ENGG2020 Digital Logic and Systems

Chapter 5: Flip-flops, Registers and Counters

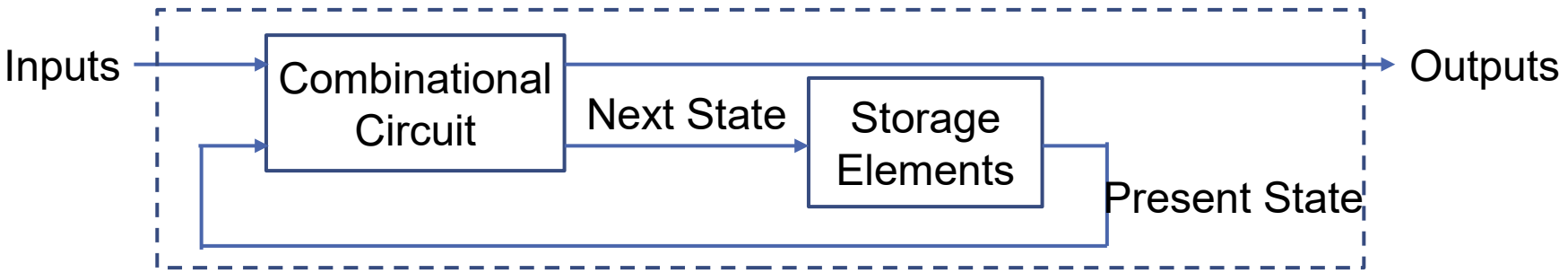
The Chinese University of Hong Kong

Sequential Circuit

The output of a sequential circuit is a function of the time sequence of inputs and the state

Asynchronous sequential circuits: State updated at any time

Synchronous sequential circuits: State updated at discrete time (fixed and regular intervals)



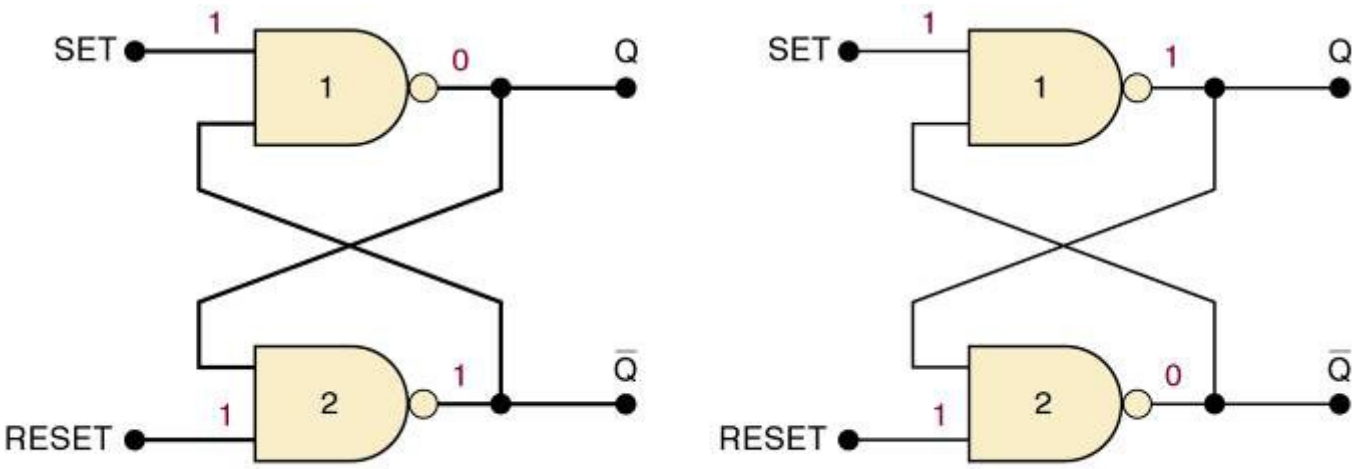
NAND Gate Latch

The **NAND** gate latch or simply latch is a basic FF.
Inputs are *SET* and *CLEAR* (*RESET*).

Inputs are active-LOW—output will change when the input is pulsed LOW.

When the latch is set: **Q** = 1 and **Q** = 0

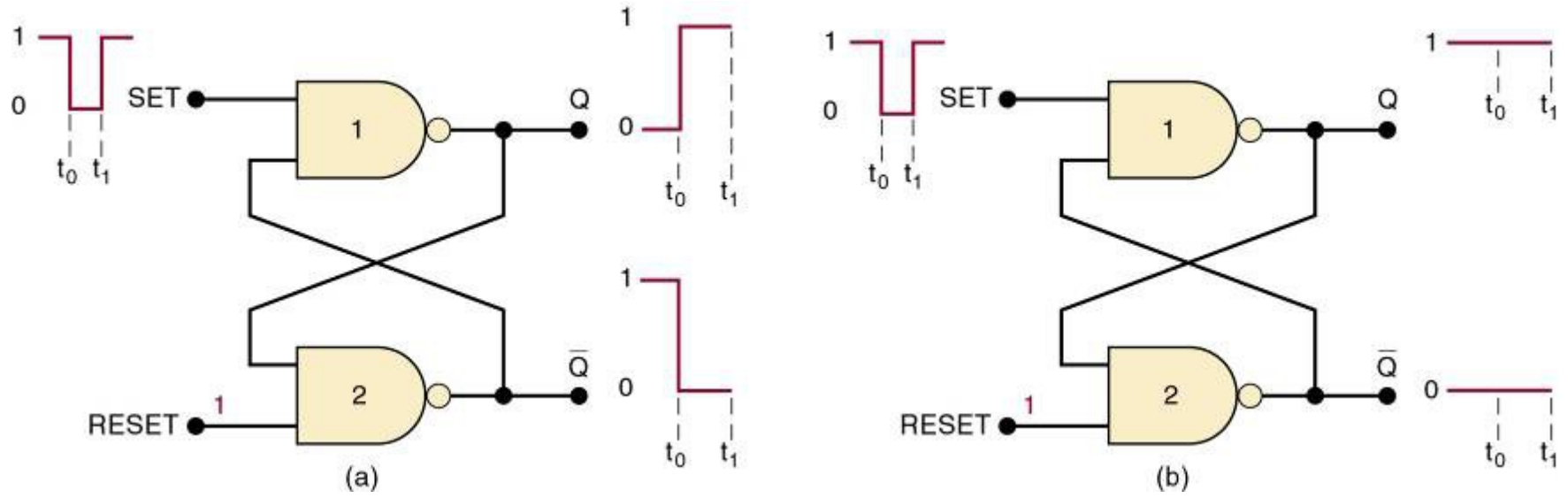
When the latch is clear or reset: **Q** = 0 and **Q** = 1



NAND Gate Latch

Pulsing the SET input to the 0 state...
(a) *Q* = 0 prior to SET pulse.
(b) *Q* = 1 prior to SET pulse.

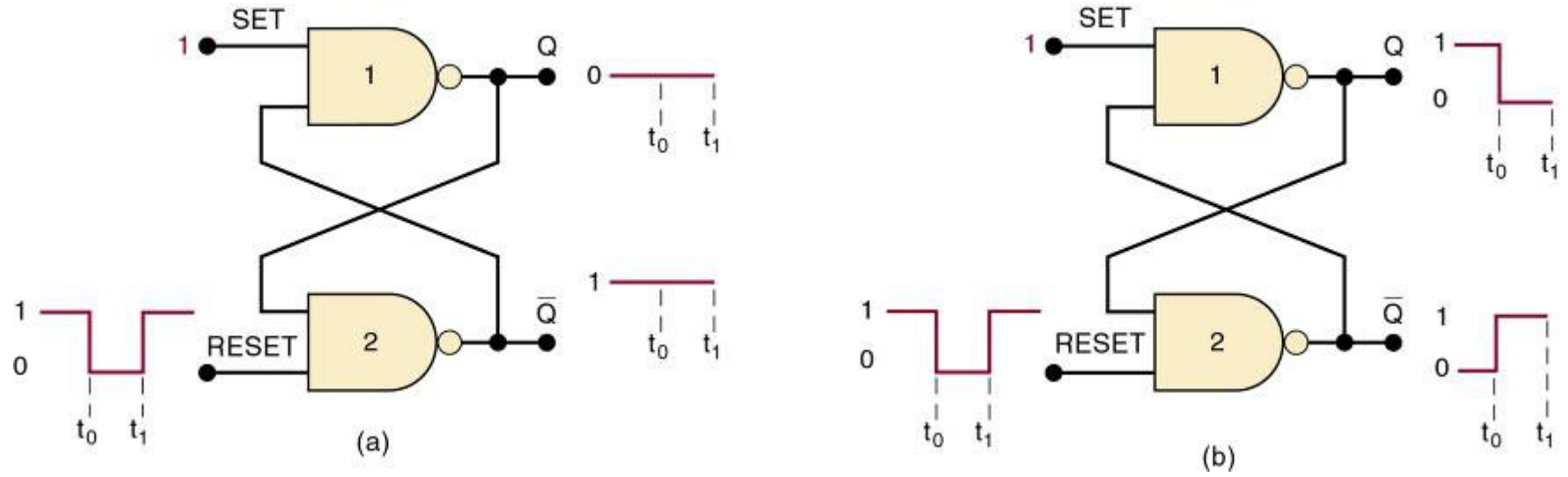
In both cases, *Q* ends up HIGH.



NAND Gate Latch

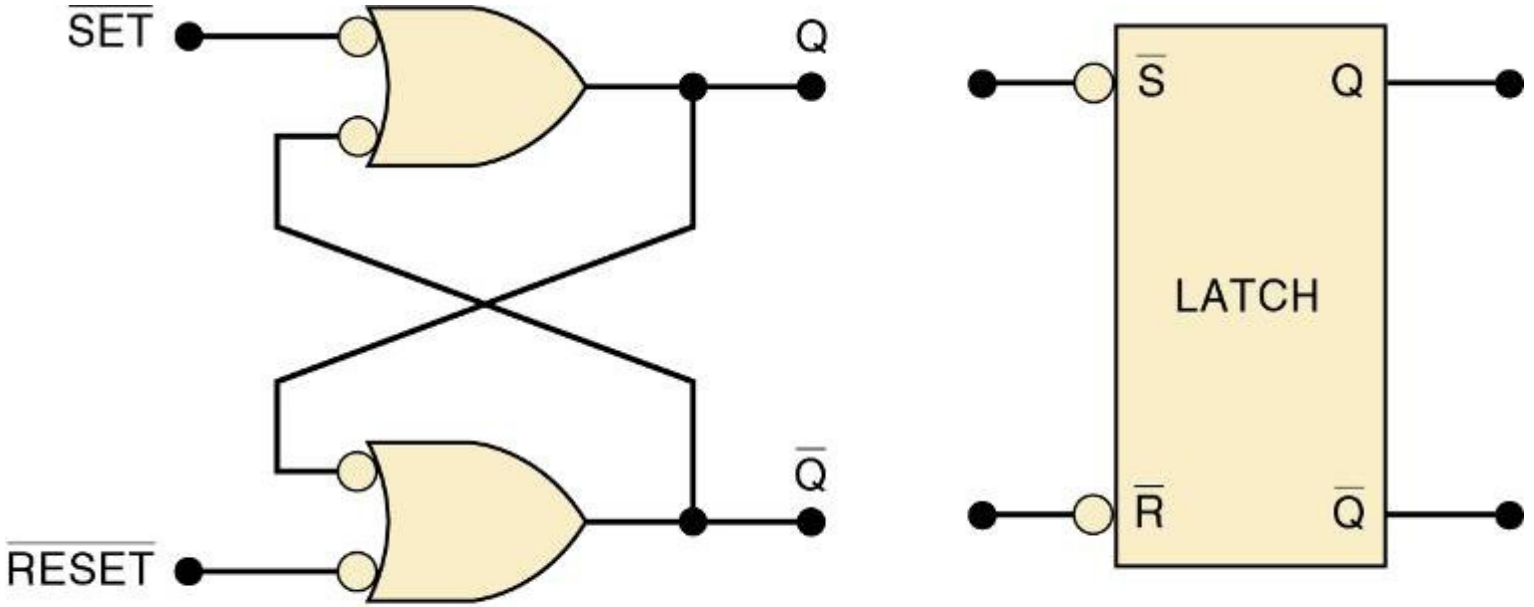
Pulsing RESET LOW when...
(a) *Q* = 0 prior to the RESET pulse.
(b) *Q* = 1 prior to the RESET pulse.

In each case, *Q* ends up LOW.



NAND Gate Latch

NAND latch equivalent representations and simplified block diagram.



NAND Gate Latch

Summary of the **NAND** latch:

SET = 1, RESET = 1—Normal resting state, outputs remain in state they were in prior to input.

SET = 0, RESET = 1—Output will go to *Q* = 1 and remains there, even after SET returns HIGH.

Called *setting* the latch.

SET = 1, RESET = 0—Will produce *Q* = 0 LOW and remains there, even after RESET returns HIGH.

Called *clearing* or *resetting* the latch.

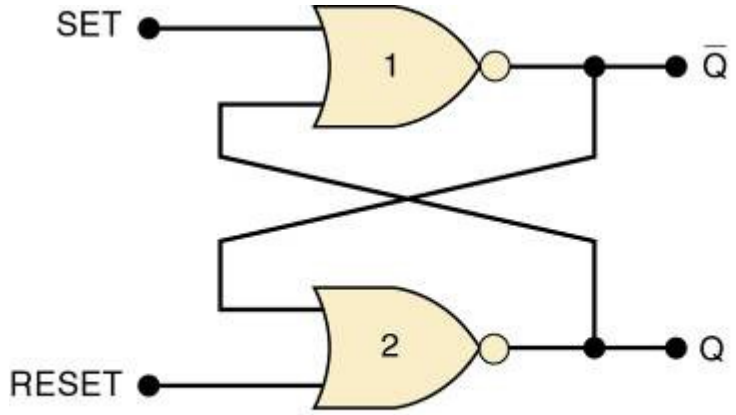
SET = 0, RESET = 0—Tries to set and clear the latch at the same time, and produces

Output is unpredictable, and this input condition should not be used.

NOR Gate Latch

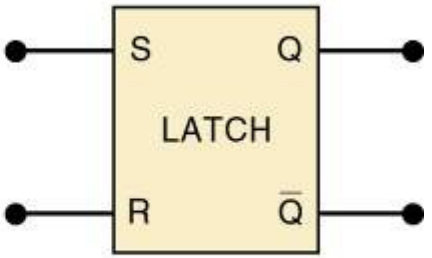
Two cross-coupled **NOR** gates can be used as a **NOR** gate latch—similar to the **NAND** latch.

The SET and RESET inputs are active-HIGH.
Output will change when the input is pulsed HIGH.



Set	Reset	Output
0	0	No change
1	0	Q = 1
0	1	Q = 0
1	1	Invalid*

*Produces $Q = \overline{Q} = 0$.



NOR Gate Latch

Summary of the **NOR** latch:

SET = 0, RESET = 0—Normal resting state, No effect on output state.

SET = 1, RESET = 0—will always set $Q = 1$, where it remains even after SET returns to 0.

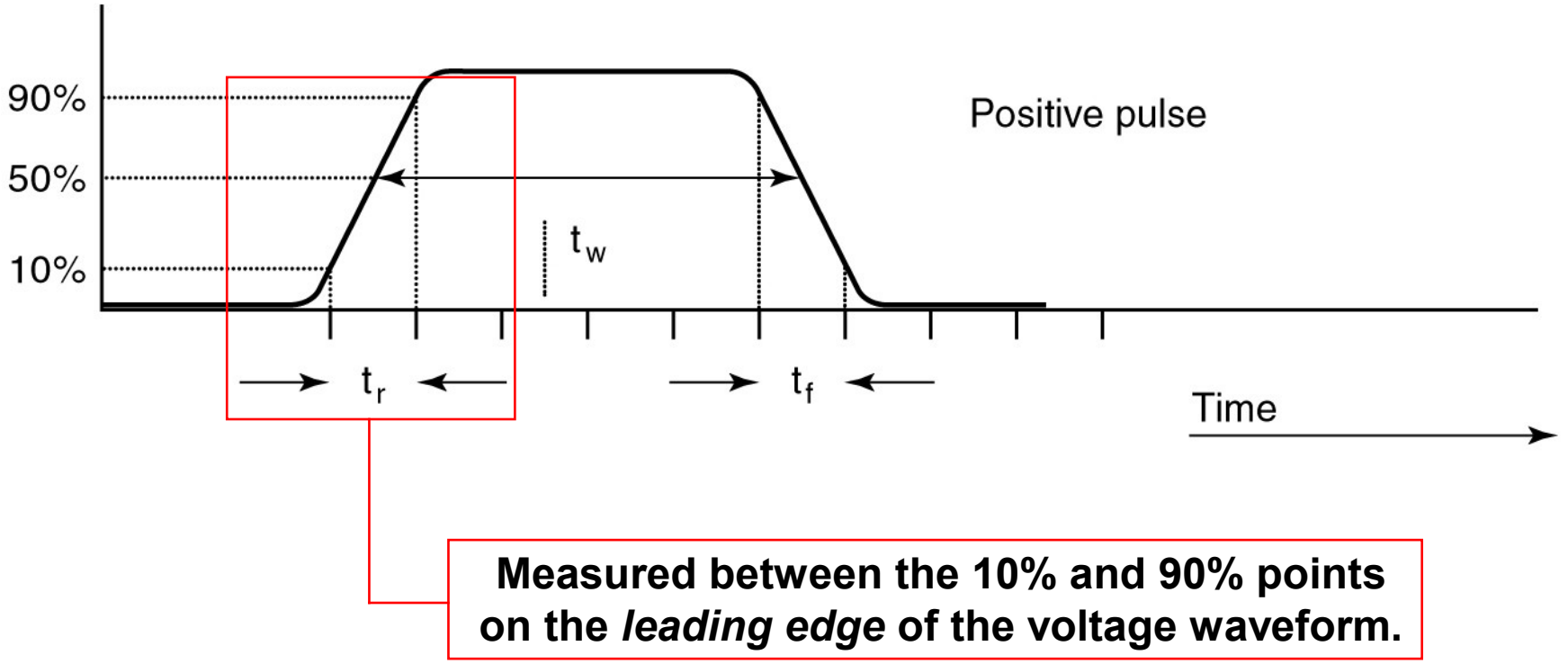
SET = 0, RESET = 1—will always clear $Q = 0$, where it remains even after RESET returns to 0.

SET = 1, RESET = 1—Tries to set and reset the latch at the same time, and produces $Q = Q' = 0$.

Output is unpredictable, and this input condition should not be used.

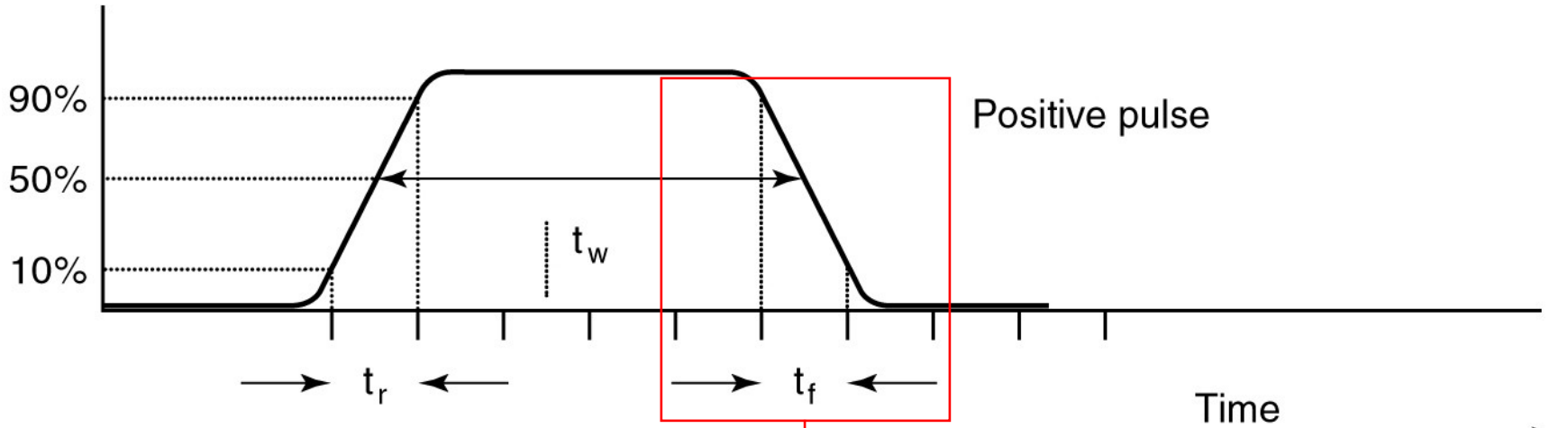
Digital Pulses

Transition from LOW to HIGH on a positive pulse is called *rise time* (t_r).



Digital Pulses

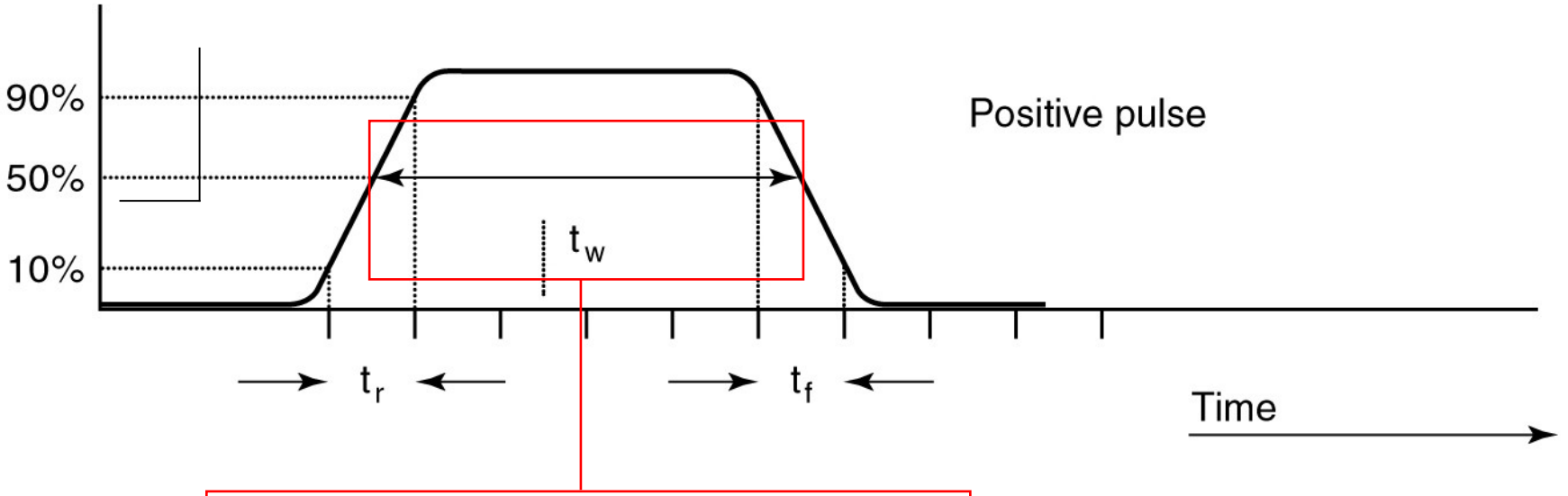
Transition from HIGH to LOW on a positive pulse is called *fall time* (t_f).



Measured between the 90% and 10% points on the *trailing edge* of the voltage waveform.

Digital Pulses

A pulse also has a *duration*—width—(t_w).



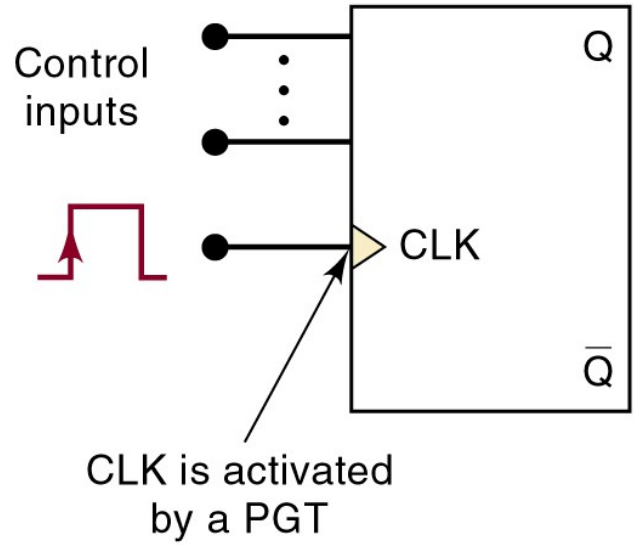
The time between the points when the leading and trailing edges are at 50% of the HIGH level voltage.

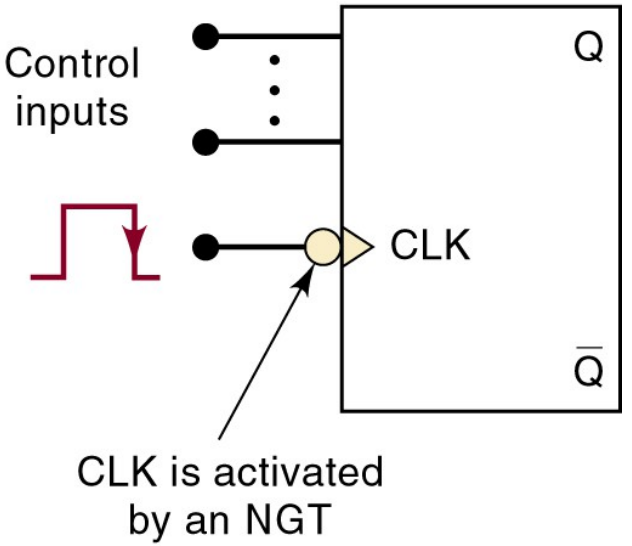
Clocked Flip-Flops

The clock signal is a rectangular pulse train or square wave.

Clocked FFs change state on one or the other clock transitions.

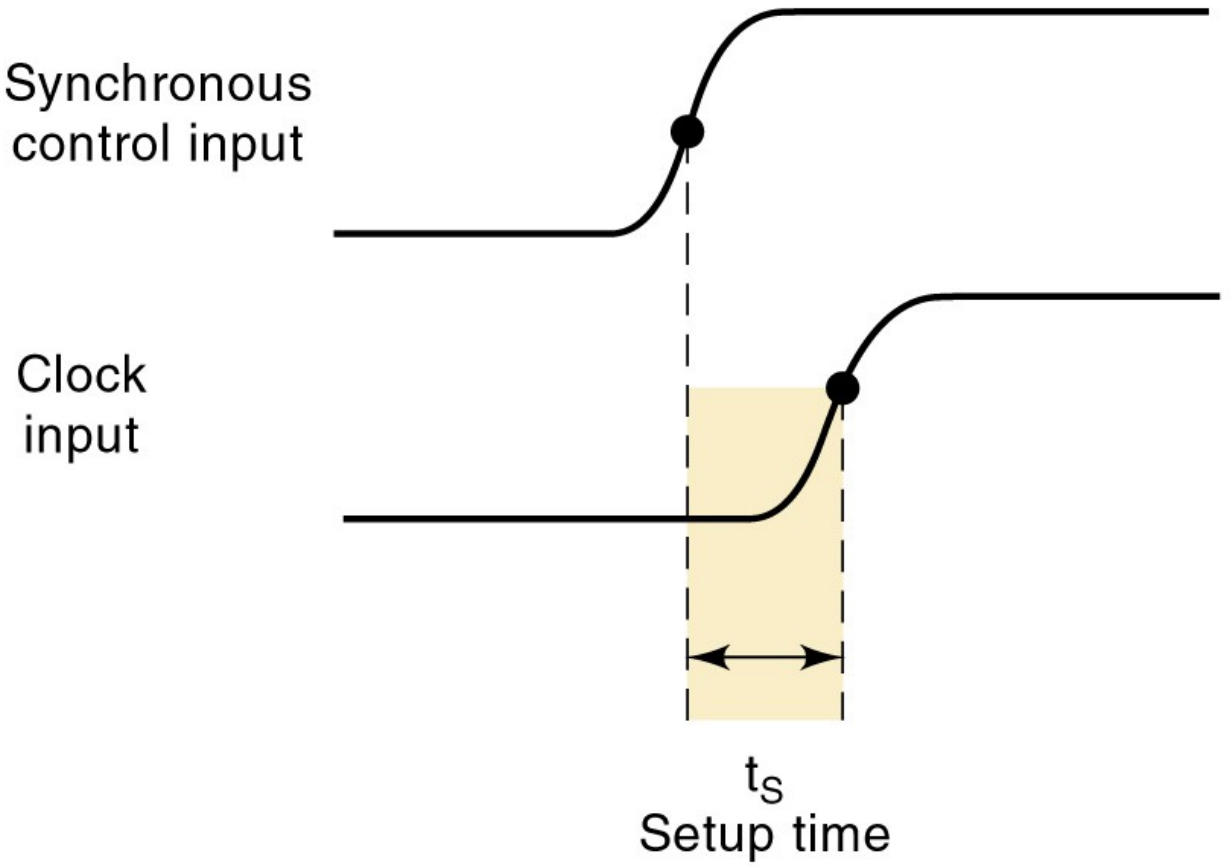
Clock inputs are labeled CLK, CK, or CP.





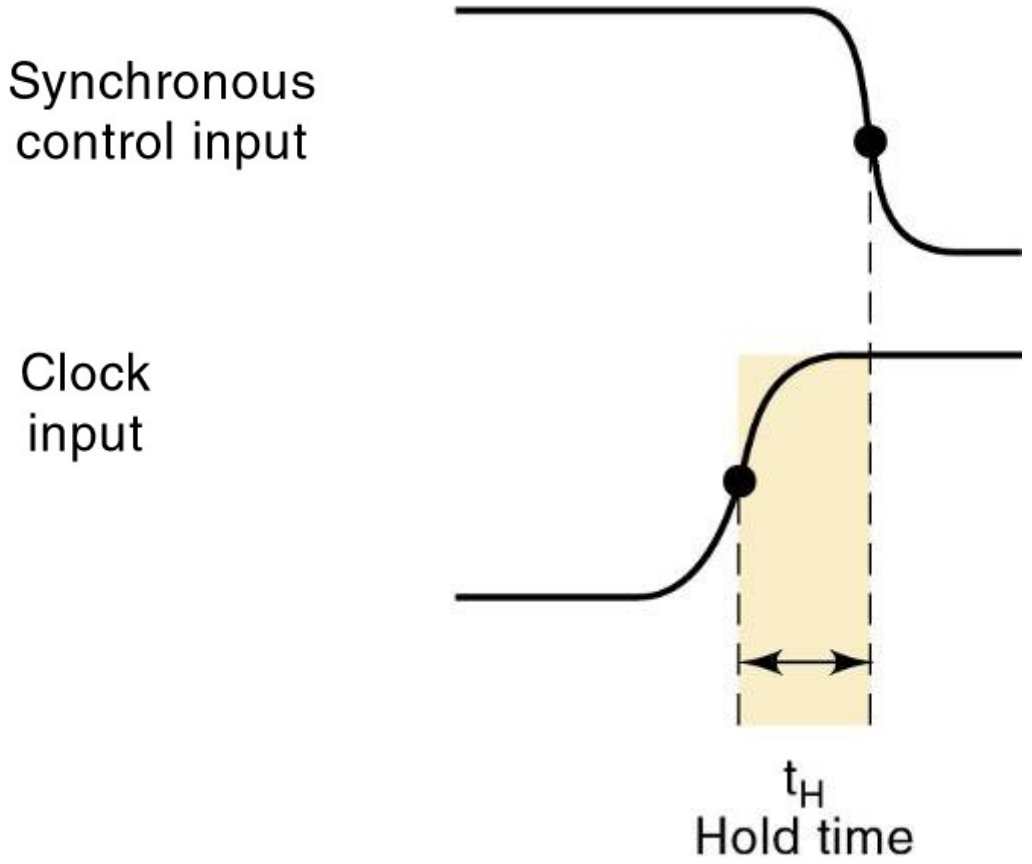
Clocked Flip-Flops

Setup time (t_S) is the minimum time interval before the active CLK transition that the control input must be kept at the proper level.



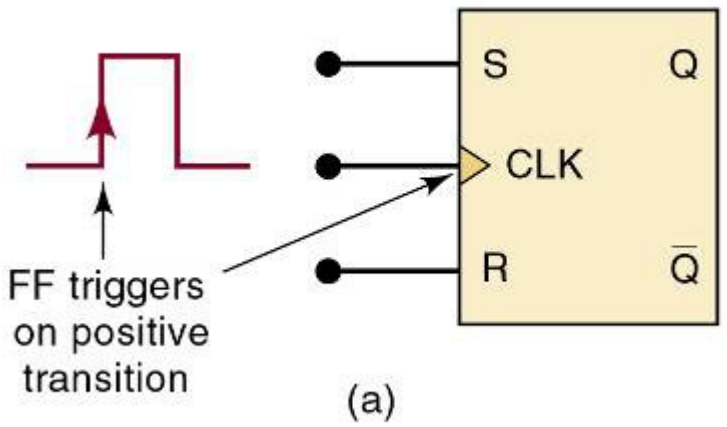
Clocked Flip-Flops

Hold time (t_H) is the time following the active transition of the CLK, during which the control input must kept at the proper level.



Clocked Flip-Flops

A clocked S-R flip-flop triggered by the positive-going edge of the clock signal.



(a)

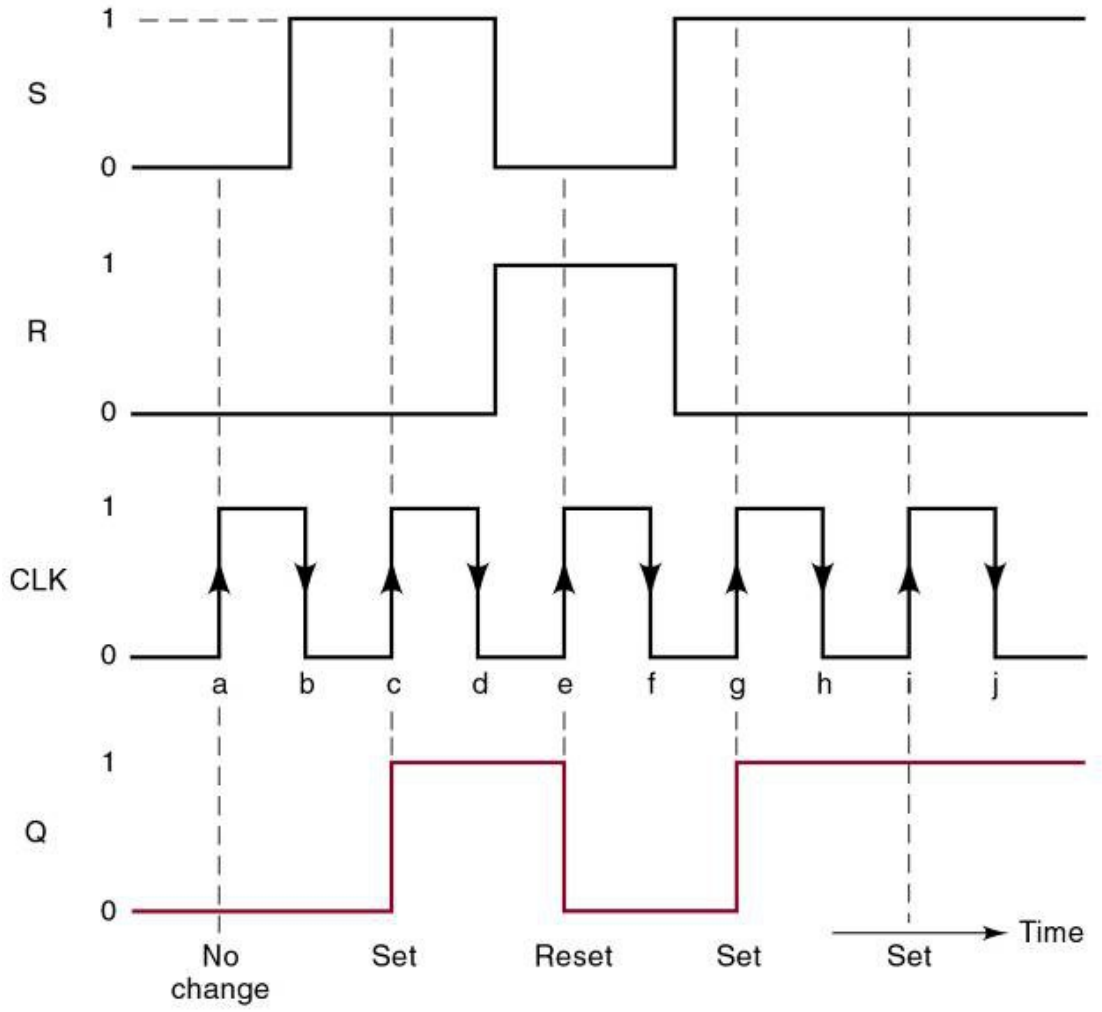
Inputs			Output
S	R	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambiguous

Q_0 is output level prior to ↑ of CLK.
↓ of CLK produces no change in Q.

(b)

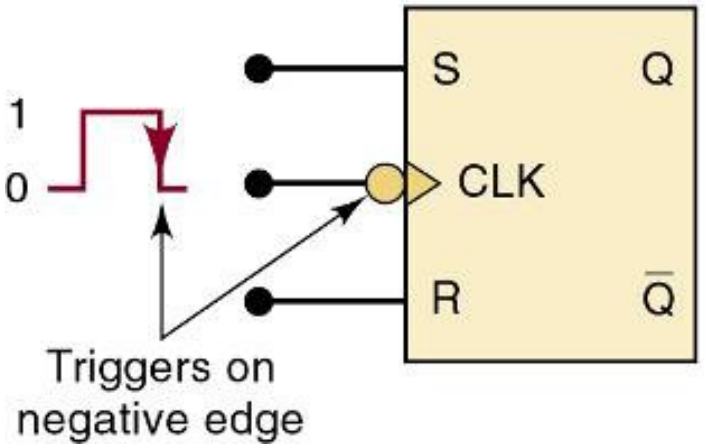
Clocked Flip-Flops

Waveforms of the operation of a clocked S-R flip-flop triggered by the positive-going edge of a clock pulse.



Clocked Flip-Flops

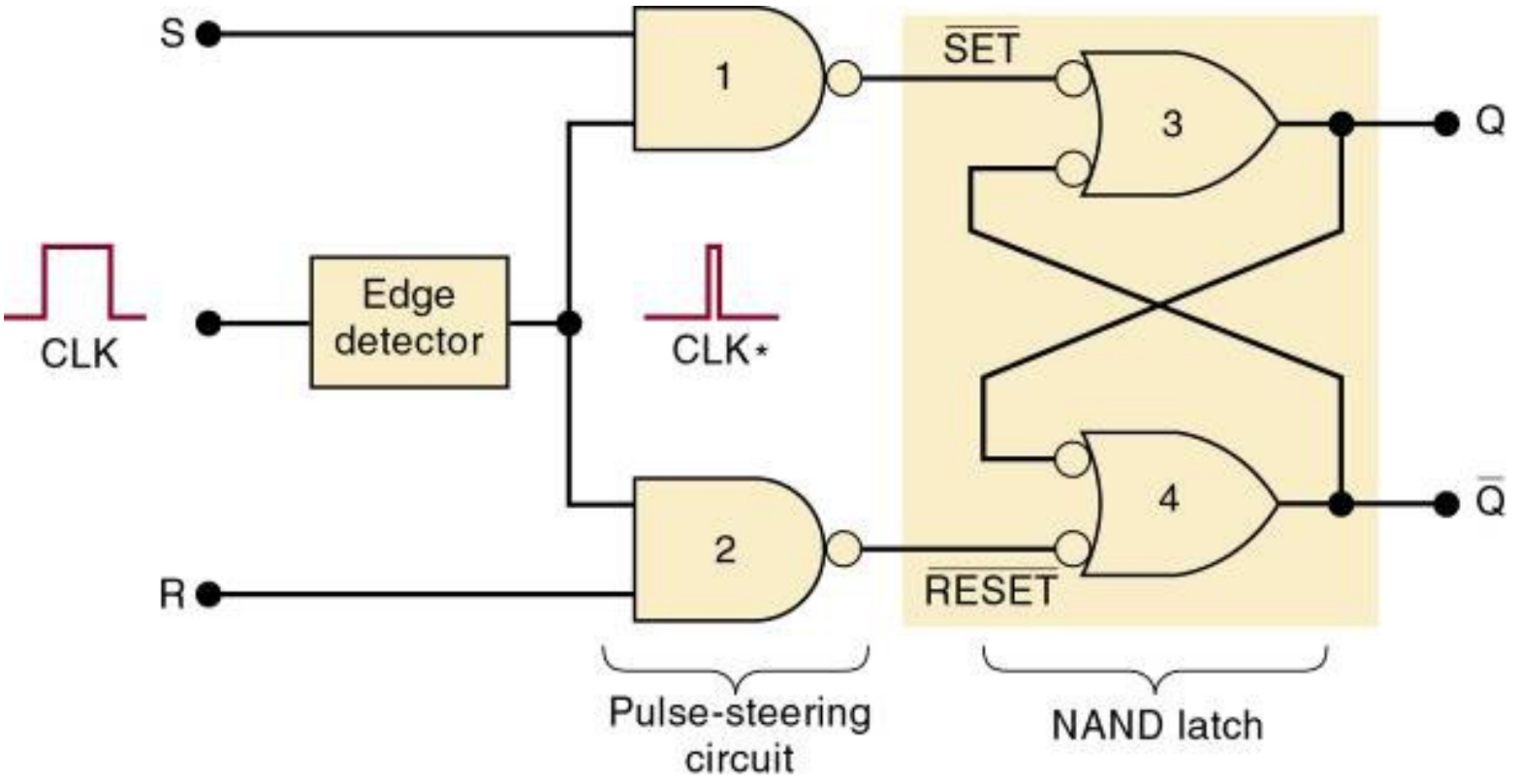
A clocked S-R flip-flop triggered by the negative-going edge of the clock signal.



Inputs			Output
S	R	CLK	Q
0	0	↓	Q_0 (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	Ambiguous

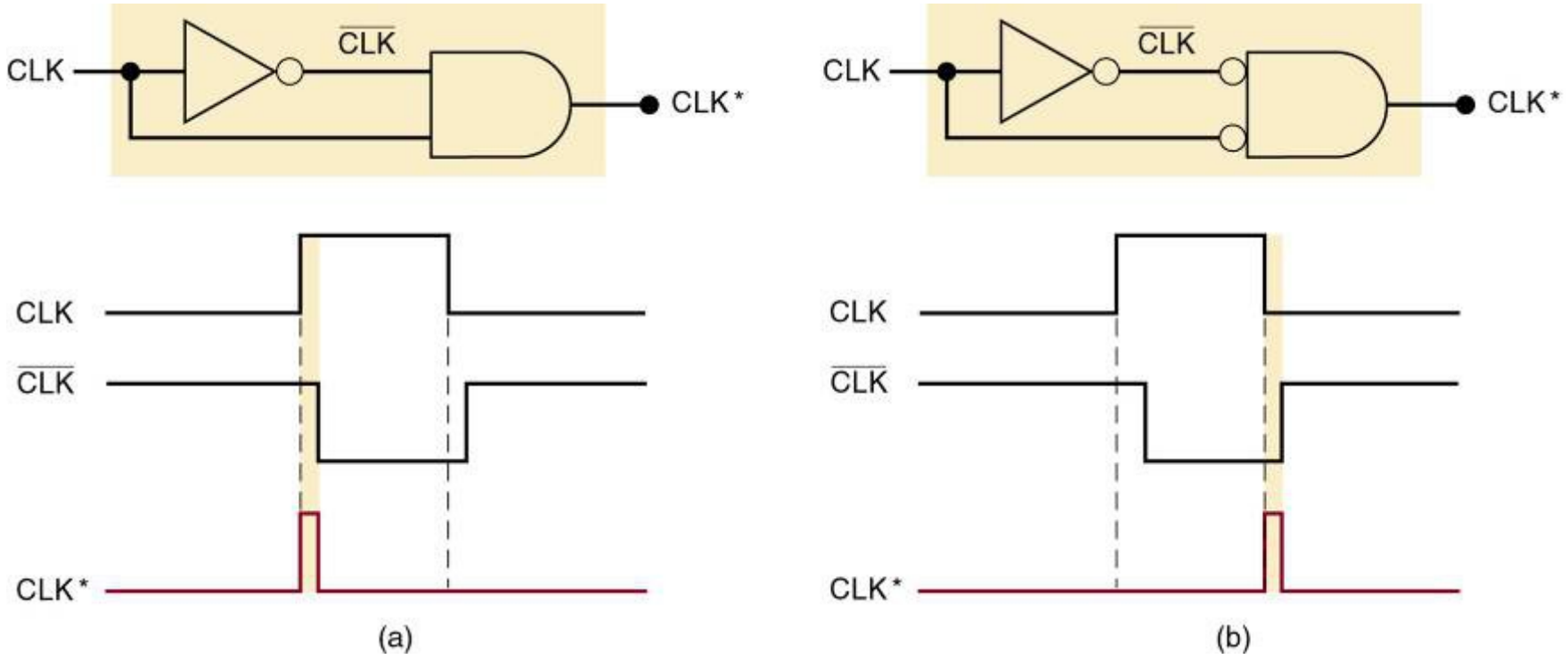
Clocked Flip-Flops

Clocked S-R Flip-Flop – Internal Circuitry



Clocked Flip-Flops

Implementation of edge-detector circuits used in edge-triggered flip-flops: (a) PGT; (b) NGT.



Clocked Flip-Flops

Clocked J-K Flip-Flop: J is SET, K is CLEAR.

When J and K are both HIGH, output is toggled to the opposite state.

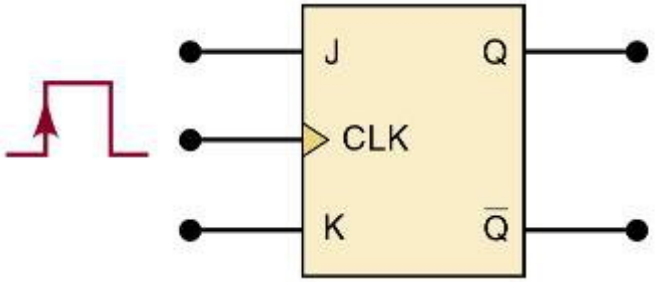
May be positive going or negative going clock trigger.

Much more versatile than the S-R flip-flop, as it has no ambiguous states.

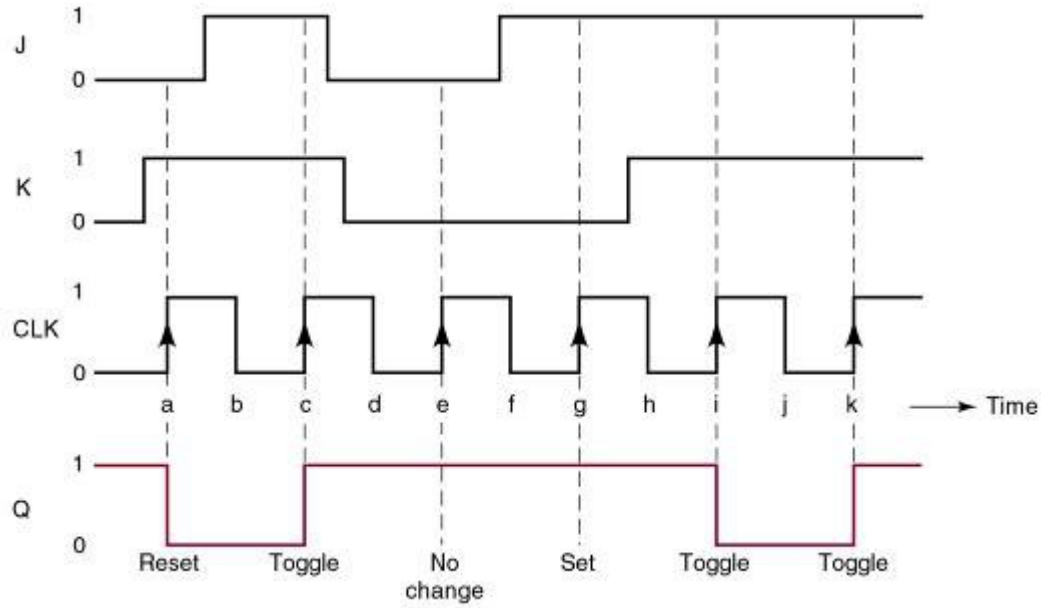
Has the ability to do everything the S-R FF does, plus operates in toggle mode.

Clocked Flip-Flops

Clocked J-K flip-flop that responds only to the positive edge of the clock.



J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q}_0$ (toggles)



Clocked Flip-Flops

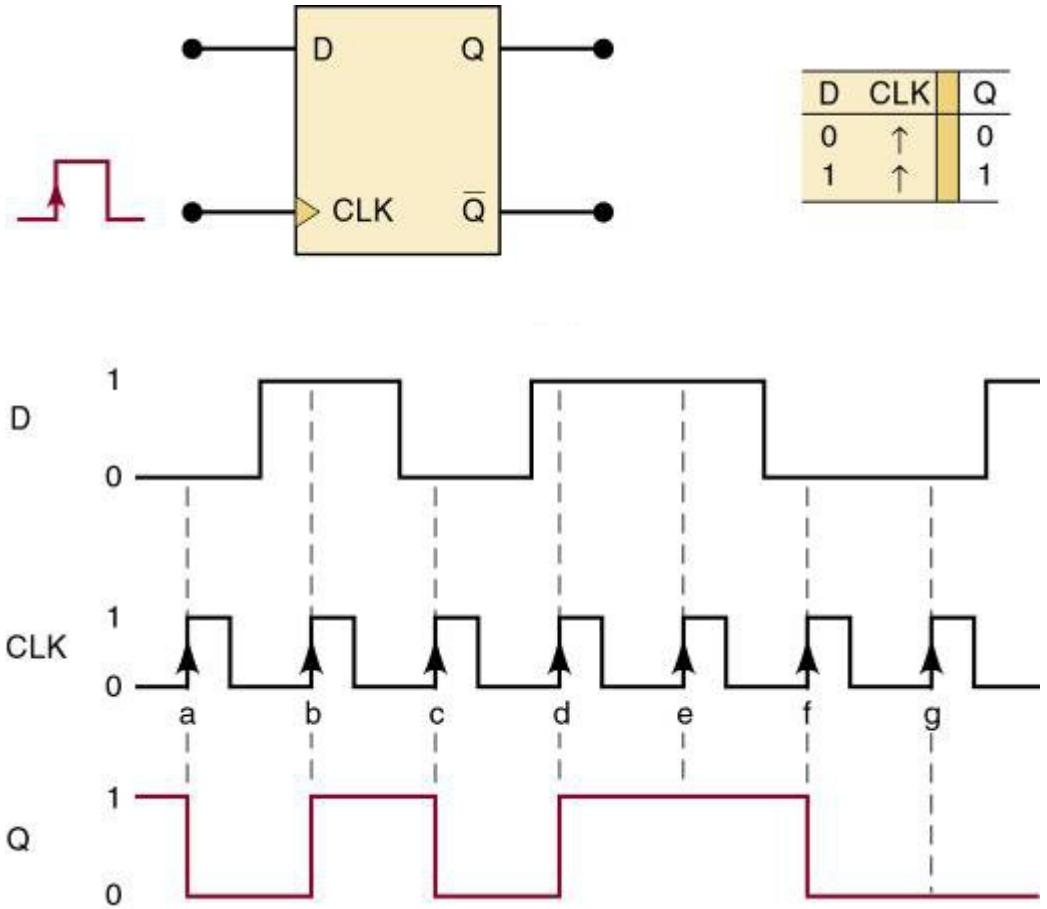
Clocked D Flip-Flop: One data input—output changes to the value of the input at either the positive- or negative-going clock trigger.

May be implemented with a J-K FF by tying the J input to the K input through an inverter.

Useful for parallel data transfer.

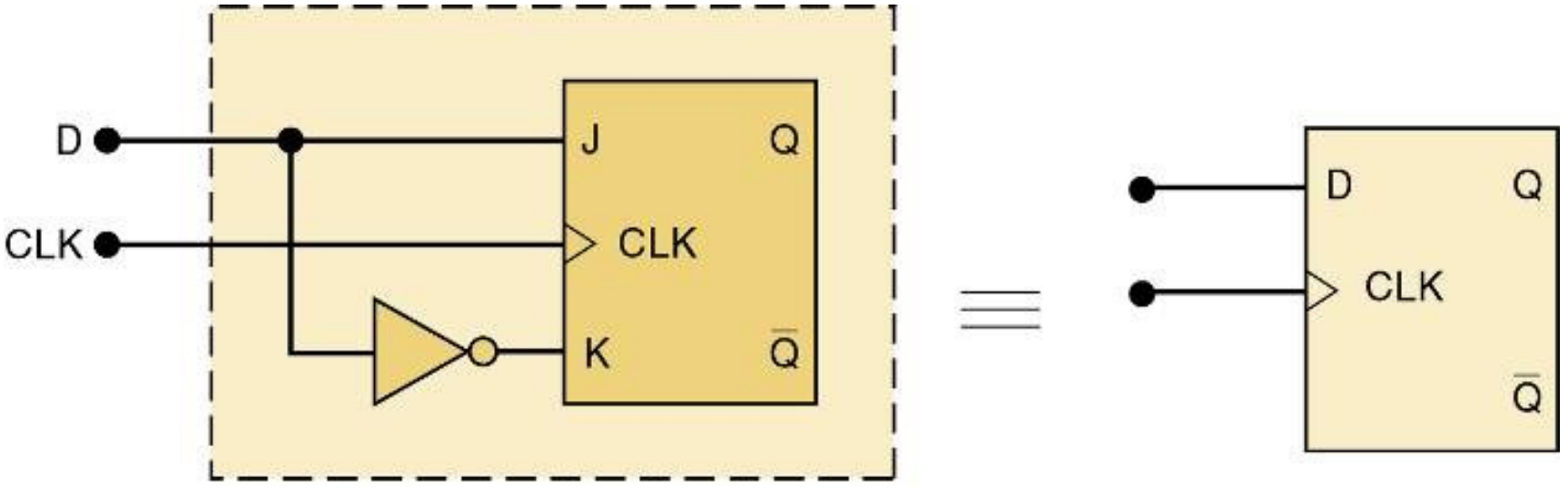
Clocked Flip-Flops

D flip-flop that triggers only on positive-going transitions.



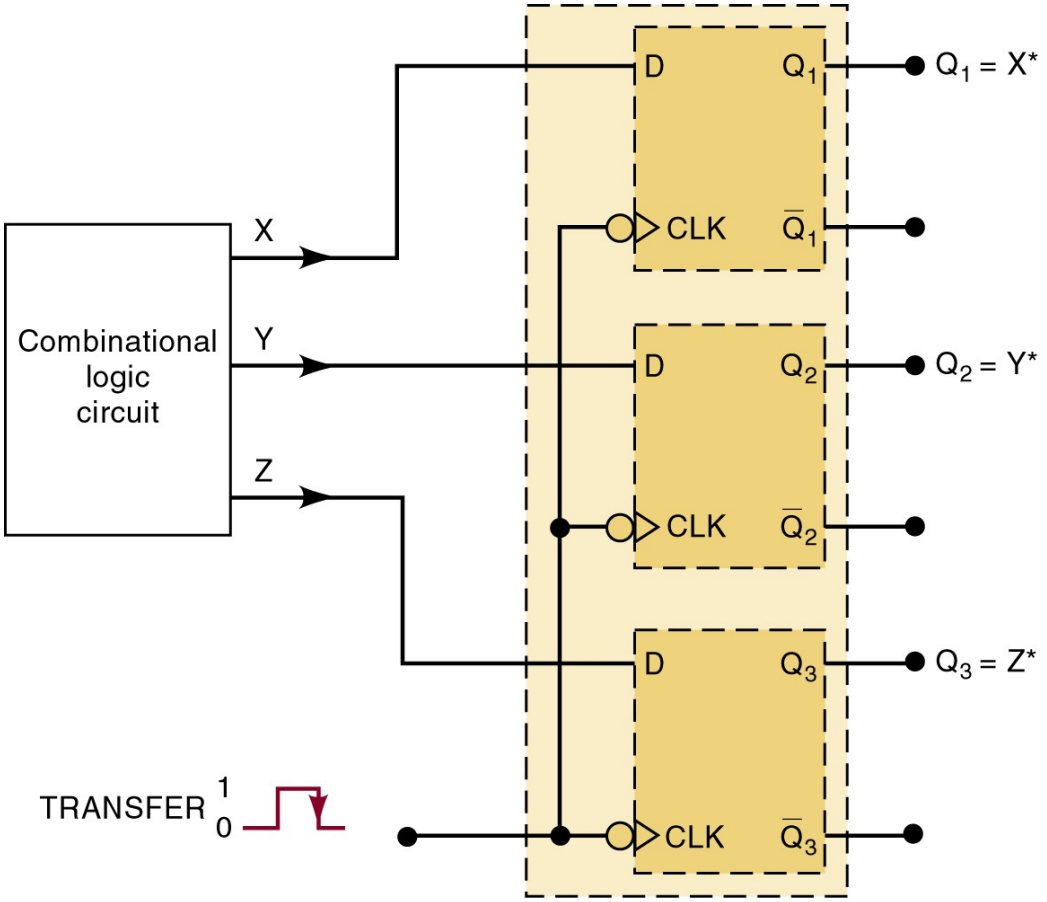
Clocked Flip-Flops

An edge-triggered D flip-flop is implemented by adding a single INVERTER to the edge-triggered J-K flip-flop.



Clocked Flip-Flops

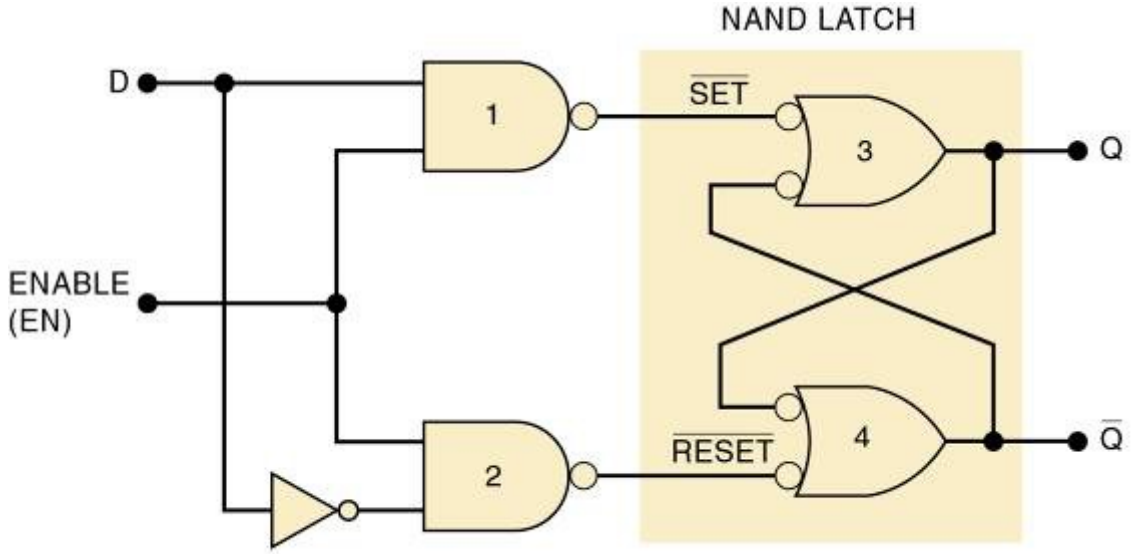
Outputs *X, Y, Z* are to be transferred to FFs *Q1*, *Q2*, and *Q3* for storage. This is an example of parallel data transfer of binary data



Clocked Flip-Flops

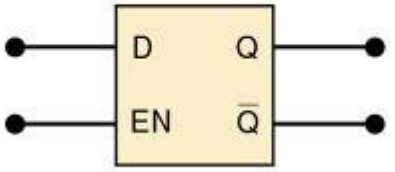
The edge-triggered D flip-flop uses an edge-detector circuit to ensure the output responds to the *D* input *only* on active transition of the clock.

If this edge detector is not used, the resultant circuit operates as a **D latch**.



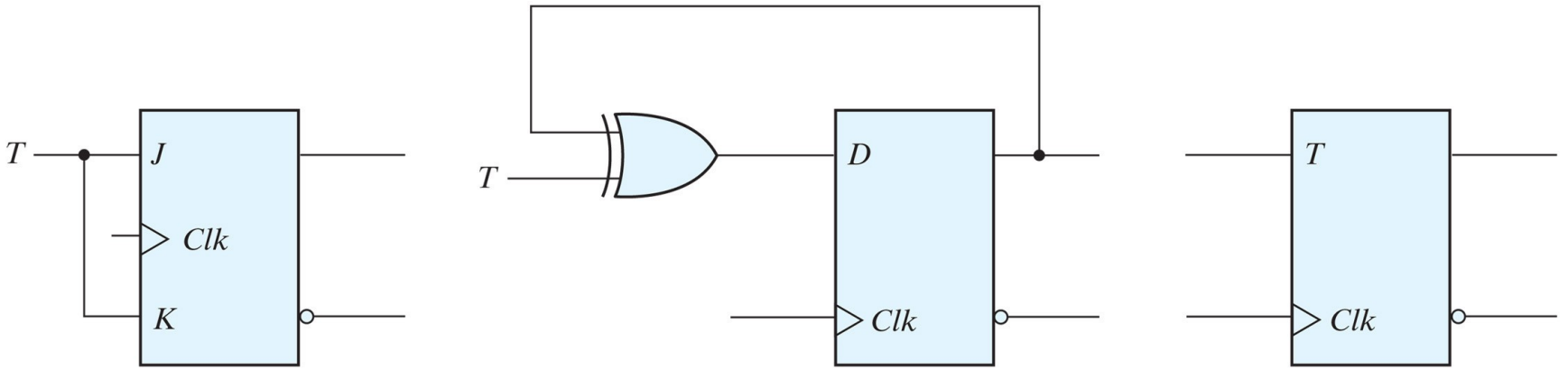
Inputs		Output
EN	D	Q
0	X	Q_0 (no change)
1	0	0
1	1	1

"X" indicates "don't care."
Q_0 is state Q just prior to EN going LOW.



Clocked Flip-Flops

T flip-flop



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

Clocked Flip-Flops

A summary of flip-flop characteristic

Table 5.1
Flip-Flop Characteristic Tables

***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

Clocked Flip-Flops

Inputs that depend on the clock are synchronous.

Most clocked FFs have asynchronous inputs that do not depend on the clock.

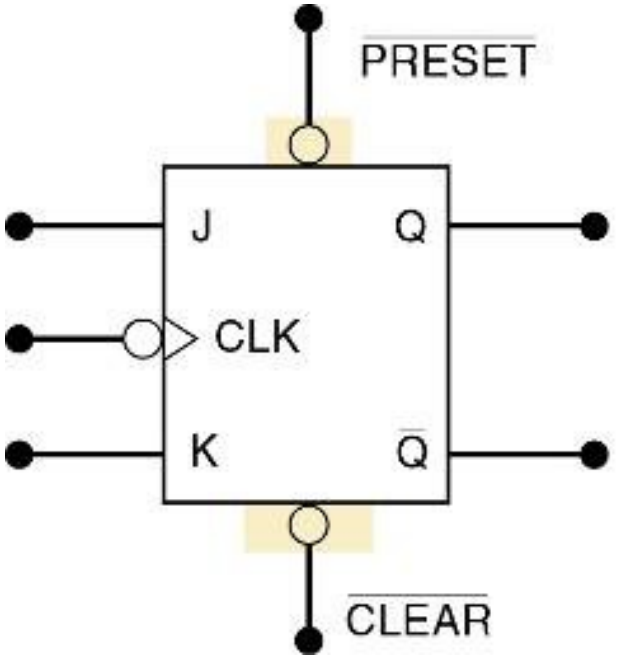
Labels PRE & CLR are used for asynchronous inputs.

Active-LOW asynchronous inputs will have a bar over the labels and inversion bubbles.

If the asynchronous inputs are not used they will be tied to their inactive state.

Clocked Flip-Flops

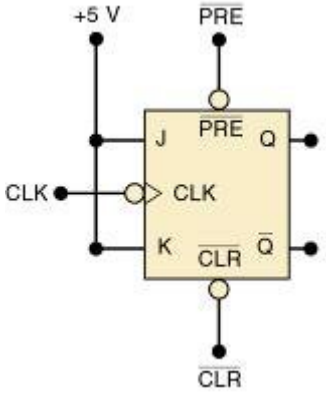
Clocked J-K flip-flop with asynchronous inputs.



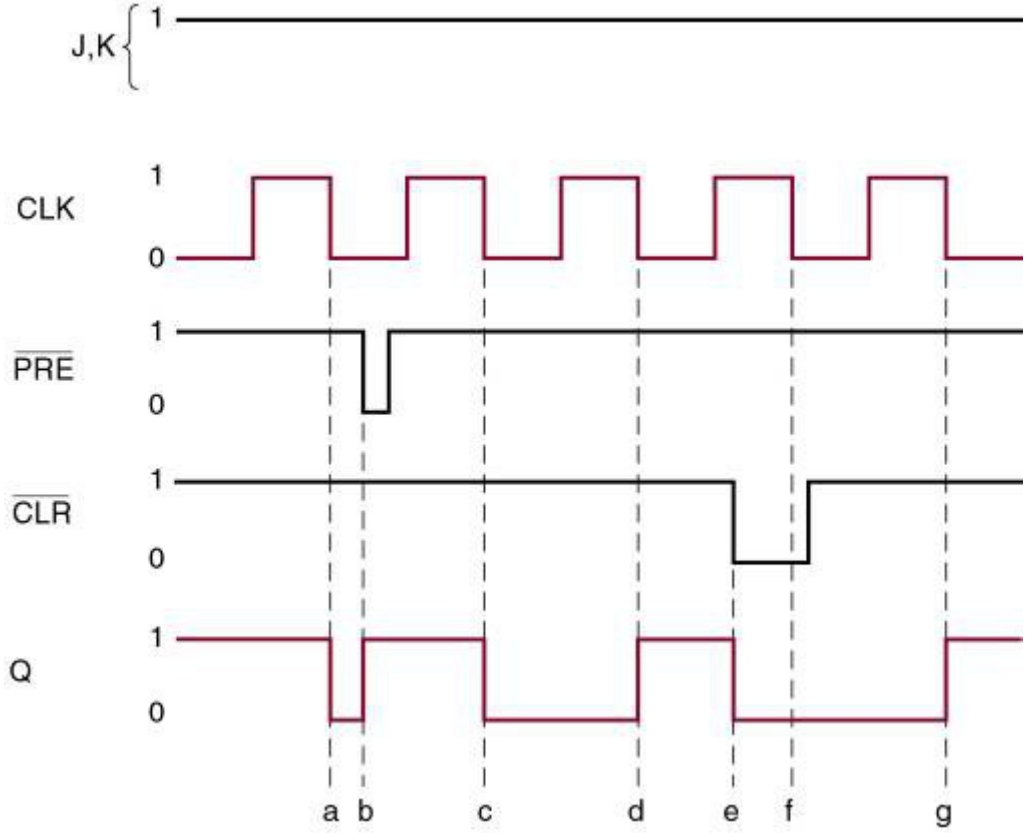
J	K	Clk	$\overline{PRE}$	$\overline{CLR}$	Q
0	0	↓	1	1	Q (no change)
0	1	↓	1	1	0 (Synch reset)
1	0	↓	1	1	1 (Synch set)
1	1	↓	1	1	$\overline{Q}$ (Synch toggle)
x	x	x	1	1	Q (no change)
x	x	x	1	0	0 (asynch clear)
x	x	x	0	1	1 (asynch preset)
x	x	x	0	0	(Invalid)

Clocked Flip-Flops

A J-K FF that responds to a NGT on its clock input and has active-LOW asynchronous inputs.



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



Clocked Flip-Flops

Important timing parameters:

Setup and hold times

Propagation delay—time for a signal at the input to be shown at the output. (t_{PLH} and t_{PHL})

Maximum clocking frequency—Highest clock frequency that will give a reliable output. (f_{MAX})

Clock pulse HIGH and LOW times—minimum clock-time between HIGH/LOW changes.($t_W(L)$; $t_W(H)$)

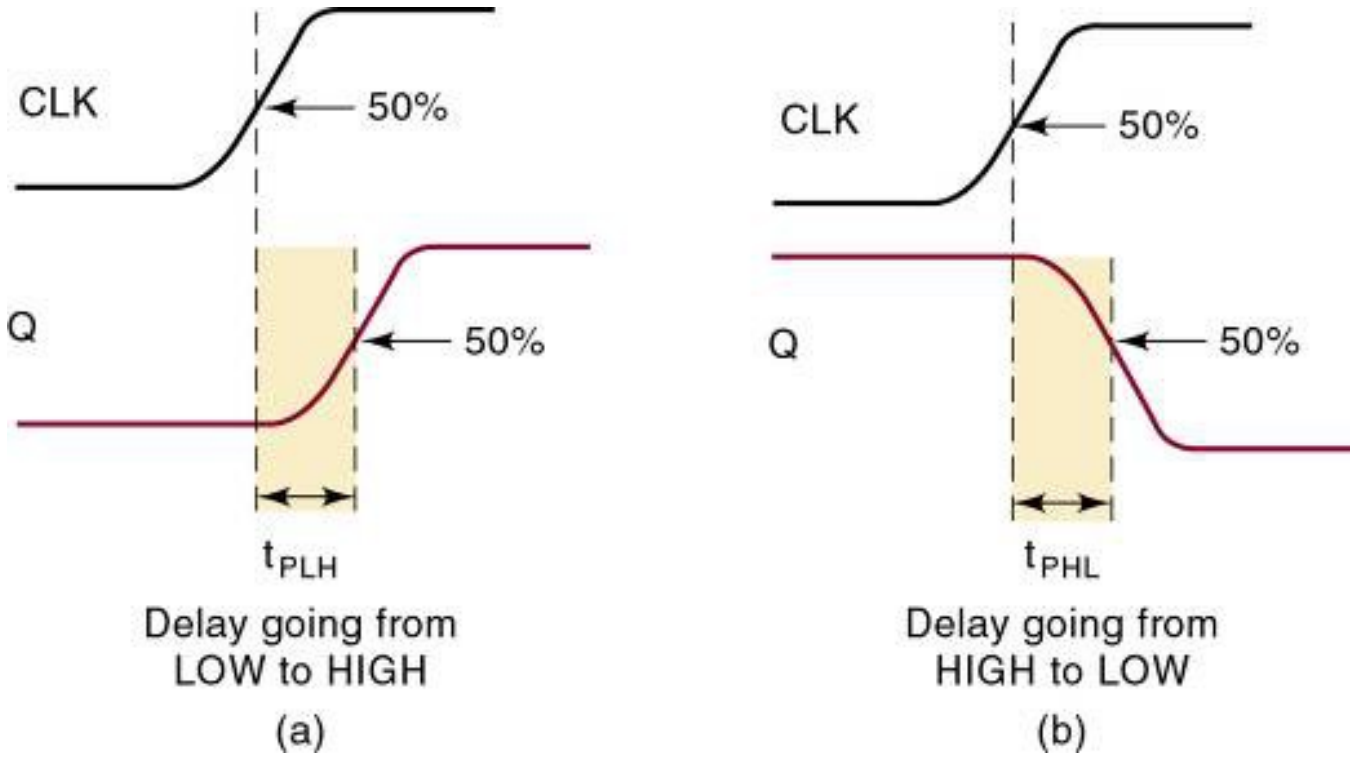
Asynchronous Active Pulse Width—time the clock must HIGH before going LOW, and LOW before going HIGH.

Clock transition times—maximum time for clock transitions,

Less than 50 ns for TTL ; 200 ns for CMOS

Clocked Flip-Flops

FF propagation delays.



Clocked Flip-Flops

When the output of one FF is connected to the input of another FF and both are triggered by the same clock, there is a *potential* timing problem.

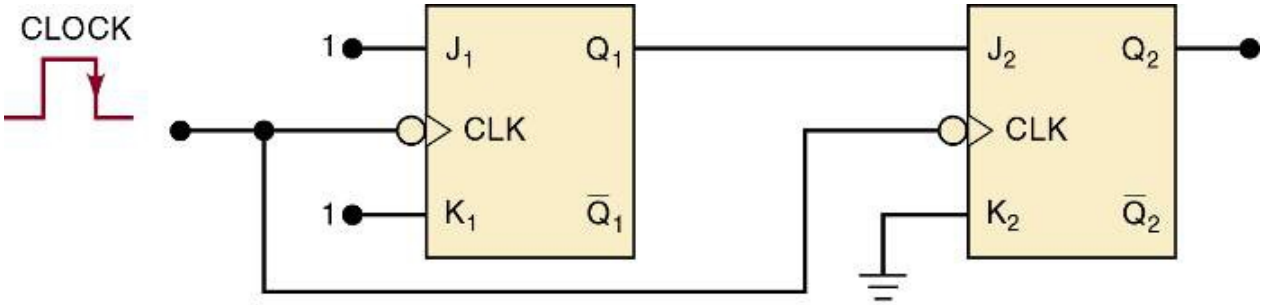
Propagation delay may cause unpredictable outputs.

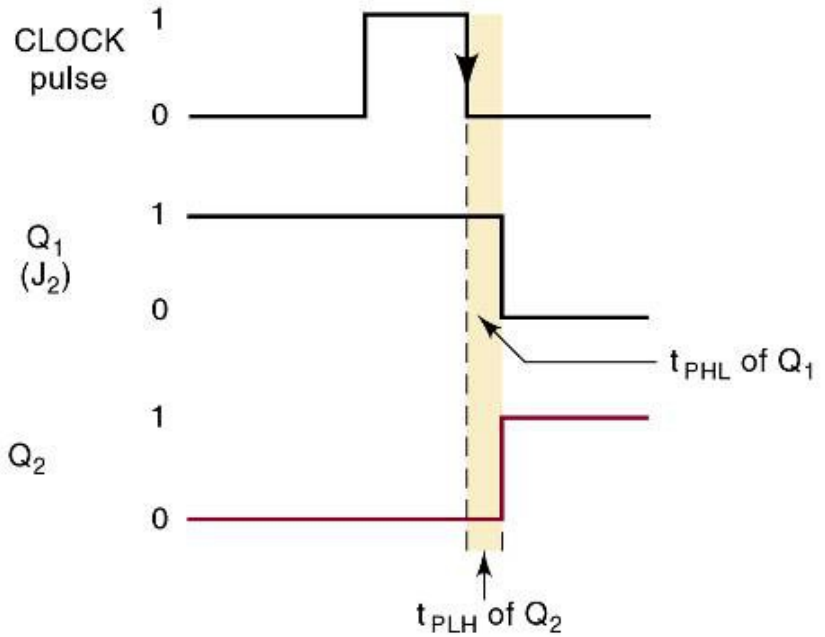
Edge-triggered FFs have hold time requirements 5 ns or less—most have $t_H = 0$.

They have *no* hold time requirement.

Clocked Flip-Flops

*Q*2 will respond properly to the level present at *Q*1 prior to NGT of CLK—*provided* *Q*2 ’s hold time requirement, t_H, is *less* than *Q*1’s propagation delay.





Registers

A flip-flop can store 1-bit of digital information

It is also referred to as a 1-bit register

An array of flip-flops is required to store binary information, the number of flip-flops required being equal to the number of bits in the binary word

Data can be entered in serial or in parallel form

Registers

Shift register: A group of FFs arranged so the binary numbers stored in the FFs are shifted from one FF to the next, for every clock pulse.

Right-shift register: Bits are shifted in the right direction

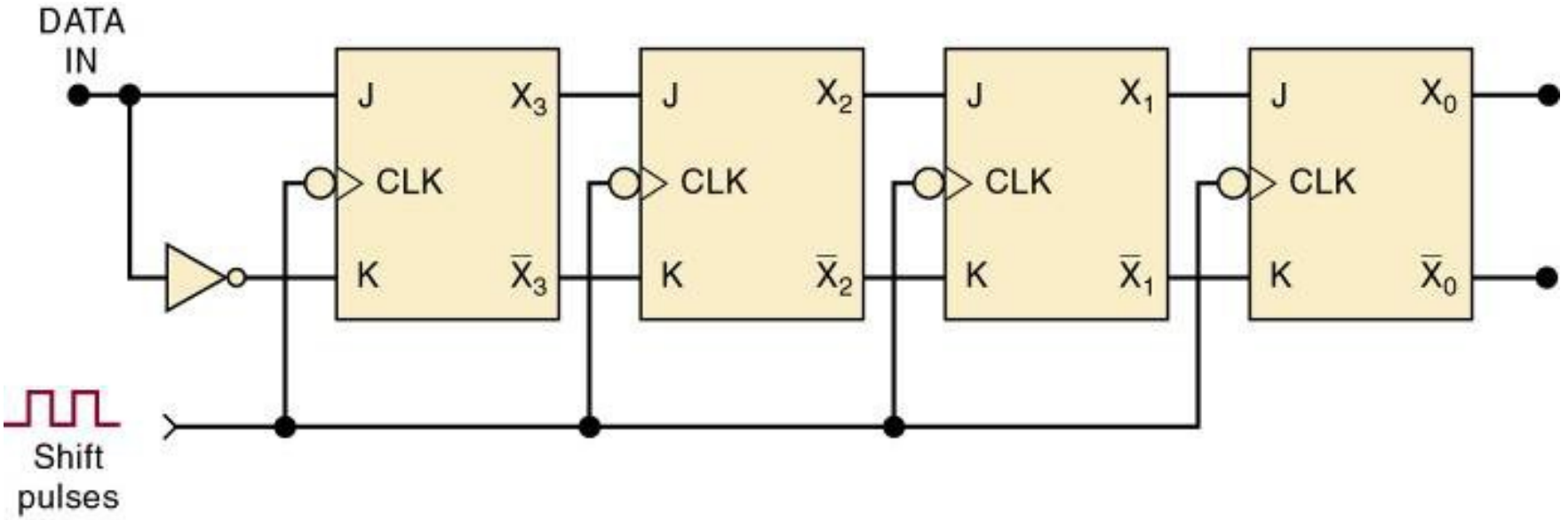
Left-shift register: Bits are shifted in the left direction

Parallel Data: Have one line or wire for each bit in the binary number or data word

Serial Data: Data are fed one bit at a time over only one line at a rate which is constant

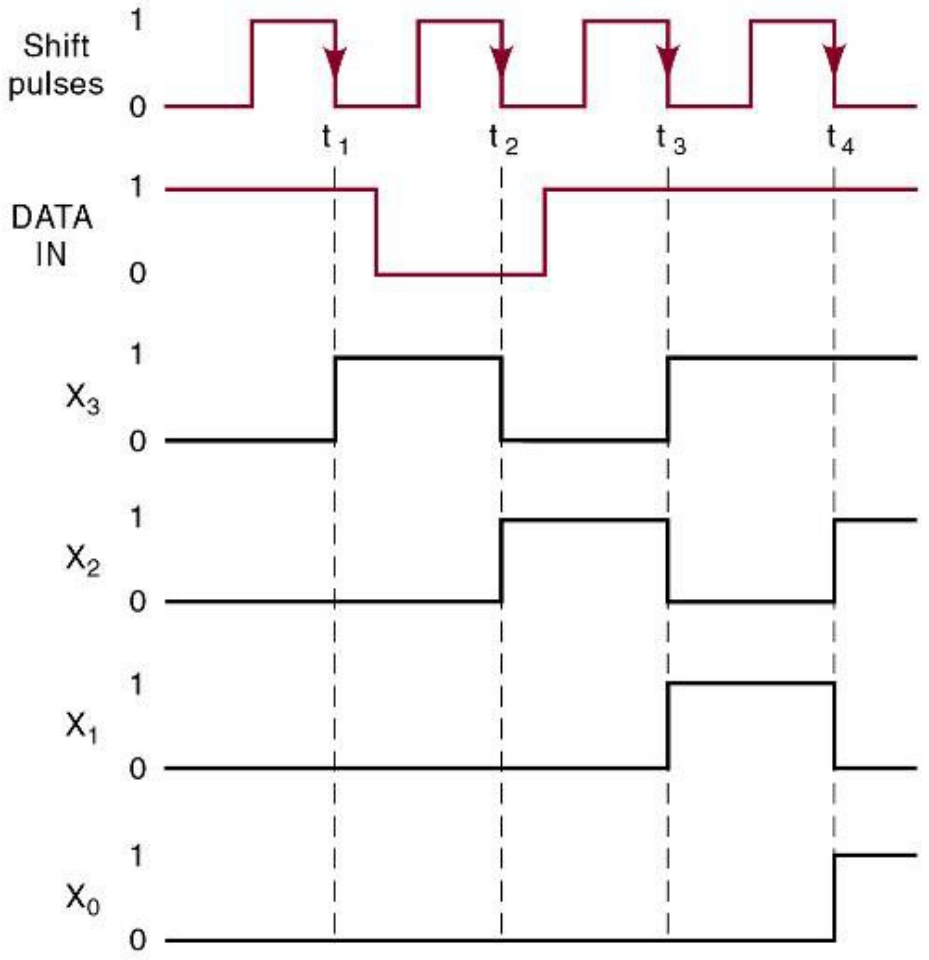
Registers

J-K flip-flops operated as a four-bit shift register.



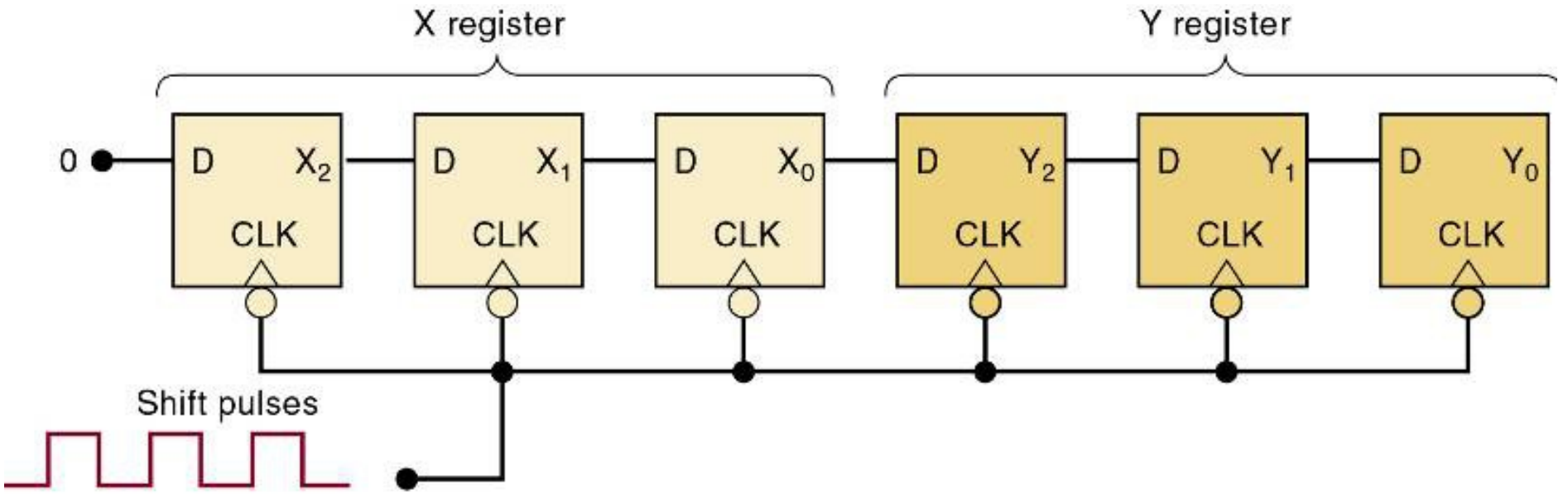
Registers

In this shift-register arrangement, it is necessary to have FFs with very small hold time requirements.



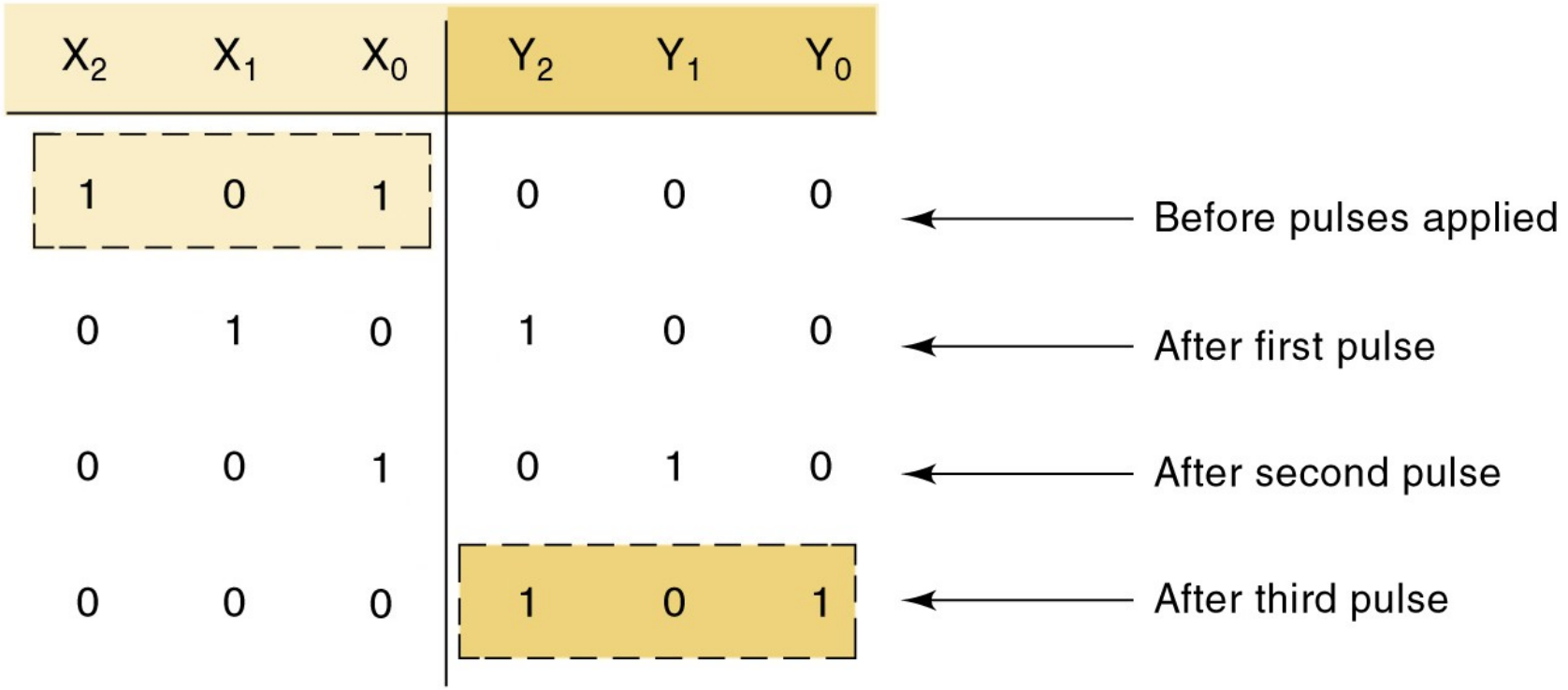
Registers

Two connected three-bit shift registers using D flip flops



Registers

The complete transfer of the three bits of data requires three shift pulses.



X_2	X_1	X_0	Y_2	Y_1	Y_0	
1	0	1	0	0	0	← Before pulses applied
0	1	0	1	0	0	← After first pulse
0	0	1	0	1	0	← After second pulse
0	0	0	1	0	1	← After third pulse

Counters

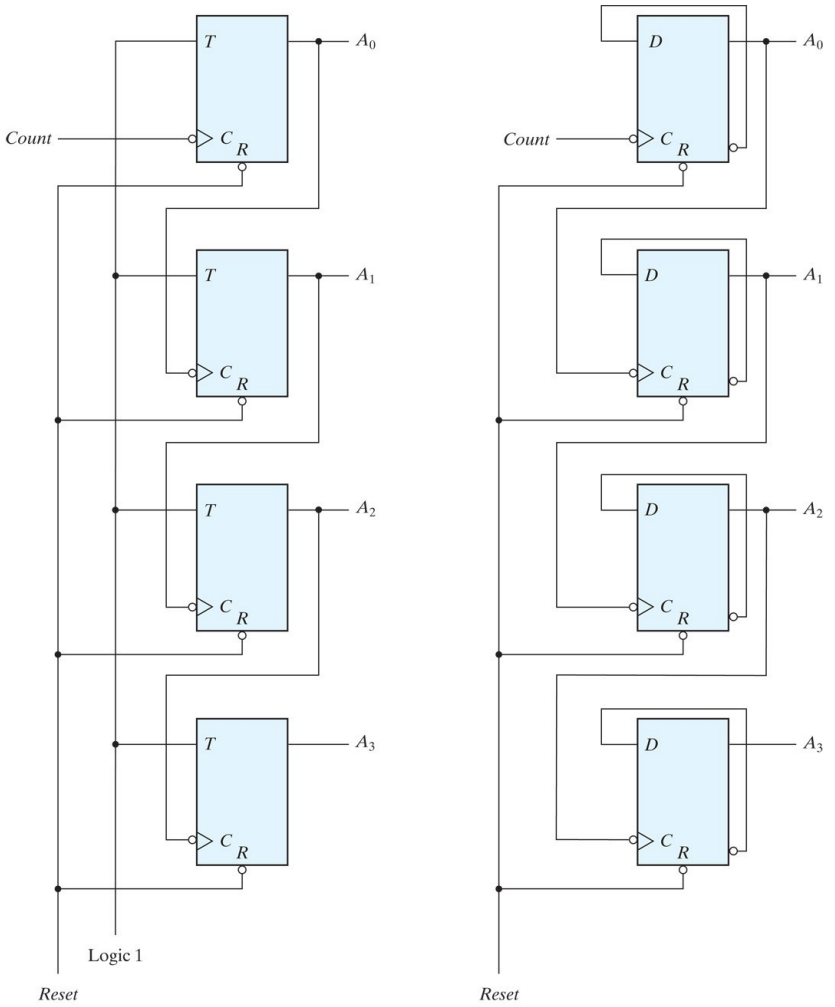
Binary Counter: Count Sequence

Table 6.4
Binary Count Sequence

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

Counters

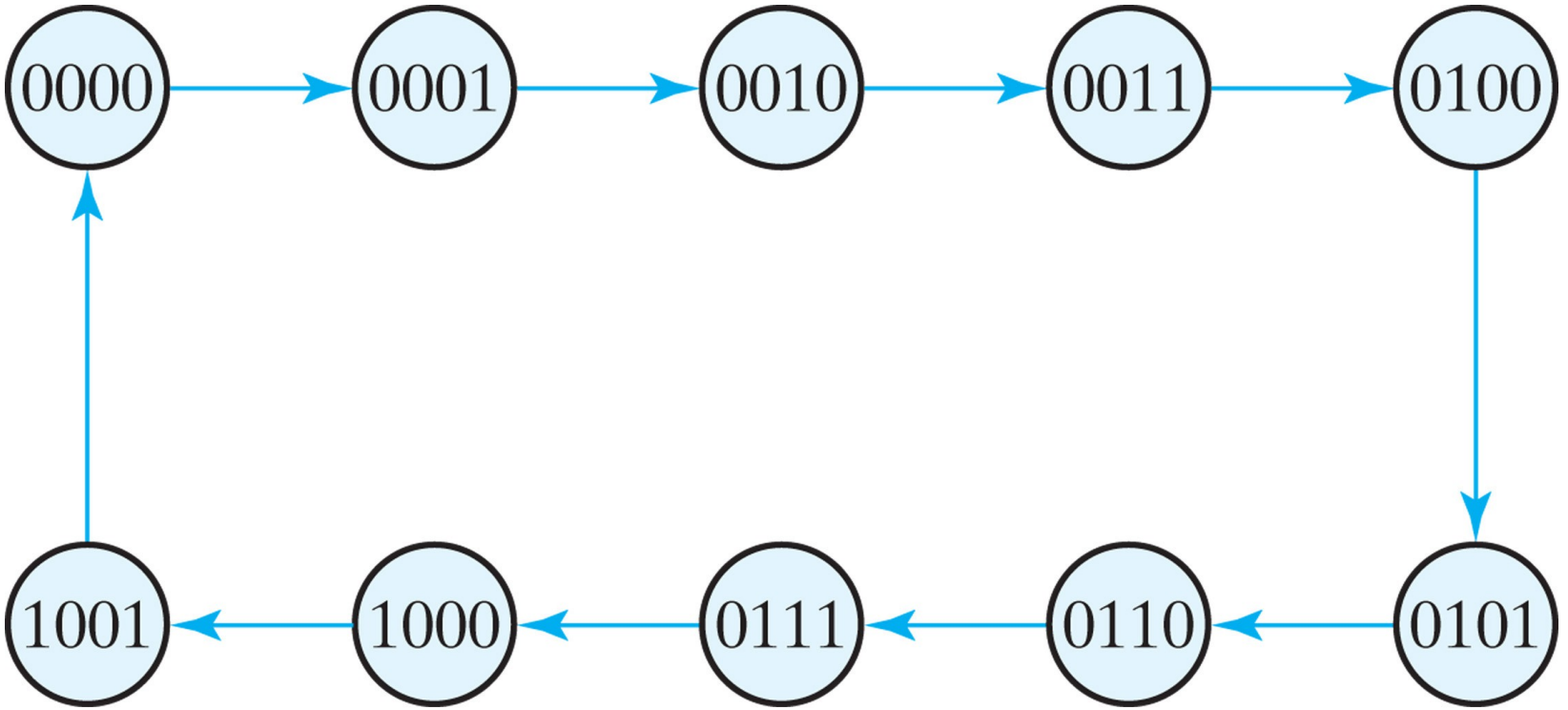
Four-bit binary ripple counter



(a) With *T* flip-flops (b) With *D* flip-flops

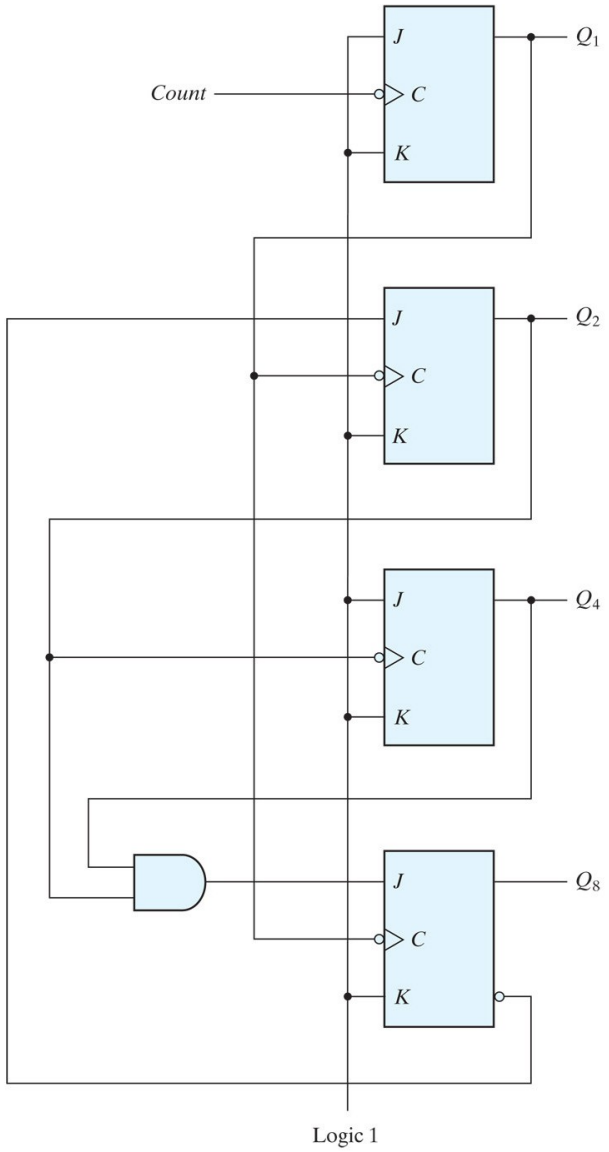
Counters

Decimal BCD counter: The state diagram



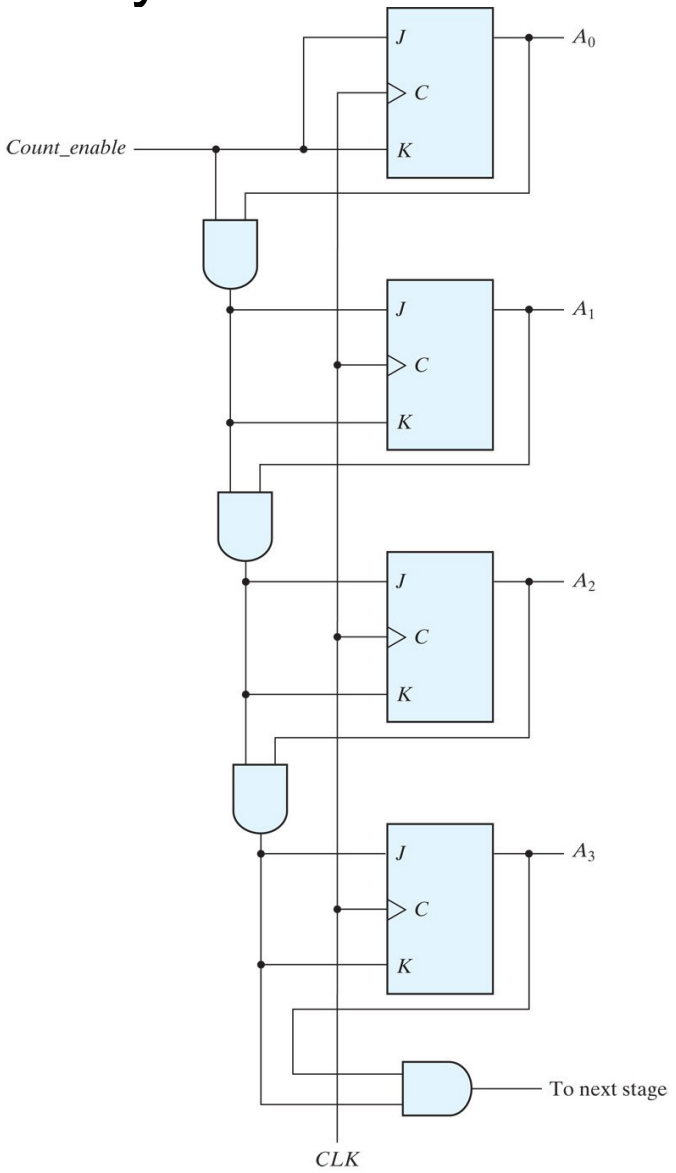
Counters

BCD ripple counter



Counters

Four-bit synchronous binary counter



Counters

Four-bit up–down binary counter