Review Problem 9

- Assuming all gates have the same delay (including inverters), complete the following timing diagram.
Converting Circuits to NAND/NOR Form

- Group gates into levels, insert double inversions on alternating levels

- Alternating AND/OR becomes all NAND or NOR
Converting to NAND/NOR Form (cont.)

- Some circuits may require internal inverters
XOR and XNOR Gates

- XOR Gate: $Z=1$ if odd # of inputs are true

\[
\begin{array}{ccc}
X & Y & Z \\
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

\[
X \oplus Y
\]

- XNOR Gate: $Z=1$ if even # of inputs are true

\[
\begin{array}{ccc}
X & Y & Z \\
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}
\]

\[
X \oplus Y
\]
Optimization via K-Maps to 2-level forms

- Readings: 2.11-2.12.2, 2.14

- Sum of Products form: the OR of several AND gates, inversions over only inputs
  - \( F = \overline{X} + Y\overline{Z} + XYZ \)

- Circuit diagram & inversions:

```
3 NANDS
```

\[ Y(x+z) = xy + yz \]
On Sets and Off Sets

- **On Set** is the set of input patterns where the function is TRUE
  \[ \overline{x} \overline{y} z, \overline{x} y \overline{z}, \overline{x} y \overline{z}, x \overline{y} z \]

- **Off Set** is the set of input patterns where the function is FALSE
  \[ \overline{x} \overline{y} z, \overline{x} y \overline{z}, \overline{x} y \overline{z}, x y z \]
Two-Level Simplification

**Key Tool: The Uniting Theorem** — \( A (\overline{B} + B) = A \)

\[ F = A \overline{B} + A B = A (\overline{B} + B) = A \]

- B's values change within the on-set rows
- \( B \) is eliminated, A remains
- A's values don't change within the on-set rows

\[ G = \overline{A} \overline{B} + A \overline{B} = (\overline{A} + A) \overline{B} = \overline{B} \]

- B's values stay the same within the on-set rows
- A is eliminated, B remains
- A's values change within the on-set rows

**Essence of Simplification:**
find two element subsets of the ON-set where only one variable changes its value. This single varying variable can be eliminated!
Karnaugh Maps

Karnaugh Map Method

K-map is an alternative method of representing the truth table that helps visualize adjacencies in up to 4 dimensions.

Beyond that, computer-based methods are needed.

2-variable K-map

3-variable K-map

4-variable K-map
Truth Tables to K-Maps

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

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Swapped
K-Map Simplification

K-Map Method Examples

Draw circles with power of 2

F = A

\[ \text{Cout} = AB + B\text{Cin} + A\text{Cin} \]

G = \overline{B}

F(A,B,C) = A
K-Map Simplification (cont.)

More K-Map Method Examples, 3 Variables

\[ F(A, B, C) = \bar{A} \bar{B} \bar{C} + A \bar{B} C + A \bar{B} C + A B C \]

\[ F = \bar{A}C + B\bar{C} \]

In the K-map, adjacency wraps from left to right and from top to bottom

\[ \bar{F}(A, B, C) = \bar{A} \bar{B} C + \bar{A} B \bar{C} + \bar{A} B C + A B \bar{C} \]

\[ \bar{F} = \bar{A}C + B\bar{C} \]
4-Variable K-Map

K-map Method Examples: 4 variables

\[ F = \overline{AD} + BD + \overline{BC} + A\overline{BD} \]

\[ F = \overline{BC} + D \]

53
K-Map Example

\[ F = (A \text{ xor } C) \times D + A\overline{C}D + \overline{A}BCD \]

\[ = \overline{A}CD + A\overline{C}D + \overline{A}CD + \overline{A}BCD \]