

**APPENDIX**

**E**

**CHARACTER CODES AND  
NUMBER CONVERSION**

## E.1 CHARACTER CODES

Information storage and processing in computers involves coding the individual items of information by using several binary variables. Positive and negative numbers are represented in some variation of the binary number system. The most usual formats are presented in Chapter 6, where both integer and floating-point numbers are discussed.

In computers used mainly for business data processing, it is useful to represent and process numbers in the base-10 (decimal) format. Table E.1 gives the most usual coding for individual digits, called the binary-coded decimal (BCD) code. This code is simply the first 10 values (0–9) of the 4-bit binary number system. Strings of these 4-bit code values can be used to represent any desired range of positive and negative integers, with an appropriate code used for the sign position.

Alphabetic characters (A–Z), operators, punctuation symbols, control characters (+ – / , ; LF CR EOT), and numbers must be represented for text storage and editing and for high-level language input, processing, and output operations. Two standard codes for this purpose are the American Standards Committee on Information Interchange (ASCII) code and the Extended Binary Coded Decimal Interchange Code (EBCDIC). The standard ASCII code is a 7-bit code, and the EBCDIC code is an 8-bit code. Tables E.2 and E.3 show the standard ASCII and EBCDIC codes, respectively. The ASCII code is by far the most frequently used.

In many applications, it is preferable to use 8-bit quantities; thus, the basic ASCII code is often extended to 8-bits. A common way of doing this is to set the high-order bit position, bit 7, to zero. Another popular possibility is to use bit 7 as a parity bit for the encoded character.

Some comments about the structure of the ASCII and EBCDIC codes are helpful. Note that in both codes the low-order 4 bits of the decimal character codes (0–9) are the BCD codes of Table E.1. This facilitates two operations. First, two characters that represent decimal digits can be compared to determine which is larger. This can be done with the same type of logic circuits that are used to perform the standard arithmetic

**Table E.1** BCD encoding of decimal digits

Decimal digit	BCD code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

**Table E.2** The 7-bit ASCII code

Bit positions	Bit positions 654								
	3210	000	001	010	011	100	101	110	111
0000	NUL	DLE	SPACE	0	@	P	'	p	
0001	SOH	DC1	!	1	A	Q	a	q	
0010	STX	DC2	"	2	B	R	b	r	
0011	ETX	DC3	#	3	C	S	c	s	
0100	EOT	DC4	\$	4	D	T	d	t	
0101	ENQ	NAK	%	5	E	U	e	u	
0110	ACK	SYN	&	6	F	V	f	v	
0111	BEL	ETB	'	7	G	W	g	w	
1000	BS	CAN	(	8	H	X	h	x	
1001	HT	EM	)	9	I	Y	i	y	
1010	LF	SUB	*	:	J	Z	j	z	
1011	VT	ESC	+	;	K	[	k	{	
1100	FF	FS	,	<	L	/	l		
1101	CR	GS	-	=	M	]	m	}	
1110	SO	RS	.	>	N	^	n	~	
1111	SI	US	/	?	O	—	o	DEL	

NUL	Null/Idle	SI	Shift in
SOH	Start of header	DLE	Data link escape
STX	Start of text	DC1-DC4	Device control
ETX	End of text	NAK	Negative acknowledgment
EOT	End of transmission	SYN	Synchronous idle
ENQ	Enquiry	ETB	End of transmitted block
ACK	Acknowledgment	CAN	Cancel (error in data)
BEL	Audible signal	EM	End of medium
BS	Back space	SUB	Special sequence
HT	Horizontal tab	ESC	Escape
LF	Line feed	FS	File separator
VT	Vertical tab	GS	Group separator
FF	Form feed	RS	Record separator
CR	Carriage return	US	Unit separator
SO	Shift out	DEL	Delete/Idle

Bit positions of code format = 

6	5	4	3	2	1	0
---	---	---	---	---	---	---

**Table E.3** The 8-bit EBCDIC code

Bit positions	Bit positions 7654																																		
	3210	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111																		
0000	NULL	SP	&	-	/																														
0001										a	j	s	A	B	J																				
0010										b	k	t	C	D	K	S																			
0011										c	l	u	E	F	L	T																			
0100										d	m	v	G	H	M	U																			
0101	PF	RES	NL	LF	RS	PN				e	n	w	I		N	V																			
0110	HT	BS	EOB	UC						f	o	x			O	W																			
0111	LC	IL	PRE	EOT						g	p	y			P	X																			
1000	DEL									h	q	z			Q	Y																			
1001										i	r				R	Z																			
1010																																			
1011																																			
1100																																			
1101																																			
1110																																			
1111																																			
NULL	Null/Idle	NL	New line	BS	Backspace	PRE	Prefix	SM	Set mode	HT	Horizontal tab	IL	Idle	PN	Punch on	LC	Lowercase	BYP	Bypass	RS	Reader stop	DEL	Delete	LF	Line feed	UC	Uppercase	RES	Restore	EOB	End of block	EOT	End of transmission	SP	Space

Bit positions of code format = 

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

operations on binary numbers. This is helpful when strings of decimal numbers must be sorted into numerical order. Second, when it is determined by context that consecutive 7- or 8-bit codes in some input string represent a decimal number that is to be stored and processed as a single entity, then it is sometimes practical to remove the leftmost 3 or 4 bits of each digit code and compress the number being represented into a string of 4-bit BCD digits. This compression or packing of data requires starting and ending delimiters, but it is justified in many situations in which storage space requirements are a concern. Similar comments apply to the codes for the alphabetic characters. The fact that their binary bit patterns are in numerical sequence facilitates alphabetic sorting.

## E.2 DECIMAL-TO-BINARY CONVERSION

This section shows how to convert a fixed-point decimal number to its binary equivalent. The value,  $V$ , represented by the binary number

$$B = b_n b_{n-1} \cdots b_0 . b_{-1} b_{-2} \cdots b_{-m}$$

is given by

$$V(B) = b_n \times 2^n + b_{n-1} \times 2^{n-1} + \cdots + b_0 \times 2^0 \\ + b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + \cdots + b_{-m} \times 2^{-m}$$

To convert a fixed-point decimal number into binary, the integer and fraction parts are handled separately. First, the integer part is converted as follows. It is divided by 2. The remainder is the least significant bit of the integer part of the binary representation. The quotient is again divided by 2, and the remainder is the next bit of the binary representation. The process is repeated up to and including the step in which the quotient becomes 0.

Second, the fraction part is converted by multiplying it by 2. The part of the product to the left of the decimal point, which is either 0 or 1, is a bit in the binary representation. The fractional part of the product is again multiplied by 2, generating the next bit of the binary representation. The first bit generated is the bit immediately to the right of the binary point. The next bit generated is the second bit to the right, and so on. The process is repeated until the required accuracy is attained.

Figure E.1 shows an example of conversion from  $(927.45)_{10}$  to binary. Note that conversion of the integer part is always exact, but the binary fraction for an exact decimal fraction may not be exact. For example, the fraction  $(0.45)_{10}$  used in Figure E.1 does not have an exact binary equivalent. This is obvious from the pattern developing in the figure. In such cases, the binary fraction is generated to some desired level of accuracy. In general, the maximum absolute error,  $e$ , in generating a  $k$ -bit fractional representation is bounded as  $e \leq 2^{-k}$ . Of course, some decimal fractions have an exact binary representation. For example,  $(0.25)_{10}$  equals  $(0.01)_2$ .

Convert  $(927.45)_{10}$ 

$$\frac{927}{2} = 463 + \frac{1}{2} \rightarrow 1 \text{ LSB}$$

$$\frac{463}{2} = 231 + \frac{1}{2} \rightarrow 1$$

$$\frac{231}{2} = 115 + \frac{1}{2} \rightarrow 1$$

$$\frac{115}{2} = 57 + \frac{1}{2} \rightarrow 1$$

$$\frac{57}{2} = 28 + \frac{1}{2} \rightarrow 1$$

$$\frac{28}{2} = 14 + \frac{0}{2} \rightarrow 0$$

$$\frac{14}{2} = 7 + \frac{0}{2} \rightarrow 0$$

$$\frac{7}{2} = 3 + \frac{1}{2} \rightarrow 1$$

$$\frac{3}{2} = 1 + \frac{1}{2} \rightarrow 1$$

$$\frac{1}{2} = 0 + \frac{1}{2} \rightarrow 1 \text{ LSB}$$

$$0.45 \times 2 = 0.90 \rightarrow 0 \text{ MSB}$$

$$0.90 \times 2 = 1.80 \rightarrow 1$$

$$0.80 \times 2 = 1.60 \rightarrow 1$$

$$0.60 \times 2 = 1.20 \rightarrow 1$$

$$0.20 \times 2 = 0.40 \rightarrow 0$$

$$0.40 \times 2 = 0.80 \rightarrow 0$$

$$0.80 \times 2 = 1.60 \rightarrow 1 \text{ LSB}$$

$$(927.45)_{10} = (1110011111.0111001 \dots)_2$$

**Figure E.1** Conversion from decimal to binary.

# INDEX

## A

- Accumulator, 40
- Access time
  - magnetic disk, 347
  - main memory, 5
- Adder
  - BCD, 405
  - carry-lookahead, 374
  - circuit, 370
  - full-adder, 368
  - half-adder, 404
  - propagation delay, 371
  - ripple-carry, 368
- Addition, 28, 368
  - carry, 28, 368
  - carry-save, 385
  - end-around carry, 409
  - floating-point, 398
  - generate function, 372
  - modular, 29
  - overflow, 32, 369
  - propagate function, 372
  - sum, 28, 368
- Addition loop. *See* Program examples
- Address, 5, 33
  - aligned/unaligned, 36
  - big-endian, 35
  - little-endian, 35
- Address pointer, 51
- Address space, 33, 205
- Addressing mode, 47
  - absolute, 49
  - ARM, 106
  - autodecrement, 57, 69
  - autoincrement, 57, 69
  - HP3000, 607
  - IA-32, 159, 772
  - immediate, 49
  - index, 52
  - indirect, 50
  - PowerPC, 592
  - register, 49
  - relative, 56
  - 68000, 131
  - 68020, 583
- Advanced Micro Devices (AMD), 591
- Alpha instructions, 596
- Alpha processors
  - 21064, 597
  - 21164, 597
  - 21264, 597
- Alphanumeric characters, 33
  - ASCII, 791
  - EBCDIC, 792
- Altera Excalibur system, 547
- Amdahl's law, 654
- Analog to digital (A/D) conversion, 515
- Annul bit, 491
- Apple computers, 591, 594
- Arbitration, 237
  - centralized, 237
  - distributed, 239
    - on SCSI bus, 270
- Architecture, 2
- Arithmetic and logic unit (ALU), 5
- ARM
  - addressing, 106
  - architecture versions, 579
  - assembly language, 118
  - condition codes, 105, 117, 736
  - hard macrocell core, 580
  - input/output (I/O), 121
  - instructions, 113, 734–750
    - conditional execution, 106
    - operand shifting, 114, 737
    - pseudo-instructions, 120
    - thumb, 579
  - interrupts, 224
  - programming experiments, 750
  - registers, 105
  - synthesizable core, 580
- ARM CPU cores
  - ARM720T, 581
  - 920T, 581
  - 1020T, 581
  - StrongARM SA-110, 581

- ARM processor cores
    - ARM7TDMI, 580
    - ARM9TDMI, 581
    - ARM10TDMI, 581
  - Array processor, 620
  - ASCII (American Standards Committee on Information Exchange) code, 4, 791
  - Assembler, 58
    - two-pass, 63
  - Assembler directives (commands), 60
  - Assembly language, 19, 58
    - ARM, 118
    - generic, 58
    - IA-32, 170
    - mnemonics, 58
    - notation, 19, 38
    - 68000, 140
    - syntax, 58
  - Associative search, 318
  - Asynchronous bus. *See* Bus
  - Asynchronous DRAM, 299
  - Asynchronous transmission, 566
  - Athlon processor, 591
  - Autovector, 214
- B**
- Bandwidth. *See also* Bit rate: Baud rate
    - communication, 565, 567, 569, 570
    - memory, 304
  - Barrier synchronization, 650
  - Base register, 55
  - Baseband, 564
  - Baud rate, 564. *See also* Bit rate
  - BCD. *See* Binary-coded decimal
  - Benchmark program, 17, 656
  - Big-endian, 35
  - Binary variable, 662
  - Binary-coded decimal (BCD), 4, 790
    - addition, 405
    - packed, 83
  - Bit, 4, 27
  - Bit map, 559
  - Bit rate, 564. *See also* Baud rate
  - Bit-ORing, 437
  - Booth algorithm, 380
    - bit-pair recoding, 384
    - skipping over 1s, 382
  - Booting, 350
  - Branch, 46
    - delay slot, 470, 491
    - delayed, 470
    - folding, 468
    - instruction, 45
    - penalty, 466
    - prediction, 472, 491
    - target, 46
  - Branching, 44
    - in ARM, 116
    - conditions, 46
    - in IA-32, 171
    - in microprogram, 437
    - in 68000, 141
  - Breakpoint, 220
  - Bridge, 259, 260, 262
  - Broadband, 564
  - Broadcast, 633
  - Buffer
    - circular, 71, 531
    - register, 10
  - Bus, 9. *See also* Bus standards
    - arbitration, 237
    - asynchronous, 244
    - cycle, 241
    - master, 237
    - propagation delay, 241
    - scheduling. *See* Arbitration
    - single-bus system, 10, 624
    - skew, 245
    - synchronous, 241, 264
    - timing, 242
  - Bus standards, 259
    - ISA, 260
    - PCI, 261
    - SCSI, 266
    - USB, 272
    - X3.131 (SCSI), 266
  - Byte, 33
  - Byte addressable, 35
- C**
- Cache memory, 5, 13, 294, 313–329
    - in ARM710T, 326
    - associative mapping, 318
    - block, 315
    - coherence, 321, 641
    - direct mapping, 317
    - dirty bit, 316, 326
    - hit, 316
    - hit rate, 332
    - levels, 313, 335
    - line, 315
    - load through, 316
    - lockup-free, 337
    - mapping function, 316
    - miss, 316
    - miss penalty, 332
    - miss rate, 332
    - in Pentium III, 326



- in Pentium 4, 329
  - replacement algorithm, 316, 321
  - set-associative mapping, 318
  - in 68040, 325
  - snoopy controls, 643
  - tag, 318
  - valid bit, 319, 326
  - write back, 316, 642
  - write buffer, 335
  - write through, 316, 642
  - Carry, 28, 368
    - flag. *See* Condition codes
  - Cartridge tape, 359
  - Cathode-ray tube (CRT), 558
  - CC-NUMA systems, 645
  - CD-ROM, 5, 355
  - Character codes, 33
    - ASCH, 791
    - EBCDIC, 792
  - Character string, 36
  - Charge-coupled device (CCD), 557
  - Circuit switching, 635
  - Circular buffer (queue), 533
  - CISC (Complex Instruction Set Computer), 17, 97
  - Clock, 14
    - cycle, 14
    - rate, 16
  - Clock recovery, 565
  - Coherence. *See* Cache memory
  - Combinational circuits, 691
  - Compiler, 4, 11
    - optimizing, 17
  - Complement, 665
  - Complementary metal-oxide semiconductor (CMOS), 681
  - Complex Instruction Set Computer. *See* CISC
  - Complex programmable logic device (CPLD), 711
  - Computer, 2
  - Computer-aided design (CAD), 677
  - Concurrency, 15
  - Condition code register, 46
  - Condition codes, 46, 84
    - ARM, 105, 117, 736
    - IA-32, 156, 171, 783
    - in pipelined processor, 465, 478
    - side effect, 478
    - 68000, 141, 768
  - Conditional branch, 46
  - Context switch, 223
  - Control store, 430
  - Control unit, 6
  - Control word, 430
  - Coprocessor
    - arithmetic (80386), 588
    - ARM instructions, 750
  - Counter
    - ripple, 702
    - synchronous, 703, 714
  - Critical section, 640
  - Crossbar, 625
  - Cycle stealing, 237
  - Cyclic redundancy check (CRC), 279
- D**
- Daisy chain, 216
  - Data, 4
  - Data communication equipment, 564
  - Data striping, 351
  - Data terminal equipment, 564
  - Data types
    - bit, 27
    - byte, 33
    - character, 33
    - floating-point, 394
    - fraction, 394
    - integer, 27
    - string, 36
    - word, 33
  - Datapath, 414
  - De Morgan's rule, 670
  - Debouncing, 249
  - Debugging, 63, 219
  - Decoder, 703
  - Demodulation, 564
  - Desktop computer, 2
  - Device driver, 223
  - Differential signaling, 308
  - Digit packing. *See* Program examples
  - Digital camera, 514
  - Digital subscriber loop (DSL), 569
  - Direct memory access. *See* DMA
  - Directory-based cache coherence, 643
  - Dirty bit. *See* Cache memory
  - Disk. *See* Magnetic disk
  - Disk arrays, 351
  - Dispatch operation, 467, 486
  - Dispatch unit. *See* Instruction
  - Display buffer, 559
  - Displays, 558
  - Distributed computer system, 618
  - Distributed computing, 21
  - Distributed memory system, 623
  - Division, 390
    - floating-point, 398
    - nonrestoring, 391
    - restoring, 391
  - DMA (Direct Memory Access), 235
    - block (burst) mode, 237
    - controller, 235

Don't-care condition, 674  
 Dot product. *See* Program examples  
 DVD, 357  
 Dynamic memory (DRAM), 299

**E**

EBCDIC (Extended Binary Coded Decimal Interchange Code), 4, 792  
 Echoback, 207  
 Edge-triggered flip-flops, 694  
 Effective address, 50  
 Effective throughput, 624  
 Elapsed time, 13  
 Embedded processor, 517  
 Embedded system, 512  
 Emulation, 445  
 Enterprise system, 2  
 Ethernet, 646  
 Exception. *See also* Interrupt  
   floating-point, 397  
   imprecise, 484  
   precise, 485  
 Execution phase, 413  
 Eye pattern, 568

**F**

Fan-in, 687  
 Fan-out, 687  
 Fat-tree network, 631  
 Fetch phase, 413  
 Field-programmable gate array (FPGA), 547, 712  
 FIFO (first-in, first-out) queue, 71, 531  
 File, 11  
 File mark, 358  
 File server, 2  
 Finite state machine, 719  
 Flash memory, 312  
 Flat panel display, 560  
 Flip-flops, 690–699  
   D, 694–697  
   edge-triggered, 694  
   gated latch, 690  
   JK, 697  
   latch, 690  
   master-slave, 694  
   SR latch, 690  
   T, 697  
 Flit, 634  
 Floating point, 393  
   addition-subtraction unit, 400  
   arithmetic operations, 398  
   chopping, 399  
   coprocessor, 588, 750

double precision, 396  
 exception, 397  
 exponent, 394  
   excess-*x* representation, 396  
 format, 395  
 guard bits, 399  
 IEEE standard, 394  
 mantissa, 394  
 normalization, 394  
 overflow, 396  
 representation, 395  
 rounding, 399  
 scale factor, 394  
 significant digits, 394  
 single precision, 396  
 special values, 397  
   denormal, 397  
   infinity, 397  
   Not a Number (NaN), 397  
 sticky bit, 400  
 truncation, 399  
   biased/unbiased, 399  
   underflow, 396  
 Floppy disk, 350  
 Four-wire link, 565  
 Frame pointer, 77  
 Frequency-shift keying (FSK), 564  
 Full-duplex (FDX) link, 565  
 Fully-connected network, 625

**G**

Gated latch, 690  
 General-purpose register, 8  
 Global address space, 637  
 GPU (Graphics Processing Unit), 561  
 Graphics  
   accelerator, 561  
   port, 562  
   processing unit, 561  
 Gray code, 726

**H**

Half-duplex (HDX) link, 565  
 Handshake, 244  
 Hardware, 2  
 Hardwired control, 425  
 Hertz, 14  
 Hexadecimal. *See* Number representation  
 High-level language, 4, 11, 26  
   C, C++, 11, 88, 527  
 History of computers, 19  
 Hold time, 697  
 HP3000  
   addressing, 607

- instructions, 606
  - registers in stack, 610
  - stack structure, 604
  - Hybrid, 566
  - Hypercube network, 628
- I**
- IBM, 591, 592
  - IEEE standards. *See* Standards
  - ILLIAC-IV, 621
  - Index register, 52
  - Input unit, 4
  - Input/output (I/O)
    - address space, 205
    - instructions in IA-32, 175
    - interface circuit, 206, 248, 259
    - memory-mapped, 66, 204
    - in operating system, 220
    - program-controlled, 64, 207
    - register, 206
    - status flag, 67, 206
    - unit, 3
  - Input/output devices. *See* Flat panel display;
    - Intellimouse; Joystick; Keyboard; Mouse;
    - Printer; Trackball; Video display
  - Input/output port, 248
    - bidirectional, 254
    - parallel, 248, 518
    - serial, 257, 521
  - Instruction, 3, 37
    - commitment, 485
    - completion queue, 485
    - encoding, 94
    - execution phase, 413
    - dispatch, 467, 486
    - fetch phase, 413
    - fields, 62, 95
    - grouping, 496
    - hazards, 504
    - operands, 39
    - privileged, 220, 343
    - queue, 467
    - reordering, 471
    - retired, 485
    - side effects, 478
    - synthetic, 489
  - Instruction encoding, 94
  - Instruction format
    - ARM, 106, 735
    - HP3000, 606
    - IA-32, 168, 770, 771
    - IA-64, 598
    - one-address, 40
    - 68000, 137
    - three-address, 39
    - two-address, 39
    - zero-address, 42
  - Instruction register (IR), 7, 43, 412
  - Instruction set architecture (ISA), 26
  - Instruction unit, 3
  - Instructions
    - arithmetic, 7, 38, 70
    - branch, 45
    - data transfer, 38
    - input/output, 66
    - logic, 81
    - shift and rotate, 82
    - subroutine, 72
  - Integrated circuit (IC), 16, 20, 688
  - Intel IA-32 Pentium
    - addressing modes, 159, 772
    - assembly language, 170
    - condition codes, 156, 171, 783
    - input/output (I/O), 174
      - block transfers, 176
      - isolated, 175
      - memory-mapped, 174
    - instructions, 164, 171, 182, 773–784
      - encoding, 770, 771
      - format, 168, 770, 771
      - multimedia (MMX), 183
      - string, 176, 783
      - vector SIMD (SSE), 184
    - interrupts, 231–234
    - memory segmentation, 586
      - protected mode, 587
      - real mode, 586
      - segment registers 156, 586
    - programming experiments, 785
    - register structure, 156
    - sixteen-bit mode, 588, 785
    - subroutines, 177
      - stack frame, 179
  - Intel IA-64
    - instructions, 598
      - bundles, 598
      - conditional execution, 598
    - Itanium processor, 602
    - register rotation, 602
    - register stack, 600
    - register windows, 602
    - speculative loads, 600
  - Intel processors
    - 8051, 542
    - 8086, 585
    - 8088, 585
    - 80286, 585
    - 80386, 588
    - 80486, 588

Intel processors—*Cont.*

- Itanium, 602
- Pentium, 589
- Pentium II, 590
- Pentium III, 590
- Pentium 4, 590
- Pentium Pro, 589

## Intellectual property (IP), 547

## Intellimouse, 555

## Interconnection network, 622–636

- crossbar, 625
- fat tree, 631
- hypercube, 628
- mesh, 630
- multistage, 626
- ring, 631
- shuffle network, 627
- single-bus, 624
- tree, 630
- torus, 630

## Interleaving, 330

## Internet, 2

Interrupt, 9. *See also* Exception

- acknowledge, 214
- in ARM, 224–229
- disabling, 212
- edge-triggered, 229, 212
- enabling, 212
- in IA-32, 231–234
- latency, 210
- mask, 224
- nesting, 213
- nonmaskable, 229
- in operating systems, 220
- priority, 215
- service routine, 9
- in 68000, 229–231
- software, 220
- vectored, 214

I/O. *See* Input/output

## IR (Instruction register), 7, 43, 412

## ISA (Instruction Set Architecture), 26

## Isochronous, 273, 281

## Itanium processor, 602

**J**

## Joystick, 556

## JTAG port, 712

**K**

## Karnaugh map, 671

## Kernel (Supervisor) mode, 343

## Keyboard, 554

**L**

## Laser printer, 560

## Latch, 690

## LIFO (last-in, first-out), 68

## Link register, 72

## Linked list, 90

insertion/deletion. *See* Program examples

## Liquid crystal display, 559

## Little-endian, 35

Load through. *See* Cache memory

## Loader, 63

## Load-store multiple operands

ARM, 112

IA-32, 783

68000, 146

## Local area networks (LAN), 646

Ethernet, 646

token ring, 647

## Locality of reference, 315

## Lock, 640

## Logic circuits, 662–724

## Logic families

CMOS, 681

## Logic functions, 662–677

AND, 664

EXCLUSIVE-OR (XOR), 664

minimization, 668

NAND, 674

NOR, 674

NOT, 665

OR, 662

synthesis, 664

## Logic gates, 662

fan-in, 687

fan-out, 687

noise margin, 686

propagation delay, 686

threshold, 678, 684

transfer characteristic, 684

transition time, 686

## Logical address, 294, 338

## Long-haul networks, 646

## Loop

with branch prediction, 473, 492

with delayed branch, 470, 492

## Loosely coupled multicomputer, 645

## LRU (least-recently used) replacement, 321

**M**

## Machine instruction, 3, 94

## Machine language, 4, 19, 26

## Magnetic disk, 5, 344–352

access time, 347

controller, 348

- data buffer/cache, 348
- data encoding, 344
- drive, 346
- floppy disk, 350
- latency, 347
- organization, 346
- rotational delay, 347
- seek time, 347
- Winchester, 345
- Magnetic tape, 358
  - cartridge, 359
  - format, 358
- Mailbox memory, 659
- Mainframe, 2
- Manchester code, 344
- Master-slave. *See* Flip-flops
- Mechanical computing devices, 19
- Memory, 4
  - access time, 5, 294
  - address, 5, 33
  - address register (MAR), 8, 293
  - address space, 292
  - asynchronous DRAM, 299
  - bandwidth, 304
  - bit line, 295
  - byte addressable, 35
  - cache. *See* Cache memory
  - cell, 298, 310
  - controller, 307
  - cycle time, 294
  - data register (MDR), 8, 293
  - DDR SDRAM, 304
  - DIMM, 306
  - dynamic (DRAM), 299
  - fast page mode, 301
  - hierarchy, 5, 313
  - interleaving, 330
  - latency, 304
  - main, 5, 13, 313
  - multiple module, 330
  - Rambus, 308
  - random-access memory (RAM), 5, 294
  - read cycle, 9
  - read-only memory (ROM), 310
  - refreshing, 301, 308
  - RIMM, 309
  - SIMM, 306
  - static (SRAM), 297, 305
  - synchronous DRAM (SDRAM), 302
  - unit, 4
  - word, 5, 33
  - word length, 33, 292
  - word line, 295
  - write cycle, 9
- Memory management unit (MMU), 294, 339
  - in ARM, 581
  - in 680X0, 584
  - in 80X86, 586
- Memory pages, 133
- Memory segmentation, 586
- Memory-mapped I/O, 66, 204
- Mesh network, 630
- Message-passing, 19, 645, 651
  - protocol, 623
- MFLOPS measure, 656
- Microcontroller, 512, 518–521, 541
  - ARM, 543
  - Intel, 542
  - Motorola, 542
- Microinstruction, 430
  - fields, 433
  - horizontal, 434
  - sequencing, 437, 440
  - vertical, 434
- Microoperation, 433
- Microprogram counter, 430
- Microprogram memory. *See* Control store
- Microprogrammed control, 429
- Microroutine, 430
- Microwave oven, 512
- MIMD system, 620
- MIPS measure, 656
- MISD system, 620
- Miss. *See* Cache memory
- Mnemonic, 58
- Modem, 564
  - cable, 569
- Modulation, 564
- Motherboard, 259, 305
- Motorola processors
  - ColdFire family, 585
  - ColdFire MCF5xxx, 543
  - 68000, 130
  - 68020, 582
  - 68030, 584
  - 68040, 584
  - 68060, 585
  - 68HC11, 542
  - 683xx, 543
- Motorola 68000, 130
  - addressing modes, 131, 752
  - assembly language, 140
  - condition codes, 141, 768
  - input/output (I/O), 145
  - instruction set, 136, 141, 151, 754–767
  - interrupts, 229–231
  - register structure, 131
  - subroutines, 146
- Mouse, 555

Multicast, 633  
 Multicomputer, 18, 638, 645  
 Multiple issue, 482  
 Multiplexer, 705  
 Multiplication, 376  
   array implementation, 376  
   Booth algorithm, 380  
   carry-save addition, 385  
   fast, 383  
   floating-point, 398  
   sequential implementation, 378  
   signed-operand, 380  
 Multiprocessor, 18, 618, 622  
   cache coherence, 641  
   caches, 637  
   global memory, 623  
   interconnection network, 624–636  
   local memory, 623  
   private address space, 638  
   program parallelism, 638  
   shared memory, 18, 637  
   shared variables, 640  
   speedup, 653  
 Multiprogramming, 12  
 Multistage network, 626  
 Multitasking, 12, 221

**N**

Network of workstations, 647  
 Noise, 686  
 Noise margin, 686  
 Nonblocking switch, 626  
 Notebook computer, 2  
 NUMA multiprocessors, 623  
 Number conversion, 793  
 Number representation  
   binary positional notation, 27  
   floating-point, 394  
   hexadecimal, 64  
   1's-complement, 27  
   sign and magnitude, 27  
   signed integer, 28  
   ternary, 404  
   2's-complement, 27

**O**

Object program, 4, 58  
 1's-complement representation, 27  
 OP code, 59, 95  
   in ARM, 106  
   in IA-32, 168  
   in 68000, 135  
 Open-drain, 211  
 OpenGL, 563

Operand forwarding, 462  
 Operands, 39  
 Operating system, 11  
   interrupts in, 220  
   multitasking, 221  
   process, 221  
   scheduling, 221  
 Optical disk, 5, 352–358  
   CD-Recordable, 356  
   CD-Rewritable, 356  
   CD-ROM, 355  
   DVD, 357  
 Output unit, 6  
 Overflow, 32, 84  
   flag. *See* Condition codes

**P**

Packet, 624  
 Page, 339  
 Parallel I/O port, 248, 518  
 Parallel processing, 619  
 Parameter passing, 74  
   by reference, 75  
   by value, 75  
 Parity, 567  
 PC. *See* Program counter  
 Pentium. *See* Intel processors  
 Performance, 13  
   basic equation, 14  
   execution time, 13  
   measurement, 17, 656  
   memory, 329  
   multiprocessors, 653  
   pipeline, 503–505  
   processor, 13  
 Peripheral device, 554  
 Personal computer, 2  
 Phase encoding, 344  
 Phase-shift keying (PSK), 564  
 Physical address, 338  
 Pipelining, 15, 454  
   addressing modes, 465, 476  
   basic concepts, 454  
   branching, 466  
   bubbles, 460  
   condition codes, 465, 478  
   data hazards, 459, 461  
   instruction hazards, 459, 465  
   performance, 503  
   stalling, 459  
   structural hazard, 460  
 Pixel, 558  
 Plasma display, 560  
 Plug-and-play, 261, 265, 274

- Pointer register, 51
  - Polling, 207, 213
  - Pop operation. *See* Stack
  - Port. *See* Input/output; Register
  - PowerPC
    - addressing modes, 592
    - instructions, 592
    - registers, 591
  - PowerPC processors
    - MPC7450, 594
    - AltiVec units, 594
    - 601/603/604/620, 593
  - Prefetching, 336
  - Primary storage, 5
  - Printer, 6, 560
  - Priority, 215. *See also* Arbitration
  - Private memory, 623
  - Privileged instruction, 220, 343
  - Process. *See* Operating system
  - Processor, 3
  - Processor stack, 73
  - Processor status word (PSW), 46
    - in ARM, 224
    - in IA-32, 232
    - in 68000, 230
  - Processor time, 13
  - Program, 4
  - Program counter (PC), 8, 43, 412
  - Program examples
    - addition loop
      - ARM, 118
      - generic, 45, 51, 57
      - IA-32, 166
      - 68000, 143
    - character transfer, 525
    - circular buffer, 531
    - digit packing
      - ARM, 116
      - generic, 82
      - IA-32, 174
      - 68000, 151
    - dot product
      - ARM, 126
      - generic, 86
      - IA-32, 184
      - 68000, 152
    - message-passing multiprocessor, 651
    - shared-memory multiprocessor, 648
  - input/output
    - ARM, 121
    - generic, 67, 207
    - IA-32, 175
    - 68000, 145
  - interrupt service routine
    - ARM, 230
    - generic, 219
    - IA-32, 234
    - 68000, 232
  - linked-list insertion/deletion
    - ARM, 127
    - generic, 92, 93
    - IA-32, 185
    - 68000, 154
  - reaction timer, 537
  - sorting
    - ARM, 127
    - generic, 87
    - IA-32, 185
    - 68000, 153
  - subroutines
    - ARM, 122
    - generic, 74, 76, 79
    - IA-32, 177
    - 68000, 146
  - Program parallelism, 638
    - shared variables, 640
  - Program state, 221
  - Program-controlled I/O. *See* Input/output
  - Programmable array logic (PAL), 710
  - Programmable logic array (PLA), 707
  - Programming experiments
    - in ARM, 750
    - in IA-32, 785
  - Propagation delay
    - on bus, 241
    - in logic circuit, 686
  - Protected mode, 587
  - Protection, 343
  - Push operation. *See* Stack
- Q**
- Quadrature amplitude modulation (QAM), 564
  - Queue, 71. *See also* Instructions
- R**
- RAID disk systems, 351
  - Rambus memory, 308
  - Random-access memory (RAM), 5
  - Raster scan, 558
  - Reaction timer, 635
  - Reactive system, 541
  - Read-modify-write, 640
  - Read-only memory (ROM), 310
    - electrically erasable (EEPROM), 311
    - erasable (EPROM), 311
    - flash, 312
    - programmable (PROM), 311
  - Real mode, 586
  - Real-time processing, 210, 535

- Reduced Instruction Set Computer. *See* RISC
- Refreshing
  - in displays, 559
  - in memories, 301, 308
- Register, 6, 699
  - base, 55
  - general-purpose, 8
  - index, 52
  - port, 423
  - renaming, 485, 498
  - windows, 488
- Register transfer notation (RTN), 37
- Rendering, 562
- Reorder buffer, 485
- Replacement algorithm, 321
- Ring network, 631
- RISC (Reduced Instruction Set Computer), 17, 97, 578
- Rounding. *See* Floating point
- RS-232-C, 571
  
- S**
- Scalability, 633
- Scaler, 702
- Scanner, 557
- Scheduling. *See* Arbitration; Operating system
- SCI Standard, 644
- SCSI bus. *See* Bus standards
- Secondary cache, 314
- Secondary storage, 5, 344–359
- Segmentation. *See* Memory segmentation
- Sequential circuits, 691, 714–723
  - finite state machine, 719
  - state assignment, 716
  - state diagram, 715
  - state table, 716
  - synchronous, 718
- Serial I/O interface, 257, 521
- Serial transmission, 563
- Server, 2, 512
- Setup time, 242, 697
- Seven-segment display, 704
- Shared memory, 18, 637, 648
- Shared variables, 640
  - busy-waiting, 658
  - critical section, 640
  - locks, 640
  - read-modify write access, 640
  - Test-and-Set instruction, 640
- Shift register, 700
- Shuffle network, 627
- Side effects. *See* Instructions
- Sign bit, 27, 34
- Sign extension, 32
- SIMD system, 620
- Simplex link, 565
- Simulated annealing, 655
- SISD system, 619
- Skew. *See* Bus
- Small-scale integration (SSI), 689
- Snoopy cache, 643
- Soft processor core, 547
- Software interrupts. *See* Interrupt
- Sorting. *See* Program examples
- Source program, 4, 58
- SP. *See* Stack, pointer
- SPARC. *See* Sun Microsystems processors
- SPMD application, 651
- SPEC rating, 18
- Speedup, 653
- Split-transaction protocol, 624
- Stack, 68
  - in ARM, 122
  - frame
    - ARM, 123
    - generic, 75
    - IA-32, 179
    - IA-64, 600
    - 68000, 146
  - frame pointer (FP), 77
  - in HP3000, 604
  - in IA-32, 177
  - in IA-64,
    - pointer (SP), 68
  - processor. *See* HP3000
  - push and pop operations, 68
  - pushdown, 68
  - in 68000, 146
  - in subroutines, 73
- Standards. *See also* Bus standards
  - IEEE floating-point, 395
  - IEEE-802, 646
  - RS-232-C, 571
  - SCI, 644
- Start-stop format, 566
- State diagram, 715
- State table, 716
- Static memory (SRAM), 297, 305
- Status flag. *See* Input/output
- Status register. *See* Input/output
- Store-and-forward network, 634
- Stored program, 4
- Strobe, 241
- Subroutine, 72
  - in ARM, 122
  - in IA-32, 177
  - in IA-64, 600
  - linkage, 72
  - nesting, 73, 78



- parameter passing, 74
    - in 68000, 146
  - Subtraction, 29
    - floating-point, 398
  - Sum-of-products form, 667
  - Sun Microsystems processors
    - microSPARC, 595
    - SPARC, 594
    - UltraSPARC I, II, III, 493, 595
      - visual instruction set (VIS), 595
  - Supercomputer, 2, 618
  - Superscalar processor, 15, 481
  - Supervisor mode (state), 215, 221, 343
  - Symbol table, 63
  - Symmetric multiprocessor, 636
  - Synchronous DRAM (SDRAM), 302
  - Synchronous sequential circuit, 718
  - Synchronous transmission, 568
  - Syntax. *See* Assembler
  - System Performance Evaluation Corporation (SPEC), 18
  - System software, 10. *See also* Operating system
  - System space, 343
  - System-on-a-chip, 546
- T**
- Testability, 546
  - Test-and-Set instruction, 640
  - Texel, 562
  - Text editor, 11
  - TFT (Thin-film transistor) display, 560
  - Thread, 650
  - Three-state. *See* Tri-state
  - Threshold, 678, 684
  - Tightly-coupled multiprocessor, 645
  - Time slicing, 221
  - Timers, 524
  - Timing signals, 6
  - Token-ring network, 647
  - Torus network, 630
  - Touchpad, 557
  - Trace exception, 220
  - Trackball, 556
  - Transition time, 686
  - Translation lookaside buffer (TLB), 341
- Transmission
    - asynchronous, 566
    - simplex and duplex, 565
    - synchronous, 568
  - Trap, 220
  - Tree network, 630
  - Tri-state, 415, 687
  - Truth table, 662
  - 2's-complement representation, 27
  - Two-wire link, 564
- U**
- UART (Universal Asynchronous Receiver Transmitter), 259
  - UltraSPARC II, 493, 595
  - UMA multiprocessors, 622
  - User mode (state), 215, 221, 343
  - User space, 343
- V**
- Very large-scale integration (VLSI), 9, 20
  - Video display, 558
  - Virtual address, 294, 338
  - Virtual memory, 294, 337–343
    - address translation, 339
    - page, 339
    - page fault, 342
    - page frame, 339
    - page table, 339
    - translation buffer (TLB), 341
    - virtual address, 294, 338
  - Von Neumann architecture, 19
- W**
- Wait loop, 66
  - Wide area networks (WAN), 646
  - Winchester, 345
  - Word, 5
  - Word alignment, 36
  - Word length, 5, 33
  - Workstation, 2
  - Wormhole routing, 634
  - Write buffer, 335
  - Write-back protocol, 316
  - Write-through protocol, 316, 642

