UNIT 14
DERIVATION OF STATE GRAPHS AND TABLES

Iris Hui-Ru Jiang
Spring 2010

Designing a Sequential Circuit

- Given the specification of a sequential circuit
- Design procedure:
  1. Construct a state table or state graph (Unit 14)
  2. Simplify (Unit 15)
  3. Derive FF input equations and output equations (Unit 12)

Sequence Detectors

- Basic unit
  Unit 11: Latch & FFs
- Simple sequential Ckt
  Unit 12: Registers & Counters
- Complex sequential Ckt
  Units 13-15: FSM
- Put it all together
  Unit 16: Summary

Derivation of State Graphs and Tables

- Contents
  - Case studies: sequence detectors
  - Guidelines for construction of state graphs
  - Serial data code conversion
  - Alphanumeric state graph notation

- Reading
  - Unit 14
Case I (1/2)

- Examine groups of 4 consecutive inputs & produce an output
- Reset after every 4 inputs
  - e.g., $X = 0101 \ 0010 \ 1001 \ 0100$
  - $Z = 0001 \ 0000 \ 0001 \ 0000$
- Observation: $X = 0101$
  - $Z = 0001 \ 0000 \ 0001 \ 0000$

- Typical sequence
  - Partial state graph

<table>
<thead>
<tr>
<th>State</th>
<th>Sequence received</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>Reset</td>
</tr>
<tr>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>1</td>
</tr>
<tr>
<td>$S_3$</td>
<td>01 or 10</td>
</tr>
<tr>
<td>$S_4$</td>
<td>010 or 100</td>
</tr>
</tbody>
</table>

Derivation of state graphs & tables

Case I (2/2)

- Complete state graph

State table

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
<th>Present output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_2$</td>
<td>0</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
<tr>
<td>$S_6$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
<tr>
<td>$S_7$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
<tr>
<td>$S_8$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
<tr>
<td>$S_9$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
</tbody>
</table>

State maps

<table>
<thead>
<tr>
<th>$AB$</th>
<th>$X = 0$</th>
<th>$X = 1$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Derivation of state graphs & tables

Case II (1/4)

- Examine groups of 3 consecutive inputs & produce an output
- No reset
  - e.g., $X = 001 \ 101 \ 100 \ 101 \ 010 \ 100$
  - $Z = 000001 \ 1000001 \ 010000$
- State graph (Mealy)
  - $S_0$: initial, $S_1$: get ...1, $S_2$: get ...10

Derivation of state graphs & tables

Case II (2/4)

State table

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
<th>Present output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_0$</td>
<td>0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_1$</td>
<td>0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$S_2$</td>
<td>0</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
<tr>
<td>$S_5$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
<tr>
<td>$S_6$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
<tr>
<td>$S_7$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
<tr>
<td>$S_8$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
<tr>
<td>$S_9$</td>
<td>$S_0$</td>
<td>1</td>
</tr>
</tbody>
</table>

State maps

$A^*X' = B^*X = Z = XA$

Derivation of state graphs & tables
Case II (3/4)

- State maps

- Realize it

Q: Check by yourself

\[
X = 0011 1011 0010 1010 1000
\]

Z = 0000 0100 0110 1000

Derivation of state graphs & tables

Case II (4/4)

- Moore?

- State for \( S_0 \): initial, \( S_1 \): get \( \ldots 1 \), \( S_2 \): get \( \ldots 10 \), \( S_3 \): get \( \ldots 101 \)

- Derivation of state graphs & tables

Case III (1/2)

- 010 & 1001 detector

- State assignment

- State for "010"

- State for "1001"

- Complete

Derivation of state graphs & tables

Case III (2/2)

- State for "010"

- State for "1001"

- Complete

Derivation of state graphs & tables
**Case IV (1/2)**

- **Specifications**
  - $Z = 1$ if total # of 1's is **odd** and at least **two** consecutive 0's have been received
  - e.g., $X = 1\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0$,
    
  - $Z = (0)\ 0\ 0\ 0\ 0\ 1\ 0\ 1$

- **State assignment**
  - Initial state and state for 1's

- **State for 0's**

<table>
<thead>
<tr>
<th>State</th>
<th>Sequence received</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>Reset or even 1's</td>
</tr>
<tr>
<td>$S_1$</td>
<td>Odd 1's</td>
</tr>
<tr>
<td>$S_2$</td>
<td>Even 1's and ends in 0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>Even 1's and 00 has occurred</td>
</tr>
<tr>
<td>$S_4$</td>
<td>00 has occurred and odd 1's</td>
</tr>
</tbody>
</table>

**Guidelines for State Graphs Construction**

- **Steps**
  1. Construct sample sequences to help you understand the problem
  2. Determine under what conditions it should reset
  3. If only one or two sequences leads to a nonzero output, construct a partial state graph
  - Another way, determine what sequences or groups of sequences must be remembered by the circuit and set up states accordingly
  4. Each time you add an arrow to the state graph, determine whether it can go to one of the previously defined states or whether a new state must be added
  5. Check your graph to make sure there is one and only one path leaving each state for each combination of values of the input variables
  6. When your graph is complete, verify it by applying the input sequences formulated in step 1

**Case IV (2/2)**

- **State graph**

- **Initial state and state for 1's**

- **Odd 1's**

- **Guidelines for State Graph Construction**

- **Steps**
  1. Construct sample sequences to help you understand the problem
  2. Determine under what conditions it should reset
  3. If only one or two sequences leads to a nonzero output, construct a partial state graph
  - Another way, determine what sequences or groups of sequences must be remembered by the circuit and set up states accordingly
  4. Each time you add an arrow to the state graph, determine whether it can go to one of the previously defined states or whether a new state must be added
  5. Check your graph to make sure there is one and only one path leaving each state for each combination of values of the input variables
  6. When your graph is complete, verify it by applying the input sequences formulated in step 1
Serial Data Transmission

Coding schemes

<table>
<thead>
<tr>
<th>Bit Sequence</th>
<th>NRZ</th>
<th>NRZI</th>
<th>RZ</th>
<th>Manchester</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 1 0 0 1 0</td>
<td>0 ⇒ 0; 1 ⇒ 1</td>
<td>0 ⇒ d; 1 ⇒ ~d</td>
<td>0 ⇒ 0; 1 ⇒ 10</td>
<td>0 ⇒ 01; 1 ⇒ 10</td>
</tr>
</tbody>
</table>

NRZ: Non-return-to-zero
NRZI: Non-return-to-zero-inverted
RZ: Return-to-zero

Mealy?

- **Mealy**:
  - Output depends on
    - Current state (synchronous)
    - Input (maybe asynchronous)
  - Fewer states

- **NRZ data**
  - Clock2
  - Conversion Network
  - Z (Manchester data)

Moore?

- **Moore**:
  - Output only depends on
    - Current state (synchronous)
  - More states (in general)
  - 1 clock period delay

- **NRZ data**
  - Clock2
  - Conversion Network
  - Z (Manchester data)

- **Starting states**: $S_0$, $S_2$
Alphanumeric State Graph Notation

When a sequential circuit has several inputs, label the state graph arcs with alphanumeric input variable names instead of 0's and 1's.

- e.g., 2 inputs: F: forward, R: reverse

Complete?

- Completely specified state graph
  - OR together all input labels on arcs emanating from a state, the result can reduce to 1
  - Cover all conditions: \( F + F'R + F'R' = F + F' = 1 \)
  - AND together any pair of input labels on arcs emanating from a state, the result can reduce to 0
  - Only one arc is valid: \( F\cdot F'R = 0, F\cdot F'R' = 0, F'R\cdot F'R' = 0 \)

- Notation in state graph
  - \( X_iX'_i/Z_2Z_3 = 1\sim0/0110 \)
  - \(-/Z_1 = ----/1000 \)
  - For any combination of input values...

Homework for Unit 14

- Problems
  - 14.32
  - 14.42
  - 14.46

- Homework #6: Units 13–14
  - Due 10am, June 1, 2010; submit to ED518
  - Quiz #6: June 8, 2010.