Review of SM.

Signed magnitude notation is not used in modern computers except for in the floating point unit.

Signed magnitude is just that; a signed magnitude tells what the sign is. Like 2's complement, we need to know the number of bits.

**Ex.** 4-bit SM.

<table>
<thead>
<tr>
<th>SB</th>
<th>Sign Bit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Negative</td>
<td>1010</td>
</tr>
<tr>
<td>0</td>
<td>Positive</td>
<td>0110</td>
</tr>
</tbody>
</table>

Given any SM number, to find the decimal equivalent, first determine the sign, then remove the sign bit. Now simply convert the remaining unsigned binary number.

Using the example above.

Sign is 1010 \( \frac{1}{2} \) Thus 1010 is -2

Going into SM notation is very similar. Simply determine the unsigned binary representation of the number and then stick a 0 or 1 in the MSB for the sign.

**Ex.** 5-bit SM notation

Given \(-7\) \(\rightarrow\) 00111 \(\rightarrow\) 10111 = -7 in SM.

Determine binary rep. of \(-7\), make MSB 1 since negative.

**Ex.** 5-bit SM notation

Given \(11\) \(\rightarrow\) 01011 = 11 in SM.

Determine binary rep. of 11.
Addition: Unlike 2's complement addition, the analysis of the numbers must first be done to determine the operation and result of sign.

If the sign of numbers is different then the problem is not an addition, but a subtraction.

ex/ 4-bits
\[
\begin{array}{c}
\text{sign differences} \\
+0100 \\
+1011 \\
\Rightarrow \\
\text{convert to subtraction} \\
-0011
\end{array}
\]

The above problem had the top bit's magnitude larger, if this was not the case, we would have to switch the numbers and make the output the sign of the larger #.

ex/ 4-bits
\[
\begin{array}{c}
\text{sign differences} \\
+0100 \\
+1010 \\
\Rightarrow \\
\text{convert to subtraction} \\
-0110
\end{array}
\]

If the signs are the same then all like unsigned addition, the magnitudes only.

ex/ 4-bits
\[
\begin{array}{c}
\text{signs are same} \\
+0100 \\
+0101 \\
\Rightarrow \\
\text{the same} \\
+0101
\end{array}
\]

Overflow: In addition, overflow if there is a carry out in the addition of the magnitudes.

ex/ 4-bits
\[
\begin{array}{c}
\text{no carry out, over flowed} \\
+0101 \\
+1001 \\
\Rightarrow \\
\text{carry out, over flowed} \\
-0100 \\
\text{(all 4-bit examples)}
\end{array}
\]

Subtraction: Similar to addition, only do after determining operation based on sign bits.

Need to switch since larger is on bottom