CSE140: Components and Design Techniques for Digital Systems

Tajana Simunic Rosing
Where we are now...

- What we covered last time:
  - ALUs, SR Latch

- What we’ll do next
  - Latches and FlipFlops (FFs)
  - Registers

- Upcoming deadlines:
  - ZyBook today: Sec 4.3-6, Tuesday: Sec 5.1-6
  - HW#4 assigned, due next Tuesday
  - Quiz #4 today

- Textbook references:
  - chap 3, Sec 6.3
SR Latch Analysis

\(- S = 1, R = 0: \text{ Set} \)
then \(Q = 1\) and \(\overline{Q} = 0\)

\(- S = 0, R = 1: \text{ Reset} \)
then \(\overline{Q} = 1\) and \(Q = 0\)

Sources: TSR, Katz, Boriello & Vahid
SR Latch Analysis

- $S = 0$, $R = 0$: Hold
  
  then $Q = Q_{\text{prev}}$

- Memory!

- $S = 1$, $R = 1$:
  
  then $Q = 0$, $\overline{Q} = 0$

- Invalid State

$Q \neq \text{NOT } Q$

Sources: TSR, Katz, Boriello & Vahid
What if a kid presses both call and cancel & then releases them?

- If S=1 and R=1 at the same time and then released, Q=?
  - Can also occur also due to different delays of different paths
  - Q may oscillate and eventually settle to 1 or 0 due to diff. path delay

Sources: TSR, Katz, Boriello & Vahid
SR Latch Symbol

- SR stands for Set/Reset Latch
  - Stores one bit of state ($Q$)
- Control what value is being stored with $S$, $R$ inputs
  - **Set:** Make the output 1
    $$(S = 1, \ R = 0, \ Q = 1)$$
  - **Reset:** Make the output 0
    $$(S = 0, \ R = 1, \ Q = 0)$$
  - **Hold:** Keep data stored
    $$(S = 0, \ R = 0, \ Q = Q_{previous})$$

Sources: TSR, Katz, Boriello & Vahid
**SR Latch Characteristic Equation**

To analyze, break the feedback path

\[
Q(t + \Delta) = S + R' Q(t)
\]

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Q(t)</th>
<th>Q(t+\Delta)</th>
</tr>
</thead>
<tbody>
<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>1</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>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
</tbody>
</table>

hold
reset
set
not allowed

characteristic equation

**State Diagram**

**SR Latch Symbol**

Sources: TSR, Katz, Boriello & Vahid
Avoiding S=R=1 Part 1:
Level-Sensitive SR Latch

- Add input “C”
  - Change C to 1 only after S and R are stable
  - C is usually a clock (CLK)
Clocks

- **Clock** -- Pulsing signal for enabling latches; ticks like a clock
- **Synchronous** circuit: sequential circuit with a clock

- **Clock period**: time between pulse starts
  - Above signal: period = 20 ns
- **Clock cycle**: one such time interval
  - Above signal shows 3.5 clock cycles
- **Clock duty cycle**: time clock is high
  - 50% in this case
- **Clock frequency**: 1/period
  - Above: freq = 1 / 20ns = 50MHz;

<table>
<thead>
<tr>
<th>Freq</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 GHz</td>
<td>0.01 ns</td>
</tr>
<tr>
<td>10 GHz</td>
<td>0.1 ns</td>
</tr>
<tr>
<td>1 GHz</td>
<td>1 ns</td>
</tr>
<tr>
<td>100 MHz</td>
<td>10 ns</td>
</tr>
<tr>
<td>10 MHz</td>
<td>100 ns</td>
</tr>
</tbody>
</table>

Sources: TSR, Katz, Boriello & Vahid
Clock question

The clock shown in the waveform below has:

A. Clock period of 4ns with 250MHz frequency  ✔
B. Clock duty cycle 75%  ✔
C. Clock period of 1ns with 1GHz frequency
D. A. & B.
E. None of the above
Avoiding $S=R=1$ Part 2: Level-Sensitive D Latch

- SR latch requires careful design so $SR=11$ never occurs
- D latch helps by inserting the inverter between S & R inputs
  - Inserted inverter ensures $R$ is always the opposite of $S$ when $C=1$
**D Latch Truth Table**

<table>
<thead>
<tr>
<th>CLK</th>
<th>D</th>
<th>$\overline{D}$</th>
<th>S</th>
<th>R</th>
<th>Q</th>
<th>$\overline{Q}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$Q_{prev}$ $\overline{Q_{prev}}$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Hold** $Q = \overline{D}$

Sources: TSR, Katz, Boriello & Vahid
D Latch Summary

• Two inputs: \( CLK, D \)
  – \( CLK \): controls when the output changes
  – \( D \) (the data input): controls what the output changes to

• Function
  – When \( CLK = 1 \),
    \( D \) passes through to \( Q \) (transparent)
  – When \( CLK = 0 \),
    \( Q \) holds its previous value (opaque)

• (Mostly) avoids invalid case \( Q = Q' \)

Sources: TSR, Katz, Boriello & Vahid
Level-Sensitive D Latches

Assume that data in all latches is initially 0. Input $Y=1$ and Clk transitions from 0->1. When Clk=0 again, the stored values in latches are:

A. $Q_1=1$, $Q_2=0$, $Q_3=0$, $Q_4=0$ for both clock A & B
B. $Q_1=1$, $Q_2=1$, $Q_3=1$, $Q_4=1$ for clock A
   $Q_1=1$, $Q_2=0$, $Q_3=0$, $Q_4=0$ for clock B
C. $Q_1=1$, $Q_2=1$, $Q_3=1$, $Q_4=1$ for both clocks
D. More information is needed to determine the answer
E. None of the above

Sources: TSR, Katz, Boriello & Vahid
**D Flip-Flop Design & Timing Diagram**

- **Flip-flop**: Bit storage that stores on the clock edge, not level
- Master-slave design: master loads when Clk=0, then slave when Clk=1

Sources: TSR, Katz, Boriello & Vahid
D Flip-Flop: Characteristic Equation

$Q(t+1) = D(t)$

<table>
<thead>
<tr>
<th>Id</th>
<th>D Q(t)</th>
<th>Q(t+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0 1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1 0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1 1</td>
<td>1</td>
</tr>
</tbody>
</table>

Sources: TSR, Katz, Boriello & Vahid
Rising vs. Falling Edge D Flip-Flop

Symbol for rising-edge triggered D flip-flop

Symbol for falling-edge triggered D flip-flop

The triangle means clock input, edge triggered

Internal design: Just invert servant clock rather than master

Sources: TSR, Katz, Boriello & Vahid
Enabled D-FFs

- **Inputs:** $CLK, D, EN$
  - The enable input ($EN$) controls when new data ($D$) is stored

- **Function**
  - $EN = 1$: $D$ passes through to $Q$ on the clock edge
  - $EN = 0$: the flip-flop retains its previous state

---

**Internal Circuit**

**Symbol**

Sources: TSR, Katz, Boriello & Vahid
Additional D-FF Features

- **Reset (set state to 0) – R**
  - synchronous: $D_{new} = R' \cdot D_{old}$ (when next clock edge arrives)
  - asynchronous: doesn't wait for clock

- **Preset or set (set state to 1) – S (or sometimes P)**
  - synchronous: $D_{new} = D_{old} + S$ (when next clock edge arrives)
  - asynchronous: doesn't wait for clock

- **Both reset and preset**
  - $D_{new} = R' \cdot D_{old} + S$ (set-dominant)
  - $D_{new} = R' \cdot D_{old} + R'S$ (reset-dominant)

- **Selective input capability (input enable or load) – LD or EN**
  - multiplexor at input: $D_{new} = LD' \cdot Q + LD \cdot D_{old}$
  - load may or may not override reset/set (usually R/S have priority)

- **Complementary outputs – Q and Q’**
D Flip-Flops

Assume that the data in all D-FFs is initially 0. Input Y=1. When Clk goes from 0->1, the stored values in D-FFs are:

A. Q1=1, Q2=0, Q3=0, Q4=0 for both clock A & B
B. Q1=1, Q2=1, Q3=1, Q4=1 for clock A
C. Q1=1, Q2=0, Q3=0, Q4=0 for clock B
D. More information is needed to determine the answer
E. None of the above
Bit Storage Overview

**SR latch**
- S (set)
- R (reset)
- S=1 sets Q to 1, R=1 resets Q to 0.
- Problem: SR=11 yield undefined Q.

**Level-sensitive SR latch**
- S and R only have effect when C=1.
- We can design outside circuit so SR=11 never happens when C=1.
- Problem: avoiding SR=11 can be a burden.

**D latch**
- SR can’t be 11 if D is stable before and while C=1, and will be 11 for only a brief glitch even if D changes while C=1.
- Problem: C=1 too long propagates new values through too many latches.
- too short may not enable a store.

**D flip-flop**
- Only loads D value present at rising clock edge, so values can’t propagate to other flip-flops during same clock cycle.
- Tradeoff: uses more gates internally than D latch, and requires more external gates than SR – but gate count is less of an issue today.

Sources: TSR, Katz, Boriello & Vahid
Comparison of latches and flip-flops

- **D Q**
- **CLK**

Positive edge-triggered flip-flop

- **D Q**
- **G CLK**

Level-sensitive latch

Sources: TSR, Katz, Boriello & Vahid
Registers and Counters
Building blocks with FFs: Basic Register

Sources: TSR, Katz, Boriello & Vahid
Shift register

• Holds & shifts samples of input
• Combinational function of input samples