Sequential Circuits

COE203
Digital Logic Laboratory

Dr. Ahmad Almulhem
COE - KFUPM
Spring 2009
Sequential Circuits

- The outputs, $Z$, depends both on the present inputs, $X$, and the present state $Y$.
- The next state depends on the inputs $X$ and the present state $Y$.

$$Z = F(X,Y)$$
Example

- 4-bit adder (ripple-carry)
  - Notice how carry-out propagates
- 4 full adders are needed
• 1-bit memory and 2 4-bit memory
• Only one full-adder!
• 4 clocks to get the output (sequential adder)
Types of Sequential Circuits

• Synchronous
  – State changes synchronized by one or more clocks
  – Easier to design and analyze

• Asynchronous
  – Changes occur independently
  – Faster
  – Hard to design and analyze
Clocks in Synchronous Circuits

(a) Block diagram

(b) Timing diagram of clock pulses
Basic Storage

- Apply low or high for longer than $t_{pd}$
- Feedback will hold value

Fig. 4-2 Logic Structures for Storing Information
SR (set-reset) Latch

- Basic storage made from gates
- When S & R are both 0 → “keep” state
- When S & R are both 1 → undefined!

(a) Logic diagram

(b) Function table

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Q</th>
<th>( \overline{Q} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</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>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Set state
Reset state
Undefined
S'R' (set-reset) Latch

- Similar – made from NANDs

(a) Logic diagram
(b) Function table

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Q</th>
<th>Q̅</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
| 1 | 1 | 1  | 0  | Set state
| 1 | 0 | 0  | 1  |
| 1 | 1 | 0  | 1  | Reset state
| 0 | 0 | 1  | 1  | Undefined
- Control when state can change
- How about the undefined state?

![Logic diagram](image)

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>R</th>
<th>Next state of Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>No change</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>No change</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Q = 0; Reset state</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Q = 1; Set state</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Undefined</td>
</tr>
</tbody>
</table>
• No undefined state

(b) Function table
Transparency of latches

- As long as C (the control) is high, state can change
  - This is called transparency
- Results are unpredictable
- Behavior depends on gate delays and not logical values
Flip-Flops (Solving Transparency)

• Flip-Flops:
  – Ensure output changes only once per clock cycle

• Two commonly-used types of flip-flops:
  – Master-Slave:
    • Use two latches
  – Edge-Triggered:
    • Implemented very different from latches
    • Triggers at signal transition (positive, negative)
SR Master-Slave Flip-Flop

- Two latches and an inverter
- Either Master or Slave is enabled, not both
Edge-Triggered Flip-Flops

- Changes at input of FF when clock high do trigger next state.
  - Is this a problem?
  - Master-Slave sometimes called pulse triggered
  - As clock goes faster, more problems

- New state latched on clock transition
- Low-to-high: positive-edge triggered
- High-to-low: negative-edge triggered
- Also: dual-edge-triggered
D-Type Edge-Triggered

- Positive or negative?
Standard Symbols

(a) Latches

(b) Master-Slave Flip-Flops
Standard Symbols

(c) Edge-Triggered Flip-Flops
Direct Inputs

- Force Set/Reset independent of clock
  - Direct set or *preset*
  - Direct reset or *clear*
Propagation Delay

The times it takes for an input to appear at the output is called the propagation delay.
Setup and Hold Times

- **Setup time** \((T_s)\) refers to a constant duration for which the inputs must be held prior to the arrival of the clock transition.

- **Hold time** \((T_h)\) refers to the duration for which the inputs must not change after the arrival of the clock transition.

![Diagram of Setup and Hold Times](image)
Design procedure

1. Verbal Description
2. State Diagram
3. State Table
4. Reduced State Table
5. Minimization Techniques
6. State Assignment
7. Circuit Diagram
8. FF I/P Equations & O/P Equations
9. Output & State Transition Table
10. Flip-flop Excitation Tables
Question

- Design a circuit to detect a sequence of 3 ones (111) or more using D-FF. The circuit has one input X and one output Z.
  - Draw the state diagram
  - How many FF do you need? Why?
  - Derive the state table
  - Obtain the simplified equations
  - Draw the circuit
Assignment

• Read experiment 1 in the lab manual