CMPE 212
Principles of Digital Design

Lecture 16

Introduction to Sequential Devices

March 30, 2016

www.csee.umbc.edu/~younis/CMPE212/CMPE212.htm

Lecture's Overview

- ***Previous Lecture:***
  - ➔Programmable logic devices
    (Concept of logic matrix, example applications)
  - ➔ Logic using Diode Biasing
    (AND-OR circuits using diodes settings)
  - ➔ Programmable devices and how to configure them
    (fuse management, port configurations and applications)
  - ➔ PLA, PAL, ROM and FPGA devices
    (architectures, key features, examples)
- ***This Lecture***
  - ➔ Introduction to sequential circuits
  - ➔ Memory devices

Sequential Logic

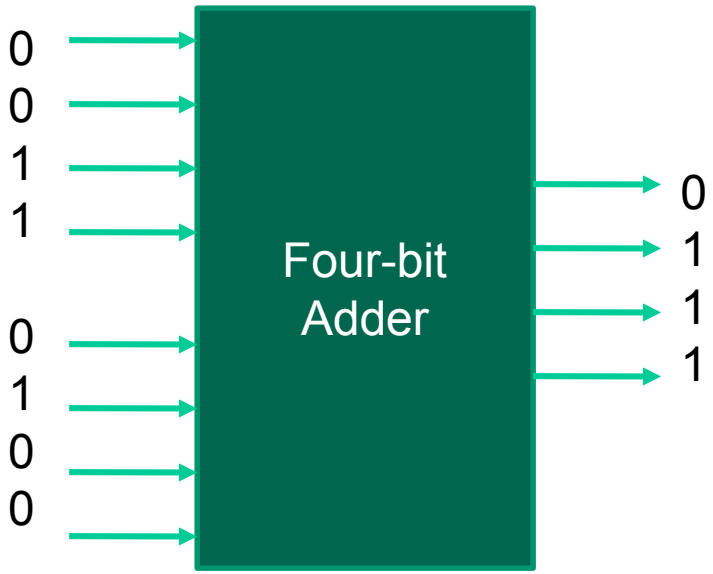
- The combinational logic circuits we have been studying so far have no memory ➔ The outputs always follow the inputs
- There is a need for circuits with memory, which behave differently depending upon their previous state
- An example is a vending machine, which must remember how many and what kinds of coins have been inserted. The machine should behave according to not only the current coin inserted, but also upon how many and what kinds of coins have been inserted previously
- These are referred to as finite state machines, because they can have at most a finite number of states
- A finite state machine responds to an input by generating an output that is function of that input and the current state and by switching to a new state
- Finite state machine are commonly used for the design of digital controllers (typically called sequencer)

* Slide is courtesy of M. Murdocca and V. Heuring

Combinational vs. Sequential

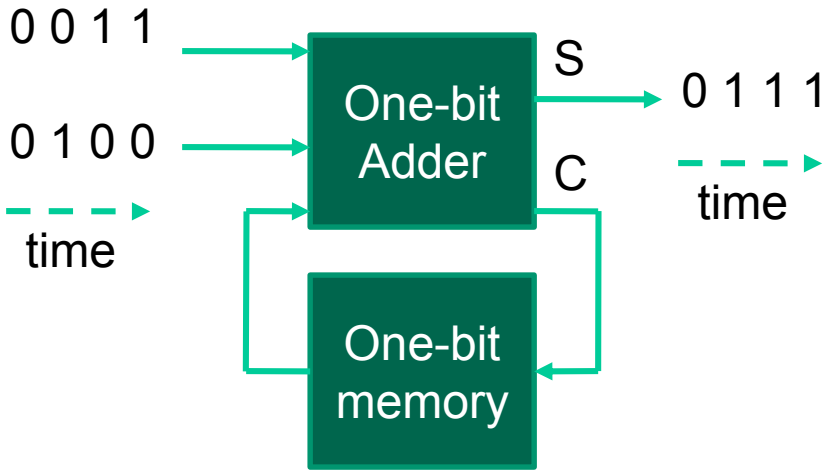
Combinational circuit:

- Output is a function of input
- No memory
- Example: a conventional 4-bit ripple-carry adder (parallel adder)



Sequential circuit:

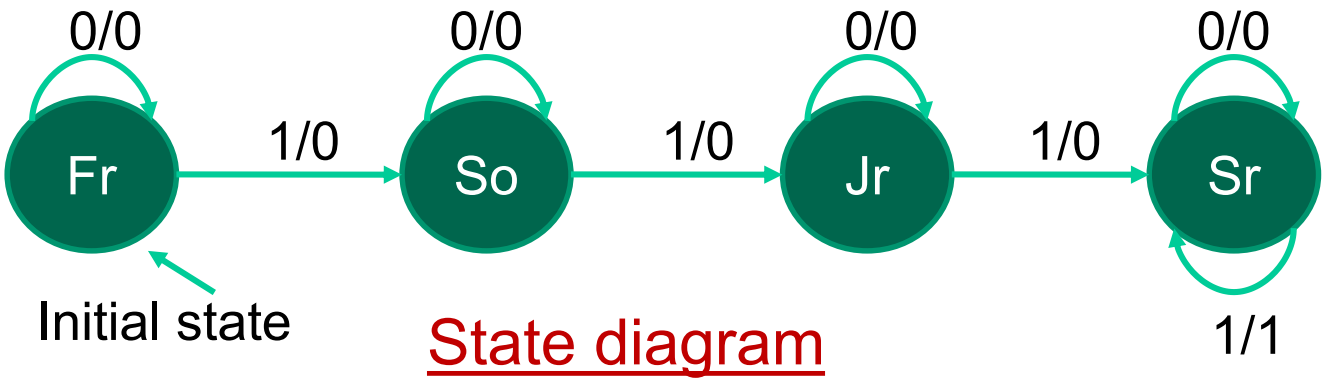
- Output is a function of input and something else stored in the circuit
- Internal memory
- Example: serial adder



1. Memory initialized to 0 (initial carry = 0)
2. Time synchronization of Inputs, output, and memory (clock)

* Contents is courtesy of Vishwani D. Agrawal

Example of Sequential Circuit



State diagram

State Table (Excitation Table)

Input	Present State	Next State	Output
0	Fr	Fr	0
0	So	So	0
0	Jr	Jr	0
0	Sr	Sr	0
1	Fr	So	0
1	So	Jr	0
1	Jr	Sr	0
1	Sr	Sr	1

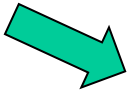
Four-year degree program

- A student can be in four states:
  (Fr, So, Jr, Sr)
- One-bit yearly input: 1 (completed), or
  0 (in progress)
- Output = 1 (degree completed), or
  0 (in progress)

* Contents is courtesy of Vishwani D. Agrawal

State Table (Alternative Form)

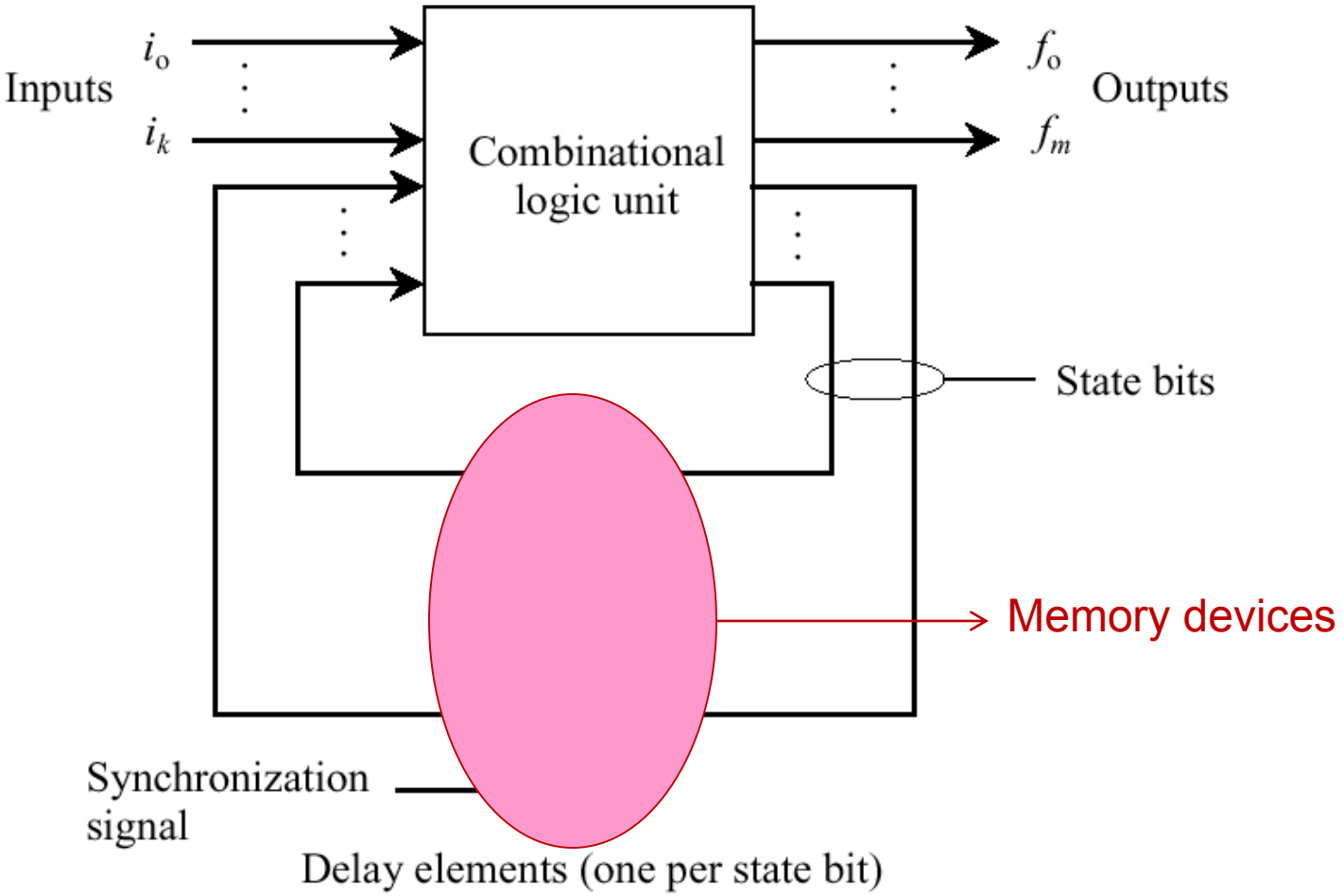
Input	Present State	Next State	Output
0	Fr	Fr	0
0	So	So	0
0	Jr	Jr	0
0	Sr	Sr	0
1	Fr	So	0
1	So	Jr	0
1	Jr	Sr	0
1	Sr	Sr	1



Present state \ Inputs	0	1
Fr	Fr/0	So/0
So	So/0	Jr/0
Jr	Jr/0	Sr/0
Sr	Sr/0	Sr/1

Next state/output

Classical Model of a Finite State Machine

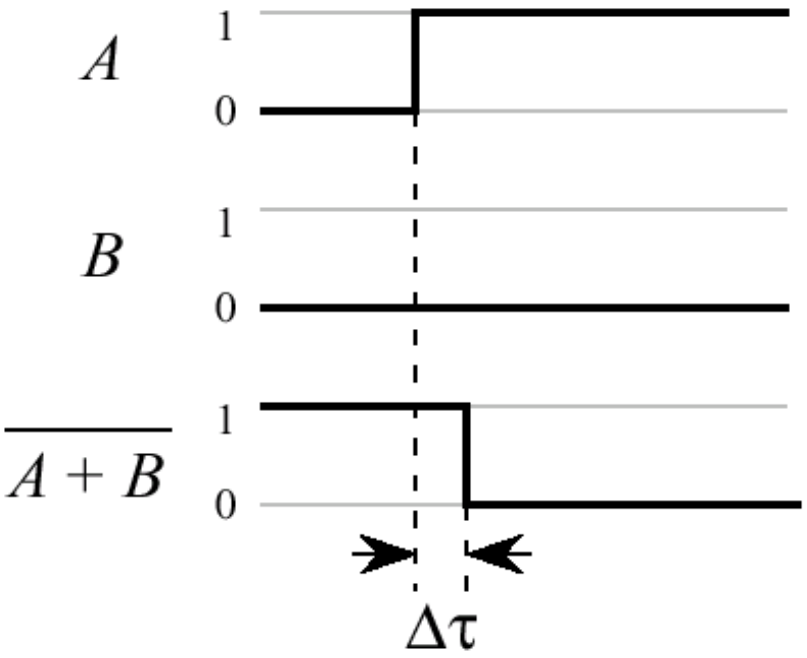


A FSM is composed of a combinational logic unit and delay elements (called latches or flip-flops) in a feedback path, which maintains state information

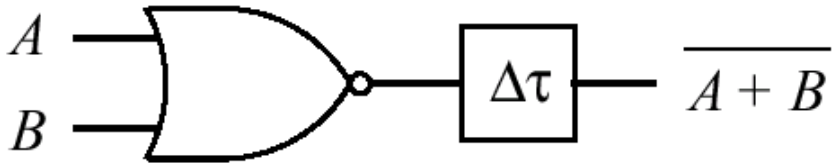
* Slide is courtesy of M. Murdocca and V. Heuring

NOR Gate with Lumped Delay

- The delay between input and output (which is lumped at the output for the purpose of analysis) is at the basis of the functioning of an important memory element, the latch
- The lumped gate delay is not normally indicated in circuit diagram but its presence is implied



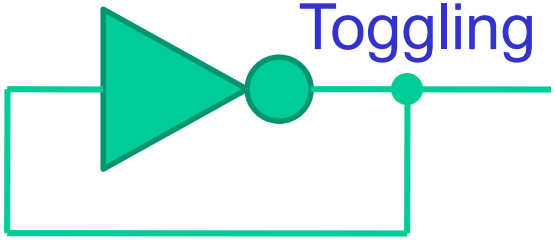
Timing Behavior



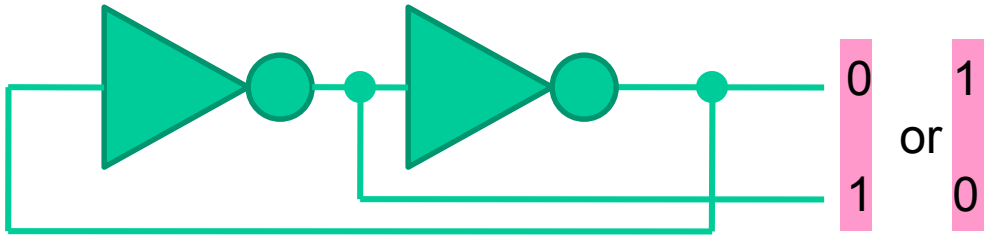
* Slide is courtesy of M. Murdocca and V. Heuring

When Is Circuit Not Combinational?

- When the present input does not completely control output.
- For a logic circuit without feedback, input uniquely determines the output.
- Examples of non-combinational (sequential) circuits:



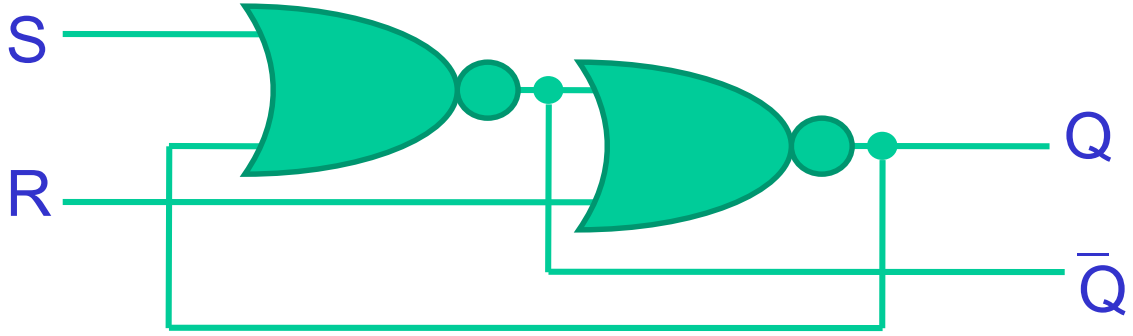
Odd inversions



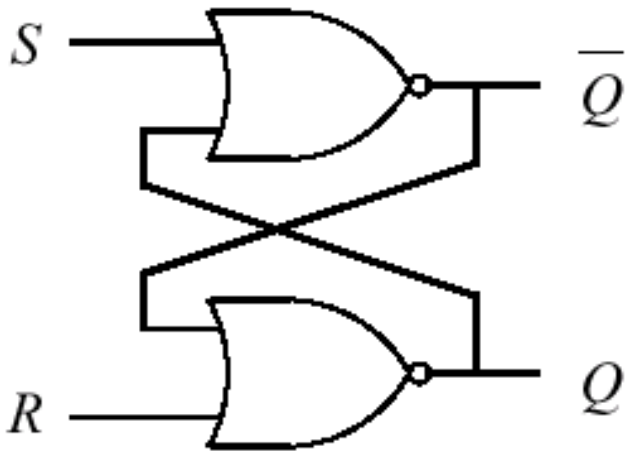
Even inversions

Basic Sequential Circuit: S-R Latch

Feedback loop with even number of inversions (no oscillation)

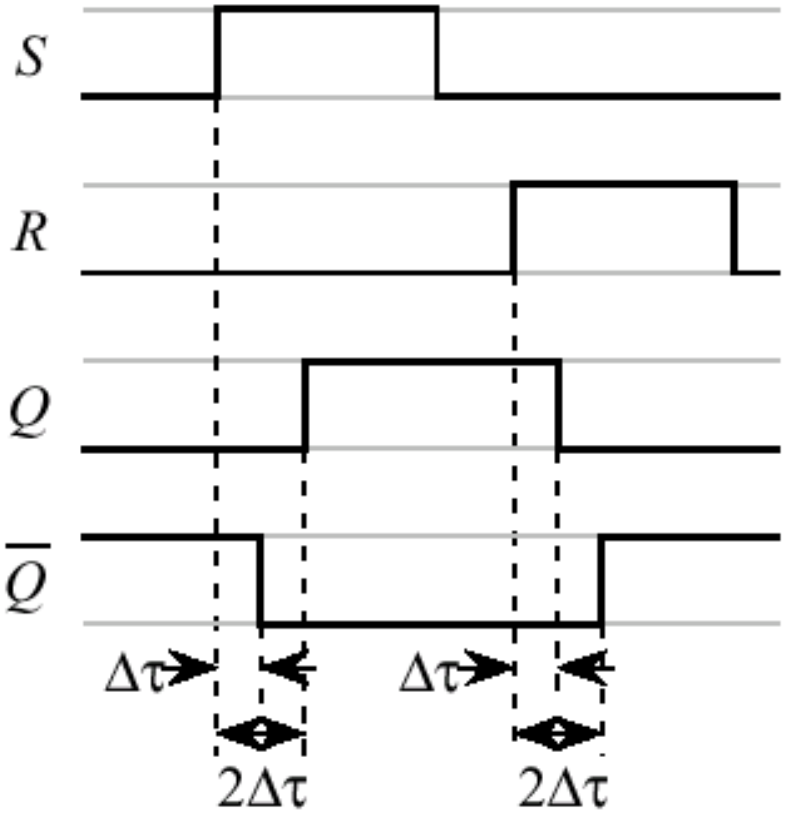


Set-Reset Latch



S=1, R=0 ➔ Q=1, S=0, R=1 ➔ Q=0
S=0, R=0 ➔ Q does not change
S=1, R=1 ➔ (does not make sense)

Q_t	S_t	R_t	Q_{i+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	(disallowed)
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	(disallowed)

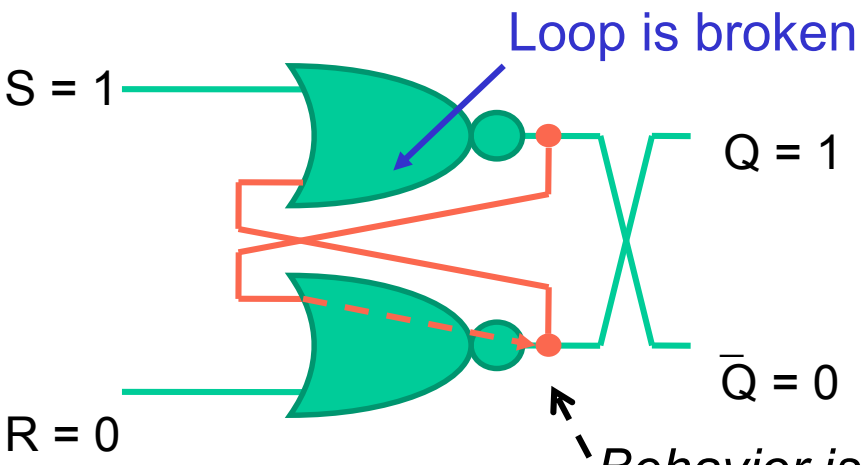


Timing Behavior

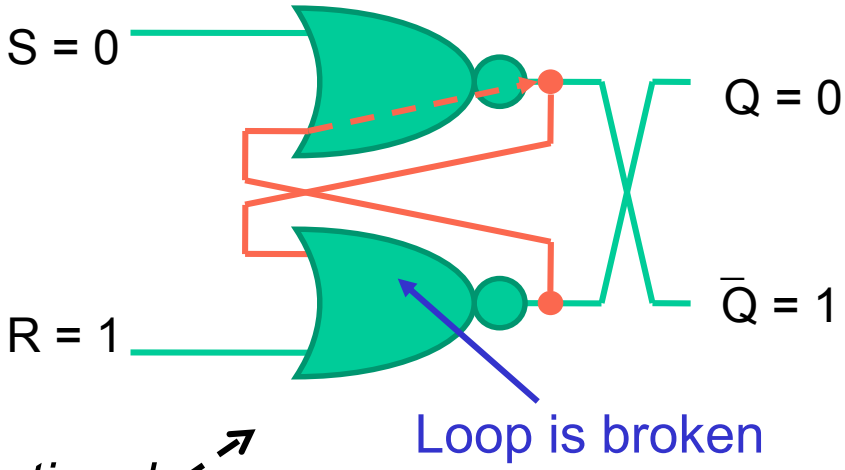
* Slide is courtesy of M. Murdocca and V. Heuring

Set-Reset Latch States

The "Set" State

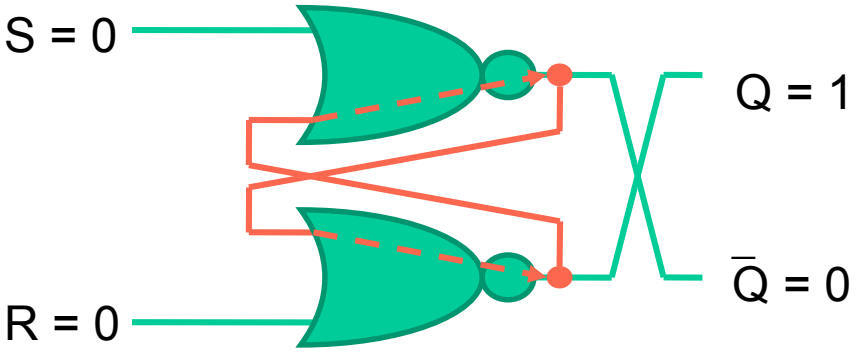


The "Reset" State



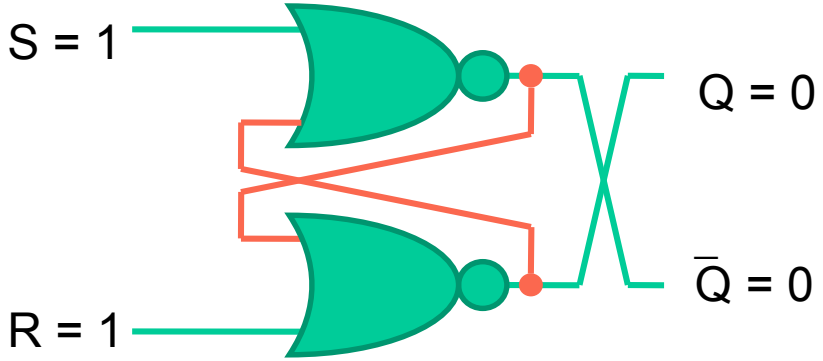
Behavior is combinational

The "Store" State



Loop is activated; behavior is sequential.

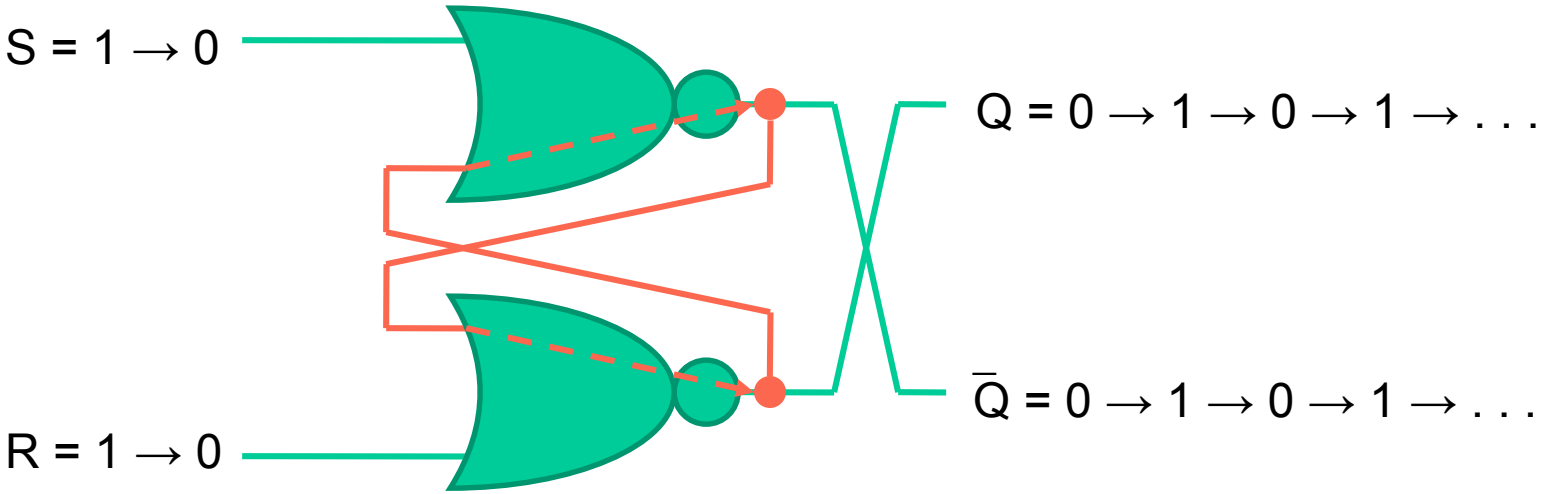
The "Illegal" State



Loop is broken in two places and inconsistent values inserted.

"Illegal" State Cannot Be Stored

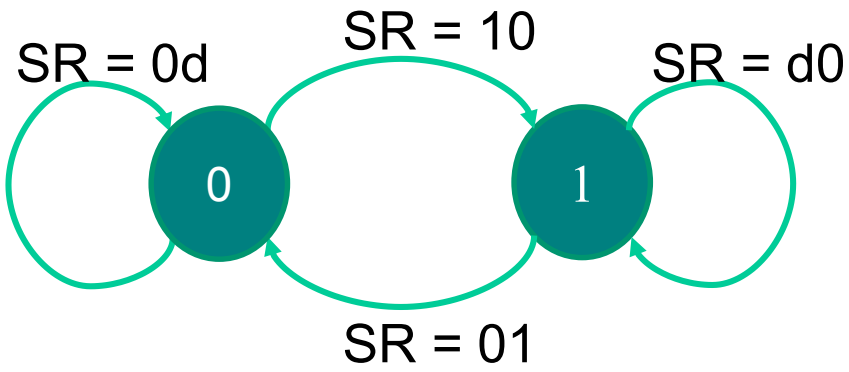
Assume two gates have equal delays.



Output oscillates with a period of loop delay. For unequal gate delays, faster gate will settle to 1 *and slower gate to* 0. *This is known as RACE CONDITION.*

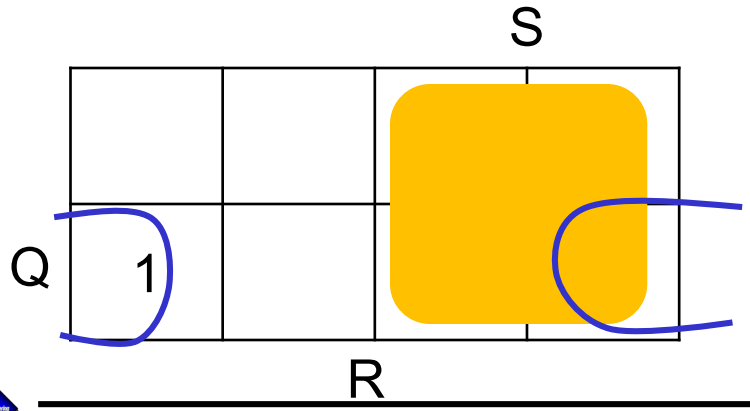
* Contents is courtesy of Vishwani D. Agrawal

SR Latch is Finite State Machine



- Next-state function:
  - Treat illegal states as don't care
  - Minimize using Karnaugh map
- Characteristic equation:

$$Q^* = S + \bar{R}Q$$



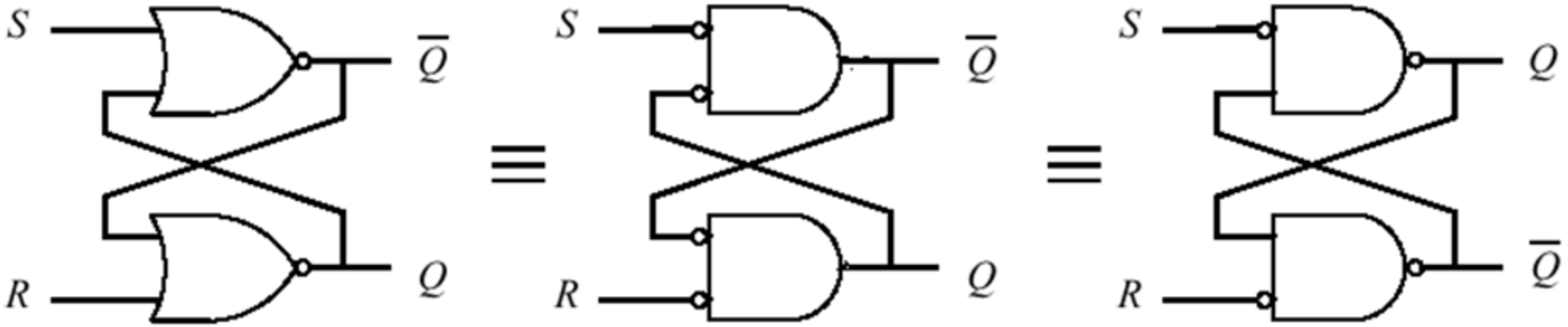
Excitation inputs		Present state	Next state	Name of State
S	R	Q	Q*	
0	0	0	0	Store
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	Illegal	Race cond.
1	1	1	Illegal	

* Contents is courtesy of Vishwani D. Agrawal

What is a Latch?

- A latch is an arrangement of logic gates that maintains a stable output in response to an excitation input
- The output of a latch is determined by the current input and the history of inputs, and thus a combinational logic unit is not powerful enough to capture this behavior
- A flip-flop is a clocked latch that is used to store a single bit of information and serves as a building block for computer memory

NAND Implementation of S-R Latch

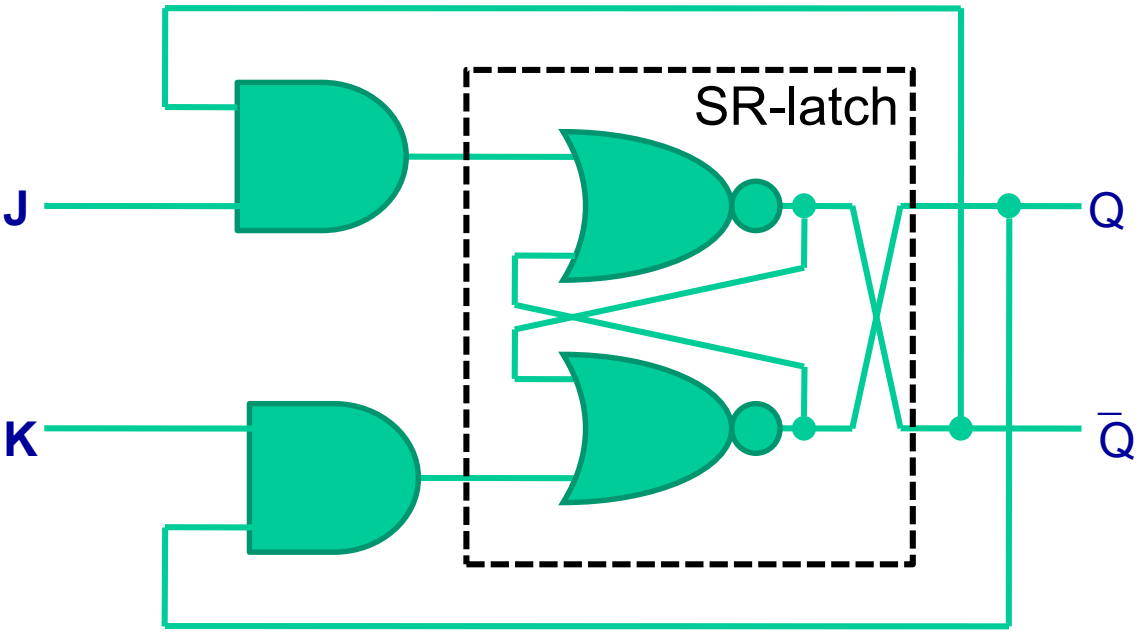


Using De Morgan's theorem

* Slide is courtesy of M. Murdocca and V. Heuring

J-K Latch

- The J-K latch eliminates the disallowed S=R=1 problem of the S-R latch, because Q enables J while Q' disables K, and vice-versa.



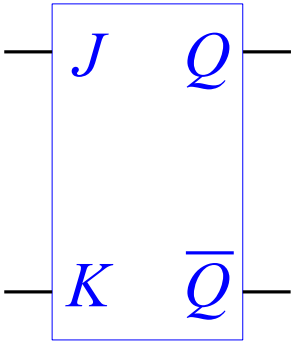
Excitation inputs		Present state	Next state
J	K	Q	Q*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Characteristic Equation:

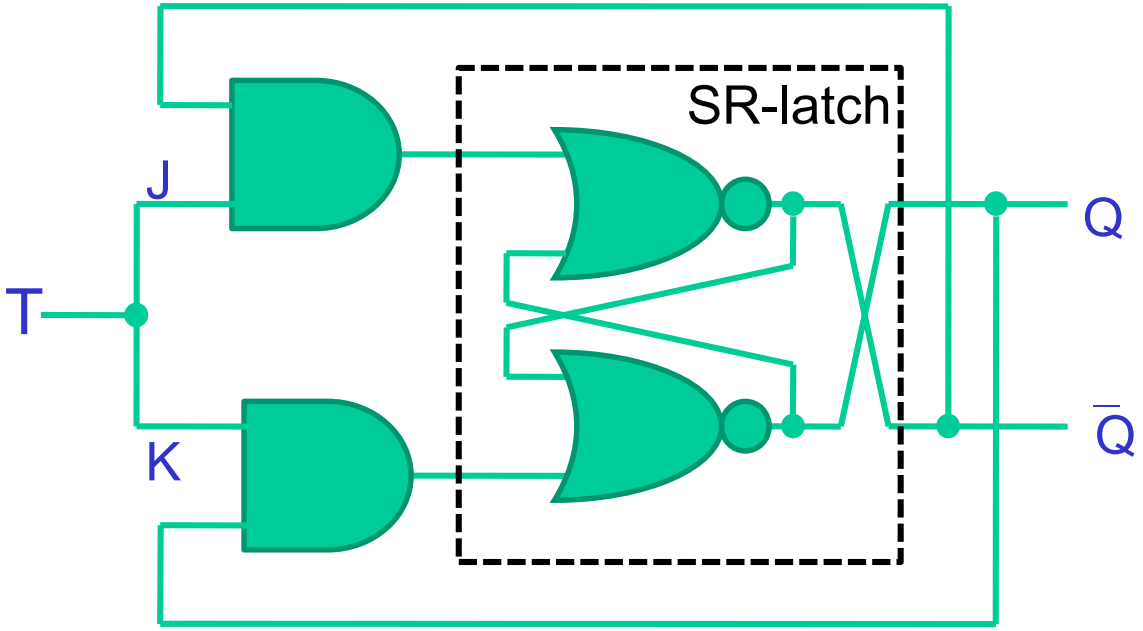
$$Q^* = J\,\overline{Q} + \overline{K}\,Q$$

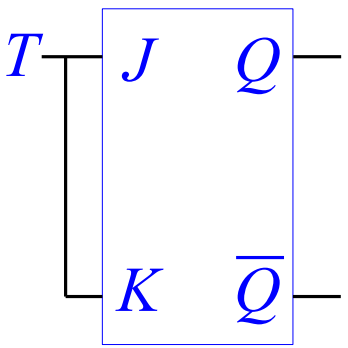
Where Q* = new state,

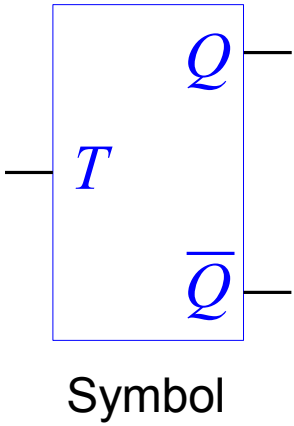
Q = present state



T-Latch (Toggle Latch)







Symbol

Characteristic Equation: $Q^* = T\,\overline{Q} + \overline{T}\,Q$

Where Q^* = next state, Q = present state

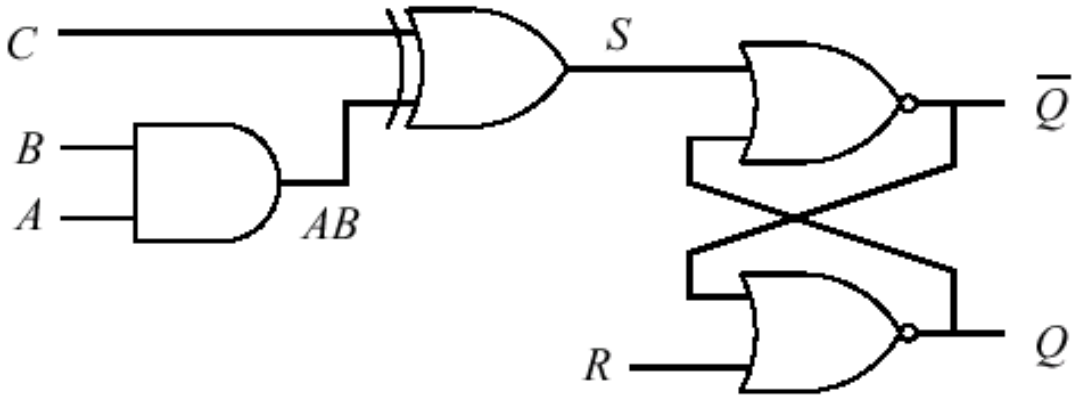
Main idea:

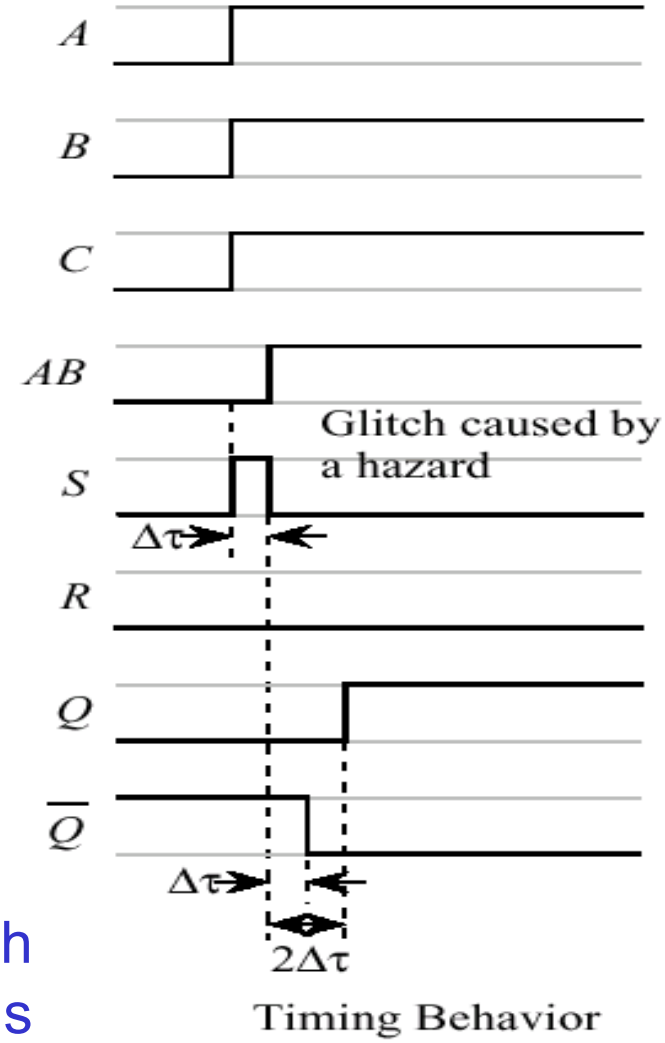
- The presence of a constant 1 at J and K means that the flip-flop will change its state from 0 to 1 or 1 to 0 each time it is clocked by the T (Toggle) input

Excitation input	Present state	Next state
T	Q	Q*
0	0	0
0	1	1
1	0	1
1	1	0

Timing Hazards

- Latch's inputs can originate from combinational logic ➔ transitions occur at any time
- Example: A, B, and C changes from 0 to 1
  - → The signal C reaches S-R flip-flop earlier
  - → The latch state gets a transient change
  - → Such a problem is called a glitch or a hazard





Timing Behavior

- It is desirable to be able to "turn off" the latch so it does not respond to this kind of hazards
- The solution is to use a clock in order to synchronize the changes ➔ becoming a flip-flop

* Some contents are courtesy of M. Murdocca and V. Heuring

The Clock Paces the System

Clock Waveform:

- A clock circuit produces a continuous stream of 1's and 0's

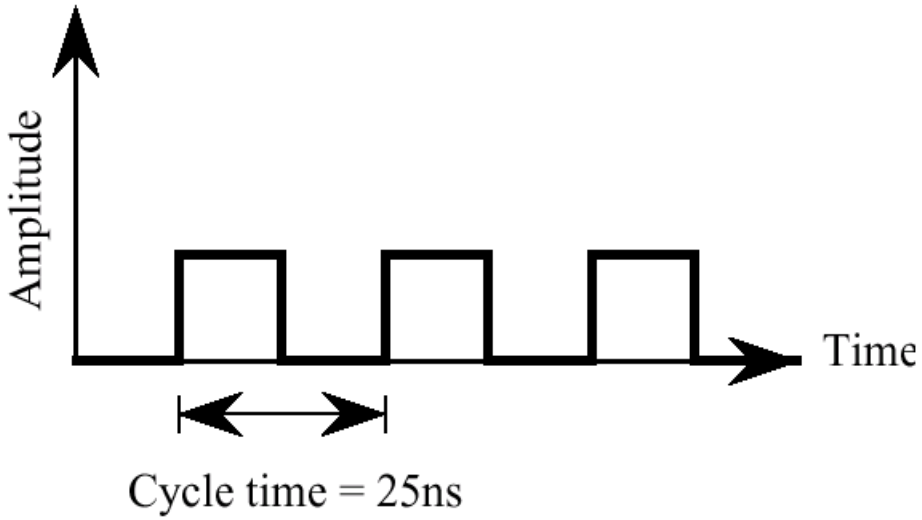
Cycle Time:

- It is the time required for the clock to rise, then fall, then begin to rise again
- In practice wave edges are not perfect (rounded) and can be asymmetric

Clock rate (frequency):

- It is the inverse of the clock cycle time (25ns/cycle → 40 MHz)

- A clock can be used to eliminate potential hazards in latches
- In a positive logic system, the "action" happens when the clock is high, or positive.
- The low part of the clock cycle allows propagation between sub-circuits, so their inputs settle at the correct value when the clock goes high next .



* Some contents are courtesy of M. Murdocca and V. Heuring

Scientific Prefixes

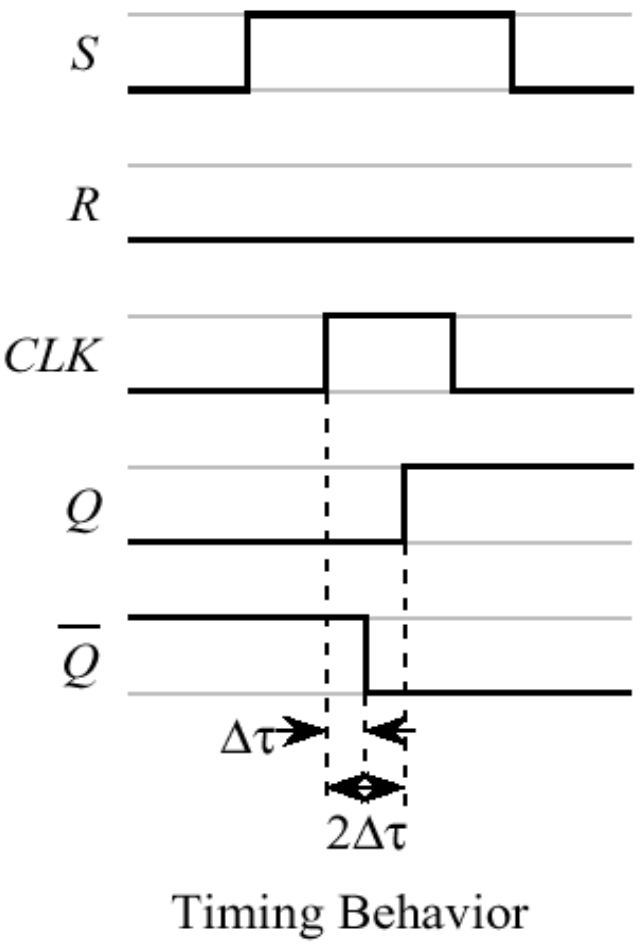
Prefix	Abbrev.	Quantity	Prefix	Abbrev.	Quantity
milli	m	10^{-3}	Kilo	K	10^{3}
micro	μ	10^{-6}	Mega	M	10^{6}
nano	n	10^{-9}	Giga	G	10^{9}
pico	p	10^{-12}	Tera	T	10^{12}
femto	f	10^{-15}	Peta	P	10^{15}
atto	a	10^{-18}	Exa	E	10^{18}

- For computer memory, 1K = 2^{10} = 1024
- For everything else, like clock speeds, 1K = 1000, and likewise for 1M, 1G, ...

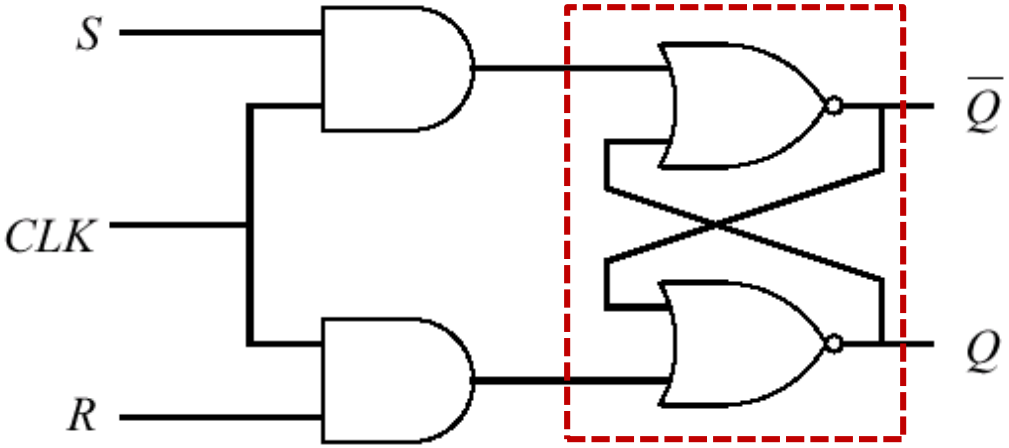
* Slide is courtesy of M. Murdocca and V. Heuring

SR Flip-Flop (Clocked Latch)

- The clock signal, CLK, enables the S and R inputs to the latch
- S and R cannot change the state of the flip-flop until the clock is high
- S and R should settle into stable states while the clock is low
- When the clock makes a transition to 1, the stable S and R values will be used in determining flip-flop's new state



Timing Behavior



SR-latch

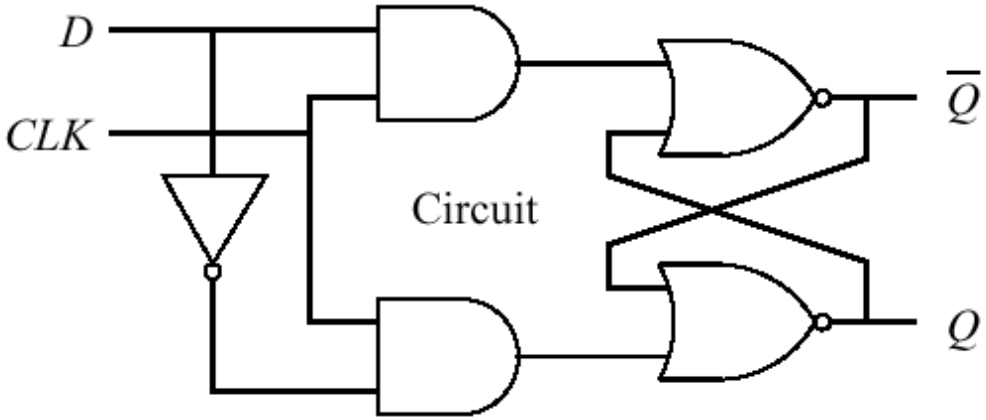
Characteristic equation:

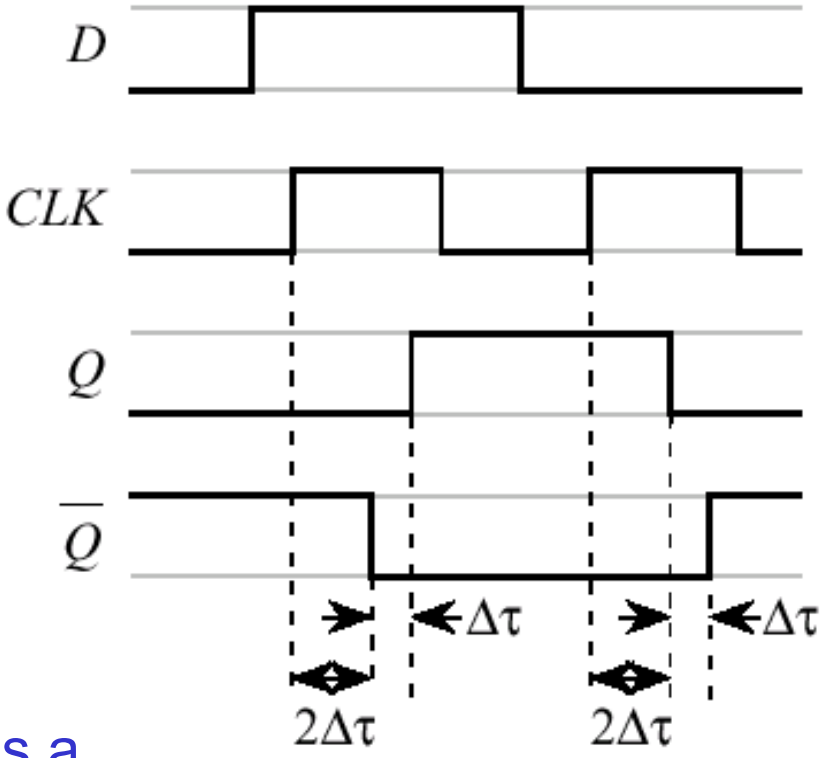
$Q^* = SC + \overline{R}Q + \overline{C}Q$

Some contents are courtesy of Murdocca and Heuring

Clocked Delay Latch (D Flip-Flop)

- S-R flip-flops requires the control of two inputs to change the flip-flop state
- The D (delay) flip-flops are commonly used for circuits with feedback from the output back to the input through some other circuitry



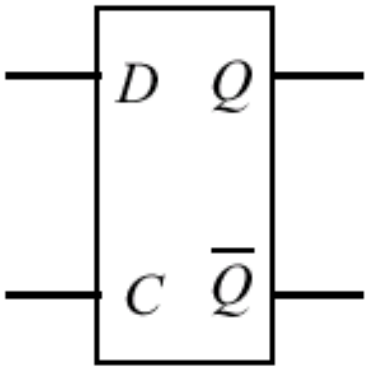


Timing Behavior

Characteristic equation:

$Q^* = DC + \overline{C}Q$

Symbol

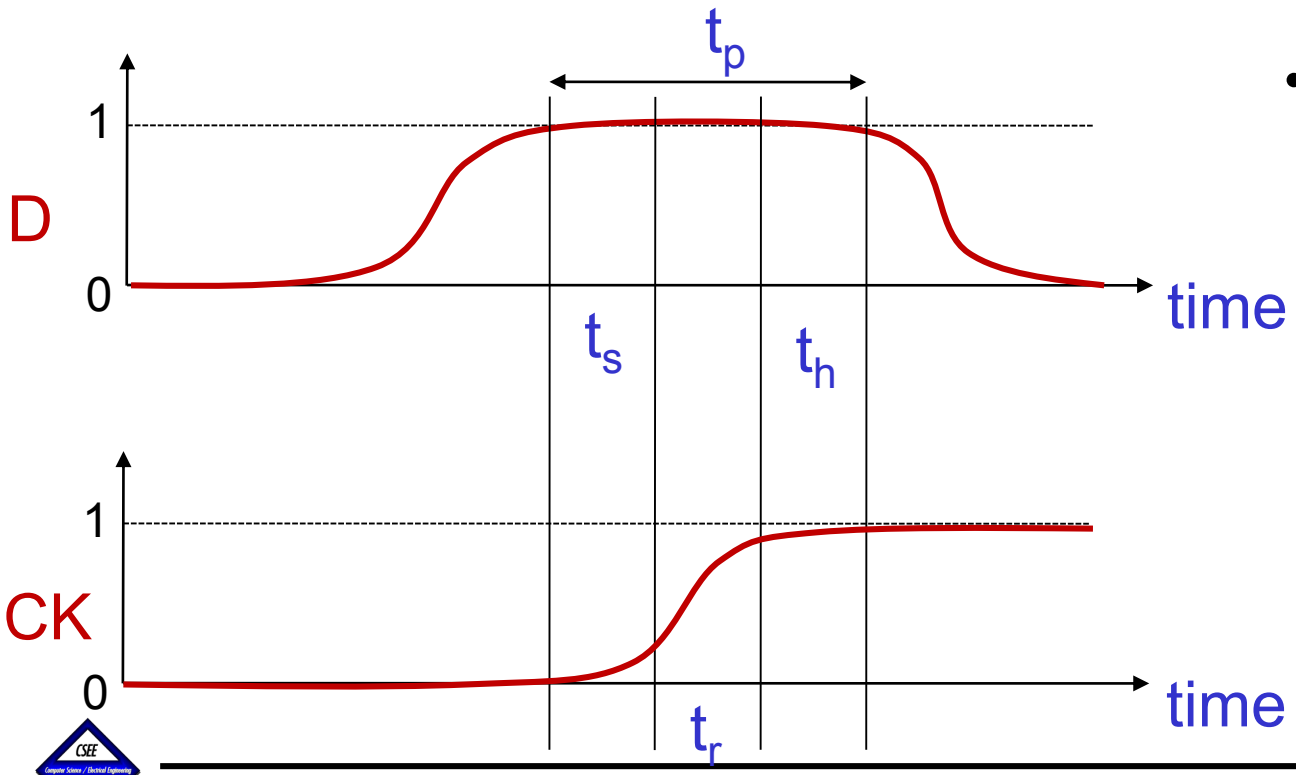


The clocked flip-flop, has a potential problem: If the input changes while the clock is high, the output will also change

* Slide is courtesy of M. Murdocca and V. Heuring

Setup and Hold Times of Latch

- Signals are synchronized with respect to clock (CK)
- ***Setup time "t_s"*** is the interval before the clock transition during which data (D) should be stable (not change). This will avoid any possible race condition
- ***Hold time "t_h"*** is the interval after the clock transition during which data should not change. This will avoid data from latching incorrectly

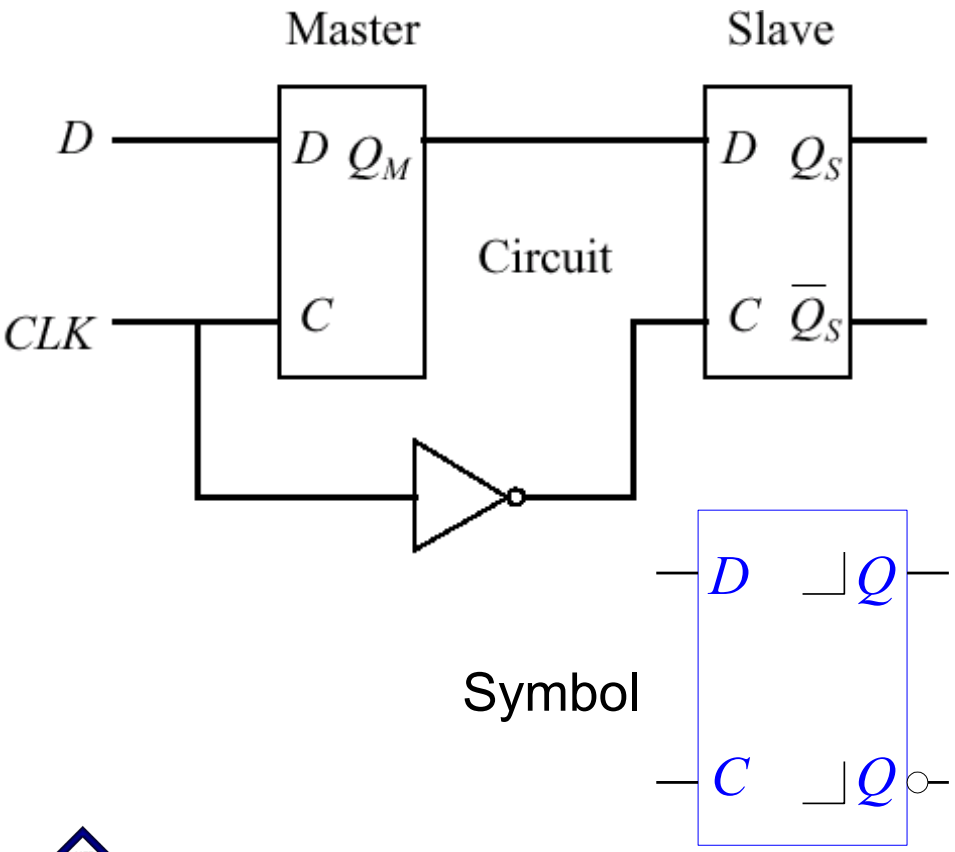


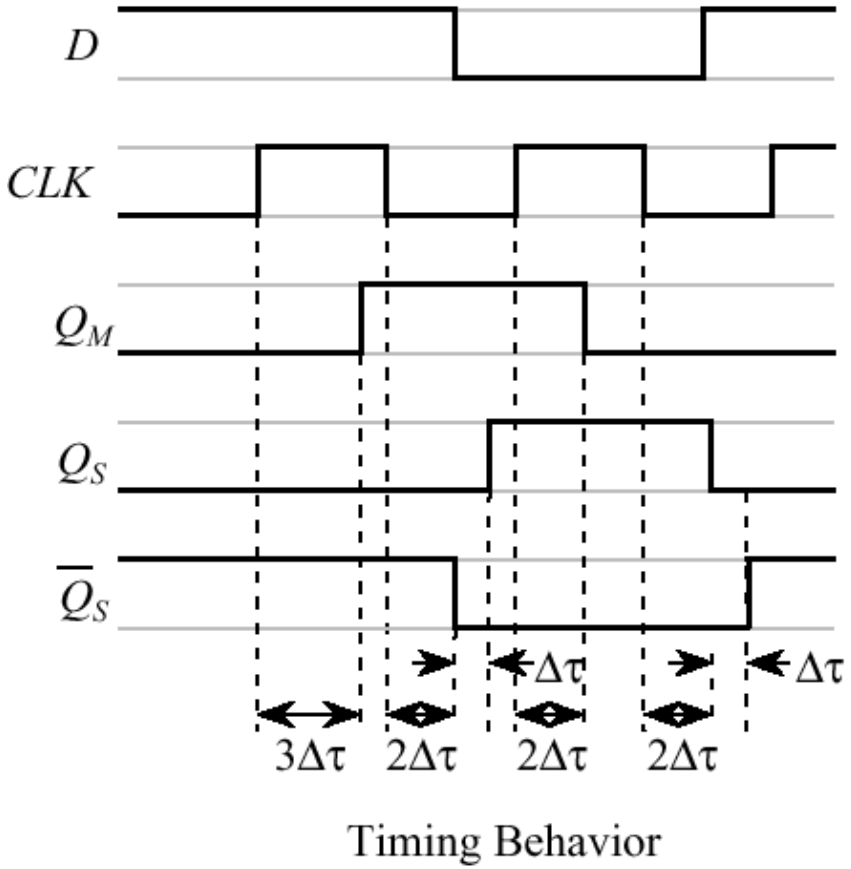
- Operation is ***level-sensitive***:
  - CK = 1 allows data (D) to pass through
  - CK = 0 holds the value of Q, ignores data (D)

* Contents is courtesy of Vishwani D. Agrawal

Master-Slave D Flip-Flop

- The clocked D flip-flop's a potential problem: If D changes while the clock is high, the output will also change ➔ use the Master-Slave flip-flop
- The rising edge of the clock loads new data into the master, while the slave continues to hold previous data. The falling edge of the clock loads the new master data into the slave

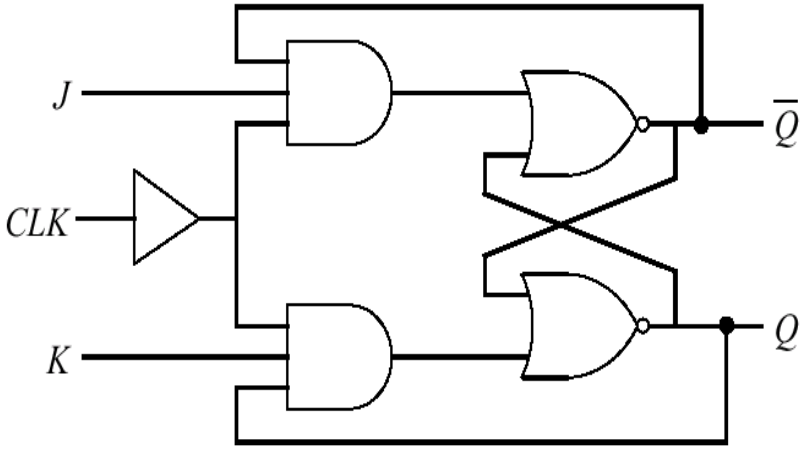




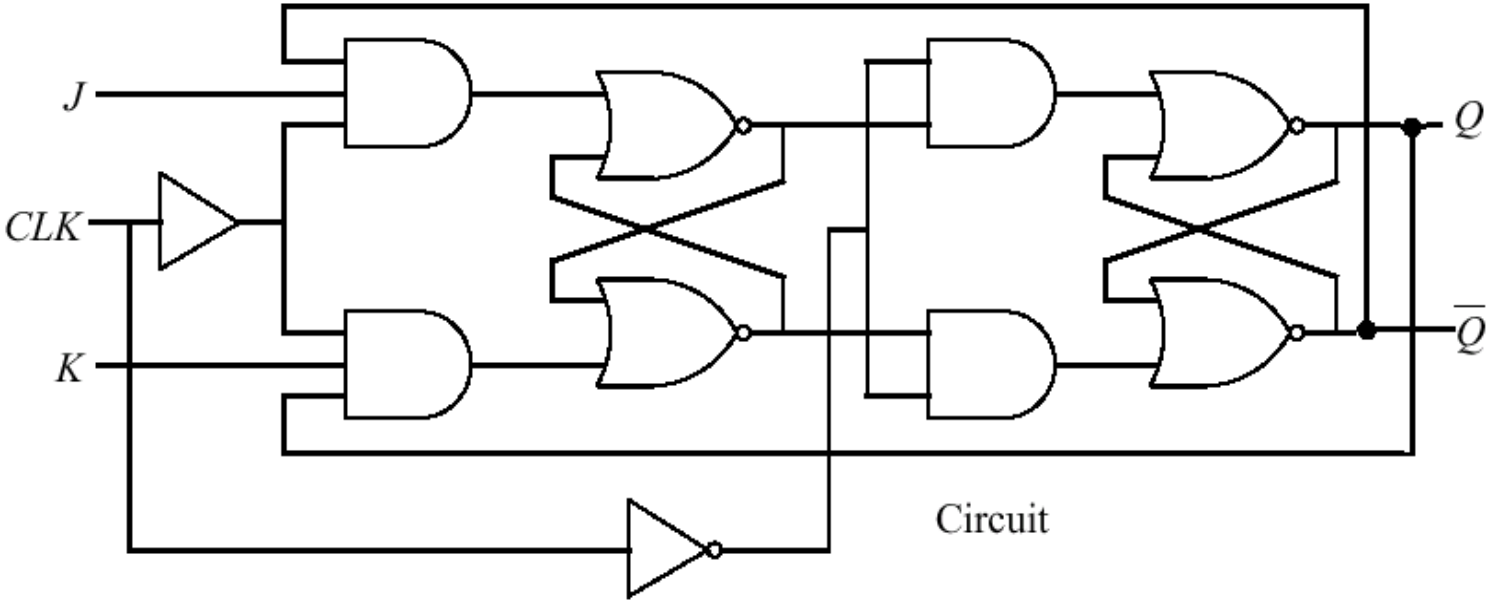
* Slide is courtesy of M. Murdocca and V. Heuring

Master-Slave J-K Flip-Flop

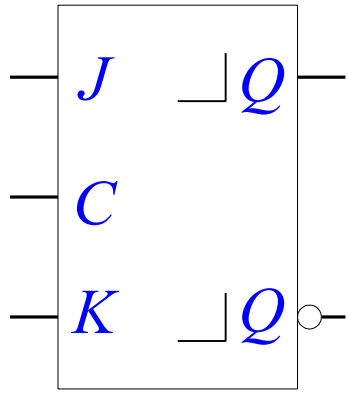
- If J or K goes momentarily to 1 and then back to 0 while the flip-flop is active and in the reset/set state, the flip-flop will toggle erroneously
- The toggle problem is fixed using the master-slave configuration



Circuit



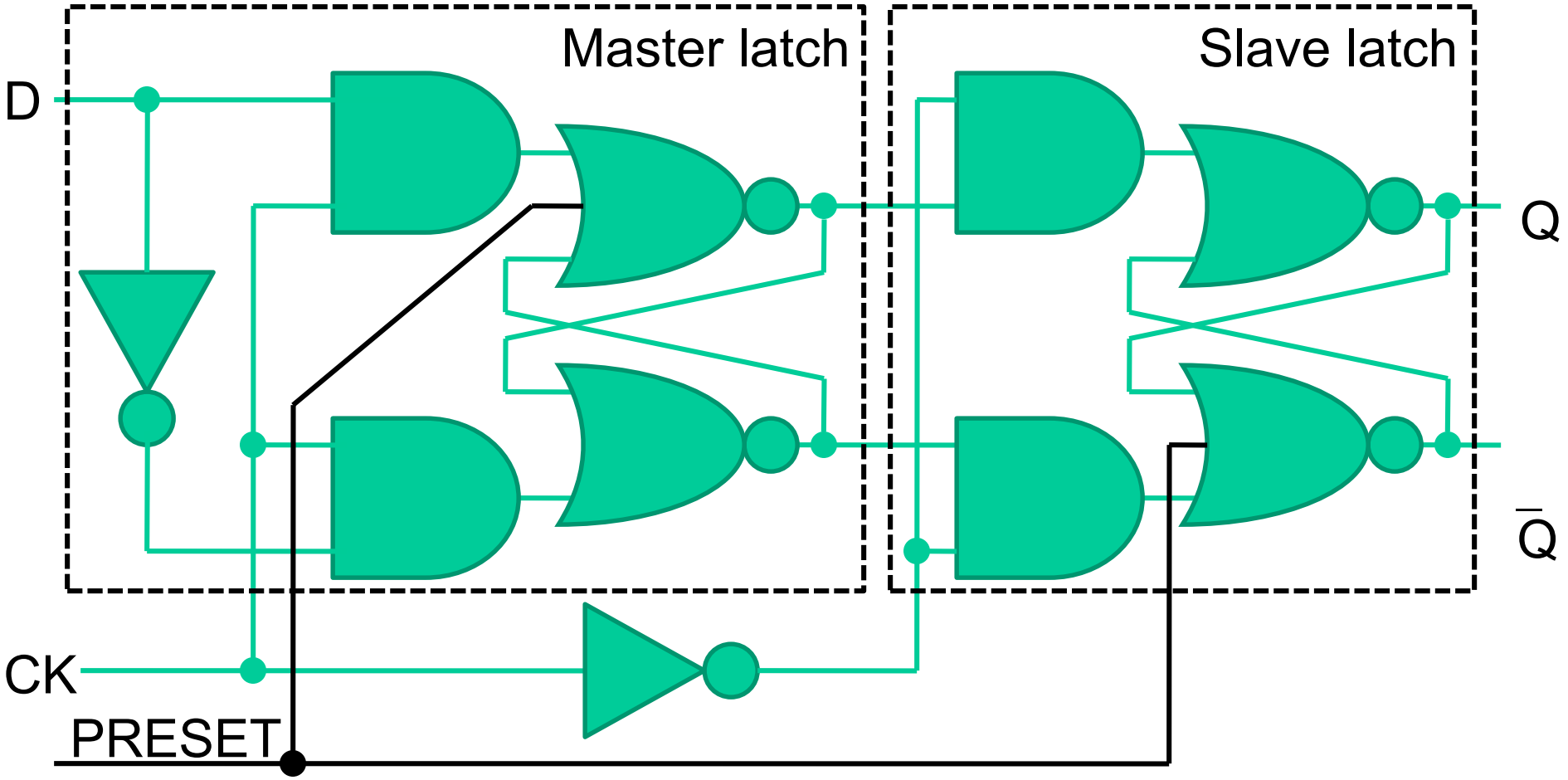
Circuit



Symbol

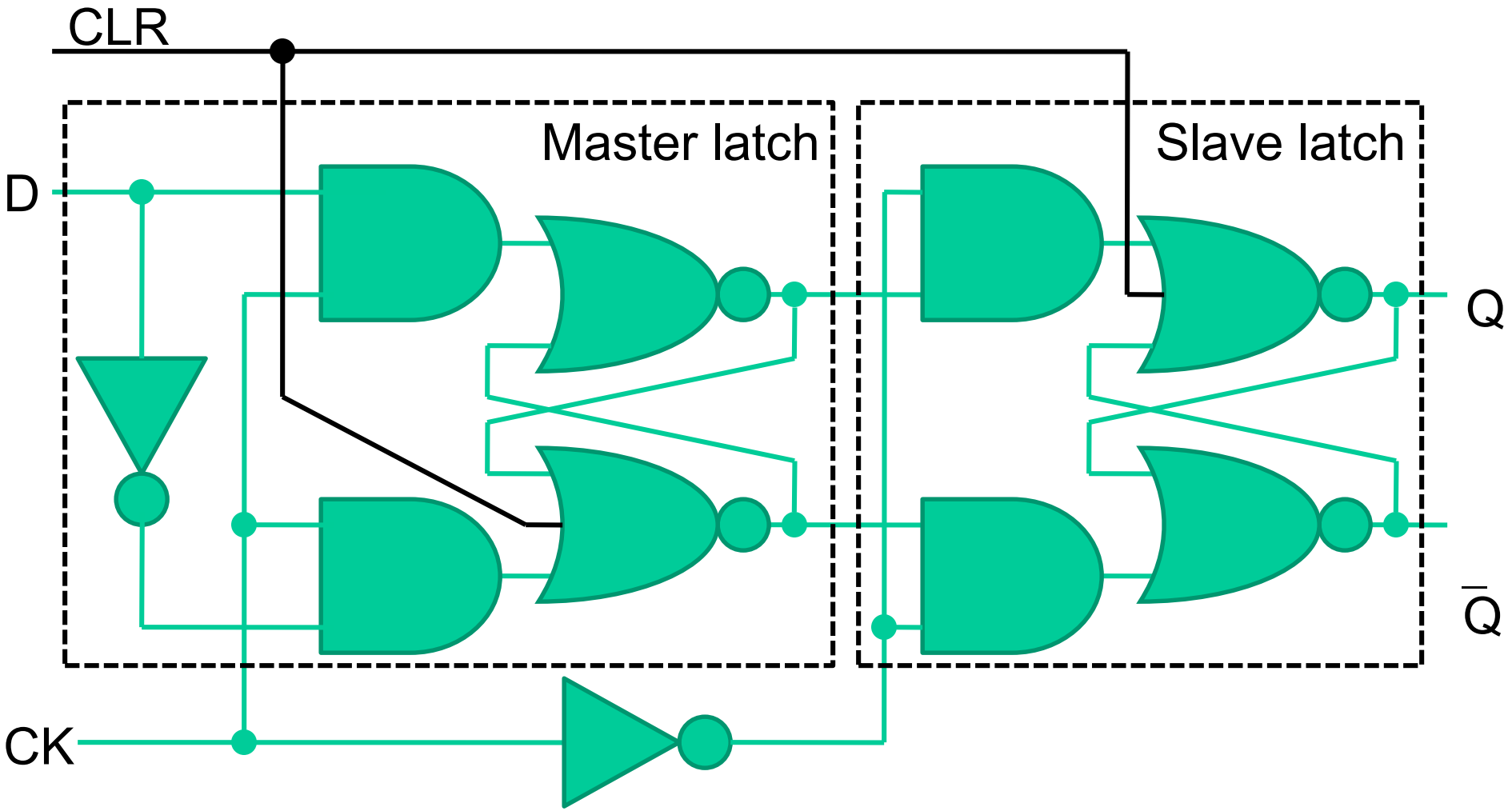
* Slide is courtesy of M. Murdocca and V. Heuring

Master-Slave D Flip-Flop with Preset



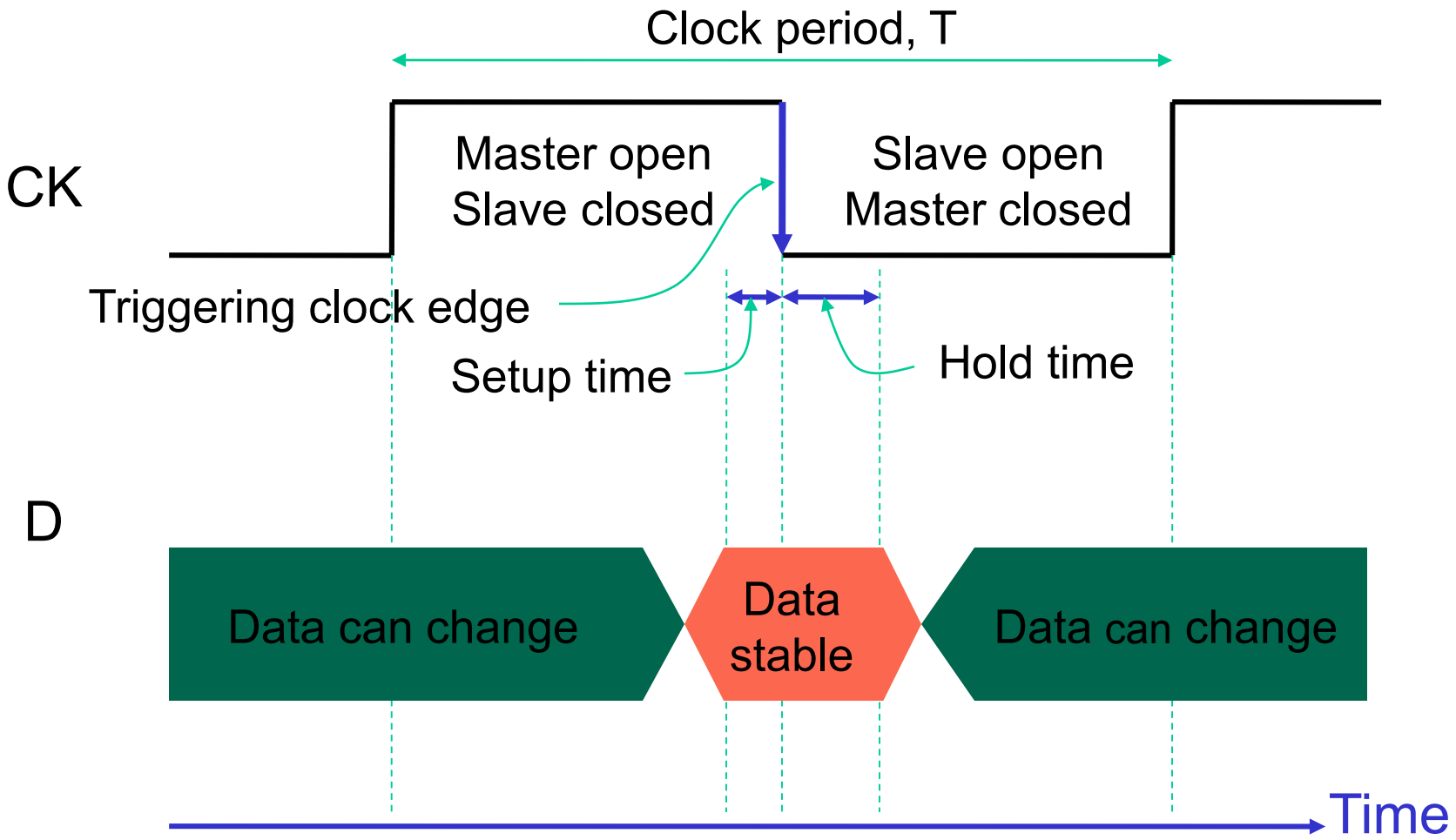
* Contents is courtesy of Vishwani D. Agrawal

Master-Slave D Flip-Flop with Clear



* Contents is courtesy of Vishwani D. Agrawal

Detailed Timing Behavior
Master-Slave D-Flip-Flop

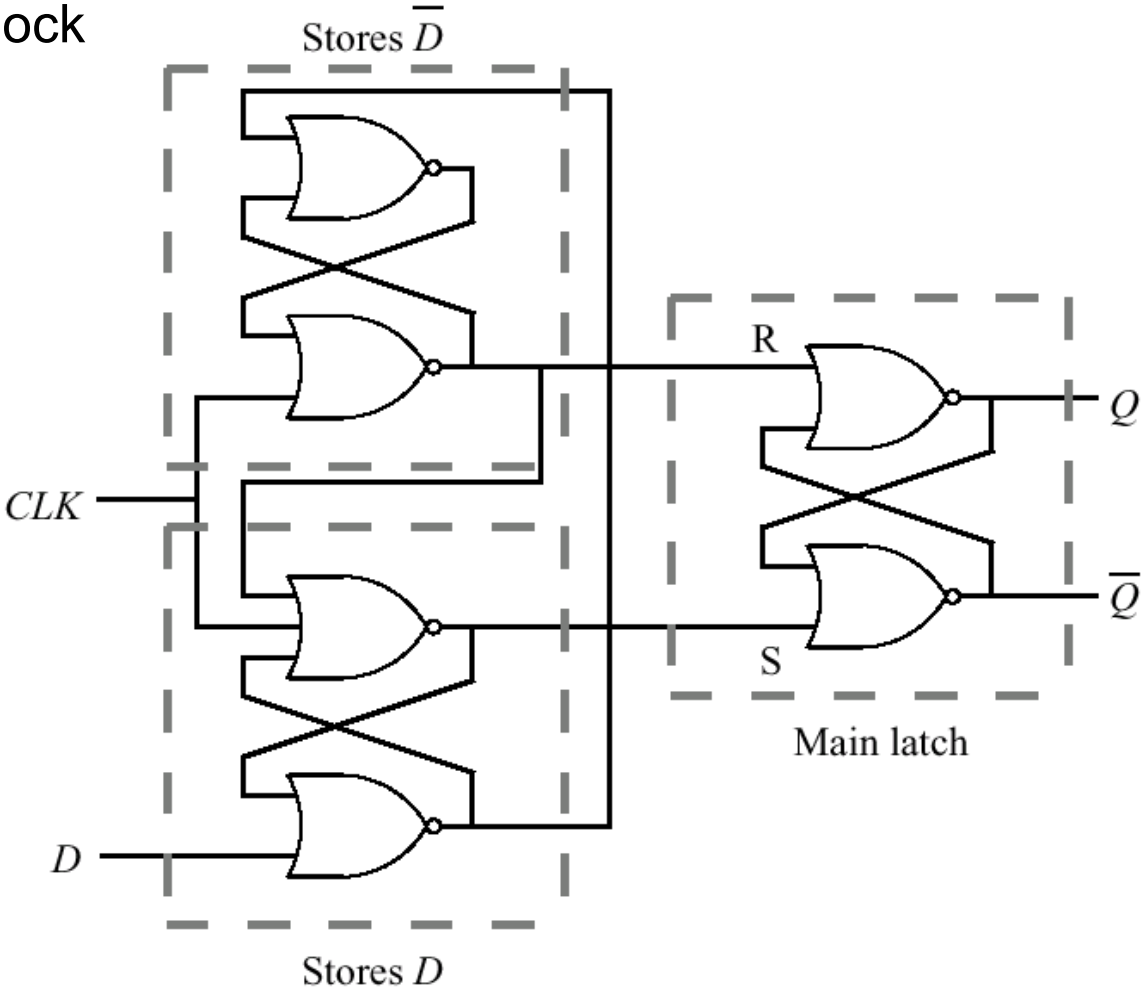


* Contents is courtesy of Vishwani D. Agrawal

Negative Edge-Triggered D Flip-Flop

- The D flip-flop transient changes problem caused by pulse-based clocks can also be solved by making the operation sensitive only to the rising or falling edge of the clock

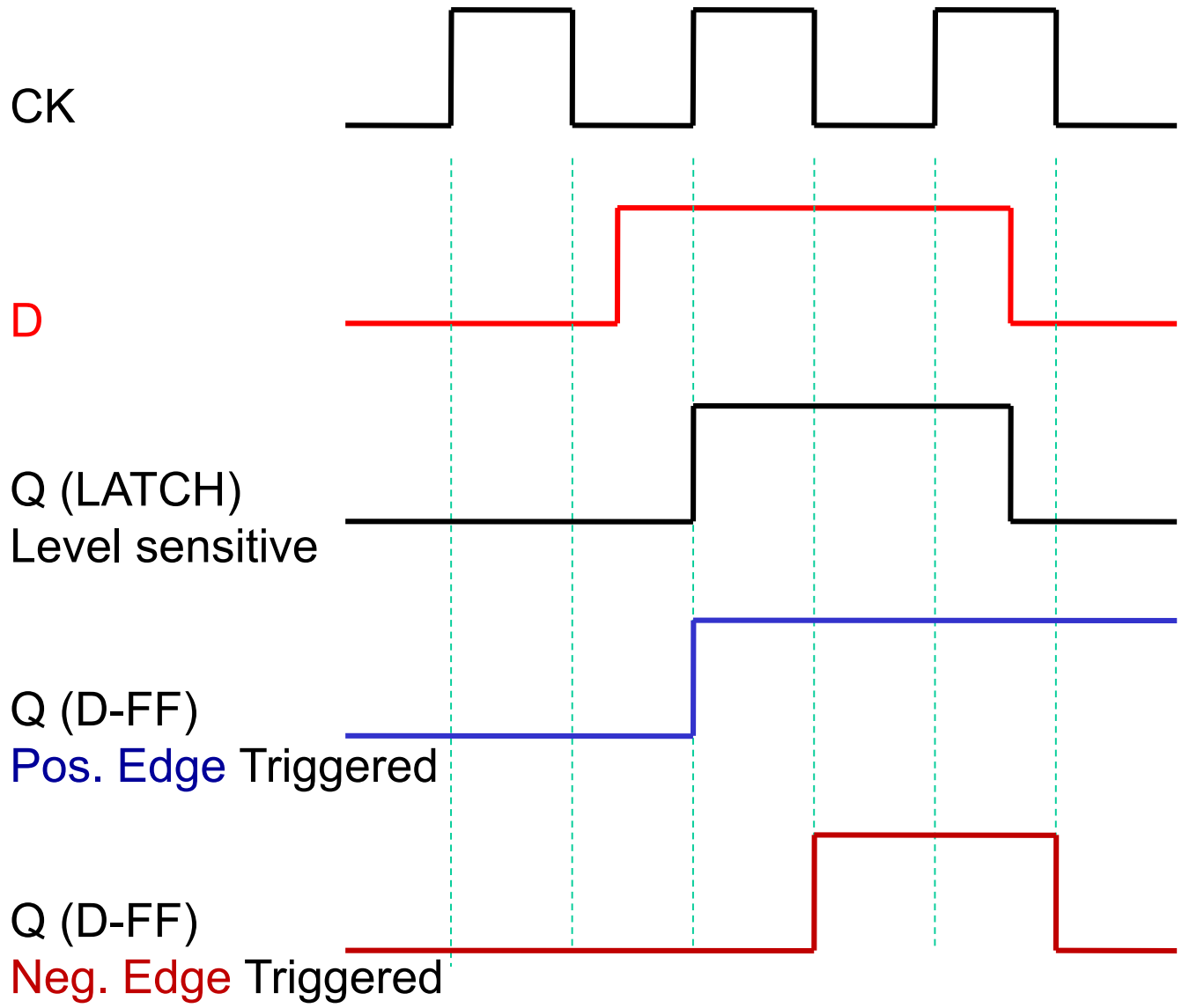
  - When the clock is high, the two input latches output 0, so the main latch remains in its previous state, regardless of changes in D
  - When the clock goes high-to-low, values in the two input latches will affect the state of the main latch.
  - While the clock is low, D cannot affect main latch.

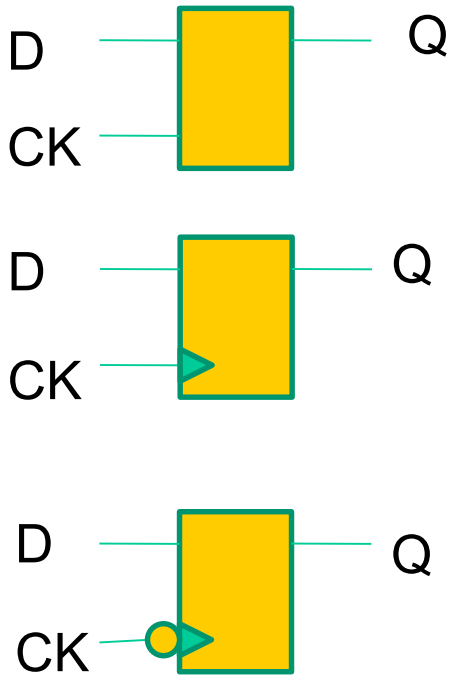


Idea: latch the input and its complement

* Figure is courtesy of M. Murdocca and V. Heuring

Compare: D-Latch and D-Flip-Flops





* Contents is courtesy of Vishwani D. Agrawal

Conclusion

- ***Summary***
  - ➔Introduction to sequential circuits
    (Concept, effect of feedback on combinational circuits, finite state machines, example applications)
  - ➔ Memory devices
    (S-R Latches, D-Latches, J-K and Toggle latches)
  - ➔ Flip-Flops
    (Role of clock, timing behavior diagrams)
  - ➔ Clocked sequential circuits
    (master-slave flip flops, level and edge triggered flip flops
- ***Next Lecture***
  - ➔ Design of sequential circuits
  - ➔ Systematic design of finite state machines

Reading assignment: Sections 6.1 – 6.4 in the textbook