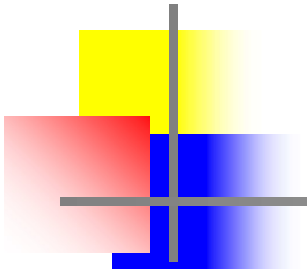
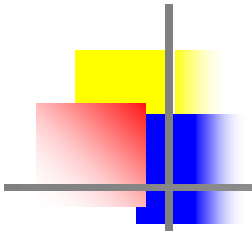


Synchronous Sequential Logic

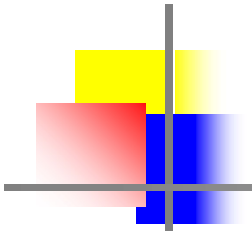
2019. 4. 15.

K-S. Sohn



Contents

- Sequential circuits
- Latches
- Flip-flops
- Analysis of clocked secuential circuits
- State reduction and assignment
- Design of secuential circuits

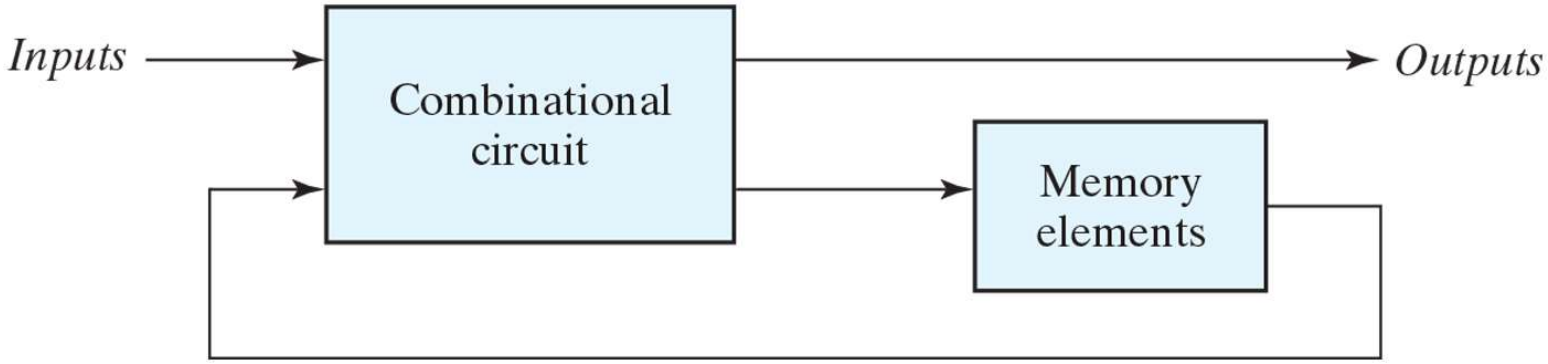


Introduction

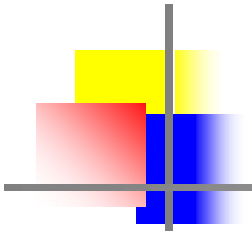
- Combinational circuits
  - neither memory element nor feedback
  - <output> = F(<input>)

Sequential circuits

- Block diagram of secuential circuit

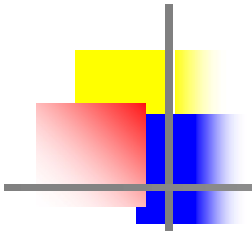


  - feedback path
  - memory element



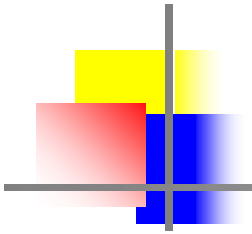
Sequential Circuits

- State
  - the binary information stored in the memory element at any given time defines the state of the sequential circuit at that time.
  - <output> = F(<state>, <input>: <control signal>)
- Syncrounous
  - the transition happens at discrete instants of time
- Asnychronous
  - the transition happens at any instant of time



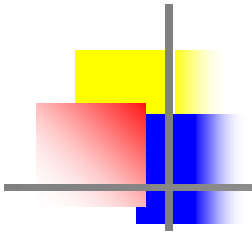
Sequential Circuits

- Asynchronous sequential circuits
  - having only feedback path (or with the "pseudo-memory")
  - <output> = F(<state>, <input>)
  - difficult to avoid instability in design



Synchronous Sequential Circuits

- Clock generator
  - a master-clock generator to generate a periodic train of clock pulses
  - the clock pulses are distributed throughout the system
- clocked sequential circuits
  - most commonly used
  - no instability problems
- clock speed
  - limited by the operational speed of the combinational circuit

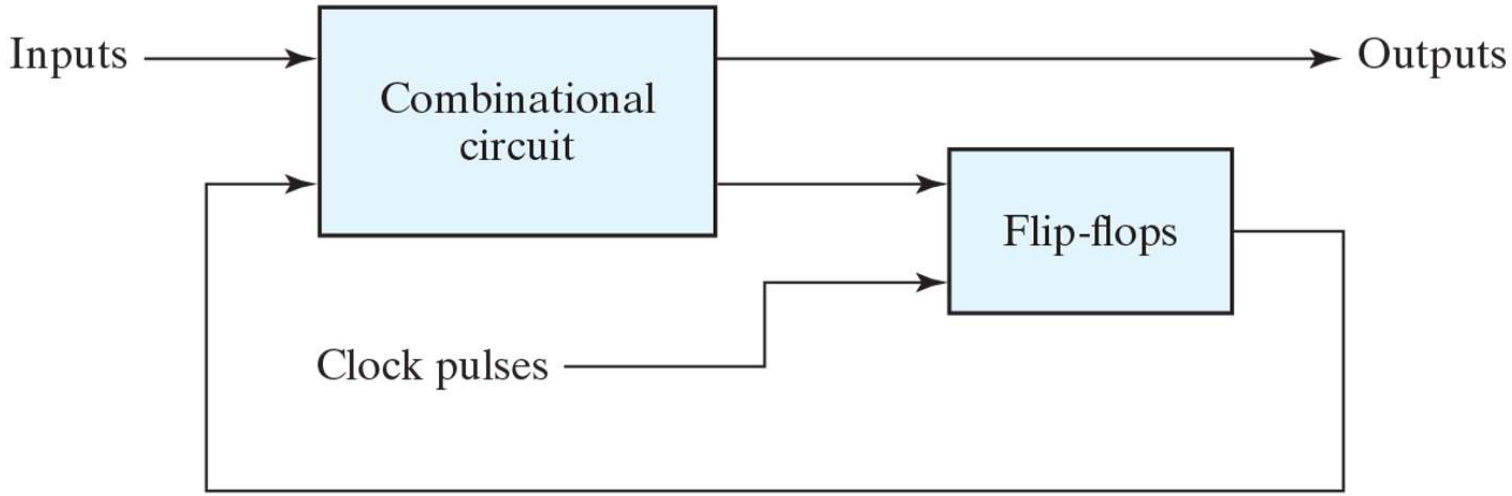


Synchronous Sequential Circuits

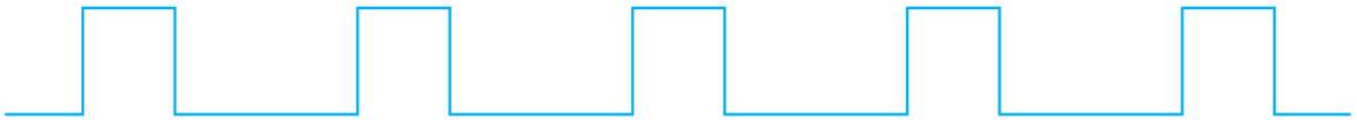
- the memory elements: flip-flops
  - binary cells capable of storing one bit of information
  - two outputs: one for the normal value and one for the
  - complement value
  - maintain a binary state indefinitely until directed by an input signal to switch states

Synchronous Sequential Circuits

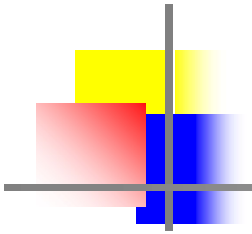
- Synchronous clocked sequential circuit



(a) Block diagram



(b) Timing diagram of clock pulses

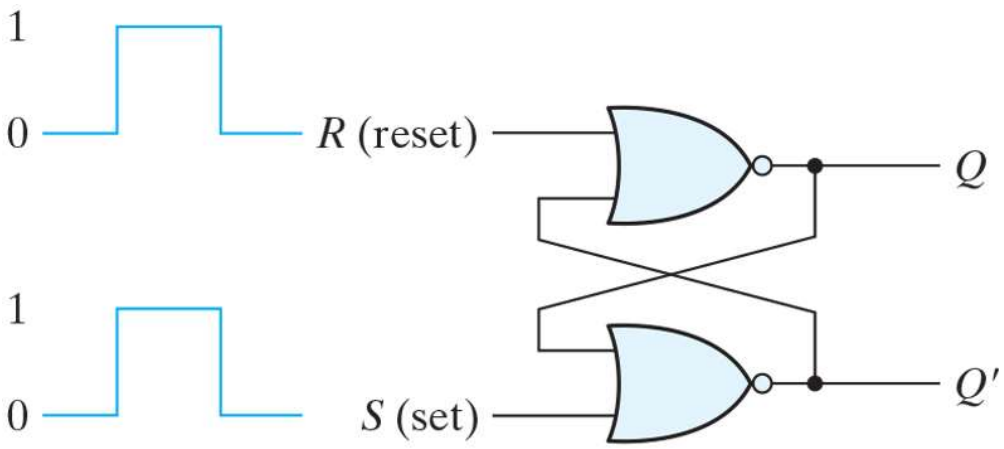


Latches

- asynchronous sequential circuits
  - binary state (0 or 1)
  - Q(t+1) = F(Q(t), <input>)
- Level sensitive circuits
  - the next state is determined by the level of inputs and the current state
  - can not be used for synchronous sequential circuits,
  - but, building blocks of flip-flop

SR Latch

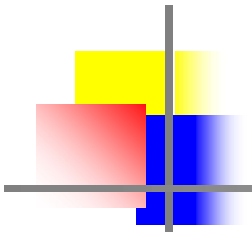
- Tow NOR gates with cross-coupled connection



(a) Logic diagram

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

(b) Function table

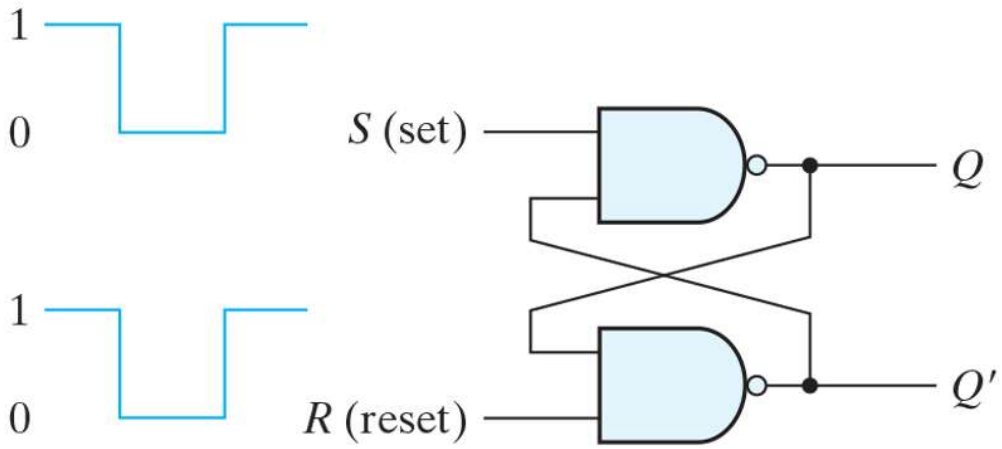


SR Latch

- Most fundamental building block
  - more complicated types can be built upon this SR latch
- Functional characteristics
  - (S,R)= (0,0): no operation (usual position of the level of inputs)
  - (S,R)=(0,1): reset (Q=0, the clear state)
  - (S,R)=(1,0): set (Q=1, the set state)
  - (S,R)=(1,1): indeterminate state (Q=Q'=0, out of SR rules)

SR Latch

- NAND implementation
  - complement of the NOR version
  - S'R' latch



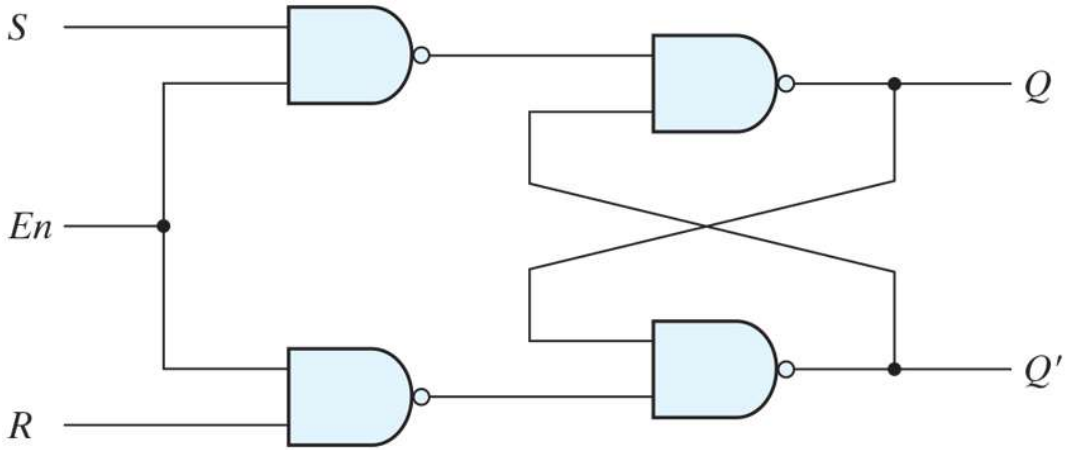
(a) Logic diagram

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

(b) Function table

SR Latch

- SR latch with control input
  - En = 0, no change
  - En = 1, enable input



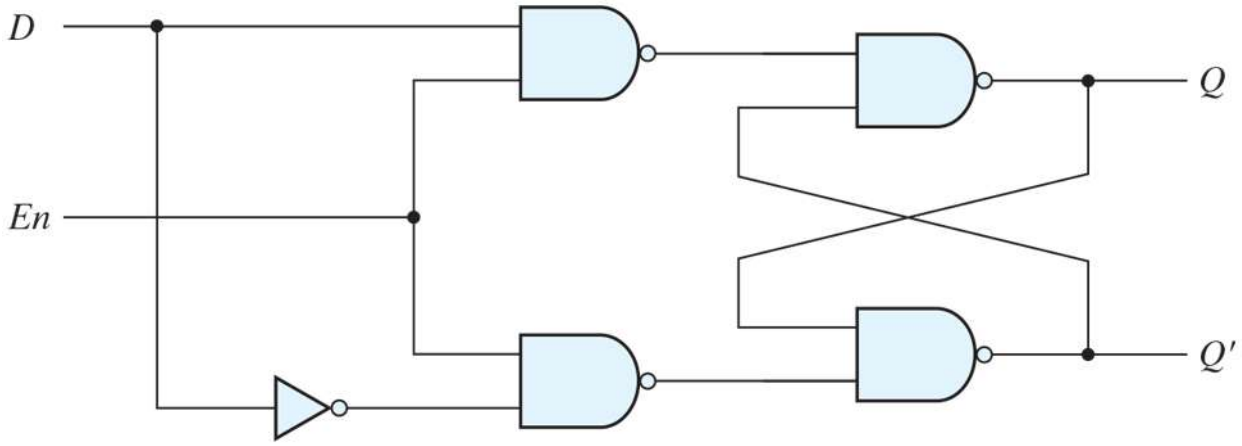
(a) Logic diagram

En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table

D Latch

- Design objectives
  - eliminate the undesirable conditions of the indeterminate state in the SR latch by using an inverter and tied inputs
- 'D' comes from:
  - Q=<input **d**ata>, transparent latch



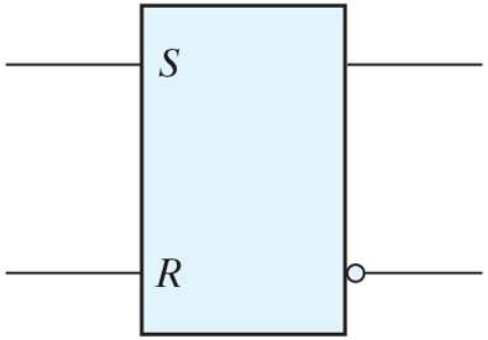
(a) Logic diagram

En	*D*	Next state of *Q*
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state

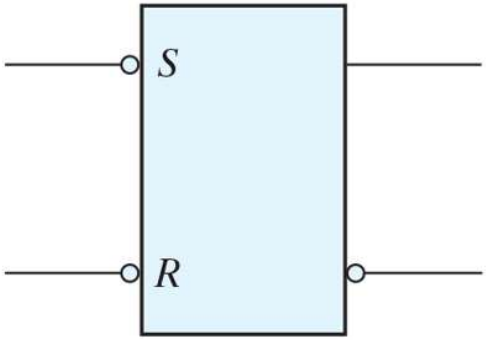
(b) Function table

Graphic Symbols

- SR latches

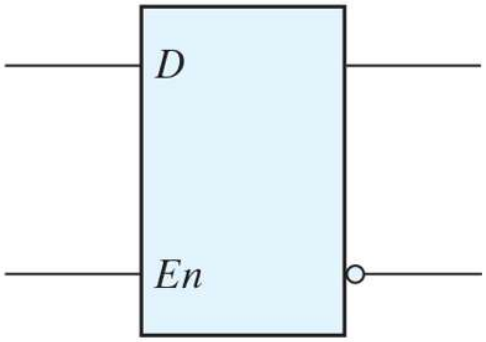


SR



$\overline{S}\overline{R}$

- D latch



D

Flip-Flops

- Trigger
  - the momentary change of control inputs
  - the state of a latch or flip-flop is switched by a change of the control input
- Level triggerd: latch

(a) Response to positive level

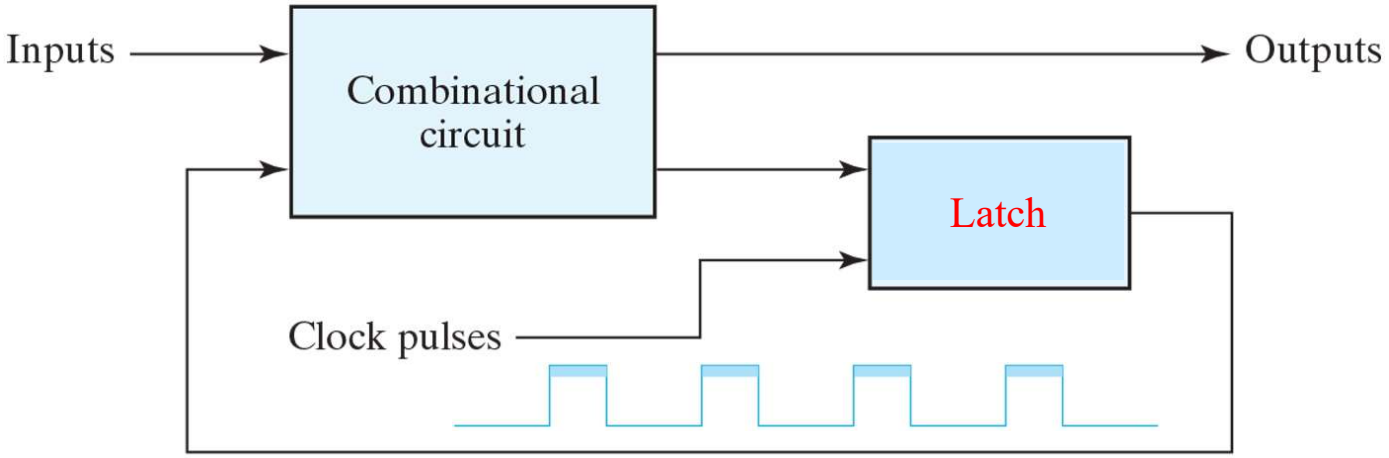
- Edge triggered: F-F

(b) Positive-edge response

(c) Negative-edge response

Flip-Flop

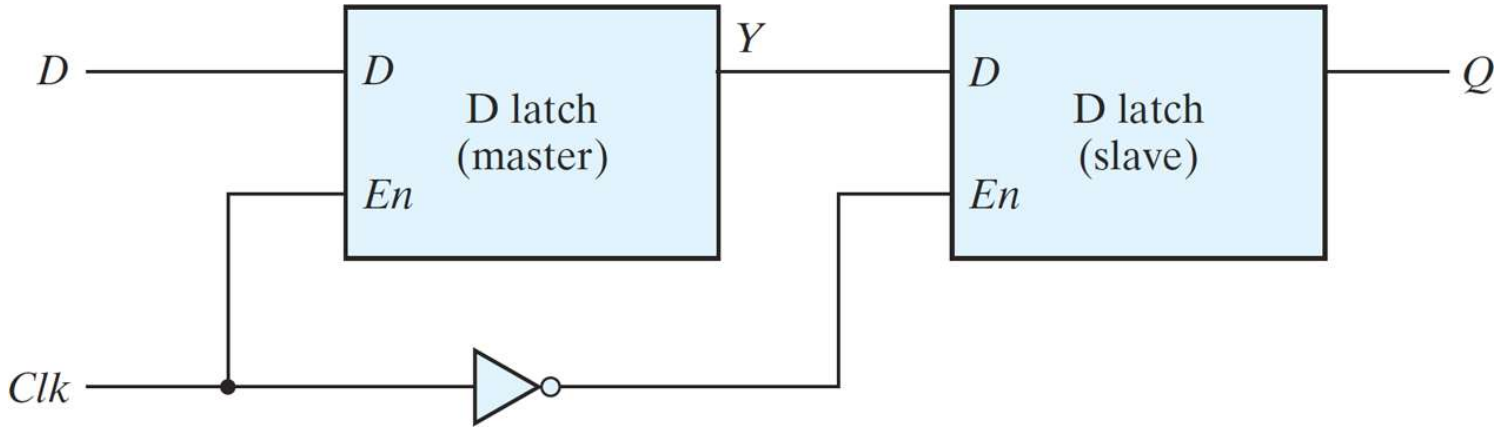
- Why F-F as the memory element in the S.S.L.?
  - If a latch used:



  - the feedback path may cause instability problem
- Edge-triggered F-Fs
  - the state transition happens only at the edge
  - eliminate the multiple-transition problem

Edge-triggered D Flip-Flop

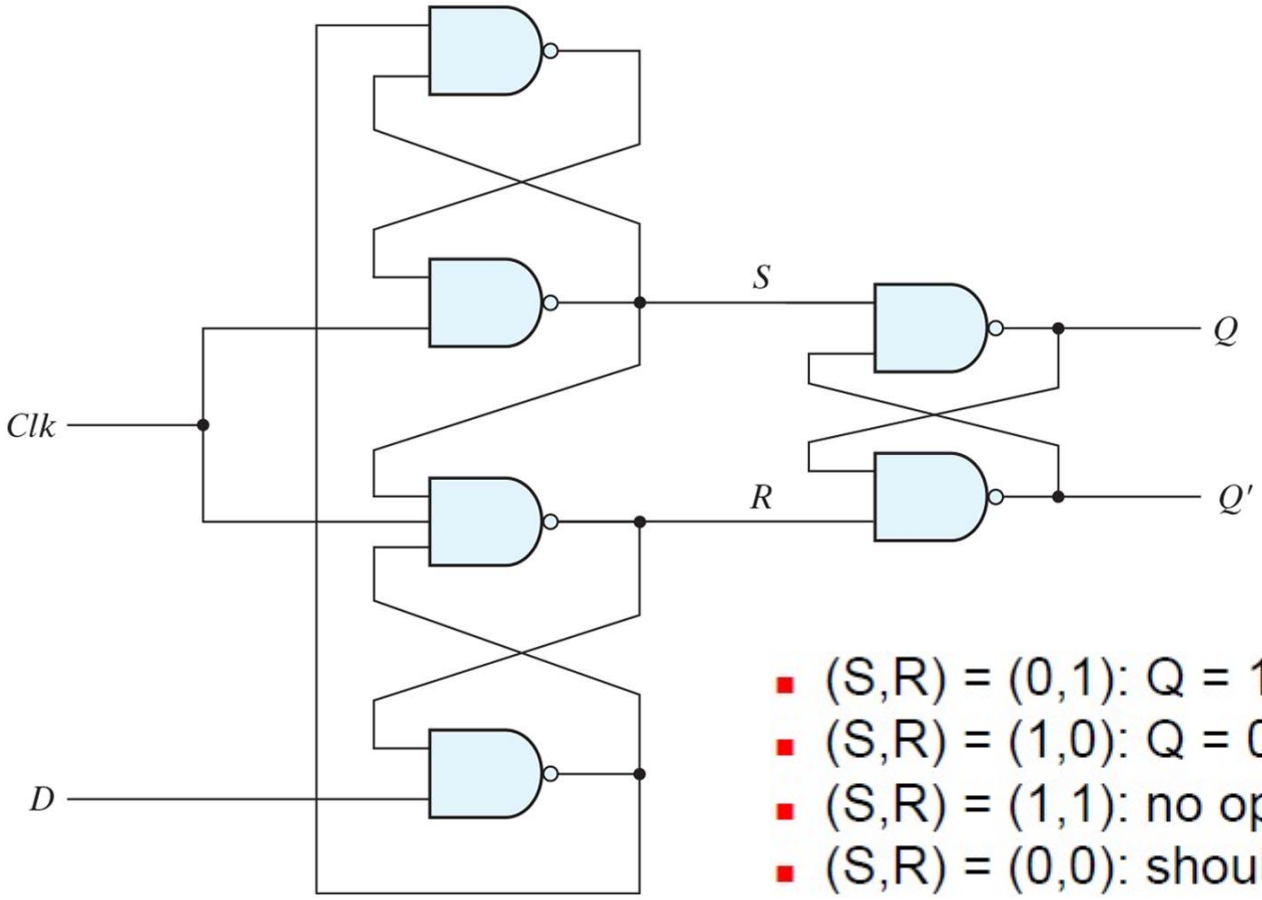
- Master-slave D flip-flop
  - two separate D latches and an inverter
  - master latch (triggered during clk's positive level)
  - slave latch(triggered during clk's negative level)



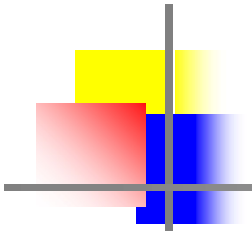
  - triggered at negative or positive edge?
  - how to reverse the direction of triggering edge?

Edge-triggered D Flip-Flop

- D type positive-edge triggered F-F



- (S,R) = (0,1): Q = 1
- (S,R) = (1,0): Q = 0
- (S,R) = (1,1): no operation
- (S,R) = (0,0): should be avoided

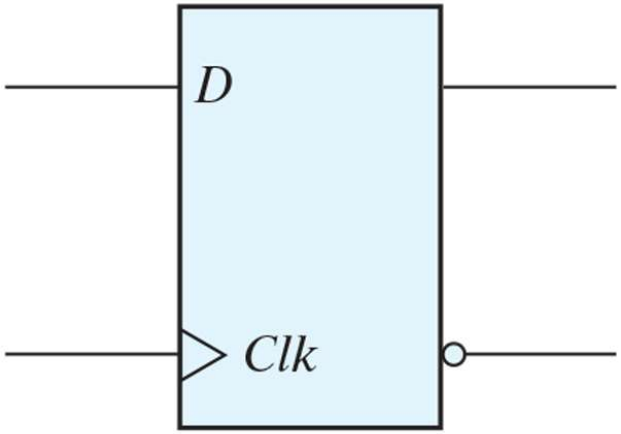


Edge-triggered D Flip-Flop

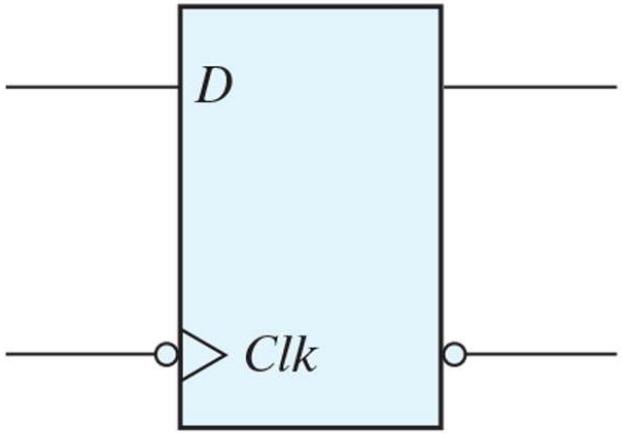
- In sum.
  - CP=0: (S,R) = (1,1), no state change
  - CP=↑: state change once
  - CP=1: state holds
  - eliminate the feedback problems in sequential circuits

Graphic Sumbols of D F-F

- The edge-triggered D F-F
  - the most economical and efficient
  - positive-edge and negative-edge



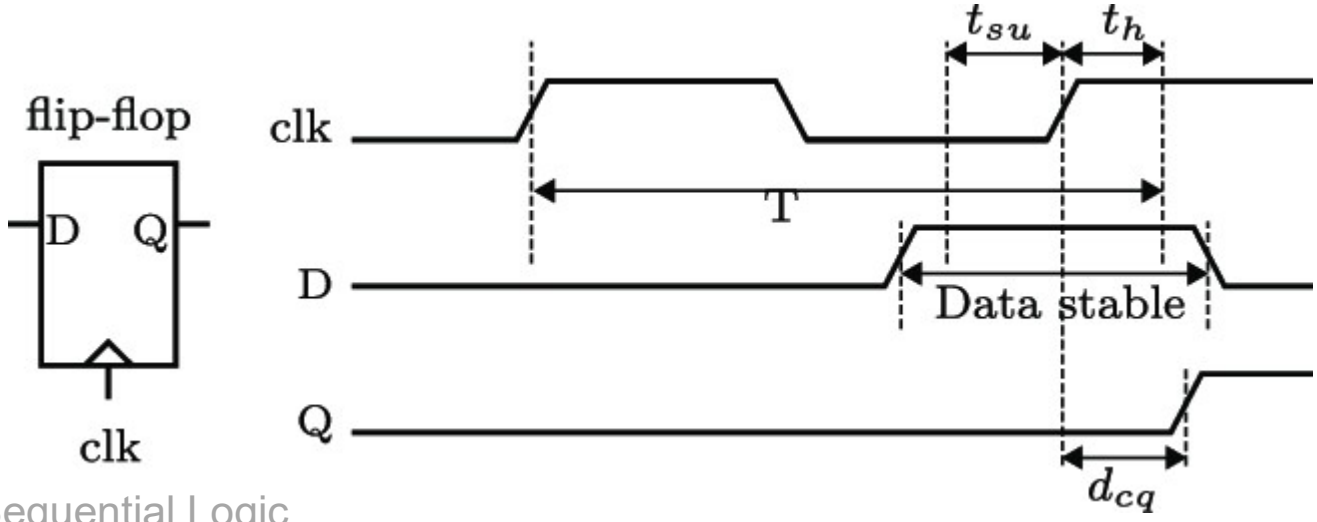
(a) Positive-edge



(b) Negative-edge

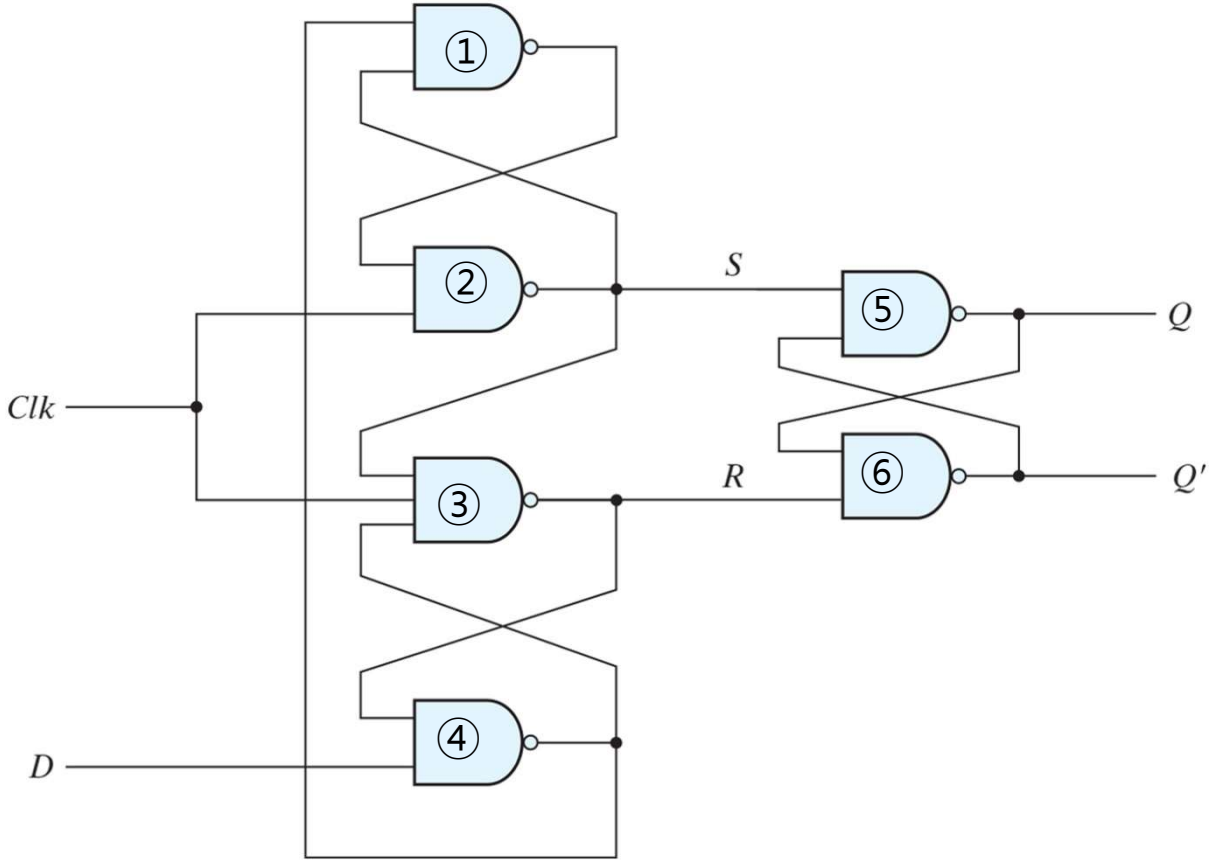
Specifications of Flip-Flop

- Setup time
  - A minimum time for which the D input must be maintained at a constant value prior to the occurrence of the clock transition
- Hold time
  - A minimum time for which the D input must not change after the application of the positive transition of the clock



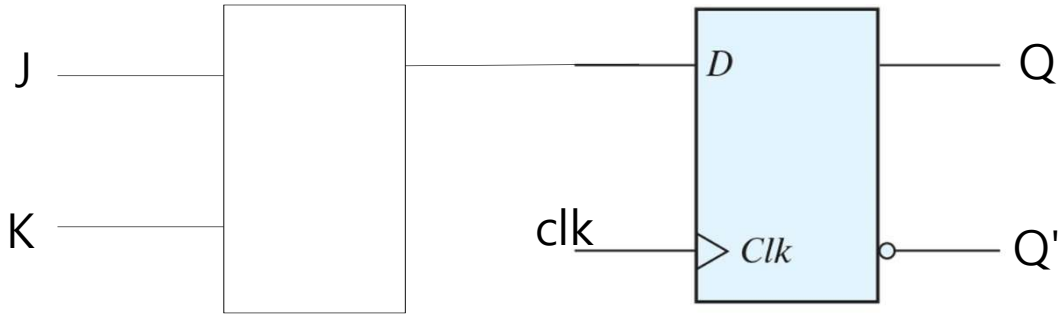
Specification of Flip-Flop

- Time contribution analysis
  - setup time > td1 + td4
  - hold time > td3



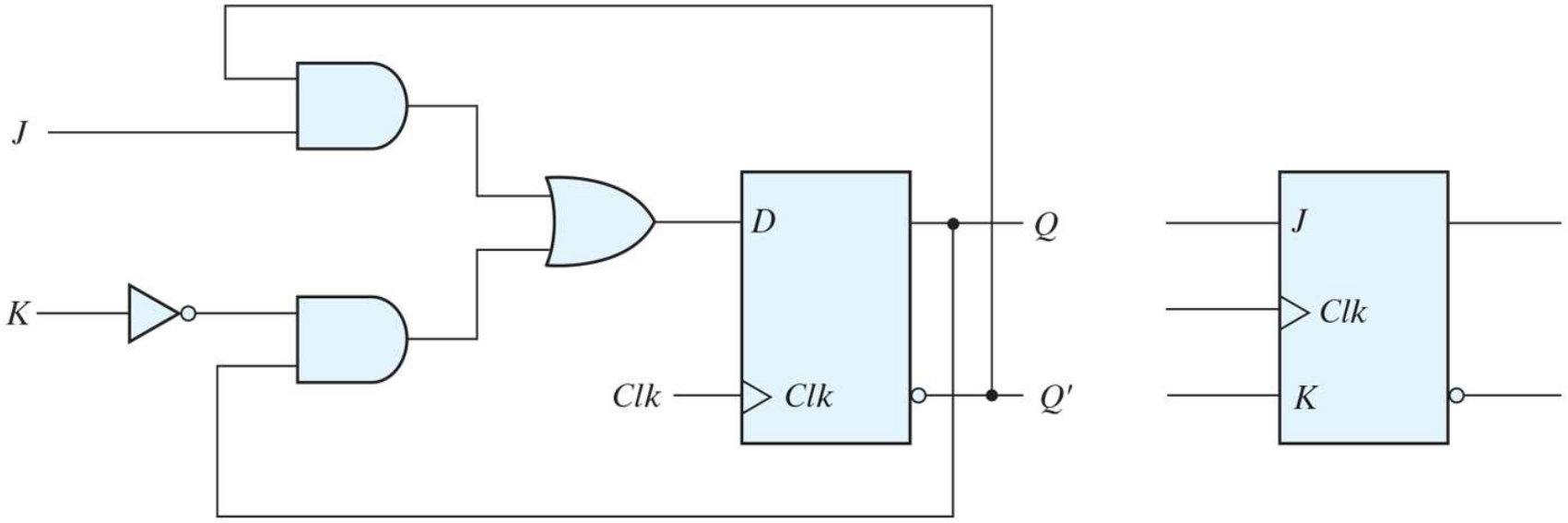
JK Flip-Flop

- Operational requirements
  - Set: J=1, K=0, Q(t+1)=1
  - Reset: J=0, K=1, Q(t+1)=0
  - Complement: J=1, K=1, Q(t+1)=Q'(t)
  - No change: J = K = 0, Q(t+1)=Q(t)
- Constraints
  - Use D F-F and some logic gates including inverters



JK Flip-Flop

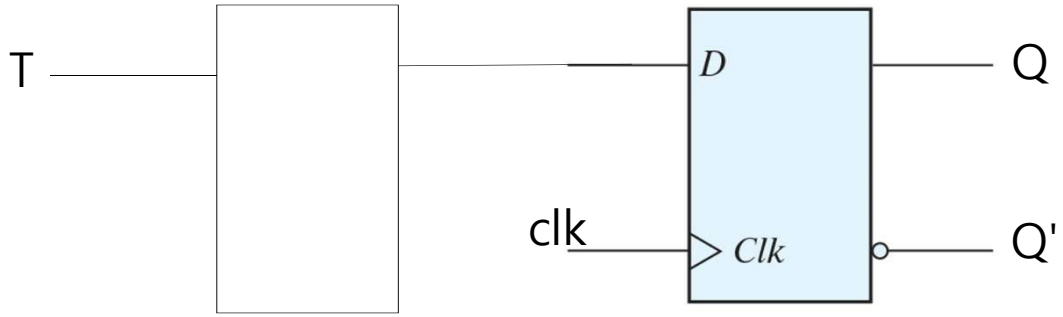
- Front logic
  - D = JQ' + K'Q
  - How?
- Logic diagram and graphical symbol



(a) Circuit diagram (b) Graphic symbol

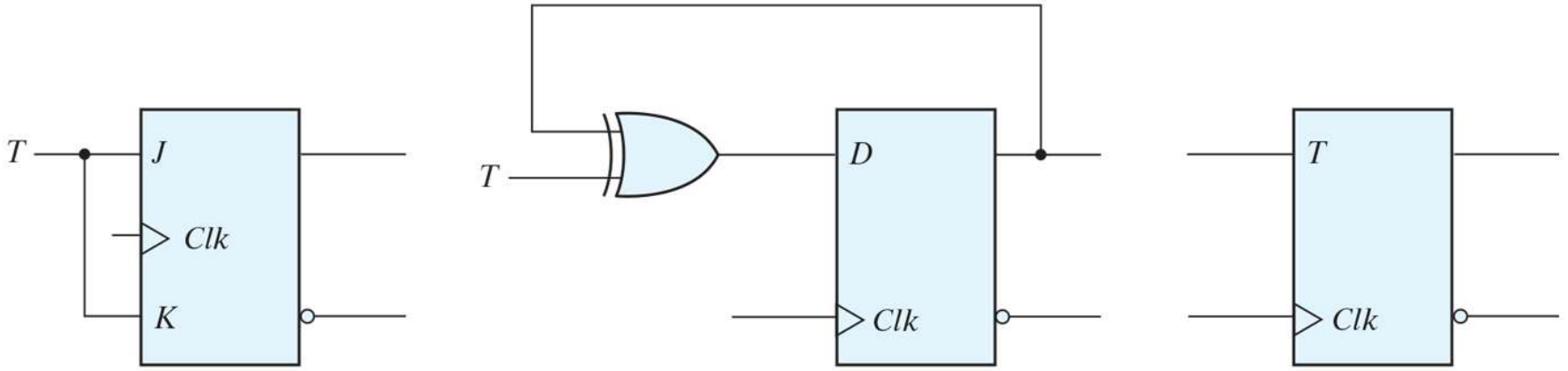
T Flip-Flop

- Functional requirements
  - toggle output with a T input
  - T=0, Q(t+1)=Q(t)
  - T=1, Q(t+1)=Q'(t)
- Constraints
  - Use D F-F and some logic gates



T Flip-Flop

- Front logic
  - D=TQ'+T'Q
  - Why?
- Logic diagram and graphic symbol



(a) From *JK* flip-flop (b) From *D* flip-flop (c) Graphic symbol

Characteristics Tables

- t
  - a period of the clock pulse
  - (t+1) means the next clock pulse period
- Q(t)
  - present state
- Q(t+1)
  - next state

JK Flip-Flop

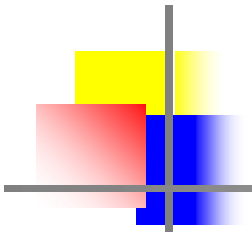
J	K	Q(t + 1)	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

D Flip-Flop

D	Q(t + 1)	
0	0	Reset
1	1	Set

T Flip-Flop

T	Q(t + 1)	
0	$Q(t)$	No change
1	$Q'(t)$	Complement

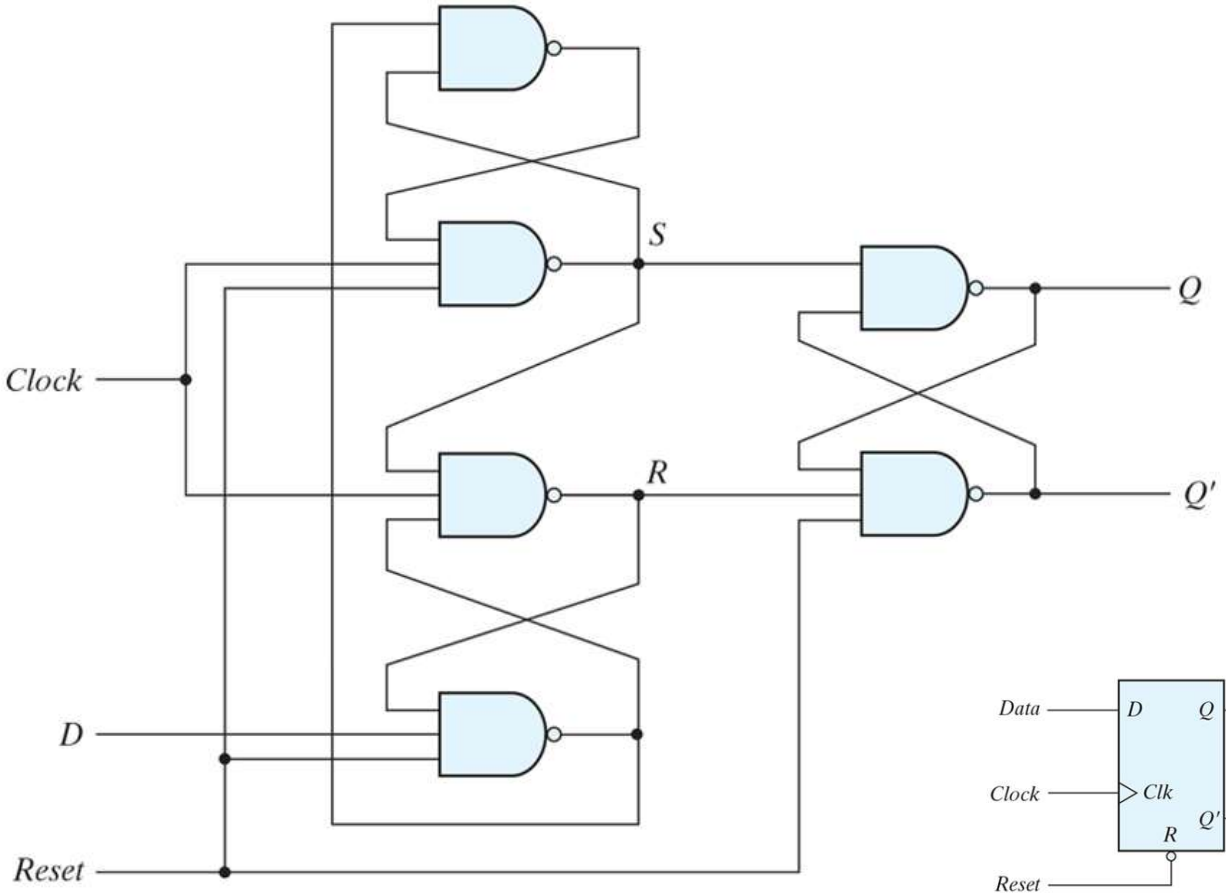


Characteristic Equations

- D flip-flop
  - Q(t+1)=D
- JK flip-flop
  - Q(t+1)=JQ'+K'Q
- T flip-flop
  - Q(t+1)=TQ'+T'Q

Direct Inputs

- Ansynchronous set and/or reset



(a) Circuit diagram

(b) Graphic symbol

R	Clk	D	Q	Q′
0	X	X	0	1
1	↑	0	0	1
1	↑	1	1	0

(c) Function table

Discussion~~~