Today’s topics

- Binary Numbers
  - Brookshear 1.1-1.6
- Slides from Prof. Marti Hearst of UC Berkeley SIMS
- Upcoming
  - Networks
    - Interactive Introduction to Graph Theory
      - [http://www.utm.edu/cgi-bin/caldwell/tutor/departments/math/graph/intro](http://www.utm.edu/cgi-bin/caldwell/tutor/departments/math/graph/intro)
  - Problem Solving

Binary Digits (Bits)

- Yes or No
- On or Off
- One or Zero
- 10010010

Digital Computers

- What are computers made up of?
  - Lowest level of abstraction: atoms
  - Higher level: transistors
- Transistors
  - Invented in 1951 at Bell Labs
  - An electronic switch
  - Building block for all modern electronics
  - Transistors are packaged as Integrated Circuits (ICs)
  - 40 million transistors in 1 IC

More on binary

- Byte
  - A sequence of bits
  - 8 bits = 1 byte
  - 2 bytes = 1 word (sometimes 4 or 8 bytes)
- Powers of two
- How do binary numbers work?
Data Encoding

- 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
  - Unicode: Uses patterns of 16-bits to represent the major symbols used in languages worldwide
  - ISO standard: Uses patterns of 32-bits to represent most symbols used in languages worldwide
- Numbers: Uses bits to represent a number in base two
- Limitations of computer representations of numeric values
  - Overflow – happens when a value is too big to be represented
  - Truncation – happens when a value is between two representable values

Decimal (Base 10) Numbers

- Each digit in a decimal number is chosen from ten symbols: \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}
- The position (right to left) of each digit represents a power of ten.
- Example: Consider the decimal number \( 2307 \)

\[
\begin{array}{cccc}
  2 & 3 & 0 & 7 \\
  \uparrow & \uparrow & \uparrow & \uparrow \\
\end{array}
\]

\[
\begin{array}{c}
\text{position:} \\
3 & 2 & 1 & 0 \\
\end{array}
\]

\[
2307 = 2 \times 10^3 + 3 \times 10^2 + 0 \times 10^1 + 7 \times 10^0
\]

Images, Sound, & Compression

- Images
  - Store as bit map: define each pixel
    - RGB
    - Luminance and chrominance
  - Vector techniques
    - Scalable
    - TrueType and PostScript
- Audio
  - Sampling
- Compression
  - Lossless: Huffman, LZW, GIF
  - Lossy: JPEG, MPEG, MP3

Binary (Base 2) Numbers

- Each digit in a binary number is chosen from two symbols: \{ 0, 1 \}
- The position (right to left) of each digit represents a power of two.
- Example: Convert binary number \( 1101 \) to decimal

\[
\begin{array}{cccc}
  1 & 1 & 0 & 1 \\
  \uparrow & \uparrow & \uparrow & \uparrow \\
\end{array}
\]

\[
\begin{array}{c}
\text{position:} \\
3 & 2 & 1 & 0 \\
\end{array}
\]

\[
1101 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
= 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 8 + 4 + 1 = 13
\]
### Powers of Two

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Binary</th>
<th>Power of 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>$2^0$</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>$2^1$</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>$2^2$</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>$2^3$</td>
</tr>
<tr>
<td>16</td>
<td>10000</td>
<td>$2^4$</td>
</tr>
<tr>
<td>32</td>
<td>100000</td>
<td>$2^5$</td>
</tr>
<tr>
<td>64</td>
<td>1000000</td>
<td>$2^6$</td>
</tr>
<tr>
<td>128</td>
<td>10000000</td>
<td>$2^7$</td>
</tr>
</tbody>
</table>

### Famous Powers of Two

<table>
<thead>
<tr>
<th>Size</th>
<th>Binary</th>
<th>Decimal</th>
<th>Hexadecimal</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kilobyte</td>
<td>$2^{10}$ bytes</td>
<td>1024</td>
<td>0x400</td>
<td>1024 bytes</td>
</tr>
<tr>
<td>Megabyte</td>
<td>$2^{20}$ bytes</td>
<td>1,048,576 bytes</td>
<td>0x00010000</td>
<td>Millions of bytes</td>
</tr>
<tr>
<td>Gigabyte</td>
<td>$2^{30}$ bytes</td>
<td>1,073,741,824 bytes</td>
<td>0x000000010000</td>
<td>Trillions of bytes</td>
</tr>
<tr>
<td>Terabyte</td>
<td>$2^{40}$ bytes</td>
<td>1,099,511,627,776 bytes</td>
<td>0x0000000000010000</td>
<td></td>
</tr>
</tbody>
</table>

### Other Number Systems

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

### Binary Addition

<table>
<thead>
<tr>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3</th>
<th>Rule 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Also: $1 + 1 + 1 = 1$ with a carry of 1

Adding Binary Numbers

\[
\begin{array}{c}
101 \\
+ \ 10 \\
\hline
111
\end{array}
\]

- \(101 + 10 = (1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0) + (1 \times 2^1 + 0 \times 2^0)\)
- \(= (1 \times 4 + 0 \times 2 + 1 \times 1) + (1 \times 2 + 0 \times 1)\)
- Add like terms: There is one 4, one 2, one 1
  \(= 1 \times 4 + 1 \times 2 + 1 \times 1 = 111\)

Converting Decimal to Binary

<table>
<thead>
<tr>
<th>Decimal</th>
<th>(\rightarrow) conversion (\rightarrow)</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0 \times 2^0)</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>(1 \times 2^0)</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>(1 \times 2^1 + 0 \times 2^0)</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>(2+1 = 1 \times 2^1 + 0 \times 2^0)</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>(4 = 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0)</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>(5+1 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0)</td>
<td>101</td>
</tr>
<tr>
<td>6</td>
<td>(6+2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0)</td>
<td>110</td>
</tr>
<tr>
<td>7</td>
<td>(7+2+1 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0)</td>
<td>111</td>
</tr>
<tr>
<td>8</td>
<td>(8 = 1 \times 2^2 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0)</td>
<td>1000</td>
</tr>
</tbody>
</table>

Converting Decimal to Binary

- Repeated division by two until the quotient is zero
- **Example:** Convert decimal number 54 to binary

\[
\begin{align*}
54 \div 2 &= 27 \, \text{remainder} 0 \\
27 \div 2 &= 13 \, \text{remainder} 1 \\
13 \div 2 &= 6 \, \text{remainder} 1 \\
6 \div 2 &= 3 \, \text{remainder} 0 \\
3 \div 2 &= 1 \, \text{remainder} 1 \\
1 \div 2 &= 0 \, \text{remainder} 1 \\
\end{align*}
\]

Binary representation of 54 is 110110
**Converting Decimal to Binary**

\[
\begin{align*}
0 & \rightarrow 1 \quad & \text{\bullet} \quad 1 \times 2^5 = 0 \quad & \text{plus 1} \quad \text{thirty-two} \\
2 & \div 1 & \rightarrow 1 \quad & \text{\bullet} \quad 6 \times 2^3 = 1 \quad & \text{plus 1} \quad \text{sixteen} \\
2 & \div 3 & \rightarrow 0 \quad & \text{\bullet} \quad 3 \times 2^2 = 3 \quad & \text{plus 0} \quad \text{eights} \\
2 & \div 6 & \rightarrow 1 \quad & \text{\bullet} \quad 13 \times 2^1 = 6 \quad & \text{plus 1} \quad \text{four} \\
2 & \div 13 & \rightarrow 1 \quad & \text{\bullet} \quad 27 \times 2^0 = 13 \quad & \text{plus 1} \quad \text{two} \\
2 & \div 27 & \rightarrow 0 \quad & \text{\bullet} \quad 54 \times 2^0 = 27 \quad & \text{plus 0} \quad \text{ones} \\
54 & \div 27 & = 2 \quad & \text{\bullet} \quad 54 - 2^5 = 22 \\
22 & \div 2^4 & = 6 \quad & \rightarrow \text{110110} \quad \text{\bullet} \quad \text{Subtracting highest} \\
\text{power of two} \quad & \text{\bullet} \quad 54 - 2^5 = 22 \\
6 & \div 2^2 & = 2 \quad \text{\bullet} \quad 22 - 2^4 = 6 \\
2 & \div 2^1 & = 0 \quad \text{\bullet} \quad 22 - 2^4 = 6 \\
\end{align*}
\]

**Solutions**

- Convert 1011000 to decimal representation
  \[
  1011000 = 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\
  = 1 \times 64 + 0 \times 32 + 0 \times 16 + 1 \times 8 + 0 \times 4 + 0 \times 2 + 0 \times 1 \\
  = 64 + 0 + 0 + 0 + 0 + 0 + 0 = 64
  \]

- Add the binary numbers 1011001 and 10101 and express their sum in binary representation
  \[
  \begin{align*}
  1011001 & + 10101 \\
  \hline
  1101110
  \end{align*}
  \]

**Problems**

- Convert 1011000 to decimal representation

- Add the binary numbers 1011001 and 10101 and express their sum in binary representation

- Convert 77 to binary representation

**Solutions**

- Convert 77 to binary representation
  \[
  \begin{align*}
  77 & \div 2^6 = \rightarrow 1 \\
 31 & \div 2^5 = \rightarrow 0 \\
15 & \div 2^4 = \rightarrow 1 \\
7 & \div 2^3 = \rightarrow 1 \\
3 & \div 2^2 = \rightarrow 1 \\
1 & \div 2^1 = \rightarrow 0 \\
0 & \div 2^0 = \rightarrow 0
  \end{align*}
  \]
  \[
  \text{Binary representation of 77 is 1001101}
  \]

- Add the binary numbers 1011001 and 10101 and express their sum in binary representation

- Convert 77 to binary representation
**Boolean Logic**
- **AND, OR, NOT, NOR, NAND, XOR**
- Each operator has a set of rules for combining two binary inputs
  - These rules are defined in a Truth Table
  - (This term is from the field of Logic)
- Each implemented in an electronic device called a gate
  - Gates operate on inputs of 0’s and 1’s
  - These are more basic than operations like addition
  - Gates are used to build up circuits that
    - Compute addition, subtraction, etc
    - Store values to be used later
    - Translate values from one format to another

**In-Class Questions**
1. How many different bit patterns can be formed if each must consist of exactly 6 bits?
2. Represent the bit pattern 1011010010011111 in hexadecimal notation.
3. Translate each of the following binary representations into its equivalent base ten representation.
   1. 1100
   2. 10.011
   4. Translate 231 in decimal to binary

**Truth Tables**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>