

# Index

*Note:* Online information is listed by chapter and section number followed by page numbers (OL3.11-7). Page references preceded by a single letter with hyphen refer to appendices.

1-bit ALU, A-614–617. *See also*  
Arithmetic logic unit (ALU)  
adder, A-615  
CarryOut, A-616  
for most significant bit, A-621  
illustrated, A-617  
logical unit for AND/OR, A-615  
performing AND, OR, and addition,  
A-619, A-621  
64-bit ALU, A-617–626. *See also*  
Arithmetic logic unit (ALU)  
defining in Verilog, A-623–626  
from 31 copies of 1-bit ALU, A-622  
illustrated, A-624  
ripple carry adder, A-617  
tailoring to MIPS, A-619–623  
with 32 1-bit ALUs, A-618  
7090/7094 hardware, OL3.12-7

## A

AArch32, 73  
AArch64, 73  
Absolute references, 131  
Abstractions  
hardware/software interface, 22  
principle, 22  
to simplify design, 11  
Accumulator architectures, OL2.22-2  
Acronyms, 9  
Active matrix, 18  
ADD (add), 64  
ADDI (add immediate), 64  
ADDIS (add immediate and set flags), 64  
Addition, 188–191. *See also* Arithmetic  
binary, 188–189  
floating-point, 212–215, 220  
operands, 189  
significands, 211  
speed, 191  
Address interleaving, 395  
Address select logic, C-24, C-25  
Address space, 442, 445  
extending, 493  
flat, 493  
ID (ASID), 460  
inadequate, OL5.17-6  
shared, 533–534  
single physical, 533–534  
virtual, 460  
Address translation  
for ARM cortex-A8, 483  
defined, 443  
fast, 452–454  
for Intel core i7, 483  
TLB for, 452–454  
Address-control lines, C-26  
Addresses  
base, 69  
byte, 70  
defined, 69  
memory, 79  
virtual, 442, 462, 463  
Addressing  
base, 120  
in branches, 117–120  
displacement, 120  
immediate, 120  
LEGv8 modes, 120–121  
PC-relative, 118, 120  
register, 120  
x86 modes, 158  
Addressing modes  
desktop architectures, D-6  
ADDS (add and set flags), 64, 164  
addu (Add Unsigned), 64  
Advanced Encryption Standard (AES)  
encryption, 488  
Advanced Vector Extensions (AVX), 232,  
240  
AGP, B-9  
Algol-60, OL2.22-7  
Aliasing, 458, 459  
Alignment restriction, 71  
All-pairs N-body algorithm, B-65  
Alpha architecture  
bit count instructions, D-29  
floating-point instructions, D-28  
instructions, D-27–29  
no divide, D-28  
PAL code, D-28  
unaligned load-store, D-28  
VAX floating-point formats, D-29  
ALU control, 271–273. *See also*  
Arithmetic logic unit (ALU)  
bits, 272  
logic, C-6  
mapping to gates, C-4–7  
truth tables, C-5  
ALU control block, 275  
defined, C-4  
generating ALU control bits, C-6  
ALUOp, 272, C-6  
bits, 272, 273  
control signal, 275  
Amazon Web Services (AWS), 439  
AMD Opteron X4 (Barcelona), 559, 560  
AMD64, 155, 156, 232, OL2.22-6  
Amdahl's law, 415, 519  
corollary, 49  
defined, 49  
fallacy, 572  
and (AND), 64  
AND gates, A-600, C-7  
AND operation, 91  
AND operation, A-594  
andi (And Immediate), 65  
Annual failure rate (AFR), 432  
*versus* MTTF of disks, 433–434  
Antidependence, 348  
Antifuse, A-666  
Apple computer, OL1.12-7  
Apple iPad 2 A1395, 20  
logic board of, 20  
processor integrated circuit of, 21  
Application binary interface (ABI), 22  
Application programming interfaces (APIs)  
defined, B-4  
graphics, B-14

- Architectural registers, 358
  - Arithmetic, 186–248
    - addition, 188–191
    - addition and subtraction, 188–191
    - division, 197–204
    - fallacies and pitfalls, 242–245
    - floating-point, 205–230
    - historical perspective, 248
    - multiplication, 191–197
    - parallelism and, 230–232
    - Streaming SIMD Extensions and
      - advanced vector extensions in x86, 232–233
    - subtraction, 188–191
    - subword parallelism, 230–232
    - subword parallelism and matrix multiply, 238–242
  - Arithmetic instructions. *See also* Instructions
    - desktop RISC, D-11
    - embedded RISC, D-14
    - logical, 263
    - operands, 67–74
  - Arithmetic intensity, 557
  - Arithmetic logic unit (ALU). *See also* ALU control; Control units
    - 1-bit, A-614–617
    - 64-bit, A-617–626
    - before forwarding, 321
    - branch datapath, 266
    - hardware, 190
    - memory-reference instruction use, 257
    - for register values, 264
    - R-format operations, 265
    - signed-immediate input, 323
  - ARM Cortex-A53, 256, 355–358
    - address translation for, 483
    - caches in, 484
    - data cache miss rates for, 485
    - memory hierarchies of, 482
    - performance of, 485–488
    - specification, 356
    - TLB hardware for, 483
  - ARM instructions, 152–154
    - 12-bit immediate field, 153
    - brief history, OL2.22-5
    - condition field, 334
    - unique, D-36–37
  - ARMv7, 62
  - ARMv8, 62, 163–169
    - common features between MIPS and, 152
  - ARPANET, OL1.12-10
  - Arrays, 429
    - logic elements, A-606–607
    - multiple dimension, 226
    - pointers *versus*, 146–150
    - procedures for setting to zero, 147
  - ASCII
    - binary numbers *versus*, 111
    - character representation, 110
    - defined, 110
    - symbols, 113
  - Assemblers, 129–131
    - defined, 14
    - function, 129
    - microcode, C-30
    - number acceptance, 130
    - object file, 130
  - Assembly language, 15
    - defined, 14, 129
    - floating-point, 221
    - illustrated, 15
    - LEGv8, 64, 86
    - programs, 129
    - translating into machine language, 86
  - Asserted signals, 262, A-592
  - Associativity
    - in caches, 419
    - degree, increasing, 418, 466
    - increasing, 423
    - set, tag size *versus*, 423
  - Atomic compare and swap, 127
  - Atomic exchange, 126
  - Atomic fetch-and-increment, 127
  - Atomic memory operation, B-21
  - Attribute interpolation, B-43–44
  - Automobiles, computer application in, 4
  - Average memory access time (AMAT), 416
    - calculating, 416
- ## B
- Bandwidth, 30
    - bisection, 551
    - external to DRAM, 412
    - memory, 394–395, 412
    - network, 549
  - Barrier synchronization, B-18
    - defined, B-20
    - for thread communication, B-34
  - Base addressing, 69, 120
  - Base registers, 70
  - Basic block, 96
  - Benchmarks, 554–556
    - defined, 46
    - Linpack, 554, OL3.12-4
    - multicores, 538–545
    - multiprocessor, 554–556
    - NAS parallel, 556
    - parallel, 555
    - PARSEC suite, 556
    - SPEC CPU, 46–48
    - SPEC power, 48–49
    - SPECrate, 554
    - Stream, 564
  - Biased notation, 82, 209
  - Big-endian byte order, 70
  - Binary numbers, 83
    - ASCII *versus*, 111
    - conversion to decimal numbers, 78
    - defined, 75
  - Bisection bandwidth, 551
  - Bit maps
    - defined, 18, 73
    - goal, 18
    - storing, 18
  - Bit-Interleaved Parity (RAID 3), OL5.11-5
  - Bits
    - ALUOp, 272, 273
    - defined, 14
    - dirty, 452
    - guard, 228
    - patterns, 228–229
    - reference, 450
    - rounding, 228
    - sign, 77
    - state, C-8
    - sticky, 228
    - valid, 397
  - Blocking assignment, A-612
  - Blocking factor, 428
  - Block-Interleaved Parity (RAID 4), OL5.11-5–5.11-6
  - Blocks
    - combinational, A-592
    - defined, 390
    - finding, 467
    - flexible placement, 416–418
    - least recently used (LRU), 423
    - locating in cache, 421–422
    - miss rate and, 405
    - multiword, mapping addresses to, 404
    - placement locations, 466
    - placement strategies, 418
    - replacement selection, 423

- replacement strategies, 468
- spatial locality exploitation, 405
- state, A-592
- valid data, 400
- Bonding, 28
- Boolean algebra, A-594
- Bounds check shortcut, 98
- Branch address, 168
- Branch datapath
  - ALU, 266
  - operations, 266
- Branch delay slots
- Branch instructions
  - pipeline impact, 329
- Branch not taken
  - assumption, 328–329
  - defined, 266
- Branch prediction
  - as control hazard solution, 295
  - buffers, 331, 333
  - defined, 294
  - dynamic, 295, 331–334
  - static, 345
- Branch predictors
  - accuracy, 333
  - correlation, 333
  - information from, 333
  - tournament, 334
- Branch register, 168
- Branch table, 169
- Branch taken
  - cost reduction, 330
  - defined, 266
- Branch target
  - addresses, 266
  - buffers, 333
- Branches. *See also* Conditional branches
  - addressing in, 117–120
  - compiler creation, 94
  - decision, moving up, 330
  - delayed, 295, 330–331, 295
  - ending, 96
  - execution in ID stage, 330
  - pipelined, 330
  - target address, 330
  - unconditional, 318
- Branch-on-zero instruction, 280
- B-type instruction format, 113
- Bubble Sort, 145
- Bubbles, 326
- Bus-based coherent multiprocessors,
  - OL6.15-7
- Buses, A-607
- Bytes
  - addressing, 70
  - order, 70
- C**
- C.mmp, OL6.15-4
- C language
  - assignment, compiling into LEGv8, 66
  - compiling, 150, OL2.15-2–2.15-3
  - compiling assignment with registers, 68
  - compiling while loops in, 95–96
  - sort algorithms, 146
  - translation hierarchy, 128
  - translation to LEGv8 assembly
    - language, 66
    - variables, 106
- C++ language, OL2.15-27, OL2.22-8
- Cache blocking and matrix multiply, 489–490
- Cache coherence, 477–481
  - coherence, 477
  - consistency, 477
  - enforcement schemes, 479
  - implementation techniques, OL5.12-5–5.12-12
  - migration, 479
  - problem, 477, 478, 481
  - protocol example, OL5.12-12–5.12-16
  - protocols, 479
  - replication, 479
  - snooping protocol, 479–481
  - snoopy, OL5.12-16–5.12-17
  - state diagram, OL5.12-16
- Cache coherency protocol, OL5.12-12–5.12-16
  - finite-state transition diagram, OL5.12-15
  - functioning, OL5.12-14
  - mechanism, OL5.12-14
  - state diagram, OL5.12-16
  - states, OL5.12-13
  - write-back cache, OL5.12-15
- Cache controllers, 482
  - coherent cache implementation
    - techniques, OL5.12-5–5.12-12
  - implementing, OL5.12-2
  - snoopy cache coherence, OL5.12-16–5.12-17
  - SystemVerilog, OL5.12-2
- Cache hits, 458
- Cache misses
  - block replacement on, 468
  - capacity, 470, 471
  - compulsory, 470
  - conflict, 470
  - defined, 406
  - direct-mapped cache, 418
  - fully associative cache, 420
  - handling, 406–407
  - memory-stall clock cycles, 413
  - reducing with flexible block placement, 416–418
  - set-associative cache, 419
  - steps, 407
  - in write-through cache, 407
- Cache performance, 412–431
  - calculating, 414
  - hit time and, 415–416
  - impact on processor performance, 414
- Cache-aware instructions, 496
- Caches, 397–412. *See also* Blocks
  - accessing, 400–403
  - in ARM cortex-A53, 484
  - associativity in, 419–420
  - bits in, 404
  - bits needed for, 404
  - contents illustration, 401
  - defined, 21, 397–398
  - direct-mapped, 398, 399, 404, 416
  - empty, 400–401
  - FSM for controlling, 472
  - fully associative, 417
  - GPU, B-38
  - inconsistent, 407
  - index, 402
  - in Intel Core i7, 484
  - Intrinsity FastMATH example, 409–412
  - locating blocks in, 421–422
  - locations, 399
  - multilevel, 412, 424
  - nonblocking, 483
  - physically addressed, 458, 459
  - physically indexed, 458
  - physically tagged, 458
  - primary, 424, 431
  - secondary, 424, 431
  - set-associative, 417
  - simulating, 491
  - size, 403
  - split, 411
  - summary, 411–412
  - tag field, 402
  - tags, OL5.12-3, OL5.12-11

- Caches (*Continued*)
  - virtual memory and TLB integration, 457–459
  - virtually addressed, 458
  - virtually indexed, 458
  - virtually tagged, 458
  - write-back, 408, 409, 469
  - write-through, 407, 409, 469
  - writes, 407–409
- Callee, 101, 103
- Caller, 101
- Capabilities, OL5.17-8
- Capacity misses, 470
- Carry lookahead, A-626–635
  - 4-bit ALUs using, A-633
  - adder, A-627
  - fast, with first level of abstraction, A-627–628
  - fast, with “infinite” hardware, A-626–627
  - fast, with second level of abstraction, A-628–634
  - plumbing analogy, A-630, A-631
  - ripple carry speed *versus*, A-634
  - summary, A-634–635
- Carry save adders, 197
- Cause register
- CDC 6600, OL1.12-7, OL4.16-3
- Cell phones, 7
- Central processor unit (CPU). *See also* Processors
  - classic performance equation, 36–40
  - defined, 19
  - execution time, 32, 33–34
  - performance, 33–35
  - system, time, 32
  - time, 413
  - time measurements, 33–34
  - user, time, 32
- Cg pixel shader program, B-15–17
- Characters
  - ASCII representation, 110
  - in Java, 113
- Chips, 19, 25, 26
  - manufacturing process, 26
- Classes
  - defined, OL2.15-15
  - packages, OL2.15-21
- Clear exclusive instruction (CLREX), 488
- Clock cycles
  - defined, 33
  - memory-stall, 413
  - number of registers and, 67
  - worst-case delay and, 283
- Clock cycles per instruction (CPI), 35, 293
  - one level of caching, 424
  - two levels of caching, 424
- Clock rate
  - defined, 33
  - frequency switched as function of, 41
  - power and, 40
- Clocking methodology, 261–263, A-636
  - edge-triggered, 261, A-636, A-661
  - level-sensitive, A-662, A-663–664
  - for predictability, 261
- Clocks, A-636–638
  - edge, A-636, A-638
  - in edge-triggered design, A-661
  - skew, A-662
  - specification, A-645
  - synchronous system, A-636–637
- Cloud computing, 549
  - defined, 7
- Cluster networking, 553–554, OL6.9-12
- Clusters, OL6.15-8–6.15-9
  - defined, 516, 546, OL6.15-8
  - isolation, 547
  - organization, 515
  - scientific computing on, OL6.15-8
- Cm\*, OL6.15-4
- CMOS (complementary metal oxide semiconductor), 41
- Coarse-grained multithreading, 530
- Cobol, OL2.22-7
- Code generation, OL2.15-13
- Code motion, OL2.15-7
- Cold-start miss, 470
- Collision misses, 470
- Column major order, 427
- Combinational blocks, A-592
- Combinational control units, C-4–8
- Combinational elements, 260
- Combinational logic, 261, A-591, A-597–608
  - arrays, A-606–607
  - decoders, A-597
  - defined, A-593
  - don’t cares, A-605–606
  - multiplexors, A-598
  - ROMs, A-602–604
  - two-level, A-599–602
  - Verilog, A-611–14
- Commercial computer development, OL1.12-4–1.12-10
- Commit units
  - buffer, 350
  - defined, 350
  - in update control, 355
- Common case fast, 11
- Common subexpression elimination, OL2.15-6
- Communication, 23–24
  - overhead, reducing, 44–45
  - thread, B-34
- Compact code, OL2.22-4
- Compare and branch zero, 330
- Comparisons
  - constant operands in, 73
  - signed *versus* unsigned, 97
- Compilers, 129
  - branch creation, 95
  - brief history, OL2.22-8–2.22-9
  - conservative, OL2.15-7
  - defined, 14
  - front end, OL2.15-3
  - function, 14, 129
  - high-level optimizations, OL2.15-4
  - ILP exploitation, OL4.16-5
  - Just In Time (JIT), 137
  - optimization, 146, OL2.22-9
  - speculation, 344–345
  - structure, OL2.15-2
- Compiling
  - C assignment statements, 66
  - C language, 95, 150, OL2.15-2–2.15-3
  - floating-point programs, 222–225
  - if-then-else, 94
  - in Java, OL2.15-19
  - procedures, 102, 104–105
  - recursive procedures, 104–105
  - while loops, 95–96
- Compressed sparse row (CSR) matrix, B-55, B-56
- Compulsory misses, 470, 471
- Computer architects, 11–12
  - abstraction to simplify design, 11
  - common case fast, 11
  - dependability via redundancy, 12
  - hierarchy of memories, 12
  - Moore’s law, 11
  - parallelism, 12
  - pipelining, 12
  - prediction, 12

- Computers
    - application classes, traditional, 5–6
    - applications, 4
    - arithmetic for, 186–248
    - characteristics, OL1.12–12
    - commercial development, OL1.12–4–1.12–10
    - component organization, 17
    - components, 17, 177
    - design measure, 53
    - desktop, 5
    - embedded, 5
    - first, OL1.12–2–1.12–4
    - in information revolution, 4
    - instruction representation, 82–89
    - performance measurement, OL1.12–10
    - post-PC era, 6–7
    - principles, 86
    - servers, 5
  - Condition codes/flags, 97
  - Conditional branches
    - changing program counter with, 333
    - compiling if-then-else into, 94
    - defined, 93
    - desktop RISC, D-16
    - embedded RISC, D-16
    - implementation, 99
    - in loops, 119
    - PA-RISC, D-34, D-35
    - PC-relative addressing, 118
    - RISC, D-10–16
    - SPARC, D-10–12
  - Conditional move instructions, 334
  - Conflict misses, 470
  - Constant memory, B-40
  - Constant operands, 73–74
    - frequent occurrence, 73
  - Content Addressable Memory (CAM), 422
  - Context switch, 460
  - Control
    - ALU, 271–273
    - challenge, 336
    - finishing, 281
    - forwarding, 320
    - FSM, C-8–21
    - implementation, optimizing, C-27–28
    - mapping to hardware, C-2–32
    - memory, C-26
    - organizing, to reduce logic, C-31–32
    - pipelined, 311–315
  - Control flow graphs, OL2.15–9–2.15–10
    - illustrated examples, OL2.15–9, OL2.15–10
  - Control functions
    - ALU, mapping to gates, C-4–7
    - defining, 276
    - PLA, implementation, C-7, C-20–21
    - ROM, encoding, C-18–19
    - for single-cycle implementation, 281
  - Control hazards, 292–295, 328–329
    - branch delay reduction, 330
    - branch not taken assumption, 328
    - branch prediction as solution, 295
    - delayed decision approach, 295
    - dynamic branch prediction, 331
    - logic implementation in Verilog, OL4.13–8
    - pipeline stalls as solution, 293
    - pipeline summary, 335–336
    - simplicity, 328
    - solutions, 293
    - static multiple-issue processors and, 345–346
  - Control lines
    - asserted, 276
    - in datapath, 275
    - execution/address calculation, 312
    - final three stages, 314
    - instruction decode/register file read, 312
    - instruction fetch, 312
    - memory access, 312
    - setting of, 276
    - values, 312
    - write-back, 312
  - Control signals
    - ALUOp, 275
    - defined, 262
    - effect of, 276
    - multi-bit, 276
    - pipelined datapaths with, 311–315
    - truth tables, C-14
  - Control units, 259. *See also* Arithmetic logic unit (ALU)
    - address select logic, C-24, C-25
    - combinational, implementing, C-4–8
    - with explicit counter, C-23
    - illustrated, 277
    - logic equations, C-11
    - main, designing, 273–276
    - as microcode, C-28
    - MIPS, C-10
    - next-state outputs, C-10, C-12–13
    - output, 271–273, C-10
  - Cooperative thread arrays (CTAs), B-30
  - Coprocessors
    - defined, 226
  - Core LEGv8 instruction set, 248. *See also* MIPS
    - abstract view, 258
    - desktop RISC, D-9–11
    - implementation, 256–260
    - implementation illustration, 259
    - overview, 257–260
    - subset, 256
  - Cores
    - defined, 43
    - number per chip, 43
  - Correlation predictor, 333
  - Cosmic Cube, OL6.15–7
  - CPU, 9
  - Cray computers, OL3.12–5–3.12–6
  - Critical word first, 406
  - Crossbar networks, 551
  - CTSS (Compatible Time-Sharing System), OL5.18–9
  - CUDA programming environment, 539, B-5
    - barrier synchronization, B-18, B-34
    - development, B-17, B-18
    - hierarchy of thread groups, B-18
    - kernels, B-19, B-24
    - key abstractions, B-18
    - paradigm, B-19–23
    - parallel plus-scan template, B-61
    - per-block shared memory, B-58
    - plus-reduction implementation, B-63
    - programs, B-6, B-24
    - scalable parallel programming with, B-17–23
    - shared memories, B-18
    - threads, B-36
  - Cyclic redundancy check, 437
  - Cylinder, 396
- D**
- D flip-flops, A-639, A-641
  - D latches, A-639, A-640
  - Data bits, 435
  - Data flow analysis, OL2.15–11
  - Data hazards, 289–292, 316–328.
    - See also* Hazards
    - forwarding, 289, 316–328
    - load-use, 290, 329
    - stalls and, 324–328
  - Data parallel problem decomposition, B-17, B-18

- Data race, 125
- Data selectors, 258
- Data transfer instructions. *See also* Instructions
  - defined, 68, 69
  - load, 69
  - offset, 70
  - store, 71–72
- Datacenters, 7
- Data-level parallelism, 524
- Datapath elements
  - defined, 263
  - sharing, 268
- Datapaths
  - branch, 266
  - building, 263–271
  - control signal truth tables, C-14
  - control unit, 277
  - defined, 19
  - design, 263
  - exception handling, 339
  - for fetching instructions, 265
  - for hazard resolution via forwarding, 323
  - for LEGv8 architecture, 269
  - for memory instructions, 267
  - in operation for branch-on-zero instruction, 280
  - in operation for load instruction, 279
  - in operation for R-type instruction, 277, 278
  - operation of, 276–280
  - pipelined, 297–315
  - for R-type instructions, 278, 276–277
  - single, creating, 267
  - single-cycle, 296
  - static two-issue, 347
- Deasserted signals, 262, A-592
- DEC PDP-8, OL2.22-3
- Decimal numbers
  - binary number conversion to, 78
  - defined, 75
- Decision-making instructions, 93–99
- Decoders, A-597
  - two-level, A-653
- Decoding machine language, 121–125
- Defect, 26
- Delayed branches, 295. *See also* Branches
  - as control hazard solution, 295
  - embedded RISCs and, D-23
  - for five-stage pipelines, 323–324
  - reducing, 330
- Delayed decision, 295
- DeMorgan's theorems, A-599
- Denormalized numbers, 230
- Dependability via redundancy, 12
- Dependable memory hierarchy, 432–437
  - failure, defining, 432
- Dependencies
  - between pipeline registers, 319
  - between pipeline registers and ALU inputs, 319
  - bubble insertion and, 326
  - detection, 318
  - name, 348
  - sequence, 316
- Design
  - compromises and, 85
  - datapath, 263
  - digital, 366
  - logic, 260–263, B-1–79
  - main control unit, 273–276
  - memory hierarchy, challenges, 472
  - pipelining instruction sets, 288
- Desktop and server RISCs. *See also* Reduced instruction set computer (RISC) architectures
  - addressing modes, D-6
  - architecture summary, D-4
  - arithmetic/logical instructions, D-11
  - conditional branches, D-16
  - constant extension summary, D-9
  - control instructions, D-11
  - conventions equivalent to MIPS core, D-12
  - data transfer instructions, D-10
  - features added to, D-45
  - floating-point instructions, D-12
  - instruction formats, D-7
  - multimedia extensions, D-16–18
  - multimedia support, D-18
  - types of, D-3
- Desktop computers, defined, 5
- Device driver, OL6.9-5
- DGEMM (Double precision General Matrix Multiply), 238, 363, 365, 427, 553
  - cache blocked version of, 429
  - optimized C version of, 241, 363, 490
  - performance, 365, 430
- Dicing, 27
- Dies, 26, 26–27
- Digital design pipeline, 366
- Digital signal-processing (DSP) extensions, D-19
- DIMMs (dual inline memory modules), OL5.17-5
- Direct Data IO (DDIO), OL6.9-6
- Direct memory access (DMA), OL6.9-4
- Direct3D, B-13
- Direct-mapped caches. *See also* Caches
  - address portions, 421
  - choice of, 422
  - defined, 398, 416
  - illustrated, 399
  - memory block location, 417
  - misses, 419
  - single comparator, 421
  - total number of bits, 404
- Dirty bit, 452
- Dirty pages,
- Disk memory, 395–397
- Displacement addressing, 120
- Distributed Block-Interleaved Parity (RAID 5), OL5.11-6
- Divide algorithm, 200
- Dividend, 198
- Division, 197–203
  - algorithm, 199
  - dividend, 198
  - divisor, 198
- Divisor, 198
- divu (Divide Unsigned). *See also* Arithmetic
  - faster, 202–203
  - floating-point, 220
  - hardware, 198–201
  - hardware, improved version, 201
  - in LEGv8, 203
  - operands, 198
  - quotient, 198
  - remainder, 198
  - signed, 201–202
  - SRT, 203
- Don't cares, A-605–606
  - example, A-605–606
  - term, 273
- Double data rate (DDR), 393
- Double Data Rate (DDR) SDRAM, 393–394, A-653
- Double precision. *See also* Single precision
  - defined, 207
  - FMA, B-45–46
  - GPU, B-45–46, B-74
  - representation, 206–207
- Doubleword, 66, 158
- Dual inline memory modules (DIMMs), 395
- Dynamic branch prediction, 331–334. *See also* Control hazards
  - branch prediction buffer, 331
  - loops and, 333



- Dynamic hardware predictors, 295
  - Dynamic multiple-issue processors, 343, 349–352. *See also* Multiple issue pipeline scheduling, 350–352 superscalar, 349
  - Dynamic pipeline scheduling, 350–352
    - commit unit, 350
    - concept, 350
    - hardware-based speculation, 352
    - primary units, 351
    - reorder buffer, 355
    - reservation station, 350
  - Dynamic random access memory (DRAM), 392, 393–395, A-651–653
    - bandwidth external to, 412
    - cost, 23
    - defined, 19, A-651
    - DIMM, OL5.17-5
    - Double Data Rate (DDR), 393–394
    - early board, OL5.17-4
    - GPU, B-37–38
    - growth of capacity, 25
    - history, OL5.17-2
    - internal organization of, 394
    - pass transistor, A-651
    - SIMM, OL5.17-5, OL5.17-6
    - single-transistor, A-652
    - size, 412
    - speed, 23
    - synchronous (SDRAM), 393–394, A-648, A-653
    - two-level decoder, A-653
  - Dynamically linked libraries (DLLs), 134–136
    - defined, 134
    - lazy procedure linkage version, 135
- ## E
- Early restart, 406
  - Edge-triggered clocking methodology, 261, 262, A-636, A-661
    - advantage, A-637
    - clocks, A-661
    - drawbacks, A-662
    - illustrated, A-638
    - rising edge/falling edge, A-636
  - EDSAC (Electronic Delay Storage Automatic Calculator), OL1.12-3, OL5.17-2
  - Eispack, OL3.12-4
  - Electrically erasable programmable read-only memory (EEPROM), 395
  - Elements
    - combinational, 260
    - datapath, 263, 268
    - memory, A-638–646
    - state, 260, 262, 264, A-636, A-638
  - Embedded computers, 5
    - application requirements, 6
    - design, 5
    - growth, OL1.12-12–1.12-13
  - Embedded Microprocessor Benchmark Consortium (EEMBC), OL1.12-12
  - Embedded RISCs. *See also* Reduced instruction set computer (RISC) architectures
    - addressing modes, D-6
    - architecture summary, D-4
    - arithmetic/logical instructions, D-14
    - conditional branches, D-16
    - constant extension summary, D-9
    - control instructions, D-15
    - data transfer instructions, D-13
    - delayed branch and, D-23
    - DSP extensions, D-19
    - general purpose registers, D-5
    - instruction conventions, D-15
    - instruction formats, D-8
    - multiply-accumulate approaches, D-19
    - types of, D-4
  - Encoding
    - defined, C-31
    - LEGv8 instruction, 86, 122
    - ROM control function, C-18–19
    - ROM logic function, A-603
    - x86 instruction, 161–162
  - ENIAC (Electronic Numerical Integrator and Calculator), OL1.12-2, OL1.12-3, OL5.17-2
  - EPIC, OL4.16-5
  - Error correction, A-653–655
  - Error Detecting and Correcting Code (RAID 2), OL5.11-5
  - Error detection, A-654
  - Error detection code, 434
  - Ethernet, 23
  - EX stage
    - load instructions, 303
    - overflow exception detection, 338, 341
    - store instructions, 305
  - Exabyte, 6
  - Exception enable, 461
  - Exception link register (ELR), 337, 459, 461
    - address capture, 340
    - defined, 338
    - in restart determination, 337
  - Exception program counters (EPCs), 326
    - address capture, 331
    - copying, 181
    - defined, 181, 327
    - in restart determination, 326–327
    - transferring, 182
  - Exception Syndrome Register (ESR), 337, 461
  - Exceptions, 336–342
    - association, 342
    - datapath with controls for handling, 339
    - defined, 207, 336
    - detecting, 336
    - event types and, 336
    - imprecise, 342
    - interrupts *versus*, 336
    - in LEGv8 architecture, 337–338
    - overflow, 339
    - pipelined computer example, 339
    - in pipelined implementation, 338–342
    - precise, 342
    - reasons for, 337–338
    - result due to overflow in add instruction, 341
    - saving/restoring stage on, 462
  - Executable files
    - defined, 131
  - Execute or address calculation stage, 303
  - Execute/address calculation
    - control line, 312
    - load instruction, 303
    - store instruction, 303
  - Execution time
    - as valid performance measure, 51
    - CPU, 32, 33–34
    - pipelining and, 297
  - Explicit counters, C-23, C-26
  - Exponents, 206
  - Extended-register instructions, 164
- ## F
- Failures, synchronizer, A-665
  - Fallacies. *See also* Pitfalls
    - add immediate unsigned, 227
    - Amdahl's law, 572
    - arithmetic, 242–245
    - assembly language for performance, 169
    - commercial binary compatibility importance, 170
    - defined, 49
    - GPUs, B-72–74, B-75

- Fallacies (*Continued*)
    - low utilization uses little power, 50
    - peak performance, 572
    - pipelining, 366
    - powerful instructions mean higher performance, 169
    - right shift, 242
  - False sharing, 480
  - Fast carry
    - with “infinite” hardware, A-626–627
    - with first level of abstraction, A-627–628
    - with second level of abstraction, A-628–634
  - Fast Fourier Transforms (FFT), B-53
  - Fault avoidance, 433
  - Fault forecasting, 433
  - Fault tolerance, 433
  - Fermi architecture, 539, 568
  - Field programmable devices (FPDs), A-666
  - Field programmable gate arrays (FPGAs), A-666
  - Fields
    - defined, 84
    - format, C-31
    - LEGv8, 84–86
    - names, 84
  - Files, register, 264, 269, A-638, A-642–644
  - Fine-grained multithreading, 530
  - Finite-state machines (FSMs), 472–477, A-655–660
    - control, C-8–22
    - controllers, 475
    - for multicycle control, C-9
    - for simple cache controller, 476–477
    - implementation, 474, A-658
    - Mealy, 475
    - Moore, 475
    - next-state function, 474, A-655
    - output function, A-655, A-657
    - state assignment, A-658
    - state register implementation, A-659
    - style of, 475
    - synchronous, A-655
    - SystemVerilog, OL5.12-7
    - traffic light example, A-656–658
  - Flash memory, 395
    - characteristics, 23
    - defined, 23
  - Flat address space, 493
  - Flip-flops
    - D flip-flops, A-639, A-641
    - defined, A-639
  - Floating point, 205–230, 232
    - assembly language, 221
    - backward step, OL3.12-4–3.12-5
    - binary to decimal conversion, 211
    - branch, 220
    - challenges, 246
    - diversity *versus* portability, OL3.12-3–3.12-4
    - division, 220
    - first dispute, OL3.12-2–3.12-3
    - form, 206
    - fused multiply add, 228
    - guard digits, 226–227
    - history, OL3.12-3
    - IEEE 754 standard, 207–211
    - intermediate calculations, 226
    - LEGv8 instruction frequency for, 248
    - LEGv8 instructions, 220–226
    - machine language, 221
    - operands, 221
    - overflow, 206
    - packed format, 232
    - precision, 243
    - procedure with two-dimensional matrices, 223–225
    - programs, compiling, 222–225
    - registers, 226
    - representation, 206–211
    - rounding, 226
    - sign and magnitude, 206
    - SSE2 architecture, 232, 233
    - subtraction, 220
    - underflow, 206
    - units, 227
    - in x86, 233
  - Floating vectors, OL3.12-3
  - Floating-point addition, 212–215
    - arithmetic unit block diagram, 216
    - binary, 213
    - illustrated, 214
    - instructions, 220
    - steps, 212
  - Floating-point arithmetic (GPUs), B-41–46
    - basic, B-42
    - double precision, B-45–46, B-74
    - performance, B-44
    - specialized, B-42–44
    - supported formats, B-42
    - texture operations, B-44
  - Floating-point instructions
    - desktop RISC, D-12
    - SPARC, D-31
  - Floating-point multiplication, 215–219
    - binary, 219
    - illustrated, 218
    - instructions, 220
    - significands, 215
    - steps, 215, 217
  - Flow-sensitive information, OL2.15-15
  - Flushing instructions, 329, 330
    - exceptions and, 340
  - For loops, 147, OL2.15-26
    - inner, OL2.15-24
    - SIMD and, OL6.15-2
  - Format fields, C-31
  - Fortran, OL2.22-7
  - Forwarding, 316–328
    - ALU before, 321
    - control, 320
    - datapath for hazard resolution, 323
    - defined, 289
    - functioning, 317
    - graphical representation, 290
    - illustrations, OL4.13-26
    - multiple results and, 292
    - multiplexors, 322
    - pipeline registers before, 321
    - with two instructions, 289–290
    - Verilog implementation, OL4.13-2–4.13-4
  - Fractions, 206, 207
  - Frame buffer, 18
  - Frame pointers, 106
  - Front end, OL2.15-3
  - Fully associative caches. *See also* Caches
    - block replacement strategies, 468–469
    - choice of, 422
    - defined, 417
    - memory block location, 417
    - misses, 420
  - Fully connected networks, 551
  - Fused-multiply-add (FMA) operation, 228, B-45–46
- ## G
- Galois/Counter Mode ( GCM )
    - encryption, 488
  - Game consoles, B-9



Gates, A-591, A-596  
 AND, A-600, C-7  
 delays, A-634–635  
 mapping ALU control function to, C-4–7  
 NAND, A-596  
 NOR, A-596, A-638  
 Gather-scatter, 527, 568  
 General Purpose GPUs (GPGPUs), B-5  
 General-purpose registers, 154  
 architectures, OL2.22-3  
 embedded RISCs, D-5  
 Generate  
 defined, A-628  
 example, A-632  
 super, A-629  
 Gigabyte, 6  
 Global common subexpression elimination, OL2.15-6  
 Global memory, B-21, B-39  
 Global miss rates, 430  
 Global optimization, OL2.15-5  
 code, OL2.15-7  
 implementing, OL2.15-8–2.15-11  
 Global pointers, 106  
 GPU computing. *See also* Graphics processing units (GPUs)  
 defined, B-5  
 visual applications, B-6–7  
 GPU system architectures, B-7–12  
 graphics logical pipeline, B-10  
 heterogeneous, B-7–9  
 implications for, B-24  
 interfaces and drivers, B-9  
 unified, B-10–12  
 Graph coloring, OL2.15-12  
 Graphics displays  
 computer hardware support, 18  
 LCD, 18  
 Graphics logical pipeline, B-10  
 Graphics processing units (GPUs), 538–543. *See also* GPU computing  
 as accelerators, 538  
 attribute interpolation, B-43–44  
 defined, 46, 522, B-3  
 evolution, B-5  
 fallacies and pitfalls, B-72–75  
 floating-point arithmetic, B-17, B-41–46, B-74  
 GeForce 8-series generation, B-5  
 general computation, B-73–74

General Purpose (GPGPUs), B-5  
 graphics mode, B-6  
 graphics trends, B-4  
 history, B-3–4  
 logical graphics pipeline, B-13–14  
 mapping applications to, B-55–72  
 memory, 538  
 multilevel caches and, 538  
 N-body applications, B-65–72  
 NVIDIA architecture, 539–541  
 parallel memory system, B-36–41  
 parallelism, 539, B-76  
 performance doubling, B-4  
 perspective, 543–545  
 programming, B-12–24  
 programming interfaces to, B-17  
 real-time graphics, B-13  
 summary, B-76  
 Graphics shader programs, B-14–15  
 Gresham's Law, 248, OL3.12-2  
 Grid computing, 549  
 Grids, B-19  
 GTX 280, 564–569  
 Guard digits  
 defined, 226  
 rounding with, 227

## H

Half precision, B-42  
 Halfwords, 114  
 Hamming, Richard, 434  
 Hamming distance, 434  
 Hamming Error Correction Code (ECC), 434–435  
 calculating, 434–435  
 Hard disks  
 access times, 23  
 defined, 23  
 Hardware  
 as hierarchical layer, 13  
 language of, 14–16  
 operations, 63–67  
 supporting procedures in, 100–110  
 synthesis, A-609  
 translating microprograms to, C-28–32  
 virtualizable, 440  
 Hardware description languages. *See also* Verilog  
 defined, A-608  
 using, A-608–614  
 VHDL, A-608–609

Hardware multithreading, 530–533  
 coarse-grained, 530  
 options, 531  
 simultaneous, 531  
 Hardware-based speculation, 352  
 Harvard architecture, OL1.12-4  
 Hazard detection units, 324  
 functions, 324  
 pipeline connections for, 327  
 Hazards. *See also* Pipelining  
 control, 292–293, 328–336  
 data, 289, 316–328  
 forwarding and, 323  
 structural, 288–289, 305  
 Heap  
 allocating space on, 107–110  
 defined, 107  
 Heterogeneous systems, B-4–5  
 architecture, B-7–9  
 defined, B-3  
 Hexadecimal numbers, 83  
 binary number conversion to, 83, 84  
 Hierarchy of memories, 12  
 High-level languages, 14–16  
 benefits, 16  
 computer architectures, OL2.22-5  
 importance, 16  
 High-level optimizations, OL2.15-4–2.15-5  
 Hit rate, 390  
 Hit time  
 cache performance and, 415–416  
 defined, 390  
 Hit under miss, 483  
 Hold time, A-642  
 Horizontal microcode, C-32  
 Hot-swapping, OL5.11-7  
 Human genome project, 4

## I

I/O, OL6.9-2, OL6.9-3  
 on system performance, OL5.11-2  
 I/O benchmarks. *See* Benchmarks  
 IBM 360/85, OL5.17-7  
 IBM 701, OL1.12-5  
 IBM 7030, OL4.16-2  
 IBM ALOG, OL3.12-7  
 IBM Blue Gene, OL6.15-9–6.15-10  
 IBM Personal Computer, OL1.12-7, OL2.22-6

- IBM System/360 computers, OL1.12-6, OL3.12-6, OL4.16-2
- IBM z/VM, OL5.17-8
- ID stage
  - branch execution in, 330, 331
  - load instructions, 303
  - store instruction in, 302
- IEEE 754 floating-point standard, 207–211, 208, OL3.12-8–3.12-10. *See also* Floating point
  - first chips, OL3.12-8–3.12-9
  - in GPU arithmetic, B-42–43
  - implementation, OL3.12-10
  - rounding modes, 227
  - today, OL3.12-10
- If statements, 118
- I-format, 87
- If-then-else, 94
- Immediate addressing, 120
- Immediate instructions, 73
- Imprecise interrupts, 342, OL4.16-4
- Index-out-of-bounds check, 98
- Induction variable elimination, OL2.15-7
- Inheritance, OL2.15-15
- In-order commit, 351
- Input devices, 16
- Inputs, 273
- Instances, OL2.15-15
- Instruction count, 36, 38
- Instruction decode/register file read stage
  - control line, 311–312
  - load instruction, 300
  - store instruction, 305
- Instruction execution illustrations, OL4.13-16–4.13-17
  - clock cycle 9, OL4.13-24
  - clock cycles 1 and 2, OL4.13-21
  - clock cycles 3 and 4, OL4.13-22
  - clock cycles 5 and 6, OL4.13-23
  - clock cycles 7 and 8, OL4.13-24
  - examples, OL4.13-20–4.13-25
  - forwarding, OL4.13-26–4.13-31
  - no hazard, OL4.13-17
  - pipelines with stalls and forwarding, OL4.13-26, OL4.13-20
- Instruction fetch stage
  - control line, 312
  - load instruction, 300
  - store instruction, 305
- Instruction formats, 161
  - ARMv7, 151
  - defined, 83
  - desktop/server RISC architectures, D-7
  - embedded RISC architectures, D-8
  - I-type, 85
  - LEGv8, 151
  - MIPS, 151
  - R-type, 85, 273
  - x86, 161
- Instruction latency, 367
- Instruction mix, 39, OL1.12-10
- Instruction set architecture
  - ARM, 152–154
  - branch address calculation, 266
  - defined, 22, 52
  - history, 173–174
  - maintaining, 52
  - protection and, 441
  - thread, B-31–34
  - virtual machine support, 440–441
- Instruction sets, B-49
  - ARMv8, 171
  - design for pipelining, 228
  - LEGv8, 247
  - MIPS-32, 151
  - x86 growth, 170
- Instruction-level parallelism (ILP), 365. *See also* Parallelism
  - compiler exploitation, OL4.16-5–4.16-6
  - defined, 43, 344
  - exploitation, increasing, 354
  - and matrix multiply, 363–365
- Instructions, 60–174, D-25–27, D-40–42. *See also* Arithmetic instructions; MIPS; Operands
  - add immediate, 73
  - addition, 190
  - Alpha, D-27–29
  - arithmetic-logical, 263
  - ARM, 152–154, D-36–37
  - assembly, 66
  - basic block, 96
  - cache-aware, 496
  - conditional branch, 93, 94
  - conditional move, 334
  - core, 246
  - data transfer, 68
  - decision-making, 93–99
  - defined, 14, 62
  - desktop RISC conventions, D-12
  - as electronic signals, 82
  - embedded RISC conventions, D-15
  - encoding, 86
  - fetching, 265
  - fields, 83
  - floating-point (x86), 232, 233
  - floating-point, 220–221
  - flushing, 329, 330, 340
  - immediate, 73
  - introduction to, 62–63
  - jump
  - left-to-right flow, 298
  - load, 69
  - logical operations, 90–93
  - M32R, D-40
  - memory access, B-33–34
  - memory-reference, 257
  - multiplication, 197
  - nop, 325–326
  - PA-RISC, D-34–36
  - performance, 35–36
  - pipeline sequence, 325
  - PowerPC, D-12–13, D-32–34
  - PTX, B-31, B-32
  - representation in computer, 82–89
  - restartable, 462
  - resuming,
  - R-type, 263, 268
  - SPARC, D-29–32
  - store, 72
  - store exclusive register (STXR), 126
  - subtraction, 190
  - SuperH, D-39–40
  - thread, B-30–31
  - Thumb, D-38
  - vector, 524
  - as words, 62
  - x86, 154–159
- Instructions per clock cycle (IPC), 343
- Integrated circuits (ICs), 19. *See also* specific chips
  - cost, 27
  - defined, 25
  - manufacturing process, 26
  - very large-scale (VLSIs), 25
- Intel Core i7, 46–49, 256, 517, 564–569
  - address translation for, 483
  - architectural registers, 358
  - caches in, 484
  - memory hierarchies of, 482–488
  - microarchitecture, 358
  - performance of, 485–486
  - SPEC CPU benchmark, 46–48
  - SPEC power benchmark, 48–49
  - TLB hardware for, 483

Intel Core i7 920, 358–360  
 microarchitecture, 358

Intel Core i7 960  
 benchmarking and rooflines of, 564–569

Intel Core i7 Pipelines, 354, 358–360  
 memory components, 359  
 performance, 361–362  
 program performance, 362  
 specification, 356

Intel IA-64 architecture, OL2.22-3

Intel Paragon, OL6.15-8

Intel Threading Building Blocks, B-60

Intel x86 microprocessors  
 clock rate and power for, 40

Interference graphs, OL2.15-12

Interleaving, 412

Interprocedural analysis, OL2.15-14

Interrupt enable, 461

Interrupt-driven I/O, OL6.9-4

Interrupts  
 defined, 207, 336  
 event types and, 336  
 exceptions *versus*, 336  
 imprecise, 342, OL4.16-4  
 precise, 342  
 vectored, 337

Intrinsity FastMATH processor, 409–412  
 caches, 410  
 data miss rates, 411, 421  
 read processing, 456  
 TLB, 454–457  
 write-through processing, 456

Inverted page tables, 451

Issue packets, 345

## J

Java  
 bytecode, 136  
 bytecode architecture, OL2.15-17  
 characters in, 113–115  
 compiling in, OL2.15-19–2.15-20  
 goals, 136  
 interpreting, 136, 150, OL2.15-15  
 keywords, OL2.15-21  
 method invocation in, OL2.15-21  
 pointers, OL2.15-26  
 primitive types, OL2.15-26  
 programs, starting, 136–137  
 reference types, OL2.15-26  
 sort algorithms, 146

strings in, 113–115  
 translation hierarchy, 136  
 while loop compilation in, OL2.15-18–2.15-19

Java Virtual Machine (JVM), 150, OL2.15-16

Jump instructions, 254, D-26  
 branch instruction *versus*, 270  
 control and datapath for, 271  
 implementing, 270  
 instruction format, 270

Just In Time (JIT) compilers, 137, 576

## K

Karnaugh maps, A-606

Kernel mode, 459

Kernels  
 CUDA, B-19, B-24  
 defined, B-19

Kilobyte, 6

## L

LAPACK, 243

Large-scale multiprocessors, OL6.15-7, OL6.15-9–6.15-10

Latches  
 D latch, A-639, A-640  
 defined, A-639

Latency  
 instruction, 367  
 memory, B-74–75  
 pipeline, 297  
 use, 346

LDUR (load register), 64

LDURB (load byte), 64

LDURH (load half), 64

LDURSW (load signed word), 64

LDXR (load exclusive register), 64, 122

Leaf procedures. *See also* Procedures  
 defined, 104  
 example, 113

Least recently used (LRU)  
 as block replacement strategy, 468–469  
 defined, 423  
 pages, 448

Least significant bits, A-620  
 defined, 75

SPARC, D-31

Left-to-right instruction flow, 298–299

LEGv8, 62, 64, 86

architecture, 204

arithmetic core, 246

arithmetic instructions, 63

arithmetic/logical instructions not in, D-21, D-23

assembly instruction, mapping, 82–83

common extensions to, D-20–25

compiling C assignment statements into, 66

compiling complex C assignment into, 66

control instructions not in, D-21

control registers, 461

control unit, C-10

data transfer instructions not in, D-20, D-22

divide in, 203

exceptions in, 337–338

fields, 84–85

floating-point instructions not in, D-22

floating-point instructions, 220–221

instruction classes, 173

instruction encoding, 86, 122

instruction formats, 124, 151

instruction set, 62, 171, 246, 247, 256–260, D-9–10

machine language, 88

memory addresses, 71

memory allocation for program and data, 108

multiply in, 197

operands, 64

Pseudo, 246

register conventions, 109

static multiple issue with, 345–347

Level-sensitive clocking, A-662, A-663–664  
 defined, A-662  
 two-phase, A-663

Link, OL6.9-2

Linkers, 131–134  
 defined, 131  
 executable files, 131  
 steps, 131  
 using, 131–134

Linking object files, 132–134

Linpack, 554, OL3.12-4

Liquid crystal displays (LCDs), 18

LISP, SPARC support, D-30

Live range, OL2.15-11

Livermore Loops, OL1.12-11

Load balancing, 521–522  
 Load byte, 167  
 Load halfword, 167  
 Load instructions. *See also* Store instructions  
   access, B-41  
   base register, 274  
   compiling with, 71–72  
   datapath in operation for, 279  
   defined, 69  
   EX stage, 303  
   halfword unsigned, 114  
   ID stage, 302  
   IF stage, 302  
   load byte unsigned, 79  
   load half, 114  
   MEM stage, 304  
   move wide with keep, 115  
   move wide with zeros, 115  
   pipelined datapath in, 307  
   signed, 79  
   unit for implementing, 267  
   unsigned, 79  
   WB stage, 304  
 Load register, 69, 72  
 Loaders, 134  
 Load-store architectures, OL2.22-3  
 Load-use data hazard, 290, 329  
 Load-use stalls, 329  
 Local area networks (LANs), 24. *See also* Networks  
 Local memory, B-21, B-40  
 Local miss rates, 430  
 Local optimization, OL2.15-5. *See also* Optimization  
   implementing, OL2.15-8  
 Locality  
   principle, 388  
   spatial, 388, 391  
   temporal, 388, 391  
 Lock synchronization, 125  
 Locks, 534  
 Logic  
   address select, C-24, C-25  
   ALU control, C-6  
   combinational, 262, A-593, A-597–608  
   components, 261  
   control unit equations, C-11  
   design, 260–263, B-1–79  
   equations, A-595  
   minimization, A-606  
   programmable array (PAL), A-666

  sequential, A-593, A-644–646  
   two-level, A-599–602  
 Logical operations, 90–93  
   AND, 91  
   ARM, 154  
   desktop RISC, D-11  
   embedded RISC, D-14  
   EOR, 92  
   NOT, 91  
   OR, 91  
   shifts, 90  
 Long instruction word (LIW), OL4.16-5  
 Lookup tables (LUTs), A-667  
 Loop unrolling  
   defined, 348, OL2.15-4  
   for multiple-issue pipelines, 348  
   register renaming and, 348  
 Loops, 95–96  
   conditional branches in, 118  
   for, 147  
   prediction and, 333–334  
   test, 147, 148  
   while, compiling, 95–96

## M

M32R, D-15, D-40  
 Machine code, 83  
 Machine instructions, 83  
 Machine language, 15  
   branch offset in, 119  
   decoding, 121–124  
   defined, 14, 83  
   floating-point, 221  
   illustrated, 15  
   LEGv8, 88  
   SRAM, 21  
   translating MIPS assembly language into, 86  
 Main memory, 442. *See also* Memory  
   defined, 23  
   page tables, 451  
   physical addresses, 442  
 Mapping applications, B-55–72  
 Mark computers, OL1.12-14  
 Matrix multiply, 238–242, 569–571  
 Mealy machine, 475, A-656, A-659, A-660  
 Mean time to failure (MTTF), 432  
   improving, 433  
   *versus* AFR of disks, 433–434  
 Media Access Control (MAC) address, OL6.9-7  
 Megabyte, 6  
 Memory  
   addresses, 79  
   affinity, 562  
   atomic, B-21  
   bandwidth, 394–395, 411  
   cache, 21, 397–412, 412–431  
   CAM, 422  
   constant, B-40  
   control, C-26  
   defined, 19  
   DRAM, 19, 393–394, A-651–653  
   flash, 23  
   global, B-21, B-39  
   GPU, 538  
   instructions, datapath for, 267  
   local, B-21, B-40  
   main, 23  
   nonvolatile, 22  
   operands, 68–69  
   parallel system, B-36–41  
   read-only (ROM), A-602–604  
   SDRAM, 393–394  
   secondary, 23  
   shared, B-21, B-39–40  
   spaces, B-39  
   SRAM, A-646–650  
   stalls, 414  
   technologies for building, 24–28  
   texture, B-40  
   virtual, 441–465  
   volatile, 22  
 Memory access instructions, B-33–34  
 Memory access stage  
   control line, 313  
   load instruction, 303  
   store instruction, 303  
 Memory bandwidth, 565, 573  
 Memory consistency model, 481  
 Memory elements, A-638–646  
   clocked, A-639  
   D flip-flop, A-639, A-641  
   D latch, A-640  
   DRAMs, A-651–655  
   flip-flop, A-639  
   hold time, A-642  
   latch, A-639  
   setup time, A-641, A-642  
   SRAMs, A-646–650  
   unclocked, A-639  
 Memory hierarchies, 559  
   of ARM cortex-A8, 482–488

- block (or line), 390
- cache performance, 412–431
- caches, 397–431
- common framework, 465–472
- defined, 389
- design challenges, 472
- development, OL5.17-6–5.17-8
- exploiting, 386–513
- of Intel Core i7, 482–488
- level pairs, 390
- multiple levels, 389
- overall operation of, 457–458
- parallelism and, 477–481, OL5.11-2
- pitfalls, 491–495
- program execution time and, 431
- quantitative design parameters, 466
- redundant arrays and inexpensive disks, 481
- reliance on, 390
- structure, 389
- structure diagram, 392
- variance, 431
- virtual memory, 441–465
- Memory rank, 395
- Memory technologies, 392–397
  - disk memory, 395–397
  - DRAM technology, 392, 393–395
  - flash memory, 395
  - SRAM technology, 392, 393
- Memory-mapped I/O, OL6.9-3
- Memory-stall clock cycles, 413
- Message passing
  - defined, 543
  - multiprocessors, 543–548
- Metastability, A-664
- Methods
  - defined, OL2.15-5
  - invoking in Java, OL2.15-20–2.15-21
- Microarchitectures, 358
  - Intel Core i7 920, 358
- Microcode
  - assembler, C-30
  - control unit as, C-28
  - defined, C-27
  - dispatch ROMs, C-30–31
  - horizontal, C-32
  - vertical, C-32
- Microinstructions, C-31
- Microprocessors
  - design shift, 517
  - multicore, 8, 43, 517
- Microprograms
  - as abstract control representation, C-30
  - field translation, C-29
  - translating to hardware, C-28–32
- Migration, 479
- Million instructions per second (MIPS), 51
- Minterms
  - defined, A-600, C-20
  - in PLA implementation, C-20
- MIP-map, B-44
- MIPS and ARMv8
  - common features between, 152
- MIPS-16
  - 16-bit instruction set, D-41–42
  - immediate fields, D-41
  - instructions, D-40–42
  - MIPS core instruction changes, D-42
  - PC-relative addressing, D-41
- MIPS-32 instruction set, 151
- MIPS-64 instructions, 151, D-25–27
  - conditional procedure call instructions, D-27
  - constant shift amount, D-25
  - jump/call not PC-relative, D-26
  - move to/from control registers, D-26
  - nonaligned data transfers, D-25
  - NOR, D-25
  - parallel single precision floating-point operations, D-27
  - reciprocal and reciprocal square root, D-27
  - SYSCALL, D-25
  - TLB instructions, D-26–27
- Mirroring, OL5.11-5
- Miss penalty
  - defined, 390
  - determination, 405–406
  - multilevel caches, reducing, 424
- Miss rates
  - block size *versus*, 406
  - data cache, 467
  - defined, 390
  - global, 430
  - improvement, 405–406
  - Intrinsity FastMATH processor, 411
  - local, 430
  - miss sources, 471
  - split cache, 411
- Miss under miss, 483
- MMX (MultiMedia eXtension), 232
- Moore machines, 475, A-656, A-659, A-660
- Moore's law, 11, 393, 538, OL6.9-2, B-72–73
- Most significant bit
  - 1-bit ALU for, A-621
  - defined, 75
- move (Move), 144
- MOVK (move wide with keep), 64, 115
- MOVZ (move wide with zero), 64, 115
- MS-DOS, OL5.17-11
- Multicore, 533–537
- Multicore multiprocessors, 8, 43
  - defined, 8, 517
- MULTICS (Multiplexed Information and Computing Service), OL5.17-9–5.17-10
- Multilevel caches. *See also* Caches
  - complications, 430
  - defined, 412, 430
  - miss penalty, reducing, 424
  - performance of, 424
  - summary, 431–432
- Multimedia extensions
  - desktop/server RISCs, D-16–18
  - as SIMD extensions to instruction sets, OL6.15-4
  - vector *versus*, 525–526
- Multiple dimension arrays, 226
- Multiple instruction multiple data (MIMD), 574
  - defined, 523, 524
  - first multiprocessor, OL6.15-14
- Multiple instruction single data (MISD), 523
- Multiple issue, 343–350
  - code scheduling, 347–348
  - dynamic, 343, 349–350
  - issue packets, 345
  - loop unrolling and, 348
  - processors, 343, 344
  - static, 343, 345–349
  - throughput and, 353
- Multiple processors, 569–571
- Multiple-clock-cycle pipeline diagrams, 308
  - five instructions, 309
  - illustrated, 309
- Multiplexors, A-598
  - controls, 473
  - in datapath, 275
  - defined, 258
  - forwarding, control values, 322
  - selector control, 271
  - two-input, A-598

- Multiplicand, 192
  - Multiplication, 191–197. *See also*
    - Arithmetic
    - fast, hardware, 196
    - faster, 196–197
    - first algorithm, 194
    - floating-point, 215–217
    - hardware, 192–194
    - instructions, 197
    - in MIPS, 197
    - multiplicand, 197
    - multiplier, 197
    - operands, 197
    - product, 197
    - sequential version, 192–194
    - signed, 196
  - Multiplier, 192
  - Multiply algorithm, 195
  - Multiply-add (MAD), B-42
  - Multiprocessors
    - benchmarks, 554–556
    - bus-based coherent, OL6.15-7
    - defined, 516
    - historical perspective, 577
    - large-scale, OL6.15-7–6.15-8, OL6.15-9–6.15-10
    - message-passing, 545–550
    - multithreaded architecture, B-26–27, B-35–36
    - organization, 515, 545
    - for performance, 573
    - shared memory, 517, 533–537
    - software, 517
    - TFLOPS, OL6.15-6
    - UMA, 534
  - Multistage networks, 551
  - Multithreaded multiprocessor
    - architecture, B-25–36
    - conclusion, B-36
    - ISA, B-31–34
    - massive multithreading, B-25–26
    - multiprocessor, B-26–27
    - multiprocessor comparison, B-35–36
    - SIMT, B-27–30
    - special function units (SFUs), B-35
    - streaming processor (SP), B-34
    - thread instructions, B-30–31
    - threads/thread blocks management, B-30
  - Multithreading, B-25–26
    - coarse-grained, 530
    - defined, 522
    - fine-grained, 530
    - hardware, 530–533
    - simultaneous (SMT), 531–533
  - Must-information, OL2.15-5
  - Mutual exclusion, 125
- ## N
- Name dependence, 348
  - NAND gates, A-596
  - NAS (NASA Advanced Supercomputing), 556
  - N-body
    - all-pairs algorithm, B-65
    - GPU simulation, B-71
    - mathematics, B-65–67
    - multiple threads per body, B-68–69
    - optimization, B-67
    - performance comparison, B-69–70
    - results, B-70–72
    - shared memory use, B-67–68
  - Negation shortcut, 79
  - Nested procedures, 104–105
    - compiling recursive procedure showing, 104–105
  - NetFPGA 10-Gigabit Ethernet card, OL6.9-2, OL6.9-3
  - Network of Workstations, OL6.15-8–6.15-9
  - Network topologies, 550–553
    - implementing, 552
    - multistage, 553
  - Networking, OL6.9-4
    - operating system in, OL6.9-4–6.9-5
    - performance improvement, OL6.9-7–6.9-10
  - Networks, 23–24
    - advantages, 23
    - bandwidth, 549
    - crossbar, 551
    - fully connected, 551
    - local area (LANs), 24
    - multistage, 551
    - wide area (WANs), 24
  - Newton's iteration, 226
  - Next state
    - nonsequential, C-24
    - sequential, C-23
  - Next-state function, 474, A-655
    - defined, 474
    - implementing, with sequencer, C-22–28
  - Next-state outputs, C-10, C-12–13
    - example, C-12–13
    - implementation, C-12
    - logic equations, C-12–13
    - truth tables, C-15
  - No Redundancy (RAID 0), OL5.11-4
  - No write allocation, 408
  - Nonblocking assignment, A-612
  - Nonblocking caches, 355, 483
  - Nonuniform memory access (NUMA), 534
  - Nonvolatile memory, 22
  - Nops, 326
  - NOR gates, A-596
    - cross-coupled, A-638
    - D latch implemented with, A-640
  - NOR operation, D-25
  - NOT operation, 91, A-594
  - Not-A-Number (NaN), 235–236
  - Numbers
    - binary, 75
    - computer *versus* real-world, 229
    - decimal, 75, 78
    - denormalized, 230
    - hexadecimal, 84
    - signed, 75–82
    - unsigned, 75–82
  - NVIDIA GeForce 8800, B-46–55
    - all-pairs N-body algorithm, B-71
    - dense linear algebra computations, B-51–53
    - FFT performance, B-53
    - instruction set, B-49
    - performance, B-51
    - rasterization, B-50
    - ROP, B-50–51
    - scalability, B-51
    - sorting performance, B-54–55
    - special function approximation statistics, B-43
    - special function unit (SFU), B-50
    - streaming multiprocessor (SM), B-48–49
    - streaming processor, B-49–50
    - streaming processor array (SPA), B-46
    - texture/processor cluster (TPC), B-47–48
  - NVIDIA GPU architecture, 539–541
  - NVIDIA GTX 280, 565, 566
  - NVIDIA Tesla GPU, 564–569
- ## O
- Object files, 132
    - debugging information, 131
    - header, 130



linking, 132–134  
 relocation information, 130  
 static data segment, 130  
 symbol table, 130  
 text segment, 130

Object-oriented languages. *See also* Java  
 brief history, OL2.22-8  
 defined, 150, OL2.15-5

One's complement, 82, A-617

Opcodes  
 control line setting and, 276  
 defined, 84, 274

OpenGL, B-13

OpenMP (Open MultiProcessing), 536, 556

Operands, 67–72. *See also* Instructions  
 32-bit immediate, 115–116  
 adding, 189  
 arithmetic instructions, 67  
 compiling assignment when in memory, 69  
 constant, 73–74  
 division, 197  
 floating-point, 221  
 LEV8, 64  
 memory, 68–69  
 multiplication, 191

Operating systems  
 brief history, OL5.17-9–5.17-12  
 defined, 13  
 encapsulation, 22  
 in networking, OL6.9-4–6.9-5

Operations  
 atomic, implementing, 126  
 hardware, 63–67  
 logical, 90–93  
 x86 integer, 157, 158

Optimization  
 class explanation, OL2.15-14  
 compiler, 146  
 control implementation,  
   C-27–28  
 global, OL2.15-5  
 high-level, OL2.15-4–2.15-5  
 local, OL2.15-5, OL2.15-8  
 manual, 150

OR operation, 91, A-594

Out-of-order execution  
 defined, 351  
 performance complexity, 430  
 processors, 355

Output devices, 16

Overflow

defined, 76, 206  
 detection, 190  
 exceptions, 339  
 floating-point, 207  
 occurrence, 77  
 saturation and, 191  
 subtraction, 189

## P

P+Q redundancy (RAID 6), OL5.11-7

Packed floating-point format, 232

Page faults, 448. *See also* Virtual memory  
 for data access, 463  
 defined, 442  
 handling, 443, 461–464  
 virtual address causing, 457, 458

Page tables, 468  
 defined, 446  
 illustrated, 449  
 indexing, 446  
 inverted, 451  
 levels, 451  
 main memory, 451  
 register, 446  
 storage reduction techniques, 451  
 updating, 446  
 VMM, 463

Pages. *See also* Virtual memory  
 defined, 442  
 dirty, 452  
 finding, 446–447  
 LRU, 448  
 offset, 443  
 physical number, 443  
 placing, 432–434  
 size, 444  
 virtual number, 443

Parallel bus, OL6.9-3

Parallel execution, 125

Parallel memory system, B-36–41.  
*See also* Graphics processing units (GPUs)  
 caches, B-38  
 constant memory, B-40  
 DRAM considerations, B-37–38  
 global memory, B-39  
 load/store access, B-41  
 local memory, B-40  
 memory spaces, B-39  
 MMU, B-38–39  
 ROP, B-41  
 shared memory, B-39–40

surfaces, B-41  
 texture memory, B-40

Parallel processing programs, 518–523  
 creation difficulty, 518–523  
 defined, 516  
 for message passing, 533  
 great debates in, OL6.15-5  
 for shared address space, 533–534  
 use of, 573

Parallel reduction, B-62

Parallel scan, B-60–63  
 CUDA template, B-61  
 inclusive, B-60  
 tree-based, B-62

Parallel software, 517

Parallelism, 12, 43, 342–355  
 and computers arithmetic, 230–232  
 data-level, 246, 524  
 debates, OL6.15-5–6.15-7  
 GPUs and, 538, B-76  
 instruction-level, 43, 342, 354  
 memory hierarchies and, 477–481,  
   OL5.11-2  
 multicore and, 533  
 multiple issue, 342–349  
 multithreading and, 531  
 performance benefits, 44  
 process-level, 516  
 redundant arrays and inexpensive  
   disks, 481  
 subword, D-17  
 task, B-24  
 task-level, 516  
 thread, B-22

Paravirtualization, 495

PA-RISC, D-14, D-17  
 branch vectored, D-35  
 conditional branches, D-34, D-35  
 debug instructions, D-36  
 decimal operations, D-35  
 extract and deposit, D-35  
 instructions, D-34–36  
 load and clear instructions, D-36  
 multiply/add and multiply/subtract,  
   D-36  
 nullification, D-34  
 nullifying branch option, D-25  
 store bytes short, D-36  
 synthesized multiply and divide,  
   D-34–35

Parity, OL5.11-5  
 bits, 435  
 code, 434, A-653

- PARSEC (Princeton Application Repository for Shared Memory Computers), 556
- Pass transistor, A-651
- PCI-Express (PCIe), 553, B-8, OL6.9-2
- PC-relative addressing, 118, 120
- Peak floating-point performance, 558
- Pentium bug morality play, 244
- Performance, 28–40
  - assessing, 28
  - classic CPU equation, 36–40
  - components, 38
  - CPU, 33–35
  - defining, 29–32
  - equation, using, 36
  - improving, 34–35
  - instruction, 35–36
  - measuring, 33–35, OL1.12-10
  - program, 39–40
  - ratio, 31
  - relative, 31–32
  - response time, 30–31
  - sorting, B-54–55
  - throughput, 30–31
  - time measurement, 32
- Personal computers (PCs), 7
  - defined, 5
- Personal mobile device (PMD)
  - defined, 7
- Petabyte, 6
- Physical addresses, 442
  - mapping to, 442–443
  - space, 533, 535
- Physically addressed caches, 458
- Pipeline registers
  - before forwarding, 320
  - dependences, 319
  - forwarding unit selection, 323
- Pipeline stalls, 291
  - avoiding with code reordering, 291
  - data hazards and, 324–328
  - insertion, 326
  - load-use, 329
  - as solution to control hazards, 293
- Pipelined branches, 331
- Pipelined control, 311–315. *See also* Control
  - control lines, 311, 312
  - overview illustration, 327
  - specifying, 312
- Pipelined datapaths, 297–315
  - with connected control signals, 315
  - with control signals, 311–315
  - corrected, 307
  - illustrated, 300
  - in load instruction stages, 307
- Pipelined dependencies, 317
- Pipelines
  - branch instruction impact, 329
  - effectiveness, improving, OL4.16-4-4.16-5
  - execute and address calculation stage, 301, 303
  - five-stage, 285, 301, 309
  - graphic representation, 290, 307–311
  - instruction decode and register file
    - read stage, 300, 303
  - instruction fetch stage, 301, 303
  - instructions sequence, 325
  - latency, 297
  - memory access stage, 301, 303
  - multiple-clock-cycle diagrams, 308
  - performance bottlenecks, 353
  - single-clock-cycle diagrams, 308
  - stages, 285
  - static two-issue, 345
  - write-back stage, 301, 305
- Pipelining, 12, 283–297
  - advanced, 354–355
  - benefits, 283
  - control hazards, 292–293
  - data hazards, 289
  - exceptions and, 338–342
  - execution time and, 297
  - fallacies, 366–367
  - hazards, 288
  - instruction set design for, 288
  - laundry analogy, 284
  - overview, 283–297
  - paradox, 285
  - performance improvement, 288
  - pitfall, 366–367
  - simultaneous executing instructions, 297
  - speed-up formula, 285
  - structural hazards, 288, 305
  - summary, 296
  - throughput and, 297
- Pitfalls. *See also* Fallacies
  - address space extension, 493
  - arithmetic, 242–245
  - associativity, 492
  - defined, 49
  - GPUs, B-74–75
  - ignoring memory system behavior, 491
  - memory hierarchies, 491–495
  - out-of-order processor evaluation, 492
  - performance equation subset, 50–51
  - pipelining, 366–367
  - pointer to automatic variables, 171
  - sequential word addresses, 171
  - simulating cache, 491
  - software development with
    - multiprocessors, 570
    - VMM implementation, 495
- Pixel shader example, B-15–17
- Pixels, 18
- Pointers
  - arrays *versus*, 146–150
  - frame, 106
  - global, 106
  - incrementing, 148
  - Java, OL2.15-26
  - stack, 101, 105
- Polling, OL6.9-8
- Pop, 101
- Power
  - clock rate and, 40
  - critical nature of, 53
  - efficiency, 354–355
  - relative, 41
- PowerPC
  - algebraic right shift, D-33
  - branch registers, D-32–33
  - condition codes, D-12
  - instructions, D-12–13
  - instructions unique to, D-31–33
  - load multiple/store multiple, D-33
  - logical shifted immediate, D-33
  - rotate with mask, D-33
- Precise interrupts, 342
- Prediction, 12
  - 2-bit scheme, 333
  - accuracy, 333
  - dynamic branch, 331–333
  - loops and, 333–334
  - steady-state, 333
- Prefetching, 496, 560
- Primitive types, OL2.15-26
- Procedure calls
  - preservation across, 106
- Procedures, 100–110
  - compiling, 102
  - compiling, showing nested procedure
    - linking, 102–104
  - execution steps, 100

frames, 106  
 leaf, 104  
 nested, 104–106  
 recursive, 108  
 for setting arrays to zero, 147  
 sort, 140–145  
 strcpy, 112–113  
 string copy, 112–113  
 swap, 138

Process identifiers, 460

Process-level parallelism, 516

Processors, 254–368  
 as cores, 43  
 control, 19  
 datapath, 19  
 defined, 17, 19  
 dynamic multiple-issue, 343  
 multiple-issue, 343  
 out-of-order execution, 355, 430  
 performance growth, 44  
 ROP, B-12, B-41  
 speculation, 344–345  
 static multiple-issue, 343, 345–349  
 streaming, B-34  
 superscalar, 349, 531–532, OL4.16-5  
 technologies for building, 24–28  
 two-issue, 346  
 vector, 523–524  
 VLIW, 345

Product, 192

Product of sums, A-599

Program counters (PCs), 263  
 changing with conditional branch, 334  
 defined, 101, 263  
 exception, 459, 461  
 incrementing, 263, 265  
 instruction updates, 300

Program performance  
 elements affecting, 39  
 understanding, 9

Programmable array logic (PAL), A-666

Programmable logic arrays (PLAs)  
 component dots illustration, A-604  
 control function implementation, C-7, C-20–21  
 defined, A-600  
 example, A-601–602  
 illustrated, A-601  
 ROMs and, A-603–604  
 size, C-20  
 truth table implementation, A-601

Programmable logic devices (PLDs), A-666

Programmable ROMs (PROMs), A-602

Programming languages. *See also* specific languages  
 brief history of, OL2.22-7–2.22-8  
 object-oriented, 150  
 variables, 67

Programs  
 assembly language, 129  
 Java, starting, 136–137  
 parallel processing, 516–523  
 starting, 128–137  
 translating, 128–137

Propagate  
 defined, A-628  
 example, A-632  
 super, A-629

Protected keywords, OL2.15-21

Protection  
 defined, 442  
 implementing, 459–460  
 mechanisms, OL5.17-9  
 VMs for, 438

Protection group, OL5.11-5

Pseudo MIPS  
 defined, 246  
 instruction set, 248

Pseudoinstructions  
 defined, 129  
 summary, 130

Pthreads (POSIX threads), 556

PTX instructions, B-31, B-32

Public keywords, OL2.15-21

Push  
 defined, 101  
 using, 104

## Q

Quad words, 158

Quicksort, 425, 426

Quotient, 198

## R

Race, A-661

Radix sort, 425, 426, B-63–65  
 CUDA code, B-64  
 implementation, B-63–65

RAID, *See* Redundant arrays of inexpensive disks (RAID)

RAM, 9

Raster operation (ROP) processors, B-12, B-41, B-50–51  
 fixed function, B-41

Raster refresh buffer, 18

Rasterization, B-50

Ray casting (RC), 568

Read-only memories (ROMs), A-602–604  
 control entries, C-16–17  
 control function encoding, C-18–19  
 dispatch, C-25  
 implementation, C-15–19  
 logic function encoding, A-603  
 overhead, C-18  
 PLAs and, A-603–604  
 programmable (PROM), A-602  
 total size, C-16

Read-stall cycles, 413

Read-write head, 395

Receive message routine, 545

Recursive procedures, 108. *See also* Procedures  
 clone invocation, 104

Reduced instruction set computer (RISC) architectures, D-2–45, OL2.22-5, OL4.16-4. *See also* Desktop and server RISCs; Embedded RISCs  
 group types, D-3–4  
 instruction set lineage, D-44

Reduction, 535

Redundant arrays of inexpensive disks (RAID), OL5.11-2–5.11-8  
 history, OL5.11-8  
 RAID 0, OL5.11-4  
 RAID 1, OL5.11-5  
 RAID 2, OL5.11-5  
 RAID 3, OL5.11-5  
 RAID 4, OL5.11-5–5.11-6  
 RAID 5, OL5.11-6–5.11-7  
 RAID 6, OL5.11-7  
 spread of, OL5.11-6  
 summary, OL5.11-7–5.11-8  
 use statistics, OL5.11-7

Reference bit, 450

References  
 absolute, 131  
 types, OL2.15-26

Register 31, 74, 102, 175

Register addressing, 120

Register allocation, OL2.15-11–2.15-13

Register files, A-638, A-642–644  
 defined, 264, A-638, A-642  
 in behavioral Verilog, A-645  
 single, 269  
 two read ports implementation, A-643  
 with two read ports/one write port,  
   A-643  
 write port implementation, A-644

Register-memory architecture, OL2.22-3

Registers, 156, 157–158  
 architectural, 336–342  
 base, 69  
 clock cycle time and, 67  
 compiling C assignment with, 68  
 defined, 67  
 destination, 85, 274  
 floating-point, 226  
 left half, 301  
 LEGv8 conventions, 108  
 mapping, 82  
 number specification, 264  
 page table, 446  
 pipeline, 319, 321, 323  
 primitives, 67  
 renaming, 348  
 right half, 301  
 spilling, 72  
 Status, 337  
 temporary, 68, 102  
 variables, 67

Relative performance, 31–32

Relative power, 41

Reliability, 432

Remainder  
 defined, 198

Reorder buffers, 355

Replication, 479

Requested word first, 406

Request-level parallelism, 548

Reservation stations  
 buffering operands in, 350  
 defined, 350

Response time, 30–31

Restartable instructions, 462

Return address, 100

Return from exception (ERET), 459

R-format, 274  
 ALU operations, 265  
 defined, 86

Ripple carry  
 adder, A-617  
 carry lookahead speed *versus*,  
   A-634–635

Roofline model, 558–559, 560, 561  
 with ceilings, 561  
 computational roofline, 559  
 illustrated, 557

Opteron generations, 558  
 with overlapping areas shaded, 563  
 peak floating-point performance, 562  
 peak memory performance, 562  
 with two kernels, 563

Rotational delay. *See* Rotational latency

Rotational latency, 397

Rounding, 226  
 accurate, 226  
 bits, 228  
 with guard digits, 227  
 IEEE 754 modes, 227

Row-major order, 225, 427

R-type instructions, 264  
 datapath for, 276  
 datapath in operation for, 278

## S

Saturation, 191

SCALAPAK, 244

Scaling  
 strong, 521  
 weak, 521

Scientific notation  
 adding numbers in, 213  
 defined, 205  
 for reals, 205

Search engines, 4

Secondary memory, 23

Sectors, 395

Secure Hash Algorithm ( SHA )  
 encryption, 488

Seek, 396

Segmentation, 445

Selector values, A-598

Semiconductors, 25

Send message routine, 545

Sensitivity list, A-612

Sequencers  
 explicit, C-32  
 implementing next-state function with,  
   C-22–28

Sequential logic, A-593

Servers, OL5. *See also* Desktop and server

RISCs  
 cost and capability, 5

Service accomplishment, 432

Service interruption, 432

Set instructions, 97

Set-associative caches, 417. *See also*  
   Caches  
     address portions, 421  
     block replacement strategies, 468  
     choice of, 467  
     four-way, 418, 421  
     memory-block location, 417  
     misses, 419–420  
     *n*-way, 417  
     two-way, 418

Setup time, A-641, A-642

Shaders  
 defined, B-14  
 floating-point arithmetic, B-14  
 graphics, B-14–15  
 pixel example, B-15–17

Shading languages, B-14

Shadowing, OL5.11-5

Shared memory. *See also* Memory  
 as low-latency memory, B-21  
 caching in, B-58–60  
 CUDA, B-58  
 N-body and, B-67–68  
 per-CTA, B-39  
 SRAM banks, B-40

Shared memory multiprocessors (SMP),  
 531–535  
 defined, 517, 531  
 single physical address space, 531  
 synchronization, 534

Shift amount, 84

Shift instructions, 90

Sign and magnitude, 206

Sign bit, 78

Sign extension, 266  
 defined, 78  
 shortcut, 80

Signals  
 asserted, 262, A-592  
 control, 262, 274–275  
 deasserted, 262, A-592

Signed division, 201–202

Signed multiplication, 196

Signed numbers, 75–82  
 sign and magnitude, 77  
 treating as unsigned, 98

Significands, 207  
 addition, 212  
 multiplication, 215

Silicon, 25  
 as key hardware technology, 53  
 crystal ingot, 26

- defined, 26
- wafers, 26
- Silicon crystal ingot, 26
- SIMD (Single Instruction Multiple Data), 522, 574
  - computers, OL6.15-2–6.15-4
  - data vector, B-35
  - extensions, OL6.15-4
  - for loops and, OL6.15-3
  - massively parallel multiprocessors, OL6.15-2
  - small-scale, OL6.15-4
  - vector architecture, 524–525
  - in x86, 524
- SIMMs (single inline memory modules), OL5.17-5, OL5.17-6
- Simple programmable logic devices (SPLDs), A-666
- Simplicity, 171
- Simultaneous multithreading (SMT), 531–533
  - support, 531
  - thread-level parallelism, 531
  - unused issue slots, 531
- Single error correcting/Double error correcting (SEC/DEC), 434–436
- Single instruction single data (SISD), 523
- Single precision. *See also* Double precision
  - binary representation, 210
  - defined, 207
- Single-clock-cycle pipeline diagrams, 308
  - illustrated, 310
- Single-cycle datapaths. *See also* Datapaths
  - illustrated, 298
  - instruction execution, 299
- Single-cycle implementation
  - control function for, 281
  - defined, 281
  - nonpipelined execution *versus* pipelined execution, 287
  - non-use of, 284
  - penalty, 283
  - pipelined performance *versus*, 285
- Single-instruction multiple-thread (SIMT), B-27–30
  - overhead, B-35
  - multithreaded warp scheduling, B-28
  - processor architecture, B-28
  - warp execution and divergence, B-29–30
- Single-program multiple data (SPMD), B-22
- Smalltalk-80, OL2.22-8
- Smart phones, 7
- Snooping protocol, 479–481
- Snoopy cache coherence, OL5.12-7
- Software optimization
  - via blocking, 427–432
- Sort algorithms, 146
- Software
  - layers, 13
  - multiprocessor, 516
  - parallel, 517
  - as service, 7, 547, 574
  - systems, 13
- Sort procedure, 140–144. *See also* Procedures
  - code for body, 140–142
  - full procedure, 143–144
  - passing parameters in, 143
  - preserving registers in, 143
  - procedure call, 143
  - register allocation for, 140
- Sorting performance, B-54–55
- Space allocation
  - on heap, 107–110
  - on stack, 106
- SPARC
  - annulling branch, D-23
  - CASA, D-31
  - conditional branches, D-10–12
  - fast traps, D-30
  - floating-point operations, D-31
  - instructions, D-29–32
  - least significant bits, D-31
  - multiple precision floating-point results, D-32
  - nonfaulting loads, D-32
  - overlapping integer operations, D-31
  - quadruple precision floating-point arithmetic, D-32
  - register windows, D-29–30
  - support for LISP and Smalltalk, D-30
- Sparse matrices, B-55–58
- Sparse Matrix-Vector multiply (SpMV), B-55, B-57, B-58
  - CUDA version, B-57
  - serial code, B-57
  - shared memory version, B-59
- Spatial locality, 388
  - large block exploitation of, 405
  - tendency, 392
- SPEC, OL1.12-11–1.12-12
  - CPU benchmark, 46–48
  - power benchmark, 48–49
- SPEC2000, OL1.12-12
- SPEC2006, 246, OL1.12-12
- SPEC89, OL1.12-11
- SPEC92, OL1.12-12
- SPEC95, OL1.12-12
- SPECrate, 554
- SPECratio, 47
- Special function units (SFUs), B-35, B-50
  - defined, B-43
- Speculation, 344–345
  - hardware-based, 352
  - implementation, 344
  - performance and, 344
  - problems, 344
  - recovery mechanism, 344
- Speed-up challenge, 518
  - balancing load, 518–519
  - bigger problem, 520–521
- Spilling registers, 72, 101
- Split algorithm, 568
- Split caches, 411
- Stack architectures, OL2.22-4
- Stack pointers
  - adjustment, 104
  - defined, 101
  - values, 104
- Stacks
  - allocating space on, 106
  - for arguments, 145
  - defined, 101
  - pop, 101
  - push, 101, 104
- Stalls, 291
  - as solution to control hazard, 292
  - avoiding with code reordering, 291
  - behavioral Verilog with detection, OL4.13-6
  - data hazards and, 324–328
  - illustrations, OL4.13-23, OL4.13-30
  - insertion into pipeline, 326
  - load-use, 329
  - memory, 414
  - write-back scheme, 413
  - write buffer, 413
- Standby spares, OL5.11-8
- State
  - in 2-bit prediction scheme, 333
  - assignment, A-658, C-27
  - bits, C-8
  - exception, saving/restoring, 462
  - logic components, 261
  - specification of, 446

- State elements
    - clock and, 262
    - combinational logic and, 262
    - defined, 260, A-636
    - inputs, 261
    - in storing/accessing instructions, 264
    - register file, A-638
  - Static branch prediction, 345
  - Static data
    - segment, 107
  - Static multiple-issue processors, 343, 345–349. *See also* Multiple issue control hazards and, 345 instruction sets, 345 with LEGv8 ISA, 345–348
  - Static random access memories (SRAMs), 392, 393, A-646–650
    - array organization, A-650
    - basic structure, A-649
    - defined, 21, A-646
    - fixed access time, A-646
    - large, A-647
    - read/write initiation, A-647
    - synchronous (SSRAMs), A-648
    - three-state buffers, A-647, A-648
  - Static variables, 106
  - Steady-state prediction, 333
  - Sticky bits, 228
  - Store buffers, 355
  - Store instructions. *See also* Load instructions
    - access, B-41
    - base register, 274
    - compiling with, 71
    - conditional, 126
    - defined, 71
    - EX stage, 305
    - ID stage, 302
    - IF stage, 302
    - instruction dependency, 323
    - MEM stage, 304
    - unit for implementing, 267
    - WB stage, 304
  - Store register, 72
  - Stored program concept, 63
    - as computer principle, 88
    - illustrated, 89
    - principles, 171
  - Strcpy procedure, 112–113. *See also* Procedures
    - as leaf procedure, 113
    - pointers, 113
  - Stream benchmark, 564
  - Streaming multiprocessor (SM), B-48–49
  - Streaming processors, B-34, B-49–50
    - array (SPA), B-41, B-46
  - Streaming SIMD Extension 2 (SSE2) floating-point architecture, 232
  - Streaming SIMD Extensions (SSE) and advanced vector extensions in x86, 232–233
  - Stretch computer, OL4.16-2
  - Strings
    - defined, 111
    - in Java, 113–115
    - representation, 111
  - Strip mining, 526
  - Striping, OL5.11-4
  - Strong scaling, 521
  - Structural hazards, 288–289, 305
  - STUR (store register), 64
  - STURB (store byte), 64
  - STURH (store half), 64
  - STURW (store word), 64
  - STXR (store exclusive register), 64, 126
  - SUB (subtract), 64
  - SUBI (subtract immediate), 64
  - SUBIS (subtract immediate and set flags), 64
  - SUBS (subtract and set flags), 64
  - Subnormals, 230
  - Subtraction, 188–191. *See also* Arithmetic binary, 188–189 floating-point, 220 negative number, 190 overflow, 190
  - Subword parallelism, 230–232, 365, D-17
    - and matrix multiply, 238–242
  - Sum of products, A-599, A-600
  - Supercomputers, OL4.16-3
    - defined, 5
  - SuperH, D-15, D-39–40
  - Superscalars
    - defined, 349, OL4.16-5
    - dynamic pipeline scheduling, 349
    - multithreading options, 516
  - Surfaces, B-41
  - Swap procedure, 138. *See also* Procedures
    - body code, 138
    - full, 139–140, 143–145
    - register allocation, 138
  - Swap space, 448
  - Symbol tables, 130
  - Synchronization, 125–127, 568
    - barrier, B-18, B-20, B-34
    - defined, 534
    - lock, 125
    - overhead, reducing, 44–45
    - unlock, 125
  - Synchronizers
    - defined, A-664
    - failure, A-665
    - from D flip-flop, A-664
  - Synchronous DRAM (SRAM), 393–394, A-648, A-653
  - Synchronous SRAM (SSRAM), A-648
  - Synchronous system, A-636
  - Syntax tree, OL2.15-3
  - System calls
    - defined, 459
  - Systems software, 13
  - SystemVerilog
    - cache controller, OL5.12-2
    - cache data and tag modules, OL5.12-6
    - FSM, OL5.12-7
    - simple cache block diagram, OL5.12-4
    - type declarations, OL5.12-2
- ## T
- Tablets, 7
  - Tags
    - defined, 398
    - in locating block, 421
    - page tables and, 448
    - size of, 423
  - Tail call, 109
  - Task identifiers, 460
  - Task parallelism, B-24
  - Task-level parallelism, 516
  - Tebibyte (TiB), 5
  - Telsa PTX ISA, B-31–34
    - arithmetic instructions, B-33
    - barrier synchronization, B-34
    - GPU thread instructions, B-32
    - memory access instructions, B-33–34
  - Temporal locality, 388
    - tendency, 392
  - Temporary registers, 68, 102
  - Terabyte (TB), 6
    - defined, 5
  - Texture memory, B-40
  - Texture/processor cluster (TPC), B-47–48
  - TFLOPS multiprocessor, OL6.15-6



Thrashing, 464  
 Thread blocks, 542  
   creation, B-23  
   defined, B-19  
   managing, B-30  
   memory sharing, B-20  
   synchronization, B-20  
 Thread parallelism, B-22  
 Threads  
   creation, B-23  
   CUDA, B-36  
   ISA, B-31–34  
   managing, B-30  
   memory latencies and, B-74–75  
   multiple, per body, B-68–69  
   warps, B-27  
 Three Cs model, 459–461  
 Three-state buffers, A-647, A-648  
 Throughput  
   defined, 30–31  
   multiple issue and, 342  
   pipelining and, 286, 342  
 Thumb, D-15, D-38  
 Timing  
   asynchronous inputs, A-664–665  
   level-sensitive, A-663–664  
   methodologies, A-660–665  
   two-phase, A-663  
 TLB misses, 453. *See also* Translation-lookaside buffer (TLB)  
   handling, 461–465  
   occurrence, 461  
   problem, 464  
 Tomasulo's algorithm, OL4.16-3  
 Touchscreen, 19  
 Tournament branch predictors, 334  
 Tracks, 395–396  
 Transfer time, 397  
 Transistors, 25  
 Translation Table Base Register (TTBR), 449  
 Translation-lookaside buffer (TLB), 452–454, D-26–27, OL5.17-6. *See also* TLB misses  
   associativities, 454  
   illustrated, 453  
   integration, 457  
   Intrinsity FastMATH, 454–457  
   typical values, 454  
 Transmit driver and NIC hardware  
   time *versus* receive driver and NIC hardware time, OL6.9-8

Tree-based parallel scan, B-62  
 Truth tables, A-593  
   ALU control lines, C-5  
   for control bits, 272  
   datapath control outputs, C-17  
   datapath control signals, C-14  
   defined, 272  
   example, A-593  
   next-state output bits, C-15  
   PLA implementation, A-601  
 Two's complement representation, 77–78  
   advantage, 78  
   negation shortcut, 79–80  
   rule, 81  
   sign extension shortcut, 80–81  
 Two-level logic, A-599–602  
 Two-phase clocking, A-663  
 TX-2 computer, OL6.15-4

## U

Unconditional branches, 94  
 Underflow, 206  
 Unicode  
   alphabets, 113  
   defined, 113  
   example alphabets, 114  
 Unified GPU architecture, B-10–12  
   illustrated, B-11  
   processor array, B-11–12  
 Uniform memory access (UMA), 534, B-9  
   multiprocessors, 534  
 Units  
   commit, 350, 355  
   control, 259, 271–273, C-4–8, C-10, C-12–13  
   defined, 227  
   floating point, 227  
   hazard detection, 324, 327–328  
   for load/store implementation, 267  
   special function (SFUs), B-35, B-43, B-50  
 UNIVAC I, OL1.12-5  
 UNIX, OL2.22-8, OL5.17-9–5.17-12  
   AT&T, OL5.17-10  
   Berkeley version (BSD), OL5.17-10  
   genius, OL5.17-12  
   history, OL5.17-9–5.17-12  
 Unlock synchronization, 125  
 Unscaled signed immediate off set, 166  
 Unsigned numbers, 75–82  
 Use latency  
   defined, 346  
   one-instruction, 346

## V

Vacuum tubes, 25  
 Valid bit, 400  
 Variables  
   C language, 106  
   programming language, 67  
   register, 67  
   static, 106  
   storage class, 106  
   type, 106  
 VAX architecture, OL2.22-4, OL5.17-7  
 Vector lanes, 526  
 Vector processors, 523–524. *See also* Processors  
   conventional code comparison, 525–526  
   instructions, 524  
   multimedia extensions and, 524–525  
   scalar *versus*, 526–527  
 Vectored interrupts, 337  
 Verilog  
   behavioral definition of MIPS ALU, A-613  
   behavioral definition with bypassing, OL4.13-4–4.13-6  
   behavioral definition with stalls for loads, OL4.13-6  
   behavioral specification, A-609, OL4.13-2–4.13-4  
   behavioral specification of multicycle MIPS design, OL4.13-12–4.13-13  
   behavioral specification with simulation, OL4.13-2  
   behavioral specification with stall detection, OL4.13-6  
   behavioral specification with synthesis, OL4.13-11–4.13-16  
   blocking assignment, A-612  
   branch hazard logic implementation, OL4.13-8–4.13-10  
   combinational logic, A-611–614  
   datatypes, A-609–610  
   defined, A-608  
   forwarding implementation, OL4.13-4  
   MIPS ALU definition in, A-623–626  
   modules, A-611  
   multicycle MIPS datapath, OL4.13-14

Verilog (*Continued*)  
 nonblocking assignment, A-612  
 operators, A-610  
 program structure, A-611  
 reg, A-609–610  
 sensitivity list, A-612  
 sequential logic  
   specification, A-644–646  
   structural specification, A-609  
   wire, A-609–610  
 Vertical microcode, C-32  
 Very large-scale integrated (VLSI)  
   circuits, 25  
 Very Long Instruction Word (VLIW)  
   defined, 345  
   first generation computers, OL4.16-5  
   processors, 345  
 VHDL, A-608–609  
 Video graphics array (VGA) controllers,  
   B-3–4  
 Virtual addresses  
   causing page faults, 462  
   defined, 442  
   mapping from, 442–443  
   size, 444  
 Virtual machine monitors (VMMs)  
   defined, 438  
   implementing, 494  
   laissez-faire attitude, 494  
   page tables, 463  
   in performance improvement, 441  
   requirements, 440  
 Virtual machines (VMs), 438–441  
   benefits, 438  
   illusion, 463  
   instruction set architecture support,  
     441  
   performance improvement, 441  
   for protection improvement, 438  
 Virtual memory, 441–465. *See also* Pages  
   address translation, 443, 452–454  
   integration, 457–459  
   for large virtual addresses, 450–451  
   mechanism, 464  
   motivations, 427–442  
   page faults, 442, 448  
   protection implementation, 459–460  
   segmentation, 445  
   summary, 463  
   virtualization of, 463  
   writes, 452

Virtualizable hardware, 440  
 Virtually addressed caches, 458  
 Visual computing, B-3  
 Volatile memory, 22

## W

Wafers, 26  
   defects, 26  
   dies, 26–27  
   yield, 27  
 Warehouse Scale Computers (WSCs), 7,  
   545–550, 574  
 Warps, 544, B-27  
 Weak scaling, 521  
 Wear levelling, 395  
 While loops, 95  
 Whirlwind, OL5.17-2  
 Wide area networks (WANs), 24. *See also*  
   Networks  
 Wide immediate operands, 115–117  
 Words  
   accessing, 69  
   defined, 67  
   double, 158  
   load, 69, 71  
   quad, 158  
   store, 71  
 Working set, 464  
 World Wide Web, 4  
 Worst-case delay, 283  
 Write buffers  
   defined, 408  
   stalls, 413  
   write-back cache, 409  
 Write invalidate protocols, 479  
 Write serialization, 479  
 Write-back caches. *See also* Caches  
   advantages, 469  
   cache coherency protocol, OL5.12-5  
   complexity, 409  
   defined, 408, 469  
   stalls, 413  
   write buffers, 409  
 Write-back stage  
   control line, 313  
   load instruction, 303  
   store instruction, 305  
 Writes  
   complications, 408  
   expense, 464

handling, 407–409  
 memory hierarchy handling of,  
   469–470  
 schemes, 408  
 virtual memory, 451  
 write-back cache, 408, 409  
 write-through cache, 408, 409

Write-stall cycles, 414  
 Write-through caches. *See also* Caches  
   advantages, 469  
   defined, 407, 469  
   tag mismatch, 408

## X

x86, 154–162  
   Advanced Vector Extensions in, 232  
   brief history, OL2.22-6  
   conclusion, 162  
   data addressing modes, 157–158  
   evolution, 154–157  
   first address specifier encoding, 162  
   instruction encoding, 161–162  
   instruction formats, 161  
   instruction set growth, 170  
   instruction types, 160  
   integer operations, 158–160  
   registers, 157  
   SIMD in, 522  
   Streaming SIMD Extensions in,  
     232–233  
   typical instructions/functions, 161  
   typical operations, 161  
 Xerox Alto computer, OL1.12-8  
 XMM, 232

## Y

Yahoo! Cloud Serving Benchmark  
   (YCSB), 556  
 Yield, 27  
 YMM, 232

## Z

Zettabyte, 6