DATA STORAGE

LECTURE SLIDES ARE ADAPTED/MODIFIED FROM SLIDES PROVIDED BY THE TEXTBOOK, **COMPUTER SCIENCE: AN OVERVIEW** BY J. GLENN BROOKSHEAR AND DENNIS BRYLOW

PUBLISHER PEARSON

Data Storage

- Bits and Their Storage
- Main Memory
- Representing Information as Bit Patterns
- The Binary System
- Storing Integers
- Storing Fractions
- Data Compression
- Communications Errors
- Mass Storage

Bits and Bit Patterns

- **Bit:** Binary Digit (0 or 1)
- Bit Patterns are used to represent information.
  - Numbers
  - Text characters
  - Images
  - Sound
  - And others

Boolean Operations

- **Boolean Operation:** An operation that manipulates one or more true/false values
- Specific operations
  - AND
  - OR
  - XOR (exclusive or)
  - NOT

Boolean Operations

The AND operation

$$\begin{array}{lr} & 0 \\ \text{AND} & 0 \\ \hline & 0 \end{array} \qquad \begin{array}{lr} & 0 \\ \text{AND} & 1 \\ \hline & 0 \end{array} \qquad \begin{array}{lr} & 1 \\ \text{AND} & 0 \\ \hline & 0 \end{array} \qquad \begin{array}{lr} & 1 \\ \text{AND} & 1 \\ \hline & 1 \end{array}$$

The OR operation

$$\begin{array}{lr} & 0 \\ \text{OR} & 0 \\ \hline & 0 \end{array} \qquad \begin{array}{lr} & 0 \\ \text{OR} & 1 \\ \hline & 1 \end{array} \qquad \begin{array}{lr} & 1 \\ \text{OR} & 0 \\ \hline & 1 \end{array} \qquad \begin{array}{lr} & 1 \\ \text{OR} & 1 \\ \hline & 1 \end{array}$$

The XOR operation

$$\begin{array}{lr} & 0 \\ \text{XOR} & 0 \\ \hline & 0 \end{array} \qquad \begin{array}{lr} & 0 \\ \text{XOR} & 1 \\ \hline & 1 \end{array} \qquad \begin{array}{lr} & 1 \\ \text{XOR} & 0 \\ \hline & 1 \end{array} \qquad \begin{array}{lr} & 1 \\ \text{XOR} & 1 \\ \hline & 0 \end{array}$$

Gates

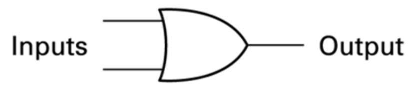
- **Gate:** A device that computes a Boolean operation
  - Often implemented as (small) electronic circuits
  - Provide the building blocks from which computers are constructed
  - VLSI (Very Large Scale Integration)

A pictorial representation of gates

AND

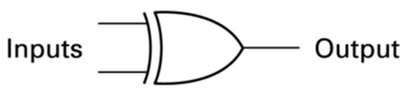
Inputs	Output
0 0	0
0 1	0
1 0	0
1 1	1

OR



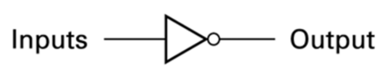
Inputs	Output
0 0	0
0 1	1
1 0	1
1 1	1

XOR



Inputs	Output
0 0	0
0 1	1
1 0	1
1 1	0

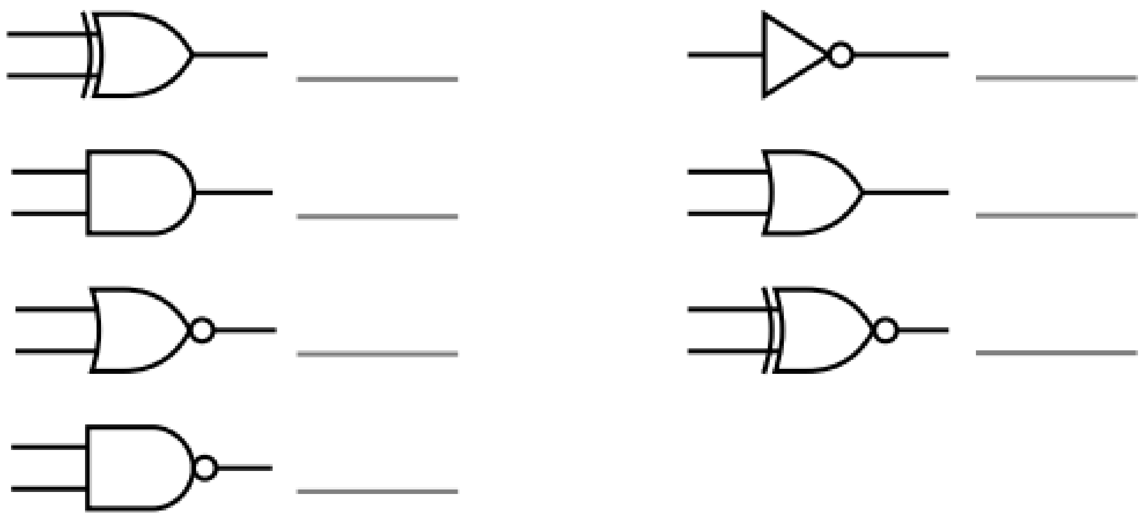
NOT



Inputs	Output
0	1
1	0

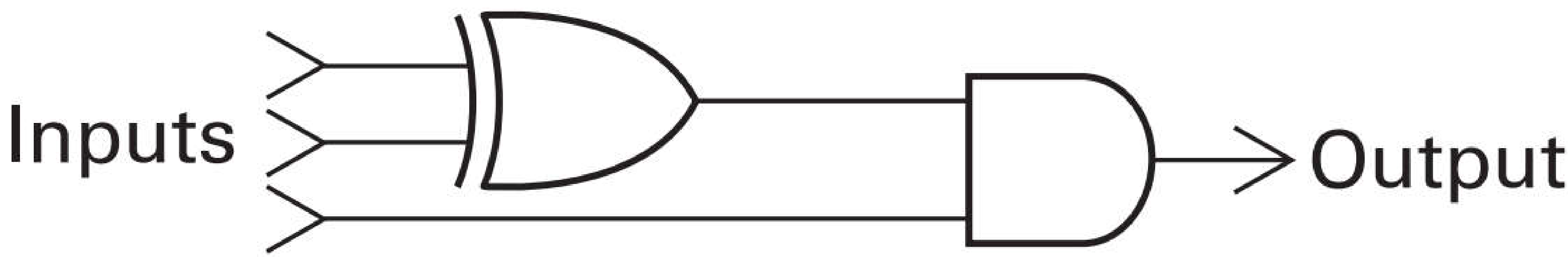
Quiz

What are the names of these gates?



Quiz

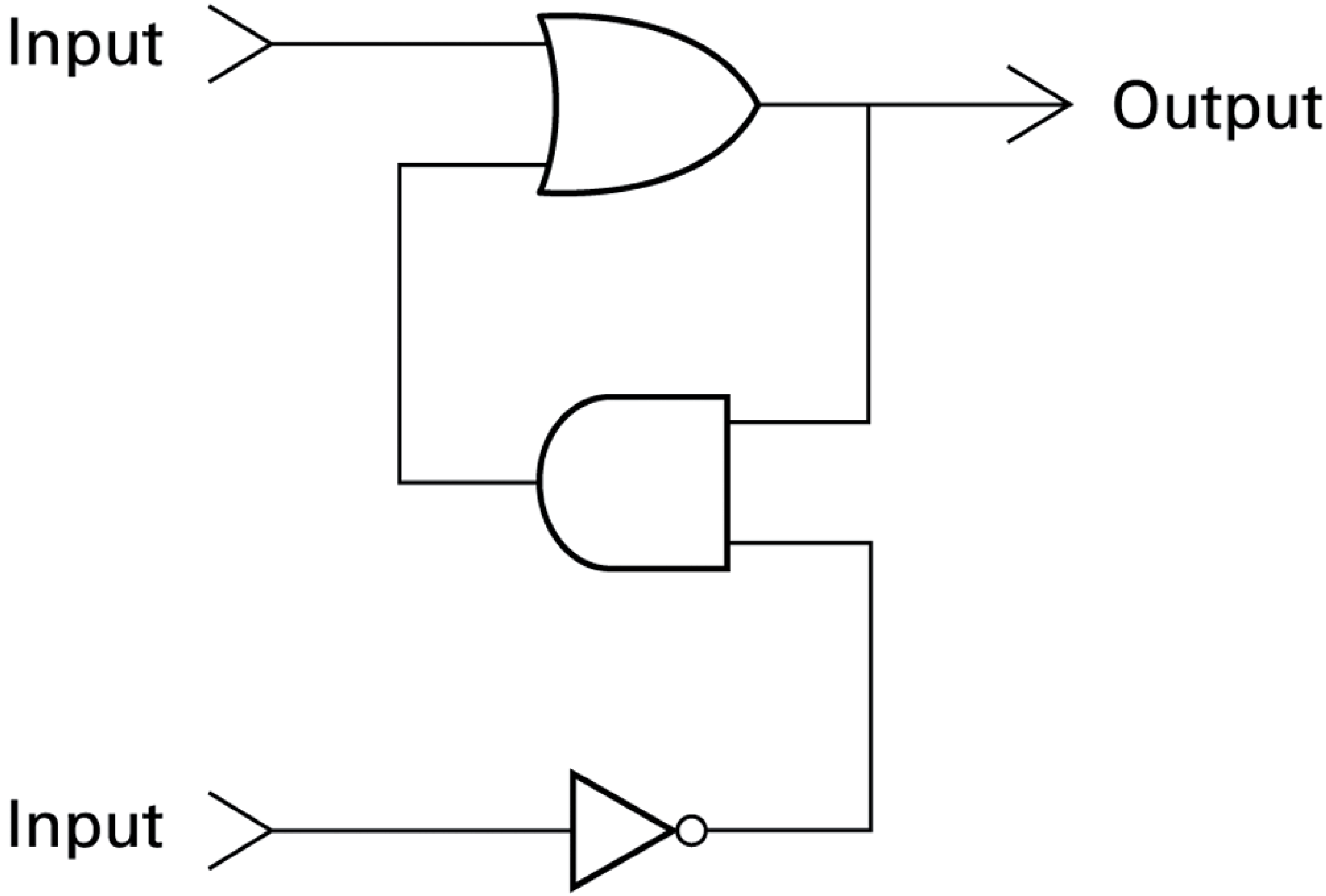
What input bit patterns will cause the following circuit to produce output of 1?



Flip-flops

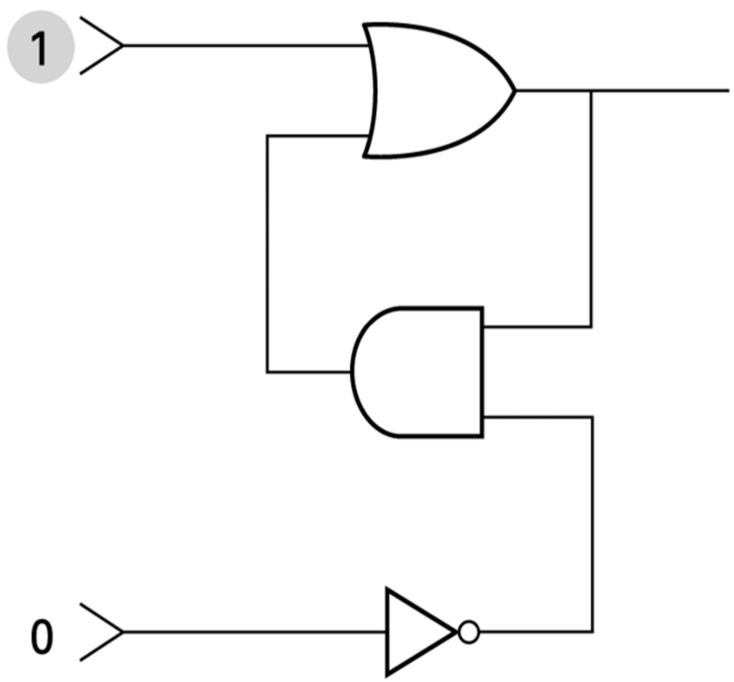
- Flip-flop: A circuit built from gates that can store one bit.
  - One input line is used to set its stored value to 1
  - One input line is used to set its stored value to 0
  - While both input lines are 0, the most recently stored value is preserved

A simple flip-flop circuit



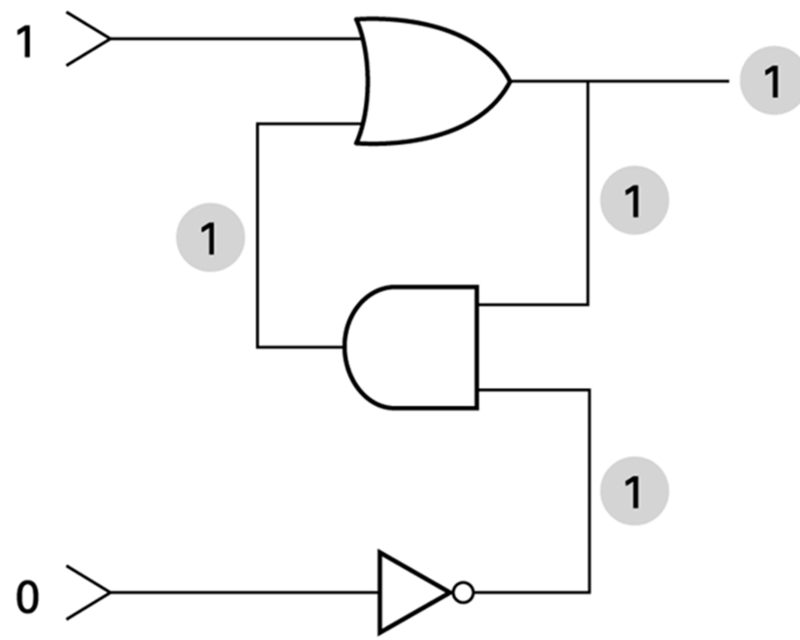
Setting the output of a flip-flop to 1

a. 1 is placed on the upper input.

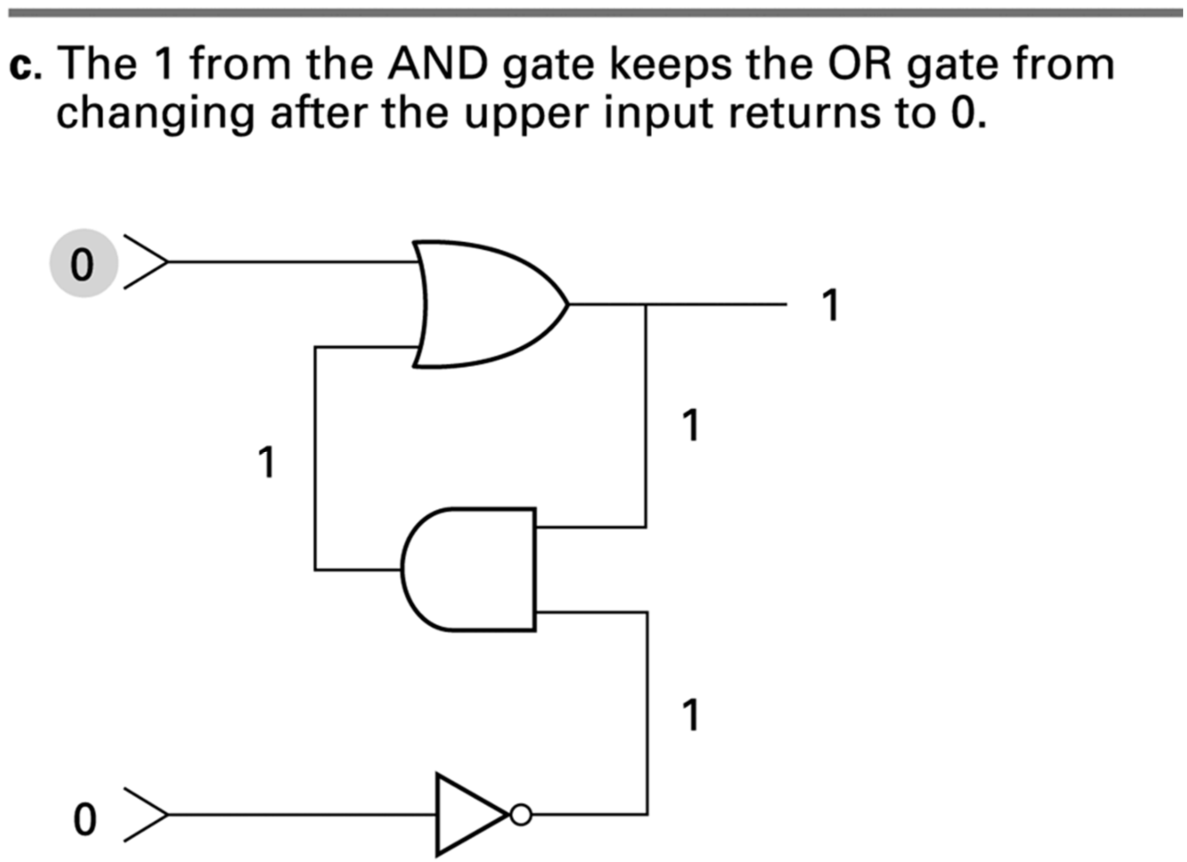


Setting the output of a flip-flop to 1

b. This causes the output of the OR gate to be 1 and, in turn, the output of the AND gate to be 1.

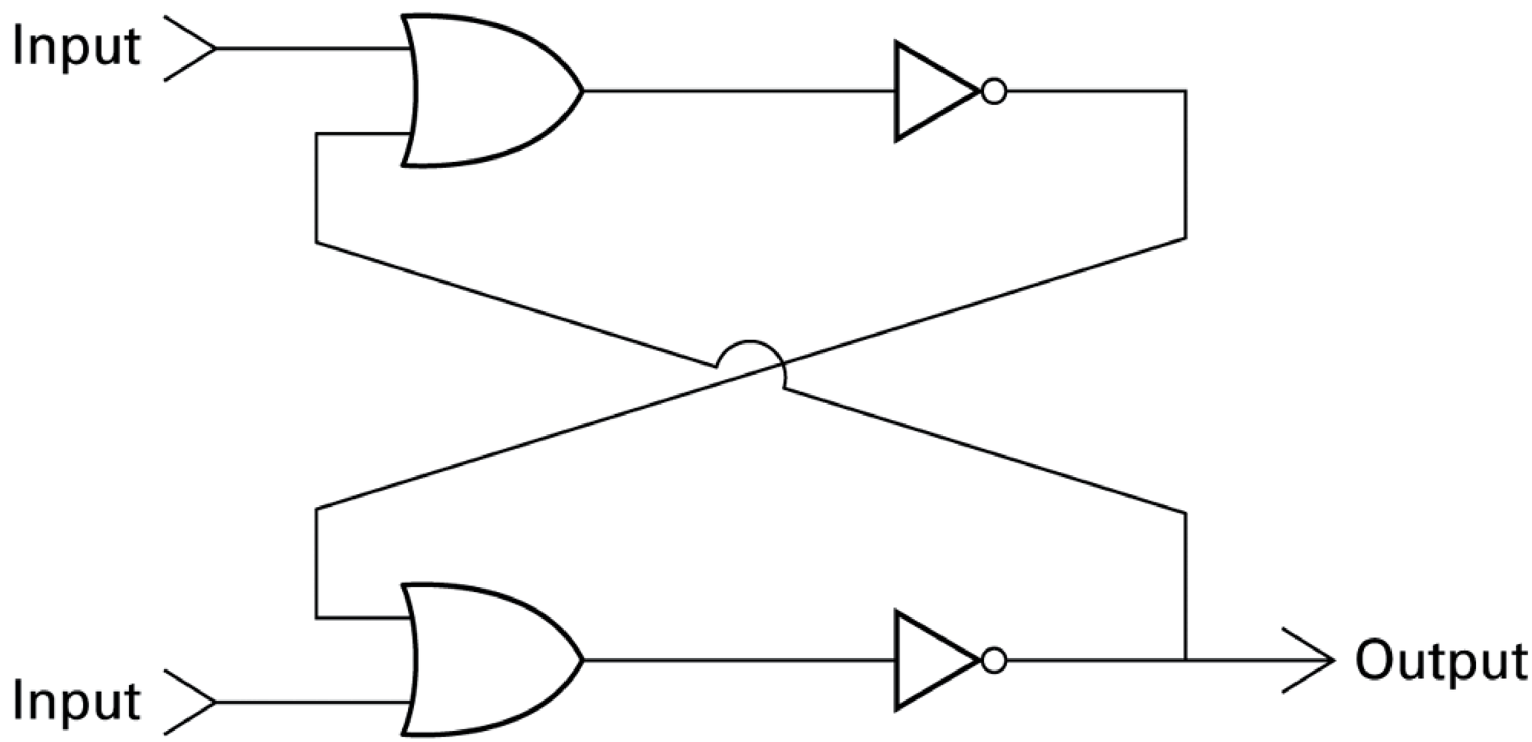


Setting the output of a flip-flop to 1



Quiz

Another flip-flop



Quiz

Quiz

Activity

Hexadecimal Notation

- **Hexadecimal notation:** A shorthand notation for long bit patterns
  - Divides a pattern into groups of four bits each
  - Represents each group by a single symbol
- Example: 10100011 becomes A3

The hexadecimal coding system

Bit pattern	Hexadecimal representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

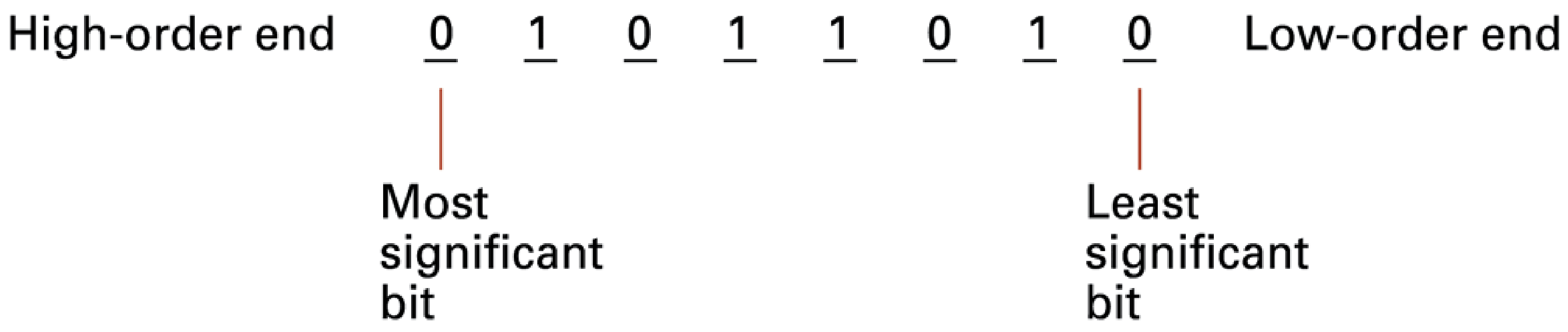
Quiz

Main Memory

Main Memory Cells

- **Cell:** A unit of main memory (typically 8 bits which is one **byte**)
  - **Most significant bit:** the bit at the left (high-order) end of the conceptual row of bits in a memory cell
  - **Least significant bit:** the bit at the right (low-order) end of the conceptual row of bits in a memory cell

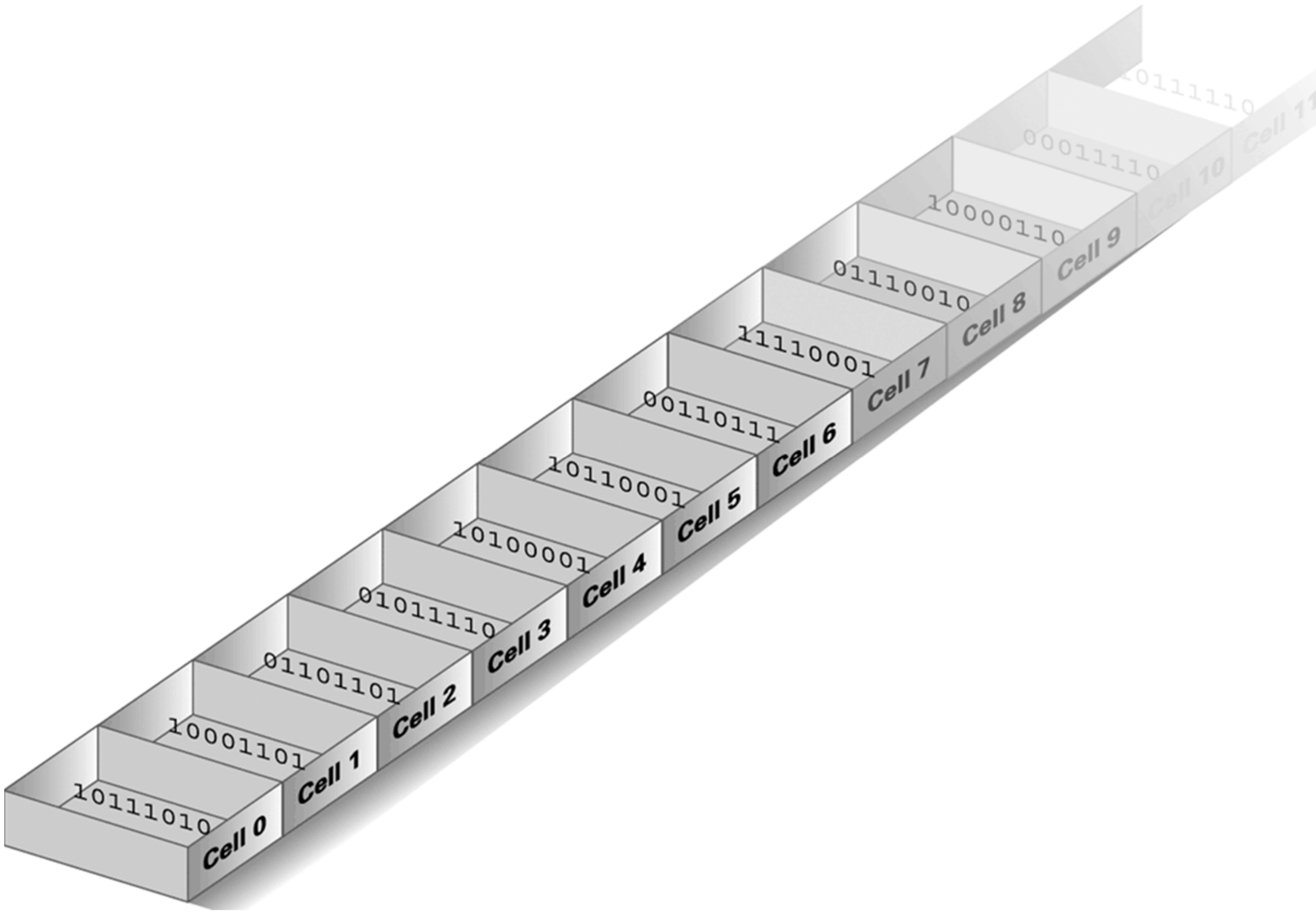
The organization of a byte-size memory cell



Main Memory Addresses

- **Address:** A "name" that uniquely identifies one cell in the computer's main memory
  - The names are actually numbers.
  - These numbers are assigned consecutively starting at zero.
  - Numbering the cells in this manner associates an order with the memory cells.

Memory cells arranged by address



Memory Terminology

- **Random Access Memory (RAM):** Memory in which individual cells can be easily accessed in any order
- **Dynamic Memory (DRAM):** RAM composed of volatile memory

Measuring Memory Capacity

- Kilobyte: 210 bytes = 1024 bytes
  - Example: 3 KB = 3 times1024 bytes
- Megabyte: 220 bytes = 1,048,576 bytes
  - Example: 3 MB = 3 times 1,048,576 bytes
- Gigabyte: 230 bytes = 1,073,741,824 bytes
  - Example: 3 GB = 3 times 1,073,741,824 bytes
- Terabyte
- Petabyte
- Exabyte

- Kibi-, Mebi-, Gibi-,..

Representing Information as Bit Patterns

Representing Text

- Each character (letter, punctuation, etc.) is assigned a unique bit pattern.
  - ASCII: Uses patterns of 7-bits to represent most symbols used in written English text
  - ISO developed a number of 8 bit extensions to ASCII, each designed to accommodate a major language group
  - Unicode: Uses patterns of 16-bits to represent the major symbols used in languages world wide (UTF-8, UTF-16,...)

The message “Hello.” in ASCII

01001000	01100101	01101100	01101100	01101111	00101110
H	**e**	**l**	**l**	**o**	**.**

Representing Numeric Values

- Binary notation: Uses bits to represent a number in base two
- Limitations of computer representations of numeric values
  - Overflow: occurs when a value is too big to be represented
  - Truncation: occurs when a value cannot be represented accurately

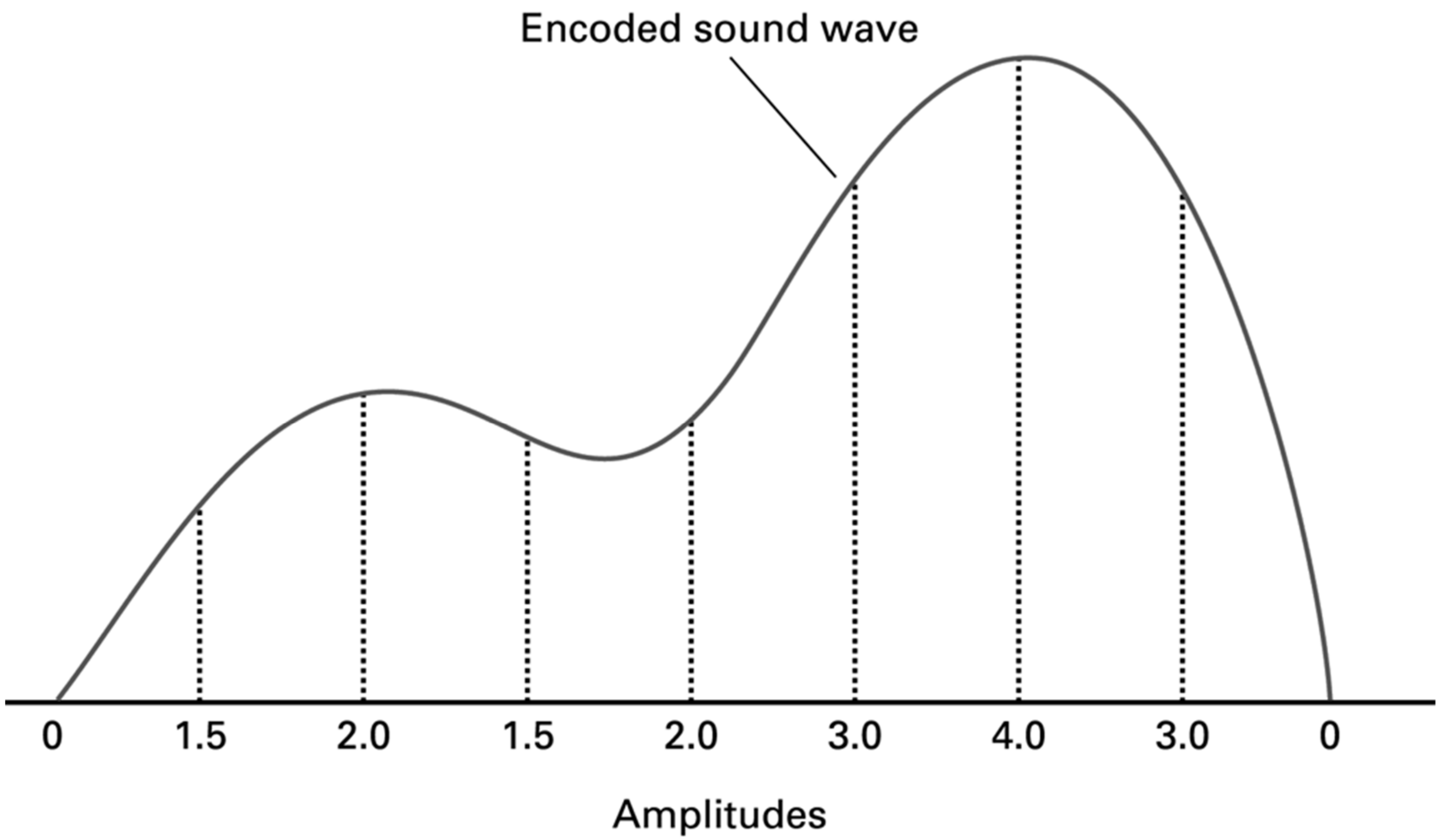
Representing Images

- Bit map techniques
  - Pixel: short for "picture element"
  - RGB
  - Luminance and chrominance
- Vector techniques
  - Scalable
  - TrueType and PostScript

Representing Sound

- Sampling techniques
  - Used for high quality recordings
  - Records actual audio
- MIDI
  - Used in music synthesizers
  - Records “musical score”

Representing Sound



The sound wave represented by the sequence 0, 1.5, 2.0, 1.5, 2.0, 3.0, 4.0, 3.0, 0

Quiz

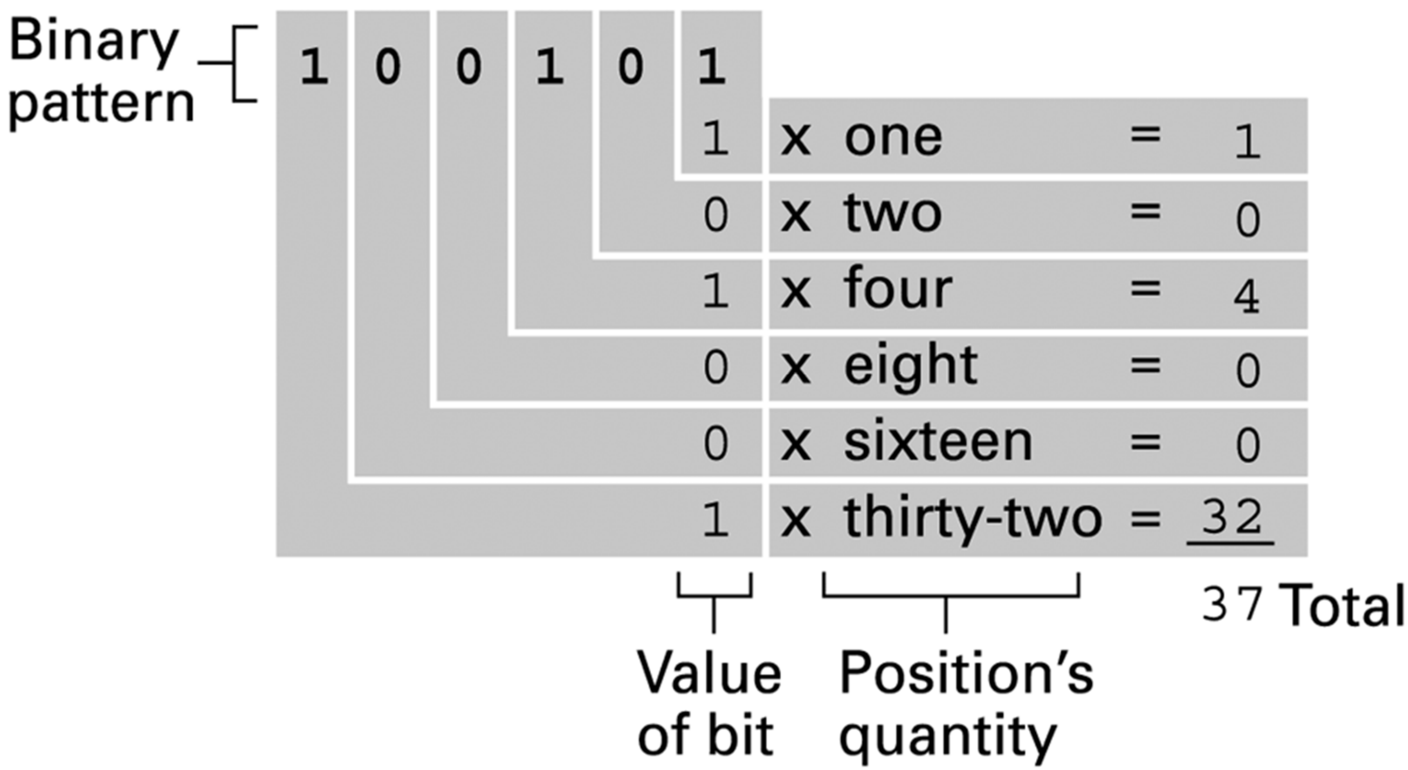
The Binary System

The Binary System

- The traditional decimal system is based on **powers of ten.**
- The Binary system is based on **powers of two.**

The base ten and binary systems

Decoding the binary representation 100101



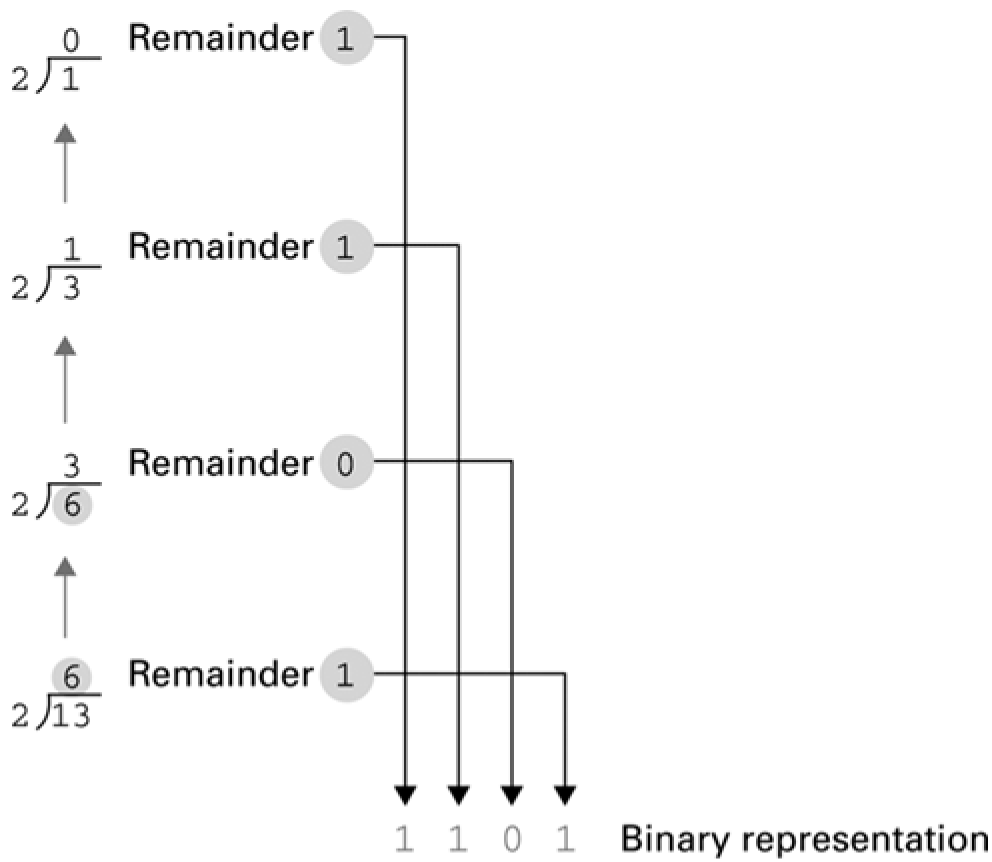
Algorithm for finding the binary representation of a positive integer

Step 1. Divide the value by two and record the remainder.

Step 2. As long as the quotient obtained is not zero, continue to divide the newest quotient by two and record the remainder.

Step 3. Now that a quotient of zero has been obtained, the binary representation of the original value consists of the remainders listed from right to left in the order they were recorded.

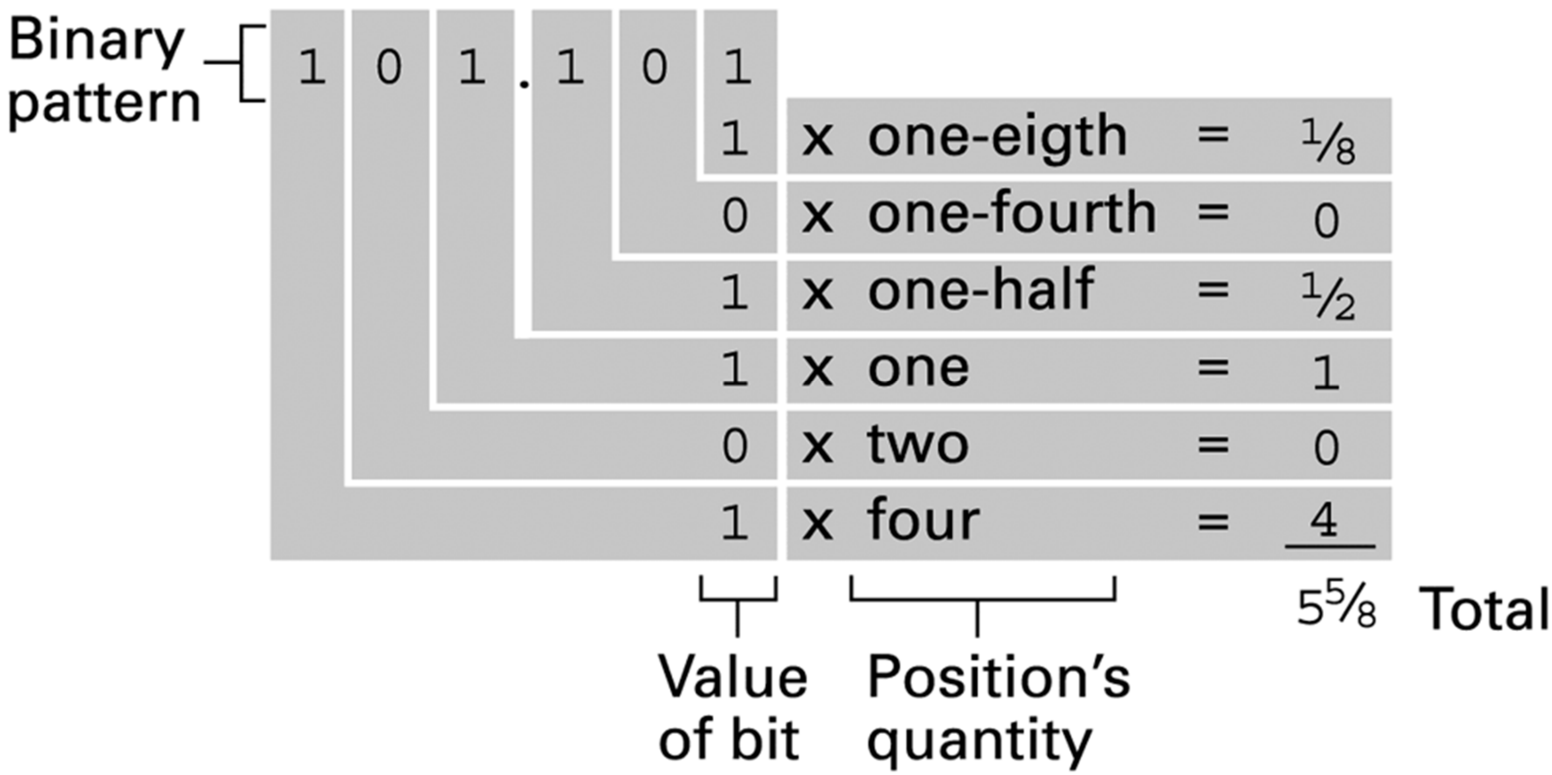
Applying the algorithm



The binary addition facts

$$\begin{array}{r} 0 \\ +\,0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ +\,0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ +\,1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\,1 \\ \hline 10 \end{array}$$

Decoding the binary representation



Quiz

Quiz

Quiz

Storing Integers

Storing Integers

- **Two's complement notation:** The most popular means of representing integer values
- **Excess notation:** Another means of representing integer values
- Both can suffer from overflow errors.

Two's complement notation systems

- leftmost bit: sign bit
  - 1: negative
  - 0: positive

- Complement: changing all the 0s to 1s, all the 1s to 0s.

  0110 and 1001 are complements.

Two's complement notation systems

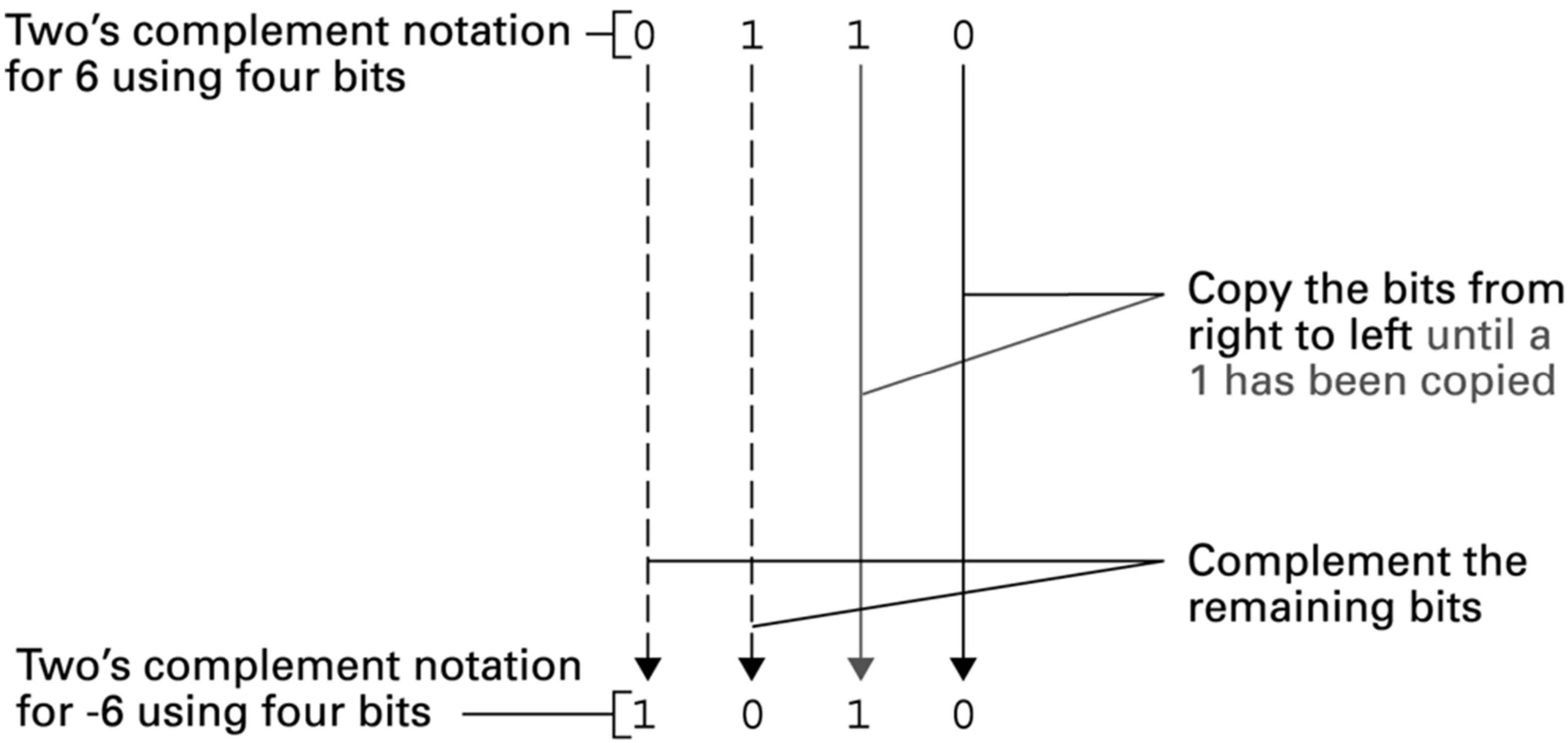
a. Using patterns of length three

Bit pattern	Value represented
011	3
010	2
001	1
000	0
111	−1
110	−2
101	−3
100	−4

b. Using patterns of length four

Bit pattern	Value represented
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	−1
1110	−2
1101	−3
1100	−4
1011	−5
1010	−6
1001	−7
1000	−8

Coding the value -6 in two's complement notation using four bits



Addition problems converted to two's complement notation

Problem in base ten		Problem in two's complement		Answer in base ten
3 + 2	→	0011 + 0010 0101	→	5
−3 + −2	→	1101 + 1110 1011	→	−5
7 + −5	→	0111 + 1011 0010	→	2

Quiz

Quiz

Quiz

Min and max of signed integers

n	minimum	maximum
8	-2^7 = **-128**	$2^7 - 1$ = **+127**
16	-2^{15} = **-32,768**	$2^{15} - 1$ = **+32,767**
32	-2^{31} = **-2,147,483,648**	$2^{31} - 1$ = **+2,147,483,647**
64	-2^{63} = **-9,223,372,036,854,775,808**	$2^{63}-1$ = **+9,223,372,036,854,775,807**

Min and max of unsigned integers

n	Minimum	Maximum
8	0	$2^8 - 1 =$ **255**
16	0	$2^{16} - 1 =$ **65,535**
32	0	$2^{32} - 1 =$ **4,294,967,295**
64	0	$2^{64} - 1 =$ **18,446,744,073,709,551,615**

An excess eight conversion table

Bit pattern	Value represented
1111	7
1110	6
1101	5
1100	4
1011	3
1010	2
1001	1
1000	0
0111	−1
0110	−2
0101	−3
0100	−4
0011	−5
0010	−6
0001	−7
0000	−8

An excess notation system using bit patterns of length three

Bit pattern	Value represented
111	3
110	2
101	1
100	0
011	−1
010	−2
001	−3
000	−4

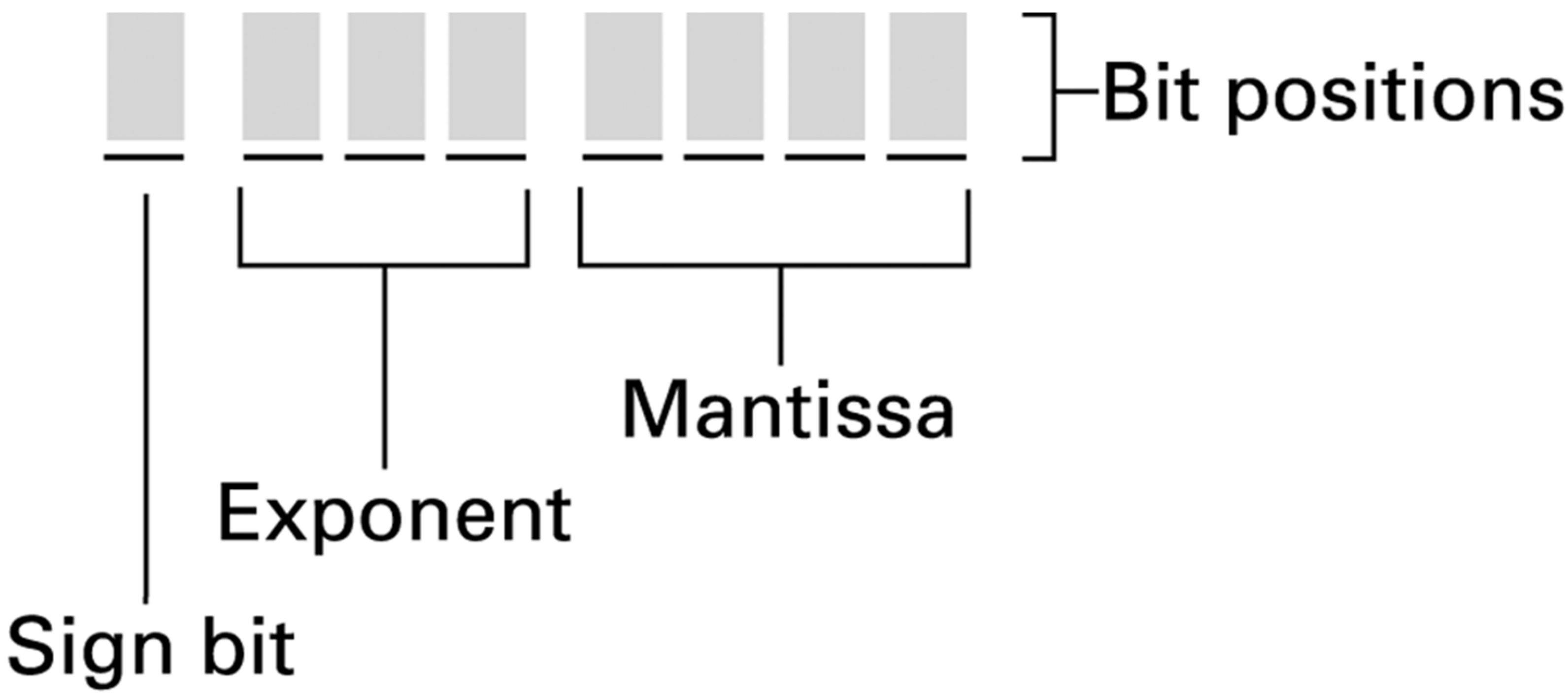
Storing Fractions

Storing Fractions

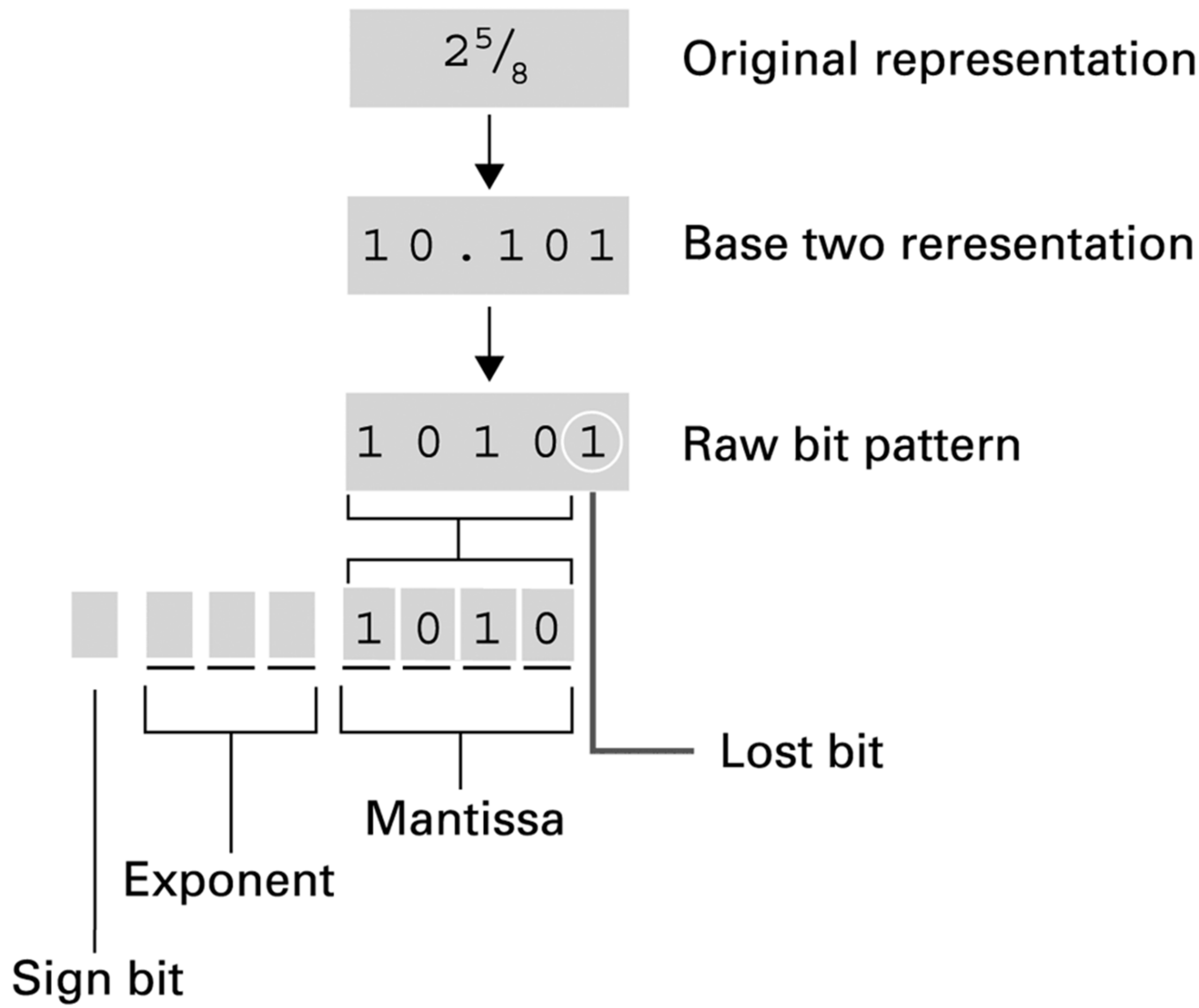
- **Floating-point Notation:** Consists of a sign bit, a mantissa field, and an exponent field.
  - Mantissa: also **fraction/significand**

- Related topics include
  - Normalized form
  - Truncation errors

Floating-point notation components



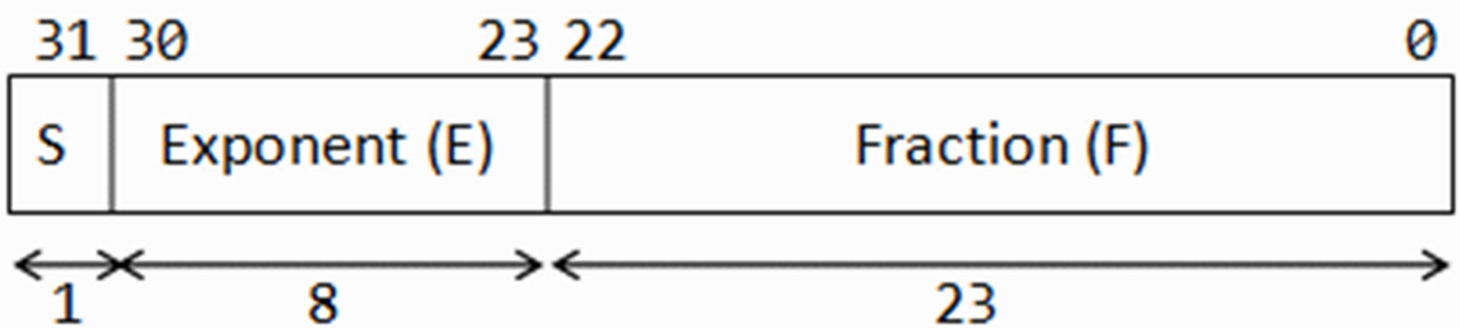
Encoding the value 2 5⁄8



IEEE-754 32-bit Single-Precision Floating-Point Numbers

- Exponent: excess-127 (bias-127)
- Fraction: implicit leading bit (before radix point).

$$N = (-1)^S \times 1.F \times 2^{(E-127)}$$



32-bit Single-Precision Floating-point Number

IEEE-754 32-bit Single-Precision Floating-Point Numbers

- Suppose that IEEE-754 32-bit floating-point representation pattern is 0 10000000 110 0000 0000 0000 0000 0000.

```
Sign bit S = 0 ⇒ positive number
E = 1000 0000B = 128D
Fraction is 1.11B (with an implicit leading 1)
= 1 + 1×2^-1 + 1×2^-2 = 1.75D

The number is +1.75 × 2^(128-127) = +3.5D
```

IEEE-754 32-bit Single-Precision Floating-Point Numbers

Suppose that IEEE-754 32-bit floating-point representation pattern is 1 01111110 100 0000 0000 0000 0000 0000.

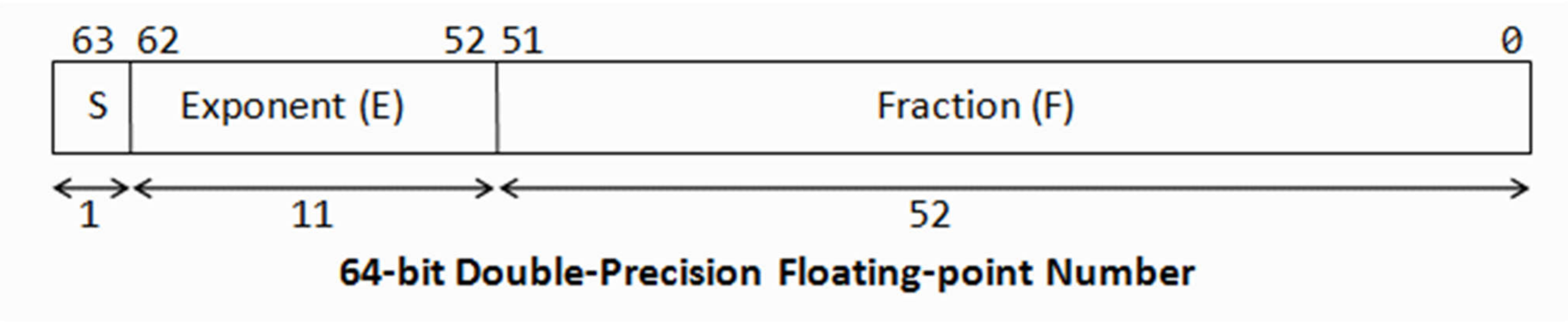
```
Sign bit S = 1 ⇒ negative number
E = 0111 1110B = 126D
Fraction is 1.1B (with an implicit leading 1) =
1 + 2^-1 = 1.5D

The number is -1.5 × 2^(126-127)=-0.75D
```

IEEE-754 64-bit Double-Precision Floating-Point Numbers

- Exponent: excess-1023
- Fraction: implicit leading bit (before radix point).

$$N = (-1)\text{^}S \times 1.F \times 2\text{^}(E-1023)$$



Min and Max of Floating-Point Numbers

Precision	Min	Max
Single	1.1754×10^{-38}	3.40282×10^{38}
Double	2.2250×10^{-308}	1.7976×10^{308}

Data Compression

Data Compression

- Lossy versus lossless
- Run-length encoding
- Frequency-dependent encoding

  (Huffman codes)
- Relative encoding
- Dictionary encoding (Includes adaptive dictionary encoding such as LZW encoding.)

Compressing Images

- GIF: Good for cartoons
- JPEG: Good for photographs
- TIFF: Good for image archiving

Compressing Audio and Video

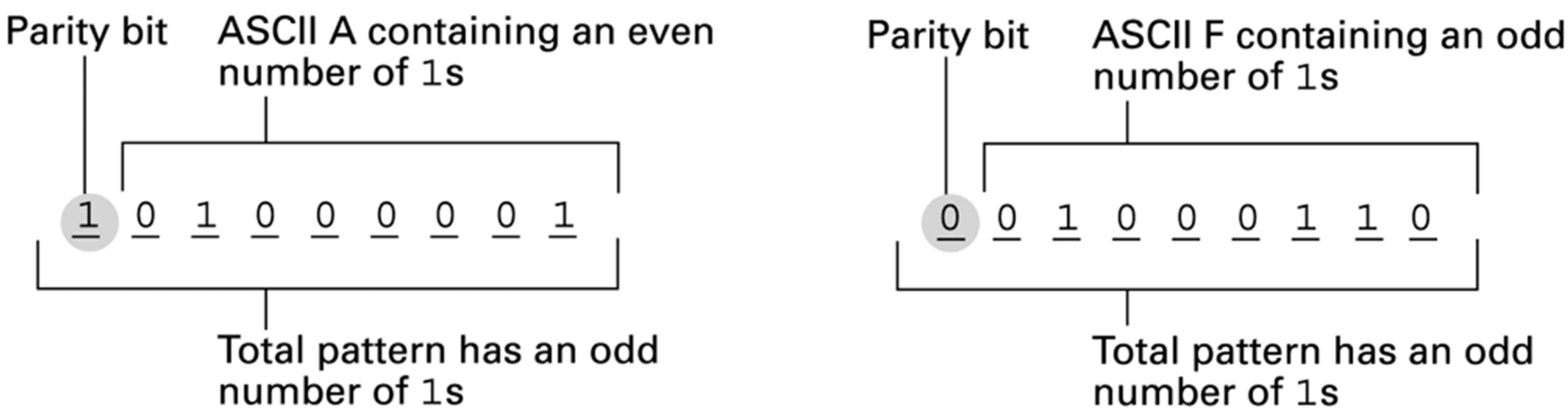
- MPEG
  - High definition television broadcast
  - Video conferencing
- MP3
  - Temporal masking
  - Frequency masking

Communication Errors

Communication Errors

- Parity bits (even versus odd)
- Checkbytes
- Error correcting codes

The ASCII codes for the letters A and F adjusted for odd parity



An error-correcting code

- Hamming distance (of two bit patterns):
  - Number of bits in which the patterns differ.

- Example:
  - Hamming(000000, 001111) = 4
  - Hamming(10101100, 01100100) = 3

An error-correcting code

Symbol	Code
A	000000
B	001111
C	010011
D	011100
E	100110
F	101001
G	110101
H	111010

Decoding the pattern 010100

Character	Code	Pattern received	Distance between received pattern and code
A	0 0 0 0 0 0	0 **1** 0 **1** 0 0	2
B	0 0 1 1 1 1	0 **1** **0** 1 **0** **0**	4
C	0 1 0 0 1 1	0 1 0 **1** **0** **0**	3
D	0 1 1 1 0 0	0 1 **0** 1 0 0	**1**
E	1 0 0 1 1 0	**0** **1** 0 1 **0** 0	3
F	1 0 1 0 0 1	**0** **1** **0** **1** 0 **0**	5
G	1 1 0 1 0 1	**0** 1 0 1 0 **0**	2
H	1 1 1 0 1 0	**0** 1 **0** **1** **0** 0	4

Smallest distance

Quiz

Quiz

Quiz

Mass Storage

Mass Storage

- On-line versus off-line
- Typically larger than main memory
- Typically less volatile than main memory
- Typically slower than main memory
- Typically lower cost than main memory

Mass Storage Systems

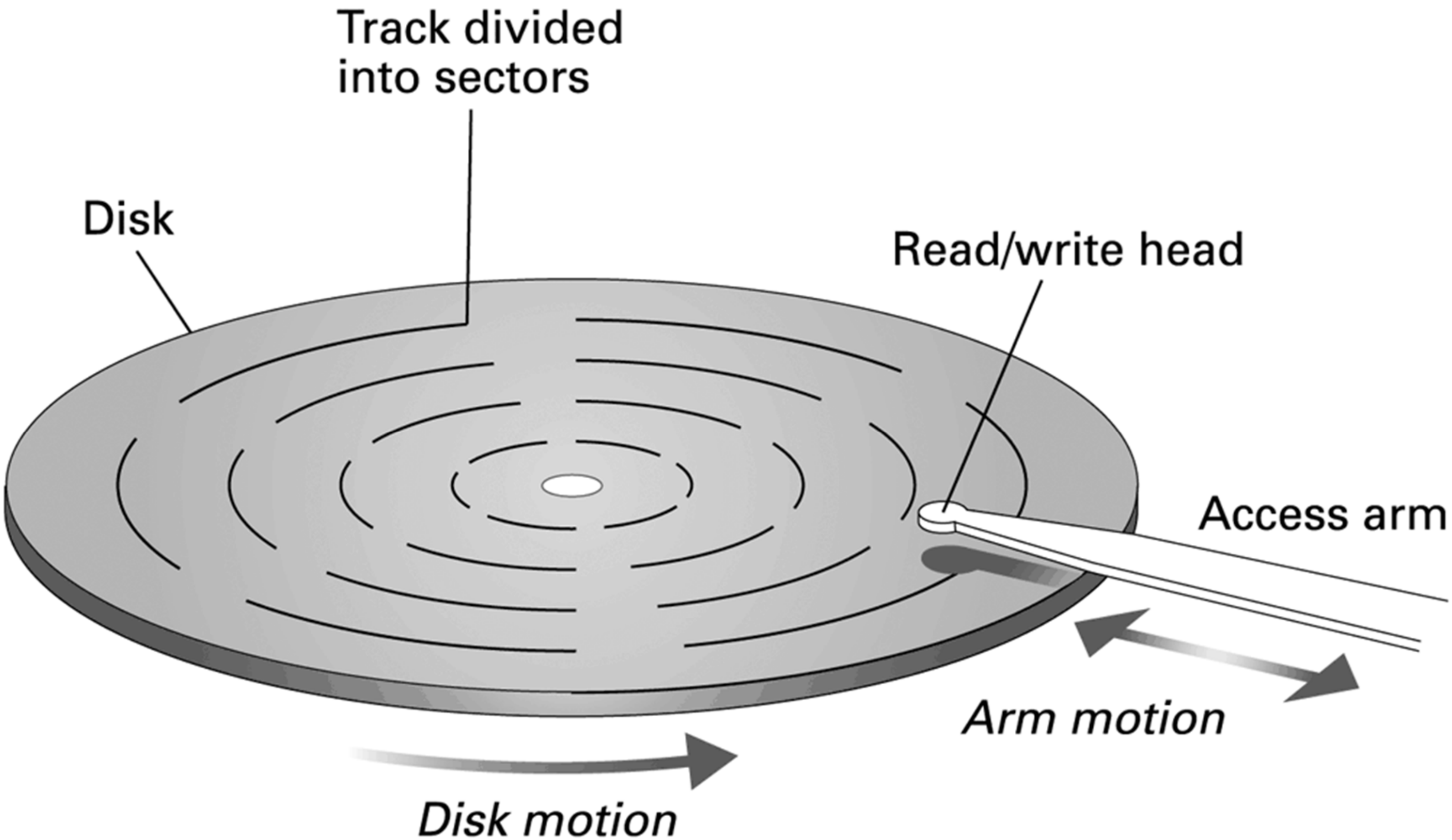
- Magnetic Systems
  - Disk
  - Tape
- Optical Systems
  - CD
  - DVD
- Flash Technology
  - Flash Drives
  - Secure Digital (SD) Memory Card

Source: Wikipedia

Source: Wikipedia

Source: Wikipedia

A magnetic disk storage system



A magnetic disk storage system

- Data stored on a spinning disk
- Disk divided into concentric rings (sectors)
- Read/write head moves from one ring to another while disk spins
- Access time depends on
  - Time to move head to correct sector
  - Time for sector to spin to data location
- Fixed number of sectors are placed in a track.

A magnetic disk storage system

- Seek time
  - Time needed to position the read/write head over the correct track
- Latency
  - Time for the beginning of the desired sector to rotate under the read/write head
- Transfer time
  - Time for the entire sector to pass under the read/write head and have its contents read into or written from memory

A magnetic disk storage system

- Given:
  - Rotation speed = 7,200 rev/min=120 rev/sec = 8.33 msec/rev
  - Arm movement time = 0.02 msec to move to an adjacent track
  - On average, the read/write head must move about 300 tracks
  - Number of tracks/surface = 1,000
  - Number of sectors/track = 64
  - Number of bytes/sector = 1,024
- Seek time
  - Best case = 0 msec;
  - Worst case = 999*0.02=19.98 msec
  - Average case = 300*0.02 = 6 msec

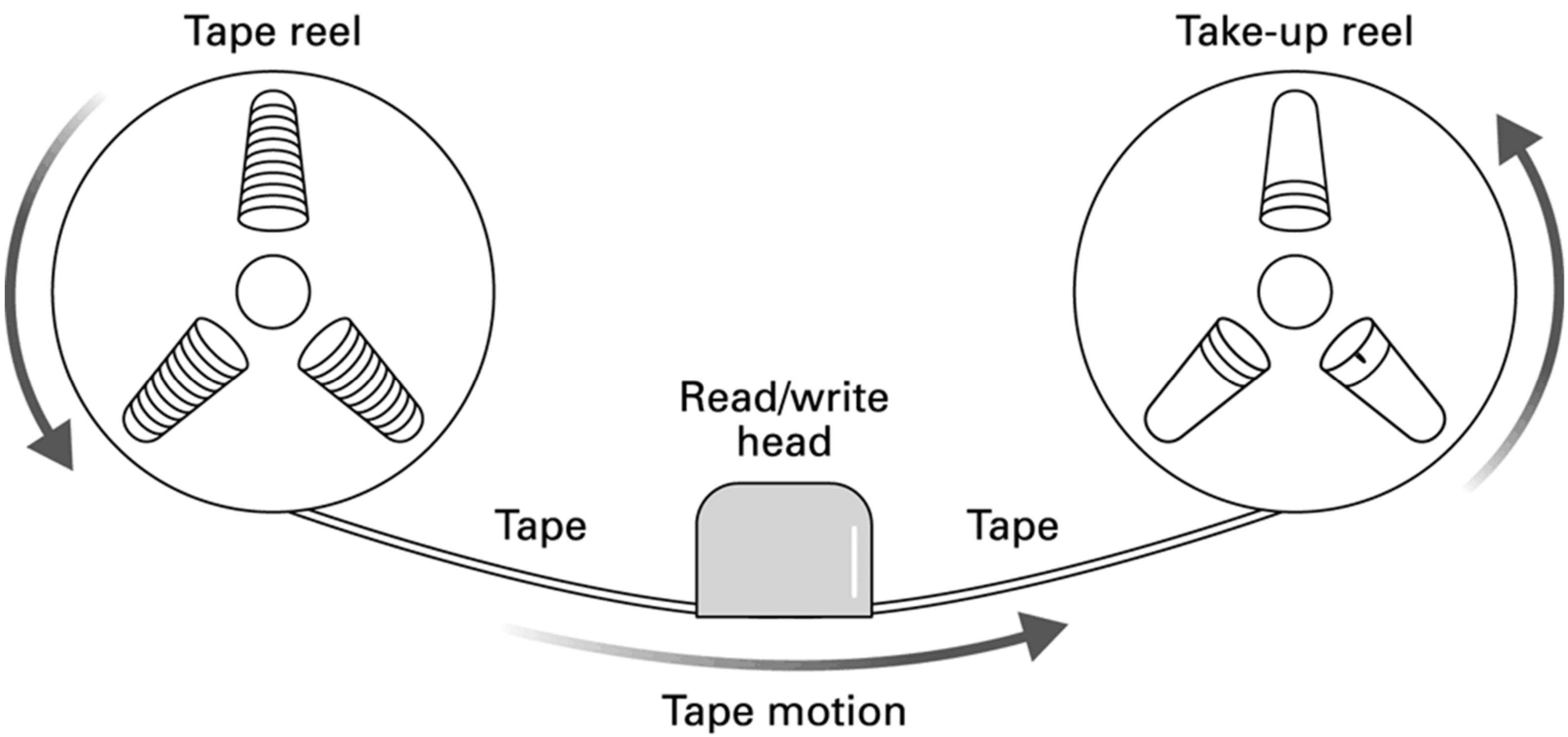
A magnetic disk storage system

- Given:
  - Rotation speed = 7,200 rev/min=120 rev/sec = 8.33 msec/rev
  - Arm movement time = 0.02 msec to move to an adjacent track
  - On average, the read/write head must move about 300 tracks
  - Number of tracks/surface = 1,000
  - Number of sectors/track = 64
  - Number of bytes/sector = 1,024
- Latency
  - Best case = 0 msec;
  - Worst case = 8.33 msec
  - Average case = 4.17 msec

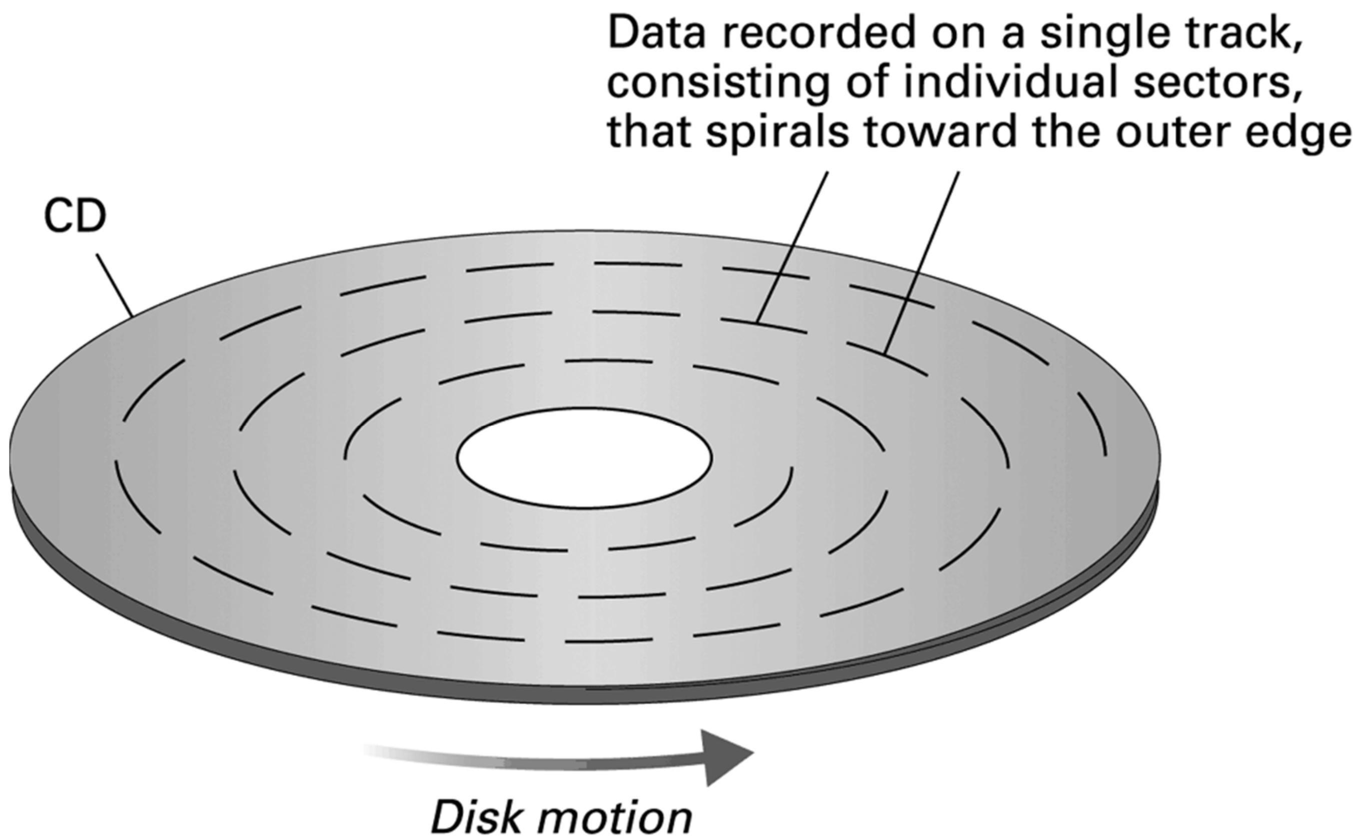
A magnetic disk storage system

- Given:
  - Rotation speed = 7,200 rev/min=120 rev/sec = 8.33 msec/rev
  - Arm movement time = 0.02 msec to move to an adjacent track
  - On average, the read/write head must move about 300 tracks
  - Number of tracks/surface = 1,000
  - Number of sectors/track = 64
  - Number of bytes/sector = 1,024
- Transfer time
  - 1/64 * 8.33 msec = 4.17 msec

Magnetic tape storage



CD storage


Data recorded on a single track, consisting of individual sectors, that spirals toward the outer edge
CD
Disk motion