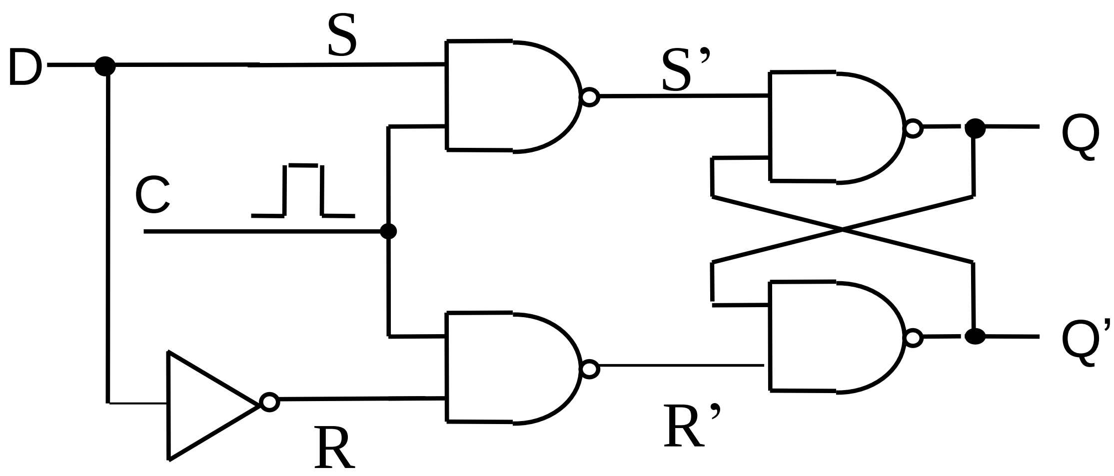
Sequential Circuits: Flip flops

Overview

- Latches respond to trigger levels on control inputs
  - Example: If G = 1, input reflected at output
- Difficult to precisely time when to store data with latches
- Flip flips store data on a rising or falling trigger edge.
  - Example: control input transitions from 0 -> 1, data input appears at output
  - Data remains stable in the flip flop until until next rising edge.
- Different types of flip flops serve different functions
- Flip flops can be defined with characteristic functions.

D Latch



D	C	Q	Q'
0	1	0	1
1	1	1	0
X	0	Q_0	Q_0'

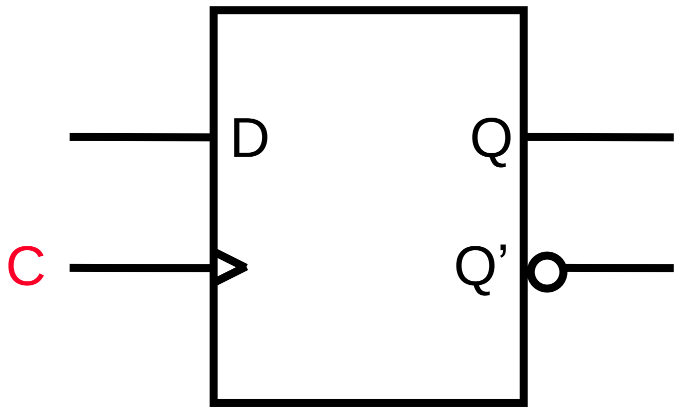
S	R	C	Q	Q'	
0	0	1	Q_0	Q_0'	Store
0	1	1	0	1	Reset
1	0	1	1	0	Set
1	1	1	1	1	Disallowed
X	X	0	Q_0	Q_0'	Store

- **When C is high, D passes from input to output (Q)**

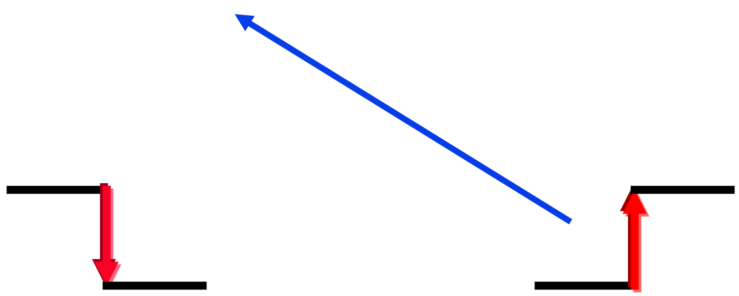
Clocking Event

- **What if the output only changed on a C *transition?***

Positive edge triggered



D	C	Q	Q'
0	↑	0	1
1	↑	1	0
X	0	Q_0	Q_0'



Hi-Lo edge **Lo-Hi edge**

Master-Slave D Flip Flop

- **Consider two latches combined together**
- **Only one *C* value active at a time**
- **Output changes on falling edge of the clock**

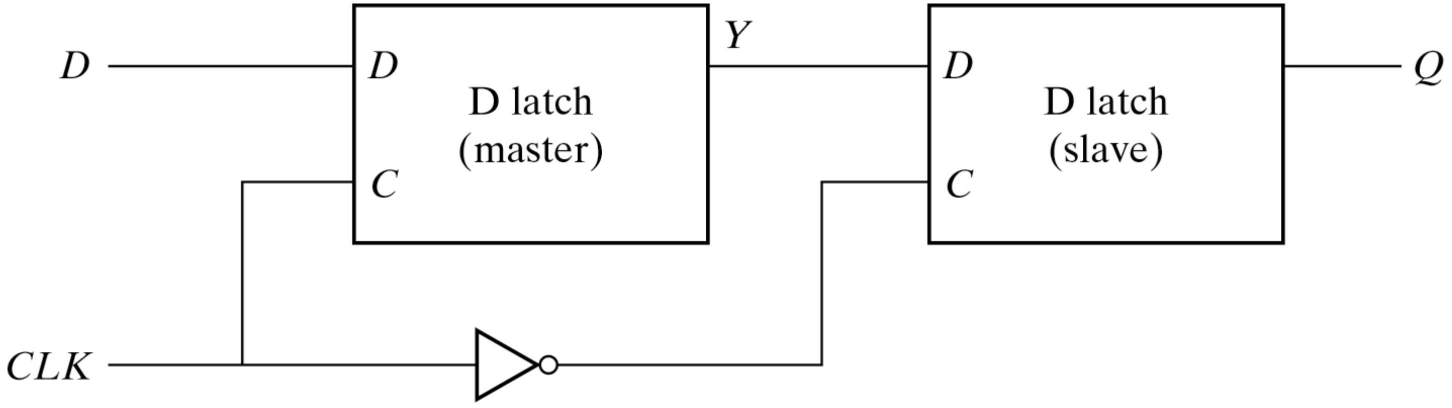
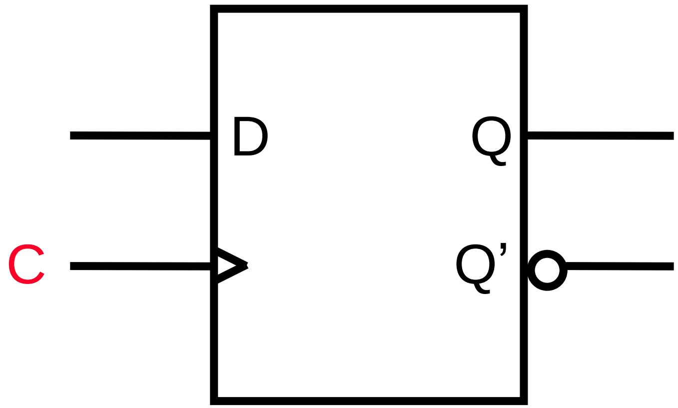


Fig. 5-9 Master-Slave *D* Flip-Flop

D Flip-Flop

- **Stores a value on the positive edge of *C***
- **Input changes at other times have no effect on output**

Positive edge triggered

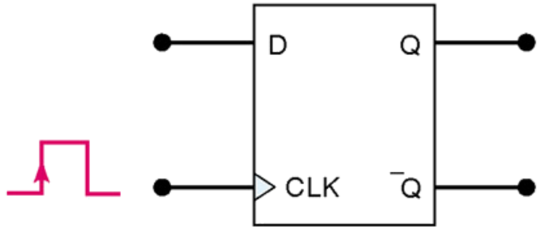


D	C	Q	Q’
0	↑	0	1
1	↑	1	0
X	0	Q_0	Q_0’

D gets latched to Q on the rising edge of the clock.

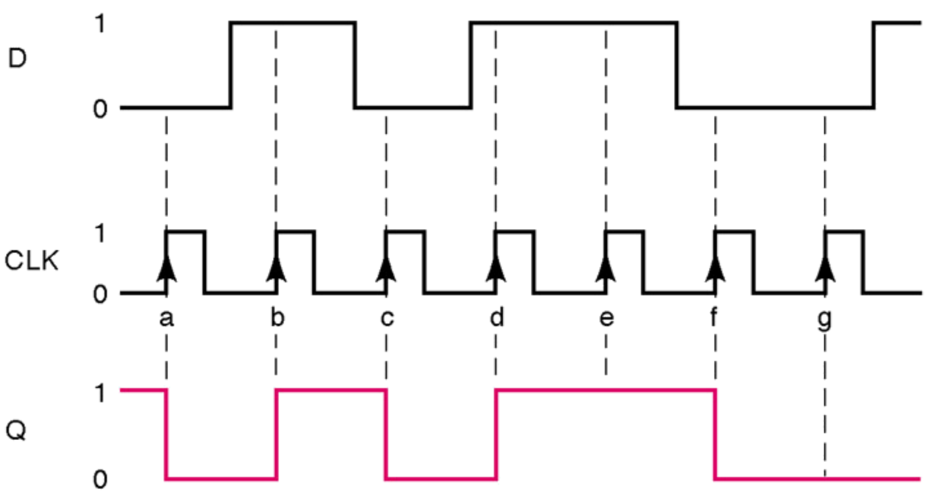
Clocked D Flip-Flop

- Stores a value on the positive edge of *C*
- Input changes at other times have no effect on output



D	CLK	Q
0	↑	0
1	↑	1

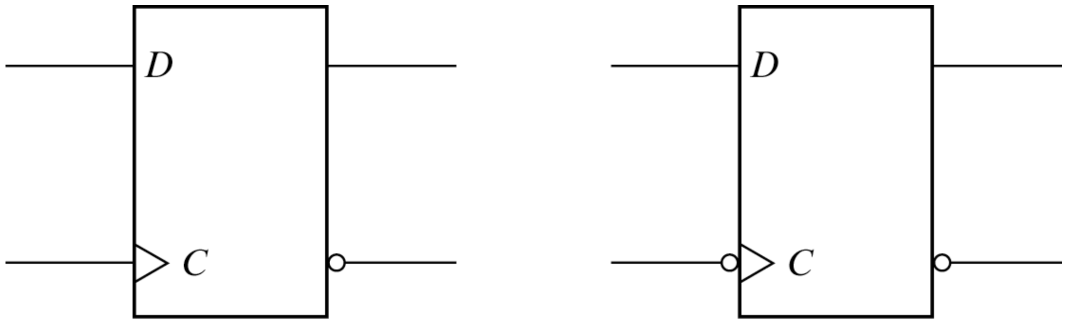
(a)



(b)

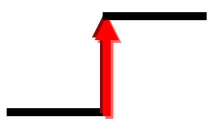
Positive and Negative Edge D Flip-Flop

- **D flops can be triggered on positive or negative edge**
- **Bubble before *Clock (C)* input indicates negative edge trigger**

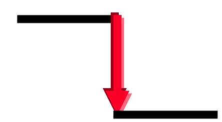


(a) Positive-edge (a) Negative-edge

Fig. 5-11 Graphic Symbol for Edge-Triggered *D* Flip-Flop

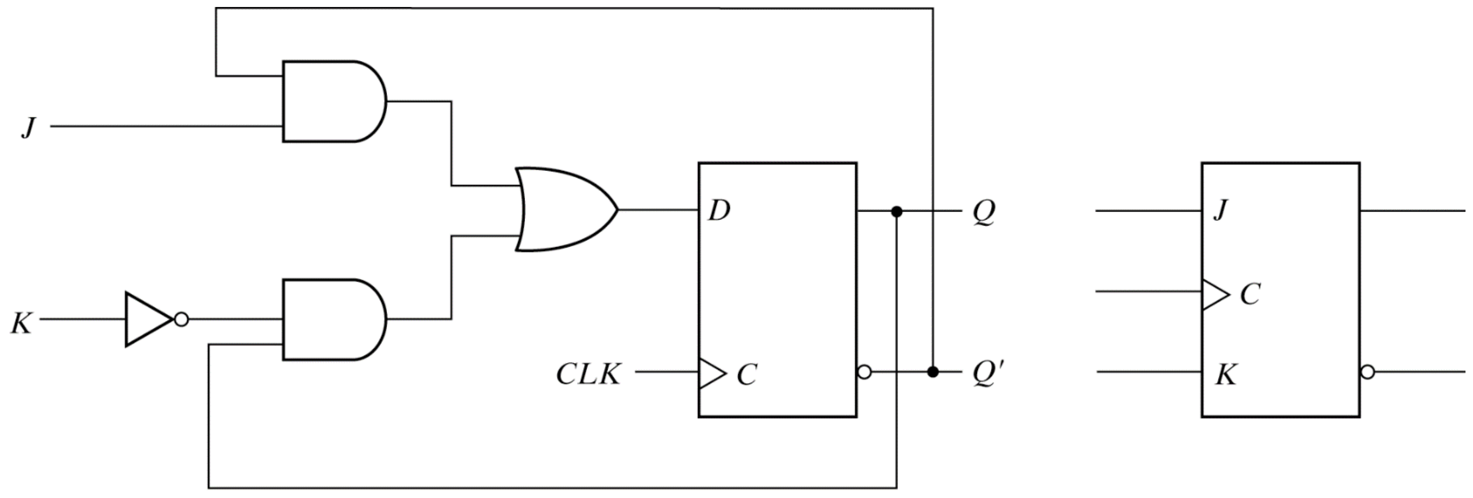


Lo-Hi edge



Hi-Lo edge

Positive Edge-Triggered J-K Flip-Flop



(a) Circuit diagram

(b) Graphic symbol

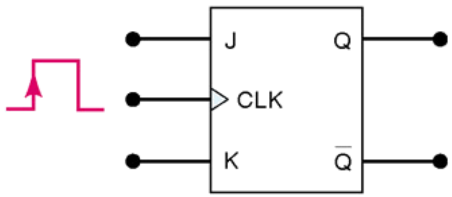
Fig. 5-12 *JK* Flip-Flop

- **Created from D flop**
- **J sets**
- **K resets**
- **J=K=1 -> invert output**

J	K	CLK	Q	Q’
0	0		Q_0	Q_0'
0	1		0	1
1	0		1	0
1	1		TOGGLE	

Clocked J-K Flip Flop

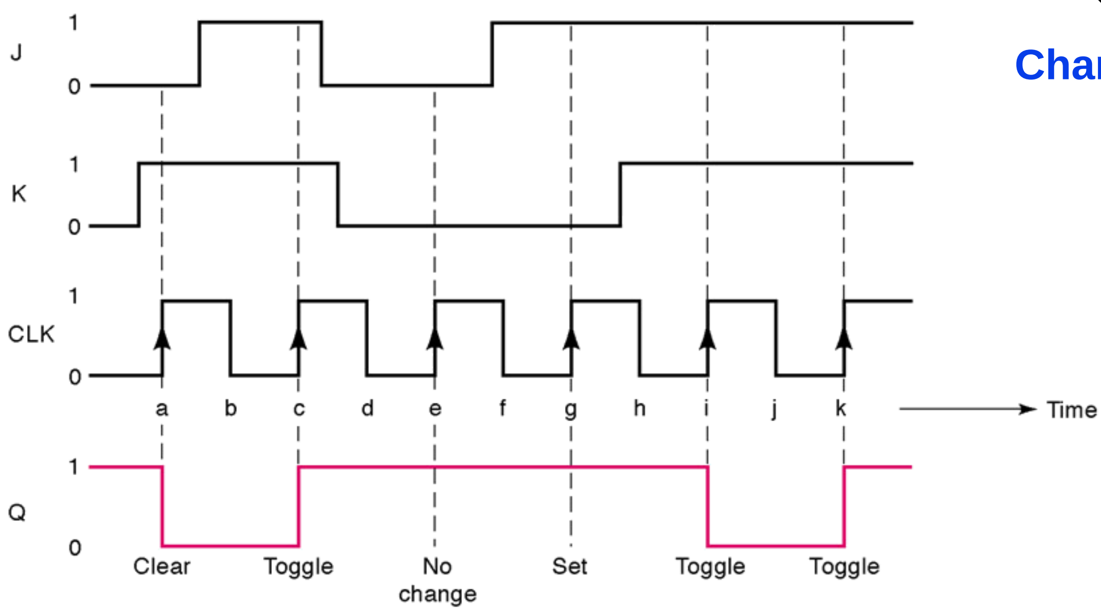
- **Two data inputs, J and K**
- **J -> set, K -> reset, if J=K=1 then toggle output**



J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q_0}$ (toggles)

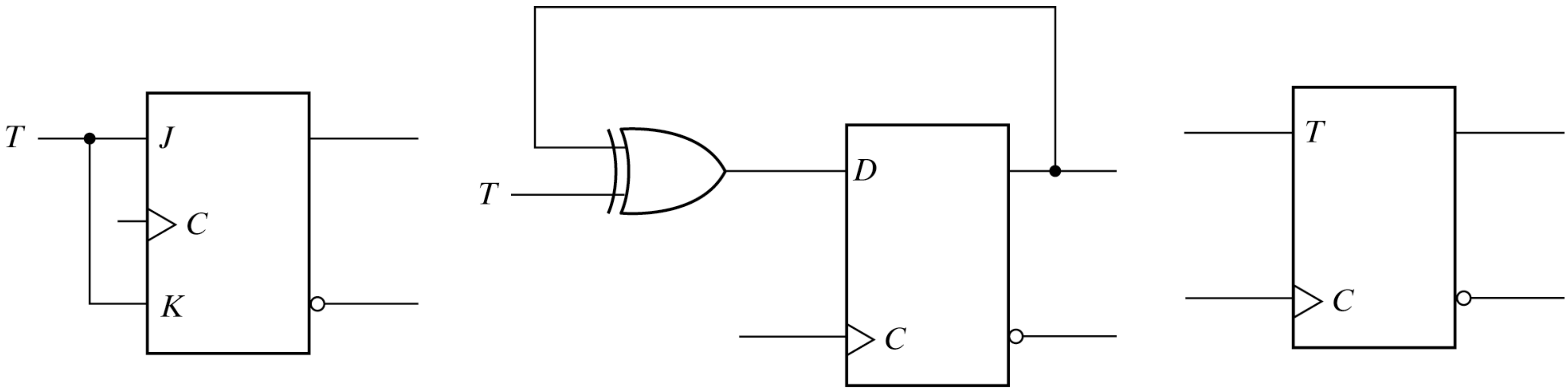
(a)

Characteristic Table



(b)

Positive Edge-Triggered T Flip-Flop



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

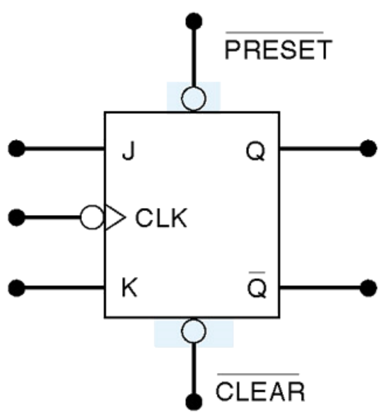
Fig. 5-13 T Flip-Flop

- **Created from D flop**
- **T=0 -> keep current**
- **K resets**
- **T=1 -> invert current**

T	C	Q	Q'
	0	Q_0	Q_0'
	1	TOGGLE	

Asynchronous Inputs

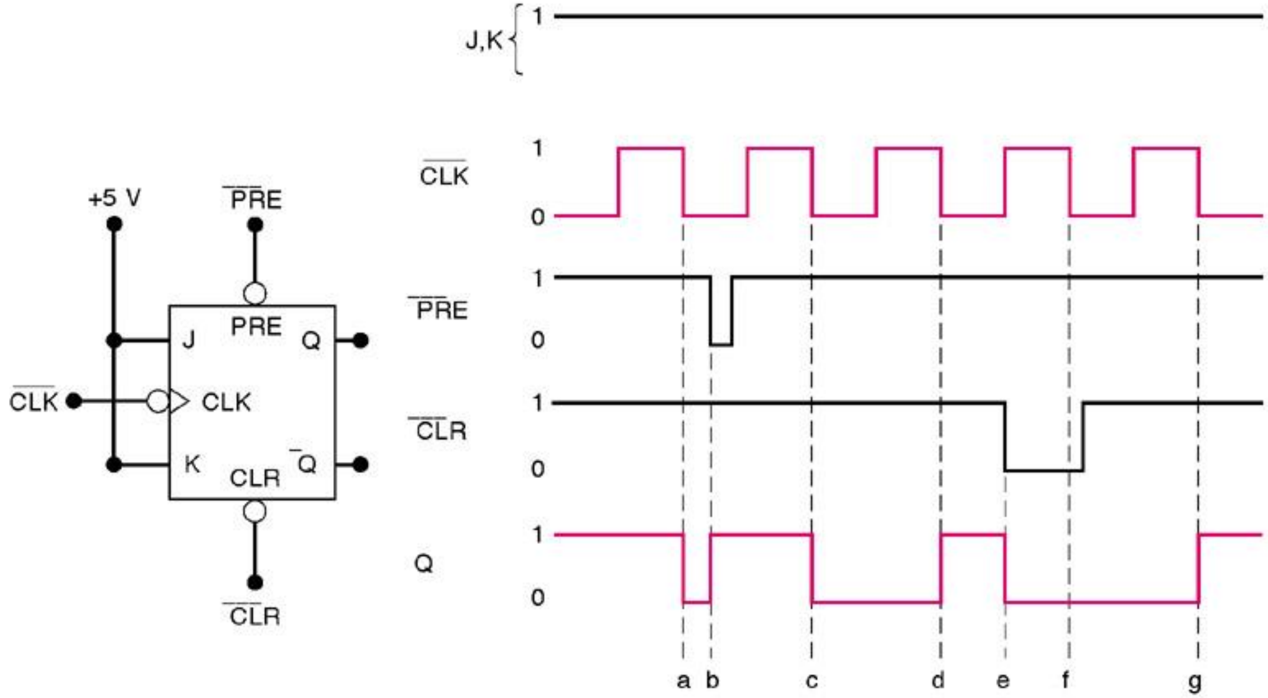
- J, K are **synchronous inputs**
  - o Effects on the output are synchronized with the *CLK* input.
- **Asynchronous inputs** operate independently of the synchronous inputs and clock
  - o Set the FF to 1/0 states *at any time.*



$\overline{\text{PRESET}}$	$\overline{\text{CLEAR}}$	FF response
1	1	Clocked operation*
0	1	Q = 1 (regardless of CLK)
1	0	Q = 0 (regardless of CLK)
0	0	Not used

*Q will respond to J, K, and CLK

Asynchronous Inputs

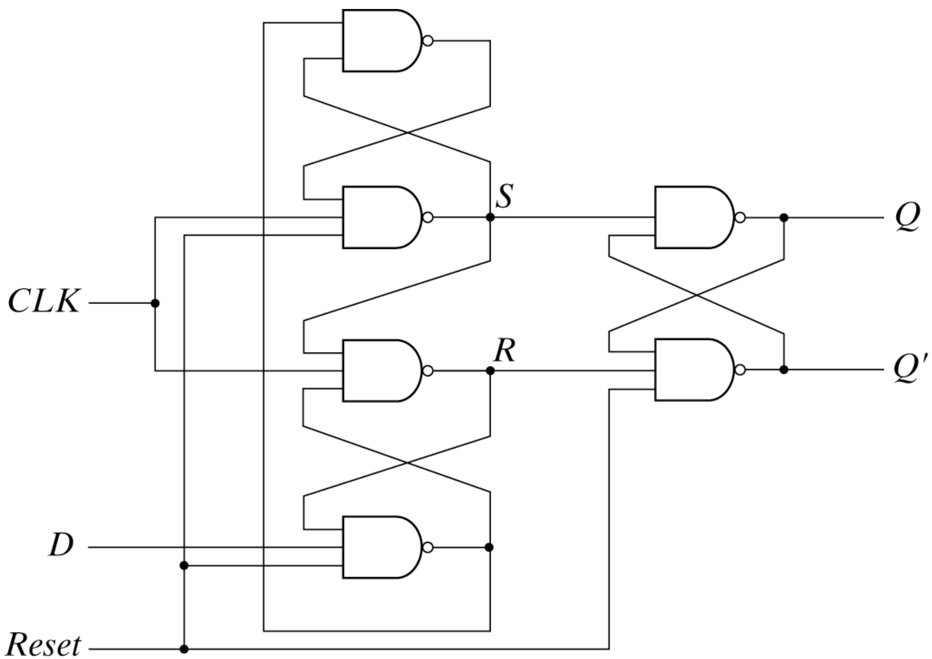


(a)

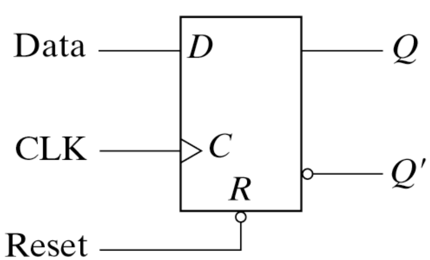
Point	Operation
a	Synchronous toggle on NGT of $\overline{CLK}$
b	Asynchronous set on $\overline{PRE}$ = 0
c	Synchronous toggle
d	Synchronous toggle
e	Asynchronous clear on $\overline{CLR}$ = 0
f	$\overline{CLR}$ over-rides the NGT of $\overline{CLK}$
g	Synchronous toggle

(b)

Asynchronous Inputs



(a) Circuit diagram



(b) Graphic symbol

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

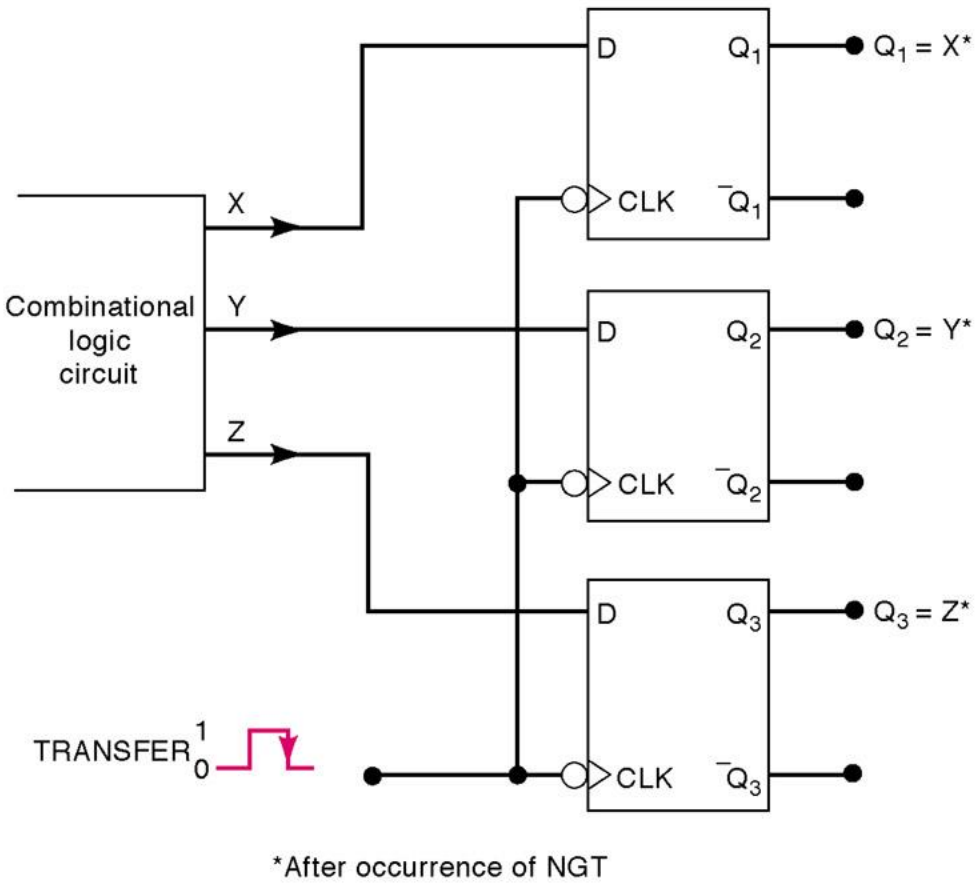
(b) Function table

Fig. 5-14 *D* Flip-Flop with Asynchronous Reset

- Note reset signal (R) for D flip flop
- If R = 0, the output Q is cleared
- This event can occur at any time, regardless of the value of the CLK

Parallel Data Transfer

- Flip flops store outputs from combinational logic
- Multiple flops can store a collection of data



Summary

- Flip flops are powerful storage elements
  - They can be constructed from gates and latches!
- D flip flop is simplest and most widely used
- Asynchronous inputs allow for clearing and presetting the flip flop output
- Multiple flops allow for data storage
  - The basis of computer memory!
- Combine storage and logic to make a computation circuit
- Next time: Analyzing sequential circuits.