THE UNIVERSITY OF TEXAS AT DALLAS

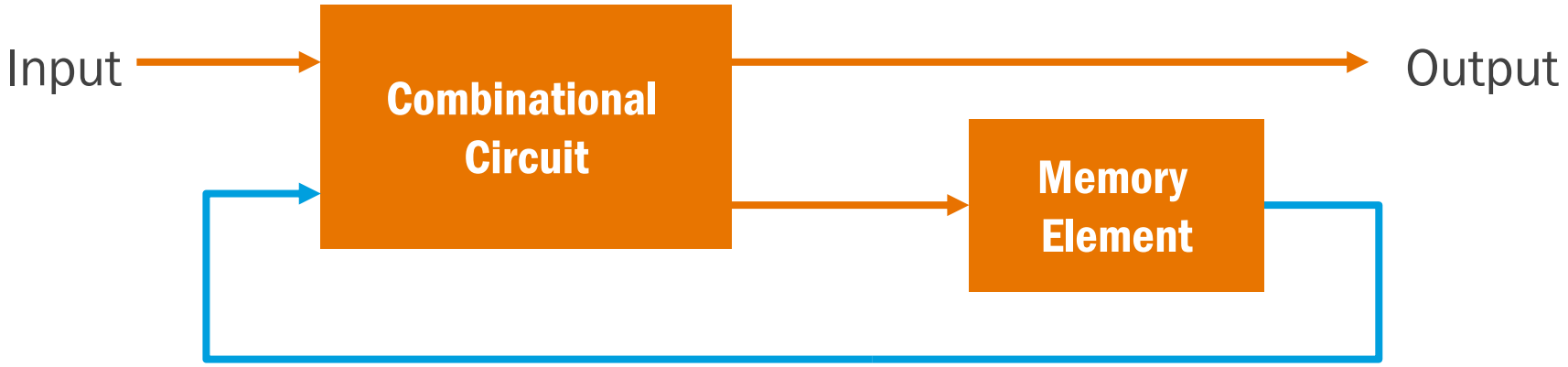
CS4341 Digital Logic & Computer Design

Lecture Notes 12

Omar Hamdy
Assistant Professor
Department of Computer Science

Sequential Circuits

- Sequential circuits are digital circuits that the output depends on:
  - current input (combinational circuit), and
  - system state: a memory device to preserve the output value



Example

- What is the key difference between these two locks?

Combinational: output depends only on input

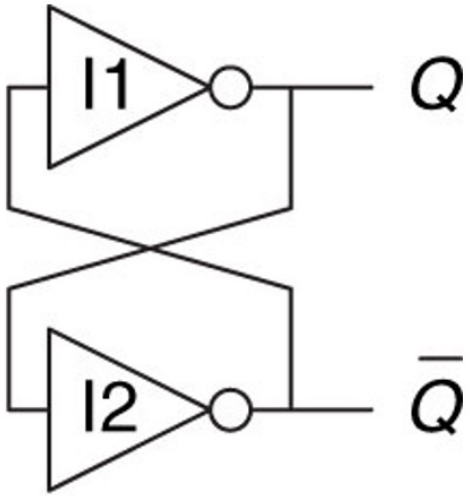
Sequential: output depends on new and previous inputs

Sequential Circuit Types

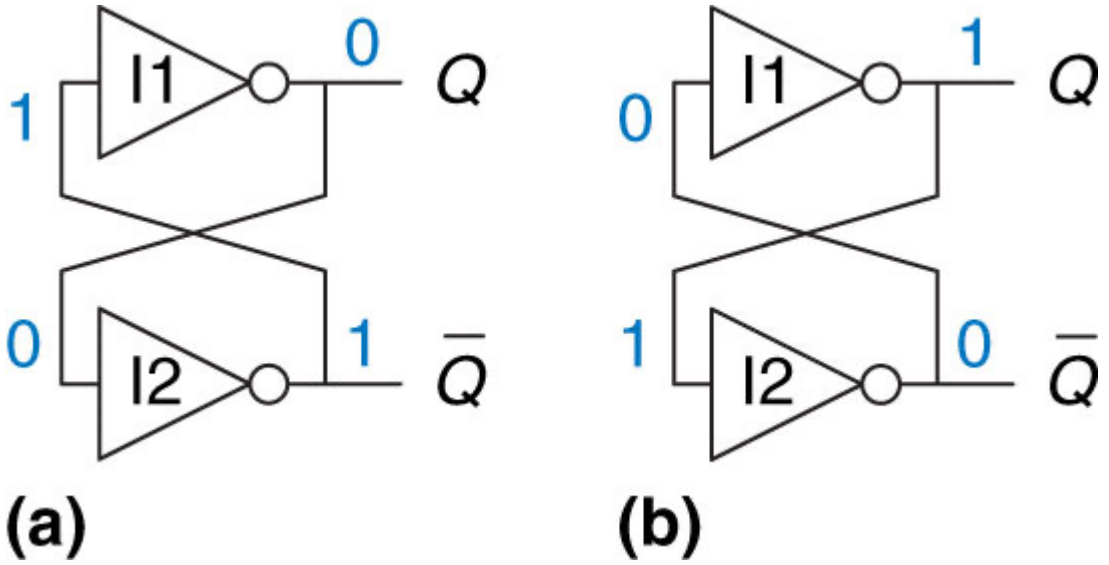
- **Sequential circuits are differentiated by times at which:**
  - storage elements observe their inputs, and
  - storage elements change their state
- **Synchronous sequential circuits:**
  - Behavior is defined from knowledge of inputs at discrete instances of time
  - Storage elements observe inputs and can change state only in relation to a timing signal (clock pulses from a clock)
- **Asynchronous sequential circuits:**
  - Behavior is defined from knowledge of inputs an any instant of time
  - Storage elements observe inputs and can change state at any instant of time

Bistable Memory Building Block

- Bistable is a digital element with two stable states 0 and 1. If the output is 0, it stays 0 (stable), and if output is 1, it stays 1 (stable)



- To analyze this cyclic element, we consider the status when Q is 0 and when Q is 1:

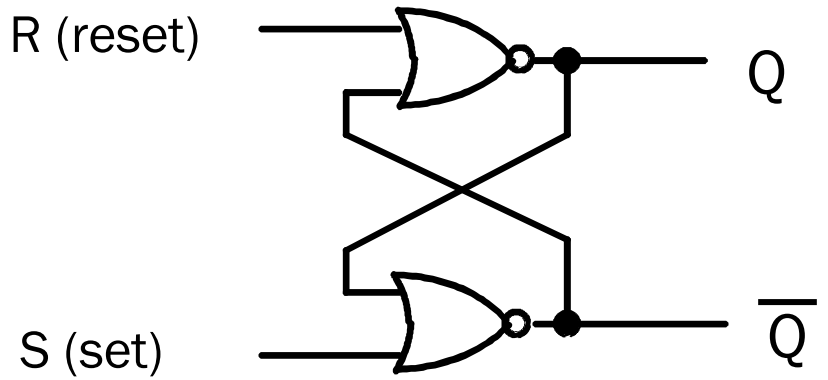


Bistable Memory Building Block

- An Element with N stable states conveys Log_2N of information
- The bistable state element has two stable states, hence it "stores" one bit of information (Q)
- Q holds a value from the past, which can be used to determine the future state of the circuit.
- The basic bistable elements are limited in their use, because we cannot control or change its status.
- More useful bistable elements should contain control inputs

SR Latches

- SR latches are the basic asynchronous memory devices
- SR latches have two control inputs, S (set) and R (reset)

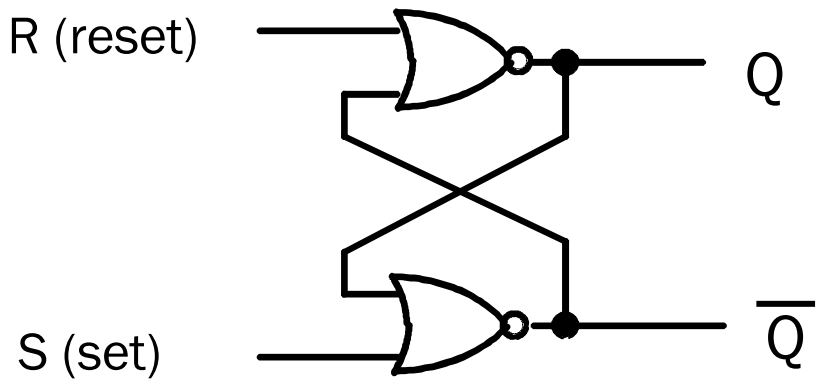


- SR latches have two famous implementations using NOR and NAND gates

SR Latch Behavior Study

- Study of the circuit behavior requires studying the input values as well as the present output values (`Current State CS`) to find the next output values (`Next State NS`)

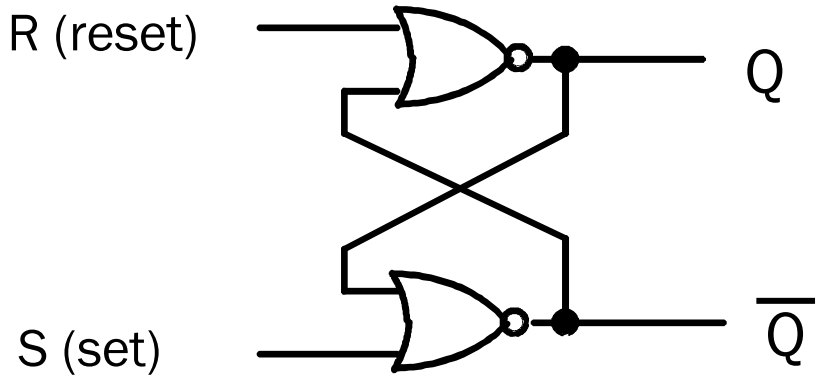
S	R	Q	Q^+	Status
0	0	0	0	Memory
0	0	1	1	Memory
0	1	0	0	Reset
0	1	1	0	Reset
1	0	0	1	Set
1	0	1	1	Set
1	1	0	?	Forbidden
1	1	1	?	Forbidden



SR Latch Behavior Study

- Using the combining row technique, we can eliminate Q (right-most variable) from the truth table, and express the output Q^+ as 0, 1 or Q

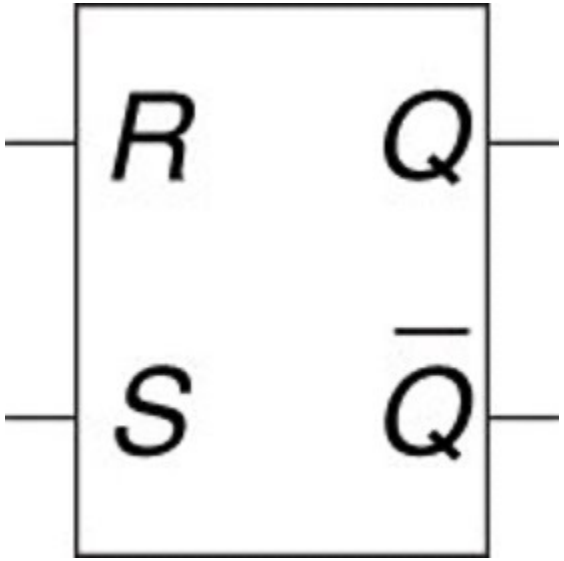
S	R	Q^+	Status
0	0	Q	Memory
0	1	0	Reset
1	0	1	Set
1	1	?	Forbidden



- Algebraically: $Q^+ = S + R'Q$ (WHY? – Home Exercise)
- Home exercise: build the SR latch using NAND gates

SR Latch Memory Element

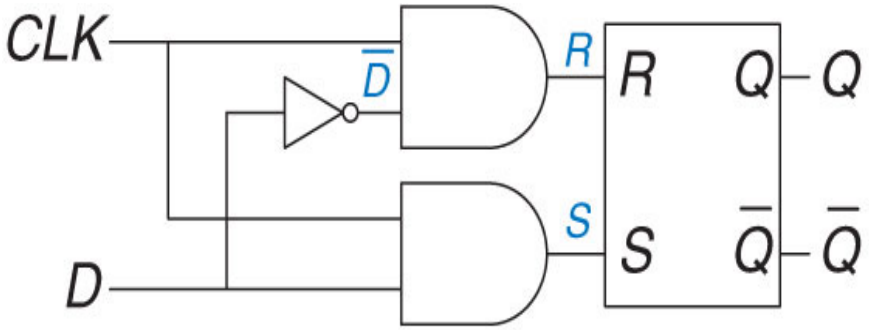
- SR latch is a bistable element with one bit of state stored in Q. However, the state can be controlled through the S and R inputs. When R is asserted, the state is reset to 0. When S is asserted, the state is set to 1. When neither is asserted, the state retains its old value



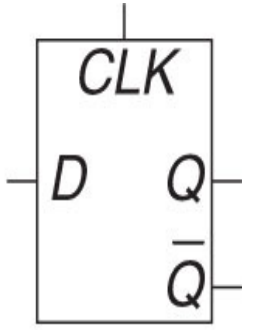
D Latch

- SR latches have multiple design concerns:
  - The undesired state with S & R are both 1.
  - The inputs are also responsible for the timing when the change should happen, and to preserve the status. Ideally, they should only be responsible for the status change.
- D Latches solve these two problems SR Latches have:
  - Instead of S/R, only one input and its inverted value are used
  - Another input *CLK* is responsible for when the element should consider the change.

D Latch



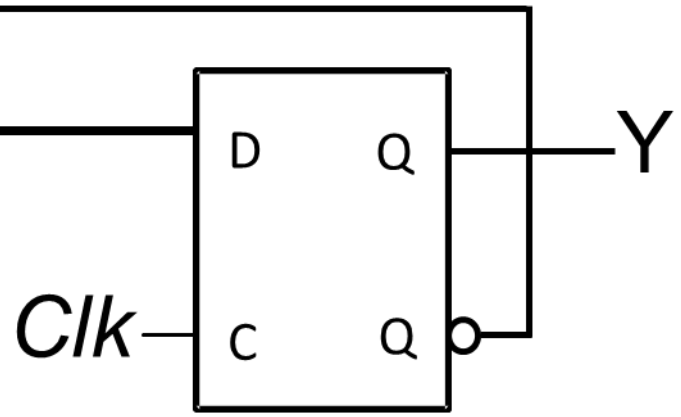
CLK	D	$\overline{D}$	S	R	Q	$\overline{Q}$
0	X	$\overline{X}$	0	0	Q_{prev}	$\overline{Q}_{prev}$
1	0	1	0	1	0	1
1	1	0	1	0	1	0

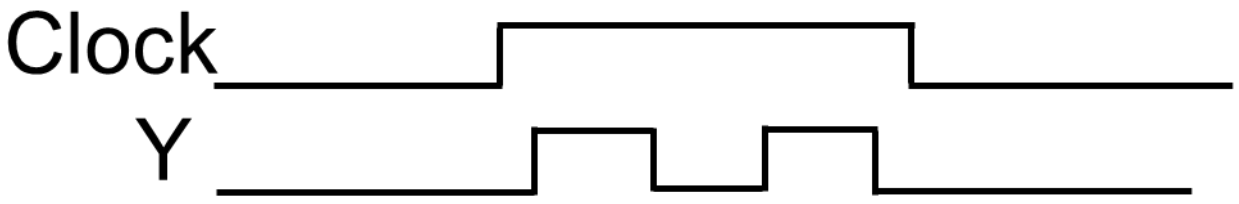


- When *CLK* = 1, the latch is *transparent*, and the data at D flows to Q.
- When *CLK* = 0, the latch is *opaque*, and blocks the data at D to flow to Q. Hence, the D latch retains its current state.
- Algebraically: $Q^+ = D$

D Latch Timing Problem

- Consider the following D Latch circuit, how will the circuit behave during an active *Clk* cycle? (Assume Y is initially 0)

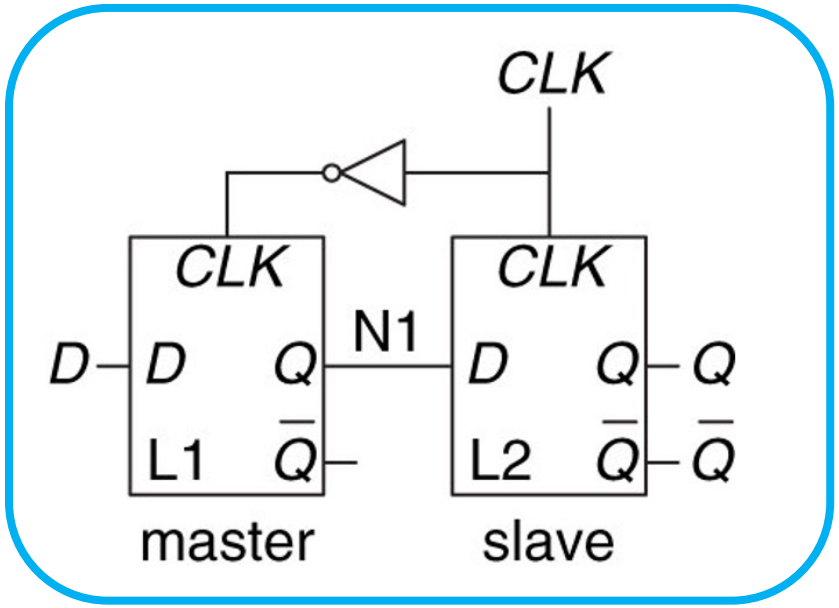


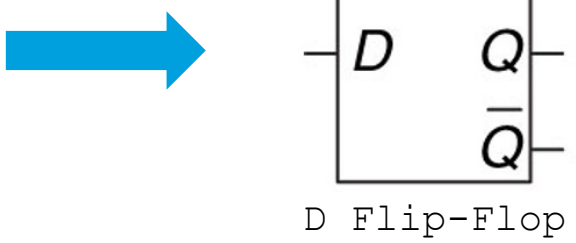


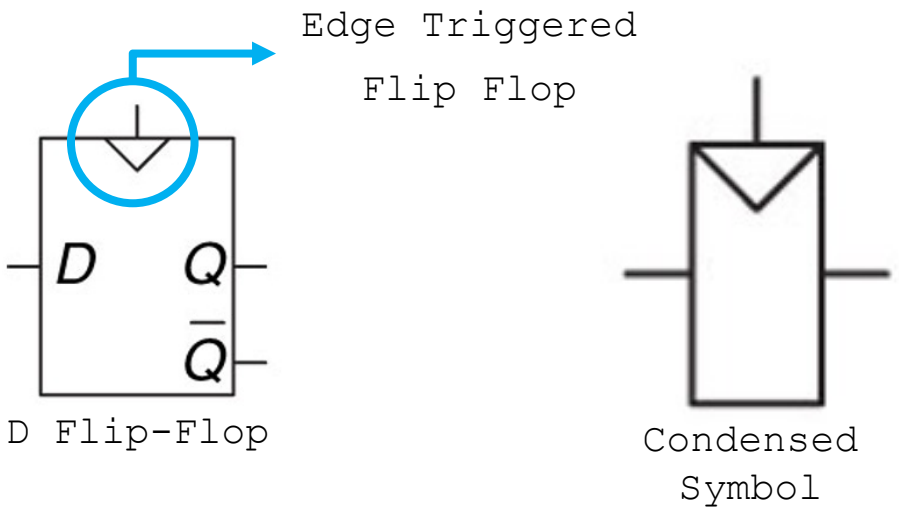
Undesirable behavior

D Flip-Flop

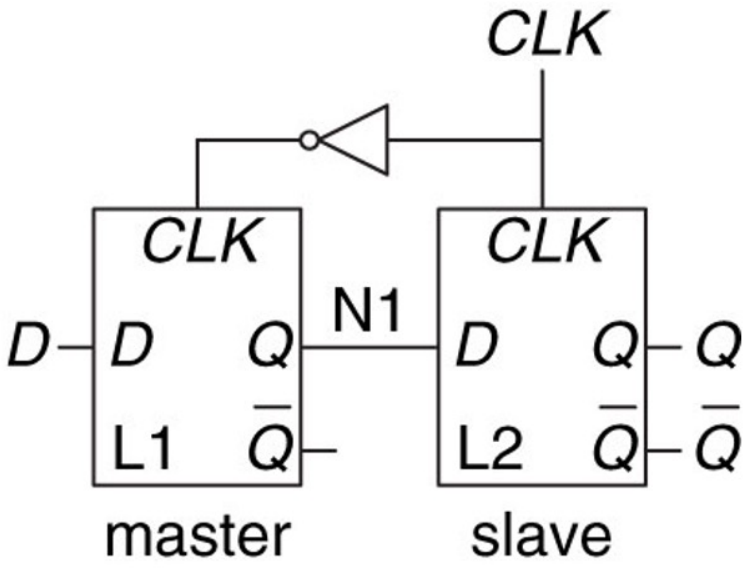
- The solution of the D-Latch undesired behavior is to break the closed path from Y to itself.
- This is done using D flip-flop, which is built from two back-to-back D latches controlled by complementary clocks

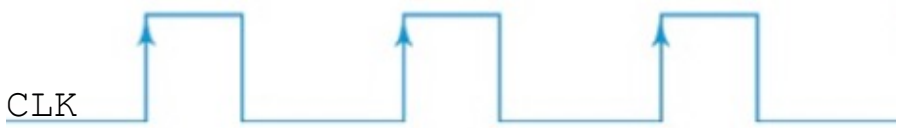






D Flip-Flop Analysis





- When CLK is 0, the master latch is transparent, and the slave is opaque. Hence, D value is propagated to N1
- When CLK is 1, the slave latch is transparent, and the master is opaque. Hence, N1 value is propagated to Q.
- From timing perspective: Whatever value was at N immediately before the *CLK* "rises" is copied to Q immediately after the *CLK* "rises".
- Therefore, a D flip-flop, copies D to Q on the *rising edge* of the *CLK*, and remembers (preserves) its state at all other times.

To Do List

- Review lecture notes
- Study chapter 3 through 3.4.4