**Introduction to Digital Design**

Week 7: Clock, Latches, and Flip-Flops

Yaoyu Zhang
Assistant Professor
University of Hawai‘i at Mānoa
Department of Electrical Engineering

---

**Overview**

- Sequential circuit
  - Output depends not just on present inputs but on past sequence of inputs.
- SR Latch
  - Feedback circuit for bit storage.
  - Race condition and level-sensitive latch.
- D Latch and D Flip-Flop
  - D Latch: inserted inverter ensures R always opposite of S.
  - D Flip-Flop: bit storage that stores on clock edge.
- Clock signal
  - Flip-flop to generates periodic pulsing signal.
- Basic register

---

**Introduction**

- Sequential circuit
  - Output depends not just on present inputs (as in combinational circuit), but on past sequence of inputs.
  - Stores bits, also known as having “state.”
- Design a new building block, a flip-flop, to store one bit.
- This chapter will:
  - Combine flip-flops to build multi-bit storage, register
  - Describe sequential behavior with finite state machines
  - Convert a finite state machine to a controller – sequential circuit with a register and combinational logic

---

**Storing One Bit – Flip-Flops**

**Example Requiring Bit Storage**

- Flight attendant call button
  - Press call: light turns on
  - Stays on after button released
- Press cancel: light turns off
  - Stays off after button released

Logic gate circuit to implement this?

Doesn't work. Q=1 when Call=1, but doesn't stay 1 when Call returns to 0

Need some form of “feedback” in the circuit

---

**First attempt at Bit Storage**

- Need some sort of feedback
  - Does circuit on the right do what we want?
    - No: Once Q becomes 1 (when S=1), Q stays 1 forever – no value of S can bring Q back to 0

---

**Bit Storage Using an SR Latch**

- Does the circuit to the right, with cross-coupled NOR gates, do what we want?
  - Yes! How did someone come up with that circuit?
    - Maybe just trial and error, a bit of insight...
Example Using SR Latch for Bit Storage

- SR latch can serve as bit storage in previous example of flight-attendant call button
  - Call=1: sets Q to 1
  - Cancel=1: resets Q to 0
- But, there’s a problem...

Problem with SR Latch

- Problem
  - If S=1 and R=1 simultaneously, we don’t know what value Q will take

Solution: Level-Sensitive SR Latch

- Add enable input “C”
- Only let S and R change when C=0
  - Ensure circuit in front of SR never sets SR=11, except briefly due to path delays
  - Set C=1 after time for S and R to be stable
  - When C becomes 1, the stable S and R value passes through the two AND gates to the SR latch’s S1 R1 inputs.

Level-Sensitive D Latch

- SR latch requires careful design to ensure SR=11 never occurs
- D latch relieves designer of that burden
  - Inserted inverter ensures R always opposite of S
Problem with Level-Sensitive D Latch

- D latch still has problem (as does SR latch)
  - When C=1, through how many latches will a signal travel?
  - Depends on how long C=1
    - Clk_A – signal may travel through multiple latches
    - Clk_B – signal may travel through fewer latches

D latch

- D1 Q1
- D2 Q2
- D3 Q3
- D4 Q4
- C1 C2 C3 C4

Y

Clk

Clk_A

Clk_B

Problem with Level-Sensitive D Latch

- FLP: Bit storage that stores on clock edge
  - One design – master-servant
    - Clk = 0 – master enabled, loads D, appears at Qm. Servant disabled.
    - Clk = 1 – Master disabled, Qm stays same. Servant latch enabled, loads Qm, appears at Qs.
    - Thus, value at D (and hence at Qm) when Clk changes from 0 to 1 gets stored into servant

D Latch vs. D Flip-Flop

- Latch is level-sensitive
  - Stores D when C=1
- Flip-flop is edge triggered
  - Stores D when C changes from 0 to 1
- Saying “level-sensitive latch” or “edge-triggered flip-flop” is redundant
- Comparing behavior of latch and flip-flop:
Clock Signal
- Flip-flop Clk inputs typically connect to one clock signal
  - Coming from an oscillator component
  - Generates periodic pulsing signal
  - “Cycle” is duration of 1 period (20 ns), below shows 3.5 cycles

Bit Storage Summary
- SR latch
  - Feature: SR can’t be 11
  - SR latch is derived from S-R flip-flop
  - SR can’t be 11
  - SR latch is derived from S-R flip-flop

Example Using Registers: Temperature Display
- Temperature history display
  - Sensor outputs temperature as 5-bit binary number
  - Timer pulses C every hour
  - Record temperature on each pulse, display last three recorded values

Flight-Attendant Call Button Using D Flip-Flop
- D flip-flop will store bit
- Inputs are Call, Cancel, and present value of D flip-flop, Q

Basic Register
- Typically, we store multi-bit items
  - e.g., storing a 4-bit binary number
  - Register: multiple flip-flops sharing clock signal
    - From this point, we’ll use registers for bit storage
    - No need to think of latches or flip-flops
    - But now you know what’s inside a register

Example Using Registers: Temperature Display
- Use three 5-bit registers
  - Sensor outputs temperature as 5-bit binary number
  - Timer pulses C every hour
  - Display last three recorded temperatures

Temperature history display
- Sensor outputs temperature as 5-bit binary number
- Timer pulses C every hour
- Record temperature on each pulse, display last three recorded values
Summary

• Sequential circuit
  – Output depends not just on present inputs but on past sequence of inputs.

• SR Latch
  – Feedback circuit for bit storage.
  – Race condition and level-sensitive latch.

• D Latch and D Flip-Flop
  – D Latch: inserted inverter ensures R always opposite of S
  – D Flip-Flop: bit storage that stores on clock edge.

• Clock signal
  – Flip-flop to generates periodic pulsing signal.

• Basic register