CS 240 Lab 3
Bits

• Logic Diagrams

• Binary and Hexadecimal Numbers

• Review of Two’s Complement and Overflow

• Bitbucket/Mercurial complete Tutorial

• Finish Bit Puzzle Exercises from Lab 2

• Start work on Bits assignment
Logic Diagrams

Not the same as pin-outs! Show information about the logical operation of the device.

Inputs on left side of diagram
Outputs on right
Voltage shown on top
Ground shown on bottom
## Binary and Hexadecimal Numbers

<table>
<thead>
<tr>
<th>Hex</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QD QC QB QA</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>0 0 0 0 1</td>
</tr>
<tr>
<td>2</td>
<td>0 0 1 0 0</td>
</tr>
<tr>
<td>3</td>
<td>0 0 1 1 1</td>
</tr>
<tr>
<td>4</td>
<td>0 1 0 0 0</td>
</tr>
<tr>
<td>5</td>
<td>0 1 0 0 1</td>
</tr>
<tr>
<td>6</td>
<td>0 1 1 0 0</td>
</tr>
<tr>
<td>7</td>
<td>0 1 1 1 1</td>
</tr>
<tr>
<td>8</td>
<td>1 0 0 0 0</td>
</tr>
<tr>
<td>9</td>
<td>1 0 0 0 1</td>
</tr>
<tr>
<td>A</td>
<td>1 0 1 0 0</td>
</tr>
<tr>
<td>B</td>
<td>1 0 1 1 1</td>
</tr>
<tr>
<td>C</td>
<td>1 1 0 0 0</td>
</tr>
<tr>
<td>D</td>
<td>1 1 0 0 1</td>
</tr>
<tr>
<td>E</td>
<td>1 1 1 0 0</td>
</tr>
<tr>
<td>F</td>
<td>1 1 1 1 1</td>
</tr>
</tbody>
</table>
Binary can be converted to decimal using positional representation of powers of 2:

\[ 0111_2 = 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0, \quad \text{result} = 7_{10} \]

Decimal can be also be converted to binary by finding the largest power of 2 which fits, subtract, and repeat with the remainders until remainder is 0 (assigning 1 to the positions where a power of 2 is used):

\[ 6_{10} = 6 - 2^2 = 2 - 2^1 = 0, \quad \text{result} = 0110_2 \]

Hex can be converted to binary and vice versa by grouping into 4 bits.

\[ 11110101_2 = \text{F5}_{16} \quad 37_{16} = 00110111_2 \]
Two’s Complement and Overflow

Given n bits, the range of binary values which can be represented using

**Unsigned representation:** $0 \rightarrow 2^n - 1$

**Signed representation:** $-2^{n-1} \rightarrow 2^{n-1} - 1$, MSB is used for sign

**Two’s Complement** (signed representation):

Most significant /leftmost bit (0/positive, 1/negative)

Example: given a fixed number of 4 bits:

$1000_2$ is negative.

$0111_2$ is positive.

**Overflow**

Given a fixed number of n available bits:

Overflow occurs if a value cannot fit in n bits.

Example: given 4 bits:

The largest negative value we can represent is $-8_{10} (1000_2)$

The largest positive value we can represent is $+7_{10} (0111_2)$
Overflow in Addition

When adding two numbers with the same sign which each can be represented with n bits, the result may cause an overflow (not fit in n bits).

An overflow occurs when adding if:

- Two positive numbers added together yield a negative result, or
- Two negative numbers added together yield a positive result, or
- The Cin and Cout bits to the most significant pair of bits being added are not the same.

An overflow cannot result if a positive and negative number are added.

Example: given 4 bits:

\[
\begin{array}{c}
0111_2 \\
+ 0001_2 \\
\hline
1000_2 \quad \text{overflow} \quad \text{NOTE: there is not a carry-out!}
\end{array}
\]

In two’s complement representation, a carry-out does not indicate an overflow, as it does in unsigned representation.

Example: given 4 bits,

\[
\begin{array}{c}
1001_2 (-7_{10}) \\
+ 1111_2 (-1_{10}) \\
\hline
1 1000_2 (-8_{10}) \quad \text{no overflow, even though there is a carry-out}
\end{array}
\]