Overview of Chapter 4

- Part 1: Types of Sequential Circuits
  - Storage Elements
  - Latches and Flip-Flops
- Part 2: Sequential Circuit Analysis
  - State Tables and State Diagrams
- Part 3: Sequential Circuit Design
  - Specification
  - Assignment of State Codes
  - Implementation
- Part 4: HDL Representation

Sequential Circuit Analysis

- General Model
  - Current State at time (t) is stored in an array of flip-flops.
  - Next State at time (t+1) is a Boolean function of current state and inputs.
  - Outputs at time (t) are a Boolean function of current state (t) and (sometimes) current inputs (t).

Example (from Fig. 4-18)

- Input: x(t)
- Output: y(t)
- State: A(t), B(t)
- What is the Output Function?
- What is the Next State Function?

Example (Fig. 4-18) (Continued)

- Boolean Equations for the functions:
  - \( A(t+1) = A(t)x(t) + B(t)x(t) \)
  - \( B(t+1) = \bar{A}(t)x(t) \)
  - \( y(t) = x(t)B(t) + A(t) \)

Example (Fig. 4-18) (Continued)

- Where in time are inputs, outputs and states defined?

Partial Simulation - Fig. 4-18 Rev a Kime
- \( t = 0 \) to \( t = 3 \)
- Inputs: x, y
- States: A, B
- Outputs: y, \( \bar{y} \)
State Table Definition

- **State table** – a multiple variable function table with the following four sections:
  - **Present State** – the values of the state variables for each allowed state.
  - **Input** – the input combinations allowed.
  - **Next-state** – the value of the state at time \( t+1 \) based on the present state and the input.
  - **Output** – the value of the output as a function of the present state and (sometimes) the input.

From the viewpoint of a truth table:
- the inputs are Input, Present State
- the outputs are Output, Next State

Example: State Table (Fig. 4-18)

- The STATE TABLE can be filled in using the next state and output equations:
  \[
  A(t+1) = A(t)X(t) + B(t)X(t)
  \]
  \[
  B(t+1) = \overline{A(t)}X(t)
  \]
  \[
  y(t) = X(t)(B(t) + A(t))
  \]

<table>
<thead>
<tr>
<th>Present State</th>
<th>Input</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(t) B(t)</td>
<td>x(t)</td>
<td>A(t+1)</td>
<td>y(t)</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td>1 1</td>
<td>0</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>1 1</td>
<td>0</td>
</tr>
</tbody>
</table>

Alternate State Table

- 2-dimensional table that matches well to a K-map.
  - Present state rows and input columns in Gray code order.
  - \( A(t+1) = A(t)X(t) + B(t)X(t) \)
  - \( B(t+1) = \overline{A(t)}X(t) \)
  - \( y(t) = X(t)(B(t) + A(t)) \)

State Diagrams

- The sequential circuit function can be represented in graphical form as a state diagram with the following components:
  - A **circle** with the state name in it for each state
  - A **directed arc** from the Present State to the Next State for each state transition
  - A label on each directed arc with the Input value which causes the state transition, and
  - A label:
    - On each circle: with the output value produced, or
    - On each directed arc: with the output value produced.

State Diagram Example

- Which type?
- Gets confusing as circuit grows in size.
- For small circuits, usually easier to understand than the state table.
- Try drawing state diagram for mod 4 counter and toggle (T) flip-flop
State Diagram Characteristics

- The Boolean state variables are a vector of \( n \) bits.
- Not all \( 2^n \) states are necessarily used!
- Similarly not all input and output combinations are used.
- The state variables may need to be initialize to a valid, appropriate initial state.
- Examples:
  - A system with 10 states requires a minimum of 4 bits (3 bits gives only 8 symbols).
  - BCD coded inputs can have 16 combinations, only 10 of which have meaning.

Flip-Flop Input Functions

- The D-Flip-Flop easy to analyze since it has only one input. Other FFs such as the JK and SR have two inputs.
- Convention used in text:
  - First Letters designate the FF input function.
  - Second Letters (or subscript) designate the state variable.
- Example with Two JKFFs:
  - \( J_A = B \quad K_A = \overline{B} \quad J_B = \overline{x} \quad K_B = \overline{A} \quad \overline{x} \quad \overline{x} + \overline{A} \quad x \)
- Example 4-18 with Two DFFs:
  - \( D_A = A \quad x + B \quad x \quad D_B = \overline{A} \quad x \quad y = (A + B) \quad \overline{x} \)

Analysis with Other Flip-Flops

- With a D Flip-Flop:
  - Next state obtained directly from the flip-flop input equation for \( D \)
- With a JK, T or SR Flip-Flop:
  - Obtain the values for each flip-flop input in terms of present state and input values
  - Use the corresponding flip-flop characteristic table from Table 4-1 (next slide) to determine the next state value of the flip-flop

Characteristic Tables

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>( Q(t+1) )</th>
<th>Comment</th>
<th>S</th>
<th>R</th>
<th>( Q(t+1) )</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>( Q(t) )</td>
<td>No change</td>
<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>Clear Q</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Clear Q</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Set Q</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Set Q</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>( \overline{Q(t)} )</td>
<td>Complement Q</td>
<td>1</td>
<td>1</td>
<td>? Indeterminate</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T</th>
<th>( Q(t+1) )</th>
<th>Comment</th>
<th>D</th>
<th>( Q(t+1) )</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( Q(t) )</td>
<td>No change</td>
<td>0</td>
<td>( Q(t) )</td>
<td>No change</td>
</tr>
<tr>
<td>1</td>
<td>( \overline{Q(t)} )</td>
<td>Complement Q</td>
<td>1</td>
<td>1</td>
<td>Set Q</td>
</tr>
</tbody>
</table>

JK Flip-Flop Circuit Analysis

- Step 1: Write the Boolean expression for each flip-flop input.
- For flip-flop A:
  - \( J_A = \) 
  - \( K_A = \) 
- For flip-flop B:
  - \( J_B = \) 
  - \( K_B = \) 

JK Flip-Flop Analysis (Cont.)

- Step 2: Using the diagram or equations, fill in the flip-flop inputs.
  - \( J_A = B \quad x \quad K_A = B \quad J_B = \overline{x} \quad K_B = A \quad x \)
JK Flip-Flop Analysis (Cont.)

• Step 3: By using JK Characteristic Table, the J and K inputs and the present state from the table, fill in the next state in the table for each flip flop.

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>Q(t)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No change</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Clear Q</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Set Q</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Complement Q</td>
</tr>
</tbody>
</table>

JK Flip-Flop Analysis (Cont.)

• The result of completion of Step 3:

<table>
<thead>
<tr>
<th>Present State</th>
<th>Input</th>
<th>Next State</th>
<th>Flip-Flop Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>0 1</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td>1 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>1 1</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0 1 0</td>
</tr>
</tbody>
</table>

Additional Concepts

• Characteristic Equations
• Moore and Mealy Models
• Diagram Examples
• Table Examples

Characteristic Equations

• Can be used instead of characteristic tables for transforming flip-flop inputs to next state information

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>Q(t)</th>
<th>Comment</th>
<th>S</th>
<th>R</th>
<th>Q(t+1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Q(t)</td>
<td>No change</td>
<td>0 0</td>
<td>Q(t)</td>
<td>No change</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Clear Q</td>
<td>0 0</td>
<td>Clear Q</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Set Q</td>
<td>1 0</td>
<td>Set Q</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Complement Q</td>
<td>1 1</td>
<td>Indeterminate</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Moore and Mealy Models

• Sequential Circuits or Sequential Machines are also called Finite State Machines (FSMs). Two formal models exist:

  • **Moore Model**
    - Named after E.F. Moore.
    - Outputs are a function ONLY of states.
    - Usually specified on the states.

  • **Mealy Model**
    - Named after G. Mealy.
    - Outputs are a function of inputs AND states.
    - Usually specified on the state transition arcs.
Moore and Mealy Example Diagrams

- **Mealy Model State Diagram**
  maps inputs and state to outputs

- **Moore Model State Diagram**
  maps states to outputs

Moore and Mealy Example Tables

- **Mealy Model State Table**
  maps inputs and state to outputs

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State ( x=0 )</th>
<th>Next State ( x=1 )</th>
<th>Output ( x=0 )</th>
<th>Output ( x=1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- **Moore Model State Table**
  maps state to outputs

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State ( x=0 )</th>
<th>Next State ( x=1 )</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>