Agenda

• Administrative details

• Course introduction

• Information representation and definitions
Instructor

Prof. Martha Kim
mak2191@columbia.edu
CSB 461

Office hours: Tuesdays and Thursdays, 2-3pm
(Email or drop by to schedule other times.)
Teaching assistants

Roopa Kakarlapudi
Nishant Shah
Harsh Parekh
Lectures

Mondays and Wednesdays

1:10-2:25pm

Fayerweather 310
Textbooks


Grading formula

Final Exam
• early May (scheduled by University)
• covers 2nd half of course

Midterm Exam
• early March (before spring break)
• covers 1st half of course

Eight problem sets
• handful of practice problems
• one week to complete
Problem sets

Due at start of class on due date.

**Collaboration policy:** In working on the problem sets, feel free to discuss the problems with your classmates. However, no collaboration is allowed in writing up the solutions. Each student is to write up his or her own solution and is expected to be able to explain and reproduce the work she or she submits.
Course webpage

http://www1.cs.columbia.edu/~martha/courses/3827/sp09/
Agenda

• Administrative details

• Course introduction

• Information representation and definitions
What does this ...

... have in common with this?
growth in performance = growth in raw resources + system design innovation

<table>
<thead>
<tr>
<th>ENIAC (1946)</th>
<th>Intel Larrabee (2009)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5,000 operations per second</td>
<td>2,000,000,000,000 operations per second</td>
<td>400,000,000x faster</td>
</tr>
<tr>
<td>8.5’ x 3’ x 80’ (2040 ft³)</td>
<td>49.5 mm²</td>
<td>1,167,000,000x smaller</td>
</tr>
<tr>
<td>$500,000</td>
<td>~$300</td>
<td>1666x cheaper</td>
</tr>
</tbody>
</table>
growth in performance = growth in raw resources + system design innovation

Gordon Moore
co-founder of Intel

Moore’s Law:
Density of transistors doubles every two years

CPU Transistor Counts 1971-2008 & Moore’s Law
growth in performance = growth in raw resources + system design innovation
Agenda

- Administrative details
- Course introduction
- Information representation and definitions
Number systems: Base 10 (Decimal)

- 10 digits = \{0,1,2,3,4,5,6,7,8,9\}

- example: $4537.8 = (4537.8)_{10}$

\[
\begin{align*}
4 & \times 10^3 & \times 10^2 & \times 10^1 & \times 10^0 & \times 10^{-1} \\
5 & \times 10^0 & \times 10^0 & \times 10^0 & \times 10^0 & \times 10^0 \\
3 & \times 10^0 & \times 10^0 & \times 10^0 & \times 10^0 & \times 10^0 \\
7 & \times 10^0 & \times 10^0 & \times 10^0 & \times 10^0 & \times 10^0 \\
. & \times 10^0 & \times 10^0 & \times 10^0 & \times 10^0 & \times 10^0 \\
8 & \times 10^0 & \times 10^0 & \times 10^0 & \times 10^0 & \times 10^0 \\
\end{align*}
\]
\[
4000 + 500 + 40 + 7 + .8 = 4537.8
\]
Number systems: Base 2 (Binary)

- 2 digits = \{0,1\}

- example: \(1011.1 = (1011.1)_2\)

\[
\begin{array}{cccccc}
1 & 0 & 1 & 1 & 1 & . & 1 \\
\times 2^3 & \times 2^2 & \times 2^1 & \times 2^0 & \times 2^{-1} \\
\hline
8 & 0 & 2 & 1 & .5 \\
\end{array}
\]

\(= (11.5)_10\)
Number systems: Base 8 (Octal)

- 8 digits = \{0,1,2,3,4,5,6,7\}

- example: \((2365.2)_8\)

\[
\begin{align*}
2 &\times 8^3 &\quad 3 &\times 8^2 &\quad 6 &\times 8^1 &\quad 5 &\times 8^0 &\quad .2 &\times 8^{-1} \\
1024 &+ &192 &+ &48 &+ &5 &+ &.25 = (1269.25)_{10}
\end{align*}
\]
Number systems: Base 16 (Hexadecimal)

• 16 digits = \{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}

• example: \( (26BA)_{16} \) \quad \text{[alternate notation for hex: 0x26BA]}

\[
\begin{align*}
2 & \times 16^3 & 6 & \times 16^2 & B & \times 16^1 & A & \times 16^0 \\
8192 & + 1536 & + 176 & + 10 & = (9914)_{10}
\end{align*}
\]
Hexadecimal (or hex) is often used for addressing.
Number ranges

• Map infinite numbers onto finite representation for a computer

• How many numbers can I represent with ...

  ... 5 digits in decimal?  $10^5$ possible values

  ... 8 binary digits?  $2^8$ possible values

  ... 4 hexadecimal digits?  $16^4$ possible values
Need a bigger range?

• Change the encoding.

• Floating point (used to represent very large numbers in a compact way)

  • A lot like scientific notation: $5.4 \times 10^5$

  • Except that it is binary: $1001 \times 2^{1011}$
What about negative numbers?

• Change the encoding.
  • Sign and magnitude
  • Ones compliment
  • Twos compliment
Sign and magnitude

- Most significant bit is sign

- Rest of bits are magnitude

\[
\begin{align*}
0110 &= (6)_{10} \\
1110 &= (-6)_{10}
\end{align*}
\]

- Two representations of zero

\[
\begin{align*}
0000 &= (0)_{10} \\
1000 &= (-0)_{10}
\end{align*}
\]
Ones compliment

• Compliment bits in positive value to create negative value

• Most significant bit still a sign bit

\[ 0110 = (6)_{10} \quad 1001 = (-6)_{10} \]

• Two representations of zero

\[ 0000 = (0)_{10} \quad 1111 = (-0)_{10} \]
Twos compliment

- Compliment bits in positive value and add 1 to create negative value

- Most significant bit still a sign bit

\[
\begin{align*}
0110 &= (6)_{10} & 1001 + 1 &= 1010 = (-6)_{10} \\
0000 &= (0)_{10} & 1000 &= (-8)_{10} & 1111 &= (-1)_{10}
\end{align*}
\]

- One representation of zero

- One more negative number than positive

MIN: 1000 = (-8)_{10} \quad MAX: 0111 = (7)_{10}
How about letters?

- Change the encoding.

```
<table>
<thead>
<tr>
<th>( B_4B_3B_2B_1 )</th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>NULL</td>
<td>DLE</td>
<td>SP</td>
<td>0</td>
<td>@</td>
<td>P</td>
<td>'</td>
<td>p</td>
</tr>
<tr>
<td>0001</td>
<td>SOH</td>
<td>DC1</td>
<td>!</td>
<td>1</td>
<td>A</td>
<td>Q</td>
<td>a</td>
<td>q</td>
</tr>
<tr>
<td>0010</td>
<td>STX</td>
<td>DC2</td>
<td>&quot;</td>
<td>2</td>
<td>B</td>
<td>R</td>
<td>b</td>
<td>r</td>
</tr>
<tr>
<td>0011</td>
<td>ETX</td>
<td>DC3</td>
<td>#</td>
<td>3</td>
<td>C</td>
<td>S</td>
<td>c</td>
<td>s</td>
</tr>
<tr>
<td>0100</td>
<td>EOT</td>
<td>DC4</td>
<td>$</td>
<td>4</td>
<td>D</td>
<td>T</td>
<td>d</td>
<td>t</td>
</tr>
<tr>
<td>0101</td>
<td>ENQ</td>
<td>NAK</td>
<td>%</td>
<td>5</td>
<td>E</td>
<td>U</td>
<td>e</td>
<td>u</td>
</tr>
<tr>
<td>0110</td>
<td>ACK</td>
<td>SYN</td>
<td>&amp;</td>
<td>6</td>
<td>F</td>
<td>V</td>
<td>f</td>
<td>v</td>
</tr>
<tr>
<td>0111</td>
<td>BEL</td>
<td>ETB</td>
<td>'</td>
<td>7</td>
<td>G</td>
<td>W</td>
<td>g</td>
<td>w</td>
</tr>
<tr>
<td>1000</td>
<td>BS</td>
<td>CAN</td>
<td>(</td>
<td>8</td>
<td>H</td>
<td>X</td>
<td>h</td>
<td>x</td>
</tr>
<tr>
<td>1001</td>
<td>HT</td>
<td>EM</td>
<td>)</td>
<td>9</td>
<td>I</td>
<td>Y</td>
<td>i</td>
<td>y</td>
</tr>
<tr>
<td>1010</td>
<td>LF</td>
<td>SUB</td>
<td>*</td>
<td>:</td>
<td>J</td>
<td>Z</td>
<td>j</td>
<td>z</td>
</tr>
<tr>
<td>1011</td>
<td>VT</td>
<td>ESC</td>
<td>+</td>
<td>;</td>
<td>K</td>
<td>[</td>
<td>k</td>
<td>{</td>
</tr>
<tr>
<td>1100</td>
<td>FF</td>
<td>FS</td>
<td>,</td>
<td>&lt;</td>
<td>L</td>
<td>\</td>
<td>l</td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td>CR</td>
<td>GS</td>
<td>-</td>
<td>=</td>
<td>M</td>
<td>]</td>
<td>m</td>
<td>}</td>
</tr>
<tr>
<td>1110</td>
<td>SO</td>
<td>RS</td>
<td>.</td>
<td>&gt;</td>
<td>N</td>
<td>^</td>
<td>n</td>
<td>~</td>
</tr>
<tr>
<td>1111</td>
<td>SI</td>
<td>US</td>
<td>/</td>
<td>?</td>
<td>O</td>
<td>_</td>
<td>o</td>
<td>DEL</td>
</tr>
</tbody>
</table>
```

**TABLE 1-5**
American Standard Code for Information Interchange (ASCII)
Some definitions

• bit = a binary digit  
  e.g., 1 or 0

• byte = 8 bits  
  e.g., 01100100

• word = a group of bytes  
  a 16-bit word = 2 bytes  
  e.g., 1001110111000101

  a 32-bit word = 4 bytes  
  e.g., 100111011100010101110111000101
Next class: binary logic, logic gates