CMSC 250
Discrete Structures

Logic Applications
(Circuits & Adders)
Circuits

- AND gate
- OR gate
- NOT gate
Combining & Determining I/O Relationship

\[ P \lor \sim (Q \land R) \]
### Draw the Circuit for:

P, Q & R are inputs

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Simplify before building the circuit.

Output RQP.
Number Conversions

- **Base of the Number System**
  - 10 (decimal), 2 (binary), 8 (octal), 16 (hexadecimal)
  - Tells how many different numerals are used
  - Determines the value of each place

- **Conversions from anything to Base 10**
  - Use the definition of the number system

- **Conversions from Base 10 to anything**
  - Use repeated integer division
Addition of Binary Numbers

- Carry if the number would be too large for the number system – if it is greater than 1

\[
\begin{array}{cccc}
1001 & 1001 & 1011 & 1011 \\
+ 10 & + 11 & + 11 & +111 \\
\end{array}
\]
Addition of Binary Numbers

- Carry if the number would be too large for the number system (larger than 7 or 15)

\[
\begin{align*}
723_8 & \quad + \quad 12_8 \\
265_8 & \quad + \quad 33_8 \\
ABC_{16} & \quad + \quad 12_{16} \\
CDE_{16} & \quad + \quad ED_{16}
\end{align*}
\]
Using a Circuit for Addition

- Write as a logic expression
- Translate to circuits

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>Q</td>
</tr>
<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>
Half Adder

P and Q are binary values (1 bit each)

sum = \((P \lor Q) \land \neg(P \land Q)\)

carry = \((P \land Q)\)
Full Adder

- P, Q and R are binary digits
- P + Q + R gives sum value and carry value
Parallel Adders

- Chain these half adders and full adders together for multi-bit addition

\[ X_1X_2X_3 + Y_1Y_2Y_3 = CA_1A_2A_3 \]
2's Compliment

To represent negative values using binary
1. Find the binary equivalent of the absolute value.
2. Pad on the left to completely fill the bits.
3. Switch all of the 1's to 0's and 0's to 1's.
4. Add 1 to the result.

Find the 8-bit 2's compliment representation of -43
1. $43_{10}$
2. $101011_2$
3. $00101011_2$
4. $11010100_2$
5. $11010101_2 = -43_{10}$