Review Problem 37

- If all gates have a delay of 1ns, how long does a 4-bit adder take to compute?
Negative Numbers

- Need an efficient way to represent negative numbers in binary
  - Both positive & negative numbers will be strings of bits
  - Use fixed-width formats (4-bit, 16-bit, etc.)
- Must provide efficient mathematical operations
  - Addition & subtraction with potentially mixed signs
  - Negation (multiply by -1)
Sign/Magnitude Representation

High order bit is sign: 0 = positive (or zero), 1 = negative

Three low order bits is the magnitude: 0 (000) thru 7 (111)

Number range for n bits = +/-2^{n-1} - 1

Representations for 0:
Sign/Magnitude Addition

Same sign -> add as usual, keep sign:

\[
\begin{array}{c}
0 \ 0 \ 1 \ 0 \ (2) \\
+ \ 0 \ 1 \ 0 \ 0 \ (4)
\end{array}
\]

Different sign -> sub small mag from larger, keep sign of smaller:

\[
\begin{array}{c}
0 \ 0 \ 1 \ 0 \ (2) \\
+ \ 1 \ 1 \ 0 \ 0 \ (-4)
\end{array}
\]

Bottom line: Basic mathematics are too complex in Sign/Magnitude
Idea: Pick negatives so that addition works

- Let \(-1 = 0 - (+1)\):

\[
\begin{array}{c}
0 \\
- 0 0 0 1 \\
\hline
1 1 1 1 \text{ done } -1
\end{array}
\]

- Does addition work?

\[
\begin{array}{c}
\times \\
0 0 1 0 \\
+ 1 1 1 1 \\
\hline
0 0 0 1
\end{array}
\]

- Result: Two’s Complement Numbers
Two’s Complement

- Only one representation for 0
- One more negative number than positive number
- Fixed width format for both pos. & neg. numbers
Negating in Two's Complement

- Flip bits & Add 1
  - Negate \((0010)_2\) (+2)
    
    \[-(0010) = 1101 + 1 = 1110\]

- Negate \((1110)_2\) (-2)
  
  \[-(1110) = 0001 + 1 = 0010\]
Addition in Two's Complement

\[
\begin{array}{c}
\times 0000 \\
0010 (+2) \\
+ 0100 (+4) \\
\hline
0110 +6 \\
\hline
\checkmark
\end{array}
\quad
\begin{array}{c}
\times 1000 \\
1110 (-2) \\
+ 1100 (-4) \\
\hline
1010 = -(-1010) \\
\hline
\checkmark
\end{array}
\]

\[
\begin{array}{c}
\times 0000 \\
0010 (+2) \\
+ 1100 (-4) \\
\hline
1110 -2\checkmark \\
\hline
\checkmark
\end{array}
\quad
\begin{array}{c}
\times 1000 \\
1110 (-2) \\
+ 0100 (+4) \\
\hline
0010 \\
+ 2 \\
\hline
\checkmark
\end{array}
\]

\[
\begin{array}{c}
\times 1000 \\
1110 (-2) \\
+ 0100 (+4) \\
\hline
0010 \\
+ 2 \\
\hline
\checkmark
\end{array}
\]

\[
\begin{array}{c}
\times 1000 \\
1110 (-2) \\
+ 0100 (+4) \\
\hline
0010 \\
+ 2 \\
\hline
\checkmark
\end{array}
\]

\[
\begin{array}{c}
\times 1000 \\
1110 (-2) \\
+ 0100 (+4) \\
\hline
0010 \\
+ 2 \\
\hline
\checkmark
\end{array}
\]

\[
\begin{array}{c}
\times 1000 \\
1110 (-2) \\
+ 0100 (+4) \\
\hline
0010 \\
+ 2 \\
\hline
\checkmark
\end{array}
\]
A - B = A + (-B) = A + \overline{B} + 1

\[2 - 6\]

\[0010 - 0110\]

\[-5 - (-7)\]

\[1011 - 1001\]

\[-5 - 1\]

\[1011 - 0001\]

\[-5 - 1 = -5 + (-1)\]

\[\text{flip the bits}\]

\[\text{add}\]

\[1100\]

\[\text{flip the bits}\]

\[\text{add}\]

\[1100\]

\[\text{add}\]

\[1100\]

\[\text{flip the bits}\]

\[\text{add}\]

\[1100\]

\[\text{add}\]

\[1100\]

\[\text{flip}\]

\[-6\]
Overflows in Two’s Complement

Add two positive numbers but get a negative number

or two negative numbers but get a positive number

\[ 5 + 3 = -8 \]

\[ -7 - 2 = +7 \]
### Overflow Detection in Two’s Complement

<table>
<thead>
<tr>
<th>Value</th>
<th>Binary</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0101</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td></td>
</tr>
<tr>
<td>-8</td>
<td></td>
<td>Overflow</td>
</tr>
<tr>
<td>-7</td>
<td>1001</td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>1110</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>No overflow</td>
</tr>
<tr>
<td>-3</td>
<td>1101</td>
<td></td>
</tr>
<tr>
<td>-5</td>
<td>1011</td>
<td></td>
</tr>
<tr>
<td>-8</td>
<td>1000</td>
<td></td>
</tr>
</tbody>
</table>

- **Overflow**: Values exceeding the range of the number of bits used.
- **No overflow**: Values within the range of the number of bits used.
Val (→) \text{Out} \\
\text{Ctrl}

<table>
<thead>
<tr>
<th>Ctrl1</th>
<th>Val</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Optional inversion
Adder/Subtractor

A - B = A + (-B) = A + B + 1

\[\text{Overflow}\]