Index

A

1-bit adder, 186 (fig.)
  input/output specifications, 186
1-bit ALU, 184–187
  for most significant bit, 192
  performing ADD, OR, addition, 188 (fig.), 190 (fig.), 191 (fig.)
4-bit adder, 196
  generate signal for, 196–197
  propagate signal for, 196
32-bit ALU, 187–188
  32 1-bit ALU construction, 189 (fig.), 193 (fig.)
  final, 194 (fig.)
  for MIPS, 189–192
32-bit constants, 182
64-bit register, 223
aborts. (*see* interrupts)
abs.d instruction, A-65
abs instruction, A-51
abs.s instruction, A-66
abstractions, **17**
  interface between levels, 18
  principle of, 18
accumulator, 150
  extended, 151
Ack signal, 553
Ada, A-9
  overflow and, 178
add.d instruction, 238, A-66
adders, 276, (*see also* addition)
  1-bit, 186 (fig.)
  (3, 2), 263
  4-bit, 196
  carry-lookahead, 196
  carry out signal hardware, 187
  function, 277 (fig.)
  ripple carry, 188
  two's compliment, 188
addi instruction, 125, 178, A-51
add immediate instruction. (*see* addi instruction)
add immediate unsigned instruction. (*see* addiu instruction)
add instruction, 178, 270, 276, 283, 365, A-51
  forwarding clock cycles, 416
addition, 175, (*see also* division); multiplication; subtraction
  binary, 176 (fig.)
  carry save, **263**
    of 4-bit numbers, 264
  floating-point, 230–233, 234 (fig.)
    arithmetic unit dedicated to, 235 (fig.)
  of significands, 231
addiu instruction, 178
AddrCtl, C-18, C-19
addresses, 98, (*see also* address space)
  byte, 137
  memory, 99 (fig.)
  physical, **482**
  return, 119
  size of, 515
  translation, 482, 493, (*see also* virtual addresses; virtual memory)
  virtual, 482 (fig.)
  word, 130
addressing modes, 129, 131 (fig.), (*see also specific types of addressing modes*)
  list of, 130
  VAX, E-5–E-6
address space, 481
  single, 603
address specifier, **E-7**
add.s instruction, 238, A-66
addu instruction, 178, A-51
add unsigned instruction. (*see.* addu instruction)
Aiken, Howard, 34
aliasing, 493
ALOG, 254
Alto, 582
ALU, **182**, 271, 276, (*see also* ALU control)
  1-bit, 184–187, 189–192
  32-bit, 187–192
  64-bit, 202
  adding 32-bit operands, 192–197
  building blocks of, 184
  construction, 197
  function performance, 286
  less operation output, 190
  operation values, 195
  in R-type instruction, 317
  sharing, 312
  signed-immediate input to, 421, 423 (fig.)
  symbol representation, 195
ALU control, 285–291
  bits, 187 (fig.)
  block, **289**, 290 (fig.)

ALU control *(continued)*
  lines, 289
ALUOp, 287, 314, 318, 319, 320, 321, 328, 329, 382
  bits, 289
  field mapping, 287
ALUoutput, 319
ALUSelA, 318, 319, 320, 321, 328, 329
ALUSelB, 314, 318, 320, 321, 328, 423
ALUSrc, 382
Amdahl, Gene, 355, 642
Amdahl's Law, 71, 89, 478, 535, 604, 638, 639
  make the common case fast corollary, 71
American Standard Code for Information Interchange. (*see* ASCII)
andi instruction, 182, A-52
and instruction, 270, 276, 283, 365, A-52
  ID stage of, 401
AND operation, 181–182, B-5
AND plane, B-12
Annals of the History of Computing, 40
ANSI, 563
Apple II, 37, 39 (fig.)
applications software, **8**
architecture, 17, (*see also* MIPS architecture)
arithmetic, (*see also* addition; division; floating-point; multiplication; overflow; rounding; subtraction)
  instructions, 149, 150 (fig.), A-51–A-55
  logic unit. (*see* ALU)
  overflow exception, 348
arithmetic mean (AM), 70, 75
  weighted, 70
ARPANET network, 547
array indices, 143
  pointers vs., 146
arrays, 143, (*see also* pointers)
  of logic elements, B-17–B-18
  setting, to zeros, 145 (fig.), 146
ASCII, **161**
  representation of characters, 161
assemblers, **6**, 109, **A-4**, A-10–A-17, (*see also* object file)
  directives, **A-5**
  function of, A-10–A-11
  microcode, C-24
  MIPS, 110, 115
  relocation information, **A-14**
  syntax, A-48–A-50
  for Unix systems, A-17
assembly, **109**
assembly language, **6**, 7 (fig.), 109, **A-3**
  code, A-4 (fig.)
  disadvantages, A-10
  high-level languages vs., A-13
  MIPS, 96, 102, 107, 118, 123, 132, 174, 180, 183, 224
  syntax for, 335
  when to use, A-8–A-10
Assessing the Speed of Algol 60, 72
Association for Computing Machinery, 256
asynchronous bus, 551–556, **553**, 556, (*see also* buses)
Atanasoff, John, 33
Atlas computer, 525
atomic swap operation, **614**
autodecrement addressing, **133**
autoincrement addressing, **133**
autoincrement deferred addressing, E-5
automatic teller machines (ATMs), 537
average memory access time (AMAT), 530

B

backpatching, **A-12**
backplane, 549
  buses, **549**, 550–551, 552 (fig.)
Bailey, David H., 637
bandwidth
  bisection, **623**
  bus, 556–558, 564
  higher memory, 474 (fig.)
  I/O, 536
  total network, **623**
barrel shifter, **197–198**
base addressing, 130, 131 (fig.)
bclf instruction, 238
bclt instruction, 238
bczf instruction, A-57
bczt instruction, A-57
Bell, C.G., 602
Bell, Gordon, 644
  Prize winners, 645 (fig.)
benchmarks, **66**, (*see also* peak performance; performance)
  Dhrystone, 73–74, 78
  kernel, 79
  selection of, 80
  small, 68
  SPEC processor, 67
  synthetic, **73–74**
  toy programs as, 79
  use of, 66
  Whetstone, 73–74, 78–79
beq instruction, 111, 115, 270, 283, 365, A-58
  datapath in operation for, 303 (fig.)
  operands, 279
  steps to executing, 302
beqz instruction, A-58
bge instruction, A-58
bgequ instruction, A-58
bgezal instruction, A-58, A-59
bgez instruction, A-58
bgt instruction, A-58
bgtu instruction, A-58
bgtz instruction, A-59
bias, 175
biased notation, 175
bidirectional links, 622
Bigelow, John, 32
Big picture, **16**
  arithmetic for computers, 243

- assemblers, linkers, SPIM simulator, A-13
- computer abstraction and technology, 18
- computer components, 16
- datapath and control, 354
- enhancing performance with pipelining, 421, 431
- instructions, 108
- interfacing processors and peripherals, 563
- I/O system performance, 579
- mapping control to hardware, C-27
- memory hierarchy design, 514
- parallel processors, 634
- principles of locality, 512
- role of performance, 58
- VAX and MIPS goals, 4, 5

BINAC, 34
binary digit, **5**, 103
binary numbers, **5**, 6, 103, (*see also* numbers)
- converting from smaller size to large, 172–173
- negating, 171
- in scientific notation, 225

b instruction, A-57
bisection bandwidth, **623**
bit map, **12**
- purpose, 13

bits, **5**, 103
- classifying groups of, 207
- dirty, **490**
- done, **567**
- least significant, 169
- line, B-28
- most significant, 169
- reference, **489**
- sign, 170
- status, 611
- sticky, 243, 266
- use, **489**
- valid, 461

Blaauw, Gerrit, 442
ble instruction, A-59
bleu instruction, A-59
blez instruction, A-59
block(s), **456**, (*see also* pages)
- cache, 612
- combinational, B-4
- "control-logic," C-4
- field selection, 461
- finding, 508–510
- fixed-size, 484
- logic, state, B-4
- modulo number, 459–460
- multiword, 469, 471
- placement, 502–508
  - strategy, 503
- replacement, 510–511
  - LRU, 510
  - random, 510
- transfers, 557
- variable-size, 484

block size
- increasing, 472–473
  - disadvantage of, 473
- miss rate vs., 472 (fig.)

blt instruction, A-17, A-59
bltu instruction, A-59
bltzal instruction, A-59
bltz instruction, A-60
Bnegate control line, **191**
bne instruction, 111, 115, A-17, A-60
bnez instruction, A-60
Boolean algebra, **B-5**-B-7
- AND operator and, B-5
- NOT operator and, B-5–B-6
- OR operator and, B-5

Booth's algorithm, 205–212
- first step cases, 208
- with negative numbers, 209 (fig.)
- for positive numbers, 209 (fig.)
- recasting, 211
- as the sum, 211

Branch, **292**, 382
branch-delay slots, 449
branch equal instruction. (*see* beq instruction)
branches
- addressing, 127–130
- address select logic types, C-18
- conditional, 111, 128, 129, 149, 150 (fig.)
- datapath for, 282 (fig.)
- delayed, 119, **443**, 449, A-37
- operations required of, 281
- target address, 281
- unconditional, 112

branch hazards. (*see* control hazards)
branch instruction, 291, A-57–A-61
- pipeline impact on, 424 (fig.)
- single state machine and, 329 (fig.)
- VAX, E-10

branch not equal instruction. (*see* bne instruction)
branch prediction, 430
break instruction, A-69
breakpoint, **A-41**
- setting, A-41

Brooks, Fred Jr., 442
Browning, Robert, 148
bubbles, **399**, (*see also* pipeline stalls)
- pipelined instruction sequence with, 401 (fig.)

buffers, (*see also* TLB)
- DECStation 3100 write, **13** (fig.)
- frame, 13
- raster refresh, 13
- store, 512
- three-state, B-28, B-29 (fig.)
- tri-state, B-28

Burks, Arthur, 32, 94, 111, 249, 521
Burroughs 5500, 72
bus arbitration, **559**–562
- centralized, parallel, 561

- bus arbitration *(continued)*
  - daisy chain, 561
  - distributed, by collision detection, 562
  - distributed, by self-selection, 561–562
- bus bandwidth, 556, 564
  - increasing, 556–558
- buses, **275**, **548**–563, B-18, (*see also* bus arbitration; bus transactions)
  - access to, 558–559
  - asynchronous, 551–556, **553**, 556
  - backplane, **549**, 550–551, 552 (fig.)
  - characteristics, 563, 619
  - contents of, 548
  - daisy chain, 561 (fig.), 562
  - design difficulties, 548
  - disadvantages of, 548
  - IBM PC, 552
  - I/O, **549**, 550
  - latency of, 548
  - processor-memory, **549**
  - SCSI, 552
  - shared, 314
  - standard, **549**
  - standards for, 562–564
    - characteristics, 564 (fig.)
  - synchronous, 551–556, **553**, 556, 558
  - types of, 549–551
- bus masters, **559**
  - multiple, 559
- bus priority, **560**
  - fairness and, **560**
- bus request, 559
- bus transactions, 549
  - initial steps in, 560 (fig.)
  - read, 549
  - write, 549
- bypassing. (*see* forwarding)
- byte order, **A-45**
  - big-endian, A-45
  - little-endian, A-45
- bytes
  - address, 137
  - size, 100
- byte/word/long displacement deferred addressing, E-5

C

- cache, **16–17**, **458**–481, 526, 635, (*see also* block(s); memory; memory hierarchy; set associative cache)
  - as a slave, 525
  - basics, 480–481
  - block, 612
    - finding, 469
  - in DECStation 3100, 466 (fig.)
  - direct mapped, **459**, 460 (fig.)
  - file, **579**
  - hit rate, 468
  - inconsistent, 467
  - index, 461
  - lower address portion, 463 (fig.)
  - memory system design and, 473–476
  - misses
    - handling, 464–465
    - reducing penalty of, 465
    - stalls and, 465
  - miss rates, 468
  - multilevel, 520
  - number of bits needed for, 464
  - performance, 476–448
  - read requests, 467
  - reference request contents, 462 (fig.)
  - referencing a word, 459 (fig.)
  - simulating, 515
  - snooping, 635
  - split, 469
  - store implementation in, 512
  - structure, 459
  - tags, **460**
  - using four-word blocks, 470 (fig.)
  - virtually addressed, **493**, 495
  - write requests, 467
- cache-coherency, 608, (*see also* multiprocessors)
  - block size and, 612
  - bus-based, 616
  - multiprocessor, 610–612
  - protocols, **609**
    - example, 612–614
    - write-invalidate, 613 (fig.)
  - steps, 616 (fig.)
- callee
  - procedure, **A-22**
  - save, 121, 124
  - saved registers, A-26
- caller
  - procedure, **A-22**
  - save, 121, 124
  - saved registers, A-26
- call instruction, 147
- capabilities, **526**
- capacity misses, 513
- CarryIn, **185**, 194–195, 196
- CarryOut, **185**, 196
  - at 1, 185
- CAS (Column Access Strobe), B-32
- case statement, 115
- cathod ray tube. (*see* CRT)
- Cause register, 345, 346, 434, 569, 570, A-32 (fig.), A-33
  - fields, A-31
- CauseWrite, 346
- CDC 6600, 35, 72, 253, 442, 443 (fig.), 582
- central processing unit. (*see* CPU)
- c.eq.d instruction, A-66
- c.eq.s instruction, A-66
- channel controllers, **573**
- Charles V, 92

chips, 13, **23**, (*see also* integrated circuits)
CISC machines, A-7, **E-3**
Clark, David, 20
clear1 array, 143
  clear2 vs., 145
c.le.d instruction, A-66
c.le.s instruction, A-66
clock, 53, 274 (fig.), **B-18**–B-21
  cycle time, B-18
  edge, 273
  frequency, B-18
  period, **53**, B-18
  rate, **53**
  signal, B-19 (fig.)
  skew, **B-40**, B-41 (fig.), B-42
  ticks, 53
  variable, 309
    implementation, 310
clock cycles, **53**, (*see also* clock cycle time)
  CPU, 57, 479
  decreasing number of, 43
  examples, 273–276
  instruction execution in, 314–322
  instruction length, 306
  of instruction types, 310
  length of, 364
  measurement of, 58
  memory-stall, 476–477, 517
  per instruction. (*see* CPI)
  required for program, 55–56
clock cycle time, 53, B-18
clocking methodology, 273
  edge-triggered, 273, 275 (fig.)
  two-phase, **B-41–B-42**
c.lt.d instruction, A-66
c.lt.s instruction, A-67
CM-1, 641, 642
CM-2, 599–601, 642
  I/O channel, 601
  operation of, 599
  organization, 600, 601 (fig.)
  performance, 600
  photo, 600 (fig.)
CM-5, 626–630, 644
  computational node, 627
  data network fat-tree topology, 628 (fig.)
  features, 627
  photo, **629**
  processing node, 627 (fig.)
Cobol, 8
Cocke, John, 442
coherency problem, **575**
collision, **591**
  misses, 513
combinational elements, 272
combinational logic, 274 (fig.), B-3, B-8–B-18
  block inputs, B-20 (fig.)
comments, 95
communication, network computers and, 20
comparison instructions, A-56–A-57
compilers, **6**, 8–9, 29, **A-6**
  assembly language coders and, 147
  compilation speed and, A-7
  C translation and, 96
  gcc, A-22
  MIPS, A-22
  optimizations, 66
  programmers vs., A-9
  in systems software level, 9 (fig.)
compiling, **6**, 7 (fig.)
compulsory misses, 513
computer-aided design (CAD), 276, 289, B-39
computer(s), (*see also specific types of computers*)
  arithmetic, 246
  commercial, characteristics, 40 (fig.)
  communicating to, 20
  components, 18
  digital, B-4
  embedded, **A-8**
  first electronic, 31–34
  generations, 38, 40 (fig.)
  hardware. (*see* hardware)
  key principles of, 108
  organization of, 30 (fig.)
  parallel, 38, 40
  price/performance prediction, 28
  systems, 8–10, 9 (fig.)
Conant, James B., 640
condition codes, **E-10**
conflict misses, 513
constant bit density, **546**
constant-manipulating instructions, A-55
constants, 124–125
  32-bit, 182
  multiplying by, 210
context switch, **497**
control, **14**, 15 (fig.), 344, (*see also* control function; control hazards; control lines)
  alternative methods for, 354 (fig.)
  ALU, 285–291
  Big picture of, 354
  checking for exceptions, 348–349
  in computer design, 352
  defining, 322–333
  definition of, 351
  design, 333–334
  for exceptions, 432 (fig.)
  finite state, 324, 332 (fig.)
  forwarding, 413
  hardwired, 352
  for hazards, 399–412
  main, unit, 291–306
  microprogrammed, 352, **C-22**
  multicycle, 322
  multiplexor, 324
  off-chip, 351

- control *(continued)*
  - optimizing, implementation, C-20–C-22
  - organizing, to reduce logic, C-24–C-27
  - pipelined, 381–390
    - overview, 422 (fig.)
  - word, C-10, C-22
    - address-control lines in, C-21 (fig.)
    - lower bits, C-13, C-14 (fig.)
- control function
  - definition, 302
  - logic, PLA implementation and, C-16 (fig.)
  - parts, C-8
  - PLA implementation and, C-14–C-15
  - ROM implementation and, C-10–C-14
  - structured implementation of, 305 (fig.)
  - truth table, 304 (fig.)
- control hazards, **424**–430, (*see also* hazard tests)
  - always stalling and, 425
  - assuming branch not taken and, 425–430
  - branch with stalls and, 426 (fig.)
  - exceptions and, 430–434
- controller
  - channel, **573**
  - disk, **544**
  - implementation, 333
  - I/O, 573
  - microcode, 343 (fig.)
  - time, **544**
  - traffic light, B-38 (fig.)
- control lines, 293 (fig.), (*see also* control)
  - buses and, 553
  - for final three stages, 388
  - labeling, 382
  - optional, 388
  - in pipelined control, 382
  - setting of, 296 (fig.)
  - values, 388 (fig.)
- control signals, 275, 334, (*see also* control)
  - action caused by setting of, 316
  - determining how pc is written, 324 (fig.)
  - function of, 294 (fig.)
  - partition of, 335
  - pipelined datapath and, 386 (fig.)
    - connected to control portion of pipeline registers, 391
- control unit, (*see also* control; main control unit)
  - address select logic for, C-20 (fig.)
  - cache misses and, 464
  - with counter, C-17
  - logic equations for, C-4, C-7 (fig.)
    - implementation of, C-8
  - as microcode, C-22–C-23 (fig.)
  - microprogram for, 338–344
  - for MIPS, C-6 (fig.)
  - techniques for implementing, C-3
  - using explicit counter, C-18 (fig.)
- Coonen, Jerome T., 255
- C operations, 182, (*see also* C programming)
- coprocessor, **255**
  - coprocessor 0, **A-30**
- copy back, **490**, 511
- CPI, **56**, 438
  - measurement of, 58
  - for MIPS instruction categories, 260
  - obtaining, 57
  - stall cycles and, 479
- C programming, 94, (*see also* sort procedure, C programming)
  - assignment statements, D-4
  - conditional statements in, D-5–D-6
  - loops in, D-6–D-7
  - procedures in, D-7
  - relational expressions in, D-5–D-6
  - routine, A-8 (fig.)
  - variable declarations in, D-3–D-4
    - standard types of, D-4
- CPU, **14**, (*see also* CPU time)
  - clock cycles, 57, 479
  - execution time, 54, 58
  - performance, **53**, 54
    - improvement, 536
- CPU time, **52**, 476–477, 516–517, (*see also* CPU)
  - system, **52**
  - user, **52**
- Cray-1, 35, 39 (fig.)
  - characteristics of, 40
- Cray-2, 65
- Cray C-90, 441
- Cray computers, 602
  - overflow on, 253
  - rounding errors and, 253
- Cray, Seymour, 35, 252, 532
- CRT, 12
  - display, 12 (fig.)
- cvt.d.s instruction, A-67
- cvt.d.w instruction, A-67
- cvt.s.d instruction, A-67
- cvt.s.w instruction, A-67
- cvt.w.s instruction, A-67
- c.x.d instruction, 238
- c.x.s instruction, 238
- cycles, 53
- cylinder, **542**

D

- daisy chain bus, 561 (fig.), (*see also* buses)
  - multiple, 562
  - protocol followed by device on, 562
- dance hall, **620**
- data alignment, A-48
- data bus width, 446
- data hazards, 390–399, **396**, 397, (*see also* hazard tests)
  - clock cycle 9 and, 396 (fig.)
  - clock cycles 1 and 2 and, 392 (fig.)
  - clock cycles 3 and 4 and, 393 (fig.)
  - clock cycles 5 and 6 and, 394 (fig.)

  clock cycles 7 and 8 and, 395 (fig.)
  control for, 399–412
  reducing, 412–424
data layout directives, **A-14**
data memory unit, 280 (fig.)
data movement instructions, A-64–A-65
data parallelism, **597**
datapath, **14**, 15 (fig.)
  adding, for branches, 284
  for branches, 282
  building, 276–282
  combining, 284 (fig.)
  with control unit, 295 (fig.)
  creating single, 283–285
  for fetching instructions, 277 (fig.)
  to handle load instruction, 379–380 (fig.)
  instruction fetch portion of, 284, 285 (fig.)
  jump instruction and, 307
  for load or store, 280 (fig.)
  for MIPS architecture, 286 (fig.)
  multicycle, 313 (fig.)
    with control lines, 315 (fig.)
    for MIPS, 313 (fig.)
    pipelined performance consequences, 438 (fig.)
  for multicycle implementation, 323
  with multiplexors and control lines, 293 (fig.)
  in operation for beq instruction, 303 (fig.)
  pipelined, 367–381, 370 (fig.)
    with control signals identified, 386 (fig.)
  for R-type instructions, 279 (fig.)
  single-cycle, 369 (fig.)
    pipeline performance consequences, 438 (fig.)
data rate, **538**
  I/O devices and, 540
DataRdy signal, 553
data selector, **283**
data transfer, 570–573
  instructions, 149, 150 (fig.)
  interrupt driven, **570**
DEC, 35, 602, (*see also* VAX)
  Alpha, 154, 436, 441, 444, 518
  NVAX, 436
  PDP-8, 35, 37 (fig.), 40
  PDP-11, 516, 525, 582, E-22, E-23
decimal numbers, 103, (*see also* numbers)
decoders, **B-9**, (*see also* combinational logic)
  three-bit, B-9 (fig.)
  three-to-eight, B-9
  two-level, B-32
    column access, B-32
    row access, B-32
decoding, **325**
DECStation 3100, 466–469
  cache, 466 (fig.)
    read requests, 467
  measurements, 469
  miss rates, 468 (fig.)
  processing read or write through, 495 (fig.)
  TLB in, 493
  virtual address to data item in, 494 (fig.)
  write buffer, 468
  write requests, 467
delayed branches, **443**, 449, A-37
delayed loads, 443
DeMorgan's Law, B-6
DeMorgan's theorems, B-10
design
  cost/performance, 76
  high-performance, 76
  low-cost, 76
  principle. (*see* hardware, design principles)
Devil's DP Dictionary, The, 182
D flip-flop, B-22–B-25, (*see also* flip-flops)
  array of, B-24
  with falling-edge trigger, B-24 (fig.)
  hold time, B-24, B-25 (fig.)
  operation of, B-24 (fig.)
  set-up time, B-24, B-25 (fig.)
Dhrystone benchmark, 73–74, 78
dies, **23**
  costs of, 26
  per wafer, 44
Digital Equipment Corporation. (*see* DEC)
direct mapped block placement, **502**, 503 (fig.)
direct mapped cache, 502
  eight-block, 504 (fig.)
  three address portions for, 508
direct memory access. (*see* DMA)
directories, **635**
disk arrays, 592, (*see also* arrays)
disk controller, **544**
disks, magnetic. (*see* magnetic disks)
dispatch, **336**
  as case statement, 339
  operation, **C-18**
  tables, 336, 342, 343
displacement addressing, 130, E-5
  types of, E-5
div.d instruction, 238, A-67
dividend, **213**
div instruction, 223, A-52
division, 212–224, (*see also* addition; multiplication; remainder; subtraction)
  algorithm, first version, 215 (fig.)
    example, 216 (fig.)
  algorithm, second version, 218 (fig.)
    example, 219 (fig.)
  algorithm, third version, 220 (fig.)
    example, 221 (fig.)
  hardware, first version, 214 (fig.)
  hardware, second version, 217 (fig.)
  hardware, third version, 221
  long division example, 212–213
  nonrestoring, 265
  restoring, **265**
  signed, 222–223

divisor, **213**
Divisor register, 213
div.s instruction, 238, A-68
divu instruction, 223, A-52
D latch, B-22–B-24, (*see also* latches)
DMA, **571**
  for hard disk interface, 573
  memory system and, 574–575
  multiple, devices, 572
  processors, 573
  transfer chains, 574
  transfer steps, 572
  virtual memory and, 574–575
don't cares, 290, **B-15**, (*see also* combinational logic)
  input, B-16
  output, B-16
  types of, B-16
double precision, 227
  integer multiplication, 261
doubles, 227
DRAM, **16**, 455, 473, 475–476, 523–525, B-30–B-33, (*see also* memory; SRAM; virtual memory)
  1-transistor, B-32 (fig.)
  4Mx1, B-33 (fig.)
  32Kx8, B-31 (fig.)
  access time, 455, 519
  board, early, 524 (fig.)
  capacity, 630
  chip interface, 520
  growth in capacity of, 22 (fig.)
  page mode, **476**
  performance, 519 (fig.)
  price per bit, 455
  refreshed charge, B-31
  sizes, 476 (fig.)
  speed, 19
  structure, 475–476
  two-level decoder, B-32
dynamic data, **A-20**
dynamic random access memory. (*see* DRAM)

E

early start, **473**
Eckert, J. Presper, 21, 31
Eckert-Mauchly patent, 32
edge-triggered clocking, **B-18**, B-21 (fig.)
  advantage of, B-20–B-21
edge-triggered timing, B-39–B-41, (*see also* level-sensitive timing)
  clock length, B-40 (fig.)
  clock skew and, **B-40**, B-41 (fig.)
  drawbacks, B-41
EDSAC, 32, 33 (fig.), 353
  mercury delay lines, 522 (fig.)
EDSAC 2, 354
EDVAC, 32
EISPACK, 250, 251
elapsed time, 52
ELXSI 6400, 256
embedded computer, **A-8**
emulation, **355**
encoder, **B-9**
encoding, **C-26**, (*see also* microcode)
  maximum, C-27
  minimum, C-27
Engelbart, Doug, 10, 582
ENIAC, 31 (fig.), 521
  described, 31–32
EPC register, 346, A-33
EPCWrite, 346
eq instruction, 238
error-correcting codes (ECC), B-34
error correction, B-33–B-35
  detection and, B-34
  distance-3 code and, B-34–B-35
  parity code and, B-33
Ethernet, 20, 547, 563, 582, 591
  controllers, 582
evolution-revolution spectrum, **640**, 641 (fig.)
EWcar signal, B-37
EWgreen signal, B-37
EWlite signal, B-37
Exabyte EXB-120, 581 (fig.)
exception handler, 499
exception instructions, A-69–A-70
exception program counter (EPC), 178, 345, 430, 498
  data write port, 346
exceptions, 178, **344**–349, A-30–A-34, (*see also* interrupts)
  causes of, 434
  control hazards and, 430–434
  controls for handling, 432 (fig.)
  detecting, 348–349
  handling of, 345–347
  imprecise, 434
  methods for communicating reasons for, 345
  multiple simultaneous, 434
  overflow, 430
  page fault, 500
  precise, 434
  processing aspects, 347 (fig.)
  as result of arithmetic overflow in add instruction, 433 (fig.)
  system call, 496
  TLB, 500
  types of, 348
execution time, 50, (*see also* performance; response time)
  after improvement, 71
  average instruction, 77
  faster, 69
  MIPS and, 60–61
  normalized, 75
  ratio, 75
  total, 70
  two program example, 69
EX hazard
  forwarding and, 413, 415

stalling and, 402
expansion factor, **A-10**
EX pipe stage, 367
control hazards and, 425
exceptions and, 434
for load instruction, 373 (fig.)
overflow exceptions and, 430
in pipelined control, 382
stalling and, 402, 412
for store instruction, 374–375, 376 (fig.)
exponent, **226**, 227
negative, 228
product calculation, 233

F

fallacies, **26**, (*see also* pitfalls)
cost/performance, 70–76
Fallacy
Amdahl's Law doesn't apply to parallel computers, 638–639
An architecture with flaws cannot be successful, E-21
Computers using antiquated computation model, 26
Geometric mean of ratios tracks total execution time, 75–76
Hardware-independent metrics predict performance, 71–72
If space in control store, new instructions are cost free, 350
Increasing pipelining depth always increases performance, 437
It is possible to design a flawless architecture, E-19
Magnetic storage is on its last legs and will be shortly replaced, 576
More powerful instructions mean higher performance, 147
Peak performance tracks observed performance, 639
Pipelining is easy, 437
Right shift instruction is same as integer division by power of two, 245
Synthetic benchmarks predict performance, 73–74
The architecture that executes fewer instructions is faster, E-21–E-22
false sharing, **612**
fields, 103
names of, 104
file caches, **579**
finite state machine, **322**, **B-35**–B-39
control, 325 (fig.), 332 (fig.)
implementation, C-4–C-15
controller implementation, 333
for controlling memory-reference instructions, 327
with exception detection, 349 (fig.)
implementation of, 322–324, 331
implemented with a state register, B-39 (fig.)
internal storage, B-36 (fig.)
next state determination, 334
next-state function, **B-35**
output function, **B-35**
R-type instructions and, 328 (fig.)
synchronous, B-35
firmware, **352**
flip-flops, **B-22**–B-25, (*see also* latches)
D, **B-22**
with falling-edge trigger, B-24 (fig.)
operation of, B-24 (fig.)
in metastable state, B-43
fllv instruction, A-54
float, 225
floating point, **225–244**, (*see also* arithmetic)
addition, 230–233, 234 (fig.)
arithmetic unit dedicated to, 235 (fig.)
exclusion of, 249–250
hardware, 250
instructions, A-65–A-69
multiplication, 233–240, 239 (fig.)
performance, 633
registers, 240
supported instructions, 238
floating-point accelerator (FPA), 599
floating-point operation, 65
floating vectors, 250
floppy disks, **19**, 542
flushing, **575**
Flynn, Michael, 442–443
for loops, 114, 134, (*see also* loops)
nested, 139–140
format field, C-26, C-27
Forrester, J., 522
Fortran, 8, A-24
overflow and, 178
forwarding, **412**
conditions for, 413–415
control, 413
disadvantage of, 420
example, 416
clock cycles 3 and 4 of, 418 (fig.)
clock cycles 5 and 6 of, 419 (fig.)
with loads and stores, 420–424
modified datapath for, 417 (fig.)
multiplexors, 415
Forwarding Unit, 416
pipeline connections for, 421, 422 (fig.)
frame buffer, **13** (fig.)
front-end machine, 600–601
fully associative block placement, **502**, 503 (fig.)
fully associative cache, 506
blocks in, 510
eight-block, 504 (fig.)
performance, 506
funct field, 104
FutureBus bus, 563, 564

G

gates, **B-7**–B-8
AND, B-7–B-8, B-14
NOR, B-21
cross-coupled, B-22 (fig.)
OR, B-7–B-8, B-14

ge instruction, 238
geometric mean, 75
  formula for, 75
Gibson, D.H., 526
gi (generate), **196**
global pointer, **A-21**, (*see also* pointers)
Goldstine, Herman, 32, 94, 111, 249, 251, 521
Goodyear MPP, 642
  photo, 644 (fig.)
goto statement, 111
Gray code, **B-46**
gray-scale displays, 12–13
gt instruction, 238
guard, **242**, 266

H

halfwords, 103
handshaking, **553**
handshaking protocol, **553**–554
  asynchronous, 554 (fig.)
  implementing, 557 (fig.)
hard disks, **19–20**, 542–546, (*see also* floppy disks; magnetic disks)
  advantages, 542
  densities, 545
  organization of, 543 (fig.)
hardware
  design principles, 96, 97, 104, 125
    good design demands compromise, 104, 149
    make the common case fast, 104, 149, 270
    simplicity favors regularity, 96, 149, 270
    smaller is faster, 97, 149
  operands of, 97–103
  operations of, 95–97
harmonic mean, 87
Harvard architecture, 34
Hazard Detection Unit, **402**, 404 (fig.), 412, 421
  pipeline connections for, 421, 422 (fig.)
hazards. (*see* control hazards; data hazards)
hazard tests, 402–403
  EX hazard, 402, 413, 415
  for forwarding, 413–415
  MEM hazard, 402, 415
  WB hazard, 402–403
helical scan tape, **580**
Hewlett Packard
  HP 9000, 40
  HPPA, 154
  PA-RISC, 518
  Precision architecture, 356
  W. Kahan and, 255
hexadecimal, 175
high-impedance state, B-29
high-level programming languages, **6**, **A-6**
  architecture, 154
  benefits of, 8
Hillis, Danny, 644
hit, **456**, (*see also* miss)
  double, 479
  rate, **457**, 468
  time, **457**
  write, 471
Hoagland, Al, 18
HP 9000, 40

I

IAS (Institute for Advanced Study), 32
IBM, 602
  360/85, 526
  360/91, 442, 444 (fig.)
  370, 567
  650, 250
  701, 34
  3330 disk, 577
  7030, 442
  7090, 254
  7094 series, 250, 251, 252
  entrance into computers, 34
  Personal Computer, 37, 40, 255
  Power PC, 154, 356
  Powerstation 550, 67
  RS/6000, 444, 551
  System/360, 34–35, 36 (fig.), 40, 153, 251–252, 355, 442, 516, 525, 582
    guard digit and, 252
icb instruction, 134, 135
ICL DAP, 642
ID pipe stage, 367, 369
  control hazards and, 425, 426
  exceptions and, 434
  for load instruction, 372 (fig.)
  in pipelined control, 382
  stalling and, 402, 412
  stalling dependent instruction in, 401
  for store instruction, 374
IEEE 754/854 floating-point standard, 256
IEEE 754 floating-point standard, **227**–228
  features, 243–244
  history, 255
  number encodings, 244
  single precision bias, 229
IEEE, 563
IF pipe stage, 367
  control hazards and, 42
  for load instruction, 371–372 (fig.)
  in pipelined control, 382
  stalling and, 402
  for store instruction, 375
if statement, 111
  options, 113 (fig.)
if-then-else statements, 115
Illiac IV, 642
  control unit, 643 (fig.)

immediate addressing, 130, 131 (fig.), E-5, E-7
immediate operand. (*see* I-type instruction format)
implementations, **18**, (*see also* multicycle implementation)
  ROM-based, 352
  simple, 283–312
  single-cycle problems, 306–312
increment-compare-and-branch. (*see* icb instruction)
indexed addressing, E-5
index register, 100
indirect addressing, E-5
input devices, 10
input operation, 549, 551 (fig.)
input/output. (*see* I/O)
instruction architecture, 281
instruction count, **57**, (*see also* instructions)
  measure of, 57
  using one, 60
instruction execution, 314, (*see also* instructions)
  in clock cycles, 314–322
  execution, memory address computation, branch completion step, 319–320
  instruction decode and register fetch step, 318–319
  instruction fetch step, 317–318
  memory access or R-type instruction completion step, 320
  step summary, 321 (fig.)
  write-back step, 320–321
instruction format, 104, (*see also* instructions)
  instruction classes and, 291
  MIPS, A-50–A-70
  observations, 291–292
  types of, 104–105
instruction mix, **77**, 78
Instruction register (IR), 312, 317
instructions, **5**, 94, (*see also specific instructions*)
  arithmetic and logical, A-51–A-55
  average time per, 56
  branch and jump, A-57–A-61
  comparison, 171, A-56–A-57
  constant-manipulating, A-55
  data movement, A-64–A-65
  data transfer, 98
  discarding, 425
  distribution of, on VAX and MIPS, E-22 (fig.)
  exception and interrupt, A-69–A-70
  execution time of, 364
  floating-point, A-65–A-69
  frequency of, 248
    on VAX and MIPS, E-23 (fig.)
  instruction fetch and decode portion of, 326 (fig.)
  length of, 104
  load, A-61–A-62
  memory-reference, 325, 327
  pipelined sequence of, 420 (fig.)
  representing, in computer, 103–110
  restartable, **500**
  stalling, 399, 401
  store, A-63–A-64
  throughput of, 364, 367
instruction set, 94, (*see also* instructions)
  architecture, **17**, 29
    maintaining as constant, 29
  design principles, 149
  history of, 151–154
  MIPS, 95, 247
  pipelining difficulties and, 436
  VAX, 152, 334
IntCause, 346
integers, 171
  double-precision, multiplication, 261
  unsigned, 171, 261
    overflow and, 178
integrated circuits, **13**, 21–26
  manufacturing process, 22
  very large scale, 21
Intel
  80x86, 567
  4004, 35, 38 (fig.)
  Delta, 631
  i860, 73
intelligent peripheral interface (IPI), 563
interconnection network, **630**
  topology, 631
interface message processor (IMP), 547
Intergraph Clipper C100, 73
interleaving, **474–475**
  disadvantages of, 475
International Business Machines. (*see* IBM)
interrupt-driven I/O, **569**, 571
interrupt handler, **A-31–A-32**, (*see also* interrupts)
  example, A-32–A-33
interrupt instructions, A-69–A-70
interrupts, 178, **344**, A-30–A-34, (*see also* exceptions)
  imprecise, 434
  precise, 434
  vectored, **345**
int instruction, 171
I/O bandwidth, 536
I/O benchmarks, (*see also* benchmarks)
  file system, 539
  supercomputer, 538
  transaction processing, 538–539
I/O buses, **549**, 550, (*see also* buses)
I/O controllers, **573**, (*see also* controller)
I/O devices, 539
  behavior, 539
  characteristics, 539–540
  collection of, 534 (fig.)
  connecting, 548–563
  data rate, 540
  diversity, 540 (fig.)
  giving commands to, 566–567
  partner, 539
  priorities of, 570
  types of, 540–548
I/O instructions, **566**, 567, (*see also* instructions)
I/O interrupt, 569, 570, (*see also* interrupts)

I/O performance, 536, (*see also* performance)
  measures, 537–539
I/O processors, **573**
I/O rate, **538**
I/O requests, 536
I/O systems, **534**
  assessing, 535
  characteristics of, 565
  designing, 535, 578
  future directions of, 578–580
  history of, 581–583
  performance, 579
IPI bus, 564
IRWrite, 318
ISAM system, 577
issues, **440**
  multiple, 440
I-type instruction format, 105, (*see also* R-type instructions)

J

jal instruction, 119, A-60
jalr instruction, A-60
j instruction, 112, 270, 283, A-60, (*see also* unconditional jumps)
  addressing, 127–130
  conditional, 121
  instruction format for, 306
  operation of, 282
  single state and, 330 (fig.)
Jobs, Steve, 37
jr instruction, 116, A-61
  in switch statement, 119
J-type instruction format, 127, (*see also* I-type instruction format; R-type instructions)
jump address table, 115
jump-and-link. (*see* jal instruction)
jump instructions, A-57–A-61, (*see also* j instruction)
jump register. (*see* jr instruction)
jumps, unconditional, 149, 150 (fig.)

K

Kahan, W., 246, 253–255
Karnaugh maps, **B-17**
Kelly-Bootle, Stan, 182, 441
kernels, 79
Kernighan, Brian, D-2
Kilburn, T., 525
KSR-1, 636

L

labels, **A-3**, (*see also* assemblers)
  external, A-11
  global, A-11
  local, A-11
  routine with, A-7 (fig.)
  str, A-11
la instruction, A-61
latches, **B-22**–B-25, (*see also* flip-flops)
  alternating use of, B-42
  closed, B-23
  D, **B-22**
    with NOR gates, B-23 (fig.)
    operation, B-23 (fig.)
  S-R, **B-21**, B-22
  transparent, B-23
latency, **439**
lb instruction, A-61
lbu instruction, A-61
ld instruction, A-61
l.d instruction, A-68
LED, 10–12
le instruction, 238
level-sensitive timing, B-41–B-42, (*see also* edge-triggered timing)
  two-phase clocking and, **B-41–B-42**
lexemes, **A-12**
lh instruction, A-61
lhu instruction, A-61
libraries. (*see* subroutine libraries)
light-emitting diode (LED), 10–12
li instruction, A-55
link editor, 117
linkers, **117**, **A-4**, A-17–A-18
  function of, A-18
linking loader, **117**
Linpack, 79, 250, 251
Lisp, 8
  overflow and, 178
Livermore laboratory, 252
Livermore Loops, 79
load byte (lb), 161
loader, **109**
load instruction, 99, A-61–A-62, (*see also* store instruction)
  base register for, 292
  datapath for, 379–380 (fig.)
  in EX pipe stage, 373 (fig.)
  five-step operation of, 296, 302
  forwarding and, 420
  in ID pipe stage, 372 (fig.)
  in IF pipe stage, 371–372 (fig.)
  implementing, 280 (fig.)
  in MEM pipe stage, 373, 374 (fig.)
  operation of, 301 (fig.)
  in WB pipe stage, 373, 375 (fig.)
load-store machine, 152, 159
load upper immediate. (*see* lui instruction)
load word. (*see* lw instruction)
local-area network (LAN), **20**, 547
locality, **454**, (*see also* memory hierarchy)
  principles of, 512
  spacial, **455**, 480

  - temporal, **454**, 456
- lock, **603**
  - steps to acquire, 615 (fig.)
  - variables, **614**
- logic, (*see also* combinational logic)
  - address select, 343
  - block, 288
    - state of, B-4
  - digital, 271
  - elements, 272
    - arrays of, B-17–B-18
    - inputs/outputs of, 275
  - minimization, **C-20**
  - sequential, B-4
    - components, 273
  - two-level, B-10–B-14
- logical instructions, A-51–A-55
- logical operations, 179–182
  - C and MIPS corresponding operations, 182
- long instruction word (LIW), **443**
- loop index, 134
- loops, 112
  - in C programming, D-6–D-4.7
  - for, 114, 139–140
  - unrolling, 448
  - while, 128
- Los Alamos laboratory, 252
- l.s instruction, A-68
- lt instruction, 238
- lui instruction, 125, A-55
  - effects of, 126 (fig.)
- lwcz instruction, A-62
- lw instruction, 99, 270, 279, 283, 365, A-62
- lwl instruction, A-62
- lwr instruction, A-62

M

- machine language, 5, 7 (fig.), 109, **A-3**
  - MIPS, 107, 118, 127, 132
    - code, A-5 (fig.)
    - floating-point, 241
- Macintosh IIs, 256
- McKevit, Jim, C-2
- macros, **A-4**, A-15
  - formal parameter, **A-16**
- magnetic disks, 19–20, **542**–546, 545 (fig.), 576, (*see also* floppy disks; hard disks)
  - arm, 20
  - arrays of, 579–580
  - characteristics, 546 (fig.)
  - floppy, **19**, 542
  - hard, (fig.)19, 542–546
  - off-line, 580
  - on-line, 580
  - types of, 19–20, 542
- main control unit, 291–306, (*see also* control unit)
- main memory, **18**, (*see also* memory)
  - magnetic disks vs., 20
- mapping, memory, **482**
  - fully associative, **485**
  - from virtual to physical address, 483 (fig.)
- Mark I-IV computers, 34
- Maspar MP-1 and MP-2, 642
- massively parallel, **606**
- matrix300, 67
- Mauchly, John, 31, (*see also* Eckert-Mauchly patent)
- Mealy, George, 331
- Mealy machine, 331, B-36
- megaFLOPS. (*see* MFLOPS)
- MEM hazard
  - control for, 415–416
  - forwarding and, 415
  - stalling and, 402
- memory, 4–5, **13**, 14 (fig.), 454–430, (*see also* cache; DRAM; memory hierarchy; SRAM; virtual memory)
  - addresses, 99 (fig.), 100 (fig.)
  - cache only, 636
  - centralized, 620
  - chips, 475
  - core, 522–524
  - data corruption in, B-33
  - distributed, 620
  - division of, A-20–A-21
  - higher, bandwidth, 474 (fig.)
  - inconsistent, 467
  - layout, A-20 (fig.)
  - magnetic core, 34
  - main, **18**, 636
  - mapping, **482**
  - primary, **18**, 481
  - secondary, **18**
  - shared, 620
  - slave, 559
  - stalls, **465**, 478
  - system design, 474
  - transferring data between, 98
  - types of, 18
  - usage, A-19–A-21
  - word in, 98
- memory-data, 320
- memory elements, B-21–B-35, (*see also* memory; memory hierarchy)
  - DRAMs, B-30–B-33
  - error correction, B-33–B-35
  - flip-flops, B-22–B-25
  - latches, B-22–B-25
  - register files, B-25–B-26
  - SRAMs, B-27–B-30
  - unlocked, **B-21**
- memory hierarchy, **455**, (*see also* locality)
  - behavior, 513–514
  - block placement choices, 510
  - designing, 514, 520
  - design parameters, 502 (fig.)

- memory hierarchy *(continued)*
  - framework for, 501–514
  - levels, 456, 457 (fig.)
  - lower level, 456
  - spacial locality and, 458
  - structure, 456 (fig.)
  - temporal locality and, 458
  - upper level, 456
  - write characteristic, 511
- memory-mapped I/O, **566**, A-34–A-36
  - alternative, 567
  - four device registers, A-35 (fig.)
  - Receiver Control register, **A-34**
  - Receiver Data register, **A-35**
  - Transmitter Control register, **A-35**
  - Transmitter Data register, **A-35–A-36**
- memory-memory machine, 152, 159
- memory-stall clock cycles, 476–477, 517, (*see also* clock cycles)
  - read-stall cycles and, 477
  - write-stall cycles and, 477
- MEM pipe stage, 367, 423
  - control hazards and, 429 (fig.)
  - for load instruction, 373, 374 (fig.)
  - in pipelined control, 382
  - stalling and, 402, 412
  - for store instruction, 375, 377 (fig.)
- MemRead signal, 318, 320, 326, 382
- MemtoReg signal, 320, 382
- MemWrite signal, 320, 326, 382
- mercury delay lines, 522
- metastability, **B-43**, B-44 (fig.)
- metrics, 54
  - hardware-independent, 71–72
  - performance, 60–66
  - relating, 54–60
- mfc0 instruction, 178
- mfcl.d instruction, A-65
- mfcz instruction, A-65
- mfhi instruction, 210, 223, A-64
- mfloi instruction, A-64
- mflo instruction, 210, 223
- MFLOPS, **64**, 78
  - formula for, 64
  - normalized, 65
  - peak, 66
  - problems, 65
  - rating, 65–66
  - sustained, 645
- MFP (Machine with Floating Point), 83
- microassembler, 336
- microcode, **C-22**, (*see also* encoding)
  - assembler, C-24
  - controller, 343 (fig.)
  - disadvantage of, 350
  - field translation, C-25 (fig.)
  - horizontal, **C-27**
  - optimizer, 340
  - vertical, **C-27**
- microinstructions, **334**, C-22, (*see also* microcode)
  - addresses of, 335
  - combining, 341
  - execution of, 335–336
  - fields of, 335, 336 (fig.)
    - values, 337 (fig.)
  - format definition, 335–337
  - function of, 338–341
  - narrowing, C-26
- microprocessors
  - first, 35
  - multiple, on single bus, 607
  - SPARC, 627
- microprogram, (*see also* microcode; microprogramming)
  - for control unit, 342 (fig.)
  - creating, 338–344
  - for memory-reference instructions, 339
  - for R-type instructions, 340
  - translating, C-23–C-27
  - translation of, 342
- microprogramming, **322**, **334**, (*see also* microprogram)
  - changes in, 353
  - history of, 353–356
  - in simplifying control design, 333–344
- Microsoft Disk Operating System, 37
- million floating-point operations per second. (*see* MFLOPS)
- million instructions per second. (*see* MIPS (million instructions per second))
- MIMD computer, **596**, 602–603
  - characteristics, 607 (fig.)
  - design, 603
  - network-connected, 619–630
    - characteristics, 622 (fig.)
  - performance, 602
  - programming, 603–607
  - with single bus connection, 607–619
  - styles of, 603
- minicomputer, 35, (*see also specific types of minicomputers*)
- minterm, **C-15**
  - unique, C-15
- MIPS architecture, **60**, 77–78, (*see also* MIPS R2000; MIPS R3000; MIPS R4000)
  - 32-bit ALU for, 189–192
  - addressing modes, 131 (fig.)
  - alternatives, 130–142
  - ALUs and, 183
  - assembler, 110, 115
  - assembly language, 96, 102, 107, 118, 124, 132, 174, 180, 183, 224
    - floating-point, 241
  - datapath for, 286 (fig.)
  - decision-making instructions in, 118
  - floating-point, 241
  - global pointer register, A-21
  - goals of, E-4
  - instruction categories, 150
  - instruction encoding, 105 (fig.), A-50
  - instruction frequency, 248
  - instruction representation in, 107

  instruction set, 95, 247
  logical operations and, 180
  M/2000, 437, E-21–E-22
  machine language, 107, 118, 123, 132
    code, A-5 (fig.)
    floating-point, 241
  memory addresses, 100 (fig.)
  multiplication/division and, 224
  negative numbers and, 174
  operands, 102, 107, 118, 123, 132, 174, 180, 183, 224
    floating-point, 241
  overflow detection, 178
  processors, A-19–A-20
  ratings, 61–64
  registers, 97, 102
    usage convention and, A-23 (fig.), A-26
  relative, 64, 77
  subset implementation, 272, 275–276
  supporting procedures in, 123
  word in, 97, 98
MIPS (million instructions per second)
  1-MIPS machines, 78
  6-MIPS machines, 78
  execution time, 60–61
  formula for, 60
  peak, 63
  problems with, 61
MIPS R2000, 466, 493, 599–600
  addressing modes, A-48
  assembler syntax, A-48–A-50
  assembly language, A-47–A-70
  CPU and FPU, A-47 (fig.)
MIPS R3000, 23 (fig.), 25 (fig.)
MIPS R4000, 23 (fig.), 444, 518, 630
  die, 24 (fig.)
  different versions of, 25 (fig.)
MIPS RISC Architecture, A-37
MISD computer, **596**
miss, **456**, (*see also* hit; miss rate)
  capacity, 513
  collision, 513
  compulsory, 513
  conflict, 513
  decomposition of, 513
  handling, 464–465
  instruction, 464
  penalty, **457**
    latency component, 473
  read, 471
  stall and, 465
  TLB, 491
    handling, 497–501
    processing speed up of, 500
  virtual memory, 482, 484
miss rate, **457**, 468, (*see also* miss)
  block size vs., 472 (fig.)
  with DECStation 3100 cache, 307 (fig.)
  for five cache sizes, 507 (fig.)
  for one/four word block size, 471 (fig.)
  of split cache, 469
  three miss sources and, 514 (fig.)
MNFP (Machine with No Floating Point), 84
modules, **A-4**, A-17
  references, A-4
Moore, Edward, 331
Moore machine, 331, B-36
Moore school, 31–32
motherboard, 14
Motorola
  68881, 255
  68882, 65
mouse, 10, 11 (fig.), 540–542, 582
  anatomy of, 10–12
  moving, 541 (fig.)
mov.d instruction, A-68
move instruction, A-64
mov.s instruction, A-68
MS-DOS, 37
mtcz instruction, A-65
mthi instruction, A-64
mtlo instruction, A-64
MTTF (mean time to failure), 546
mul.d instruction, 238, A-68
mul instruction, A-52
mulo instruction, A-52
mulou instruction, A-53
mul.s instruction, 238, A-68
multicomputers, **643**
multicycle control, 322
multicycle implementation, **312**
  datapath for, 323
mult instruction, 210, A-53
multiple-clock-cycle diagram, 378, 381 (fig.)
  converting from single-clock-cycle diagrams to, 380
multiplexor, **283**, 312, B-8–B-9, (*see also* combinational logic)
  arrayed 32 times, B-19 (fig.)
  controls, 324
  expanding existing, 312
  forwarding, control values, 415 (fig.)
  four-input, 314
  two-input, B-8 (fig.)
multiplicand, **198**
Multiplicand register, 200
  32-bit, 202
multiplication, 198–212, (*see also* addition; division; subtraction)
  algorithm, first, 200 (fig.)
    example, 201 (fig.)
  algorithms, comparing, 209 (fig.)
  algorithm, second, 202 (fig.), 203 (fig.)
    example, 204 (fig.)
  algorithm, third, 206
    example, 207 (fig.)
  of binary floating-point numbers, 237
  floating-point, 233–240, 239 (fig.)
  hardware, first version, 199 (fig.)
  signed, 205, 212

multiplication *(continued)*
of significands, 236
multiplier, **198**
least significant bit of, 200–201
Multiplier register, 199
32-bit, 202
multiprocessors, **643**, (*see also* cache-coherency)
cache-based, 611
cache coherency, 610–612
network-connected, 619 (fig.)
Sequent Symmetry, 610, 616–619
single-bus, 608 (fig.)
with snooping cache coherency, 610 (fig.)
multu instruction, 210, A-53

N

NaN, 244
National Income Tax Service (NITS), 536
National Research and Education Network (NREN), 580
Nave, Rafi, 255
nCube/2, 631
negative numbers, 168–175
neg.d instruction, A-68
neg instruction, A-53
neg.s instruction, A-68
negu instruction, A-53
neq instruction, 238
networks, 546–548, (*see also* network topologies)
advantages of, 20
bidirectional links and, 622
characteristics of, 546
control, **626**
costs, 622
cross bar, 624–625 (fig.)
diagnostics, 627
fully connected, 621, **623**, 624–625 (fig.)
interconnection, **630**, 631
local area, 547
long-haul, **547**
multistage, **623**
packet-switched, 547
switches in, **622**
terminal, 547
network topologies, 621–624
fat tree, **628**
implementing, 624–626
multistage, 625 (fig.)
in parallel processors, 624 (fig.)
next-state function, **322**, B-35, B-38
explicit vs. encoded, C-27
sequencer implementation, C-15–C-22
nonuniform memory access, **619**
machines, **620**
nonvolatile memory technology, 18
nop instruction, 402, 421, A-70
nor instruction, 261, A-53
not instruction, 262, A-53
NOT operator, B-5
NScar signal, B-37
NSgreen signal, B-37
Nslite signal, B-36
NuBus bus, 564
numbers
base, 168
binary, 5, 6, 103
computer, 243
decimal, 103
denormalized, 244
floating-point, 226
as instructions, 5
negative, 168–175
two's complement, 110

O

object file, A-4, A-13 (fig.), (*see also* assemblers)
data segment, A-13
debugging information, A-14
format, A-13–A-14
header, A-13
relocation information, A-13
symbol table, A-13
text segment, A-13
one's compliment representation, 174
opcode, 105, **291**, C-26
operands
constant, 124–125
memory-based, 134
operating systems, **8**
in loading program, A-19
operations
AND, 181–182
complex, 134–135
logical, 179–182
memory-based, 134
OR, 181–182
op field, 104
optical fiber, 580
ori instruction, 182, A-53
or instruction, 270, 276, 283, 365, A-53
OR operation, 181–182, B-5
OR plane, B-12
OS (operating system), 565
functions, 565–566
output devices, 10
output operation, 549
three steps of, 550 (fig.)
output. (*see* I/O)
overflow, 176, (*see also* underflow)
Ada, 178
conditions, 177 (fig.)
divide instructions and, 223
floating point and, 227

  handling, 178
  on Cray computers, 253
  in programming languages, 178
  in subtraction, 176–177
  unsigned integers and, 178

P

packets, **547**
page faults, **482**, 487–490, (*see also* pages; page table; virtual memory)
  cause of, 499
  exceptions, 500
  handling, 497–501
  instruction, 498
page offset, **483**
pages, 484, (*see also* blocks; page faults; virtual memory)
  finding, 485–487
  LRU, 488–489
  placing, 485–487
page table, **485**, (*see also* page fault; pages; virtual memory)
  changing, 497
  indexing of, 486 (fig.), 508
  location, 485
  page mapping in, 488 (fig.)
  reducing size of, 489, 490
  register, **485**
  two, 489
paging, 484
Palmer, John F., 255
PALs, **B-14**
parallel processing program, **602**
  development of, 604
parallel processors, (*see also* processors)
  broken components in, 632–633
  constraints of, 631
  costs, 631
  future directions for, 630–637
  large-scale, addressing in, 634–636, 637 (fig.)
  memory organization in, 620
  message overhead on, 632 (fig.)
  network characteristics, 626 (fig.)
  organization of, 634 (fig.)
  programs for, 634
parameters, 121
parity code, **B-33**
partitions, **630**
PC-relative addressing, E-5, E-7
PC. (*see* program counter (PC))
PCSource, 329, 346
PCSrc control line, 292
PCWrite, 322
PCWriteCond, 322, 329
peak performance, **639**, (*see also* benchmarks; performance)
  for 12 Perfect Club Benchmarks, 639 (fig.)
  for Cray YMP and IPSC/860, 640
Perfect Club, 80, 639 (fig.)
performance, 48, (*see also* benchmarks; peak performance)
  calculating, 51–52
  comparing and summarizing, 68–70
  components of, 58
  cost-performance, 27–28
  CPU, **53**
  defining, 49–52
  equation factors, 57
  importance of, 48
  improving, 52
  measuring, 52–53
  programs for evaluating, 66–68
  ratio example, 67
  system, **53**
peripheral processors, 582, (*see also* processors)
personal computer, 37
physical addresses, **482**
physical page number, **483**
pipelined control, 381–390
  overview, 422 (fig.)
pipelined datapath, 367–381
pipelined dependencies, 397 (fig.), 398 (fig.)
pipelined systems, performance, 435–436
pipeline registers, 400, 412
  before forwarding, 414 (fig.)
  dependencies between, 413 (fig.)
pipelines, (*see also* pipeline stalls; pipelining)
  data hazards. (*see* data hazards)
  figure styles, 378
  five-stage, 366
  graphically representing, 378–381
  impact on branch instruction, 424 (fig.)
  superscalar, 440
pipeline stalls, **399**
  example, 404–405
    clock cycles 2 and 3 of, 406 (fig.)
    clock cycles 4 and 5 of, 407
    clock cycles 6 and 7 of, 408 (fig.)
    clock cycles 8 and 9 of, 409 (fig.)
    clock cycles 10 and 11 of, 410 (fig.)
    clock cycles 12 and 13 of, 411 (fig.)
  insertion of, 403 (fig.)
pipelining, **364**, (*see also* pipelines)
  depth vs. obtained speedup, 437 (fig.)
  difficulties, 436
  performance improvement, 367
  throughput and, 439
  time per instruction and, 431
pipe segment, **364**
pipe stage, **364**, 368 (fig.)
pi (propagate), **196**
Pitfall
  Choosing set associative cache solely for better miss rate, 518
  Comparing computers using only one of performance metrics, 73
  Expecting improvement in one aspect of machine performance to increase performance proportionally, 70–71

- Pitfall (continued)
  - Extending addresses space by adding segments on flat address space, 518
  - Failure to consider instruction set design can impact pipelining, 436
  - Forgetting floating-point addition is not associative, 245
  - Forgetting sequential word addresses do not differ by 1, 148
  - Forgetting to account for byte addressing, 515
  - Ignoring inexorable progress of hardware, 26
  - Measuring parallel processor performance by linear speedup vs. execution time, 537–538
  - Microcode implementing a complex instruction may not be faster, 350
  - Moving functions from CPU to I/O processor performance improvement, 576–578
  - Predicting price-performance more than 5 years, 27
  - Selecting too small an address space, 515–516
  - Using AM of execution times to predict performance, 74
  - Using miss rate as only metric for evaluating memory hierarchy, 516–517
  - Using peak performance to compare machines, 73
  - Using peak transfer rate of a portion of I/O system, 576
  - Write in assembly language to obtain highest performance, 147
- pitfalls, **26**, (*see also* fallacies)
  - cost/performance, 70–76
- PLA, **304**, 305 (fig.), 331, B-10–B-14, B-11, C-14–C-15, (*see also* combinational logic)
  - characteristics, B-12
  - control function implementation, C-14–C-15
  - form of, B-12 (fig.)
  - for implementing logic function, B-13 (fig.)
  - ROMs and, B-15
  - ROM vs, C-27
  - truth tables and, B-12
  - using dots, B-14 (fig.)
- pointers, 143, (*see also* arrays)
  - argument, E-11
  - array indices vs., 146
  - in code to zero arrays, 146
  - frame, E-11
  - global, 121
  - stack, 124, E-11
- polling, **567**
  - disadvantage of, 567
- pop, 120
- prefetching, **521**
- primary memory, **18**
- print_int macro, A-16–A-17
- procedure call frame, **A-24**
- procedure calls, 122, A-22–A-26
  - conventions, **A-21**–A-30
  - example, A-26–A-30
  - LIFO order, A-24
  - in sort procedure, 141
- procedures, 119
  - in C programming, D-7
  - swapping two locations in memory, E-13
- process, **487**, 497
  - active, 487
  - identifier, **497**
  - inactive, 487
  - switch, **497**
  - user, 496
  - virtual address space and, 496
- processor-memory buses, **549**, (*see also* buses)
- processors, 14, 15 (fig.), (*see also* parallel processors)
  - communicating with, 567–570
  - components, 14
  - DMA, 573
  - I/O, 573
  - MIPS, A-19–A-20
  - performance time, 519 (fig.)
  - peripheral, 582
  - shared-memory, **603**
- product, **198**
  - partial, **264**
  - shifting, to right, 202
  - sign, 237
  - unnormalized, 236
- product-of-sums definition, B-10
- Product register, 199, 204
- product term, **C-15**
- program counter (PC), 119, 128, 271, 276, 346
  - relative addressing, **128**, 130, 131 (fig.)
  - value sources for, 321
- program library, A-4
- programmable logic array. (*see* PLA)
- PROMs, B-14
- protection implementation, 496–497
  - hardware and, 496
  - history of, 526
- pseudoinstructions, 110, **A-17**, A-37, A-50–A-51, (*see also* instructions)
- push instruction, 120
- Puzzle, 79

Q

- Quicksort, 79
- quotient, **213**
- Quotient register, 213, 219

R

- RAID (redundant arrays of inexpensive disks), 580
- RAMs, (*see also* ROM)
  - nibble-mode, B-33
  - page-mode, B-33
  - static-column-mode, B-33
- RAS (Row Access Strobe), B-32
- raster refresh buffer, 13
- Raymond, Eric, 150, 168, E-23
- rd field, 104

ReadReq signal, 553
read/write head, **19**
reals, **225**
  scientific notation for, 226
receiver, **A-34**
recursive procedures, A-24
reduced instruction set computer (RISC), 154, 641
references, A-4, (*see also* modules)
  external, A-18
  forward, **A-11**
  relocated, A-18
  unresolved, A-4
refresh
  buffer, 13
  rate, 12
RegDst signal, 382
register addressing, 130, 131 (fig.), E-5
register deferred addressing, E-5
register files, **276**–278, **B-25**–B-27, (*see also* registers)
  with 2 read ports and 1 write port, B-26 (fig.)
  reading, B-25
  two read ports for, B-26 (fig.)
  write port implementation, B-27 (fig.)
  writing, B-25
register-memory architecture, 152
register-register machine, 152
registers, 97, (*see also* pipeline registers; register files; *specific registers*)
  bounds, 489
  contents, passing, 377
  dedicated, 151
  destination, 292
  floating point, 240
  general-purpose, 151, 152 (fig.)
  index, 100
  instruction address, 119
  MIPS, 97, 102
  numbers, passing, 377
  parameter, 121
  special-purpose, 151
  spilling, 102
  Target, 318
register-use conventions. (*see* procedure calls, conventions)
RegWrite signal, 320, 401, C-22
relative speedup, **638**
relocation, **483**
remainder, **213**
  setting sign of, 222
  shifting, to left, 216
Remainder register, 217
rem instruction, A-54
remu instruction, A-54
reproducibility, 68
requested word first, **473**
response time, **50**, 52, (*see also* execution time)
  minimizing, 50–51
return address, 119
rfe instruction, A-69
ring topology, **622**
RISC (reduced instruction set computer), 154, 641
Ritchie, Dennis, D-2
"robo-line" storage, 580
rol instruction, A-54
ROM (read-only memory), 331, **B-14**–B-15, (*see also* RAMs)
  control contents, C-19
  control memory contents, C-22 (fig.)
  control store in, 350
  dispatch, C-19, C-21 (fig.)
    microcode, C-26 (fig.)
  height, **B-15**
  next-state, C-13–C-14
  PLAs vs., B-15
  PLAs vs, C-27
  programmable, B-14
  shape, **B-15**
  truth tables in, C-10–C-14
  width, **B-15**
ror instruction, A-54
rotational delay, **543**
rotational latency, **543**–**544**
round, **242**, 266
rounding, 242
  modes, 243
rs field, 104
rt field, 104
R-type instructions, 104–105, **276**, 291, (*see also* I-type instruction format; J-type instruction format)
  datapath for, 279 (fig.)
  elements for implementation, 278
  execution steps, 294–296
  first step of, 297 (fig.)
  fourth step of, 300 (fig.)
  implemented with finite state machine, 328 (fig.)
  register operands, 278
  second step of, 298 (fig.)
  third step of, 299 (fig.)
Russell, Bertrand, 244

S

sb instruction, A-63
sbn instruction, 163
scaled addressing, E-6
Scarrott, 525
scientific notation, **225**
  normalized, 231
  for reals, 226
SCSI bus, 564
sct.w.d instruction, A-67
sd instruction, A-63
s.d instruction, A-69
secondary memory, **18**, (*see also* memory)
sectors, **542**
seek operation, **542**
seek time, **543**

segmentation, **484**, 489
segment number, 484
segment offset, 484
semiconductor, **21**
separate compilation, A-17
seq instruction, A-56
sequencer, 600–601
  next-state function implementation with, C-15–C-22
Sequent, 602
Sequent Symmetry multiprocessor, 610, 616–619
  microprocessors, 618 (fig.)
  photo, 617 (fig.)
set associative block placement, **502**, 503 (fig.)
  uses, 510
set associative cache, 502, (*see also* cache)
  blocks in, 510
  costs of, 508, 510
  eight-block, 504 (fig.)
  finding block in, 508
  four-way, 508
    implementation, 509 (fig.)
  three address portions for, 508
set on less than. (*see* slt instruction)
sge instruction, A-56
sgeu instruction, A-56
sgt instruction, A-56
sgtu instruction, A-56
shamt field, 104, 181
Shannon, Claude, B-2
SHARE, 251, 252
shift left logical. (*see* sll instruction)
shift right logical. (*see* srl instruction)
shifts, 179
  arithmetic right, 209
sh instruction, A-63
Sieve of Erastosthenes, 79
signals, (*see also specific signals*)
  asserted, B-4
  complements, B-4
  deasserted, B-4
sign and magnitude representation, 174, 226
sign bit, 170
sign extension, 173
sign-extension unit, 280 (fig.)
significand, **226**, 227
  addition of, 231
  multiplication of, 236
silicon, **21**, 29
Silicon Graphics
  Crimson workstation, 520
  microprocessors, 551
SIMD computer, **596**–599
  characteristics, 599 (fig.)
  code for distributing partial sums, 598
  code for network-connected MIMD, 621
  instructions, 597
  for loops and, 597
  vector data operation, 597
  virtues of, 597
SIMMs (Single In-line Memory Modules), 17
  modern 1MB, 524 (fig.)
single-clock-cycle diagram, 378, 379
  for clock cycles 1 and 2, 383 (fig.)
  for clock cycles 3 and 4, 384 (fig.)
  for clock cycles, 5 and 6, 385 (fig.)
  converting to multi-clock-cycle diagram, 380
Single Instruction Computer (SIC), 163
single precision, 227
  exponent bias, 229
  IBM S/360 and, 251
  loading, numbers, 240
single program multiple data. (*see* SPMD)
SISD computer, **596**
sle instruction, A-56
sleu instruction, A-56
sll instruction, 181, A-54
slti instruction, 171, A-57
slt instruction, 114, 115, 171, 270, 276, 283, A-17, A-56
sltiu instruction, 171, A-57
sltu instruction, 171, A-56
small computer system interface (SCSI), 563, 582
snooping, **610**
  bus, 635
  caches, 635
  protocol types, 611
software, portable, 250–251
sort procedure, 139
  C language, 139
  inner loop code, 140–141
  MIPS assembly version, 144
  outer loop code, 139–140
  passing parameters, 141–142
  performance comparison, 148
  preserving registers, 142
  procedure call, 141
  prologue, 142
  register allocation, 139
sort procedure, C programming
  code for body of, E-16–E-18
  complete, E-19
  inner loop, E-17–E-18
  MIPS vs. VAX for, E-20 (fig.)
  outer loop, E-16–E-17
  passing parameters, E-18
  preserving registers, E-18–E-19
  procedure call, E-18
  register allocation for, E-15–E-16
source file, A-4
source language, **A-6**
spatial locality, **455**, 480, (*see also* locality)
  enhancing, 521
SPEC, **79**
  for IBM Powerstation 550, 67
  SPEC89, 80
  SPEC92, 80
SPECFP, 80

SPECINT, 80
SPECMark, 80
speedup, 89
  formula for, 89
speedup. (*see* relative speedup; true speedup)
spice program, 124
SPIM simulator, A-3, A-30–A-34, **A-36**–A-46, (*see also* spim (plain version); xspim)
  byte orders, A-45
  command-line options, A-41–A-42
  computer vs., A-45
  getting started with, A-37–A-41
  instruction format, A-50–A-70
  I/O simulation, A-34–A-36
  MIPS assembler support, A-49
  name, A-36
  speed, A-36
  system call instruction, A-45–A-46
  versions, A-37
  virtual machine and, A-37
  X-window interface. (*see* xspim)
spim (plain version), **A-37**, (*see also* SPIM simulator; xspim)
  terminal interface commands, A-43–A-44
spin waiting, **614**
split transaction protocol, **558**
SPMD, **606**–607
sra instruction, A-54
SRAM, 455, **B-27**–B-30, (*see also* memory; SRAM)
  4x2 structure, B-30 (fig.)
  32Kx8, B-28 (fig.)
  access time, 455
  chip configuration, B-27
  height, **B-27**
  price per bit, 455
  read access time, B-28
  width, **B-27**
  writes, B-28
srav instruction, A-54
srl instruction, 181, A-54
srlv instruction, A-55
s.s instruction, A-69
stack frames, **A-24**, A-29 (fig.), E-11, (*see also* stacks)
  layout, A-24 (fig.)
  set up, A-25
stack model execution, 153, 159
stack pointer, 124
stacks, **119**, 120, (*see also* stack frames; stack pointer; stack segment)
  placing data on, 120
  saving/restoring on, 122
stack segment, **A-20**
stale data problem, **575**
stalls, **465**, (*see also* pipeline stalls)
stalls **465,** execution time, 478
state assignment, **B-38**, **C-21**
state elements, **272**–275, 274 (fig.), (*see also* states, control)
  function, 277 (fig.)
  inputs/outputs, 273
state machine. (*see* finite state machine)
states, control, C-4
  control word and, C-17
static data, **A-20**
static random access memory. (*see* SRAM)
Status register, 345, 567, A-31 (fig.), A-34
  fields, A-30
Stewart, Robert G., 255
Stone, Harold S., 255
store buffer, **512**
store byte (sb) instruction, 161–162
stored-program concept, 108
  principles of, 149
store instruction, 101, A-63–A-64, (*see also* load instruction)
  base register for, 292
  EX pipe stage for, 374–374, 376 (fig.)
  ID pipe stage for, 374
  IF pipe stage for, 374
  implementing, 280
  MEM pipe stage for, 375, 377 (fig.)
  WB pipe stage for, 375, 378 (fig.)
store word instruction. (*see* sw instruction)
Strecker, William, E-2, E-3, E-22
Stretch computer, 442 (fig.)
sub.d instruction, 238, A-69
sub instruction, 178, 270, 276, 283, 365, A-55
subroutine libraries, **8**
  input/output, 8
subroutines. (*see* procedures)
sub.s instruction, 238, A-69
Subtract and Branch instruction. (*see* sbn instruction)
subtraction, 175, (*see also* addition; division; multiplication
  overflow in, 176–177
subu instruction, 178
Sum bit, 187
sum-of-products definition, B-10
Sun
  4/280, 73
  3, 255
  SPARCstation, 27, 518
  SuperSPARC, 444
"superclusters," 636
supercomputers, 35
super-linear performance, **647**, (*see also* performance)
superpipelined machines, **440**
superpipelined pipeline, 440, (*see also* pipelines; pipelining)
  superscalar pipeline vs, 441 (fig.)
superscalar machines, **440**
superscalar pipeline, 440
  superpipelined pipeline vs, 441 (fig.)
Sutherland, Ivan, 12
swap procedure, 135, (*see also* procedures)
  assembly code for, 138 (fig.)
  C language, 136
  code, 136–137
  code for, E-13
  complete, E-14–E-15
  MIPS vs. VAX code for, E-15 (fig.)

swap procedure (continued)
  performance comparison, 148
  preserving registers, 137
  procedure calls and, 137
  register allocation, 136, E-11–E-12
swcz instruction, A-63
sw instruction, 101, 270, 279, 283, 365, A-63
switch statement, 115
  example, 116
swl instruction, A-63
swr instruction, A-63
symbol table, **A-12**
  object file, A-13
synchronization, **603**, (*see also* cache-coherency)
  barrier, **609**
  lock and, 603, 614–615
  message sending and, 603
  using coherency, 614–616
synchronizers, **555**, B-43
  building, B-43
  failure, **B-43**
  metastability and, B-44 (fig.)
synchronous bus, 551–556, **553**, 556, 558
synchronous system, B-20
synthetic benchmarks, **73–74**
  drawbacks, 73–74
syscall instruction, A-45, A-46, A-70
system
  CPU time, **52**
  performance, **53**
System Performance Evaluation Cooperative. (*see* SPEC)
systems software, **8**

T

tags, **460**
  field selection, 461
target language, **A-6**
Target register, 322, 326
task identifier, **497**
Taylor, George S., 256
temporal locality, **454**, (*see also* locality)
  enhancing, 521
  principle of, 456
TeraFLOPS, 644
Thinking Machines, 626–630, 642
Thomas, Lewis, 30
Thornton, James, 381
three-state buffer, B-28, B-29 (fig.)
throughput, **50**
  improving, 50
  pipelining and, 364
ticks, 53
time, **52**
timing methodologies, B-39–B-44
  edge-triggered, B-39–B-41
  level-sensitive, B-41–B-42

TLB, 15 (fig.), **491**, 497, 525
  in DECStation 3100, 493
  entry, 491
  exceptions, 500
  misses, 491, 493
    handling, 497–501
    processing speed up of, 500
  page table and, 492 (fig.)
  typical values, 492
  virtual page number in, 491
Tomasulo's algorithm, 442
topology-specific algorithms, 633
total network bandwidth, **623**
TPC-B, 538
TPS (transactions per second), 538–539
tracks, **542**, 546
traffic light controller, B-38 (fig.)
transaction processing (TP), **538**
transfer time, **544**
transistors, **21**
translation hierarchy, 109
translation-lookaside buffer. (*see* TLB)
transmitter, **A-34**
tri-state buffer, B-28
true speedup, **638**
truncation, 232
truth table, 288, **B-4**–B-5
  control function, 305 (fig.)
  for datapath-control outputs, C-12 (fig.)
  for datapath-control signals, C-6 (fig.)
  full, 290
  for next-state output bits, C-11 (fig.)
  for three ALU control bits, 288 (fig.)
  for three ALU control lines, 289 (fig.)
  without don't cares, B-16–B-17
Tucker, Stewart, 355
Twain, Mark, 5
two-level representation, **B-10**
two's compliment representation, **170**
  name, 174
  shortcuts, 172–173
  two's compliment adder, 188
  in words, 173

U

ulh instruction, A-62
ulhu instruction, A-62
ulps (units in last place), **242**
ulw instruction, A-62
unconditional jumps, 149, 150 (fig.)
undefined instruction exception, 348
underflow, **227**, (*see also* overflow)
  gradual, 244
uniform memory access, **618–619**
UNIVAC I, 34, 35 (fig.), 40
Unix

assemblers for, A-13 (fig.), A-17
tools on, A-17
ush instruction, A-63
usw instruction, A-64

V

"variable allocation," E-11
VAX-11/780, 67–68, 78, E-23–E-24, (*see also* VAX)
microcode, 350
VAX-11 instruction set, E-3
VAX, 153, 516, 518, A-7, E-3–E-24, (*see also* VAX-11/780; VAX 8700; VAX instructions)
8700, 437
addressing modes, E-5–E-6
architecture, 152, E-4, E-22
C compiler, E-14
disappearance of, 355–356
goals of, E-4
Index instruction, 350
instruction set, 152, 334
introduction of, E-23
operands, E-4–E-7
operand specifiers, E-6 (fig.)
operations, E-9
registers, E-4
units of performance (VUPS), 78
VAX 8700, E-21–E-22
VAX and MIPS goals, 4, 5
VAX instructions, (*see also* VAX)
argument pointer, E-11
branch, E-10
branch byte, E-10
branch word, E-10
call and return, E-11
classes of, E-12 (fig.)
compare, E-10
encoding, E-7–E-9, E-8 (fig.)
frame pointer, E-11
move, E-9
move address, E-9
names of, E-9–E-10
for preserving registers, E-14
push, E-10
stack pointer, E-11
vectored interrupts, **345**, (*see also* interrupts)
very long instruction word (VLIW), **443**
virtual addresses, 482 (fig.), (*see also* addresses)
virtually addressed cache, **493**, 495
virtual machine, **A-37**
virtual memory, **481**–501, (*see also* cache; memory; page faults; pages)
design, 483–484
fully associative placement and, 510
invention of, 481
miss, 482, 484
protection implementation with, 496–497
summary, 501
virtual page, (*see also* pages)
mapping, 485
number, **483**
in TLB, 491
VLSI (very large scale integrated circuit), **21**, 351
VME bus, 562–564
VMS, E-24
C compiler, E-19
volatile memory, 18
von Neumann computer, 32
von Neumann, John, 32, 94, 111, 249, 251, 521
"von Neumann syndrome," 640
VUP, 78

W

wafers, 22, 23 (fig.)
costs of, 24
defects of, **22–23**
diced, 23
dies per, **44**
wall-clock time, 52
WB pipe stage, 367, 369
for load instruction, 373, 375 (fig.)
in pipelined control, 382
stalling and, 402, 412
for store instruction, 375, 378 (fig.)
weighted arithmetic mean, 70
Whetstone benchmark, 73–74, 78–79
while loop, 128, (*see also* loops)
Whirlwind project, 34, 522–523
core memory plane, 523 (fig.)
Whitehead, Alfred North, 1
Wilkes, Maurice, 32, 249, 353–354, 355, 452, 525
word, 97, (*see also* halfwords)
accessing, in memory, 98
address, 130
R-format instructions and, 278
workload, **66**, 76
workstation, 10, 11 (fig.)
inside, 14 (fig.)
performance increase of, 27
processor board, 17 (fig.)
wormhole routing, **632**
Wozniak, Steve, 37
write back, **490**, 511, 512
advantages, 490, 511
write-invalidate protocol, **611**
write-through, **467**, 511, 512
advantages, 511
write-update protocol, **611**

X

xori instruction, A-55
xor instruction, 261, A-55
xspim, **A-37–A-38**, (*see also* SPIM simulator; spim (plain version))
  commands, A-44
  control buttons pane, A-38
  data segments pane, A-38
  loading and running program on, A-40
  pop-up window, A-38, A-39 (fig.)
  program debugging and, A-40–A-41
  program terminal reads/writes and, A-40
  register display pane, A-38
  SPIM messages pane, A-38
  starting, A-38
  text segments pane, A-38

Y

yield, 23

Z

Zuse, Konrad, 33–34