Exercise 2.6

1. Design a State Diagram for a blinking traffic sign as follows:
   - Switch OFF
     - No lights
   - Switch ON, repeat
     - No lights on
     - 1 & 2 on
     - 1, 2, 3, & 4 on
     - 1, 2, 3, 4, & 5 on
Exercise 2.6 (solution)

1. Design a State Diagram for a blinking traffic sign.

```
Switch = 1
00 All off ───> 01 1,2 on
  Switch = 0  
    └──> 11 All on
Switch = 0,1

10 1-4 on ───> 01 1,2 on
  Switch = 0  
    └──> 00 All off
Switch = 1
```

Transition on each clock cycle.
Exercise 2.7

2. Create truth table for all values. (X=1,2 Y=3,4 Z=5)

<table>
<thead>
<tr>
<th>Current State</th>
<th>Next State</th>
<th>Lights</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S'₁</td>
<td>S'₀</td>
</tr>
<tr>
<td>S_w S₁ S₀</td>
<td>S'₁ S'₀</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. Create equations for next state and output values.

S'₁ = 
S'₀ = 
X = 
Y = 
Z =
Exercise 2.7 (solution)

2. Create truth table for all values. (X=1,2 Y=3,4 Z=5)

<table>
<thead>
<tr>
<th>Current State</th>
<th>Next State</th>
<th>Lights</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_w S_1 S_0</td>
<td>S'_1 S'_0</td>
<td>X Y Z</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 1 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 1 0</td>
<td>1 0 1</td>
<td>1 0 0</td>
</tr>
<tr>
<td>1 1 0</td>
<td>1 1 1</td>
<td>1 1 0</td>
</tr>
<tr>
<td>1 1 1</td>
<td>0 0 1</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

3. Create equations for next state and output values.

$S'_1 = (S_w \sim S_1 S_0) + (S_w S_1 \sim S_0)$

$S'_0 = (S_w \sim S_1 \sim S_0) + (S_w S_1 \sim S_0)$

$X = (S_w \sim S_1 S_0) + (S_w S_1 \sim S_0) + (S_w S_1 S_0)$

$Y = (S_w S_1 \sim S_0) + (S_w S_1 S_0)$

$Z = (S_w S_1 S_0)$
Exercise 2.8

4. Implement your state machine with combinational and sequential logic.
Exercise 2.8 (solution)

4. Implement your state machine with combinational and sequential logic.

\[ S'_1 = (S_w \sim S_1 \sim S_0) + (S_w S_1 \sim S_0) \]
\[ S'_0 = (S_w \sim S_1 \sim S_0) + (S_w S_1 \sim S_0) \]
\[ X = (S_w \sim S_1 \sim S_0) + (S_w S_1 \sim S_0) + (S_w S_1 S_0) \]
\[ Y = (S_w S_1 \sim S_0) + (S_w S_1 S_0) \]
\[ Z = (S_w S_1 S_0) \]