Latches, the D Flip-Flop & Counter Design

ECE 152A – Summer 2009

---

Reading Assignment

- Brown and Vranesic
  - 7 Flip-Flops, Registers, Counters and a Simple Processor
    - 7.1 Basic Latch
    - 7.2 Gated SR Latch
      - 7.2.1 Gated SR Latch with NAND Gates
    - 7.3 Gated D Latch
      - 7.3.1 Effects of Propagation Delays
Reading Assignment

- **Brown and Vranesic** (cont)
  - 7 Flip-Flops, Registers, Counters and a Simple Processor (cont)
    - 7.4 Master-Slave and Edge-Triggered D Flip-Flops
      - 7.4.1 Master-Slave D Flip-Flop
      - 7.4.2 Edge-Triggered D Flip-Flop
      - 7.4.3 D Flip-Flop with Clear and Preset
      - 7.4.4 Flip-Flop Timing Parameters (2nd edition)

---

Reading Assignment

- **Roth**
  - 11 Latches and Flip-Flops
    - 11.1 Introduction
    - 11.2 Set-Reset Latch
    - 11.3 Gated D Latch
    - 11.4 Edge-Triggered D Flip-Flop
Reading Assignment

- **Roth (cont)**
  - 12 Registers and Counters
    - 12.1 Registers and Register Transfers
    - 12.2 Shift Registers
    - 12.3 Design of Binary Counters
    - 12.4 Counters for Other Sequences

Combinational vs. Sequential Logic

- **Combinational logic**
  - Function of present inputs only
    - Output is known if inputs (some or all) are known

- **Sequential logic**
  - Function of past and present inputs
    - Memory or “state”
    - Output known if present input and present state are known
      - Initial conditions often unknown (or undefined)
Gate Delays

- Recall from earlier lecture
  - When gate inputs change, outputs don’t change instantaneously

Feedback

- Outputs connected to inputs
  - Single inverter feedback
    - If propagation delay is long enough, output will oscillate
Feedback

- If the propagation delay is not long enough, the output will settle somewhere in the middle
  - \( V_{in} = V_{out} \)

Feedback

- Ring Oscillator
  - Any odd number of inverters will oscillate
    - \( \frac{1}{2} \) period = total prop delay of chain

<table>
<thead>
<tr>
<th>Name</th>
<th>20.0ns</th>
<th>40.0ns</th>
<th>60.0ns</th>
<th>80.0ns</th>
<th>100.0ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN_OSC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[1]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[0]OSC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Feedback

- What about an even number of inversions?
  - Two inverter feedback
    - Memory (or State)
    - Static 1 or 0 "stored" in memory

The Latch

- Replace inverters with NOR gates
The Set-Reset (SR) Latch

- **NOR implementation**
  - Inverted feedback

![Circuit Diagram](image)

(a) Circuit

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Q₀</th>
<th>Q₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</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>
</tbody>
</table>

(b) Truth table

---

The SR Latch

- **R = Reset (clear)**
  - Q → 0, Q* → 1
- **S = Set (preset)**
  - Q → 1, Q* → 0
- **NOR gate implementation**
  - Either input = 1 forces an output to 0
The SR Latch (cont)

- **Terminology**
  - Present state, $Q$
    - Current value of $Q$ and $Q^+$
  - Next state, $Q^+$
    - Final value of $Q$ and $Q^+$ after input changes


The SR Latch (cont)

- **Operation**
  - $S=1$, $R=0$: set to 1, $Q^+ = 1$
  - $S=0$, $R=1$: reset to 0, $Q^+ = 0$
  - $S=0$, $R=0$: hold state, $Q^+ = Q$
  - $S=1$, $R=1$: not allowed
    - $Q^+ = Q^{++} = 0$, lose state
The SR Latch (cont)

- Timing Diagram
  - RS inputs are “pulses”
    - Temporarily high, but normally low

- Characteristic Equation
  - Algebraic expression of flip-flop behavior
  - Plot characteristic table on map, find $Q^+$
    - $Q^+ = S + R'Q$ ($S = R = 1$ not allowed)
The SR Latch (cont)

- Characteristic Equation
  - $Q^+ = S + R'Q$ ($S = R = 1$ not allowed)
    - $Q$ becomes 1 when $S = 1$, $R = 0$
    - Stays $Q$ when $S = R = 0$
    - $Q$ becomes 0 when $S = 0$, $R = 1$

---

The SR Latch (cont)

- State Table

<table>
<thead>
<tr>
<th>PS (Q)</th>
<th>SR=00</th>
<th>01</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>X</td>
</tr>
</tbody>
</table>
The SR Latch (cont)

- State Diagram

![State Diagram](image)

The SR Latch with NANDS

- NAND Based $S’R’$ Latch
  - $S’ = R’ = 0$ not allowed
  - Either input = 0 forces output to 1

![NAND Based Latch Diagram](image)
The Gated SR Latch

- Also known as “transparent” latch
  - Output follows input (transparent) when enabled

The Gated SR Latch (cont)

- Timing Diagram
The Gated SR Latch (cont)

- NAND Implementation

![Gated SR Latch Diagram]

The Gated Data (D) Latch

- NAND Implementation of transparent D latch

![Gated Data (D) Latch Diagram]
The Gated D Latch

- Timing Diagram

![Timing Diagram](image)

The Edge Triggered D Flip-Flop

- The D Flip-Flop
  - Input D, latched and passed to Q on clock edge
  - Rising edge triggered or falling edge triggered
    - Characteristic table and function

<table>
<thead>
<tr>
<th>D</th>
<th>Q'</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

![Truth Table](image)
The Edge Triggered D Flip-Flop

- Most commonly used flip-flop
- Output follows input after clock edge
  - \( Q \) and \( Q^* \) change only on clock edge
  - Timing diagram for negative edge triggered flip-flop


The D Flip-Flop

- State Table

<table>
<thead>
<tr>
<th>PS (Q)</th>
<th>NS (Q*) D = 0</th>
<th>D = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
The D Flip-Flop (cont)

- State Diagram

The Master-Slave D Flip-Flop

- Construct edge triggered flip-flop from 2 transparent latches
  - Many other topologies for edge triggered flip-flops
  - Falling edge triggered (below)
The Master-Slave D Flip-Flop (cont)

- Timing Diagram
  - Falling edge triggered

- A Second Timing Diagram
  - Rising edge triggered
The Edge Triggered D Flip-Flop

- “True” Edge Triggered D Flip-Flop
  - Never transparent (unlike Master Slave)

The Edge Triggered D Flip-Flop

- Operation of Flip-Flop
Types of D Flip-Flops

- Gated, Positive Edge and Negative Edge

Timing Parameters

- CLK → Q
  - Delay from clock edge (CLK) to valid (Q, Q*) output
- Setup time $t_{su}$
  - Stable, valid data (D) before clock edge (CLK)
- Hold time $t_{hold}$
  - Stable, valid data (D) after clock edge (CLK)
Maximum Frequency

- Maximum frequency (minimum clock period) for a digital system
  - CLK → Q + propagation delay + $t_{su}$

Counter Design with D Flip-Flops

- Design Example #1: Modulo 3 counter
  - 00 → 01 → 10 ↓
  - Requires 2 flip-flops
    - One for each “state variable”
Counter Design with D Flip-Flops

- State Diagram

```
00 01

Transitions on clock edge

10
```

- State Table

<table>
<thead>
<tr>
<th>PS</th>
<th>NS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Counter Design with D Flip-Flops

- Next State Maps

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A' = B</th>
<th>B' = A'B'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</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>X</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>0</td>
</tr>
</tbody>
</table>

Implementation with D Flip-Flops

- What are the D inputs to flip-flops A and B?
  - Recall characteristic equation for D flip-flop
    - \( Q^+ = D \)
    - Therefore, \( A^+ = B \) \( \rightarrow D_A = B \)
    - and… \( B^+ = A'B' \) \( \rightarrow D_B = A'B' \)
Counter Design with D Flip-Flops

- Implementation with positive edge triggered flip-flops
  - Timing diagram

```
00 01 10 00
01
```
Counter Design with D Flip-Flops

- Design Example #2:
  - Modulo 3 counter with up/down* input
    - Counter counts up with input = 1 and down with input = 0
  - Implement with D flip-flops

Counter Design with D Flip-Flops

- State diagram
Counter Design with D Flip-Flops

- State table

<table>
<thead>
<tr>
<th>U</th>
<th>A</th>
<th>B</th>
<th>A'</th>
<th>B'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<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>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

- Next state maps and flip-flop inputs

\[ A' = D_A = U'B + U'A'B' \]
\[ B' = D_B = U'A + UA'B' \]