227, 385
 - event types and, 385
 - exceptions versus, 384–85
 - imprecise, 390, CD4.15:3
 - instructions, B-80
 - precise, 390
 - vectored, 386
- Intrinsity FastMATH processor, 468–70
 - caches, 469
 - data miss rates, 470, 484

defined, 468
 read processing, 506
 TLB, 504
 write-through processing, 506

Inverted page tables, 500

I/O, B-38–40, CD6.14:1–8
 bandwidth, 618
 chip sets, 586
 coherence problem for, 595
 controllers, 593, 615
 future directions, 618
 instructions, 589
 interrupt-driven, 589
 memory-mapped, 588, B-38
 parallelism and, 599–606
 performance, 572
 performance measures, 596–98
 processor communication,
 589–90
 rate, 596, 610, 611
 requests, 572, 618
 standards, 584
 system performance impact,
 599–600
 systems, 570
 transactions, 583

I/O benchmarks, 596–97
 file system, 597–98
 transaction processing,
 596–97
 Web, 597–98
See also Benchmarks

I/O devices
 characteristics, 571
 commands to, 588–89
 diversity, 571
 expandability, 572
 illustrated, 570
 interfacing, 586–95
 maximum number, 617
 multiple paths to, 618
 priorities, 590–92
 reads/writes to, 572
 transfers, 585, 592–93

I/O interconnects
 function, 583
 of x86 processors, 584–86

I/O systems
 design, 598–99
 design example, 609–11
 history, 618

operating system responsibilities
 and, 587–88
 organization, 585
 peak transfer rate, 617
 performance, 618
 power evaluation, 611–12
 weakest link, 598

Issue packets, 393

J

Java
 bytecode, 147
 bytecode architecture, CD2.15:16
 characters in, 126–27
 compiling in, CD2.15:18–19
 goals, 146
 interpreting, 148, 161, CD2.15:14–15
 keywords, CD2.15:20
 method invocation in,
 CD2.15:19–20
 pointers, CD2.15:25
 primitive types, CD2.15:25
 programs, starting, 146–48
 reference types, CD2.15:25
 sort algorithms, 157
 strings in, 126–27
 translation hierarchy, 148
 while loop compilation in,
 CD2.15:17–18

Java Virtual Machine (JVM), 147,
 CD2.15:15

Job-level parallelism, 632

J-type instruction format, 129

Jump instructions, 312
 branch instruction versus, 328
 control and datapath for, 329
 implementing, 328
 instruction format, 328
 list of, B-63–64
 MIPS-64, E-26

Just In Time (JIT) compilers,
 148, 687

K

Karnaugh maps, C-18

Kernel mode, 509

Kernels
 CUDA, A-19, A-24
 defined, A-19

L

Labels
 global, B-10, B-11
 local, B-11

LAPACK, 271

Laptop computers, 18

Large-scale multiprocessors,
 CD7.14:6–7, CD7.14:8–9

Latches
 defined, C-51
 D latch, C-51, C-52

Latency
 constraints, 598
 instruction, 408
 memory, A-74–75
 pipeline, 344
 rotational, 576
 use, 395, 396

Lattice Boltzmann Magneto-
 Hydrodynamics (LBMHD),
 680–82
 defined, 680
 optimizations, 681–82
 performance, 682

Leaf procedures
 defined, 116
 example, 126
See also Procedures

Least recently used (LRU)
 as block replacement strategy, 521
 defined, 485
 pages, 499

Least significant bits, C-32
 defined, 88
 SPARC, E-31

Left-to-right instruction flow, 346

Level-sensitive clocking, C-74,
 C-75–76
 defined, C-74
 two-phase, C-75

Lines. *See* Blocks

Linkers, 142–45, B-18–19
 defined, 142, B-4
 executable files, 142, B-19
 function illustration, B-19
 steps, 142
 using, 143–45

Linking object files, 143–45

Linpack, 664, CD3.10:3

Liquid crystal displays (LCDs), 16

- LISP, SPARC support, E-30
 - Little-endian byte order, B-43
 - Live range, CD2.15:10
 - Livermore Loops, CD1.10:10
 - Load balancing, 637–38
 - Loaders, 145
 - Loading, B-19–20
 - Load instructions
 - access, A-41
 - base register, 319
 - block, 165
 - compiling with, 85
 - datapath in operation for, 325
 - defined, 83
 - details, B-66–68
 - EX stage, 350
 - floating-point, B-76–77
 - halfword unsigned, 126
 - ID stage, 349
 - IF stage, 349
 - linked, 138, 139
 - list of, B-66–68
 - load byte unsigned, 124
 - load half, 126
 - load upper immediate, 128, 129
 - MEM stage, 351
 - pipelined datapath in, 355
 - signed, 124
 - unit for implementing, 311
 - unsigned, 124
 - WB stage, 351
 - See also* Store instructions
 - Load-store architectures, CD2.20:2
 - Load-use data hazard, 338, 377
 - Load-use stalls, 377
 - Load word, 83, 85
 - Local area networks (LANs), CD6.11:5–8, CD6.14:8
 - defined, 25
 - Ethernet, CD6.11:5–6
 - hubs, CD6.11:6, CD6.11:7
 - routers, CD6.11:6
 - switches, CD6.11:6–7
 - wireless, CD6.11:8–11
 - See also* Networks
 - Locality
 - principle, 452, 453
 - spatial, 452–53, 456
 - temporal, 452, 453, 456
 - Local labels, B-11
 - Local memory, A-21, A-40
 - Local miss rates, 489
 - Local optimization, CD2.15:4–6
 - defined, CD2.15:4
 - implementing, CD2.15:7
 - See also* Optimization
 - Locks, 639
 - Lock synchronization, 137
 - Logic
 - address select, D-24, D-25
 - ALU control, D-6
 - combinational, 306, C-5, C-9–20
 - components, 305
 - control unit equations, D-11
 - design, 303–7, C-1–79
 - equations, C-7
 - minimization, C-18
 - programmable array (PAL), C-78
 - sequential, C-5, C-56–58
 - two-level, C-11–14
 - Logical operations, 102–5
 - AND, 103–4, B-52
 - ARM, 165
 - defined, 102–5
 - desktop RISC, E-11
 - embedded RISC, E-14
 - MIPS, B-51–57
 - NOR, 104–5, B-54
 - NOT, 104, B-55
 - OR, 104, B-55
 - shifts, 102
 - Long-haul networks, CD6.11:5
 - Long instruction word (LIW), CD4.15:4
 - Lookup tables (LUTs), C-79
 - Loops, 107–8
 - conditional branches in, 130
 - defined, 107
 - for, 157, CD2.15:25
 - prediction and, 380
 - test, 158, 159
 - while, compiling, 107–8
 - Loop unrolling
 - defined, 397, CD2.15:3
 - for multiple-issue pipelines, 397
 - register renaming and, 397
- M**
- M32R, E-15, E-40
 - Machine code, 95
 - Machine instructions, 95
 - Machine language
 - branch offset in, 131–32
 - decoding, 134
 - defined, 11, 95, B-3
 - floating-point, 260
 - illustrated, 12
 - MIPS, 100
 - SRAM, 20
 - translating MIPS assembly language into, 98–99
 - Macros
 - defined, B-4
 - example, B-15–17
 - use of, B-15
 - Magnetic disks. *See* Hard disks
 - Magnetic tapes, 615–16
 - defined, 23
 - use history, 615–16
 - Main memory, 493
 - defined, 21
 - page tables, 501
 - physical addresses, 492, 493
 - See also* Memory
 - Mapping applications, A-55–72
 - Mark computers, CD1.10:3
 - Mealy machine, 532, C-68, C-71, C-72
 - Mean time between failures (MTBF), 573
 - Mean time to failure (MTTF), 573, 574
 - fallacies, 613
 - ratings, 600
 - Mean time to repair (MTTR), 573, 574
 - Memory
 - addresses, 91
 - affinity, 680, 681
 - atomic, A-21
 - bandwidth, 471, 472
 - cache, 20, 457–92
 - CAM, 485
 - constant, A-40
 - control, D-26
 - defined, 17
 - DRAM, 18–19, 453, 471, 473, C-63–65
 - efficiency, 642
 - flash, 22, 23, 580–82, CD6.14:4
 - global, A-21, A-39
 - GPU, 656
 - instructions, datapath for, 314
 - layout, B-21
 - local, A-21, A-40
 - main, 21

- nonvolatile, 21
 - operands, 82–83
 - parallel system, A-36–41
 - read-only (ROM), C-14–16
 - SDRAM, 473
 - secondary, 22
 - shared, A-21, A-39–40
 - spaces, A-39
 - SRAM, C-58–62
 - stalls, 478
 - technologies for building, 25–26
 - texture, A-40
 - usage, B-20–22
 - virtual, 492–517
 - volatile, 21
- Memory access instructions, A-33–34
- Memory access stage
- control line, 362
 - load instruction, 350
 - store instruction, 352
- Memory consistency model, 538
- Memory elements, C-50–58
- clocked, C-51
 - D flip-flop, C-51, C-53
 - D latch, C-52
 - DRAMs, C-63–67
 - flip-flop, C-51
 - hold time, C-54
 - latch, C-51
 - setup time, C-53, C-54
 - SRAMs, C-58–62
 - unclocked, C-51
- Memory hierarchies
- block (or line), 454
 - cache performance, 475–92
 - caches, 457–75
 - common framework, 518–25
 - defined, 453
 - design challenges, 525
 - development, CD5.13:5–7
 - exploiting, 450–548
 - inclusion, 542
 - level pairs, 455
 - multiple levels, 454
 - overall operation of, 507
 - parallelism and, 534–38
 - pitfalls, 543–47
 - program execution time and, 491
 - quantitative design parameters, 518
 - reliance on, 455
 - structure, 454
- structure diagram, 456
 - variance, 491
 - virtual memory, 492–517
- Memory-mapped I/O
- defined, 588
 - use of, B-38
- Memory-stall clock cycles, 475, 476
- Message passing
- defined, 641
 - multiprocessors, 641–45
- Metastability, C-76
- Methods
- defined, CD2.15:14
 - invoking in Java, CD2.15:19–20
 - static, B-20
- Microarchitectures
- AMD Opteron X4 (Barcelona), 405
 - defined, 404
- Microcode
- assembler, D-30
 - control unit as, D-28
 - defined, D-27
 - dispatch ROMs, D-30–31
 - field translation, D-29
 - horizontal, D-32
 - vertical, D-32
- Microinstructions, D-31
- Microprocessors
- design shift, 633
 - multicore, 8, 41, 632
- Microprograms
- as abstract control representation, D-30
 - translating to hardware, D-28–32
- Migration, 536
- Million instructions per second (MIPS), 53
- Minterms
- defined, C-12, D-20
 - in PLA implementation, D-20
- MIP-map, A-44
- MIPS, 78, 98–99, B-45–80
- addressing for 32-bit immediates, 128–36
 - addressing modes, B-45–47
 - arithmetic core, 280
 - arithmetic instructions, 77, B-51–57
 - ARM similarities, 162
 - assembler directive support, B-47–49
 - assembler syntax, B-47–49
 - assembly instruction, mapping, 95
 - branch instructions, B-59–63
 - comparison instructions, B-57–59
 - compiling C assignment statements into, 79
 - compiling complex C assignment into, 79–80
 - constant-manipulating instructions, B-57
 - control registers, 511
 - control unit, D-10
 - CPU, B-46
 - divide in, 241–42
 - exceptions in, 385–86
 - fields, 96–97
 - floating-point instructions, 259–61
 - FPU, B-46
 - instruction classes, 179
 - instruction encoding, 98, 135, B-49
 - instruction formats, 136, 164, B-49–51
 - instruction set, 77, 178, 279
 - jump instructions, B-63–66
 - logical instructions, B-51–57
 - machine language, 100
 - memory addresses, 84
 - memory allocation for program and data, 120
 - multiply in, 235
 - opcode map, B-50
 - operands, 78
 - Pseudo, 280, 281
 - register conventions, 121
 - static multiple issue with, 394–97
- MIPS-16, E-15–16
- 16-bit instruction set, E-41–42
 - immediate fields, E-41
 - instructions, E-40–42
 - MIPS core instruction changes, E-42
 - PC-relative addressing, E-41
- MIPS-32 instruction set, 281
- MIPS-64 instructions, E-25–27
- conditional procedure call instructions, E-27
 - constant shift amount, E-25
 - jump/call not PC-relative, E-26
 - move to/from control registers, E-26
 - nonaligned data transfers, E-25
 - NOR, E-25
 - parallel single precision floating-point operations, E-27
 - reciprocal and reciprocal square root, E-27
 - SYSCALL, E-25
 - TLB instructions, E-26–27

- MIPS core
 - architecture, 243
 - arithmetic/logical instructions
 - not in, E-21, E-23
 - common extensions to, E-20–25
 - control instructions not in, E-21
 - data transfer instructions not in, E-20, E-22
 - floating-point instructions
 - not in, E-22
 - instruction set, 282, 300–303, E-9–10
 - Mirroring, 602
 - Miss penalty
 - defined, 455
 - determination, 464
 - multilevel caches, reducing, 487–91
 - reduction techniques, 541–43
 - Miss rates
 - block size versus, 465
 - data cache, 519
 - defined, 454
 - global, 489
 - improvement, 464
 - Intrinsity FastMATH processor, 470
 - local, 489
 - miss sources, 524
 - split cache, 470
 - Miss under miss, 541
 - Modules, B-4
 - Moore machines, 532, C-68, C-71, C-72
 - Moore's law, 654, A-72–73
 - Most significant bit
 - 1-bit ALU for, C-33
 - defined, 88
 - Motherboards, 17
 - Mouse anatomy, 16
 - Move instructions, B-70–73
 - coprocessor, B-71–72
 - details, B-70–73
 - floating-point, B-77–78
 - MS-DOS, CD5.13:10–11
 - Multicore multiprocessors, 41
 - benchmarking with roofline model, 675–84
 - characteristics, 677
 - defined, 8, 632
 - system organization, 676
 - two sockets, 676
 - MULTICS (Multiplexed Information and Computing Service), CD5.13:8–9
 - Multilevel caches
 - complications, 489
 - defined, 475, 489
 - miss penalty, reducing, 487–91
 - performance of, 487–88
 - summary, 491–92
 - See also* Caches
 - Multimedia arithmetic, 227–28
 - Multimedia extensions
 - desktop/server RISCs, E-16–18
 - vector versus, 653
 - Multiple-clock-cycle pipeline
 - diagrams, 356
 - defined, 356
 - five instructions, 357
 - illustrated, 357
 - Multiple dimension arrays, 266
 - Multiple instruction multiple data (MIMD), 659
 - defined, 648
 - first multiprocessor, CD7.14:3
 - Multiple instruction single data (MISD), 649
 - Multiple issue, 391–400
 - code scheduling, 396
 - defined, 391
 - dynamic, 392, 397–400
 - issue packets, 393
 - loop unrolling and, 397
 - processors, 391, 392
 - static, 392, 393–97
 - throughput and, 401
 - Multiplexors, C-10
 - controls, 531
 - in datapath, 320
 - defined, 302
 - forwarding, control values, 370
 - selector control, 314
 - two-input, C-10
 - Multiplicand, 230
 - Multiplication, 230–36
 - fast, hardware, 236
 - faster, 235
 - first algorithm, 232
 - floating-point, 255–58, B-78
 - hardware, 231–33
 - instructions, 235, B-53–54
 - in MIPS, 235
 - multiplicand, 230
 - multiplier, 230
 - operands, 230
 - product, 230
 - sequential version, 231–33
 - signed, 234
 - See also* Arithmetic
 - Multiplier, 230
 - Multiply-add (MAD), A-42
 - Multiply algorithm, 234
 - Multiprocessors
 - benchmarks, 664–66
 - bus-based coherent, CD7.14:6
 - defined, 632
 - historical perspective, 688
 - large-scale, CD7.14:6–7, CD7.14:8–9
 - message-passing, 641–45
 - multithreaded architecture, A-26–27, A-35–36
 - organization, 631, 641
 - for performance, 686–87
 - shared-memory, 633, 638–40
 - software, 632
 - TFLOPS, CD7.14:5
 - UMA, 639
 - Multistage networks, 662
 - Multithreaded multiprocessor
 - architecture, A-25–36
 - conclusion, A-36
 - ISA, A-31–34
 - massive multithreading, A-25–26
 - multiprocessor, A-26–27
 - multiprocessor comparison, A-35–36
 - SIMT, A-27–30
 - special function units (SFUs), A-35
 - streaming processor (SP), A-34
 - thread instructions, A-30–31
 - threads/thread blocks management, A-30
 - Multithreading, A-25–26
 - coarse-grained, 645–46
 - defined, 634
 - fine-grained, 645, 647
 - hardware, 645–48
 - simultaneous (SMT), 646–48
 - Must-information, CD2.15:14
 - Mutual exclusion, 137
- ## N
- Name dependence, 397
 - NAND flash memory, CD6.14:4
 - NAND gates, C-8

- NAS (NASA Advanced Supercomputing), 666
 - N-body
 - all-pairs algorithm, A-65
 - GPU simulation, A-71
 - mathematics, A-65–67
 - multiple threads per body, A-68–69
 - optimization, A-67
 - performance comparison, A-69–70
 - results, A-70–72
 - shared memory use, A-67–68
 - Negation instructions, B-54, B-78–79
 - Negation shortcut, 91–92
 - Nested procedures, 116–18
 - compiling recursive procedure showing, 117–18
 - defined, 116
 - Network of Workstations, CD7.14:7–8
 - Networks, 24–25, 612–13, CD6.11:1–11
 - advantages, 24
 - bandwidth, 661
 - characteristics, CD6.11:1
 - crossbar, 662
 - fully connected, 661, 662
 - local area (LANs), 25, CD6.11:5–8, CD6.14:8
 - long-haul, CD6.11:5
 - multistage, 662
 - OSI model layers, CD6.11:2
 - peer-to-peer, CD6.11:2
 - performance, CD6.11:7–8
 - protocol families/suites, CD6.11:1
 - switched, CD6.11:5
 - wide area (WANs), 25, CD6.14:7–8
 - Network topologies, 660–63
 - implementing, 662–63
 - multistage, 663
 - Newton's iteration, 266
 - Next state
 - nonsequential, D-24
 - sequential, D-23
 - Next-state function, 531, C-67
 - defined, 531
 - implementing, with sequencer, D-22–28
 - Next-state outputs, D-10, D-12–13
 - example, D-12–13
 - implementation, D-12
 - logic equations, D-12–13
 - truth tables, D-15
 - Nonblocking assignment, C-24
 - Nonblocking caches, 403, 541
 - Nonuniform memory access (NUMA), 639
 - Nonvolatile memory, 21
 - Nonvolatile storage, 575
 - Nops, 373
 - NOR flash memory, 581, CD6.14:4
 - NOR gates, C-8
 - cross-coupled, C-50
 - D latch implemented with, C-52
 - NOR operation, 104–5, B-54, E-25
 - North bridge, 584
 - NOT operation, 104, B-55, C-6
 - No write allocation, 467
 - Numbers
 - binary, 87
 - computer versus real-world, 269
 - decimal, 87, 90
 - denormalized, 270
 - hexadecimal, 95–96
 - signed, 87–94
 - unsigned, 87–94
 - NVIDIA GeForce 3, CDA.11:1
 - NVIDIA GeForce 8800, A-46–55, CDA.11:3
 - all-pairs N-body algorithm, A-71
 - dense linear algebra computations, A-51–53
 - FFT performance, A-53
 - instruction set, A-49
 - performance, A-51
 - rasterization, A-50
 - ROP, A-50–51
 - scalability, A-51
 - sorting performance, A-54–55
 - special function approximation statistics, A-43
 - special function unit (SFU), A-50
 - streaming multiprocessor (SM), A-48–49
 - streaming processor, A-49–50
 - streaming processor array (SPA), A-46
 - texture/processor cluster (TPC), A-47–48
 - NVIDIA GPU architecture, 656–59
- O**
- Object files, 141, B-4
 - debugging information, 142
 - defined, B-10
 - format, B-13–14
 - header, 141, B-13
 - linking, 143–45
 - relocation information, 141
 - static data segment, 141
 - symbol table, 141, 142
 - text segment, 141
 - Object-oriented languages
 - brief history, CD2.20:7
 - defined, 161, CD2.15:14
 - See also* Java
 - One's complement, 94, C-29
 - Opcodes
 - control line setting and, 323
 - defined, 97, 319
 - OpenGL, A-13
 - OpenMP (Open MultiProcessing), 666
 - Open Systems Interconnect (OSI) model, CD6.11:2
 - Operands, 80–87
 - 32-bit immediate, 128–29
 - adding, 225
 - arithmetic instructions, 80
 - compiling assignment when in memory, 83
 - constant, 86–87
 - division, 237
 - floating-point, 260
 - memory, 82–83
 - MIPS, 78
 - multiplication, 230
 - shifting, 164
 - See also* Instructions
 - Operating systems
 - brief history, CD5.13:8–11
 - defined, 10
 - disk access scheduling pitfall, 616–17
 - encapsulation, 21
 - Operations
 - atomic, implementing, 138
 - hardware, 77–80
 - logical, 102–5
 - x86 integer, 168–71
 - Optical disks
 - defined, 23
 - technology, 24
 - Optimization
 - class explanation, CD2.15:13
 - compiler, 160
 - control implementation, D-27–28
 - global, CD2.15:4–6

- Optimization (*continued*)
 - high-level, CD2.15:3
 - local, CD2.15:4–6, CD2.15:7
 - manual, 160
 - OR operation, 104, B-55, C-6
 - Out-of-order execution
 - defined, 400
 - performance complexity, 489
 - processors, 403
 - Output devices, 15
 - Overflow
 - defined, 89, 245
 - detection, 226
 - exceptions, 387
 - floating point, 245
 - occurrence, 90
 - saturation and, 227–28
 - subtraction, 226
- P**
- Packed floating-point format, 274
 - Page faults, 498
 - for data access, 513
 - defined, 493, 494
 - handling, 495, 510–16
 - virtual address causing, 514
 - See also* Virtual memory
 - Pages
 - defined, 493
 - dirty, 501
 - finding, 496
 - LRU, 499
 - offset, 494
 - physical number, 494
 - placing, 496
 - size, 495
 - virtual number, 494
 - See also* Virtual memory
 - Page tables, 520
 - defined, 496
 - illustrated, 499
 - indexing, 497
 - inverted, 500
 - levels, 500–501
 - main memory, 501
 - register, 497
 - storage reduction techniques, 500–501
 - updating, 496
 - VMM, 529
 - Parallelism, 41, 391–403
 - data-level, 649
 - debates, CD7.14:4–6
 - GPUs and, 655, A-76
 - instruction-level, 41, 391, 402
 - I/O and, 599–606
 - job-level, 632
 - memory hierarchies and, 534–38
 - multicore and, 648
 - multiple issue, 391–400
 - multithreading and, 648
 - performance benefits, 43
 - process-level, 632
 - subword, E-17
 - task, A-24
 - thread, A-22
 - Parallel memory system, A-36–41
 - caches, A-38
 - constant memory, A-40
 - DRAM considerations, A-37–38
 - global memory, A-39
 - load/store access, A-41
 - local memory, A-40
 - memory spaces, A-39
 - MMU, A-38–39
 - ROP, A-41
 - shared memory, A-39–40
 - surfaces, A-41
 - texture memory, A-40
 - See also* Graphics processing units (GPUs)
 - Parallel-processing programs, 634–38
 - creation difficulty, 634–38
 - defined, 632
 - for message passing, 642–43
 - for shared address space, 639–40
 - use of, 686
 - Parallel reduction, A-62
 - Parallel scan, A-60–63
 - CUDA template, A-61
 - defined, A-60
 - inclusive, A-60
 - tree-based, A-62
 - Parallel software, 633
 - Paravirtualization, 547
 - PA-RISC, E-14, E-17
 - branch vectored, E-35
 - conditional branches, E-34, E-35
 - debug instructions, E-36
 - decimal operations, E-35
 - extract and deposit, E-35
 - instructions, E-34–36
 - load and clear instructions, E-36
 - multiply/add and multiply/ subtract, E-36
 - nullification, E-34
 - nullifying branch option, E-25
 - store bytes short, E-36
 - synthesized multiply and divide, E-34–35
 - Parity, 602
 - bit-interleaved, 602
 - block-interleaved, 602–04
 - code, C-65
 - disk, 603
 - distributed block-interleaved, 603–4
 - PARSEC (Princeton Application Repository for Shared-Memory Computers), 666
 - Pass transistor, C-63
 - PCI-Express (PCIe), A-8
 - PC-relative addressing, 130, 133
 - Peak floating-point performance, 668
 - Peak transfer rate, 617
 - Peer-to-peer networks, CD6.11:2
 - Pentium bug morality play, 276–79
 - Performance, 26–38
 - assessing, 26–27
 - classic CPU equation, 35–37
 - components, 37
 - CPU, 30–32
 - defining, 27–30
 - equation, using, 34
 - improving, 32–33
 - instruction, 33–34
 - measuring, 30–32, CD1.10:9
 - networks, CD6.11:7–8
 - program, 38
 - ratio, 30
 - relative, 29
 - response time, 28, 29
 - sorting, A-54–55
 - throughput, 28
 - time measurement, 30
 - Petabytes, 5
 - Physical addresses, 493
 - defined, 492
 - mapping to, 494
 - space, 638, 640
 - Physically addressed caches, 508
 - Physical memory. *See* Main memory
 - Pipelined branches, 378

- Pipelined control, 359–63
 - control lines, 360, 361
 - overview illustration, 375
 - specifying, 361
 - See also* Control
- Pipelined datapaths, 344–58
 - with connected control signals, 362
 - with control signals, 359
 - corrected, 355
 - illustrated, 347
 - in load instruction stages, 355
- Pipelined dependencies, 364
- Pipeline registers
 - before forwarding, 368
 - dependences, 366, 367
 - forwarding unit selection, 371
- Pipelines
 - AMD Opteron X4 (Barcelona), 404–6
 - branch instruction impact, 376
 - effectiveness, improving, CD4.15:3–4
 - execute and address calculation stage, 350, 352
 - five-stage, 333, 348–50, 358
 - fixed-function graphics, CDA.11:1
 - graphic representation, 337, 356–58
 - instruction decode and register file read stage, 348, 352
 - instruction fetch stage, 348, 352
 - instructions sequence, 372
 - latency, 344
 - memory access stage, 350, 352
 - multiple-clock-cycle diagrams, 356
 - performance bottlenecks, 402
 - single-clock-cycle diagrams, 356
 - stages, 333
 - static two-issue, 394
 - write-back stage, 350, 352
- Pipeline stalls, 338–39
 - avoiding with code reordering, 338–39
 - data hazards and, 371–74
 - defined, 338
 - insertion, 374
 - load-use, 377
 - as solution to control hazards, 340
- Pipelining, 330–44
 - advanced, 402–3
 - benefits, 331
 - control hazards, 339–43
 - data hazards, 336–39
 - defined, 330
 - exceptions and, 386–91
 - execution time and, 344
 - fallacies, 407
 - hazards, 335–43
 - instruction set design for, 335
 - laundry analogy, 331
 - overview, 330–44
 - paradox, 331
 - performance improvement, 335
 - pitfall, 407–8
 - simultaneous executing instructions, 344
 - speed-up formula, 333
 - structural hazards, 335–36, 352
 - summary, 343
 - throughput and, 344
- Pitfalls
 - address space extension, 545
 - associativity, 545
 - defined, 51
 - GPUs, A-74–75
 - ignoring memory system behavior, 544
 - magnetic tape backups, 615–16
 - memory hierarchies, 543–47
 - moving functions to I/O processor, 615
 - network feature provision, 614–15
 - operating system disk accesses, 616–17
 - out-of-order processor evaluation, 545
 - peak transfer rate performance, 617
 - performance equation subset, 52–53
 - pipelining, 407–8
 - pointer to automatic variables, 175
 - sequential word addresses, 175
 - simulating cache, 543–44
 - software development with multiprocessors, 685
 - VMM implementation, 545–47
 - See also* Fallacies
- Pixel shader example, A-15–17
- Pizza boxes, 607
- Pointers
 - arrays versus, 157–61
 - frame, 119
 - global, 118
 - incrementing, 159
 - Java, CD2.15:25
 - stack, 114, 116
- Polling, 589
- Pop, 114
- Power
 - clock rate and, 39
 - critical nature of, 55
 - efficiency, 402–3
 - relative, 40
- PowerPC
 - algebraic right shift, E-33
 - branch registers, E-32–33
 - condition codes, E-12
 - instructions, E-12–13
 - instructions unique to, E-31–33
 - load multiple/store multiple, E-33
 - logical shifted immediate, E-33
 - rotate with mask, E-33
- P + Q redundancy, 604
- Precise interrupts, 390
- Prediction
 - 2-bit scheme, 381
 - accuracy, 380, 381
 - dynamic branch, 380–83
 - loops and, 380
 - steady-state, 380
- Prefetching, 547, 680
- Primary memory. *See* Main memory
- Primitive types, CD2.15:25
- Priority levels, 590–92
- Procedure calls
 - convention, B-22–33
 - examples, B-27–33
 - frame, B-23
 - preservation across, 118
- Procedures, 112–22
 - compiling, 114
 - compiling, showing nested procedure linking, 117–18
 - defined, 112
 - execution steps, 112
 - frames, 119
 - leaf, 116
 - nested, 116–18
 - recursive, 121, B-26–27
 - for setting arrays to zero, 158
 - sort, 150–55
 - strcpy, 124–25, 126
 - string copy, 124–26
 - swap, 149–50
- Process identifiers, 510
- Process-level parallelism, 632
- Processor-memory bus, 582

- Processors, 298–409
 - control, 19
 - as cores, 41
 - datapath, 19
 - defined, 14, 19
 - dynamic multiple-issue, 392
 - I/O communication with, 589–90
 - multiple-issue, 391, 392
 - out-of-order execution, 403, 489
 - performance growth, 42
 - ROP, A-12, A-41
 - speculation, 392–93
 - static multiple-issue, 392, 393–97
 - streaming, 657, A-34
 - superscalar, 397, 398, 399–400, 646, CD4.15:4
 - technologies for building, 25–26
 - two-issue, 395
 - vector, 650–53
 - VLIW, 394
 - Product, 230
 - Product of sums, C-11
 - Program counters (PCs), 307
 - changing with conditional branch, 383
 - defined, 113, 307
 - exception, 509, 511
 - incrementing, 307, 309
 - instruction updates, 348
 - Program libraries, B-4
 - Programmable array logic (PAL), C-78
 - Programmable logic arrays (PLAs)
 - component dots illustration, C-16
 - control function implementation, D-7, D-20–21
 - defined, C-12
 - example, C-13–14
 - illustrated, C-13
 - ROMs and, C-15–16
 - size, D-20
 - truth table implementation, C-13
 - Programmable logic devices (PLDs), C-78
 - Programmable real-time graphics, CDA.11:2–3
 - Programmable ROMs (PROMs), C-14
 - Programming languages
 - brief history of, CD2.20:6–7
 - object-oriented, 161
 - variables, 81
 - See also specific languages*
 - Program performance
 - elements affecting, 38
 - understanding, 9
 - Programs
 - assembly language, 139
 - Java, starting, 146–48
 - parallel-processing, 634–38
 - starting, 139–48
 - translating, 139–48
 - Propagate
 - defined, C-40
 - example, C-44
 - super, C-41
 - Protected keywords, CD2.15:20
 - Protection
 - defined, 492
 - group, 602
 - implementing, 508–10
 - mechanisms, CD5.13:7
 - VMs for, 526
 - Protocol families/suites
 - analogy, CD6.11:2–3
 - defined, CD6.11:1
 - goal, CD6.11:2
 - Protocol stacks, CD6.11:3
 - Pseudodirect addressing, 133
 - Pseudoinstructions
 - defined, 140
 - summary, 141
 - Pseudo MIPS
 - defined, 280
 - instruction set, 281
 - Pthreads (POSIX threads), 666
 - PTX instructions, A-31, A-32
 - Public keywords, CD2.15:20
 - Push
 - defined, 114
 - using, 116
- ## Q
- Quad words, 168
 - Quicksort, 489, 490
 - Quotient, 237
- ## R
- Race, C-73
 - Radix sort, 489, 490, A-63–65
 - CUDA code, A-64
 - implementation, A-63–65
 - RAID. *See* Redundant arrays of inexpensive disks
 - RAMAC (Random Access Method of Accounting and Control), CD6.14:1, CD6.14:2
 - Rank units, 606, 607
 - Rasterization, A-50
 - Raster operation (ROP) processors, A-12, A-41
 - fixed function, A-41
 - GeForce 8800, A-50–51
 - Raster refresh buffer, 17
 - Read-only memories (ROMs), C-14–16
 - control entries, D-16–17
 - control function encoding, D-18–19
 - defined, C-14
 - dispatch, D-25
 - implementation, D-15–19
 - logic function encoding, C-15
 - overhead, D-18
 - PLAs and, C-15–16
 - programmable (PROM), C-14
 - total size, D-16
 - Read-stall cycles, 476
 - Receive message routine, 641
 - Receiver Control register, B-39
 - Receiver Data register, B-38, B-39
 - Recursive procedures, 121, B-26–27
 - clone invocation, 116
 - defined, B-26
 - stack in, B-29–30
 - See also* Procedures
 - Reduced instruction set computer (RISC) architectures, E-2–45, CD2.20:4, CD4.15:3
 - group types, E-3–4
 - instruction set lineage, E-44
 - See also* Desktop and server RISCs; Embedded RISCs
 - Reduction, 640
 - Redundant arrays of inexpensive disks (RAID), 600–606
 - calculation of, 605
 - defined, 600
 - example illustration, 601
 - history, CD6.14:6–7
 - PCI controller, 611
 - popularity, 600
 - RAID 0, 601
 - RAID 1, 602, CD6.14:6
 - RAID 1 + 0, 606

- RAID 2, 602, CD6.14:6
 - RAID 3, 602, CD6.14:6, CD6.14:7
 - RAID 4, 602–3, CD6.14:6
 - RAID 5, 603–4, CD6.14:6, CD6.14:7
 - RAID 6, 604
 - spread of, CD6.14:7
 - summary, 604–5
 - use statistics, CD6.14:7
 - Reference bit, 499
 - References
 - absolute, 142
 - forward, B-11
 - types, CD2.15:25
 - unresolved, B-4, B-18
 - Register addressing, 132, 133
 - Register allocation, CD2.15:10–12
 - Register files, C-50, C-54–56
 - in behavioral Verilog, C-57
 - defined, 308, C-50, C-54
 - single, 314
 - two read ports implementation, C-55
 - with two read ports/one write port, C-55
 - write port implementation, C-56
 - Register-memory architecture, CD2.20:2
 - Registers
 - architectural, 404
 - base, 83
 - callee-saved, B-23
 - caller-saved, B-23
 - Cause, 386, 590, 591, B-35
 - clock cycle time and, 81
 - compiling C assignment with, 81–82
 - Count, B-34
 - defined, 80
 - destination, 98, 319
 - floating-point, 265
 - left half, 348
 - mapping, 94
 - MIPS conventions, 121
 - number specification, 309
 - page table, 497
 - pipeline, 366, 367, 368, 371
 - primitives, 80–81
 - Receiver Control, B-39
 - Receiver Data, B-38, B-39
 - renaming, 397
 - right half, 348
 - spilling, 86
 - Status, 386, 590, 591, B-35
 - temporary, 81, 115
 - Transmitter Control, B-39–40
 - Transmitter Data, B-40
 - usage convention, B-24
 - use convention, B-22
 - variables, 81
 - x86, 168
 - Relational databases, CD6.14:5
 - Relative performance, 29
 - Relative power, 40
 - Reliability, 573
 - Relocation information, B-13, B-14
 - Remainder
 - defined, 237
 - instructions, B-55
 - Reorder buffers, 399, 402, 403
 - Replication, 536
 - Requested word first, 465
 - Reservation stations
 - buffering operands in, 400
 - defined, 399
 - Response time, 28, 29
 - Restartable instructions, 513
 - Restorations, 573
 - Return address, 113
 - Return from exception (ERET), 509
 - R-format, 319
 - ALU operations, 310
 - defined, 97
 - Ripple carry
 - adder, C-29
 - carry lookahead speed versus, C-46
 - RISC. *See* Desktop and server RISCs; Embedded RISCs; Reduced instruction set computer (RISC) architectures
 - Roofline model, 667–75
 - benchmarking multicores with, 675–84
 - with ceilings, 672, 674
 - computational roofline, 673
 - IBM Cell QS20, 678
 - illustrated, 669
 - Intel Xeon e5345, 678
 - I/O intensive kernel, 675
 - Opteron generations, 670
 - with overlapping areas shaded, 674
 - peak floating-point performance, 668
 - peak memory performance, 669
 - Sun UltraSPARC T2, 678
 - with two kernels, 674
 - Rotational latency, 576
 - Rounding
 - accurate, 266
 - bits, 268
 - defined, 266
 - with guard digits, 267
 - IEEE 754 modes, 268
 - Routers, CD6.11:6
 - Row-major order, 265
 - R-type instructions, 308–9
 - datapath for, 323
 - datapath in operation for, 324
- ## S
- Saturation, 227–28
 - Scalable GPUs, CDA.11:4–5
 - SCALAPAK, 271
 - Scaling
 - strong, 637, 638
 - weak, 637
 - Scientific notation
 - adding numbers in, 250
 - defined, 244
 - for reals, 244
 - Secondary memory, 22
 - Sectors, 575
 - Seek time, 575
 - Segmentation, 495
 - Selector values, C-10
 - Semiconductors, 45
 - Send message routine, 641
 - Sensitivity list, C-24
 - Sequencers
 - explicit, D-32
 - implementing next-state function with, D-22–28
 - Sequential logic, C-5
 - Servers
 - cost and capability, 5
 - defined, 5
 - See also* Desktop and server RISCs
 - Set-associative caches, 479–80
 - address portions, 484
 - block replacement strategies, 521
 - choice of, 520
 - defined, 479
 - four-way, 481, 486
 - memory-block location, 480
 - misses, 482–83

- Set-associative caches (*continued*)
 - n*-way, 479
 - two-way, 481
 - See also* Caches
- Set instructions, 109
- Setup time, C-53, C-54
- Shaders, CDA.11:3
 - defined, A-14
 - floating-point arithmetic, A-14
 - graphics, A-14–15
 - pixel example, A-15–17
- Shading languages, A-14
- Shared memory
 - caching in, A-58–60
 - CUDA, A-58
 - defined, A-21
 - as low-latency memory, A-21
 - N-body and, A-67–68
 - per-CTA, A-39
 - SRAM banks, A-40
 - See also* Memory
- Shared-memory multiprocessors (SMP), 638–40
 - defined, 633, 638
 - single physical address
 - space, 638
 - synchronization, 639
- Shift amount, 97
- Shift instructions, 102, B-55–56
- Signals
 - asserted, 305, C-4
 - control, 306, 320, 321, 322
 - deasserted, 305, C-4
- Sign and magnitude, 245
- Sign bit, 90
- Signed division, 239–41
- Signed multiplication, 234
- Signed numbers, 87–94
 - sign and magnitude, 89
 - treating as unsigned, 110
- Sign extension, 310
 - defined, 124
 - shortcut, 92–93
- Significands, 246
 - addition, 250
 - multiplication, 255
- Silicon
 - crystal ingot, 45
 - defined, 45
 - as key hardware technology, 54
 - wafers, 45
- SIMD (Single Instruction Multiple Data), 649, 659
 - computers, CD7.14:1–3
 - data vector, A-35
 - extensions, CD7.14:3
 - for loops and, CD7.14:2
 - massively parallel multiprocessors, CD7.14:1
 - small-scale, CD7.14:3
 - vector architecture, 650–53
 - in x86, 649–50
- SIMMs (single inline memory modules), CD5.13:4, CD5.13:5
- Simple programmable logic devices (SPLDs), C-78
- Simplicity, 176
- Simultaneous multithreading (SMT), 646–48
 - defined, 646
 - support, 647
 - thread-level parallelism, 647
 - unused issue slots, 648
- Single-clock-cycle pipeline
 - diagrams, 356
 - defined, 356
 - illustrated, 358
- Single-cycle datapaths
 - illustrated, 345
 - instruction execution, 346
 - See also* Datapaths
- Single-cycle implementation
 - control function for, 327
 - defined, 327
 - nonpipelined execution versus pipelined execution, 334
 - non-use of, 328–30
 - penalty, 330
 - pipelined performance versus, 332–33
- Single-instruction multiple-thread (SIMT), A-27–30
 - defined, A-27
 - multithreaded warp scheduling, A-28
 - overhead, A-35
 - processor architecture, A-28
 - warp execution and divergence, A-29–30
- Single instruction single data (SISD), 648
- Single precision
 - binary representation, 248
 - defined, 245
 - See also* Double precision
- Single-program multiple data (SPMD), 648, A-22
- Small Computer Systems Interface (SCSI) disks, 577, 613
- Smalltalk
 - Smalltalk-80, CD2.20:7
 - SPARC support, E-30
- Snooping protocol, 536–37, 538
- Snoopy cache coherence, CD5.9:16
- Software
 - GPU driver, 655
 - layers, 10
 - multiprocessor, 632
 - parallel, 633
 - as service, 606, 686
 - systems, 10
- Sort algorithms, 157
- Sorting performance, A-54–55
- Sort procedure, 150–55
 - code for body, 151–53
 - defined, 150
 - full procedure, 154–55
 - passing parameters in, 154
 - preserving registers in, 154
 - procedure call, 153
 - register allocation for, 151
 - See also* Procedures
- Source files, B-4
- Source language, B-6
- South bridge, 584
- Space allocation
 - on heap, 120–22
 - on stack, 119
- SPARC
 - annulling branch, E-23
 - CASA, E-31
 - conditional branches, E-10–12
 - fast traps, E-30
 - floating-point operations, E-31
 - instructions, E-29–32
 - least significant bits, E-31
 - multiple precision floating-point results, E-32
 - nonfaulting loads, E-32
 - overlapping integer operations, E-31
 - quadruple precision floating-point arithmetic, E-32
 - register windows, E-29–30
 - support for LISP and Smalltalk, E-30

- Sparse matrices, A-55–58
- Sparse Matrix-Vector multiply (SpMV), 679–80, 681, A-55, A-57, A-58
 - CUDA version, A-57
 - serial code, A-57
 - shared memory version, A-59
- Spatial locality, 452–53
 - defined, 452
 - large block exploitation of, 464
 - tendency, 456
- SPEC, CD1.10:10–11
 - CPU benchmark, 48–49
 - defined, CD1.10:10
 - power benchmark, 49–50
 - SPEC89, CD1.10:10
 - SPEC92, CD1.10:11
 - SPEC95, CD1.10:11
 - SPEC2000, CD1.10:11
 - SPEC2006, 282, CD1.10:11
 - SPECPower, 597
 - SPECrate, 664
 - SPECratio, 48
- Special function units (SFUs), A-35
 - defined, A-43
 - GeForce 8800, A-50
- Speculation, 392–93
 - defined, 392
 - hardware-based, 400
 - implementation, 392
 - performance and, 393
 - problems, 393
 - recovery mechanism, 393
- Speed-up challenge, 635–38
 - balancing load, 637–38
 - bigger problem, 636–37
- Spilling registers, 86, 115
- SPIM, B-40–45
 - byte order, B-43
 - defined, B-40
 - features, B-42–43
 - getting started with, B-42
 - MIPS assembler directives support, B-47–49
 - speed, B-41
 - system calls, B-43–45
 - versions, B-42
 - virtual machine simulation, B-41–42
- SPLASH/SPLASH 2 (Stanford Parallel Applications for Shared-Memory), 664–66
- Split caches, 470
- Square root instructions, B-79
- Stack architectures, CD2.20:3
- Stack pointers
 - adjustment, 116
 - defined, 114
 - values, 116
- Stacks
 - allocating space on, 119
 - for arguments, 156
 - defined, 114
 - pop, 114
 - push, 114, 116
 - recursive procedures, B-29–30
- Stack segment, B-22
- Stalls, 338–39
 - avoiding with code reordering, 338–39
 - behavioral Verilog with detection, CD4.12:5–9
 - data hazards and, 371–74
 - defined, 338
 - illustrations, CD4.12:25, CD4.12:28–30
 - insertion into pipeline, 374
 - load-use, 377
 - memory, 478
 - as solution to control hazard, 340
 - write-back scheme, 476
 - write buffer, 476
- Standby spares, 605
- State
 - in 2-bit prediction scheme, 381
 - assignment, C-70, D-27
 - bits, D-8
 - exception, saving/restoring, 515
 - logic components, 305
 - specification of, 496
- State elements
 - clock and, 306
 - combinational logic and, 306
 - defined, 305, C-48
 - inputs, 305
 - register file, C-50
 - in storing/accessing instructions, 308
- Static branch prediction, 393
- Static data
 - defined, B-20
 - as dynamic data, B-21
 - segment, 120
- Static multiple-issue processors, 392, 393–97
 - control hazards and, 394
 - instruction sets, 393
 - with MIPS ISA, 394–97
 - See also* Multiple issue
- Static random access memories (SRAMs), C-58–62
 - array organization, C-62
 - basic structure, C-61
 - defined, 20, C-58
 - fixed access time, C-58
 - large, C-59
 - read/write initiation, C-59
 - synchronous (SSRAMs), C-60
 - three-state buffers, C-59, C-60
- Static variables, 118
- Status register, 590
 - fields, B-34, B-35
 - illustrated, 591
- Steady-state prediction, 380
- Sticky bits, 268
- Storage
 - disk, 575–79
 - flash, 580–82
 - nonvolatile, 575
- Storage area networks (SANs), CD6.11:11
- Store buffers, 403
- Stored program concept, 77
 - as computer principle, 100
 - illustrated, 101
 - principles, 176
- Store instructions
 - access, A-41
 - base register, 319
 - block, 165
 - compiling with, 85
 - conditional, 138–39
 - defined, 85
 - details, B-68–70
 - EX stage, 353
 - floating-point, B-79
 - ID stage, 349
 - IF stage, 349
 - instruction dependency, 371
 - list of, B-68–70
 - MEM stage, 354
 - unit for implementing, 311
 - WB stage, 354
 - See also* Load instructions

- Store word, 85
 - Strcpy procedure, 124–25
 - defined, 124
 - as leaf procedure, 126
 - pointers, 126
 - See also* Procedures
 - Stream benchmark, 675
 - Streaming multiprocessor (SM), A-48–49
 - Streaming processors, 657, A-34
 - array (SPA), A-41, A-46
 - GeForce 8800, A-49–50
 - Streaming SIMD Extension 2 (SSE2)
 - floating-point architecture, 274–75
 - Stretch computer, CD4.15:1
 - Strings
 - defined, 124
 - in Java, 126–27
 - representation, 124
 - Striping, 601
 - Strong scaling, 637, 638
 - Structural hazards, 335–36, 352
 - Structured Query Language (SQL), CD6.14:5
 - Subnormals, 270
 - Subtracks, 606
 - Subtraction, 224–29
 - binary, 224–25
 - floating-point, 259, B-79–80
 - instructions, B-56–57
 - negative number, 226
 - overflow, 226
 - See also* Arithmetic
 - Subword parallelism, E-17
 - Sum of products, C-11, C-12
 - Sun Fire x4150 server, 606–12
 - front/rear illustration, 608
 - idle and peak power, 612
 - logical connections and bandwidths, 609
 - minimum memory, 611
 - Sun UltraSPARC T2 (Niagara 2), 647, 658
 - base versus fully optimized performance, 683
 - characteristics, 677
 - defined, 677
 - illustrated, 676
 - LBMHD performance, 682
 - roofline model, 678
 - SpMV performance, 681
 - Supercomputers, 5, CD4.15:1
 - SuperH, E-15, E-39–40
 - Superscalars
 - defined, 397, CD4.15:4
 - dynamic pipeline scheduling, 398, 399–400
 - multithreading options, 646
 - Surfaces, A-41
 - Swap procedure, 149–50
 - body code, 150
 - defined, 149
 - full, 150, 151
 - register allocation, 149–50
 - See also* Procedures
 - Swap space, 498
 - Switched networks, CD6.11:5
 - Switches, CD6.11:6–7
 - Symbol tables, 141, B-12, B-13
 - Synchronization, 137–39
 - barrier, A-18, A-20, A-34
 - defined, 639
 - lock, 137
 - overhead, reducing, 43
 - unlock, 137
 - Synchronizers
 - defined, C-76
 - from D flip-flop, C-76
 - failure, C-77
 - Synchronous bus, 583
 - Synchronous DRAM (SRAM), 473, C-60, C-65
 - Synchronous SRAM (SSRAM), C-60
 - Synchronous system, C-48
 - Syntax tree, CD2.15:3
 - System calls, B-43–45
 - code, B-43–44
 - defined, 509
 - loading, B-43
 - System Performance Evaluation Cooperative. *See* SPEC
 - Systems software, 10
 - SystemVerilog
 - cache controller, CD5.9:1–9
 - cache data and tag modules, CD5.9:5
 - FSM, CD5.9:6–9
 - simple cache block diagram, CD5.9:3
 - type declarations, CD5.9:1, CD5.9:2
- ## T
- Tags
 - defined, 458
 - in locating block, 484
 - page tables and, 498
 - size of, 486–87
 - Tail call, 121
 - Task identifiers, 510
 - Task parallelism, A-24
 - TCP/IP packet format, CD6.11:4
 - Telsa PTX ISA, A-31–34
 - arithmetic instructions, A-33
 - barrier synchronization, A-34
 - GPU thread instructions, A-32
 - memory access instructions, A-33–34
 - Temporal locality, 453
 - defined, 452
 - tendency, 456
 - Temporary registers, 81, 115
 - Terabytes, 5
 - Tesla multiprocessor, 658
 - Text segment, B-13
 - Texture memory, A-40
 - Texture/processor cluster (TPC), A-47–48
 - TFLOPS multiprocessor, CD7.14:5
 - Thrashing, 517
 - Thread blocks, 659
 - creation, A-23
 - defined, A-19
 - managing, A-30
 - memory sharing, A-20
 - synchronization, A-20
 - Thread dispatch, 659
 - Thread parallelism, A-22
 - Threads
 - creation, A-23
 - CUDA, A-36
 - ISA, A-31–34
 - managing, A-30
 - memory latencies and, A-74–75
 - multiple, per body, A-68–69
 - warps, A-27
 - Three Cs model, 523
 - Three-state buffers, C-59, C-60
 - Throughput
 - defined, 28
 - multiple issue and, 401
 - pipelining and, 344, 401

- Thumb, E-15, E-38
- Timing
 - asynchronous inputs, C-76–77
 - level-sensitive, C-75–76
 - methodologies, C-72–77
 - two-phase, C-75
- TLB misses, 503
 - entry point, 514
 - handler, 514
 - handling, 510–16
 - minimization, 681
 - occurrence, 510
 - problem, 517
 - See also* Translation-lookaside buffer (TLB)
- Tomasulo's algorithm, CD4.15:2
- Tournament branch predictors, 383
- Tracks, 575
- Transaction Processing Council (TPC), 596
- Transaction processing (TP)
 - defined, 596
 - I/O benchmarks, 596–97
- Transfer time, 576
- Transistors, 26
- Translation-lookaside buffer (TLB), 502–4, CD5.13:5
 - associativities, 503
 - defined, 502
 - illustrated, 502
 - integration, 504–8
 - Intrinsic FastMATH, 504
 - MIPS-64, E-26–27
 - typical values, 503
 - See also* TLB misses
- Transmitter Control register, B-39–40
- Transmitter Data register, B-40
- Trap instructions, B-64–66
- Tree-based parallel scan, A-62
- Truth tables, C-5
 - ALU control lines, D-5
 - for control bits, 318
 - datapath control outputs, D-17
 - datapath control signals, D-14
 - defined, 317
 - example, C-5
 - next-state output bits, D-15
 - PLA implementation, C-13
- Two-level logic, C-11–14
- Two-phase clocking, C-75
- Two's complement representation, 89, 90
 - advantage, 90
 - defined, 89
 - negation shortcut, 91–92
 - rule, 93
 - sign extension shortcut, 92–93
- TX-2 computer, CD7.14:3
- U**
- Unconditional branches, 106
- Underflow, 245
- Unicode
 - alphabets, 126
 - defined, 126
 - example alphabets, 127
- Unified GPU architecture, A-10–12
 - illustrated, A-11
 - processor array, A-11–12
- Uniform memory access (UMA), 638–39, A-9
 - defined, 638
 - multiprocessors, 639
- Units
 - commit, 399, 402
 - control, 303, 316–17, D-4–8, D-10, D-12–13
 - defined, 267
 - floating point, 267
 - hazard detection, 372, 373
 - for load/store implementation, 311
 - rank, 606, 607
 - special function (SFUs), A-35, A-43, A-50
- UNIVAC I, CD1.10:4
- UNIX, CD2.20:7, CD5.13:8–11
 - AT&T, CD5.13:9
 - Berkeley version (BSD), CD5.13:9
 - genius, CD5.13:11
 - history, CD5.13:8–11
- Unlock synchronization, 137
- Unresolved references
 - defined, B-4
 - linkers and, B-18
- Unsigned numbers, 87–94
- Use latency
 - defined, 395
 - one-instruction, 396
- V**
- Vacuum tubes, 26
- Valid bit, 458
- Variables
 - C language, 118
 - programming language, 81
 - register, 81
 - static, 118
 - storage class, 118
 - type, 118
- VAX architecture, CD2.20:3, CD5.13:6
- Vectored interrupts, 386
- Vector processors, 650–53
 - conventional code comparison, 650–51
 - instructions, 652
 - multimedia extensions and, 653
 - scalar versus, 652
 - See also* Processors
- Verilog
 - behavioral definition of MIPS ALU, C-25
 - behavioral definition with bypassing, CD4.12:4–5
 - behavioral definition with stalls for loads, CD4.12:6–7, CD4.12:8–9
 - behavioral specification, C-21, CD4.12:2–3
 - behavioral specification of multicycle MIPS design, CD4.12:11–12
 - behavioral specification with simulation, CD4.12:1–5
 - behavioral specification with stall detection, CD4.12:5–9
 - behavioral specification with synthesis, CD4.12:10–16
 - blocking assignment, C-24
 - branch hazard logic implementation, CD4.12:7–9
 - combinational logic, C-23–26
 - datatypes, C-21–22
 - defined, C-20
 - forwarding implementation, CD4.12:3
 - MIPS ALU definition in, C-35–38
 - modules, C-23
 - multicycle MIPS datapath, CD4.12:13
 - nonblocking assignment, C-24
 - operators, C-22
 - program structure, C-23

- Verilog (*continued*)
 - reg, C-21–22
 - sensitivity list, C-24
 - sequential logic specification, C-56–58
 - structural specification, C-21
 - wire, C-21–22
 - Vertical microcode, D-32
 - Very large-scale integrated (VLSI) circuits, 26
 - Very Long Instruction Word (VLIW)
 - defined, 393
 - first generation computers, CD4.15:4
 - processors, 394
 - VHDL, C-20–21
 - Video graphics array (VGA) controllers, A-3–4
 - Virtual addresses
 - causing page faults, 514
 - defined, 493
 - mapping from, 494
 - size, 495
 - Virtualizable hardware, 527
 - Virtually addressed caches, 508
 - Virtual machine monitors (VMMs)
 - defined, 526
 - implementing, 545–47
 - laissez-faire attitude, 546
 - page tables, 529
 - in performance improvement, 528
 - requirements, 527
 - Virtual machines (VMs), 525–29
 - benefits, 526
 - defined, B-41
 - illusion, 529
 - instruction-set architecture support, 527–28
 - performance improvement, 528
 - for protection improvement, 526
 - simulation of, B-41–42
 - Virtual memory, 492–517
 - address translation, 493, 502–4
 - defined, 492
 - integration, 504–8
 - mechanism, 516
 - motivations, 492–93
 - page faults, 493, 498
 - protection implementation, 508–10
 - segmentation, 495
 - summary, 516
 - virtualization of, 529
 - writes, 501
 - See also* Pages
 - Visual computing, A-3
 - Volatile memory, 21
- W**
- Wafers, 46
 - defects, 46
 - defined, 45
 - dies, 46
 - yield, 46
 - Warps, 657, A-27
 - Weak scaling, 637
 - Wear leveling, 581
 - Web server benchmark (SPECWeb), 597
 - While loops, 107–8
 - Whirlwind, CD5.13:1, CD5.13:3
 - Wide area networks (WANs), CD6.14:7–8
 - defined, 25
 - history of, CD6.14:7–8
 - See also* Networks
 - Winchester disk, CD6.14:2–4
 - Wireless LANs, CD6.11:8–10
 - Words
 - accessing, 82
 - defined, 81
 - double, 168
 - load, 83, 85
 - quad, 168
 - store, 85
 - Working set, 517
 - Worst-case delay, 330
 - Write-back caches
 - advantages, 522
 - cache coherency protocol, CD5.9:12
 - complexity, 468
 - defined, 467, 521
 - stalls, 476
 - write buffers, 468
 - See also* Caches
 - Write-back stage
 - control line, 362
 - load instruction, 350
 - store instruction, 352
 - Write buffers
 - defined, 467
 - stalls, 476
 - write-back cache, 468
 - Write invalidate protocols, 536, 537
 - Writes
 - complications, 467
 - expense, 516
 - handling, 466–68
 - memory hierarchy handling of, 521–22
 - schemes, 467
 - virtual memory, 501
 - write-back cache, 467, 468
 - write-through cache, 467, 468
 - Write serialization, 535–36
 - Write-stall cycles, 476
 - Write-through caches
 - advantages, 522
 - defined, 467, 521
 - tag mismatch, 468
 - See also* Caches
- X**
- X86, 165–74
 - brief history, CD2.20:5
 - conclusion, 172
 - data addressing modes, 168, 170
 - evolution, 165–68
 - first address specifier encoding, 174
 - floating point, 272–74
 - floating-point instructions, 273
 - historical timeline, 166–67
 - instruction encoding, 171–72
 - instruction formats, 173
 - instruction set growth, 176
 - instruction types, 169
 - integer operations, 168–71
 - I/O interconnects, 584–86
 - registers, 168
 - SIMD in, 649–50
 - typical instructions/functions, 171
 - typical operations, 172
 - Xerox Alto computer, CD1.10:7–8