Computer Architecture
CSCI-UA.0436 Fall 2013

Handout - First Day of Class (9/4/13)

This will be the only paper handout of the semester. All other material will be found on the course web page, http://cs.nyu.edu/courses/fall13/CSCI-UA.0436-001/.

**What you must do before the next class (Monday 9/9)**

- You must install the Logisim logic simulator. See the link on the course web page.

**What you must do before next Wednesday (9/11)**

- You must study the material on the back of this page, covering basic information about numbers that you probably learned in middle school. Although the material is very easy, it's important that you have this information memorized. There will be a short **quiz** on the material on Wednesday. The quiz will only cover what is on the back of this page.
Number Review for Quiz on Wednesday, September 11

Powers of two

\[
egin{array}{|c|c|c|}
\hline
2^0 = 1 & 2^5 = 32 & 2^{10} = 1024 = 1 \text{K} \\
2^1 = 2 & 2^6 = 64 & 2^{20} = 1 \text{M} \\
2^2 = 4 & 2^7 = 128 & 2^{30} = 1 \text{G} \\
2^3 = 8 & 2^8 = 256 & 2^{40} = 1 \text{T} \\
2^4 = 16 & 2^9 = 512 & \\
\hline
\end{array}
\]

Rule: \(2^{m+n} = 2^m \times 2^n\)

Useful examples: \(2^{16} = 2^{10} \times 2^6 = 1 \text{K} \times 64 = 64 \text{K}\) \(2^{32} = 2^{30} \times 2^2 = 1 \text{G} \times 4 = 4 \text{G}\)

Logarithms (base 2)

\[
egin{array}{|c|c|c|}
\hline
\log_2 1 = 0 & \log_2 32 = 5 & \log_2 1024 = \log_2 1 \text{K} = 10 \\
\log_2 2 = 1 & \log_2 64 = 6 & \log_2 1 \text{M} = 20 \\
\log_2 4 = 2 & \log_2 128 = 7 & \log_2 1 \text{G} = 30 \\
\log_2 8 = 3 & \log_2 256 = 8 & \log_2 1 \text{T} = 40 \\
\log_2 16 = 4 & \log_2 512 = 9 & \\
\hline
\end{array}
\]

Rule: \(\log_2 (m \times n) = \log_2 m + \log_2 n\)

Useful examples:

\[
\begin{align*}
\log_2 16 \text{M} &= \log_2 16 + \log_2 1 \text{M} = 4 + 20 = 24 \\
\log_2 32 \text{G} &= \log_2 32 + \log_2 1 \text{G} = 5 + 30 = 35
\end{align*}
\]

In a computer system, an n-bit memory location can store one of \(2^n\) different numbers. Example: A 32-bit variable can store any of \(2^{32}\) different numbers.

In a computer system, an n-bit address can refer to any of \(2^n\) different memory locations. Example: A 16-bit address can refer to any of \(2^{16} = 64 \text{K}\) different memory locations.

Number Systems (Decimal, Hexadecimal, Binary)

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Hex</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0001</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0010</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0011</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0100</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0101</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0110</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>0111</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>1000</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>1001</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>1010</td>
</tr>
<tr>
<td>11</td>
<td>B</td>
<td>1011</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>1100</td>
</tr>
<tr>
<td>13</td>
<td>D</td>
<td>1101</td>
</tr>
<tr>
<td>14</td>
<td>E</td>
<td>1110</td>
</tr>
<tr>
<td>15</td>
<td>F</td>
<td>1111</td>
</tr>
</tbody>
</table>

In computer science, the value of a number is often expressed in hex because the translation between binary and hex is very easy. Each hex digit represents four binary digits.

Examples:
The 16-bit number 4AE3 hex corresponds to 0100 1010 1110 0011 in binary. The 32-bit number FFFF hex corresponds to 1111 1111 1111 1111 1111 in binary.