Outline

1. Exercise (1) Sequential Circuit Analysis
2. Exercise (2) Sequential Circuit Analysis
3. Exercise (3) Sequential Circuit Analysis
4. Ref. Construction of State Diagram
5. Ref. State Reduction
6. Exercise (4) Sequential Circuit Design
Exercise(1) Sequence Circuit Analysis

Assume that we have such a circuit and inputs. what are the outputs of Y and Z (The initial state of the flip-flops are both Q=0)
Exercise (1) Sequence Circuit Analysis

\[ Y = \overline{Q_1} \overline{Q_2} = Q_1 + \overline{Q_2} \]
\[ Z = \overline{Q_1} \overline{Q_2} = \overline{Q_1} + Q_2 \]
Exercise(1) Sequence Circuit Analysis

- **Sequential Circuit**
  - State equations
    - Specify next state as a function of ‘present state’ and ‘input(s)’
    - Similar to Boolean equations
  - State tables
    - List out all combination of input and present state to show the output and next state
    - Similar to truth table
  - State diagrams
    - Graphical representation for state tables

- You can know the others if you already have 1 clue.
- **Easy** to convert between sequential circuit and state equations
- **Easy** to convert between state tables and state diagrams
- **Difficult** to find relationship between state table / state equations.
  We need some tools (e.g. K-map, excitation table, etc...) for help.
Exercise (2) Sequential Circuit Analysis

A sequential circuit has one flip-flop Q, two inputs x and y, and one output S(Sum). It consists of a full-adder circuit connected to a D flip-flop, as shown in the figure below.

Derive the state table, state equation and state diagram of the circuit.
Exercise (2) Sequential Circuit Analysis

Tabulate present state and inputs as three independent Boolean variables, the produce the output $S$ (Sum) and $C$ (Carry) according to the full-adder operation, followed by producing the next states according the D flip-flop operation. The state diagram has two states only.

State equation and output equation:

$$Q(t+1) = xy + xQ + yQ$$
$$S = x \oplus y \oplus Q$$
Exercise (2) Sequential Circuit Analysis

State Table

<table>
<thead>
<tr>
<th>Present States</th>
<th>Inputs</th>
<th>Next States</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q(t)</td>
<td>x</td>
<td>y</td>
<td>Q(t+1)</td>
</tr>
<tr>
<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>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<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>
</tr>
</tbody>
</table>

State Diagram
A basic state table for our example circuit is shown right.

1. Complete the state table
2. Draw the state diagram

<table>
<thead>
<tr>
<th>Present State</th>
<th>Inputs</th>
<th>Next State</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Q₁ Q₀</td>
<td>X</td>
<td>Q₁ Q₀</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
1. Determine the output

- The output depends on the current state (Q0 and Q1) as well as the inputs.
- From the diagram, you can see that

\[ Z = Q_1 Q_0 X \]

Output at the current time

<table>
<thead>
<tr>
<th>Present State</th>
<th>Inputs</th>
<th>Next State</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1 Q0</td>
<td>X</td>
<td>Q1 Q0</td>
<td>Z</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
2. Flip-Flop Input Equations

- Finding the next states is harder. To do this, we have to figure out how the flip-flops are changing.

**Step 2a:**
Find Boolean expressions for the flip-flop inputs.
- i.e. How do the inputs (say, J & K) to the flipflops depend on the current state and input

**Step 2b:**
Use these expressions to find the actual flip-flop input values for each possible combination of present states and inputs.
- i.e. Fill in the state table (with new intermediate columns)

**Step 2c:**
Use flip-flop characteristic tables or equations to find the next states, based on the flip-flop input values and the present states.
Step 2a: Flip-flop input equations

- For our example, the flip-flop input equations are:
  
  \[ J_1 = X' Q_0 \]
  \[ K_1 = X + Q_0 \]
  \[ J_0 = X + Q_1 \]
  \[ K_0 = X' \]

- JK flip-flops each have two inputs, J and K. (D and T flip-flops have one input each.)
Step 2b: Flip-flop input values

With these equations, we can make a table showing $J_1$, $K_1$, $J_0$ and $K_0$ for the different combinations of present state $Q_1Q_0$ and input $X$.

$$J_1 = X' Q_0$$
$$K_1 = X + Q_0$$
$$J_0 = X + Q_1$$
$$K_0 = X'$$

<table>
<thead>
<tr>
<th>Present State $Q_1$</th>
<th>Present State $Q_0$</th>
<th>Inputs $X$</th>
<th>Flip-flop Inputs $J_1$</th>
<th>Flip-flop Inputs $K_1$</th>
<th>Flip-flop Inputs $J_0$</th>
<th>Flip-flop Inputs $K_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<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>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Step 2c: Find the next states

- Finally, use the JK flip-flop characteristic tables or equations to find the next state of each flip-flop, based on its present state and inputs.
- The general JK flip-flop characteristic equation is:

  \[ Q(t+1) = K'Q(t) + JQ'(t) \]

- In our example circuit, we have two JK flip-flops, so we have to apply this equation to each of them:

  \[ Q_1(t+1) = K_1'Q_1(t) + J_1Q_1'(t) \]
  \[ Q_0(t+1) = K_0'Q_0(t) + J_0Q_0'(t) \]

- We can also determine the next state for each input/current state combination directly from the characteristic table.

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>Q(t+1)</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Q(t)</td>
<td>No change</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Reset</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Set</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Q'(t)</td>
<td>Complement</td>
</tr>
</tbody>
</table>
Step 2c Conclusion

Finally, here are the next states for $Q_1$ and $Q_0$, using these equations:

$$Q_1(t+1) = K_1'Q_1(t) + J_1Q_1'(t)$$

$$Q_0(t+1) = K_0'Q_0(t) + J_0Q_0'(t)$$

<table>
<thead>
<tr>
<th>Present State</th>
<th>Inputs</th>
<th>FF Inputs</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_1$</td>
<td>$Q_0$</td>
<td>$X$</td>
<td>$J_1$</td>
</tr>
<tr>
<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>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<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>
</tr>
</tbody>
</table>
3. State diagrams

- We can also represent the state table graphically with a state diagram.
- A diagram corresponding to our example state table is shown below.
4. Sizes of state diagrams

- Always check the size of your state diagrams.
  - If there are \( n \) flip-flops, there should be \( 2^n \) nodes in the diagram.
  - If there are \( m \) inputs, then each node will have \( 2^m \) outgoing arrows.
- In our example,
  - We have two flip-flops, and thus four states or nodes.
  - There is one input, so each node has two outgoing arrows.
5. State diagram

Construction

• Identify all possible states (they are the nodes)

• Identify how the process moves from a state to another (they are the links)

• Identify how the outputs are generated
The directed lines are labelled with two binary numbers separated by a slash (/). The input value that causes the state transition is labelled first. The number after the slash symbol / gives the value of the output.

For example, the directed line from state 00 to 01 is labelled 1/0, meaning that, if the sequential circuit is in a present state and the input is 1, then the next state is 01 and the output is 0. If it is in a present state 00 and the input is 0, it will remain in that state. A directed line connecting a circle with itself indicates that no change of state occurs. The state diagram provides exactly the same information as the state table and is obtained directly from the state table.
5. Ref. State Reduction

• Equivalent property
  
  – Two states said to be equivalent if, for all sequence of inputs, the sequential machine produces the same output sequence when it is started in either state.

  • Output must be identical

  • Next states may not be identical provided that it is possible to establish an equivalence between the unlike states
5. Ref. State Reduction

- Distinct states generate different sequences of outputs for the same set of input sequence.
- Equivalent states generate the same output sequence for all possible input sequence patterns.

∴ One state can be assigned to the equivalent states without affecting the output sequence pattern.
1. Find the Equivalent table based on the reduced table:

<table>
<thead>
<tr>
<th>Present States</th>
<th>X=0</th>
<th>X=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>C</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>A</td>
</tr>
</tbody>
</table>

2. Find out the terms that should be reduced based on the reduced table: A → B with input 1, since A and B are equivalent, A → A with input 1. State B can be reduced, or state A can be reduced. (if one choose to reduce state A, then use B instead of A)
Exercise (4)

- Design, using D flip-flop, circuit is based on state table below.
Solution

- Determine input expression for D flip-flop and y output

\[
\begin{array}{|c|c|c|c|c|}
\hline
\text{Present state} & \text{Input} & \text{Next state} & \text{Output} \\
\hline
A & B & x & A^+ & B^+ & y \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 \\
\hline
\end{array}
\]

\[
DA(A, B, x) = \sum m(2, 4, 5, 6)
\]

\[
DB(A, B, x) = \sum m(1, 3, 5, 6)
\]

\[
y(A, B, x) = \sum m(1, 5)
\]

\[
DA = A \cdot B' + B \cdot x'
\]

\[
DB = A' \cdot x + B' \cdot x + A \cdot B \cdot x'
\]

\[
y = B' \cdot x
\]
Solution

- From expression built, draw logic diagram

\[
DA = A \cdot B' + B \cdot x' \\
DB = A' \cdot x + B' \cdot x + A \cdot B \cdot x' \\
y = B' \cdot x
\]