Chapter 1: Data Storage

- 1.1 Bits and Their Storage
- 1.2 Main Memory
- 1.3 Mass Storage
- 1.4 Representing Information as Bit Patterns
- 1.5 The Binary System
- 1.6 Storing Integers
- 1.7 Storing Fractions (skip for now)
- 1.8 Data Compression (skim)
- 1.9 Communications Errors (skip for now)

1-1

Bits and Bit Patterns

- **Bit:** Binary Digit (0 or 1)
- Bit Patterns are used to represent information.
  - Numbers
  - Text characters
  - Images
  - Sound
  - Combinations of the above (e.g., video)
  - and *Instructions*

1-2

Boolean Operations

- Think of 0 as *false* and 1 as *true*
- **Boolean Operation:** An operation that manipulates one or more true/false values
- Specific operations
  - AND
  - OR
  - XOR (exclusive or)
  - NOT

1-3

The Boolean operations AND, OR, and XOR (exclusive or)

The AND operation

```
  0      0      1      1
AND 0  AND 1  AND 0  AND 1
  0      0      0      1
```

The OR operation

```
  0      0      1      1
OR  0  OR  1  OR  0  OR  1
  0      1      1      1
```

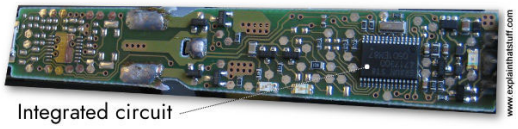
The XOR operation

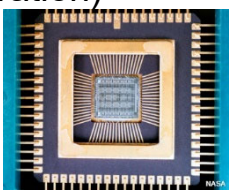
```
  0      0      1      1
XOR 0  XOR 1  XOR 0  XOR 1
  0      1      1      0
```

1-4

Gates

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





1-5

A pictorial representation of AND, OR, XOR, and NOT gates as well as their input and output values

AND

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

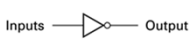
OR

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

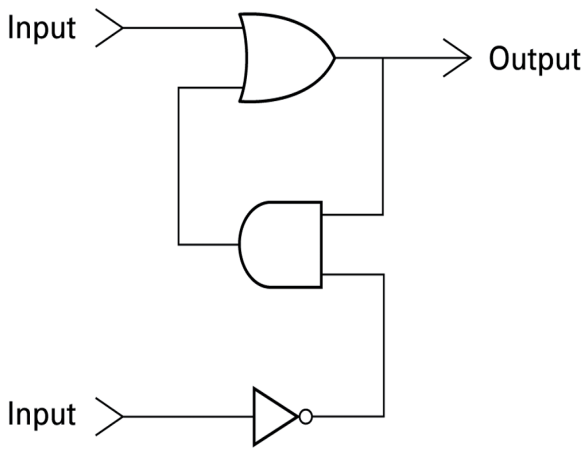
1-6

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

1-7

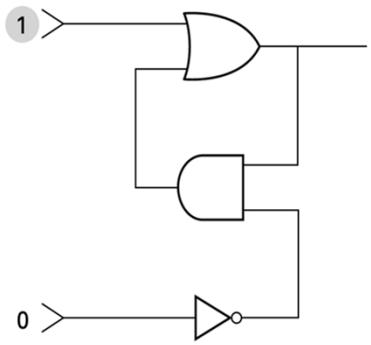
A simple flip-flop circuit



1-8

Setting the output of a flip-flop to 1

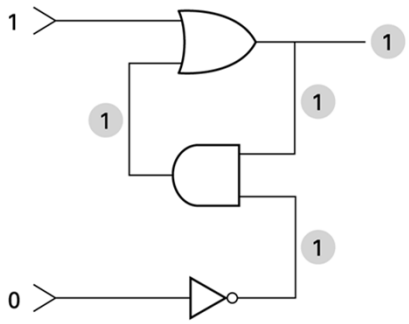
a. 1 is placed on the upper input.



1-9

Setting the output of a flip-flop to 1 (continued)

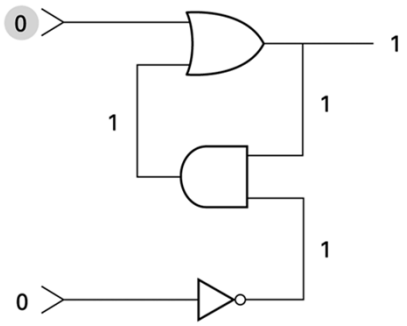
b. This causes the output of the OR gate to be 1 and, in turn, the output of the AND gate to be 1.



1-10

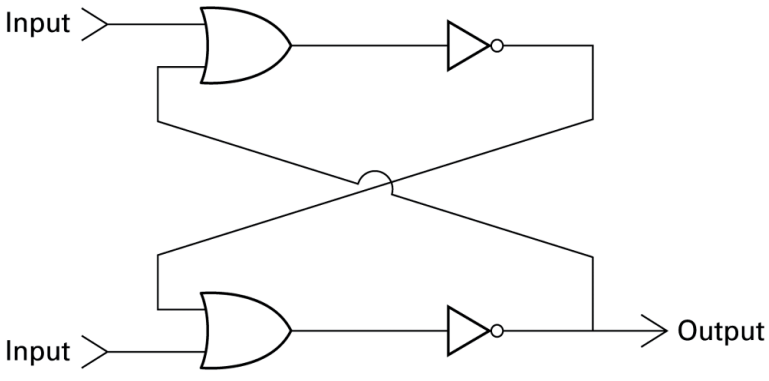
Setting the output of a flip-flop to 1 (continued)

c. The 1 from the AND gate keeps the OR gate from changing after the upper input returns to 0.



1-11

Another way of constructing a flip-flop



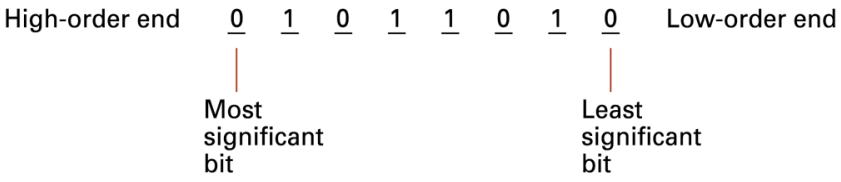
1-12

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

1-13

The organization of a byte-size memory cell



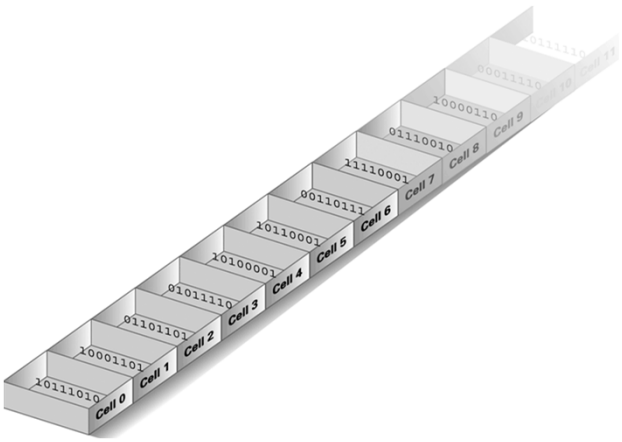
1-14

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.

1-15

Memory cells arranged by address



1-16

Measuring Memory Capacity

- **Kilobyte:** 2^{10} bytes = 1024 bytes
  - Example: 3 KB = 3 times 1024 bytes
  - Sometimes “kibi” rather than “kilo”
- **Megabyte:** 2^{20} bytes = 1,048,576 bytes
  - Example: 3 MB = 3 times 1,048,576 bytes
  - Sometimes “megi” rather than “mega”
- **Gigabyte:** 2^{30} bytes = 1,073,741,824 bytes
  - Example: 3 GB = 3 times 1,073,741,824 bytes
  - Sometimes “gigi” rather than “giga”

1-17

Kilo, Mega, Giga, Tera, Peta, Exa, Zetta, Yotta

physics.nist.gov/cuu/Units/binary.html

- Common use prefixes (all SI, except K [= k in SI])

Name	Abbr	Factor	SI size
Kilo	K	2^{10} = 1,024	10^{3} = 1,000
Mega	M	2^{20} = 1,048,576	10^{6} = 1,000,000
Giga	G	2^{30} = 1,073,741,824	10^{9} = 1,000,000,000
Tera	T	2^{40} = 1,099,511,627,776	10^{12} = 1,000,000,000,000
Peta	P	2^{50} = 1,125,899,906,842,624	10^{15} = 1,000,000,000,000,000
Exa	E	2^{60} = 1,152,921,504,606,846,976	10^{18} = 1,000,000,000,000,000,000
Zetta	Z	2^{70} = 1,180,591,620,717,411,303,424	10^{21} = 1,000,000,000,000,000,000,000
Yotta	Y	2^{80} = 1,208,925,819,614,629,174,706,176	10^{24} = 1,000,000,000,000,000,000,000,000

- **Confusing! Common usage of “kilobyte” means 1024 bytes, but the “correct” SI value is 1000 bytes**
- **Hard Disk manufacturers & Telecommunications are the only computing groups that use SI factors, so what is advertised as a 30 GB drive will actually only hold about 28 x 2^{30} bytes, and a 1 Mbit/s connection transfers 10^{6} bps.**

RAM (Main Memory)

- **Random Access Memory (RAM):**
  - Memory in which individual cells can be easily accessed in any order
  - Constructed from flip-flop circuits
  - Electronic means:
    - Fast
    - Expensive
    - Small
    - Volatile
  - Working space for a computer processor

1-19

Mass Storage (Secondary Memory)

- On-line versus off-line
- Larger than main memory
- Less volatile than main memory
- Slower access time than main memory
- Cheaper (per byte) than main memory

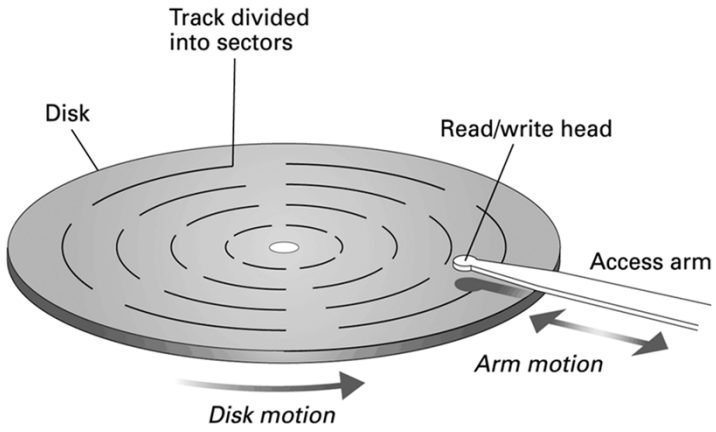
1-20

Mass Storage Systems

- Magnetic Systems
  - Disk (HDD or “floppy”)
  - Tape
- Optical Systems
  - CD
  - DVD
- Flash-based Solid State Drive (SSD)
  - Flash drives, SD cards, smartphone memory
- Performance comparison

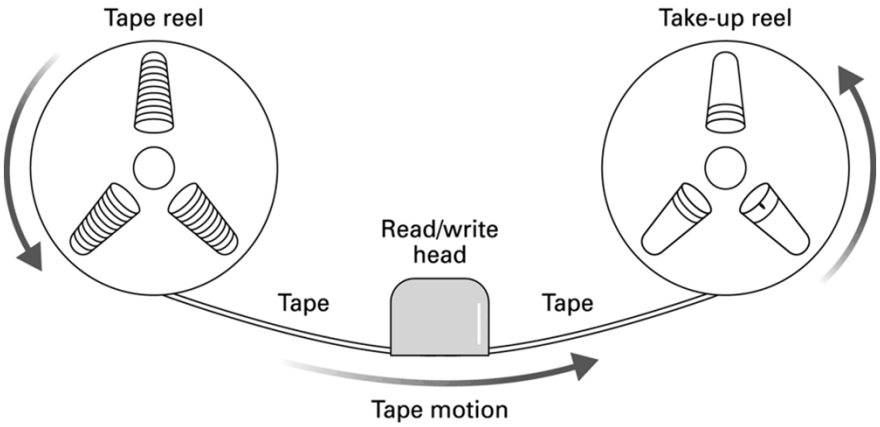
1-21

A magnetic disk storage system



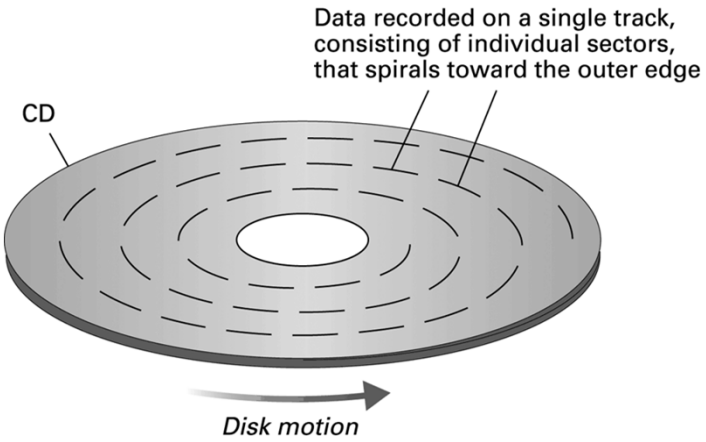
1-22

Magnetic tape storage



1-23

CD storage

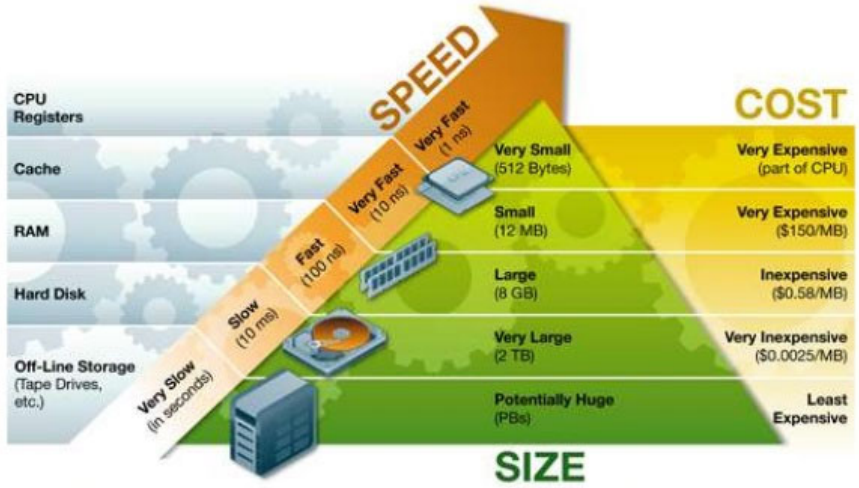


1-24

Flash memory (solid state)

- Solid state = no moving parts
- Non-volatile
- Can store and erase data electronically
- Degrades over time, so unsuitable for RAM (for now), but taking over other types of secondary memory

Extended Memory Hierarchy



Source: http://www.ts.avnet.com/uk/products_and_solutions/storage/hierarchy.html

Representing Text

- **Each character (letter, punctuation, etc.) is assigned a unique bit pattern.**
  - ASCII: Uses patterns of 8 bits to represent most symbols used in written English text
  - Unicode: Uses patterns of 16 bits to represent the major symbols used in languages world wide

1-27

The message “Hello.” in ASCII

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

1-28

Representing Numeric Values

- Binary notation: Uses bits to represent a number in *base two*
  - Instead of ASCII symbols like '2' and '7'
- 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

https://studio.code.org/s/odometer/stage/1/puzzle/1

1-29

Representing Images

- Bit map techniques
  - Pixel: short for "picture element"
  - RGB
- Vector techniques
  - Geometric shapes
  - Scalable
  - E.g., TrueType and PostScript fonts

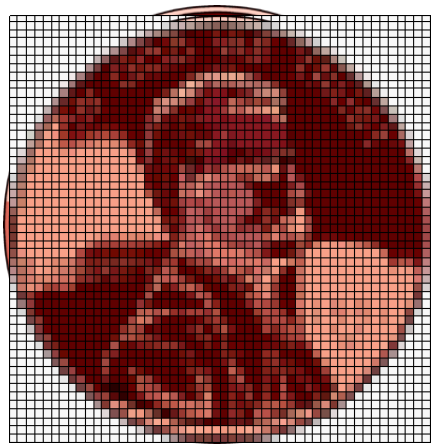
1-30

Lights, Sound, Magic

- Representation
  1. Objective is to store as binary numbers
  2. Since the information is analog, must use approximation for representation
     a. Sampling
     b. Quantizing
  3. The quality of this approximation depends on
     1. Resolution
     2. Dynamic range
     3. Mode (Grayscale or color)

Light, Sound, Magic…

- Sampling
  - Grid of *pixels*
  - Resolution
    - More sampling → higher quality
- Quantizing
  - Each sample assigned a value
  - Dynamic Range
    - Range of values used
    - i.e. # bits used per pixel
    - More bits → higher quality



Changes in Resolution

- 272 x 416 136 x 208 68 x 104

Changes to Dynamic Range

Grayscale images

- Black & white images
  1) Simplest form: each pixel either 0 (black) or 1 (white)
     - Binary image
     - 1 bitplane dynamic range
     - Requires little storage space, but only appropriate for images with no shades of gray
  2) Grayscale images
     - Use *range* of values for each pixel
     - Range from black to white
     - Able to represent more intensities
     - More memory

Color images

- Pixels contain *ordered sets* of values
- Different strategies exist
  1) RGB color (most common)
     - Natural color represented as a combination of three *channels*
       - Red, Green, Blue
       - Each has its own *dynamic range*
     - RGB are *additive* primaries
       - Combining different colors of *light*
       - Start from black (0,0,0) which is the absence of light/color (e.g. a turned-off monitor)
       - All light combined is white
     - RGB color is employed by most color video displays

Image Storage

- How many bytes?
  - Resolution: 1024x768
  - Mode: Color (RGB)
  - Dynamic range: 8 bitplanes

$$\frac{1024x768\ pixels}{picture} \times \frac{3\ values\ (RGB)}{pixel} \times \frac{8\ bits}{value} \times \frac{1\ byte}{8\ bits} = 2359296 \text{ bytes}$$

- Higher resolution? Grayscale? Lower dynamic range (posterized)?

Representing Sound

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

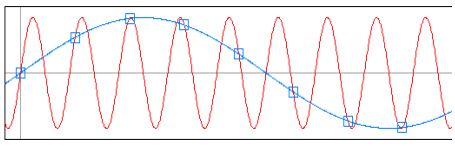
1-38

Basic Audio

- Waves of ***air pressure*** due to vibration
- Wave cycles per second is the ***frequency*** of sound (measured in ***Hertz –*** *1 cycle per second*)
  - Low frequency == bass; High frequency == piccolo
  - A above middle C is 440 Hz
  - Human Speech – less than 4 KHz
    - 4 K**Hz** == 4000 wave cycles per second
  - *Pitch* of a sound
- Human ear can hear between 20Hz to 20,000 Hz

Digital Sound

- Sample and Quantize
  - Detect sound pressure at regular intervals
  - Assign a numeric value
  - Reproduce by “connecting the dots”
- Sampling Rate
  - Frequency that we sample sound
  - How good is good enough?
    - Nyquist Rule
    - CD quality is 44.1 KHz
    - Telephone Quality is 8KH



Digital Sound

File Sizes

- 10 seconds high quality digital audio (no compression)

  $\frac{44100 \text{ samples}}{\text{sec}} * \frac{16 \text{ bits}}{\text{sample}} * 10 \text{ sec} * \frac{1 \text{ byte}}{8 \text{ bits}} = 882 \text{ Kbytes}$

- 60 minutes of high quality audio (no compression)

  44100 samples/sec * 16 bits/sample * 3600 sec * 1 byte/8 bits = 317 Mbytes

- 10 Seconds telephone quality speech:

  8 KHz sampling at 8 bits

  8000 samples/sec * 8 bits/sample * 10 sec * 1 byte/8 bits = 80 Kbytes

MIDI File Format

- Don't capture/recreate sound waves…
  - Capture the parameters of playing the notes!
- Notes store the parameters:
  - Pitch (i.e. frequency of note)
  - Velocity (how hard it is played)
  - Others…
- Playback requires matching each note's parameters with choice of an output "instrument"
- Much smaller file sizes

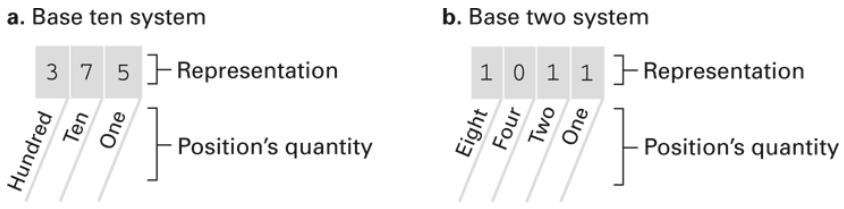
The Binary System

The traditional decimal system is based on powers of ten.

The Binary system is based on powers of two.

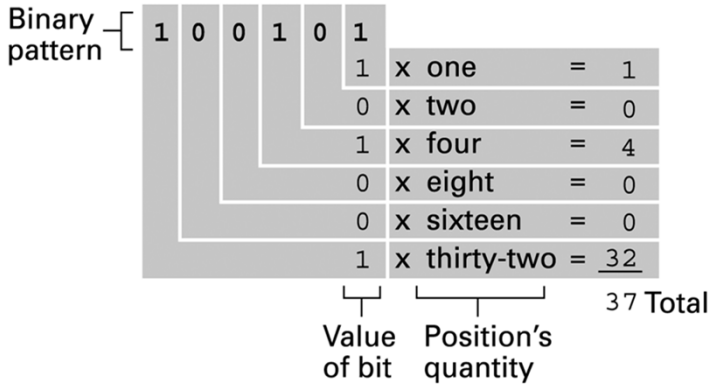
1-43

The base ten and binary systems



1-44

Decoding the binary representation 100101



1-45

An algorithm for finding the binary representation of a positive integer

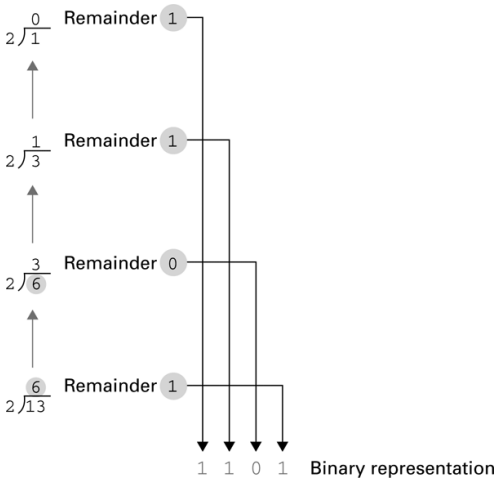
Step 1. Divide the value by two and record the remainder.

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.

1-46

Applying the algorithm to obtain the binary representation of thirteen



1-47

The binary addition facts

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

1-48

Binary addition

- Doing Math in Binary – base 2

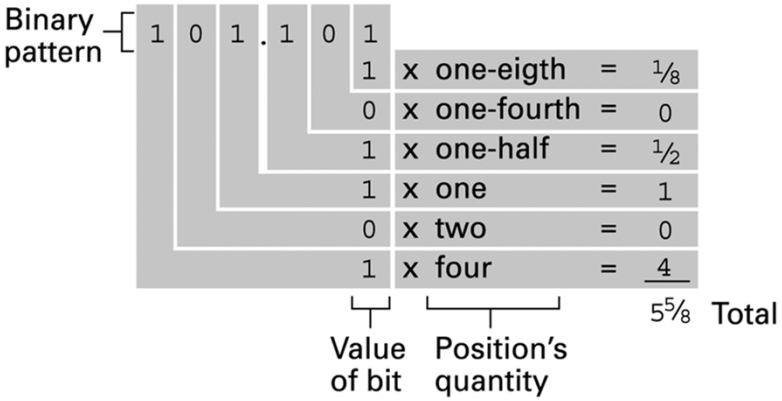
```
  00001001   (9)
+ 00000100   (4)   Simply add digits position-by-position
  00001101   (13)
```

```
    1
  00000111
+ 00000101
    0
```

What about "carry" situations?
1 + 1 = 2?? (Binary for 2 is 10)

Enter the first digit value and "carry" the second

Decoding the binary representation 101.101



1-50

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.

1-51

Two's complement notation systems

a. Using patterns of length three

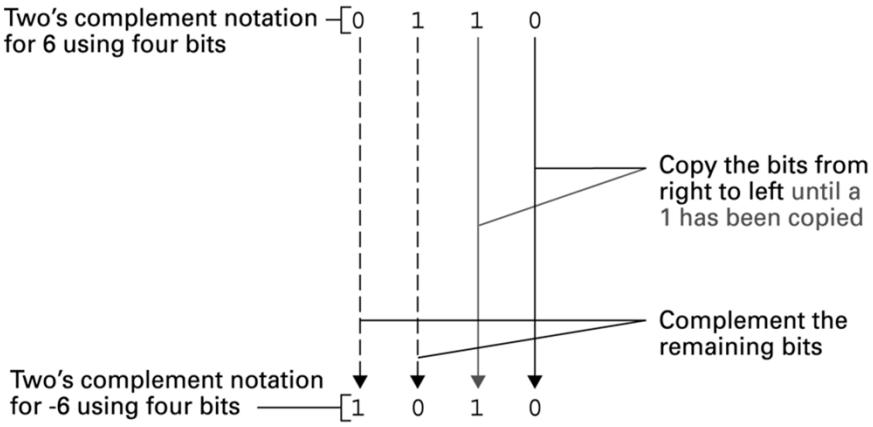
Bit pattern	Value represented
011	3
010	2
001	1
000	0
111	-1
110	-2
101	-3
100	-4

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

1-52

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



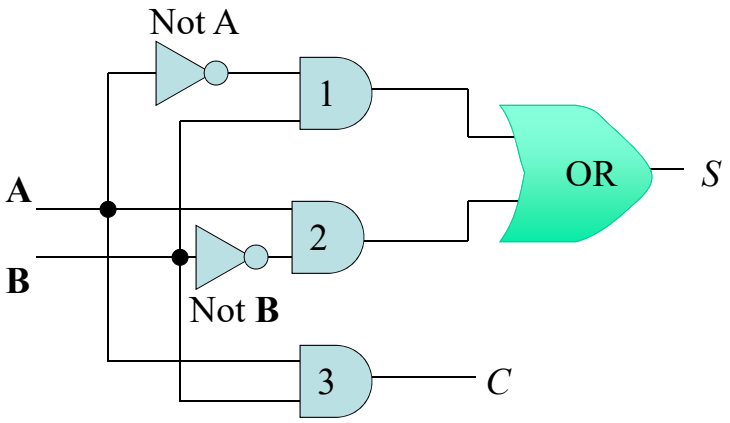
1-53

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

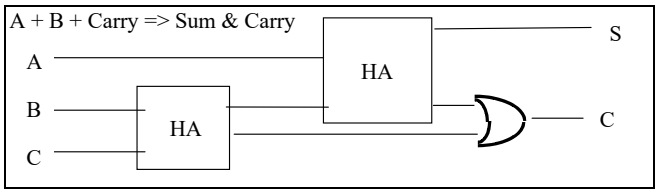
1-54

Circuit review: What does it do?



Adding Numbers

Full adder circuit: adding three single bits



How would you add two 8-bit integers?

→ Cascading adder

Storing Fractions

- **Floating-point Notation:** Consists of a sign bit, a mantissa field, and an exponent field.
- Related topics include
  - Normalized form
  - Truncation errors

1-57

Digital Representation

- All data is digitized into some pattern of symbols
- Meaning of the pattern depends on how we *interpret* the representation
- What does 0110 0001 0010 0101 1010 1001 … represent?
  - Could be text: a%©
  - Could be three ***unsigned*** integers: 97, 37, 169
  - Could be three ***signed*** integers: 97, 37, -87
  - Could be colors for one pixel: R:97 G:37: B169 = ■
  - Could be ???

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.)

1-59

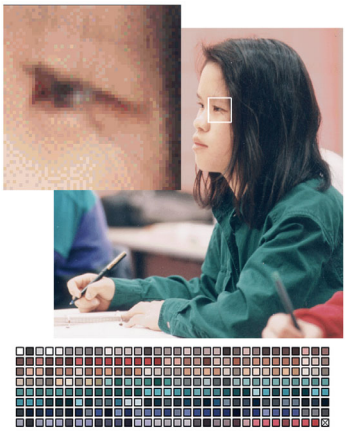
Compressing Images

- GIF: indexed color
  - Good for logos, icons, cartoons, etc.
- JPEG: lossy compression
  - Good for photographs, natural images
- TIFF: optional loss-less compression
  - Good for image archiving

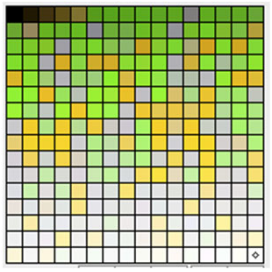
1-60

Color images

4) Indexed color
   - Only one channel used, rather than three or four
     - Only one number to store
   - Strategy for sacrificing quality for storage size
     - Smaller, more compact
   - Images are derived from full color images
   - Composed of pixels selected from a limited palette of colors or shades
   - Can convert from RGB, etc., but color information is lost



Color images



Compressing Audio and Video

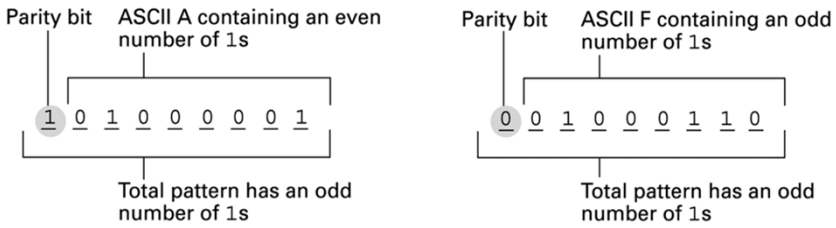
- MPEG
  - High definition television broadcast
  - Video conferencing
- MP3
  - Temporal masking
  - Frequency masking

Communication Errors

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

1-64

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



1-65

An error-correcting code

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

1-66

Decoding the pattern 010100 using the Hamming code in Figure 1.30

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**	Smallest distance
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	

1-67