Chapter 5 – Flip-Flops and Related Devices

ELEVENTH EDITION

Digital Systems

Principles and Applications

2 6 12 24 50 100 202

203 406 812 813 1626

1627

1

41

86

42

2 4 10 20 1744

Ronald J. Tocci
Monroe Community College

Neal S. Widmer
Purdue University

Gregory L. Moss
Purdue University

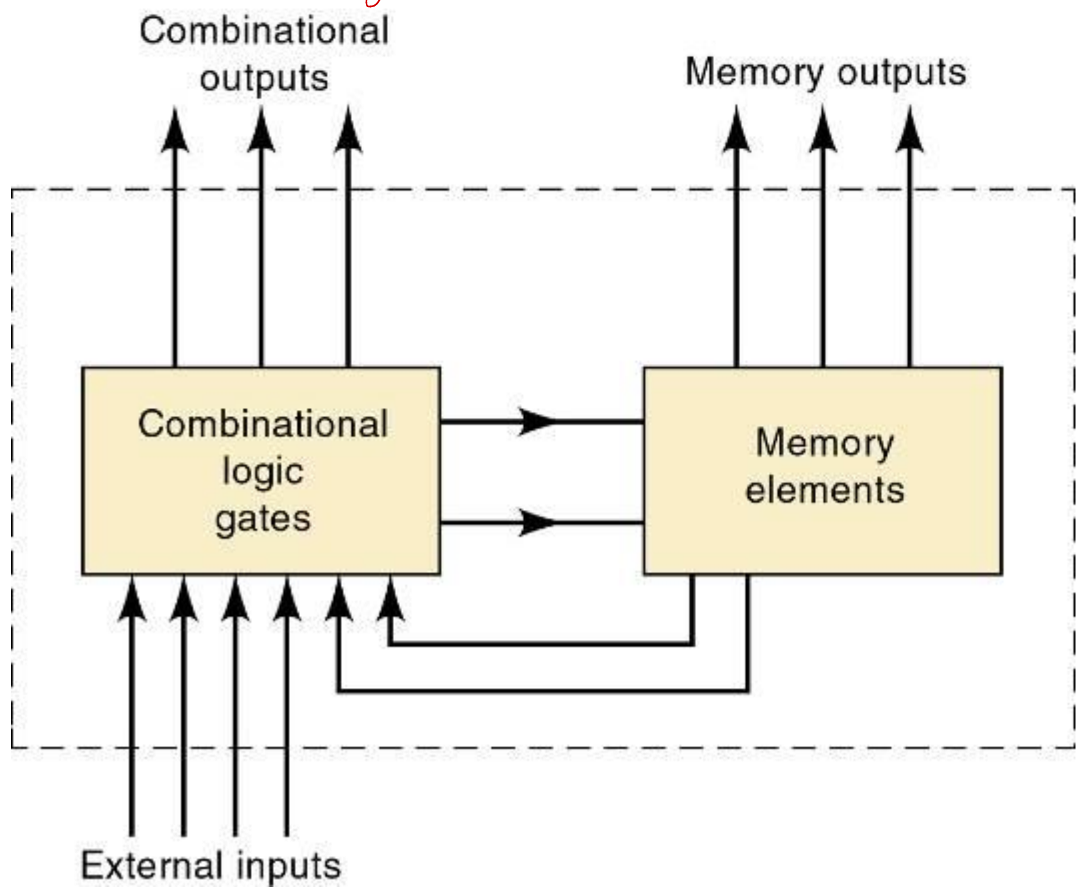
PEARSON

Chapter 5 Objectives

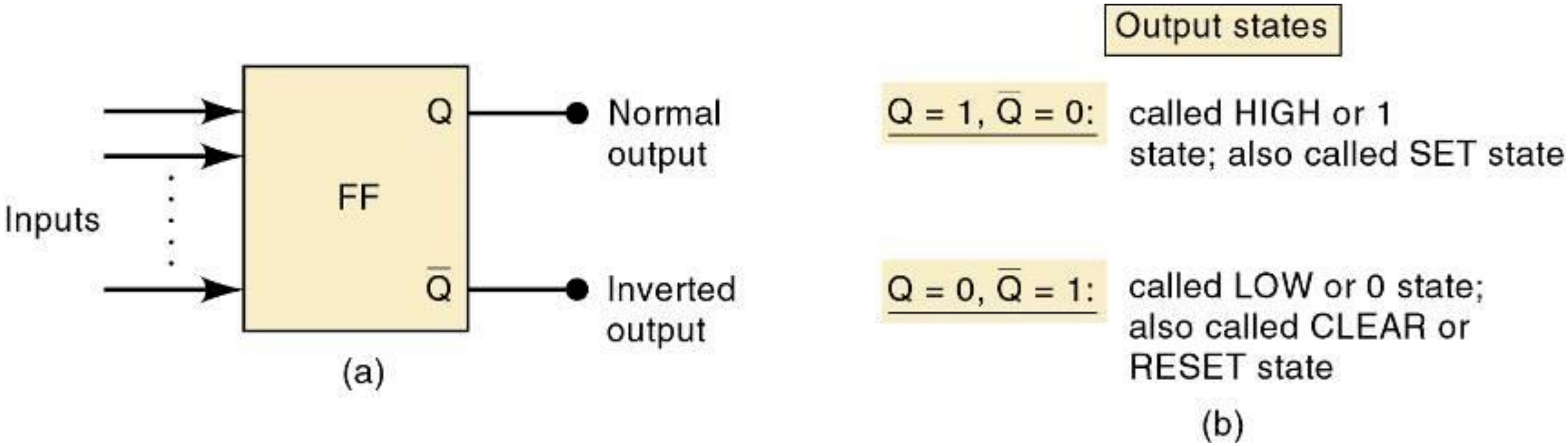
- *Selected areas covered in this chapter*:
  - Constructing/analyzing operation of latch flip-flops made from NAND or NOR gates.
  - Differences of synchronous/asynchronous systems.
  - Major differences between parallel & serial transfers.
  - Operation of edge-triggered flip-flops.
  - Typical characteristics of Schmitt triggers.
  - Effects of clock skew on synchronous circuits.
  - Troubleshoot various types of flip-flop circuits.
  - Sequential circuits with PLDs using schematic entry.
  - Logic primitives, components & libraries in HDL code.
  - Structural level circuits from components.

- Block diagram of a general digital system that combines combinational logic gates with memory devices.

รวม logic gates ไว้กับ memory devices



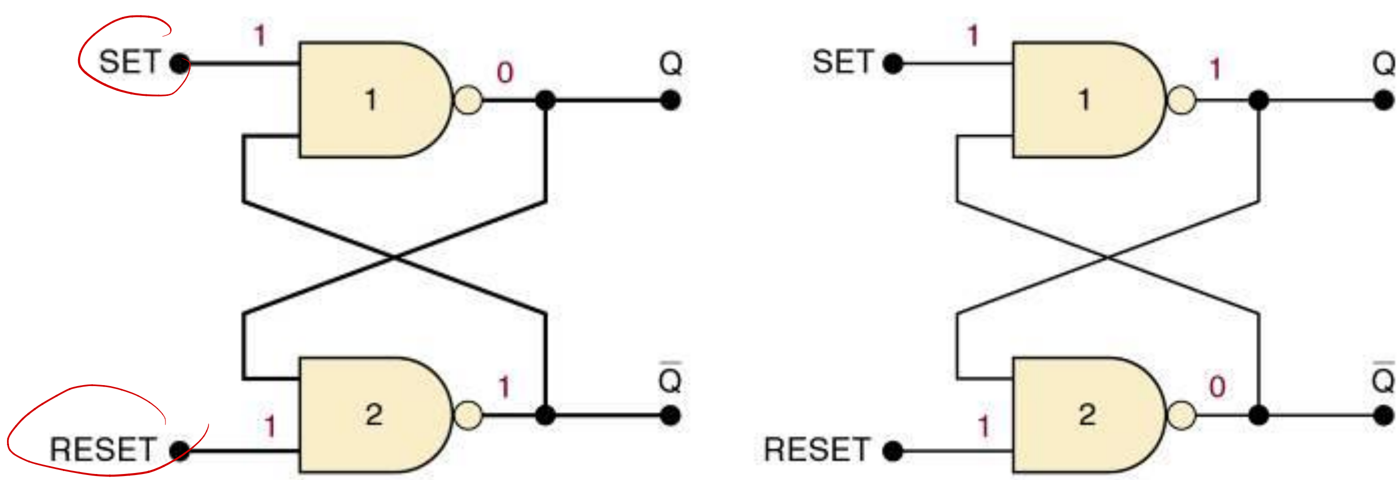
- The most important memory element is the **flip-flop** (FF)—made up of an assembly of logic gates.



The flip-flop is known by other names, including *latch* and *bistable multivibrator*.

5-1 NAND Gate Latch

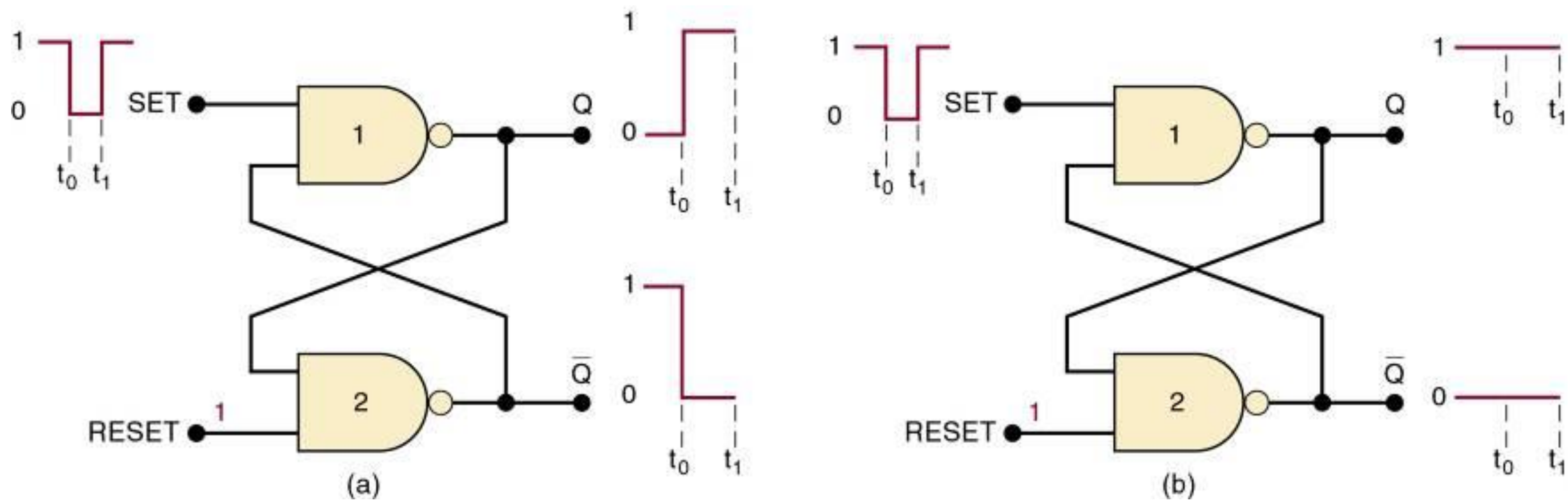
- The **NAND** gate latch or simply latch is a basic FF.
  - Inputs are *SET* and *CLEAR* (*RESET*).
- Inputs are active-LOW—output will change when the input is pulsed LOW.
  - When the latch is set: **Q** = 1 and $\overline{\mathbf{Q}}$ = 0
  - When the latch is clear or reset: **Q** = 0 and $\overline{\mathbf{Q}}$ = 1



Sync
অন্য Clock

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

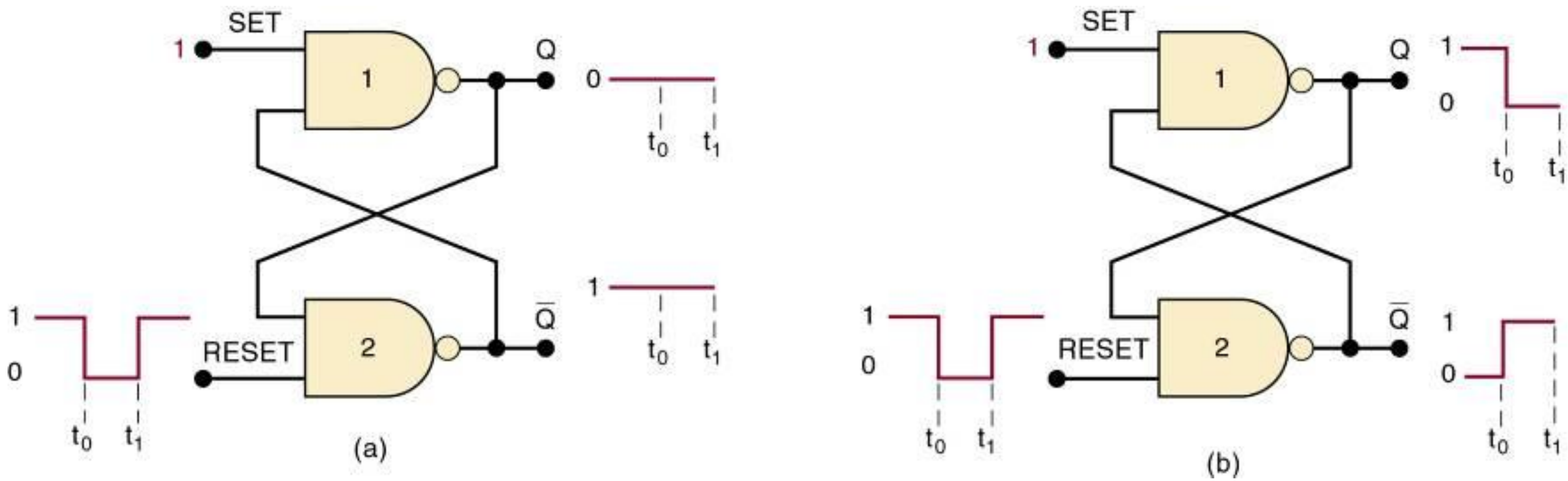
ก่อน สัญญาณ



In both cases, *Q* ends up HIGH.

5-1 NAND Gate Latch – Resetting the Latch (FF)

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

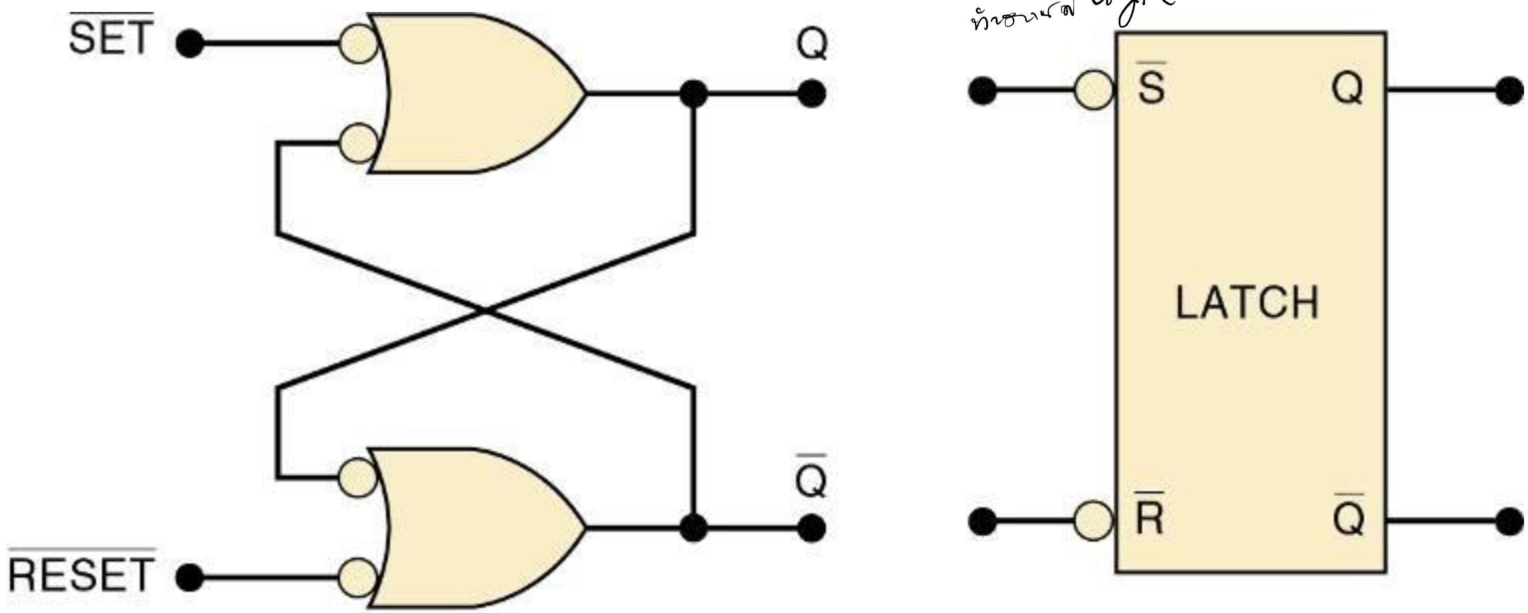


In each case, *Q* ends up LOW.

5-1 NAND Gate Latch – Alternate Representations

NAND latch equivalent representations and simplified block diagram.

ทำงานที่ logic 0

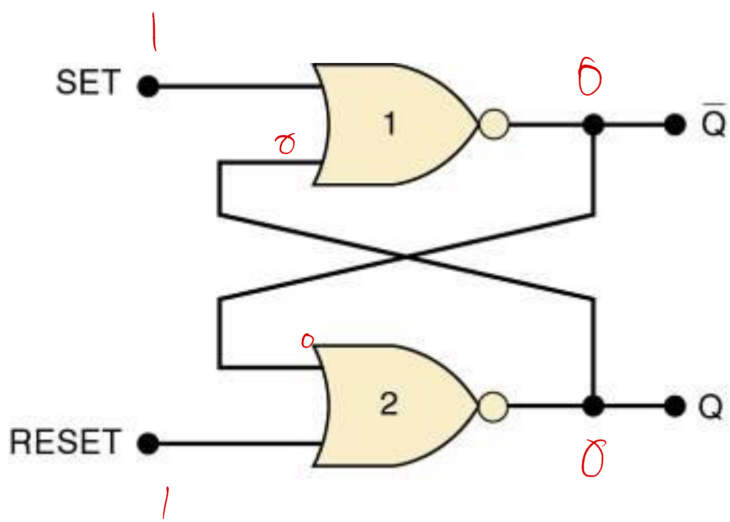


- Summary of the **NAND** latch:
  - **SET = 1, RESET = 1**—Normal resting state, outputs remain in state they were in prior to input.
  - **SET = 0, RESET = 1**—Output will go to Q = 1 and remains there, even after SET returns HIGH.
    - Called *setting* the latch.
  - **SET = 1, RESET = 0**—Will produce 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 $\boxed{Q = \overline{Q} = 1}$. ไม่รู้
    - Output is unpredictable, and this input condition should not be used.

SET	RESET	Q	$\overline{Q}$	
0	0	1	1	ไม่รู้
0	1	1	0	
1	0	0	1	
1	1			ไม่เปลี่ยน

5-2 NOR Gate Latch

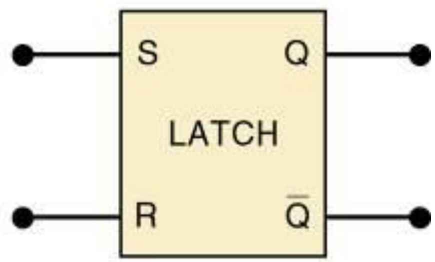
- Two cross-coupled **NOR** gates can be used as a **NOR** gate latch—similar to the **NAND** latch.
  - The **Q** and $\overline{\mathbf{Q}}$ outputs are reversed.



$\overline{Q} \neq Q$

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

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



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

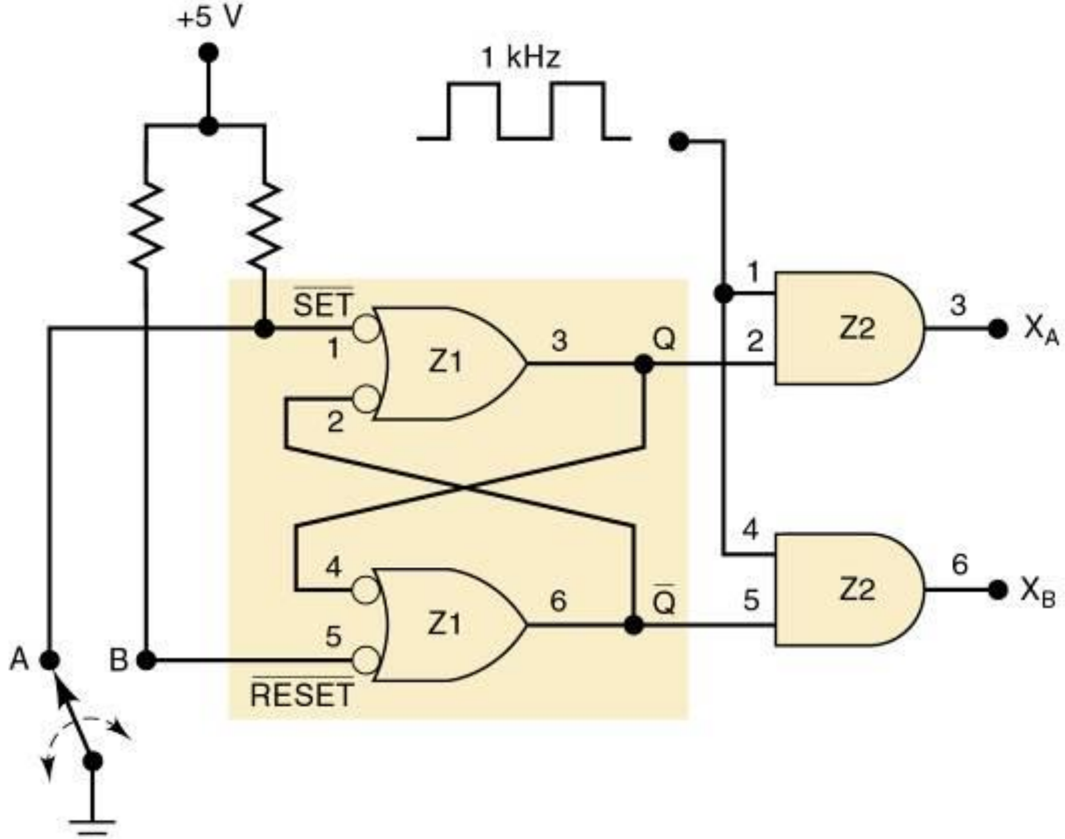
5-1 NOR Gate Latch - Summary

- 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 $Q = \overline{Q} = 0$.
    - Output is unpredictable, and this input condition should not be used.

- When power is applied, it is not possible to predict the starting state of a flip-flop's output.
  - If SET and RESET inputs are in their inactive state.
- To start a latch or FF in a particular state, it must be *placed* in that state by momentarily activating the SET or RESET input, at the start of operation.
  - Often achieved by application of a pulse to the appropriate input.

5-3 Troubleshooting Case Study

Switch Position	$\overline{SET}$ (Z1-1)	$\overline{RESET}$ (Z1-5)	Q (Z1-3)	$\overline{Q}$ (Z1-6)	X_A (Z2-3)	X_B (Z2-6)
A	LOW	HIGH	LOW	HIGH	LOW	Pulses
B	HIGH	LOW	LOW	HIGH	LOW	Pulses



Troubleshoot the circuit.

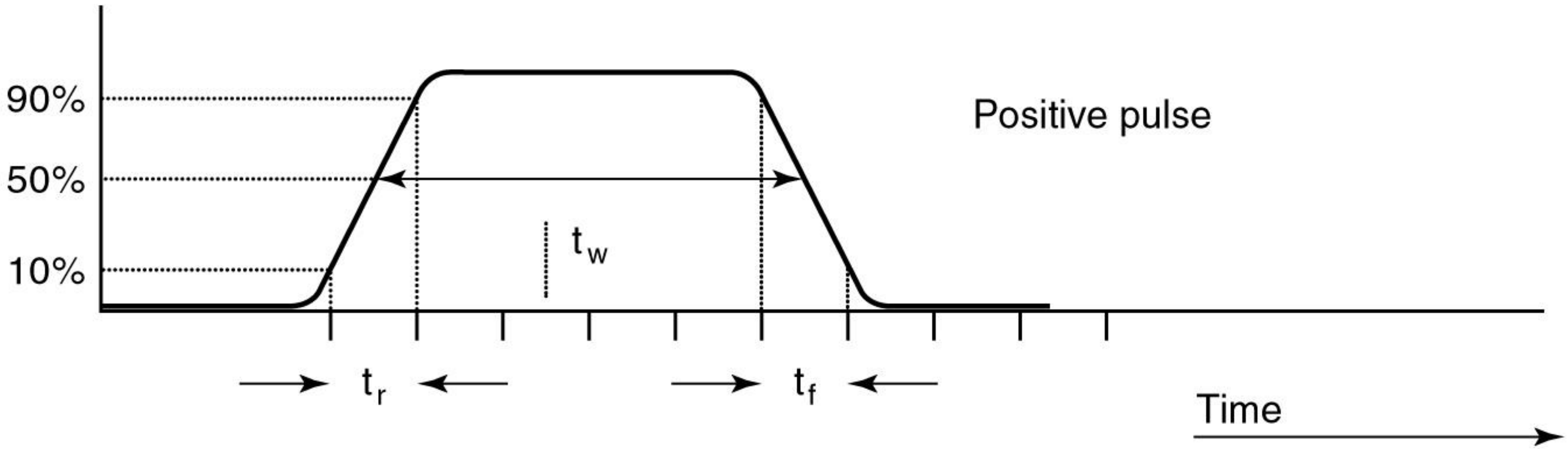
Switch position	X_A	X_B
A	Pulses	LOW
B	LOW	Pulses

5-3 Troubleshooting Case Study

Switch Position	$\overline{\text{SET}}$ (Z1-1)	$\overline{\text{RESET}}$ (Z1-5)	Q (Z1-3)	$\overline{Q}$ (Z1-6)	X_A (Z2-3)	X_B (Z2-6)
A	LOW	HIGH	LOW	HIGH	LOW	Pulses
B	HIGH	LOW	LOW	HIGH	LOW	Pulses

- There are several possibilities:
  - An internal open connection at Z1-1, which would prevent *Q* from responding to the input.
  - An internal component failure in NAND gate Z1 that prevents it from responding properly.
  - *Q* output is stuck LOW, which could be caused by:
    - Z1-3 internally shorted to ground
    - Z1-4 internally shorted to ground
    - Z2-2 internally shorted to ground
    - The *Q* node externally shorted to ground

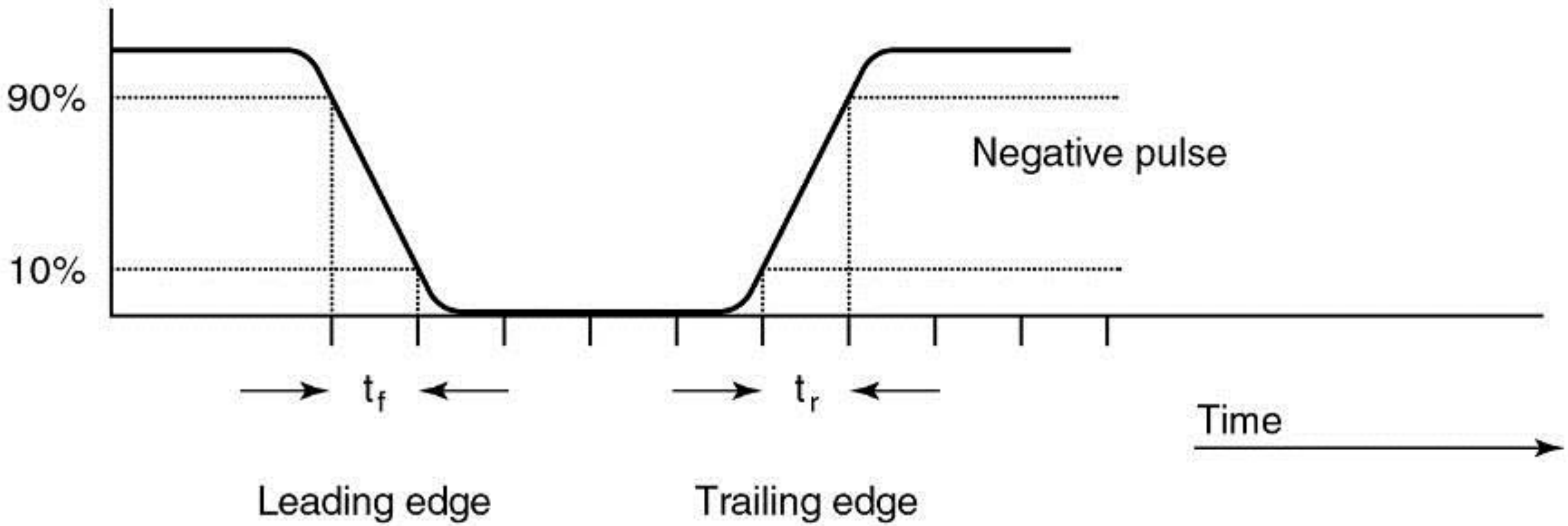
Signals that switch between active and inactive states are called pulse waveforms.



A positive pulse has an active-HIGH level.

5-4 Digital Pulses

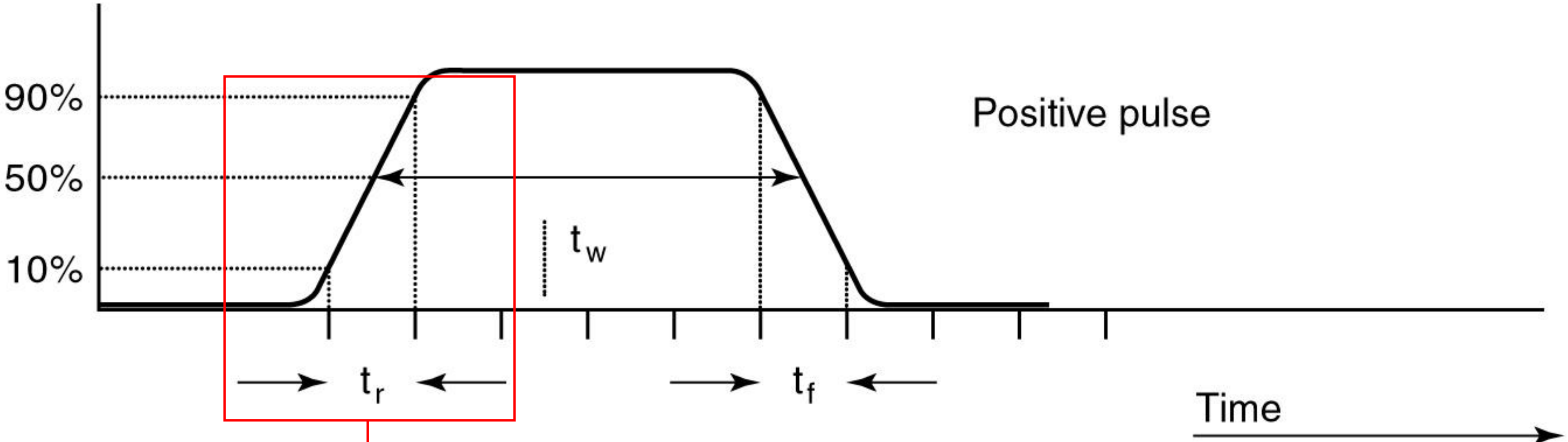
Signals that switch between active and inactive states are called pulse waveforms.



A negative pulse has an active-LOW level.

5-4 Digital Pulses

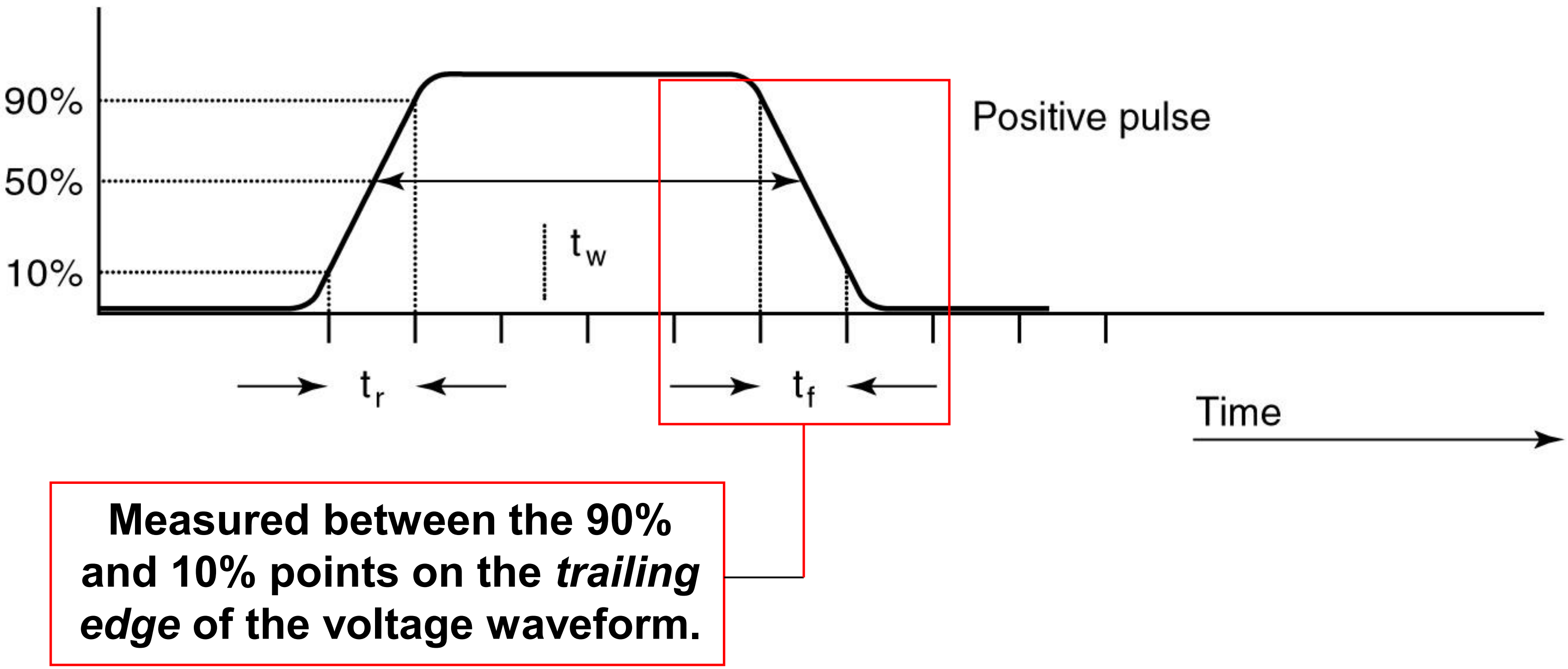
- In actual circuits it takes time for a pulse waveform to change from one level to the other.
  - Transition from LOW to HIGH on a positive pulse is called *rise time* (t_r).



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

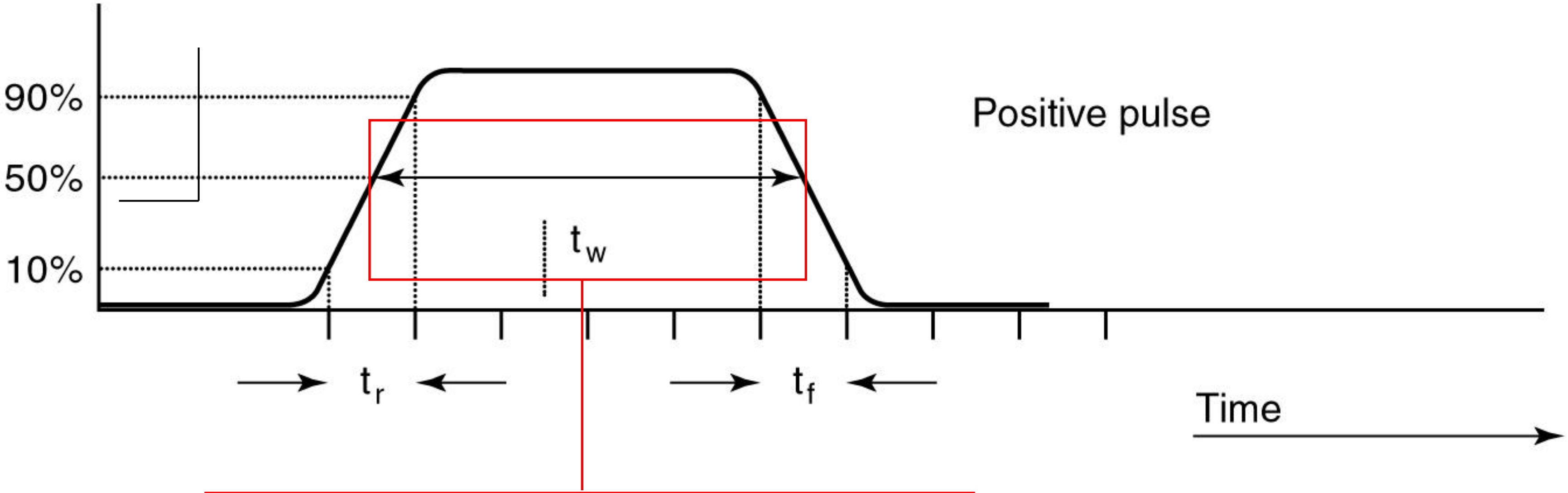
5-4 Digital Pulses

- In actual circuits it takes time for a pulse waveform to change from one level to the other.
  - Transition from HIGH to LOW on a positive pulse is called *fall time* (t_f).



5-4 Digital Pulses

- In actual circuits it takes time for a pulse waveform to change from one level to the other.
  - 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.

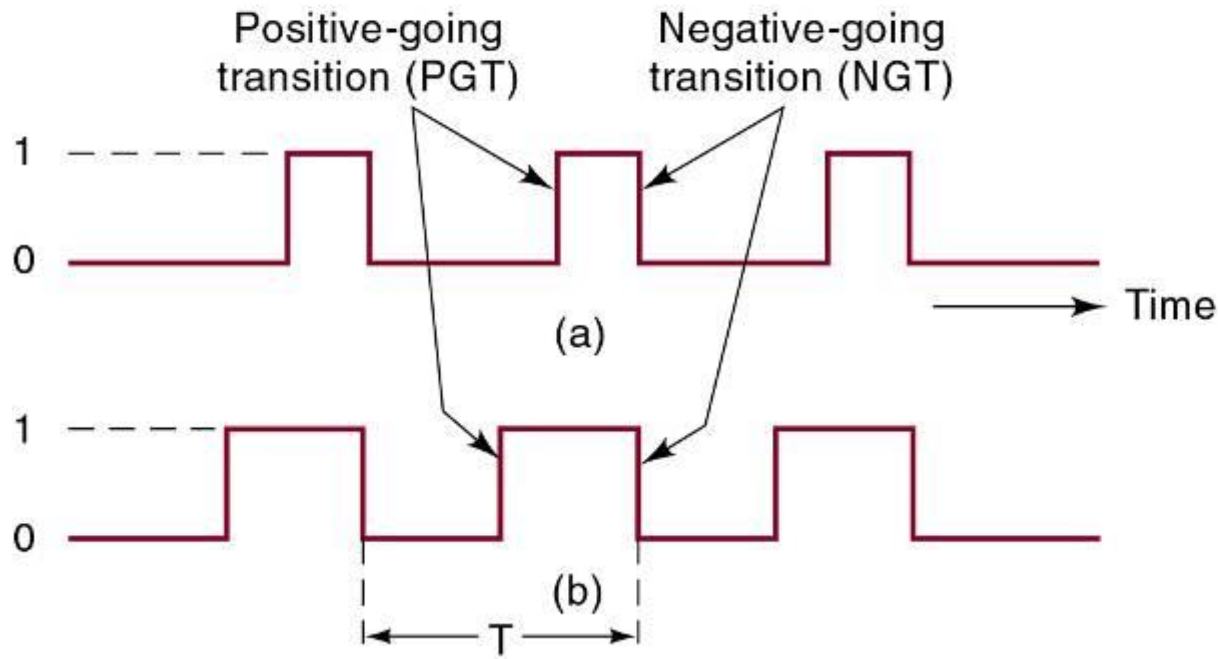
5-5 Clock Signals and Clocked Flip-Flops

- Digital systems can operate either *asynchronously* or *synchronously.*
  - **Asynchronous system**—outputs can change state at any time the input(s) change.
  - **Synchronous system**—output can change state only at a specific time in the clock cycle.

5-5 Clock Signals and Clocked Flip-Flops

- The clock signal is a rectangular pulse train or square wave.
  - Positive going transition (PGT)—clock pulse goes from 0 to 1.
  - Negative going transition (NGT)—clock pulse goes from 1 to 0.

Transitions are also called *edges*.

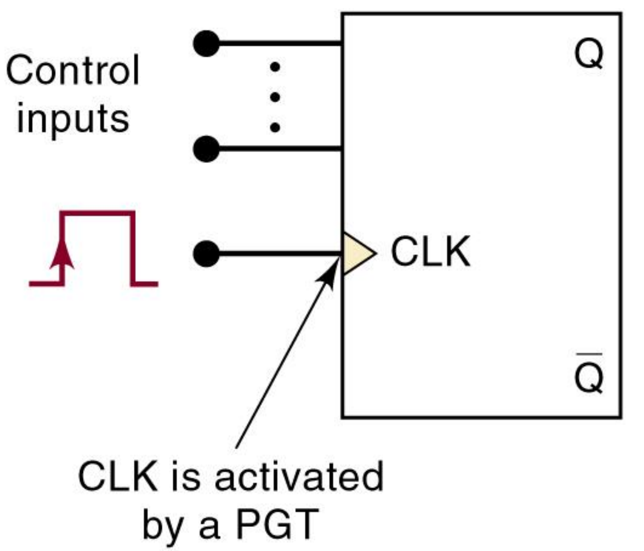


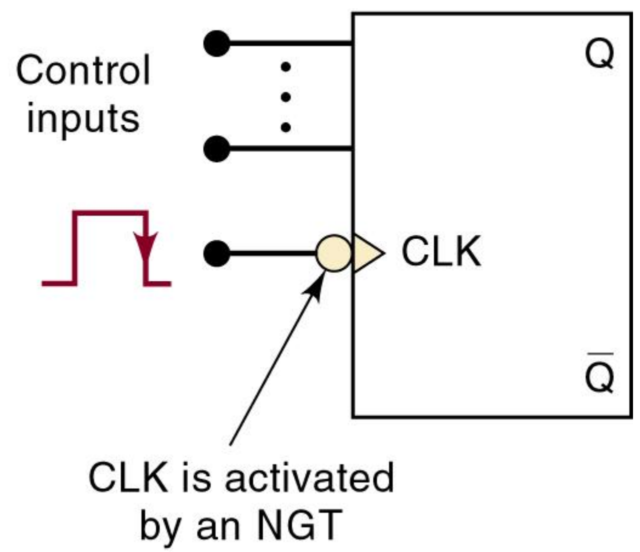
5-5 Clock Signals and Clocked Flip-Flops

- Clocked FFs change state on one or the other clock transitions.
  - Clock inputs are labeled CLK, CK, or CP.

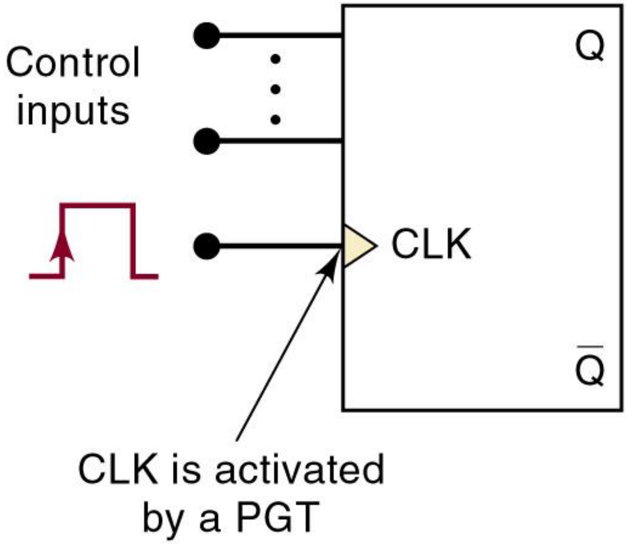
A small triangle at the CLK input indicates that the input is activated with a PGT.

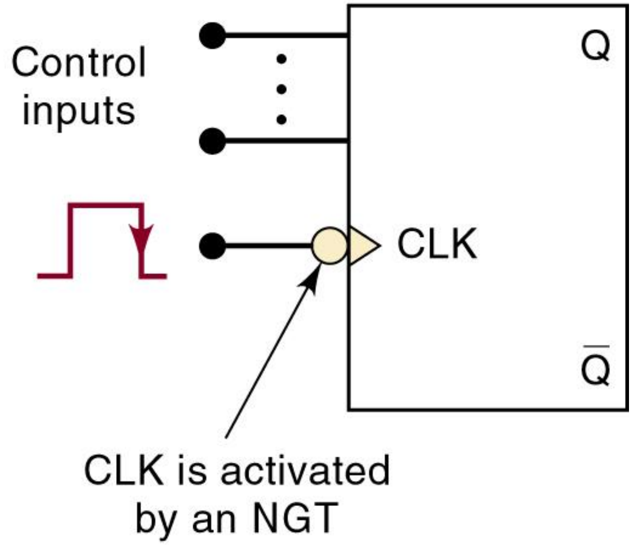
A bubble and a triangle indicates that the CLK input is activated with a NGT.



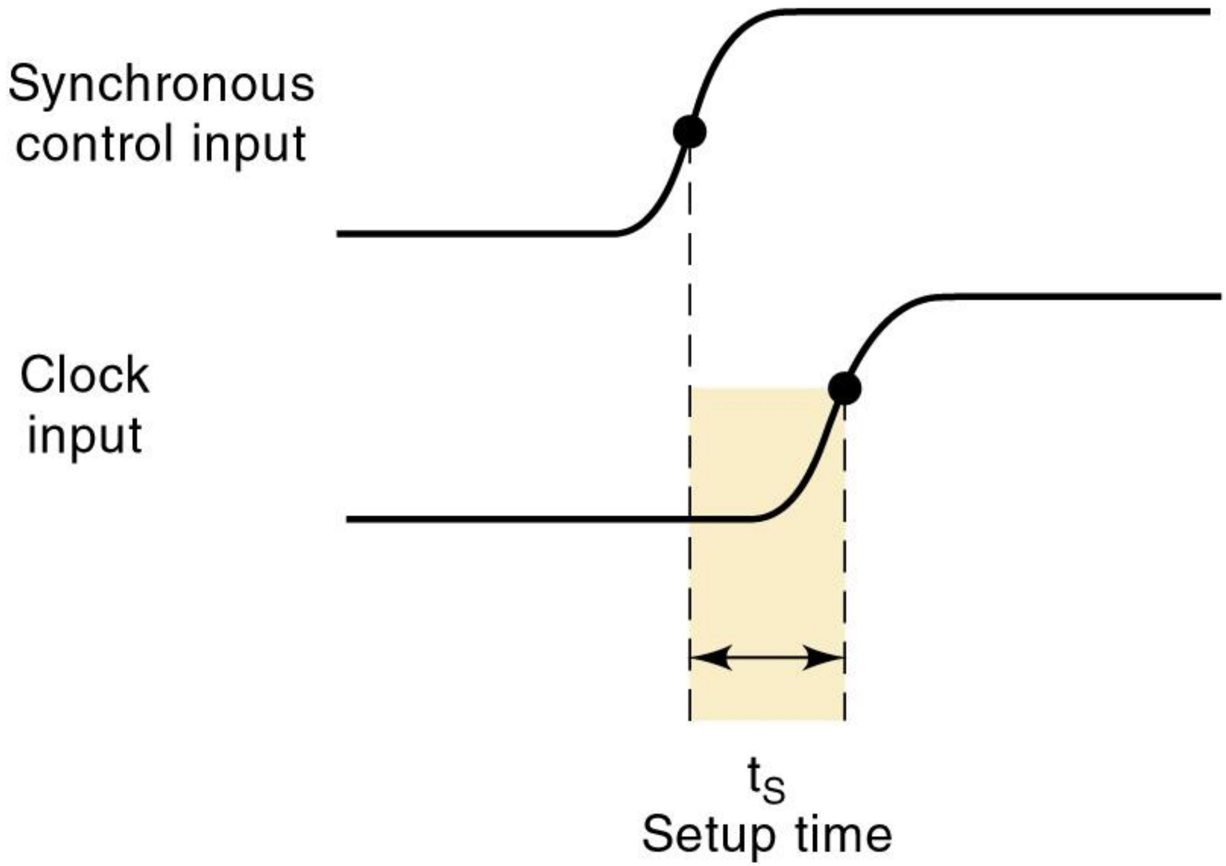


- Control inputs have an effect on the output only at the active clock transition (NGT or PGT)—also called synchronous control inputs.
  - The control inputs get the outputs ready to change, but the change is not triggered until the CLK edge.

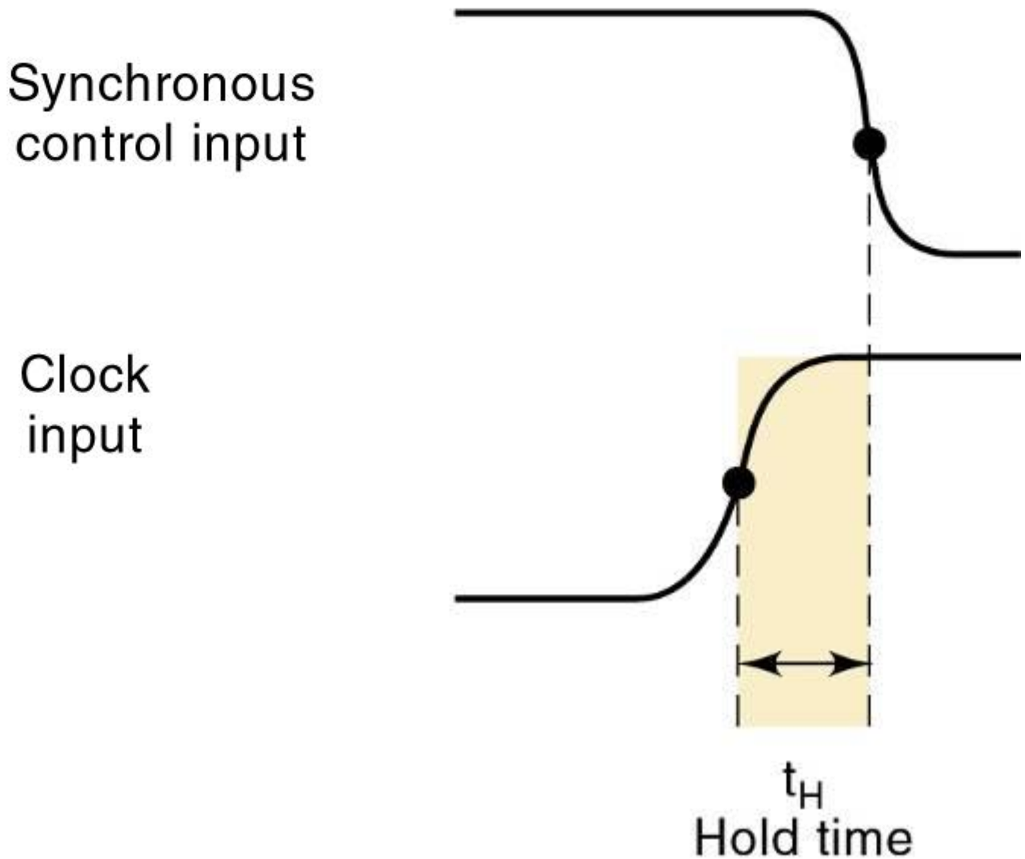




- *Setup time* (t_S) is the minimum time interval before the active CLK transition that the control input must be kept at the proper level.



- *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.

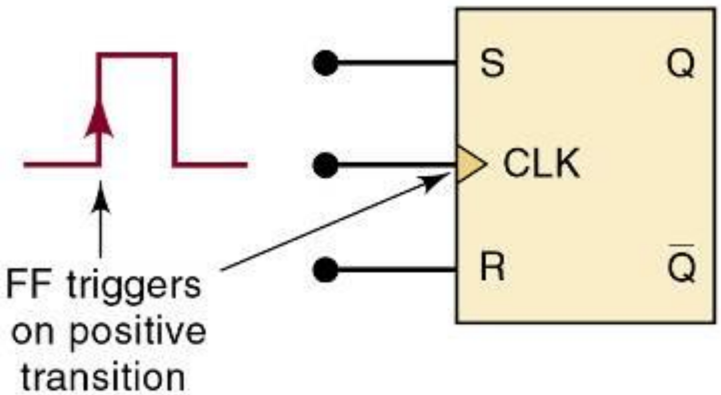


5-6 Clocked S-R Flip-Flop

- The *S* and *R* inputs are synchronous *control* inputs, which control the state the FF will go to when the clock pulse occurs.
  - The *CLK* input is the **trigger** input that causes the FF to change states according to the *S* and *R* inputs.
- SET-RESET (or SET-CLEAR) FF will change states at positive- or negative-going clock edges.

5-6 Clocked S-R Flip-Flop

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



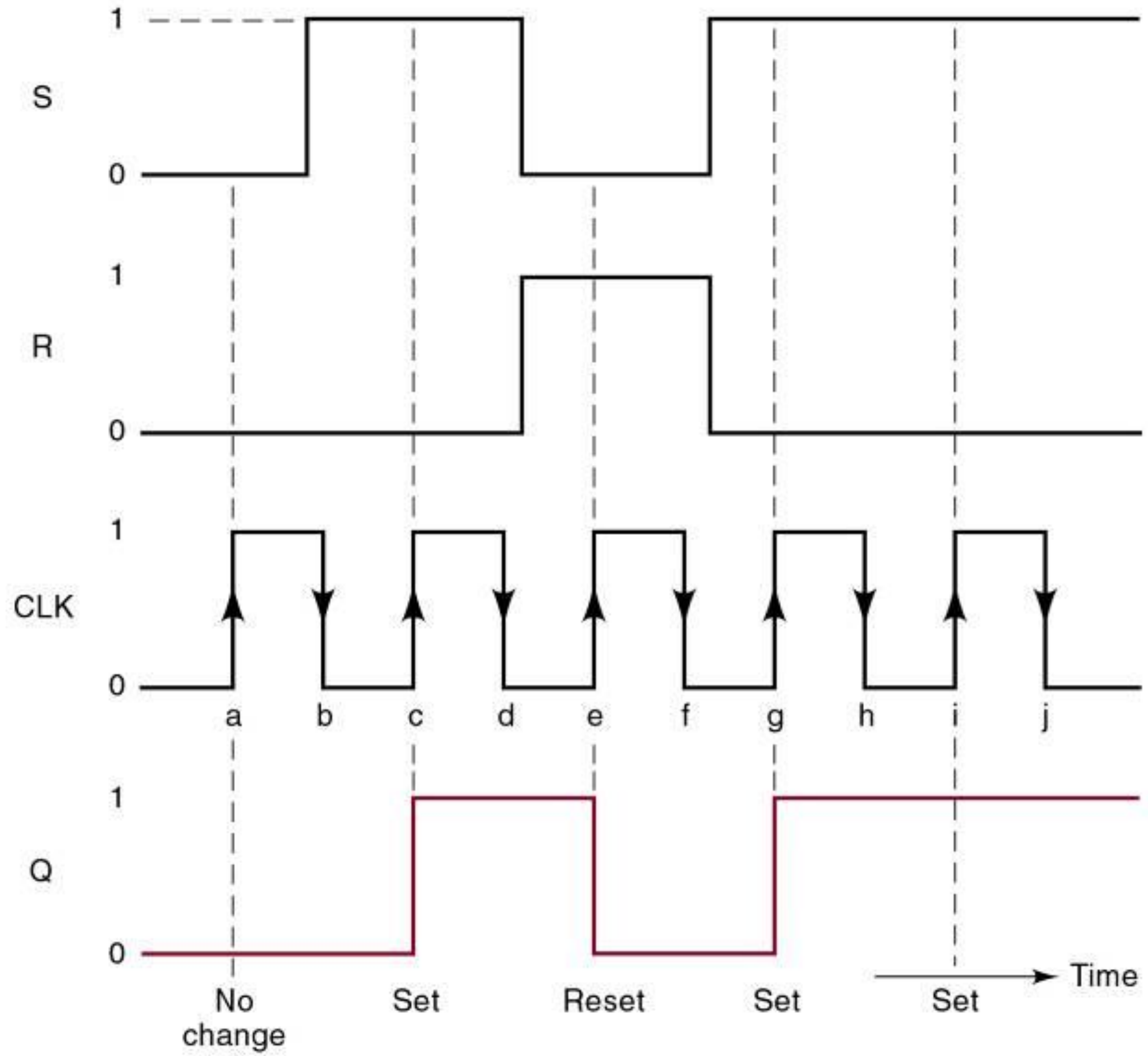
Inputs			Output
S	R	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambiguous

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

The *S* and *R* inputs control the state of the FF in the same manner as described earlier for the NOR gate latch, but the FF does *not* respond to these inputs *until* the occurrence of the PGT of the clock signal.

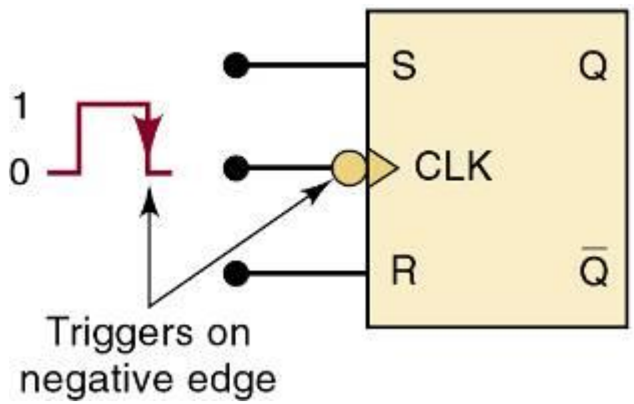
5-6 Clocked S-R Flip-Flop

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



5-6 Clocked S-R Flip-Flop

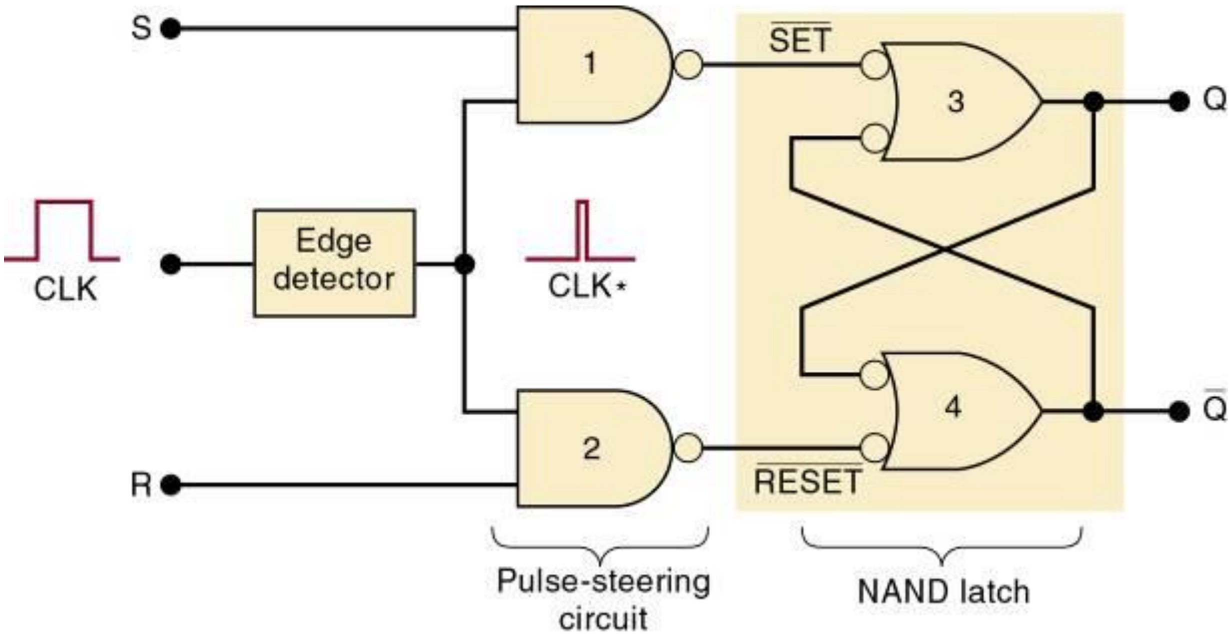
A clocked S-R flip-flop triggered by the negative-going edge of the clock signal.



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

Both positive-edge and negative-edge triggering FFs are used in digital systems.

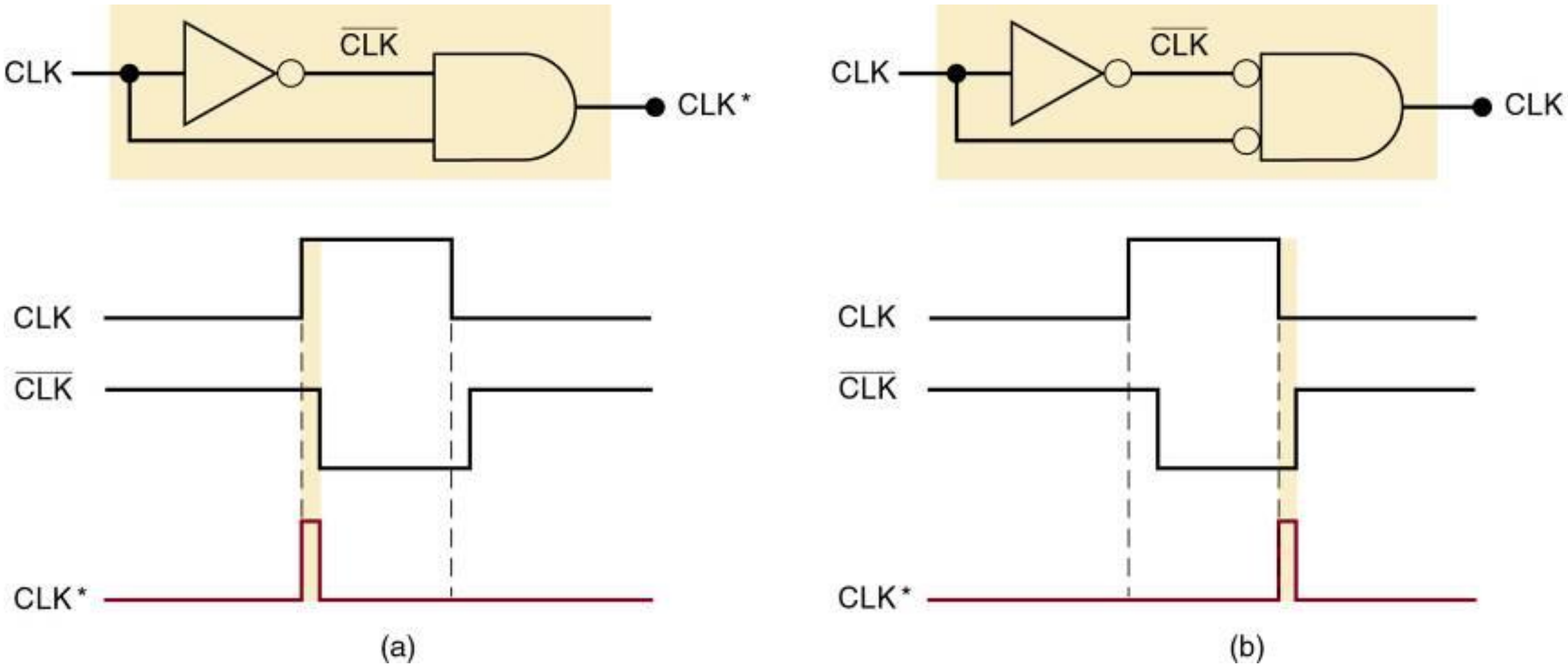
5-6 Clocked S-R Flip-Flop – Internal Circuitry



- An edge-triggered S-R flip-flop circuit features:
  - A basic **NAND** gate latch formed by **NAND**-3 and **NAND**-4.
  - A **pulse-steering circuit** formed by **NAND**-1 and **NAND**-2.
  - An **edge-detector circuit.**

5-6 Clocked S-R Flip-Flop – Internal Circuitry

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

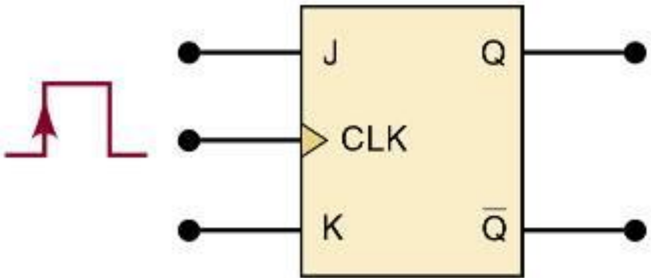


The duration of the *CLK** pulses is typically 2–5 ns.

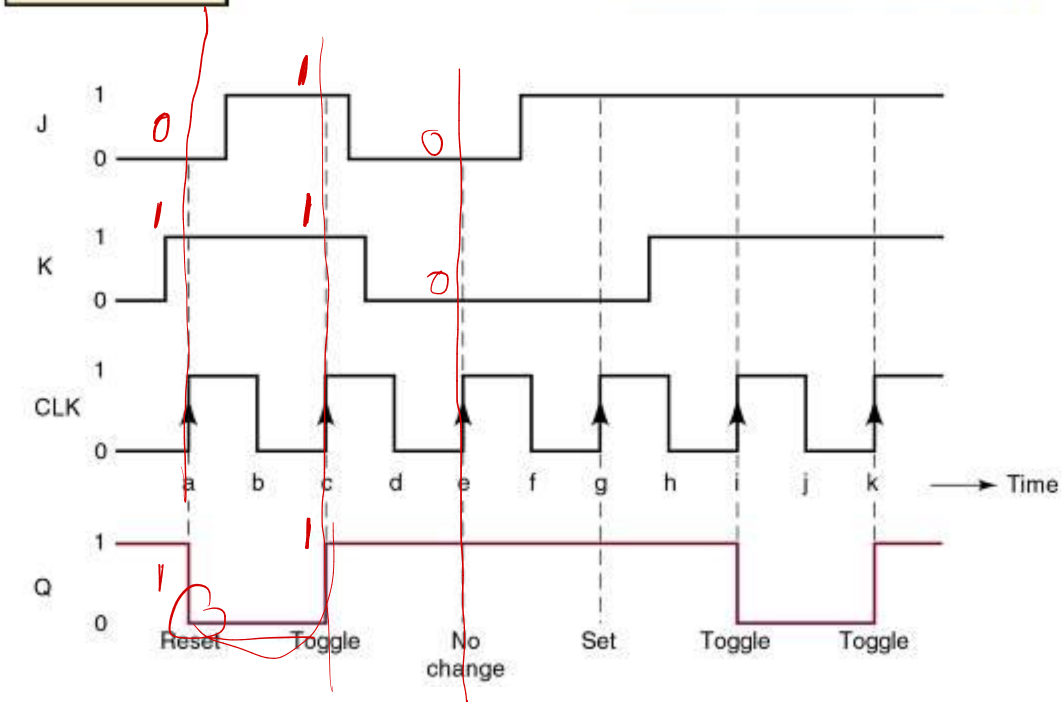
5-7 Clocked J-K Flip-Flop

- Operates like the S-R FF.
  - 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.

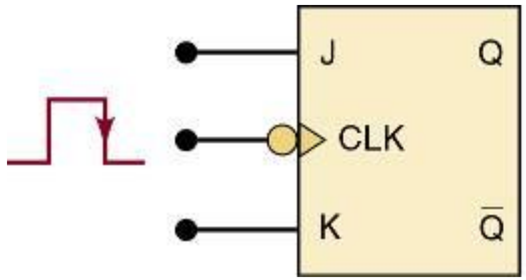
Clocked J-K flip-flop that responds only to the positive edge of the clock.



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



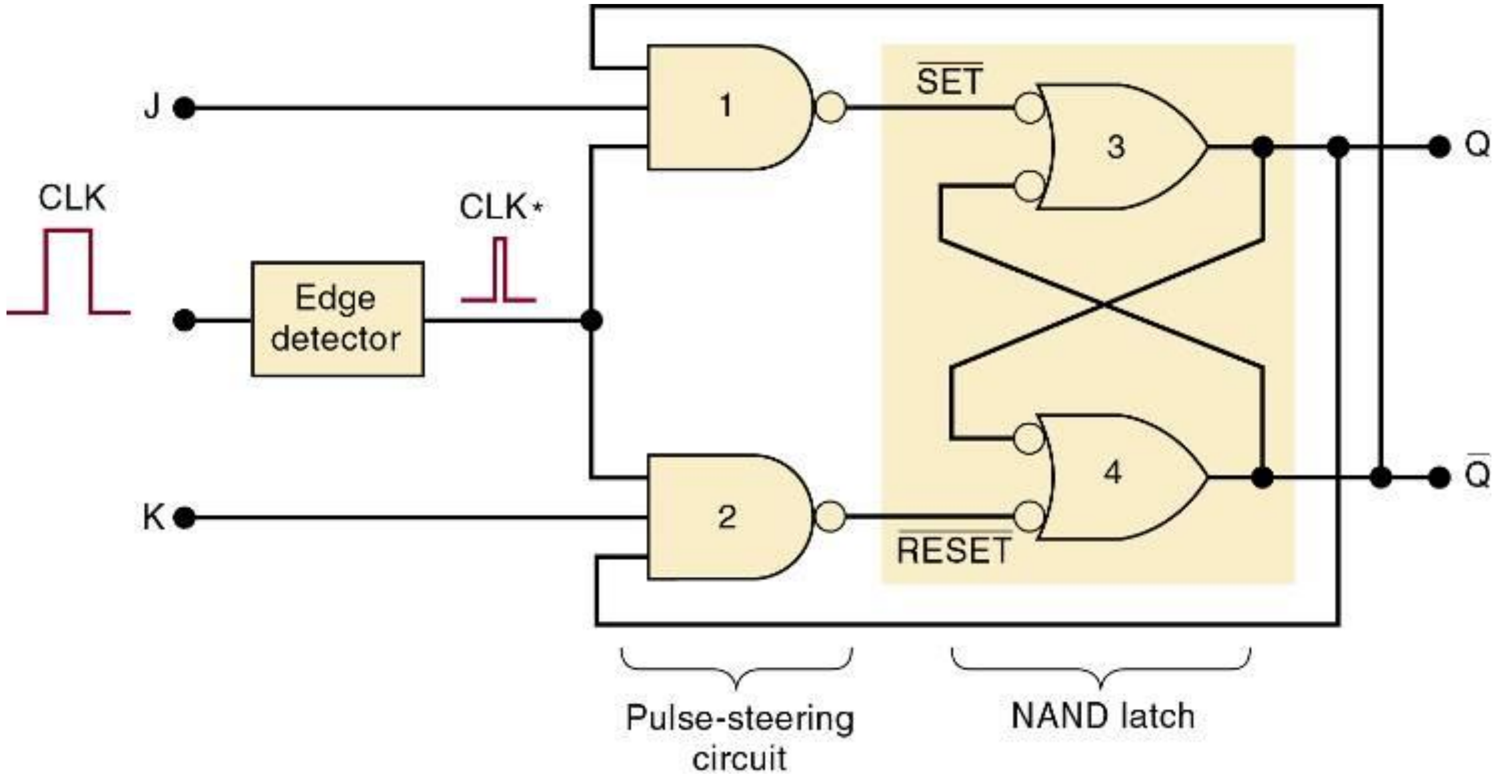
Clocked J-K flip-flop that responds only to the negative edge of the clock.



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

5-7 Clocked J-K Flip-Flop – Internal Circuitry

- The internal circuitry of an edge-triggered J-K flip-flop contains the same three sections as the edge-triggered S-R flip-flop.

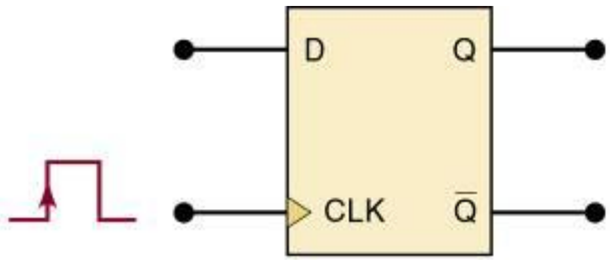


5-8 Clocked D Flip-Flop

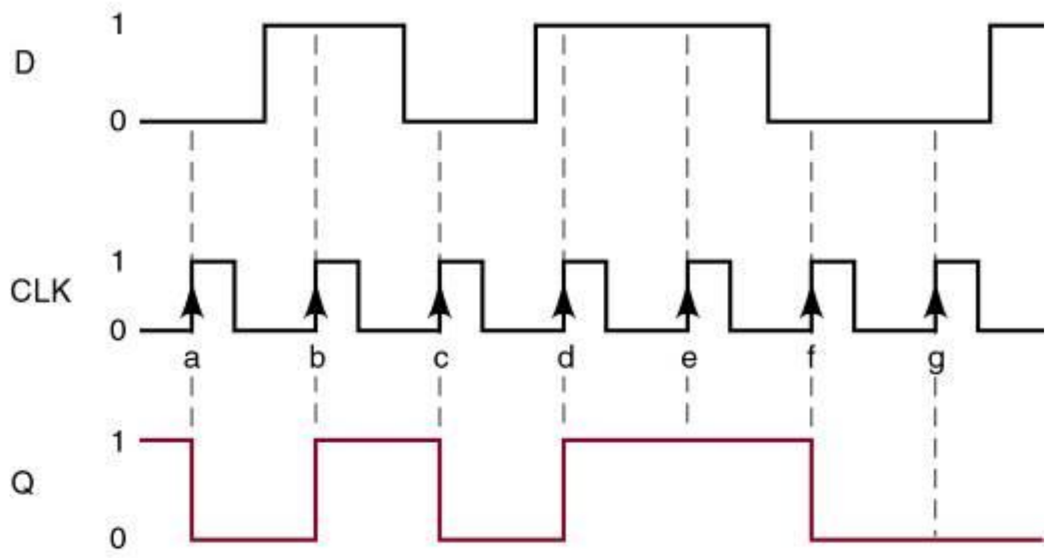
- One data input—output changes to the value of the input at either the positive- or negative-going clock trigger.
- May be implemented with a J-K FF by tying the J input to the K input through an inverter.
- Useful for parallel data transfer.

5-8 Clocked D Flip-Flop

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



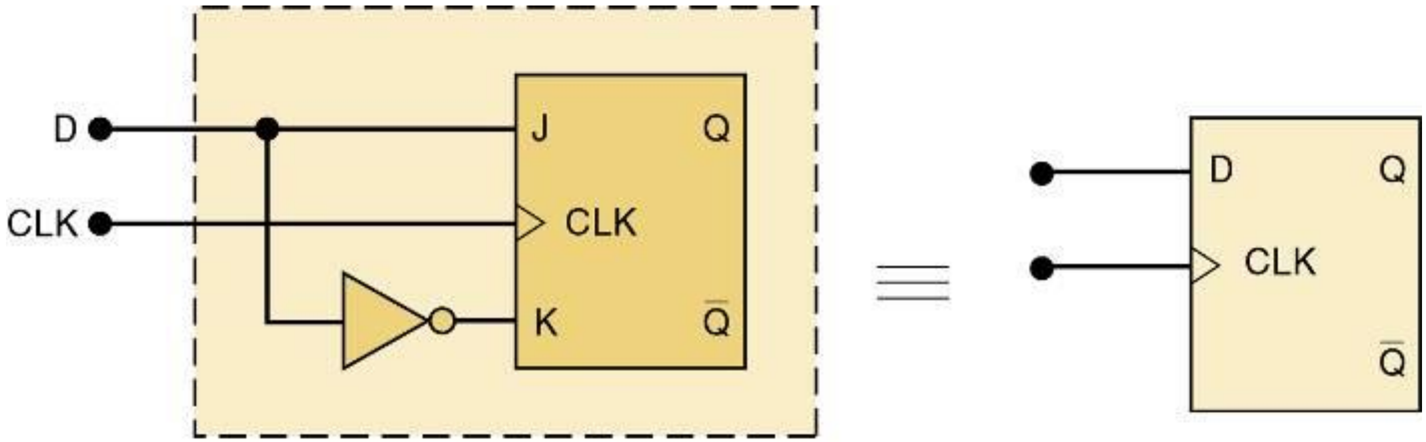
D	CLK	Q
0	↑	0
1	↑	1



5-8 Clocked D Flip-Flop - Implementation

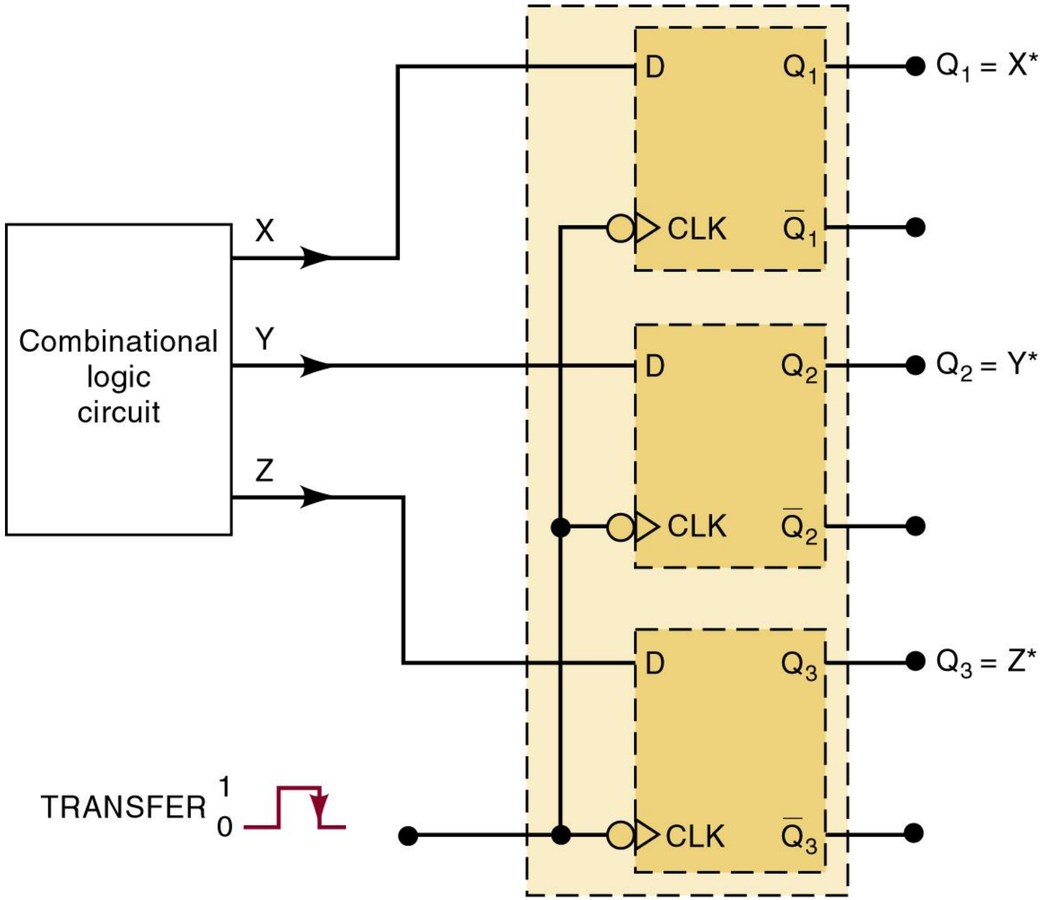
- An edge-triggered D flip-flop is implemented by adding a single INVERTER to the edge-triggered J-K flip-flop.
  - The same can be done to convert a S-R flip-flop to a D flip-flop.

Edge-triggered D flip-flop implementation from a J-K flip-flop.



Outputs *X, Y, Z* are to be transferred to FFs *Q1*, *Q2*, and *Q3* for storage.

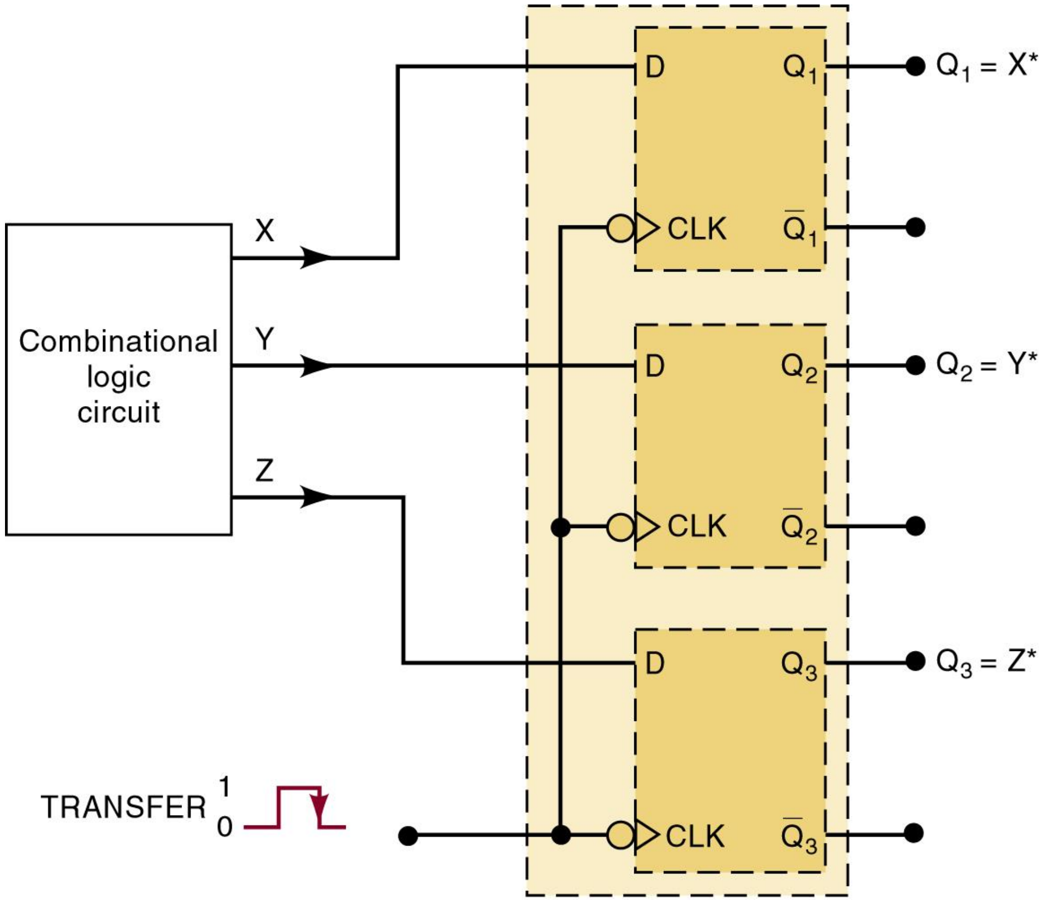
Using D flip-flops, levels present at *X*, *Y* & *Z* will be transferred to *Q*1, *Q*2 & *Q*3, upon application of a TRANSFER pulse to the common *CLK* inputs.



*After occurrence of NGT

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—the three bits X, Y & *Z* are transferred *simultaneously.*

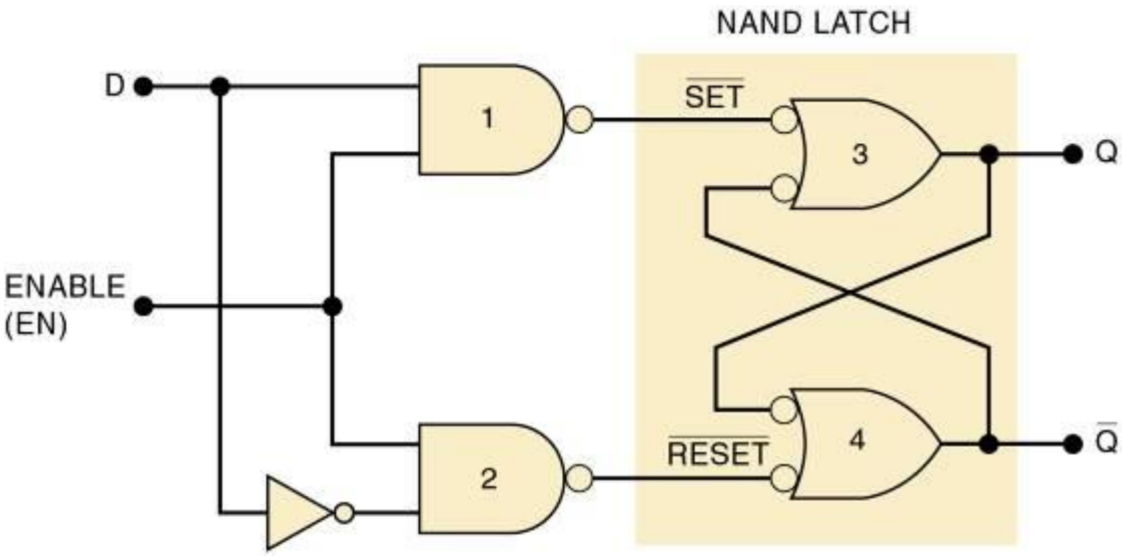


*After occurrence of NGT

5-9 D Latch (Transparent Latch)

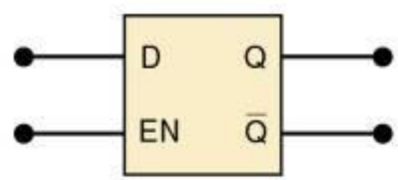
- The edge-triggered D flip-flop uses an edge-detector circuit to ensure the output responds to the *D* input *only* on active transition of the clock.
  - If this edge detector is not used, the resultant circuit operates as a **D latch**.

D latch structure, function table, logic symbol.



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.



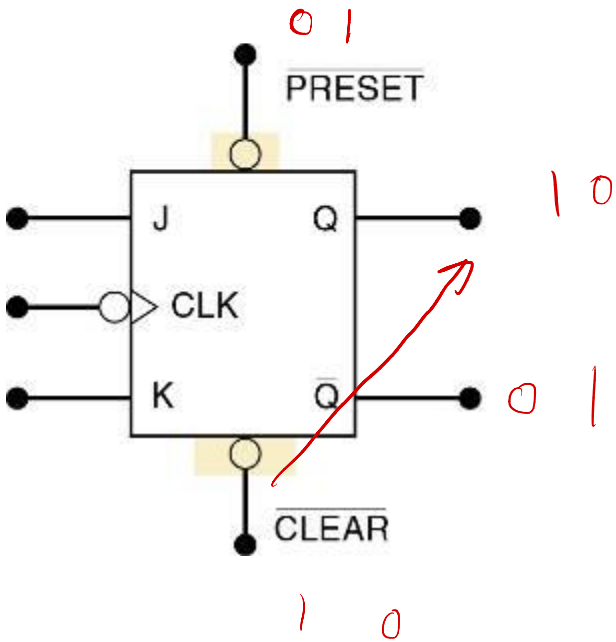
5-9 D Latch (Transparent Latch)

- The circuit contains the **NAND** latch and the steering **NAND** gates 1 and 2 *without* the edge-detector circuit.
- The common input to the steering gates is called an *enable* input (abbreviated *EN*)—rather than a clock input.
  - Its effect on the Q and $\overline{Q}$ outputs is not restricted to occurring only on its transitions

5-10 Asynchronous Inputs

- 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 J-K flip-flop with asynchronous inputs.



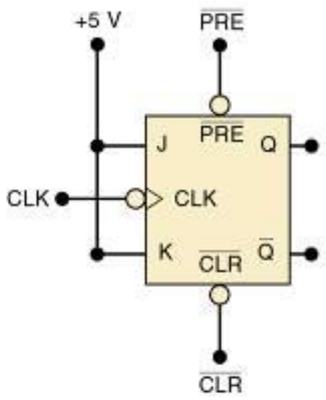
J	K	Clk	$\overline{PRE}$	$\overline{CLR}$	Q
0	0	↓	1	1	Q (no change)
0	1	↓	1	1	0 (Synch reset)
1	0	↓	1	1	1 (Synch set)
1	1	↓	1	1	$\overline{Q}$ (Synch toggle)
x	x	x	1	1	Q (no change)
x	x	x	1	0	0 (asynch clear)
x	x	x	0	1	1 (asynch preset)
x	x	x	0	0	(Invalid)

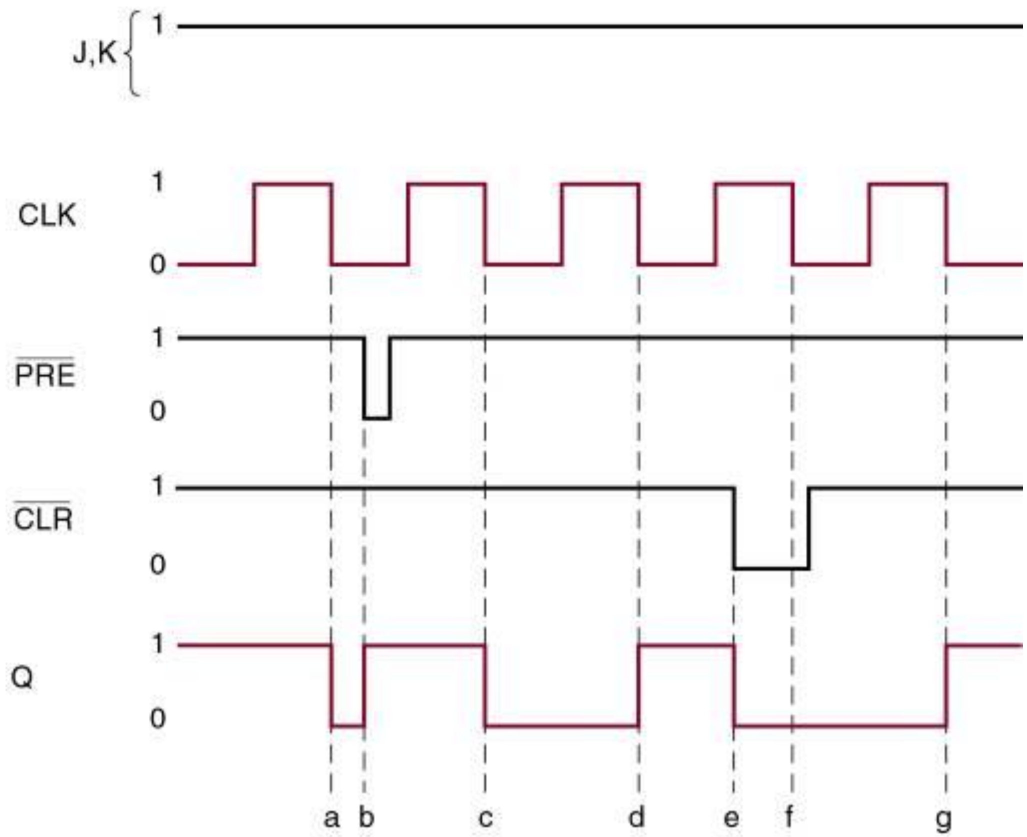
5-10 Asynchronous Inputs - Designations

- IC manufacturers do not agree on nomenclature for asynchronous inputs.
  - The most common designations are *PRE* (PRESET) and *CLR* (CLEAR).
    - Clearly distinguished from synchronous SET & RESET.
  - Labels such as *S-D* (direct SET) and *R-D* (direct RESET) are also used.

5-10 Asynchronous Inputs

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



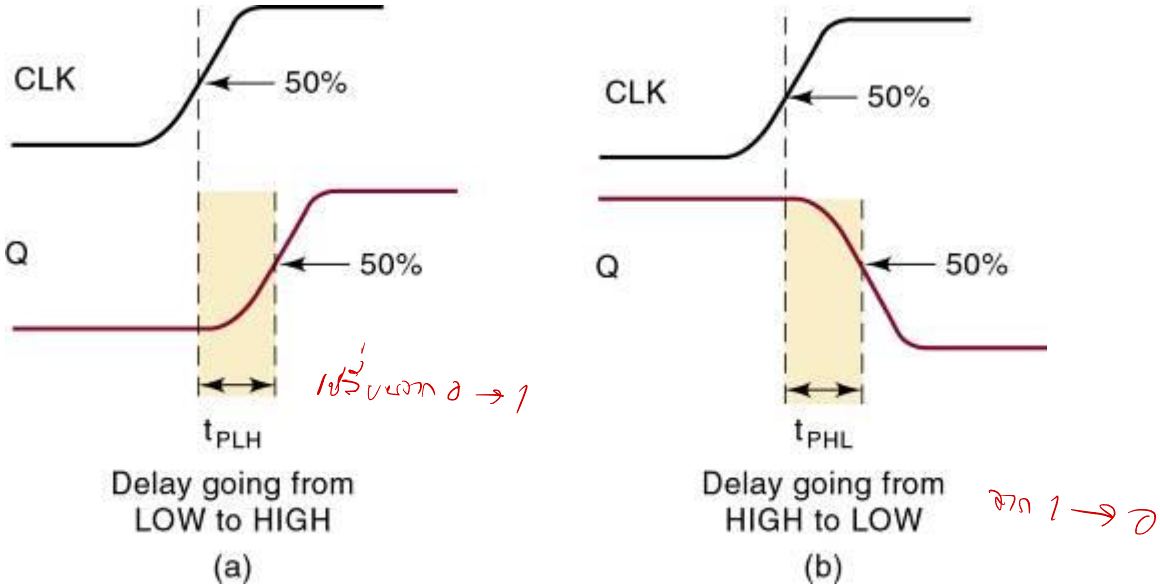


Point	Operation
a	Synchronous toggle on NGT of CLK
b	Asynchronous set on $\overline{PRE}$ = 0
c	Synchronous toggle
d	Synchronous toggle
e	Asynchronous clear on $\overline{CLR}$ = 0
f	$\overline{CLR}$ overrides the NGT of CLK
g	Synchronous toggle

5-11 Flip-Flop Timing Considerations - Parameters

- 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

FF propagation delays.



Clock Pulse HIGH and LOW and Asynch pulse width.

5-11 Flip-Flop Timing Considerations – Actual IC Values

7474	Dual edge-triggered D flip-flop (standard TTL)
74LS112	Dual edge-triggered J-K flip-flop (low-power Schottky TTL)
74C74	Dual edge-triggered D flip-flop (metal-gate CMOS)
74HC112	Dual edge-triggered J-K flip-flop (high-speed CMOS)

Timing values for FFs from manufacturer data books.

All of the listed values are *minimum* values, except propagation delays, which are *maximum* values.

		TTL		CMOS	
		7474	74LS112	74C74	74HC112
t_S		20 ns	20 ns	60 ns	25 ns
t_H		5	0	0	0
t_{PHL}	from *CLK* to *Q*	40	24	200	31
t_{PLH}	from *CLK* to *Q*	25	16	200	31
t_{PHL}	from *CLR* to *Q*	40	24	225	41
t_{PLH}	from *PRE* to *Q*	25	16	225	41
$t_W(L)$	*CLK* LOW time	37	15	100	25
$t_W(H)$	*CLK* HIGH time	30	20	100	25
$t_W(L)$	at *PRE* or *CLR*	30	15	60	25
f_{MAX}	in MHz	15	30	5	20

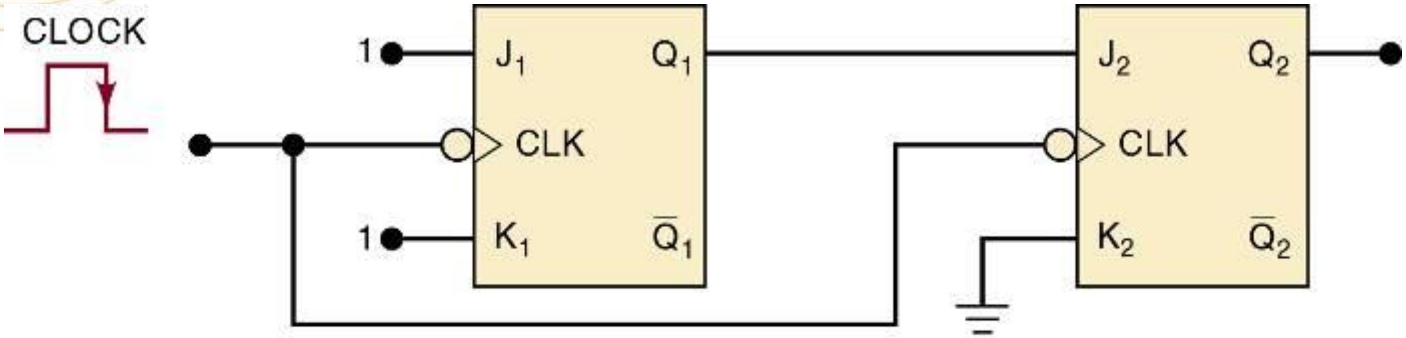
5-12 Potential Timing Problems in FF Circuits

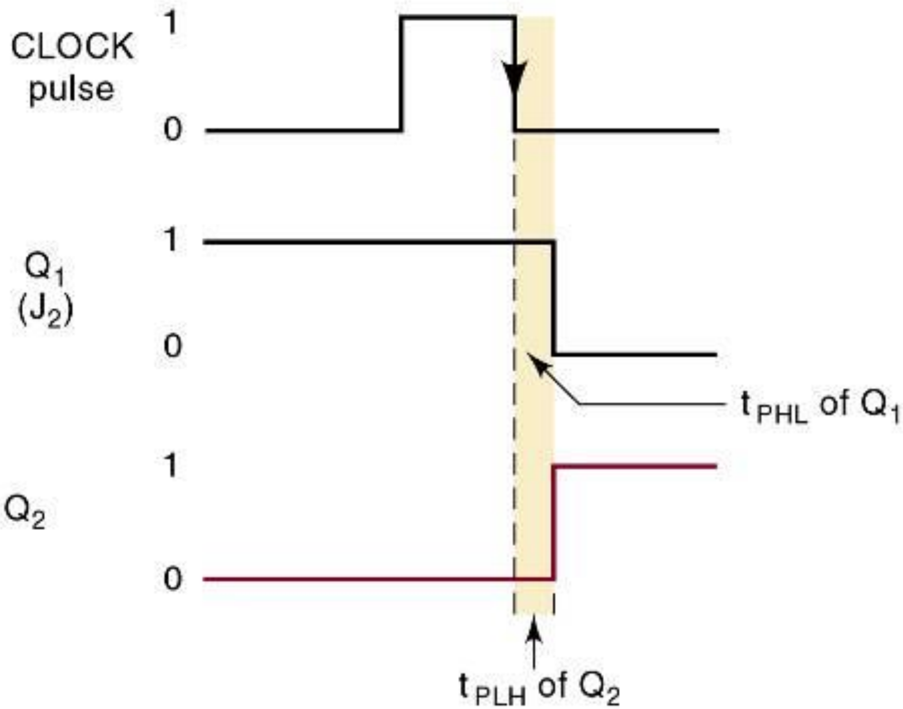
- When the output of one FF is connected to the input of another FF and both are triggered by the same clock, there is a *potential* timing problem.
  - Propagation delay may cause unpredictable outputs.
- Edge-triggered FFs have hold time requirements 5 ns or less—most have $t_H = 0$.
  - They have *no* hold time requirement.

Assume the FF hold time requirement is short enough to respond reliably according to the following rule:

Flip-Flop output will go to a state determined by logic levels present at its synchronous control inputs just prior to the active clock transition.

5-12 Potential Timing Problems in FF Circuits





*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.

5-13 Flip-Flop Applications

- Examples of applications:
  - Counting; Storing binary data
  - Transferring binary data between locations
- Many FF applications are categorized sequential.
  - Output follows a predetermined sequence of states.

RS

JK

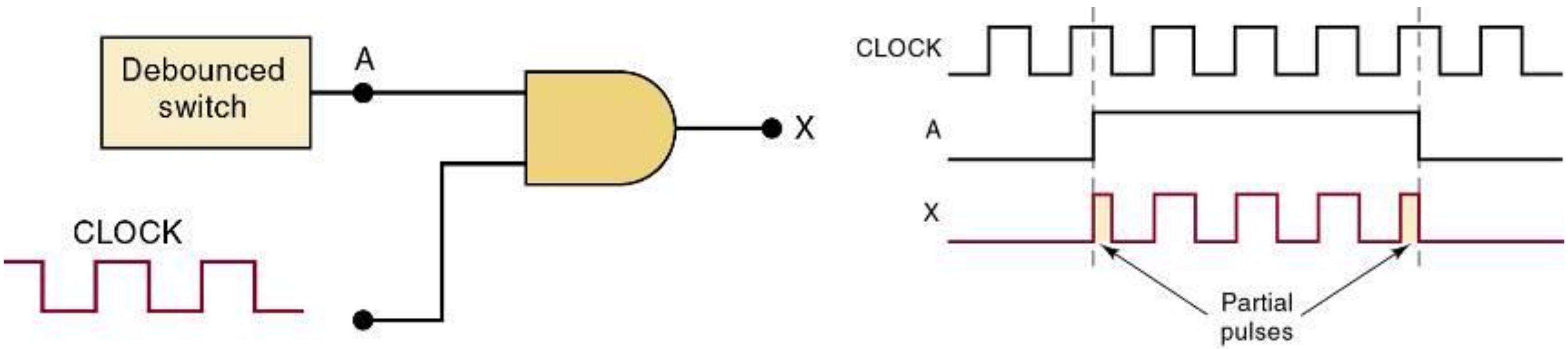
D

T

5-14 Flip-Flop Synchronization

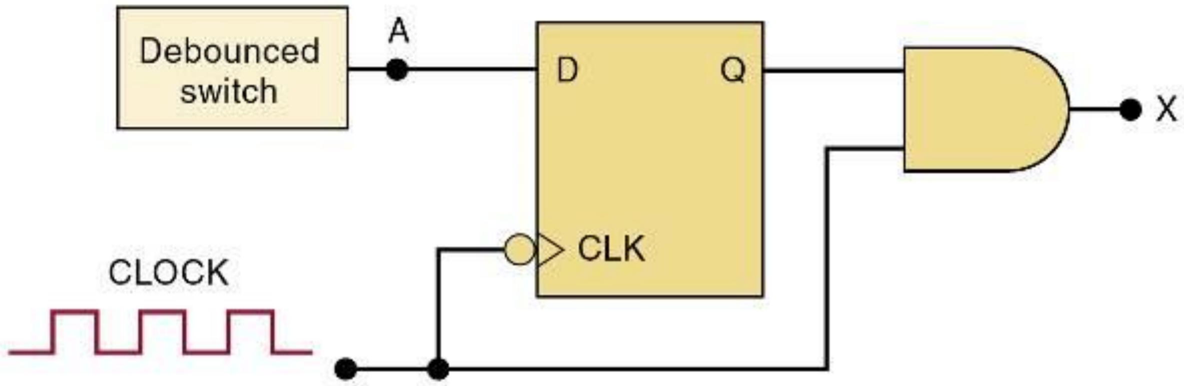
- Most systems are primarily synchronous in operation—in that changes depend on the clock.
- Asynchronous and synchronous operations are often combined—frequently through human input.
  - The random nature of asynchronous inputs *can* result in unpredictable results.

The asynchronous signal *A* can produce partial pulses at *X*.

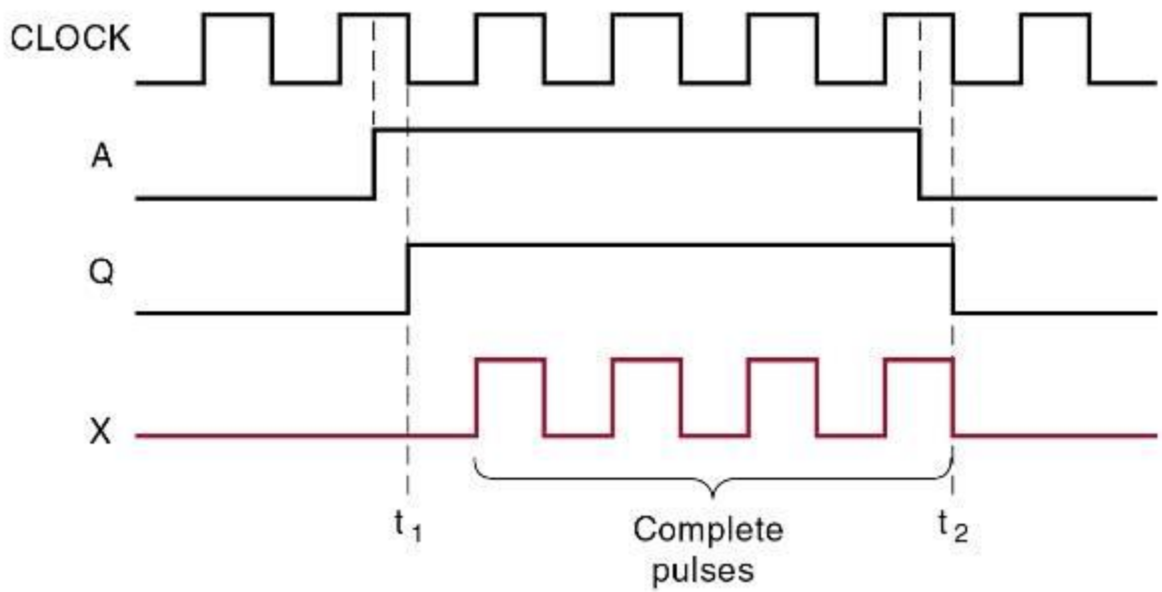


5-14 Flip-Flop Synchronization

An edge-triggered D flip-flop synchronizes the enabling of the **AND** gate to the NGTs of the clock.



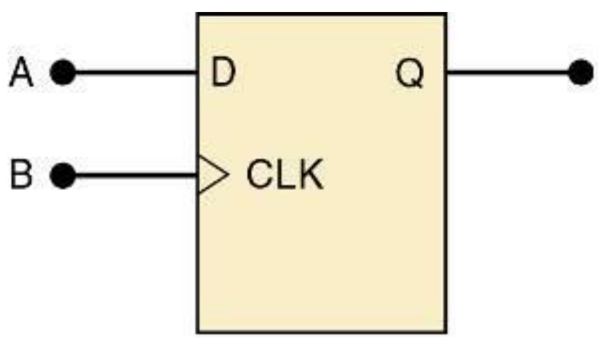
(a)



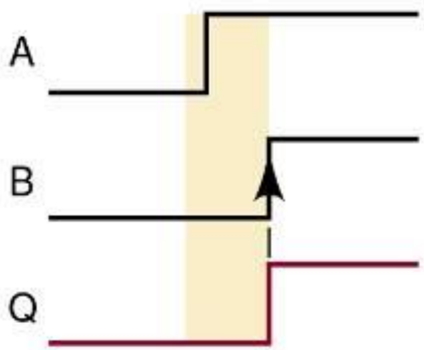
5-15 Detecting an Input Sequence

- FFs provide features pure combinational logic gates do not—in many situations, output activates *only* when inputs activate in a *certain sequence*
  - This requires the storage characteristic of FFs.

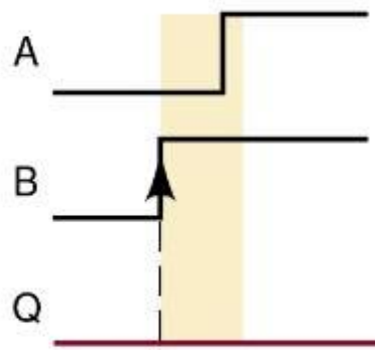
Clocked D flip-flop used to respond to a particular *sequence* of inputs.



To work properly, *A* must go HIGH, prior to *B*, by *at least* an amount of time *equal* to FF setup time.



A goes HIGH before B



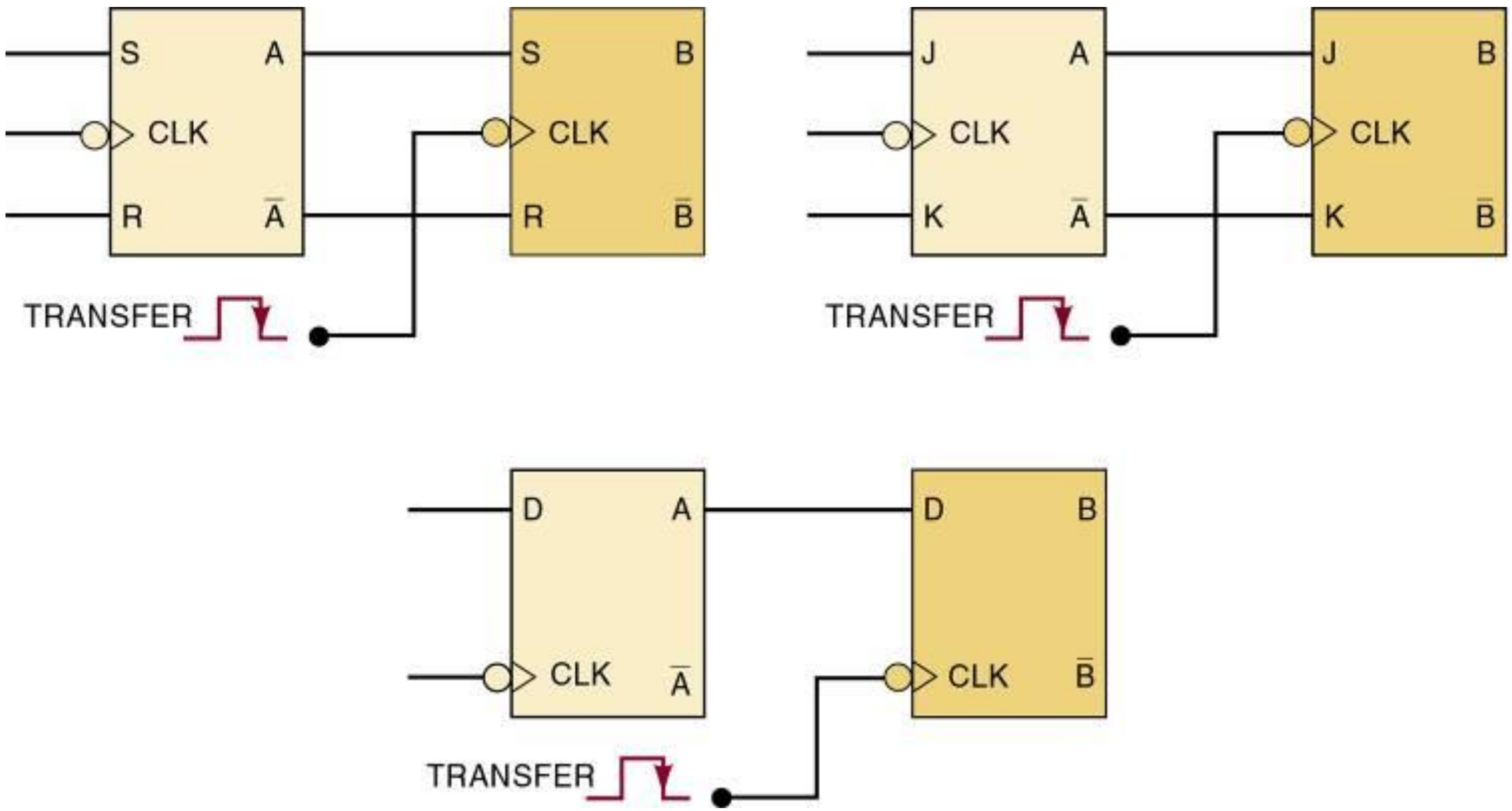
B goes HIGH before A

5-16 Data Storage and Transfer

- FFs are commonly used for storage and transfer of binary data.
  – Groups used for storage are **registers**.
- Data transfers take place when data is moved between registers or FFs.
  – *Synchronous* transfers take place at clock PGT/NGT.
  – *Asynchronous* transfers are controlled by PRE & CLR.

5-16 Data Storage and Transfer – Synchronous

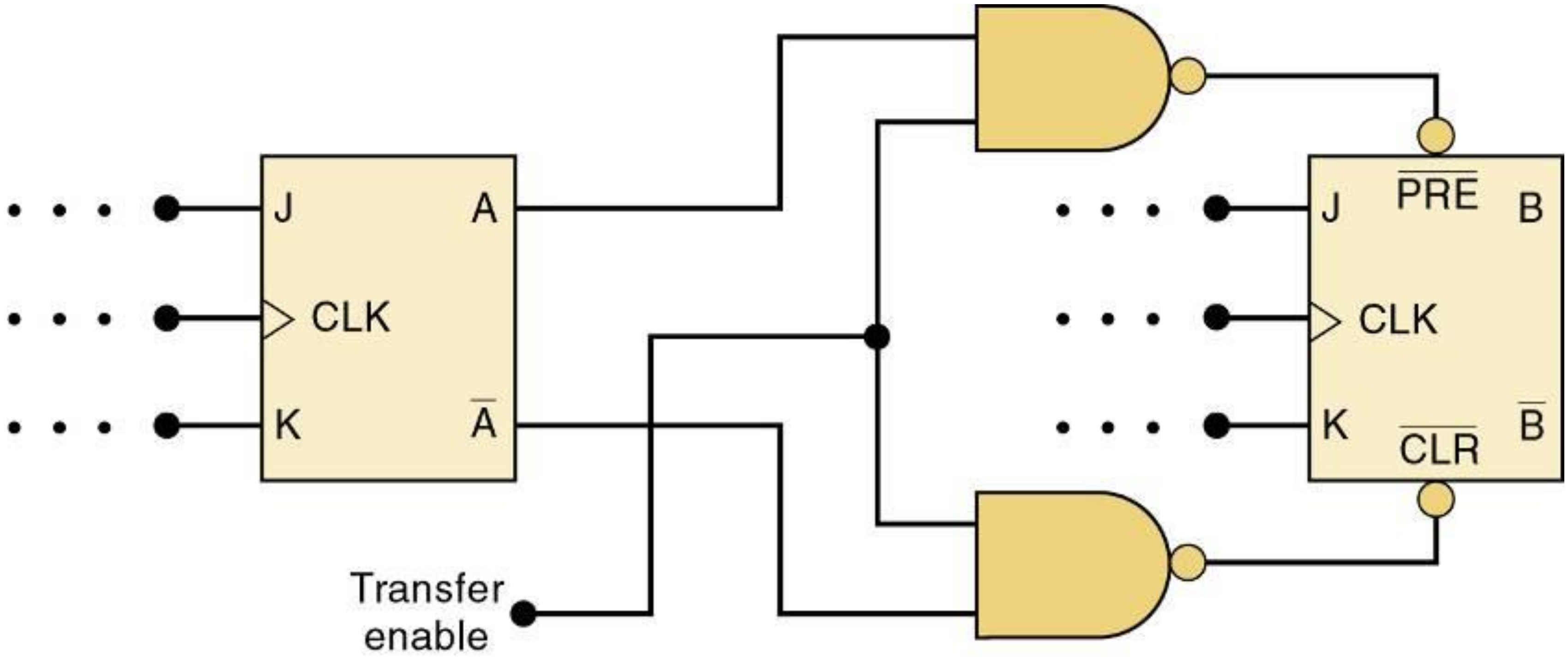
Synchronous data transfer operation by various clocked FFs.



CLK **inputs are used to perform the transfer.**

5-16 Data Storage and Transfer – Asynchronous

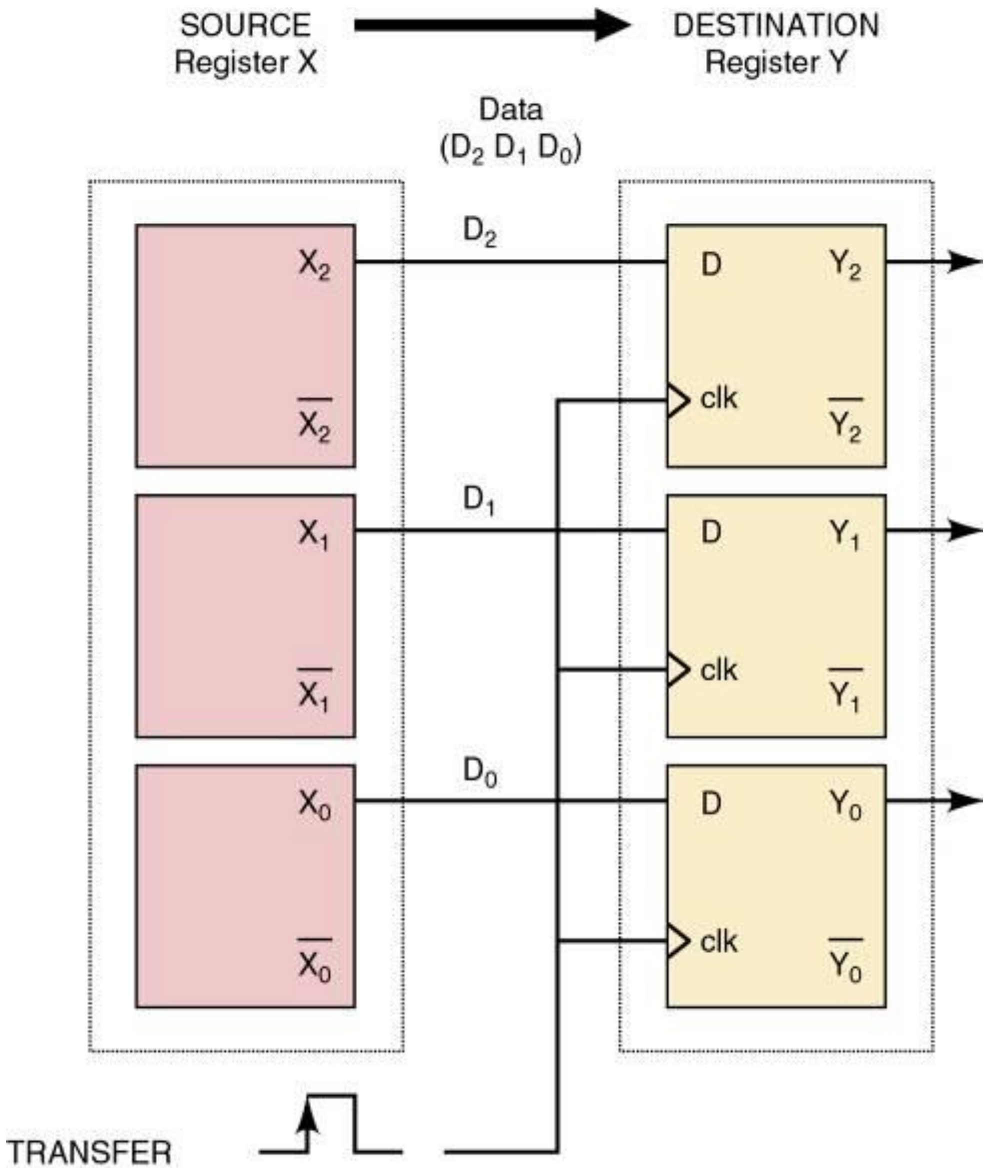
Asynchronous data transfer operation.



PRE and CLR **inputs are used to perform the transfer.**

5-16 Data Storage and Transfer – Parallel

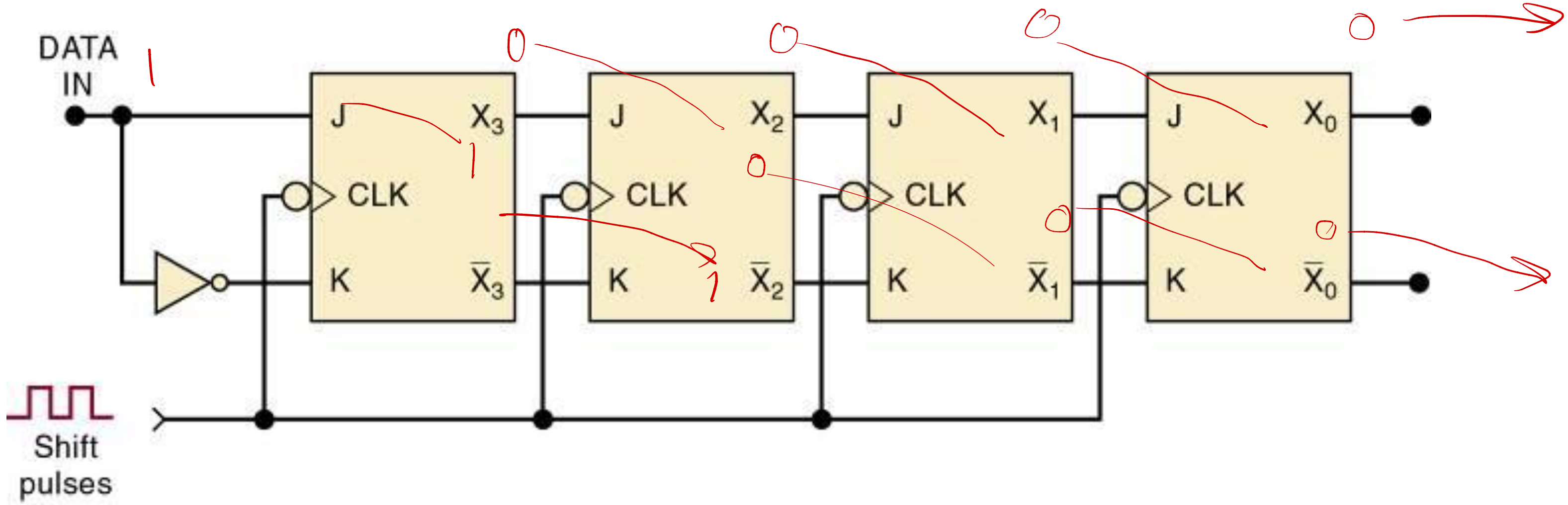
Transferring the bits of a register *simultaneously* is a *parallel* transfer.



5-17 Serial Data Transfer

- Transferring the bits of a register *a bit at a time* is a *serial* transfer.

- A **shift register** is 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.



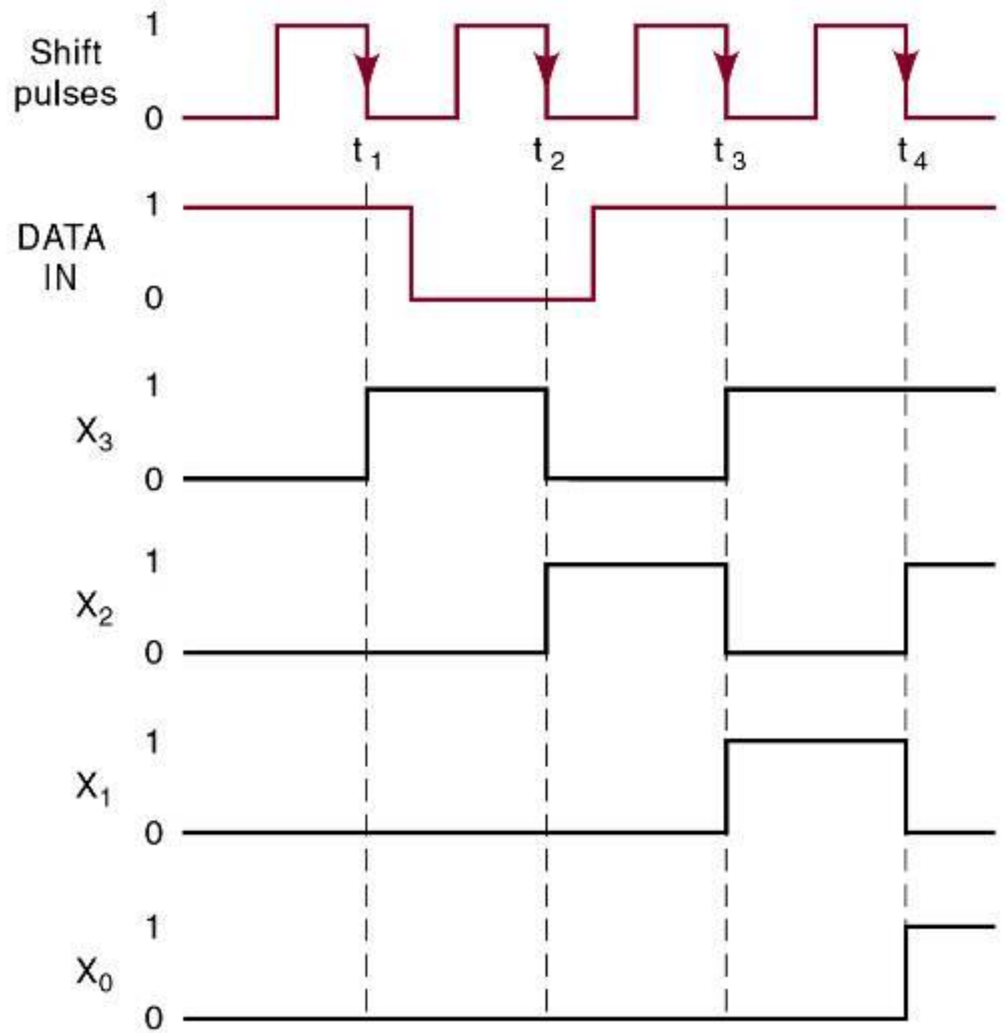
J-K flip-flops operated as a four-bit shift register.

5-17 Serial Data Transfer – Shift Register

Input data are shifted left to right from FF to FF as shift pulses are applied.

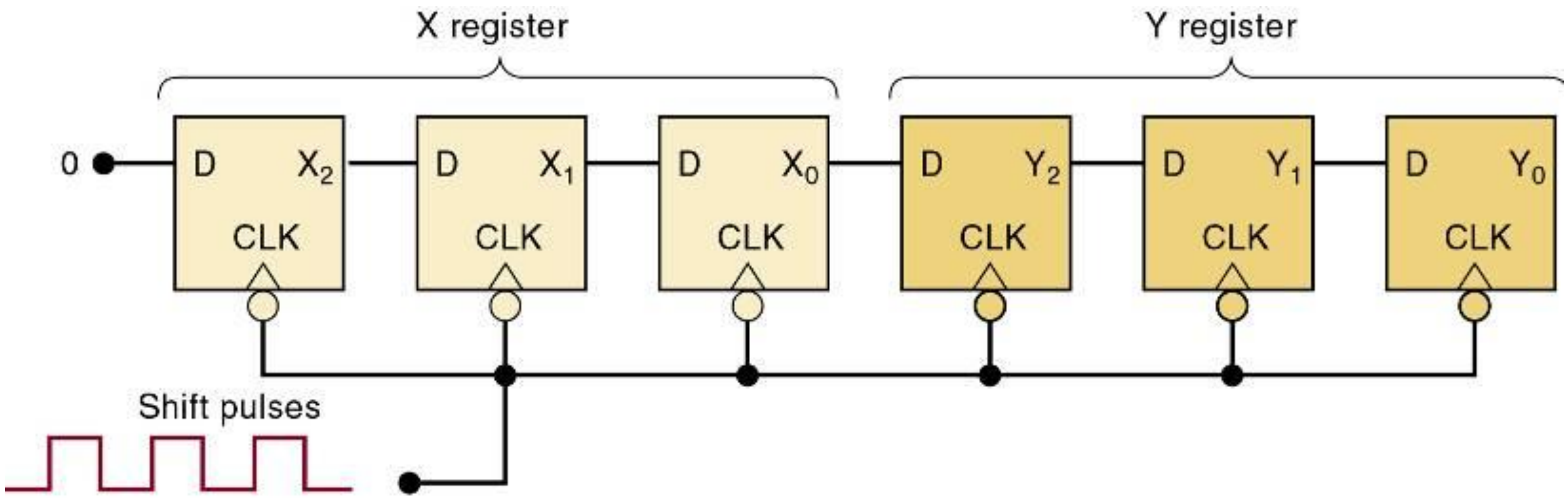
In this shift-register arrangement, it is necessary to have FFs with very small hold time requirements.

There are times when the *J*, *K* inputs are changing at about the same time as the *CLK* transition.



5-17 Serial Data Transfer – Shift Register

Two connected three-bit shift registers.



The contents of the X register will be serially transferred (shifted) into register Y.

The D flip-flops in each shift register require fewer connections than J-K flip-flops.

5-17 Serial Data Transfer – Shift Register

Two connected three-bit shift 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	
0	0	1	0	1	0	
0	0	0	1	0	1	

5-17 Serial Data Transfer – Shift Register

Two connected three-bit shift registers.

On each pulse NGT, each FF takes on the value stored in the FF on its *left* prior to the pulse.

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	
0	0	0	1	0	1	

5-17 Serial Data Transfer – Shift Register

Two connected three-bit shift registers.

On each pulse NGT, each FF takes on the value stored in the FF on its *left* prior to the pulse.

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	

Two connected three-bit shift registers.

On each pulse NGT, each FF takes on the value stored in the FF on its *left* prior to the pulse.

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

5-17 Serial Data Transfer – Shift Register

Two connected three-bit shift registers.

After three pulses:

The 1 initially in *X*2 is in *Y*2.
The 0 initially in *X*1 is in *Y*1.
The 1 initially in *X*0 is in *Y*0.

X_2	X_1	X_0	Y_2	Y_1	Y_0
1	0	1	0	0	0
0	1	0	1	0	0
0	0	1	0	1	0
0	0	0	1	0	1

The 101 stored in the X register has now been shifted into the Y register.

The X register has lost its original data, and is at 000.

5-17 Serial Data Transfer vs. Parallel

- FFs in can just as easily be connected so that information shifts from *right* to *left*.
  - No general advantage of one direction over another.
    - Often dictated by the nature of the application.
- Parallel transfer requires more interconnections between sending & receiving registers than serial.
  - More critical when a greater number of bits of are being transferred.
- Often, a combination of types is used
  - Taking advantage of parallel transfer *speed* and serial transfer the *economy and simplicity* of serial transfer.

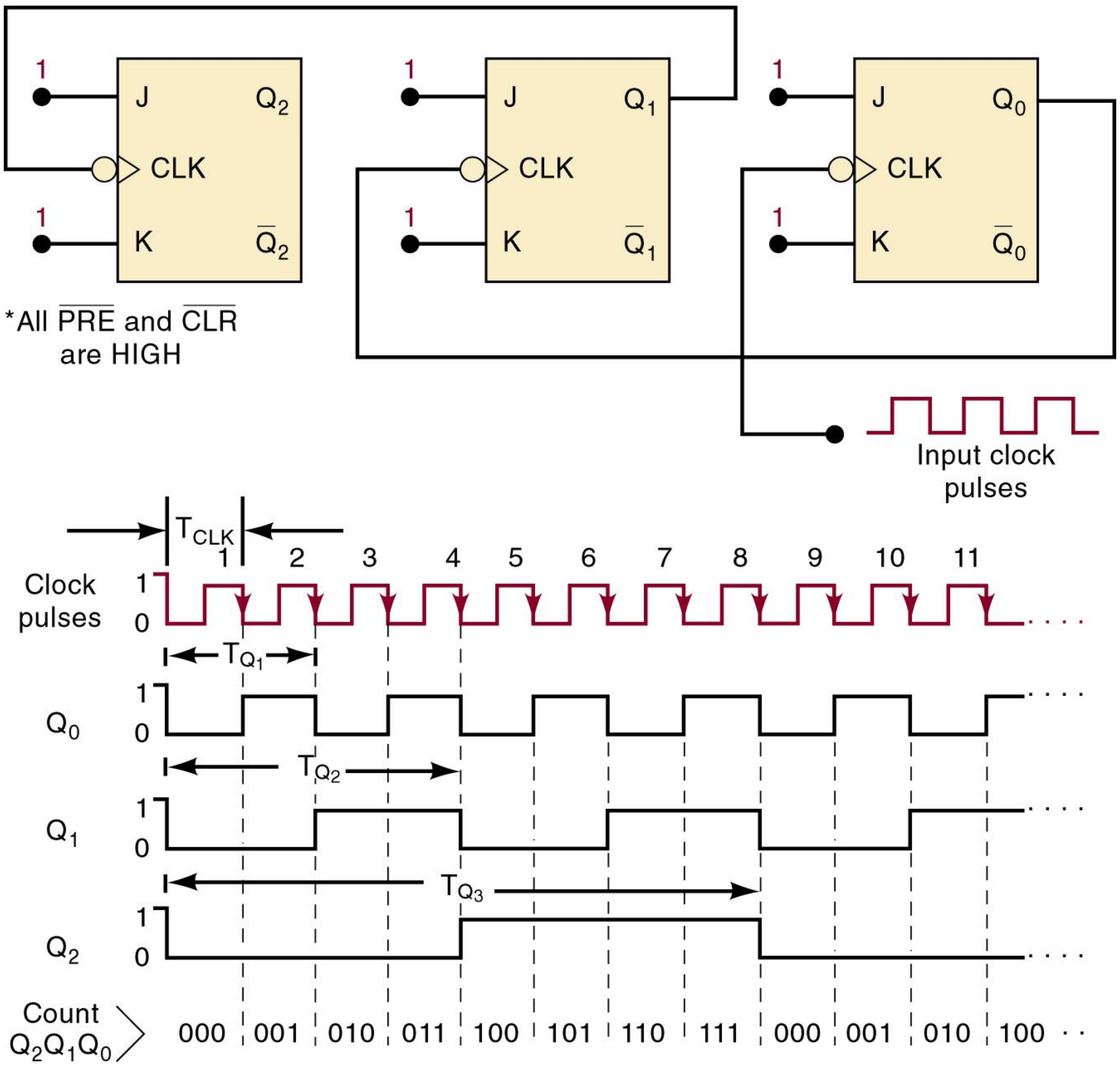
5-18 Frequency Division and Counting

J-K flip-flops wired as a three-bit binary counter (MOD-8).

Each FF divides the input frequency by 2.

Output frequency is 1/8 of the input (clock) frequency.

A fourth FF would make the frequency 1/16 of the clock.



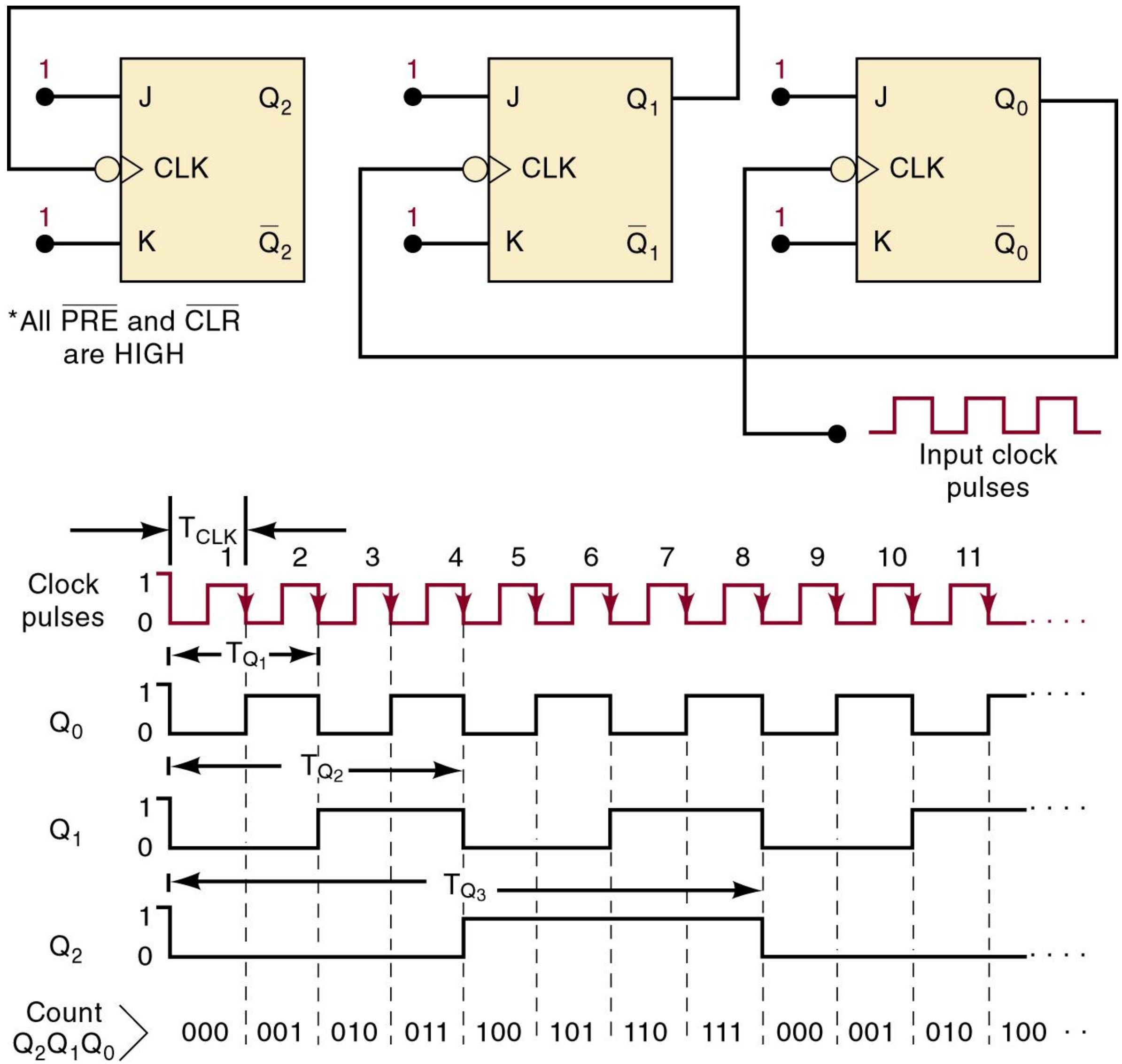
5-18 Frequency Division and Counting

J-K flip-flops wired as a three-bit binary counter (MOD-8).

This circuit also acts as a binary counter.

Outputs will count from 000_2 to 111_2 or 0_{10} to 7_{10}.

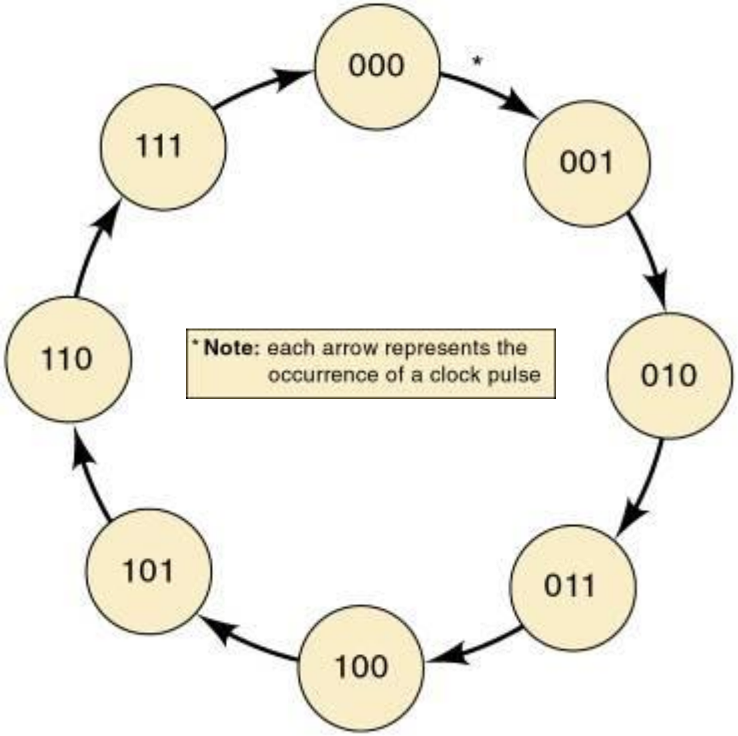
The number of states possible in a counter is the modulus or MOD number.



A MOD-8 (2^3) counter.

If another FF is added it would become a MOD-16 (2^4) counter.

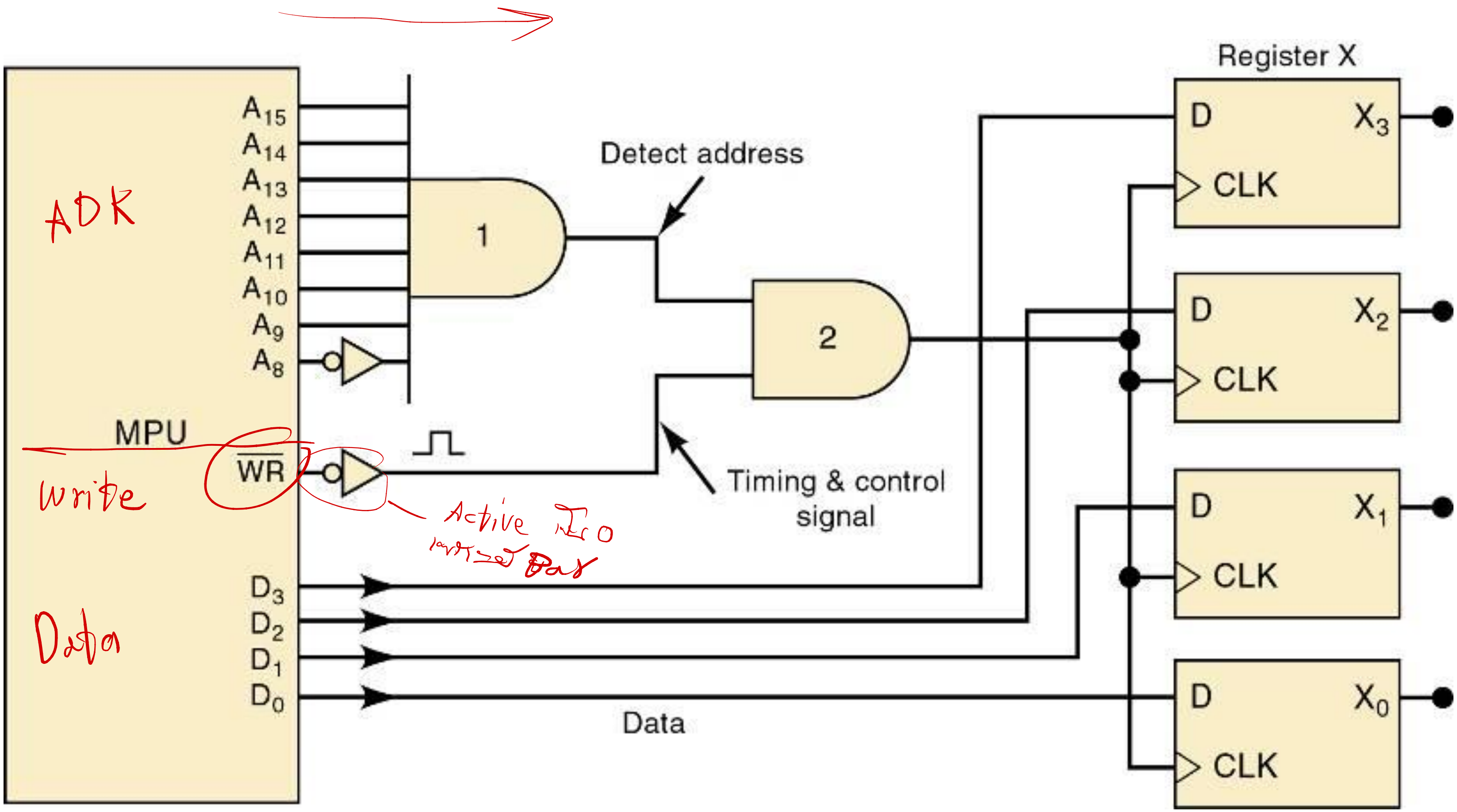
2^2 Q_2	2^1 Q_1	2^0 Q_0	
0	0	0	Before applying clock pulses
0	0	1	After pulse #1
0	1	0	After pulse #2
0	1	1	After pulse #3
1	0	0	After pulse #4
1	0	1	After pulse #5
1	1	0	After pulse #6
1	1	1	After pulse #7
0	0	0	After pulse #8 recycles to 000
0	0	1	After pulse #9
0	1	0	After pulse #10
0	1	1	After pulse #11
.	.	.	. . .
.	.	.	. . .
.	.	.	. . .



5-19 Microcomputer Application

- Microprocessor units (MPUs) perform many functions involving use of registers for data transfer and storage.
- MPUs may send data to external registers for many purposes, including:
  - Solenoid/relay control; Device positioning.
  - Motor starting & speed controls.

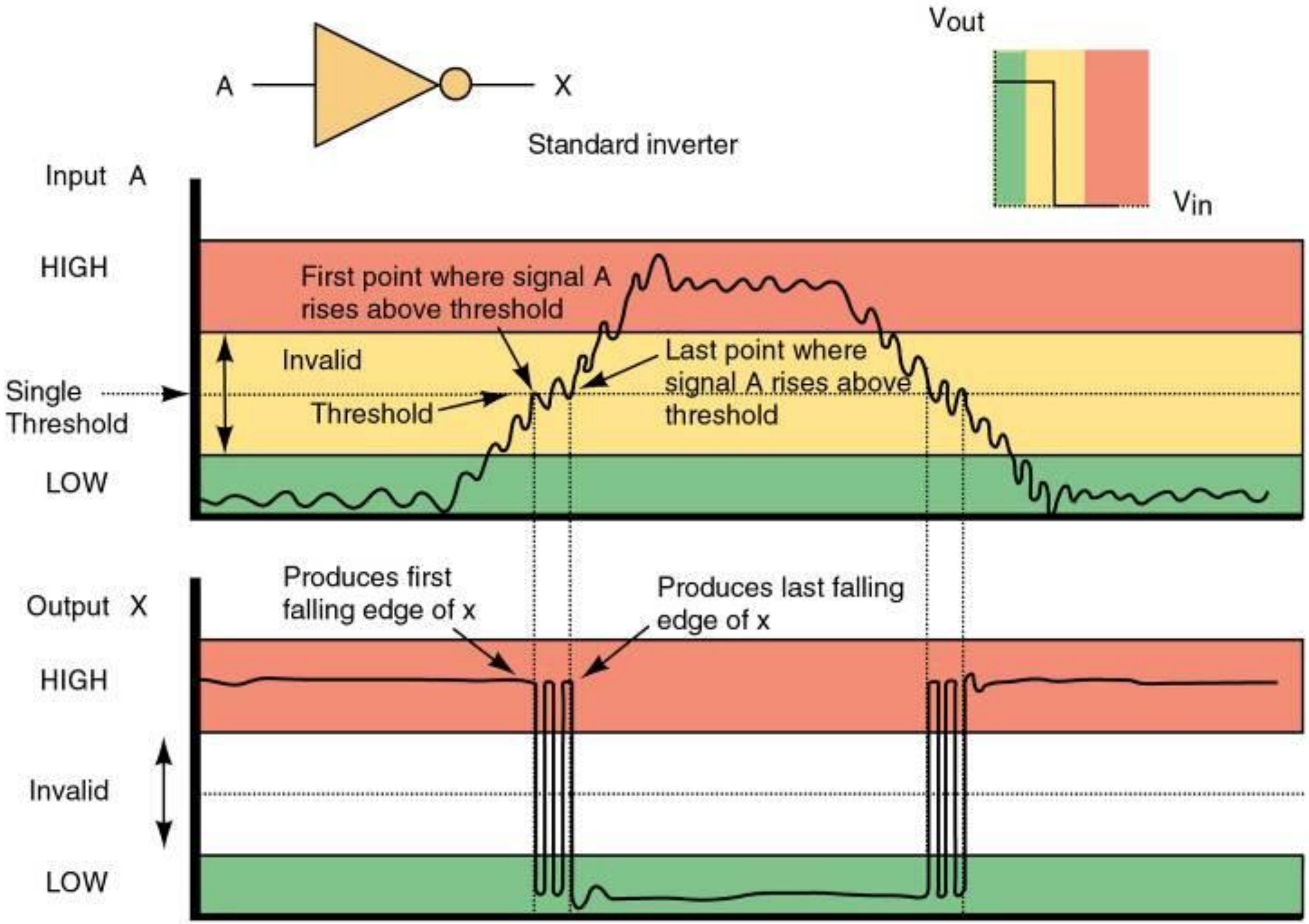
Microprocessor transferring binary data to an external register.



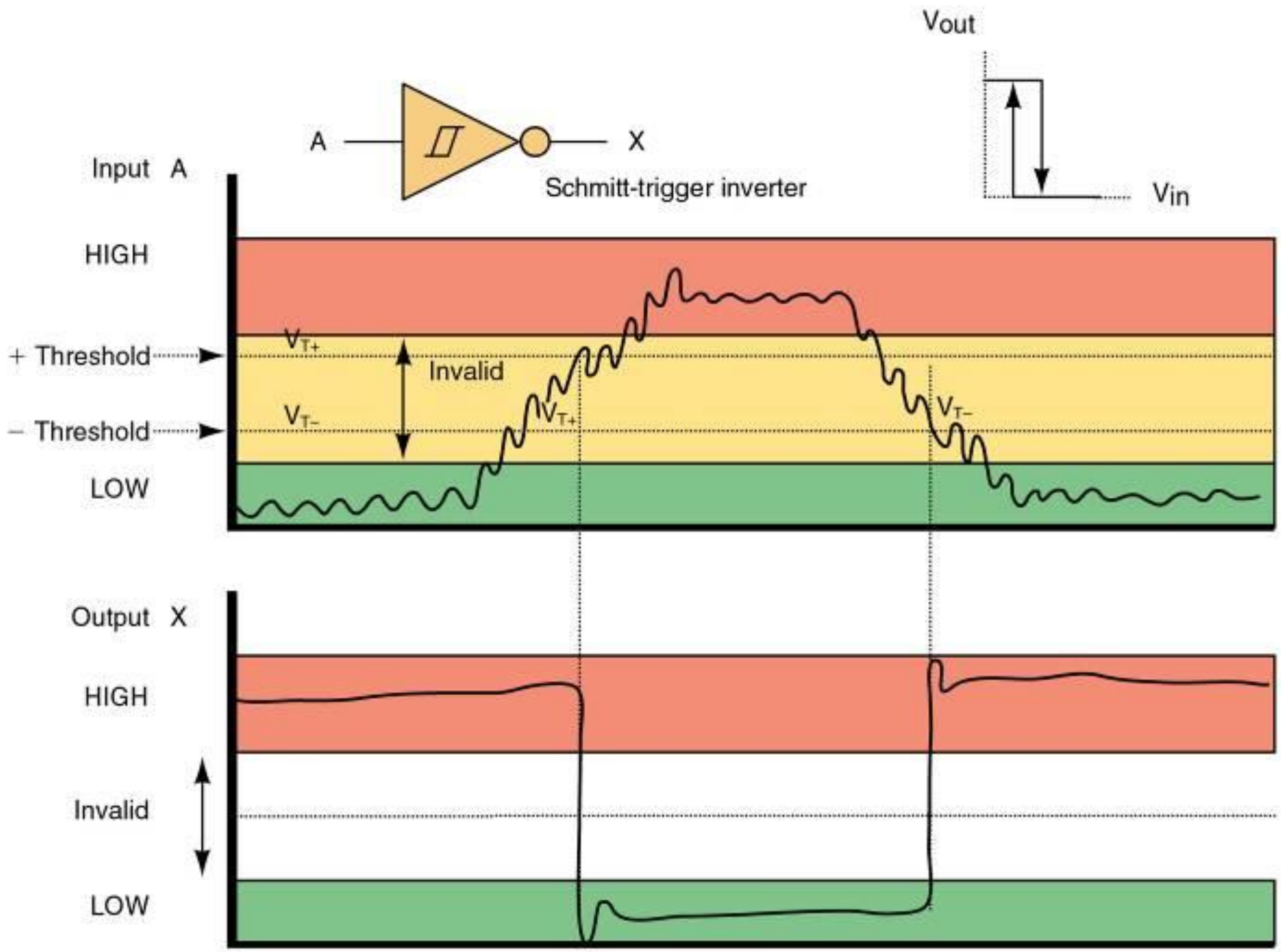
5-20 Schmitt-Trigger Devices

- Not classified as a FF—but has a useful a memory characteristic in certain situations.
- Accepts slow changing signals and produces a signal that transitions quickly, oscillation-free.
- A Schmitt trigger device will not respond to input until it exceeds the positive-(V_{T+}) or negative-(V_{T-}) going threshold.
- Separation between the threshold levels means the device will "remember" the last threshold exceeded.
  - Until the input goes to the opposite threshold.

Standard inverter response to slow noisy input.



Schmitt-trigger response to slow noisy input.



5-21 One-shot (Monostable Multivibrator)

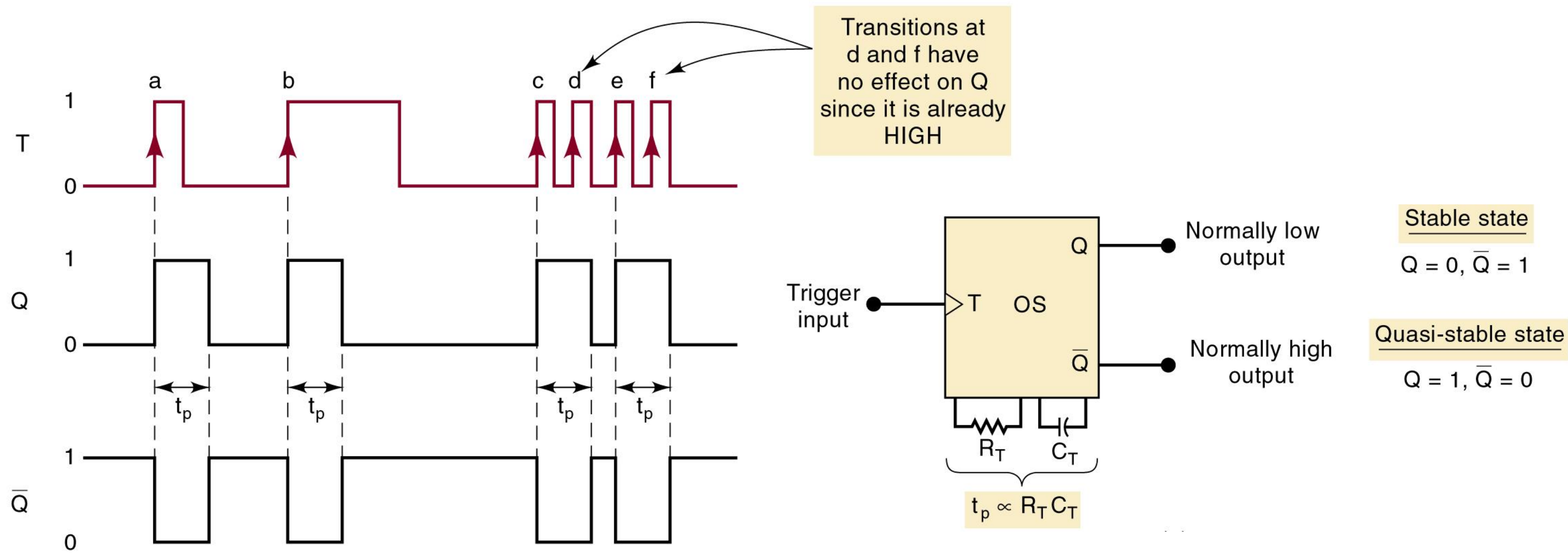
- Like the FF, the OS has two outputs, Q and $\overline{Q}$.
  - The inverse of each other.
- One shots are called monostable multivibrators because they have only one stable state.
  - Prone to triggering by noise.
- Changes from stable to quasi-stable state for a fixed time-period (t_p).
  - Usually determined by an *RC* time constant from external components.

5-21 One-shot (Monostable Multivibrator)

- **Nonretriggerable** devices trigger & return to stable.
- **Retriggerable** devices can be triggered while in the quasi-stable state, to begin another pulse.

5-21 One-shot (Monostable Multivibrator)

OS symbol and typical waveforms for nonretriggerable operation.

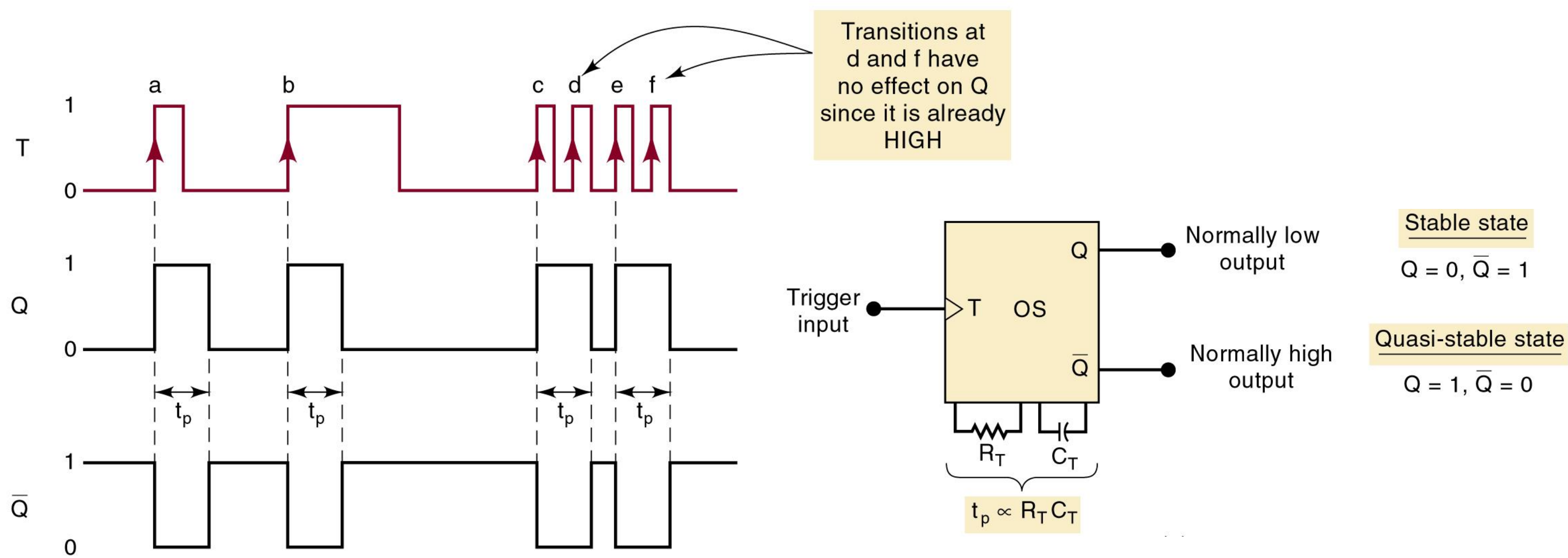


PGTs at points *a, b, c,* and *e* will trigger the OS to its quasi-stable state for a time t_p.

After which it automatically returns to the stable state.

5-21 One-shot (Monostable Multivibrator)

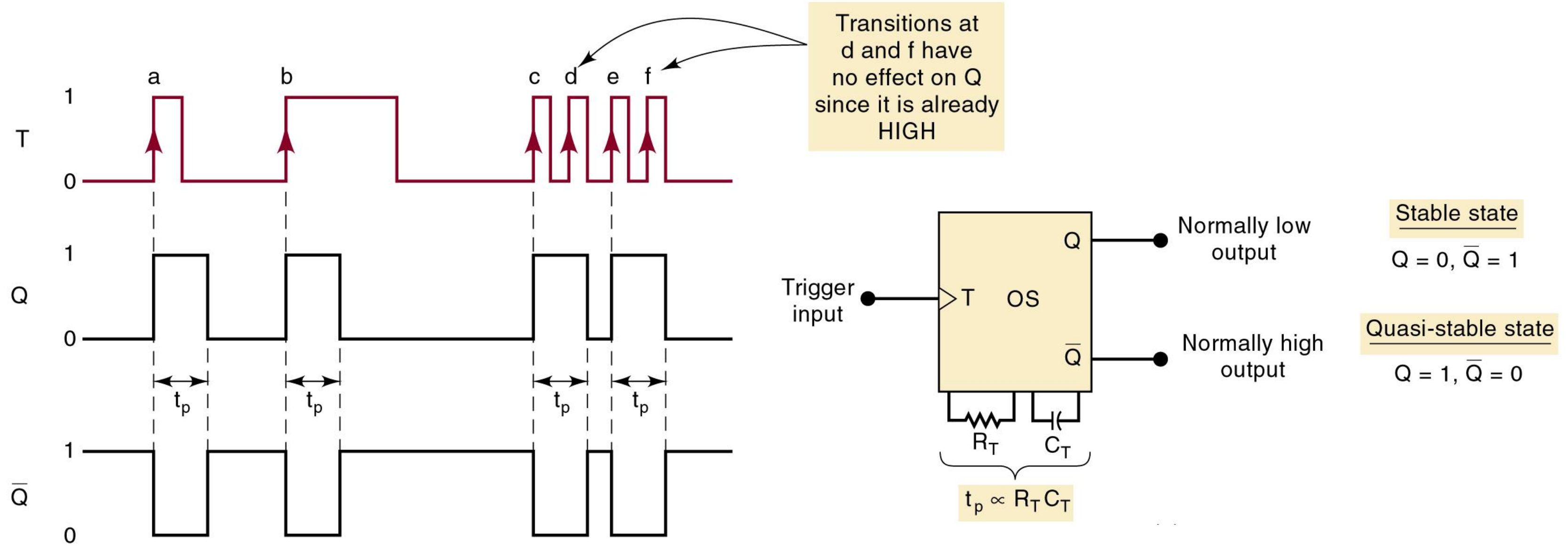
OS symbol and typical waveforms for nonretriggerable operation.



PGTs at points *d* and *f* have no effect on the OS because it has already been triggered quasi-stable.

OS must return to the stable before it can be triggered.

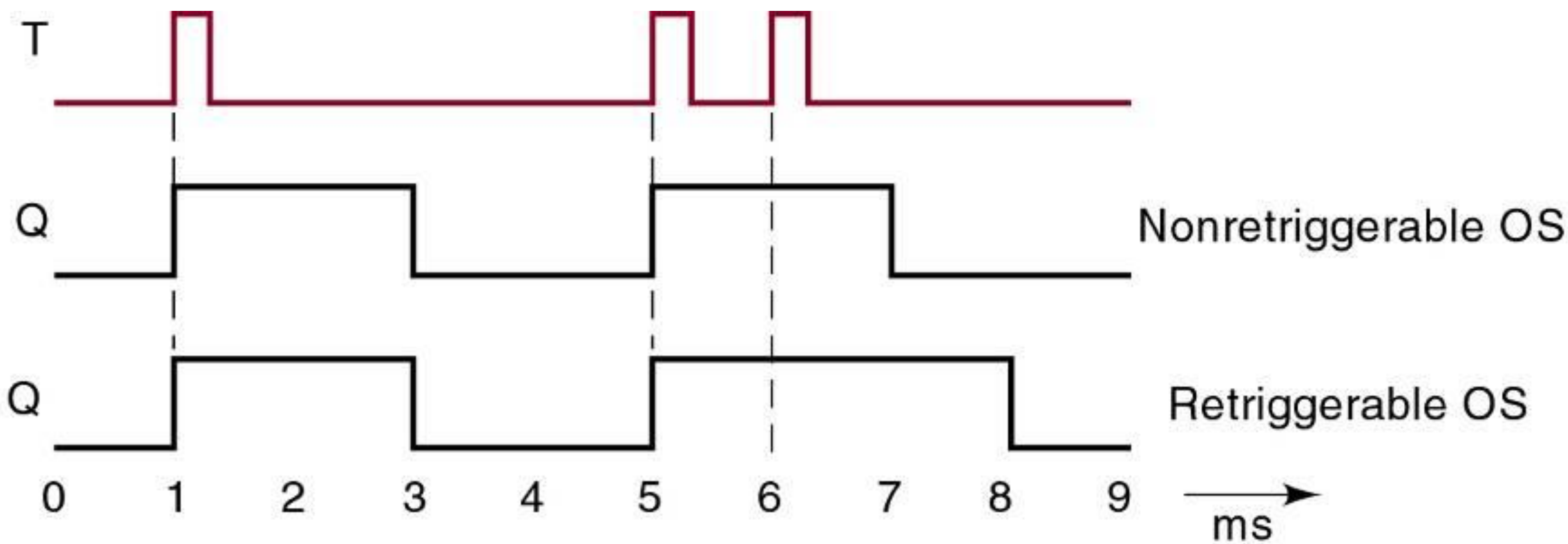
OS symbol and typical waveforms for nonretriggerable operation.



OS output-pulse duration is always the same, regardless of the duration of the input pulses.

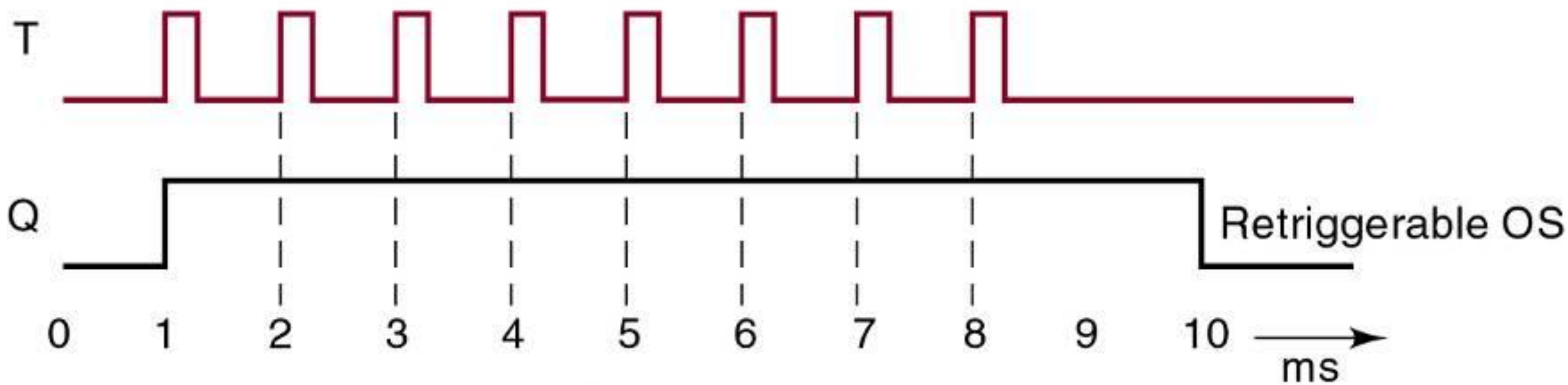
Time t_p depends only on R_T, C_T & internal OS circuitry.

Comparison of nonretriggerable and retriggerable OS responses for t_p = 2ms.



5-21 One-shot (Monostable Multivibrator)

Retriggerable OS begins a new t_p interval each time it receives a trigger pulse.

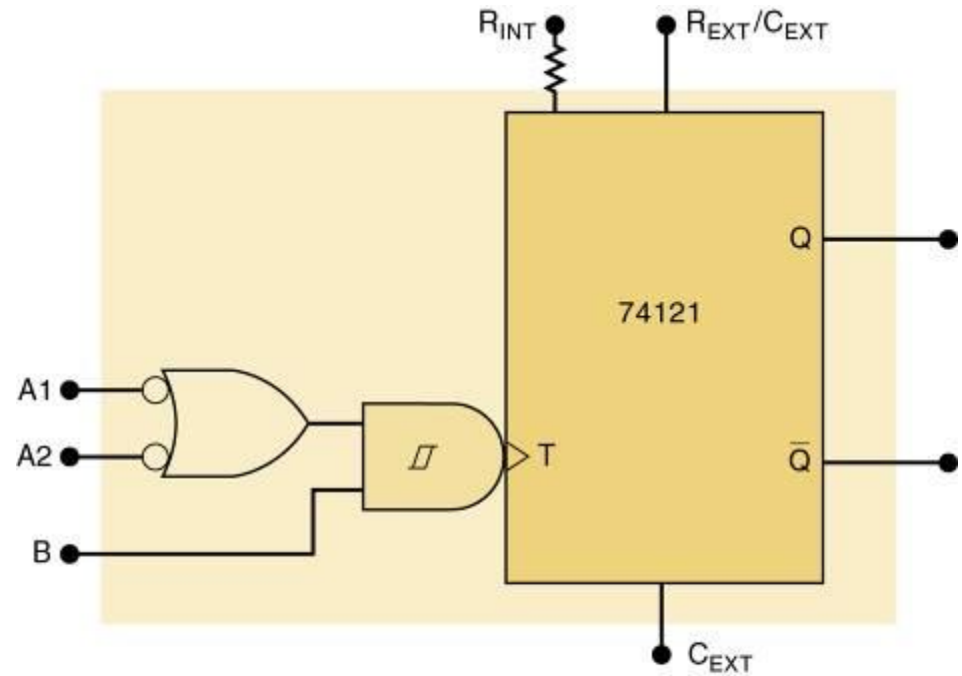


74121 nonretriggerable one-shot IC.

Contains internal logic gates to allow inputs A_1 , A_2 , and B to trigger OS.

Input B is a Schmitt-trigger—allowed to have slow transition times & still reliably trigger OS.

Pins R_{INT}, R_{EXT}/C_{INT}, , and C_{EXT} connect to an external resistor & capacitor to achieve desired output pulse duration.

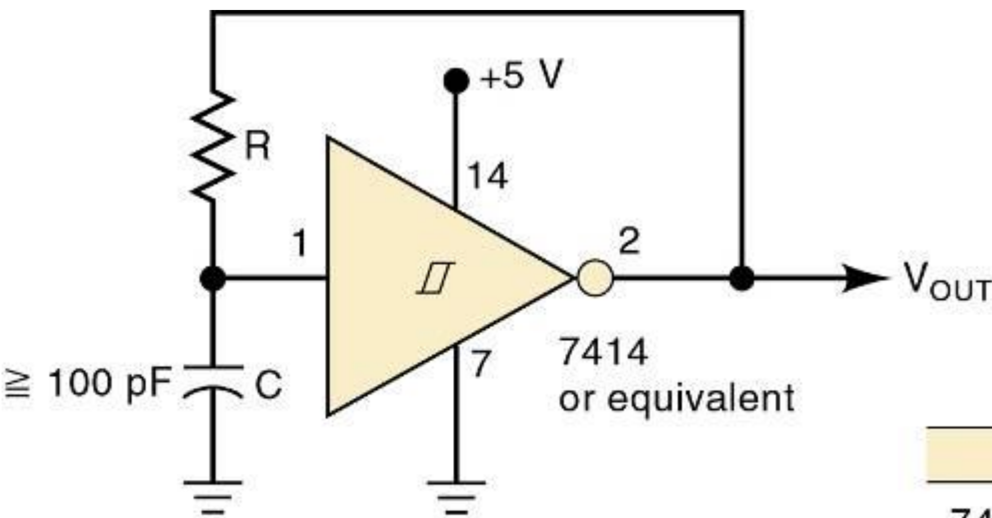


5-22 Clock Generator Circuits

- A third type multivibrator has no stable states—an **astable** or **free-running multivibrato**r.
  - Astable or free-running multivibrators switch back and forth between two unstable states.
  - Useful for generating clock signals for synchronous circuits.

Schmitt-trigger oscillator using a 7414 INVERTER.

A 7413 Schmitt-trigger NAND may also be used.



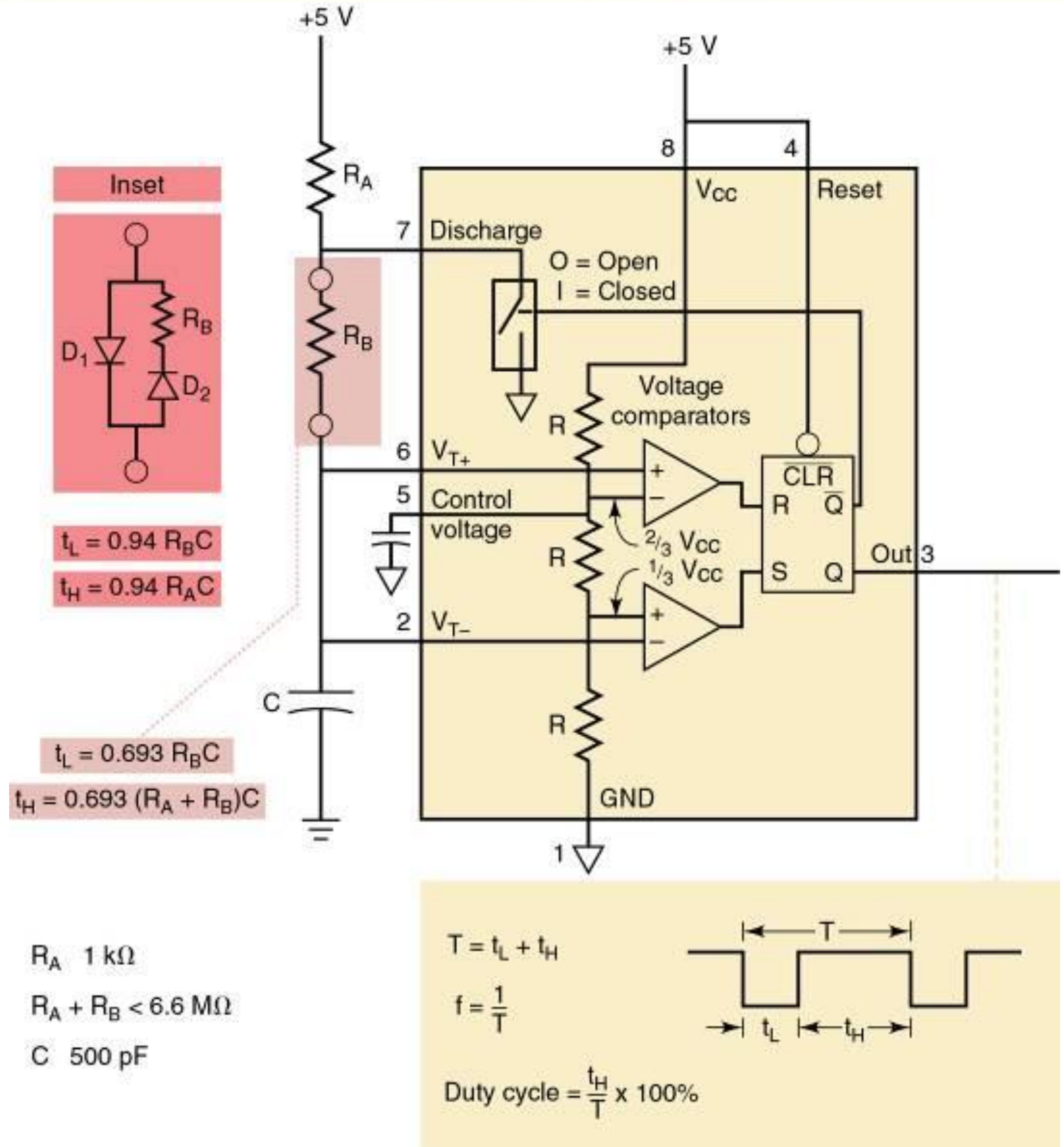
IC	Frequency	
7414	≈ 0.8/RC	(R ≦ 500 Ω)
74LS14	≈ 0.8/RC	(R ≦ 2 kΩ)
74HC14	≈ 1.2/RC	(R ≦ 10 MΩ)

5-22 Clock Generator Circuits

- The **555 timer** IC is a TTL-compatible device that can operate in several different modes.
  - Output is a repetitive rectangular waveform that switches between two logic levels.
  - The time intervals at each logic level are determined by the *R* and *C* values.
- The heart of the 555 timer is two voltage comparators and an S-R latch.
  - The comparators produce a HIGH out when voltage on the (+) input is greater than on the (-) input.

5-22 Clock Generator Circuits

555 Timer IC used as an astable multivibrator.



5-22 Clock Generator Circuits

- Crystal control may be used if a very stable clock is needed—used in microprocessor systems and microcomputers where accurate timing intervals are essential.

5-23 Troubleshooting Flip-Flop Circuits

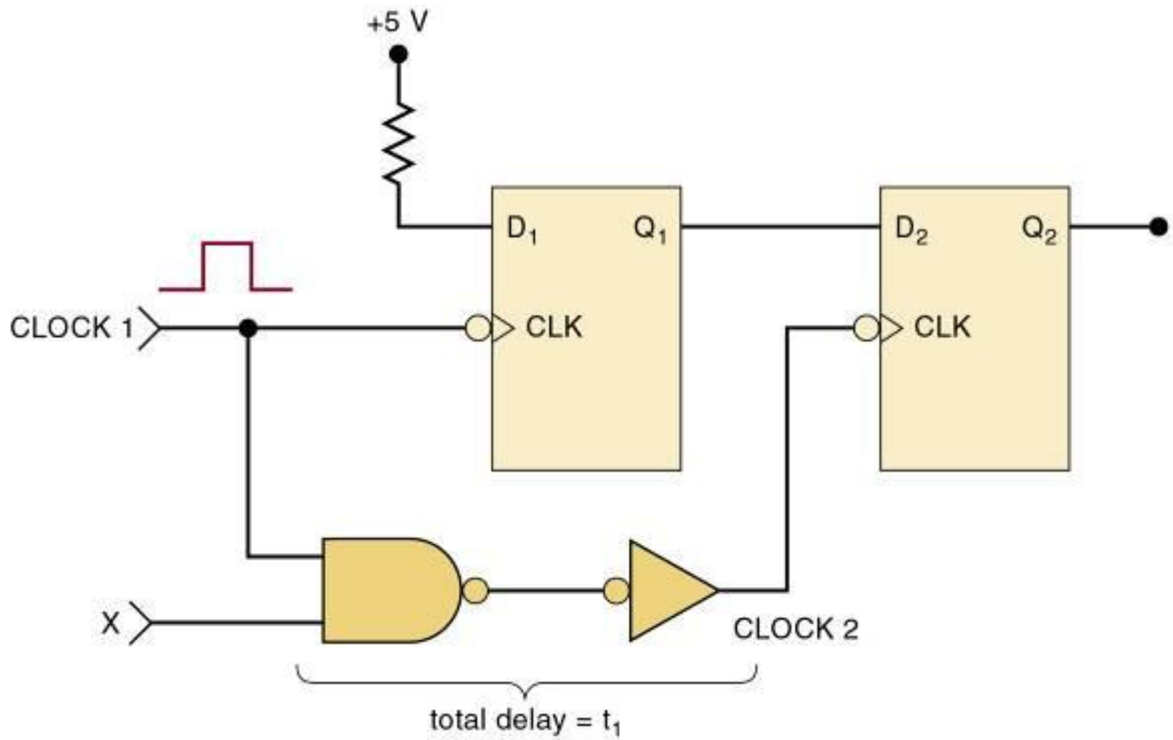
- FFs are subject to the same faults that occur in combinational logic circuits.
  - Timing problems create some faults and symptoms that are not seen in combinational logic circuits.
- Unconnected or floating inputs are particularly susceptible spurious voltage fluctuations—*noise.*
- Given sufficient noise amplitude and duration, logic circuit output may change states in response.
  - In a logic gate, output will return to its original state when the noise signal subsides.
  - In a FF, output will remain in its new state due to its memory characteristic.

5-23 Troubleshooting Flip-Flop Circuits

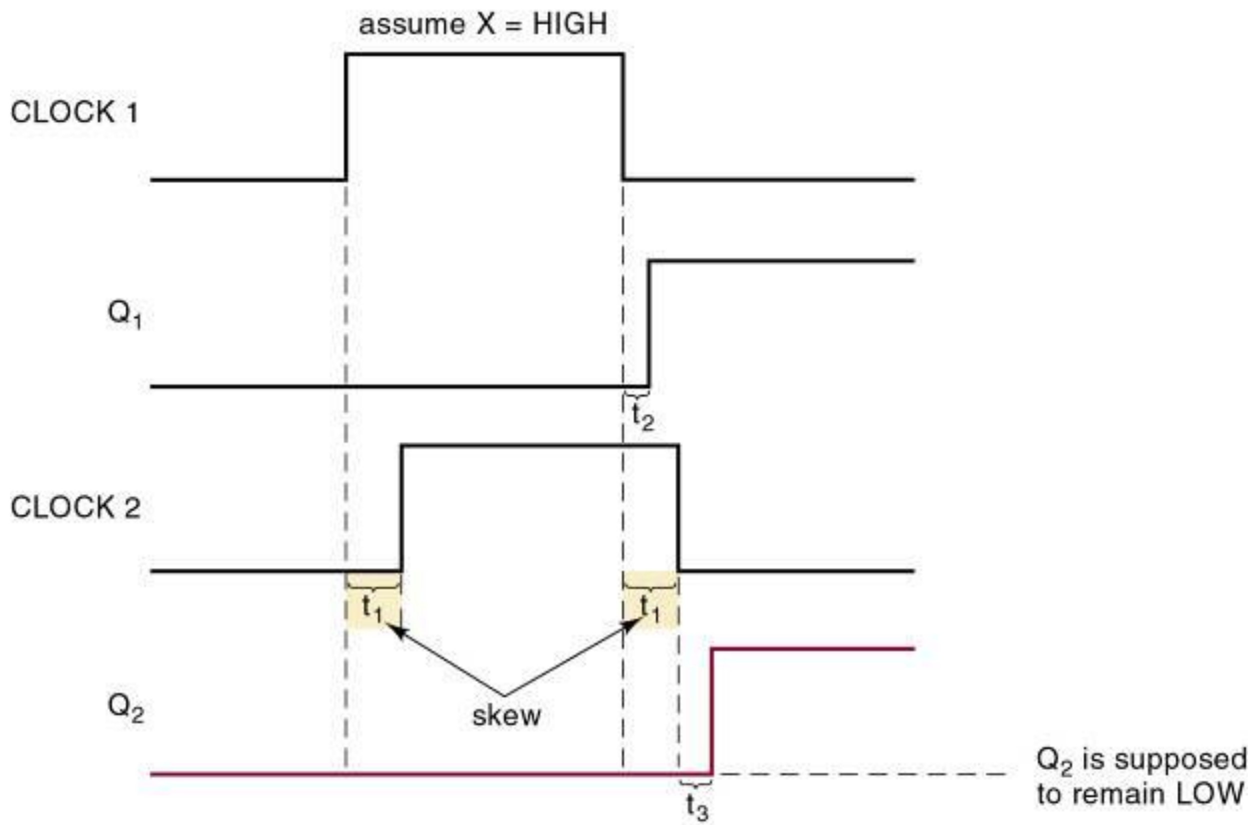
- Clock skew occurs when CLK signals arrive at different FFs at different times.
  - The fault may be seen only intermittently, or may disappear during testing.

5-23 Troubleshooting Flip-Flop Circuits

Extra gating circuits can cause clock skew.



Extra gating circuits can cause clock skew.



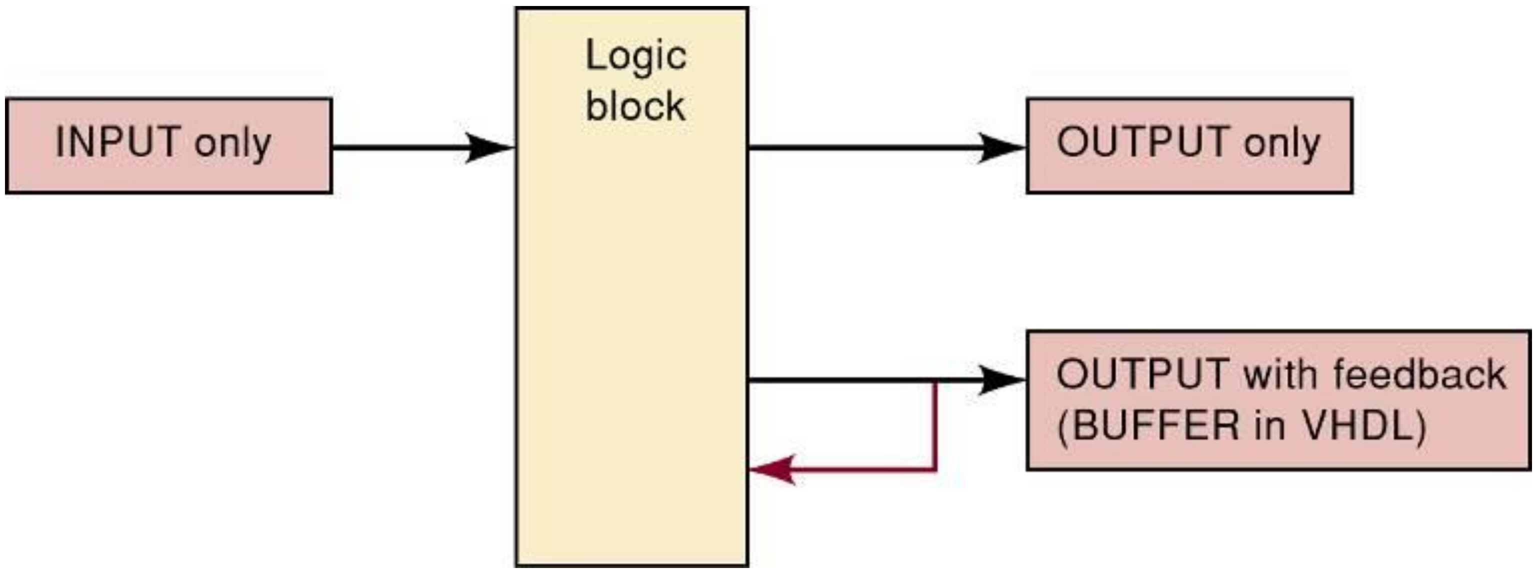
t_1 = skew = combined delay of NAND gate and INVERTER
$t_2 = t_{PLH}$ of Q_1
$t_3 = t_{PLH}$ of Q_2

5-24 Sequential Circuits In PLDs Using Schematic Entry

- Altera's Quartus II development system software allows designers to describe the desired circuit using schematics.
  - The megafunction library contains high-level modules that can be used to create logic designs.
- The Quartus II simulator can be used to verify the sequential circuits by schematic capture before you program a PLD.

5-25 Sequential Circuits Using HDL

- Most PLDs have the ability to feed back the output signal to the input circuitry—to accommodate latching action.
  - The port bit is an output with feedback.

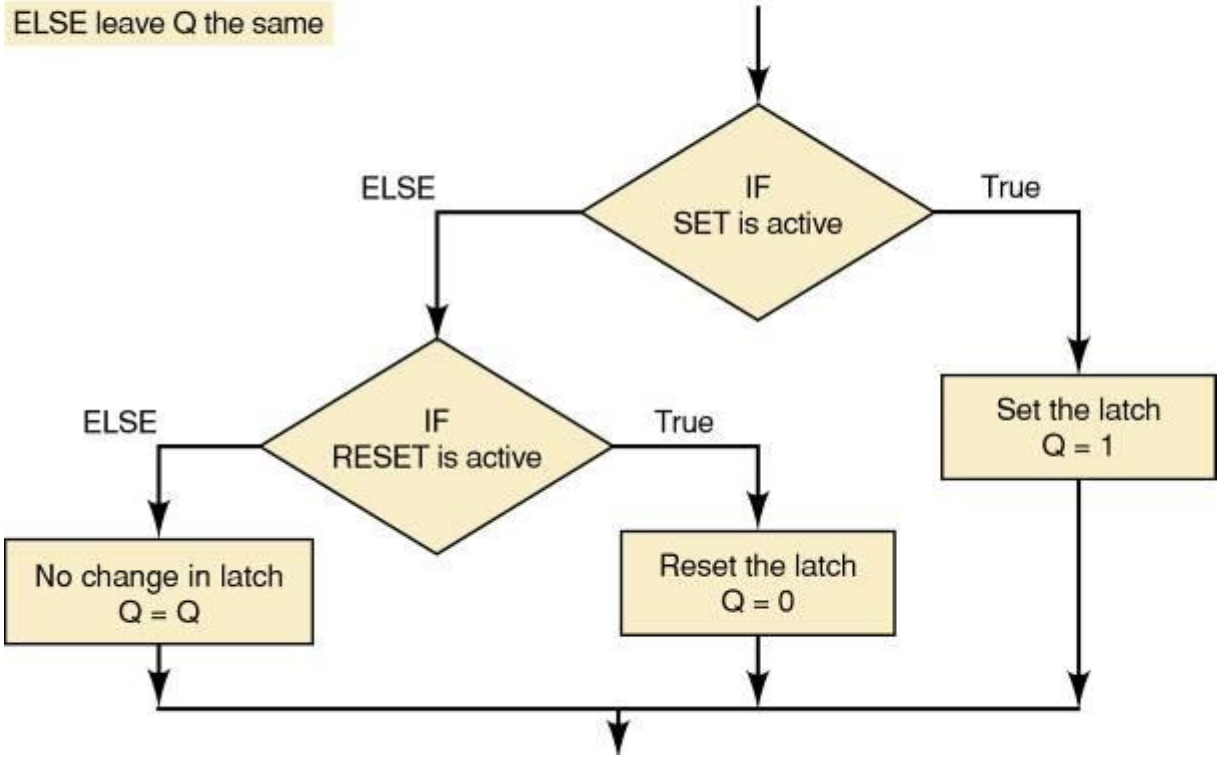


The logic of a behavioral description of an S-R latch.

IF (SET is active) THEN Make Q HIGH

ELSE IF (RESET is active) THEN Make Q LOW

ELSE leave Q the same



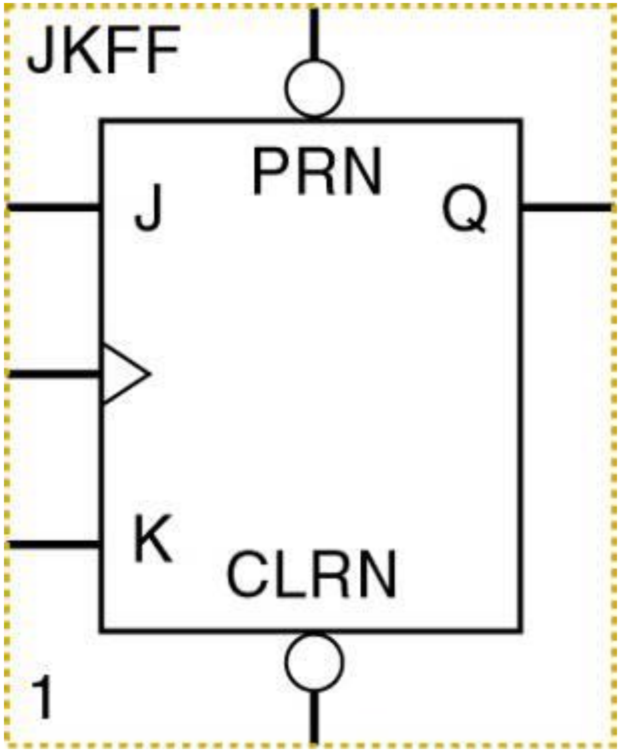
5-25 Sequential Circuits Using HDL

- Sequential circuits that feed output value back to inputs, may possibly create an unstable system.
  - A change in the output state might be fed back to the inputs, which changes the output state again, which feeds back to the inputs, which changes the output back again….
- It is very important to make sure no combination of inputs & outputs can make this undesirable oscillation undesirable happen.

5-26 Edge Triggered Devices

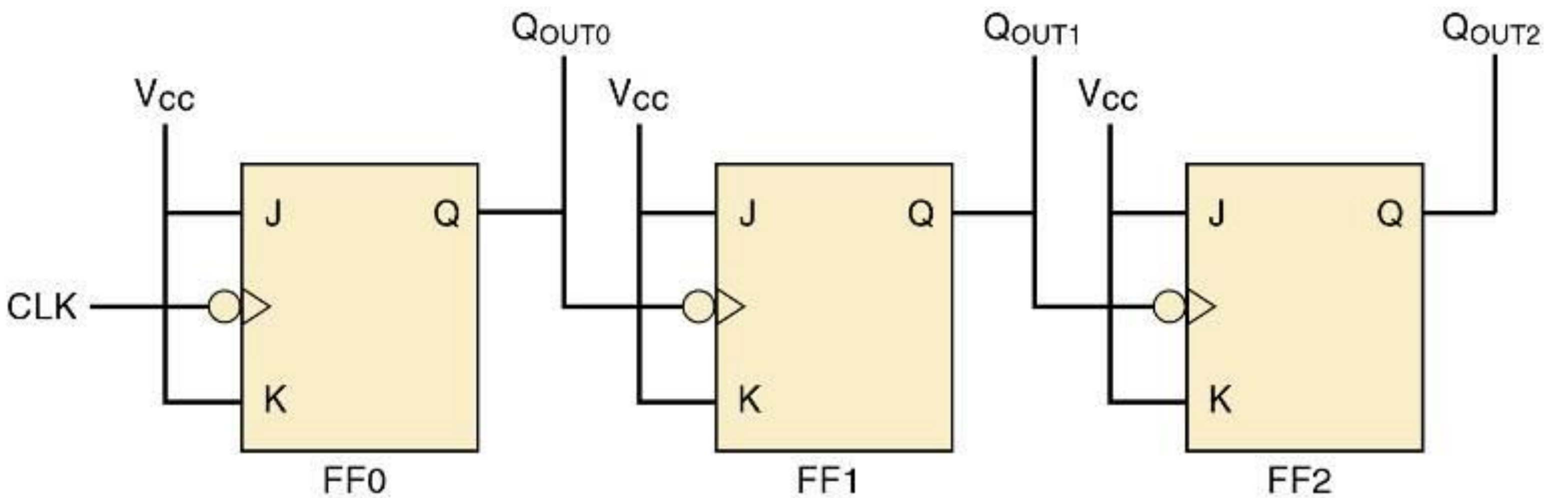
- Edge-triggered device output respond to the inputs when the clock input sees an “edge.”
  – An edge is a transition from HIGH to LOW, or vice versa—and is often referred to as an **event**.

The J-K flip-flop is a standard building block of clocked (sequential) logic circuits known as a **logic primitive**.



5-27 HDL Circuits with Multiple Components

A three-bit binary counter.

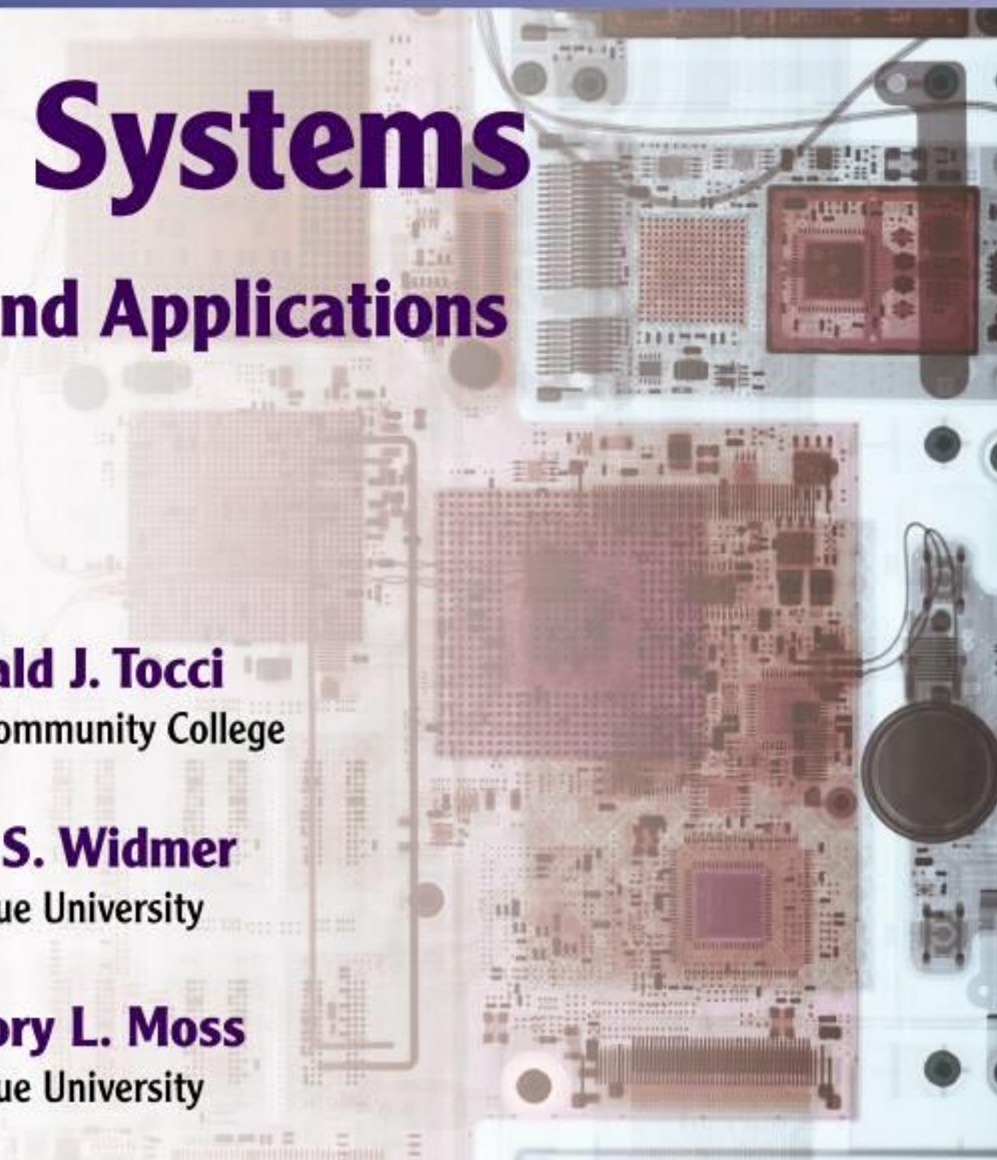


These logic symbols are negative edge-triggered.

These flip-flops do not have asynchronous inputs prn or clrn.

END

ELEVENTH EDITION



Digital Systems

Principles and Applications

Ronald J. Tocci
Monroe Community College

Neal S. Widmer
Purdue University

Gregory L. Moss
Purdue University

PEARSON

หา แปลง ฐาน 2 → 10

ดึงลงตัวแรก ×2

loop บวก ×2

1 0 1 0 1 0 0 0 1

1

×2

=

2 + 2 = 4

×2

4 + 1 = 5

5 ×2

10

118

336

337

42 ×8

336

339

$2^0 = 1$	$2^{10} = 1024 = 1k$		1 = k
$2^1 = 2$	$2^{11} = 2048 = 2k$		2 = m
$2^2 = 4$	$2^{12} = 4096 = 4k$		3 = G
$2^3 = 8$	$2^{13} = 8192 = 8k$		4 = T
$2^4 = 16$	$2^{14} = 16k$		5 = P
$2^5 = 32$	$2^{15} = 32k$		
$2^6 = 64$	$2^{16} = 64k$		
$2^7 = 128$	$2^{17} = 128k$		
$2^8 = 256$	$2^{18} = 256k$		
$2^9 = 512$	$2^{19} = 512k$		

$A \quad \bar{A} \quad 0 \quad 1$

กฎ 11 ข้อ

$A \cdot 0 = 0$ | $A + 0 = A$

$A \cdot 1 = A$ | $A + 1 = 1$

$A \cdot A = A$ | $A + A = A$

$A \cdot \bar{A} = 0$ | $A + \bar{A} = 1$

$A + \cancel{\bar{A}}B$ | $A(B + C) = AB + AC$

$\bar{A} + \cancel{A}B$

$\bar{\bar{A}} = A$

D' Morgan

$\overline{AB} = \bar{A} + \bar{B}$

$\overline{A + B} = \bar{A} \cdot \bar{B}$

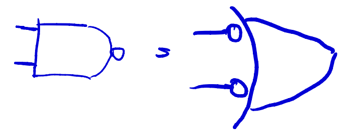
ตัด บาร์ครึ่ง เป็น or → and

and → or

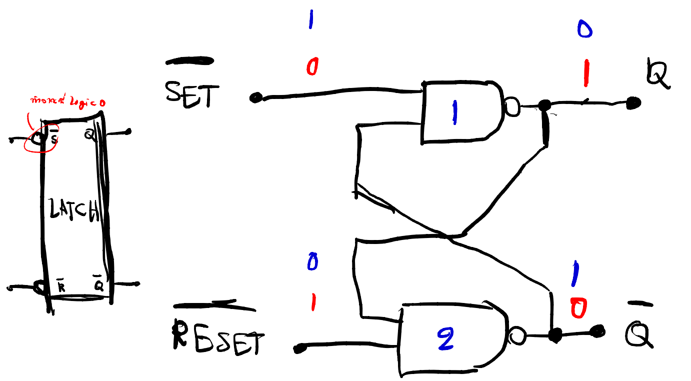
Flip-flops เรียกว่า latch ได้, bistable multivibrator

Q เป็น 1 เรียกว่า SET state

Q เป็น 0 เรียกว่า RESET state, CLEAR

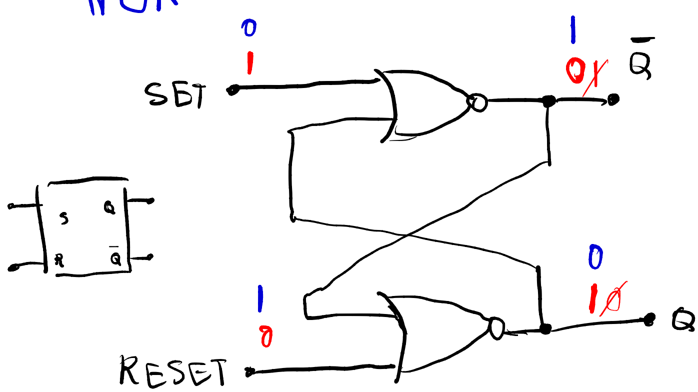


NAND



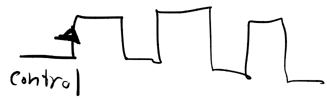
ค่า	$\overline{SET}$	$\overline{RESET}$	Q | $\bar{Q}$
	1	1	ได้ค่าเดิม
	0	1	Q = 1
	1	0	Q = 0
	0	0	Q=1, $\bar{Q}$=1 ไม่ใช้งาน

NOR



SET	RESET	Q | $\bar{Q}$
0	0	ได้ค่าเดิม
1	0	Q = 1
0	1	Q = 0
1	1	Q = 0, $\bar{Q}$=0 ไม่ใช้

Synchronous sys output จะเปลี่ยน ตามช่วงเวลาที่กำหนด



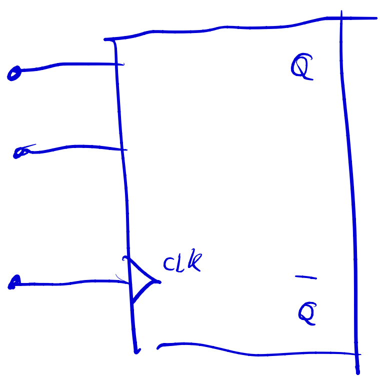
Asynchronous sys output จะเปลี่ยนได้ตลอด ไม่รอ clk

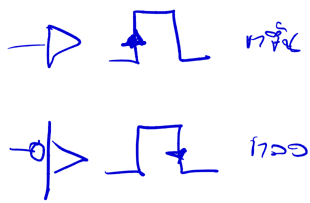
เรียก control → clock

Flip-Flops ต้องมี clk

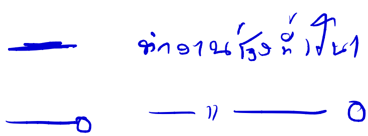
ถ้าไม่มี clk เรียก LATCH

Clk สัญญาณ ซ้ำๆ ที่เปลี่ยนจาก 0 1 แบบเหลี่ยม





edge triggering

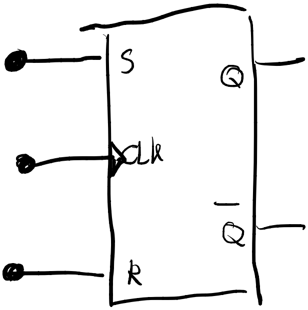


Level triggering

S – R ff

R – S ff

Reset set



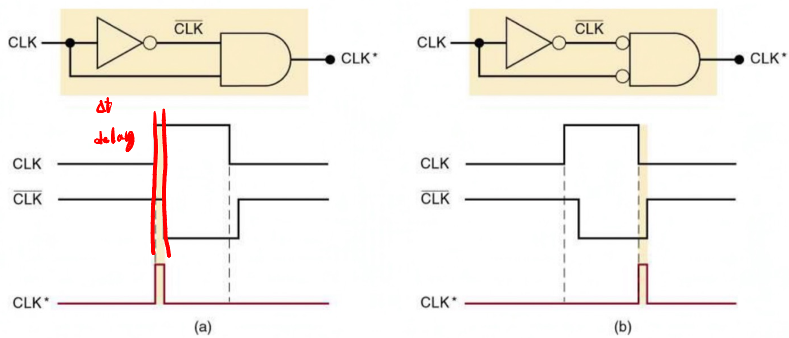
S	R	CLK	Q
0	0		Q_0 ไม่เปลี่ยน
1	0	↑	1
0	1		0
1	1		Ambiguous

Q_0 เดิม output CLK ที่ ↑

↓ CLK no change ที่ Q

5-6 Clocked S-R Flip-Flop – Internal Circuitry

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



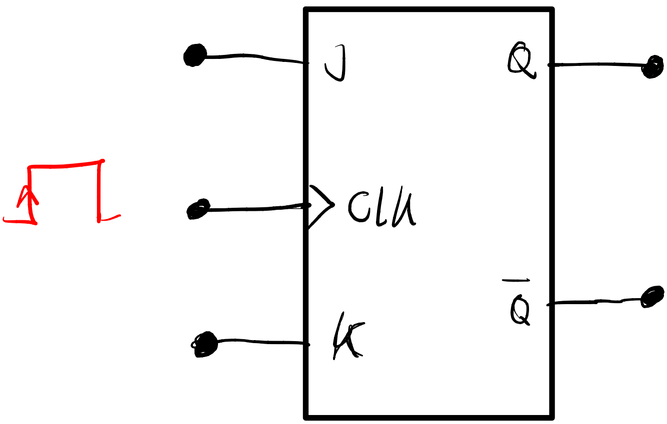
The duration of the *CLK** pulses is typically 2–5 ns.

JK ff

J → SET

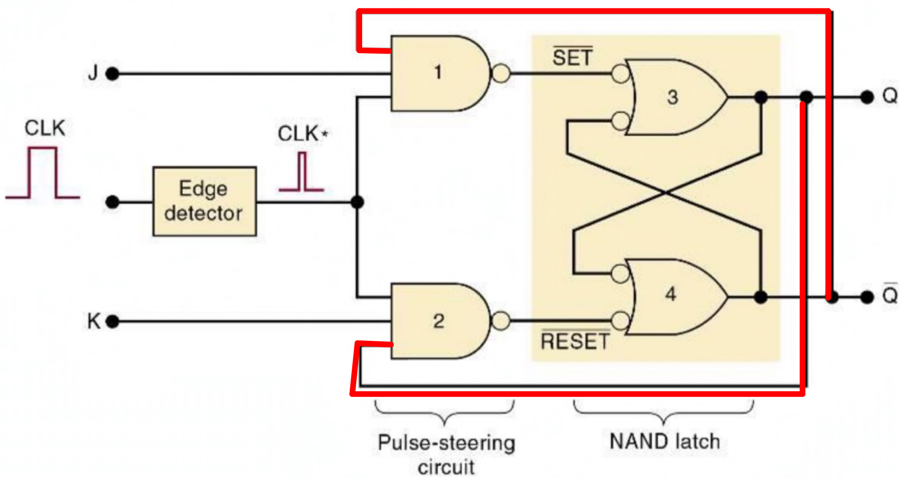
K → CLEAR

Toggle 1 → 0
  0 → 1



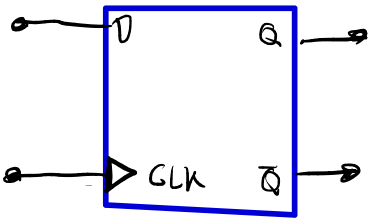
J	K	Clk	Q
0	0		Q_0 (ไม่เปลี่ยน)
1	0	↑	1
0	1		0
1	1	↓	$\bar{Q}_0$ (Toggle)

R-S ff + 2 เส้น

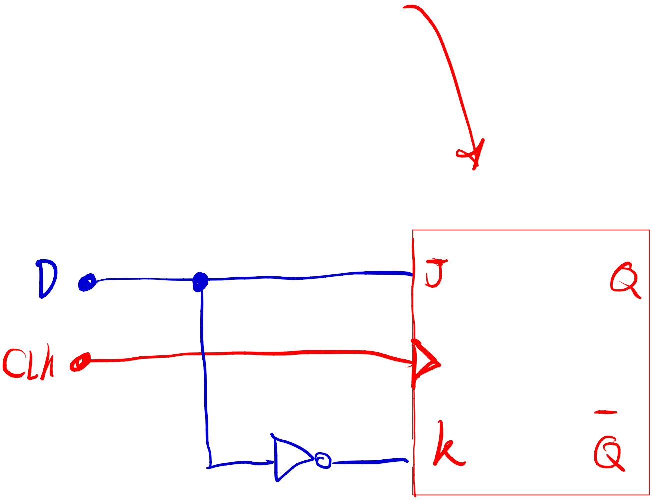


J
K
CLK
CLK*
Edge detector
1
2
3
4
SET
RESET
Q
Q̄
Pulse-steering circuit
NAND latch

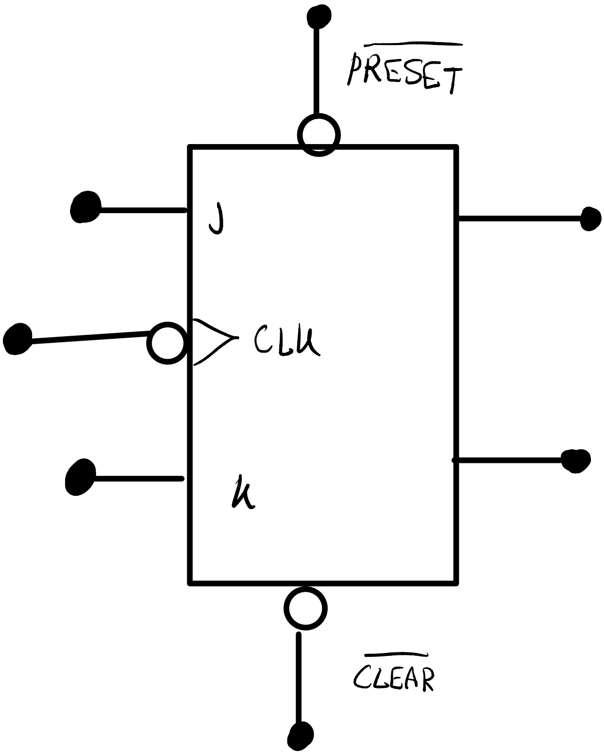
D ff



D	CLK	Q
0	↑	0
1		1

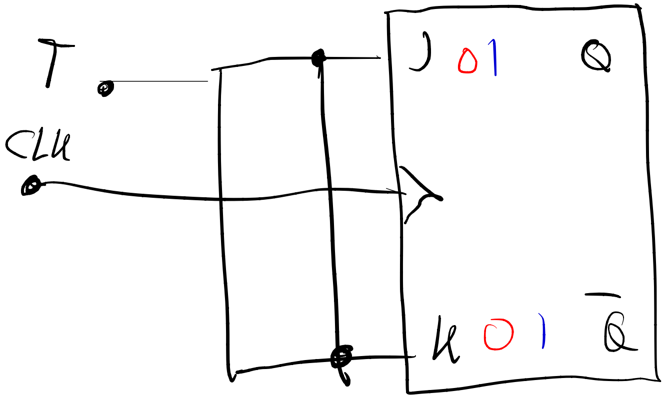


Clocked J-K ff asynchronous input



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

T ff



Toggle ff

1 → Invert

↑ ขอบclock

0 → ค่าเดิม

↑ ขอบclock