16.070

Introduction to Computers & Programming

Machine architecture:
the binary system, storing numbers, data compression

Prof. Kristina Lundqvist
Dept. of Aero/Astro, MIT
The Binary System

- **Decimal**: Position represents a power of 10
  - \( 5382_{10} = 5 \times 10^3 + 3 \times 10^2 + 8 \times 10^1 + 2 \times 10^0 \)

- **Binary**: Position represents a power of 2
  - \( 1011_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \)
  - \( = 8 + 0 + 2 + 1 = 11_{10} \)

- **Binary addition**

  \[
  \begin{array}{ccccccc}
  & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
  + & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
  \hline
  & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
  \end{array}
  \]

  bit number: 7 6 5 4 3 2 1 0

- **A byte**

<table>
<thead>
<tr>
<th>bit value</th>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 0 1 0 0 1 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

16.070 — February 14/2003 — Prof. I. K. Lundqvist — kristina@mit.edu
Representing Negative Numbers

- Using one byte, any suggestions for representing negative numbers?
  - E.g., 00010011₂ = 19₁₀. How to represent -19₁₀ in binary?

- Reserve 1 bit (#7) for sign, 7 bits for number (sign magnitude)
  - 00000001₂ = 1₁₀
  - 10000001₂ = -1₁₀
  - 00010011₂ = 19₁₀
  - 10010011₂ = -19₁₀
Representing Negative Numbers

- **One’s complement** – invert each bit
  - \(00010011_2 = 19_{10}\)
  - \(11101100_2 = -19_{10}\)
  - \(0_{10}: 00000000_2 \text{ and } 11111111_2\)

- **Two’s complement** – invert each bit and add 1
  - \(00010011_2 = 19_{10}\)
  - \(11101101_2 = -19_{10}\)
  - Try to negate 0:
    - \(0_{10} = 00000000_2\)
    - invert: \(00000000_2 \rightarrow 11111111_2\)
    - add 1: \(11111111_2 + 00000001_2 = 000000000_2\)
# Two’s Complement Notation Systems

<table>
<thead>
<tr>
<th>Bit pattern</th>
<th>Value represented</th>
</tr>
</thead>
<tbody>
<tr>
<td>011</td>
<td>3</td>
</tr>
<tr>
<td>010</td>
<td>2</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
</tr>
<tr>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>111</td>
<td>-1</td>
</tr>
<tr>
<td>110</td>
<td>-2</td>
</tr>
<tr>
<td>101</td>
<td>-3</td>
</tr>
<tr>
<td>100</td>
<td>-4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit pattern</th>
<th>Value represented</th>
</tr>
</thead>
<tbody>
<tr>
<td>0111</td>
<td>7</td>
</tr>
<tr>
<td>0110</td>
<td>6</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>1111</td>
<td>-1</td>
</tr>
<tr>
<td>1110</td>
<td>-2</td>
</tr>
<tr>
<td>1101</td>
<td>-3</td>
</tr>
<tr>
<td>1100</td>
<td>-4</td>
</tr>
<tr>
<td>1011</td>
<td>-5</td>
</tr>
<tr>
<td>1010</td>
<td>-6</td>
</tr>
<tr>
<td>1001</td>
<td>-7</td>
</tr>
<tr>
<td>1000</td>
<td>-8</td>
</tr>
</tbody>
</table>
## Addition

<table>
<thead>
<tr>
<th>Problem in base ten</th>
<th>Problem in two's complement</th>
<th>Answer in base ten</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 + 2</td>
<td>0011 + 0010 = 0101</td>
<td>5</td>
</tr>
<tr>
<td>-3 + -2</td>
<td>1101 + 1110 = 1011</td>
<td>-5</td>
</tr>
<tr>
<td>7 + -5</td>
<td>0111 + 1011 = 0010</td>
<td>2</td>
</tr>
</tbody>
</table>
Example: we are using 8 bit two's complement and have the problem 97-81.

\[
\begin{array}{c|cccccccc}
97 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
-81 & +1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\
\hline
16 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}
\]
Summary: One’s/ Two’s Complement

- Note that in *sign magnitude* and in *one's complement* there are two possible representations of 0, and we can represent every integer $n$ with $-(2^k - 1) \leq n \leq 2^k - 1$ with a k-bit field.

- With *two's complement* there is only one representation of 0, and we can represent the integers $n$ with $-2^k \leq n \leq 2^k - 1$.

- The most significant advantage of two's complement is the fact that subtraction can be performed with addition circuitry.
The Problem of Overflow

- Overflow: when a value to be represented falls outside the range of values that can be represented.
- An overflow is indicated if the addition of two positive values results in the pattern for a negative value or vice versa.
- Remember: small values can accumulate to produce large numbers.
Fractions in Binary

- Each position is assigned a quantity of twice the size of the one to its right.
- \[101.101 = ?\]

<table>
<thead>
<tr>
<th>Binary pattern</th>
<th>[2^2]</th>
<th>[2^1]</th>
<th>[2^0]</th>
<th>[-1]</th>
<th>[-2]</th>
<th>[-3]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- \(x\) one-eighth = \(\frac{1}{8}\)
- \(x\) one-fourth = 0
- \(x\) one-half = \(\frac{1}{2}\)
- \(x\) one = 1
- \(x\) two = 0
- \(x\) four = 4

\[
\begin{array}{c}
10.001 \\
+ \ 100.000 \\
\hline
111.001
\end{array}
\]

Value of bit | Position's quantity | \(5\frac{5}{8}\) | Total
Floating Point representation

- To represent a floating point number the number is divided into two parts: the integer and the fraction
  - 3.1415 has integer 3 and fraction 0.1415
- Converting FP to binary:
  - Convert integer part to binary
  - Convert fraction part to binary
  - Put a decimal point between the two numbers
- $71.3125 = +1000111.0101$
Normalization

- To represent +1000111.0101
  - Store sign, all bits, and the position of decimal point

- Instead we use Normalization

  - Move the decimal point so that there is only one 1 to the left of the decimal point.

  - To indicate the original value of the number, multiply by $2^e$ where $e$ is the number of bits that the decimal point moved, positive for left movement, negative for right movement

- Store:
  - The sign
  - The exponent
  - The mantissa
Excess (or bias) Notation

- Alternative to two’s complement used to store the exponent for floating point numbers.

<table>
<thead>
<tr>
<th>Bit pattern</th>
<th>Value represented</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>3</td>
</tr>
<tr>
<td>110</td>
<td>2</td>
</tr>
<tr>
<td>101</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>011</td>
<td>−1</td>
</tr>
<tr>
<td>010</td>
<td>−2</td>
</tr>
<tr>
<td>001</td>
<td>−3</td>
</tr>
<tr>
<td>000</td>
<td>−4</td>
</tr>
</tbody>
</table>

With 3 bits we have excess 4 \((=2^{3-1})\)
Excess (or bias) Notation

With 4 bits we have excess 8 ($=2^{4-1}$)

With N bits we have excess $2^{N-1}$

We add the magic number, $2^{N-1}$, to the integer, change the result to binary, and add 0’s (on the left) to make up N bits.

<table>
<thead>
<tr>
<th>Bit pattern</th>
<th>Value represented</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>7</td>
</tr>
<tr>
<td>1110</td>
<td>6</td>
</tr>
<tr>
<td>1101</td>
<td>5</td>
</tr>
<tr>
<td>1100</td>
<td>4</td>
</tr>
<tr>
<td>1011</td>
<td>3</td>
</tr>
<tr>
<td>1010</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td>0111</td>
<td>-1</td>
</tr>
<tr>
<td>0110</td>
<td>-2</td>
</tr>
<tr>
<td>0101</td>
<td>-3</td>
</tr>
<tr>
<td>0100</td>
<td>-4</td>
</tr>
<tr>
<td>0011</td>
<td>-5</td>
</tr>
<tr>
<td>0010</td>
<td>-6</td>
</tr>
<tr>
<td>0001</td>
<td>-7</td>
</tr>
<tr>
<td>0000</td>
<td>-8</td>
</tr>
</tbody>
</table>
The standard IEEE representation for FP uses 32 bits for single-precision format:

- 1 bit sign
- 8 bits exponent (excess 127 = $2^{8-1}-1$)
- 23 bits mantissa

Representation. Store the:

- Sign as 0 (positive), 1 (negative)
- Exponent (power of 2) as excess 127
- Mantissa as an unsigned integer
Example: FP Representation

- Consider a 16-bit representation with
  - 1 bit sign
  - 5 bits exponent (excess 16 = $2^{5-1}$)
  - 10 bits mantissa
- 0 10011 1101100000
  - The sign 0 represents positive
  - The exponent 10011 = 19_{10} represents 3
  - The mantissa is .1101100000 (1 to left of . is not stored, it is understood)
Coding the value \(2 \frac{5}{8}\)

Original representation

Base two representation

Raw bit pattern

Sign bit

Exponent

Mantissa

Lost bit

Truncation Errors
Data Compression

- Storing data in a format that requires less space than usual
  - E.g., transformation of a string of characters in some representation (such as ASCII) into a new string (of bits, for example) which contains the same information but whose length is as small as possible.

- CCITT (Comité Consultatif International Téléphonique et Télégraphique)
  - Transmitting faxes
  - Communication through modems (CCITT V.42bis)

- File compression formats
  - ZIP
  - ARC (BBSs)

- Also widely used in
  - Backup utilities, spreadsheet applications, database management systems
  - Certain types of data, such as bit mapped graphics, can be compressed to a small fraction of their normal size.
Generic Data Compression Techniques

- Run-length encoding
- Relative encoding
- Frequency-dependent encoding
  - Huffman codes
- Adaptive dictionary encoding
  - Lempel-Ziv encoding
- Graphic Interchange Format
- Joint Photographic Experts Group
Lempel-Ziv Encoding

Decompressing

xyxxxyzy (5, 4, x)

a. Count backward 5 symbols.

```
  x  y  x  x  y  z  y
```

b. Identify the four-bit segment to be appended to the end of the string.

```
  x  y  x  x  y  z  y  x  x  y  z
```

C. Copy the four-bit segment onto the end of the message.

```
  x  y  x  x  y  z  y  x  x  y  z  x
```

d. Add the symbol identified in the triple to the end of the message.
Error Detection and Correction

- Errors in data representation can be caused by factors such as noisy transmission (phone lines, radio links), power surges, …
- Bits can flip during transmission or on orbit inside spacecraft processors
- What if one bit is in the wrong state?
  - One bit flip can have dramatic ramifications
  - Must develop/use routines to
    - Detect single bit errors
    - Correct for single bit errors (for system critical info)
    - Make sure communication link and on-board processors are robust enough that more than one error per word is extremely unlikely
Communication Errors

- **Parity bit**
  - ASCII A containing an even number of 1s
    - Parity bit 1
    - Total pattern has an odd number of 1s
  - ASCII F containing an odd number of 1s
    - Parity bit 0
    - Total pattern has an odd number of 1s

- **Odd** parity
- **Even** parity

- **Checksum**
Error-Correcting Codes

- Hamming distance between two bit patterns = number of bits that differ
  - Hamming distance between A and B is 4
  - Hamming distance between B and C is 3

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>000000</td>
</tr>
<tr>
<td>B</td>
<td>001111</td>
</tr>
<tr>
<td>C</td>
<td>010011</td>
</tr>
<tr>
<td>D</td>
<td>011100</td>
</tr>
<tr>
<td>E</td>
<td>100110</td>
</tr>
<tr>
<td>F</td>
<td>101001</td>
</tr>
<tr>
<td>G</td>
<td>110101</td>
</tr>
<tr>
<td>H</td>
<td>111010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Character</th>
<th>Distance between the received pattern and the character being considered</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>1 (Smallest distance)</td>
</tr>
<tr>
<td>E</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>5</td>
</tr>
<tr>
<td>G</td>
<td>2</td>
</tr>
<tr>
<td>H</td>
<td>4</td>
</tr>
</tbody>
</table>
Evening Quiz

- Day: Tuesday
- Date: Apr. 8, 2003
- Time: 7:30p – 9:30p
- Room: 35-225

- Monday lecture Apr. 7, 2003 is canceled
Colors

- Primary colors of **pigment**: Red, Yellow, Blue, that combined produce secondary colors.
- Primary colors of **light**: Red, Green, Blue (RGB).
- Pigment color system does not apply to computer monitors because colors are created on monitors by adding light.
  - RGB is an “additive” color system, i.e., color is added to a black background.

A monochrome monitor can display only two colors, one for the background and one for the foreground.

Color monitors implement the RGB color model by using three different phosphors that appear Red, Green, and Blue when activated. Placing the phosphors directly next to each other, and activating them with different intensities, colour monitors can create an unlimited number of colours. (in practice though, the real number of colors that any monitor can display is controlled by the “video adapter”)