1. Draw Karnaugh maps for the following two input gates: NOR, NAND, XOR. Give the Boolean expression for each gate.

### NOR

\[
\begin{array}{c|cc}
A \quad B & 0 & 1 \\
\hline
0 & 0 & 1 \\
1 & 0 & 0 \\
\end{array}
\]

\[
\overline{A+B} = \overline{A} \cdot \overline{B}
\]

### NAND

\[
\begin{array}{c|cc}
A \quad B & 0 & 1 \\
\hline
0 & 1 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

\[
\overline{A \cdot B} = \overline{A} + \overline{B}
\]

### XOR

\[
\begin{array}{c|cc}
A \quad B & 0 & 1 \\
\hline
0 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

\[
A \oplus B = \overline{A \cdot B}
\]
2. Show how to construct an exclusive NOR gate (XNOR) using only NAND gates. The truth table for XNOR is:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

By truth table, since XNOR is

\[ A \odot B = AB + \overline{AB} = AB + \overline{A + B} \]

which looks like

Naive form with NANDs, put bubbles in:

Then OR

So get
MORE SYMMETRIC VERSION, FROM CLASS NOTES.

SAME NUMBER OF GATES (5)
3. Simplify the logic circuit below by the following steps.

(a) Make a Karnaugh map.
(b) Draw "covers" on the K-map to find minimal Boolean terms.
(c) Manipulate the logical expression, if needed, to reduce the number of gates needed.
(d) Redraw the diagram with the minimal number of gates and types. You should be able to make it with two two-input gates of more than one type or three two-input gates of a single type (either NAND or NOR). (Hint: one input will not matter.)

\[ B \cdot C = C(B + D) \]
Thus \( Q = C(B+D) \) (Does not depend on A)

With only **NOR** gates:

\[ \text{NOR} = \overline{D} \]

Note \( C(B+D) = \overline{C}B + \overline{C}D = \overline{(C B)(C D)} \)