"World Acclaimed" Wisconsin Bowling Ball Waxoner Machine

Controller:

Inputs: Start.H (S)  Gloss.H (G)

Outputs: Wax.H (W)  Spin Ball.H (SB)  HEA.L (H)  Buff.L (B)

Assumptions:
- When the person puts in their quarters, start goes true.
- Start stays true for the entire process until the state machine returns to state 000.

Possible Start Signal

- SW is normally closed.
- Put in your quarters and SW opens, sets START = H.
- Gloss resets the START signal.

ASM Diagram
Procedure:

1. Determine ASM flow chart (given in this problem)

2. Count states $\Rightarrow 5$
   * Compute # of F/Fs needed $\Rightarrow 3$

3. Label states on ASM flow chart
   * Try to label such that only one bit will change for each state transition... this will simplify the state variable equations

4. Functional Block Diagram & select F/F Type

5. Next State Table
   * See next page
Find State Variable Equations

- since $S = T$ we count
- when $S = F$ we go to $\phi$ and stay

- we can remove $S$ from $K$-Maps and place in equations via inspection

\[ D_2 = \text{ (from } K\text{-Map) } S \]
\[ D_1 = \text{ (from } K\text{-Map) } S \]
\[ D_0 = \text{ (from } K\text{-Map) } S \]

*Also, if our F/Fs had a clear input signal (low true) we could connect $S$ to this and remove from the equations.*
\[ D_2 = (GQ_2Q_1Q_0)s \]

\[ D_1 = (Q_2 + Q_1Q_0 + Q_0\overline{G} + Q_1\overline{G})s \]

```
check
\[ Q_2 = 0 \] should go to 000
\[ Q_1 = 1 \] else go to 110
\[ Q_0 = 0 \] when \( G = 1 \)
\[ G = 0 \]
```

\[ D_0 = (\overline{Q_1})s \]
Find Equations for Output Variables

- \( H \) is only true in state 010 when \( G = \text{False} \) (Conditional Output)
  
  then by inspection \( H = \overline{Q}_2 \overline{Q}_1 Q_0 \overline{G} \)

\( W, S_B, B \) are all unconditional outputs

and \( \therefore \) only dependent on \( Q_2, Q_1, Q_0 \)

\( W = Q_2 + Q_0 \)

Check in ASM flow chart

\( S_B = Q_2 + Q_0 \)

\( B = \overline{Q}_2 \overline{Q}_1 \overline{Q}_0 \)

\( \therefore \) also you can get this by inspection
(8A) Physical Implementation w/Gates

State Eq. \( D_2 = \overline{Q}_2 Q_1 \overline{Q}_0 \overline{G} S \), \( D_1 = SQ_2 + SQ_1 Q_0 + SQ_0 \overline{G} + SQ_1 \overline{G} \)
\( D_0 = \overline{Q}_1 S \)

These ARE Logic Equations!

Outputs
\( H = \overline{Q}_2 Q_1 \overline{Q}_0 \overline{G} \), \( W = Q_2 + Q_0 \), \( SB = Q_2 + Q_0 \), \( B = \overline{Q}_2 Q_1 Q_0 \)

Circuitry

State Generator

Output Generation

\( \overline{Q}_2 \)
\( \overline{Q}_1 \)
\( \overline{Q}_0 \)
\( W \cdot H \)
\( H \cdot L \)
\( B \cdot L \)
**Physical Implementation w/ EPROM**

Flip Flops synchronize the events to a clock (they are still needed!)

```
D2 ---- D ---- Q ---- Q2
    clk

D1 ---- D ---- Q ---- Q1
     clk
     Next State

D0 ---- D ---- Q ---- Q0
     clk
     Present State
```

Note: we don't need $\overline{Q}$ in F/F now

---

**Memory Contents**

| Address | Hex Value | Binary | SG | Q2 | Q0 | Data | Logic
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>000000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>000001</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>FF</td>
<td>OF</td>
<td>011111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Data**

| Address | Hex Value | Binary | SG | Q2 | Q0 | Data | Logic
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>000000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>000001</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>FF</td>
<td>OF</td>
<td>011111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Actual Voltage to be Programmed**

| Address | Hex Value | Binary | SG | Q2 | Q0 | Data | Logic
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>000000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>000001</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>FF</td>
<td>OF</td>
<td>011111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Note: $X$ will be programmed "$L$"