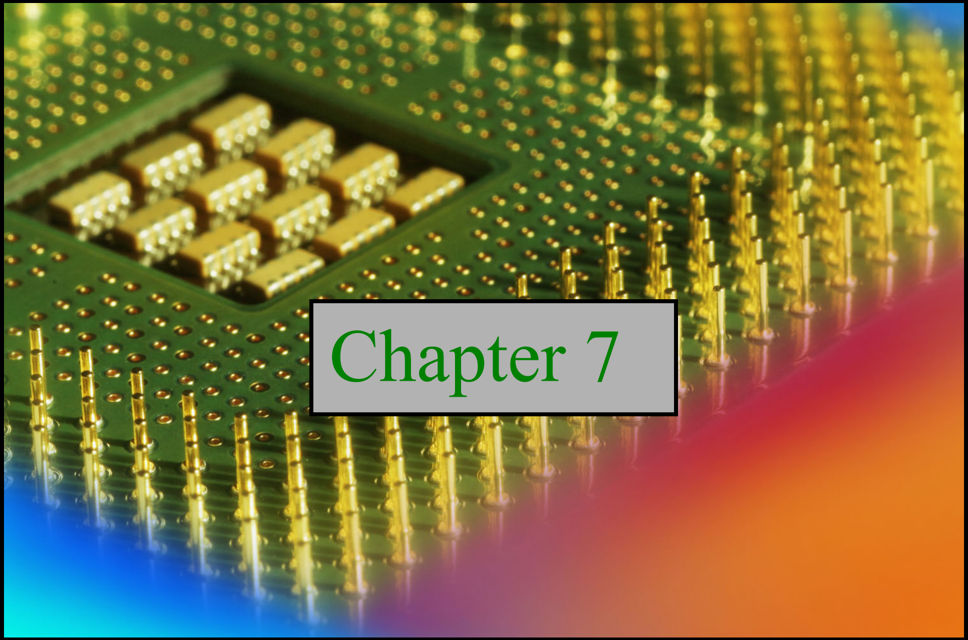
# Digital Fundamentals

Tenth Edition

Floyd



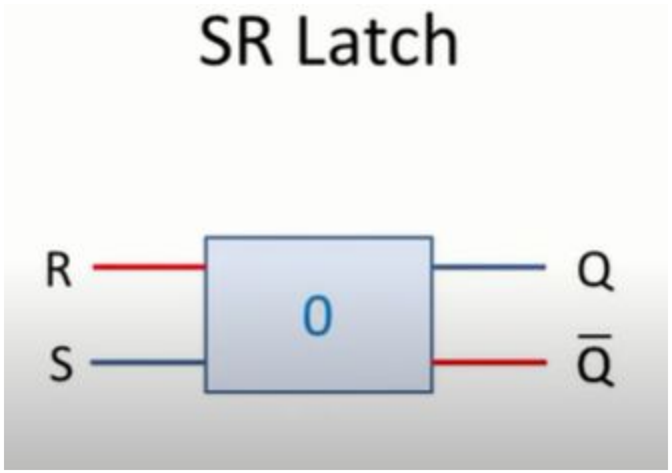
Chapter 7

# Summary

## Latches

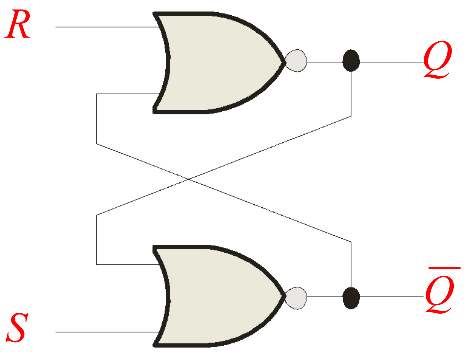
A **latch** is a temporary storage device that has two stable states (bistable). It is a basic form of memory. (1-bit memory)

The S-R (Set-Reset) latch is the most basic type. It can be constructed from NOR gates or NAND gates. With NOR gates, the latch responds to active-HIGH inputs; with NAND gates, it responds to active-LOW inputs.



# Summary

## Latches



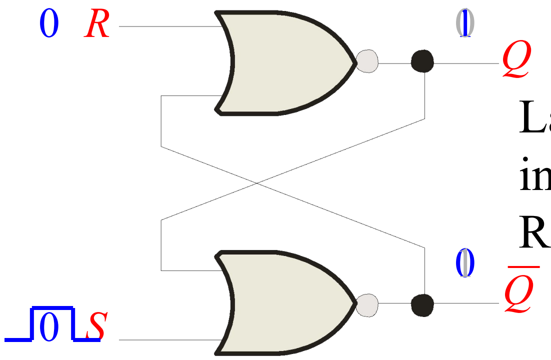
NOR Active-HIGH Latch

# Summary

## Latches

The active-HIGH *S*-*R* latch is in a stable (latched) condition when both inputs are LOW.

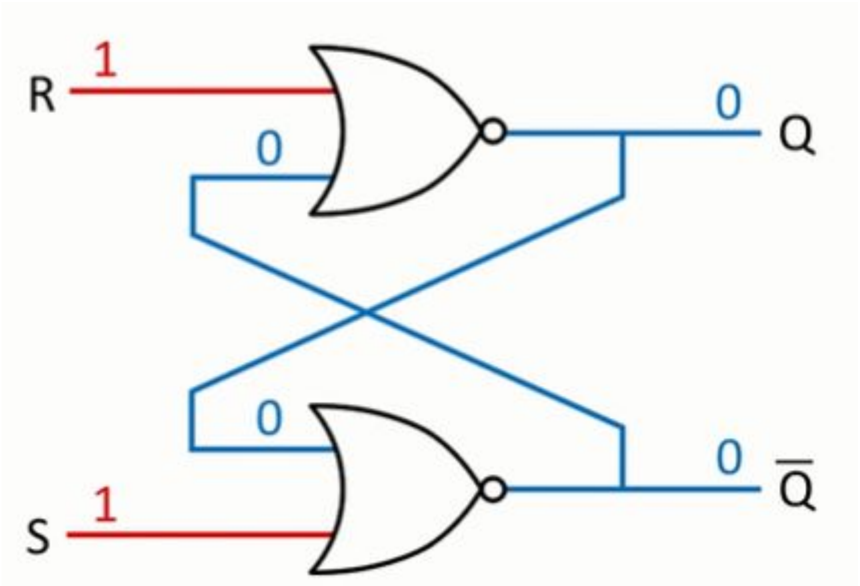
Assume the latch is initially RESET ($Q = 0$) and the inputs are at their inactive level (0). To SET the latch ($Q = 1$), a momentary HIGH signal is applied to the *S* input while the *R* remains LOW.

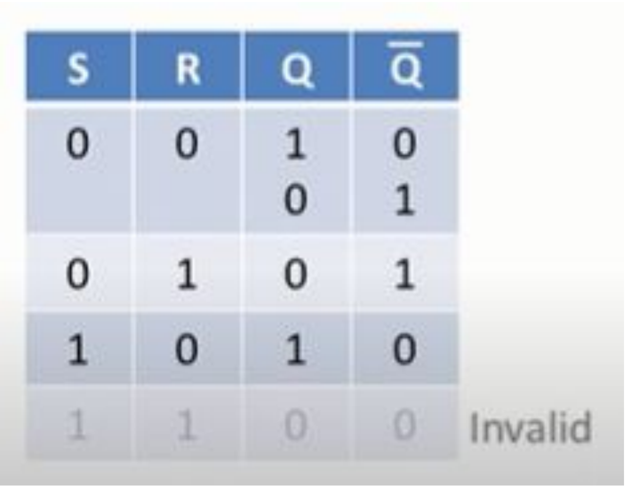


Latch initially RESET

To RESET the latch ($Q = 0$), a momentary HIGH signal is applied to the *R* input while the *S* remains LOW.

# What Happens when Both SET and RESET Pulses are Applied?





| S | R | Q | $\overline{Q}$ | |
|---|---|---|---|---|
| 0 | 0 | 1<br>0 | 0<br>1 | |
| 0 | 1 | 0 | 1 | |
| 1 | 0 | 1 | 0 | |
| 1 | 1 | 0 | 0 | Invalid |

# Summary

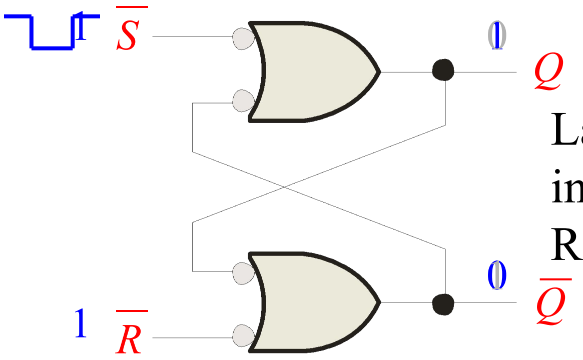
## Latches

The active-LOW $\overline{S}$-$\overline{R}$ latch is in a stable (latched) condition when both inputs are HIGH.

Assume the latch is initially RESET ($Q$ = 0) and the inputs are at their inactive level (1). To SET the latch ($Q$ = 1), a momentary LOW signal is applied to the $\overline{S}$ input while the $\overline{R}$ remains HIGH.

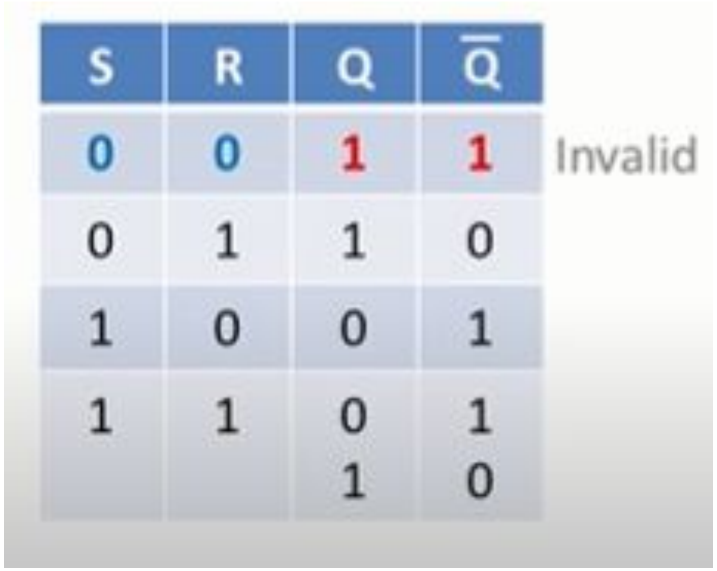
To RESET the latch a momentary LOW is applied to the $\overline{R}$ input while $\overline{S}$ is HIGH.

Never apply an active set and reset at the same time (invalid).



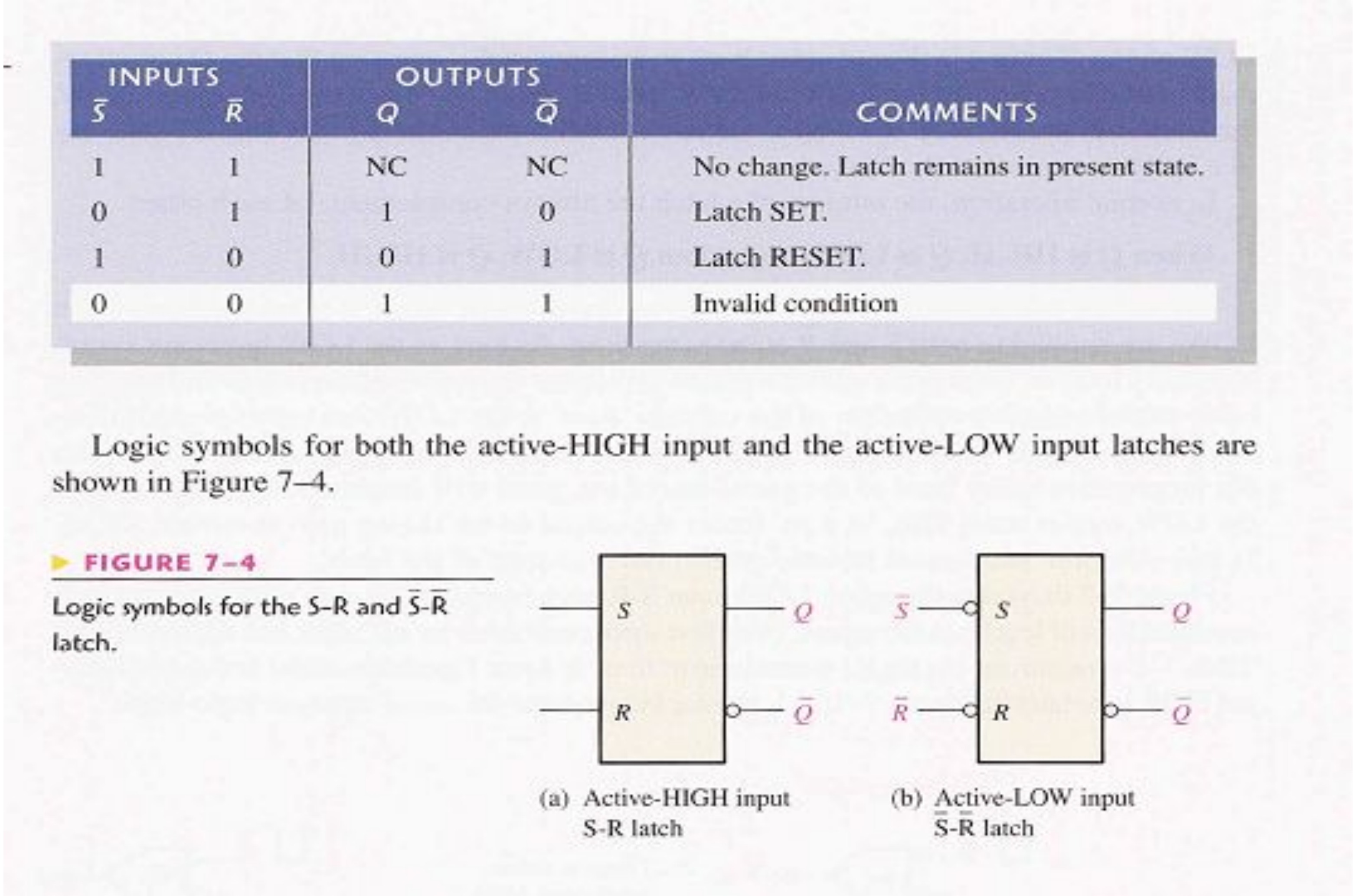
Latch initially RESET

# Truth Table



| S | R | Q | $\overline{Q}$ | |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | Invalid |
| 0 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 1 | |
| 1 | 1 | 0<br>1 | 1<br>0 | |

# Summary



| INPUTS | | OUTPUTS | | COMMENTS |
|---|---|---|---|---|
| $\overline{S}$ | $\overline{R}$ | $Q$ | $\overline{Q}$ | |
| 1 | 1 | NC | NC | No change. Latch remains in present state. |
| 0 | 1 | 1 | 0 | Latch SET. |
| 1 | 0 | 0 | 1 | Latch RESET. |
| 0 | 0 | 1 | 1 | Invalid condition |

Logic symbols for both the active-HIGH input and the active-LOW input latches are shown in Figure 7–4.

FIGURE 7–4

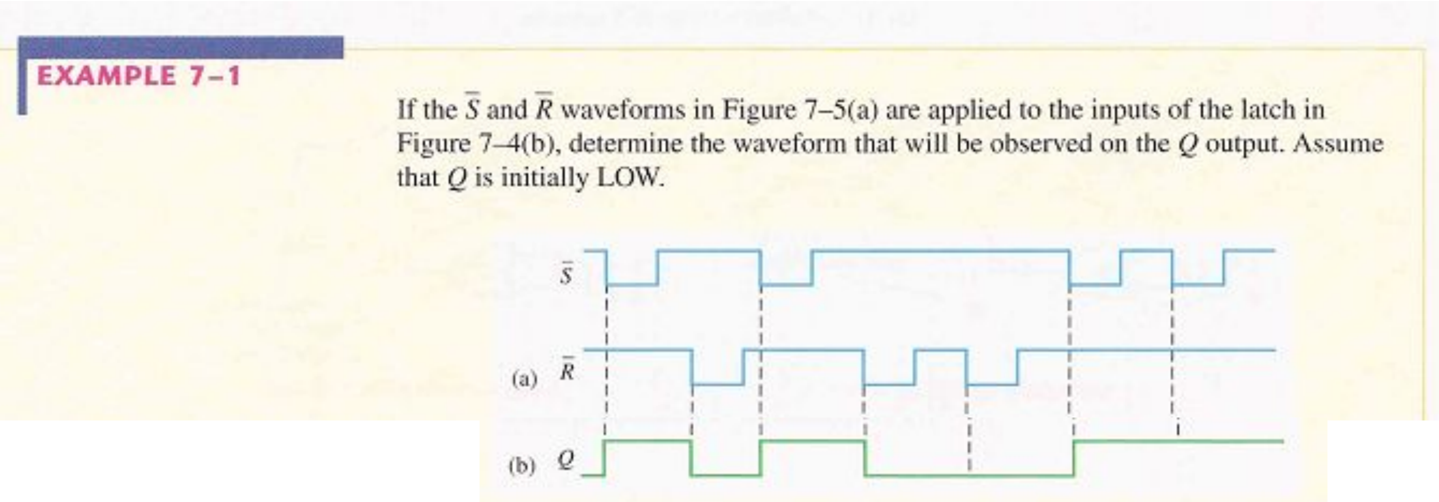
Logic symbols for the S-R and $\overline{S}$-$\overline{R}$ latch.

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

(a) Active-HIGH input S-R latch

(b) Active-LOW input $\overline{S}$-$\overline{R}$ latch

# Summary

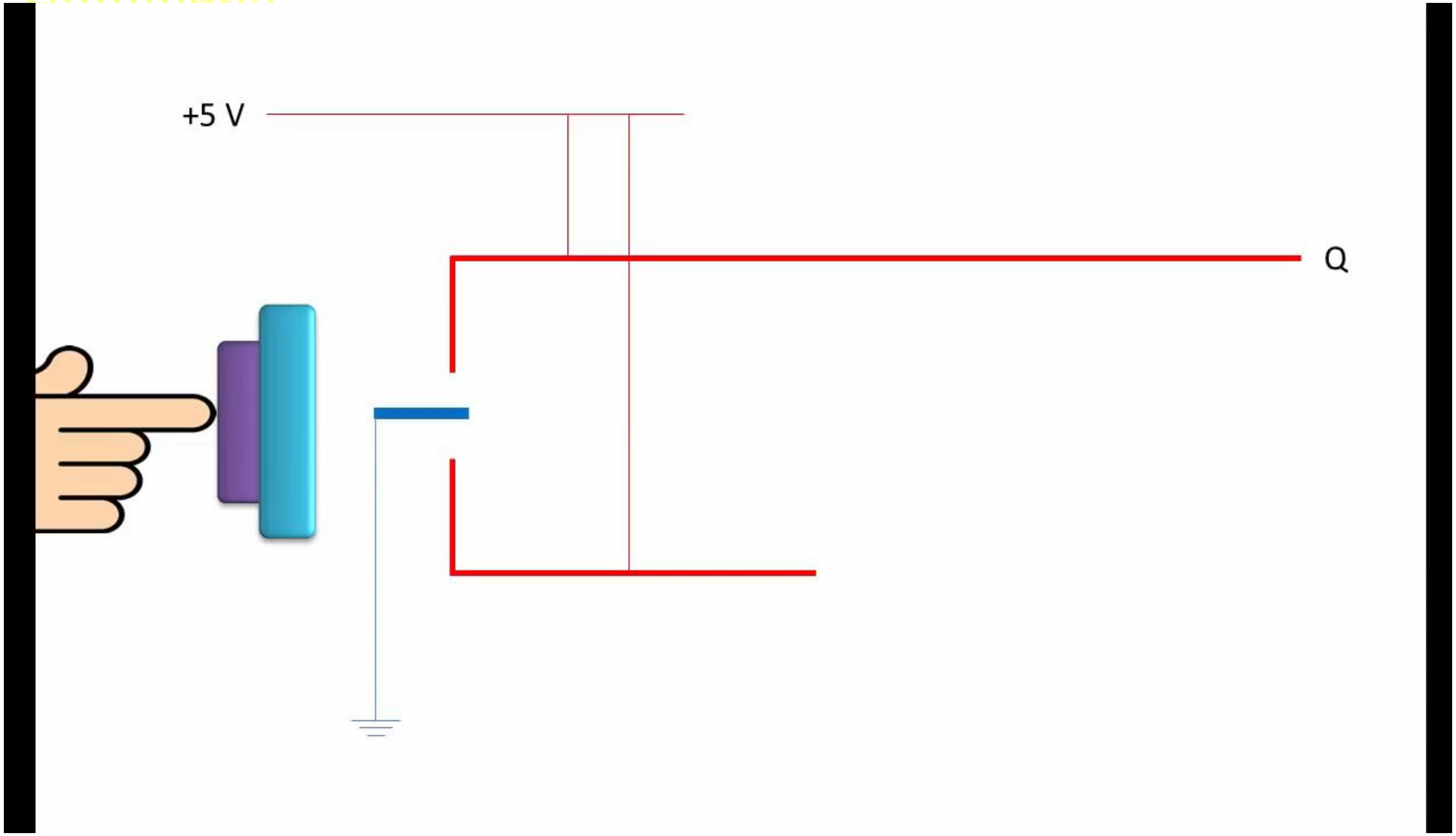


**EXAMPLE 7–1**

If the $\overline{S}$ and $\overline{R}$ waveforms in Figure 7–5(a) are applied to the inputs of the latch in Figure 7–4(b), determine the waveform that will be observed on the $Q$ output. Assume that $Q$ is initially LOW.

# Summary

## Latches Application as Contact Bounce Eliminator

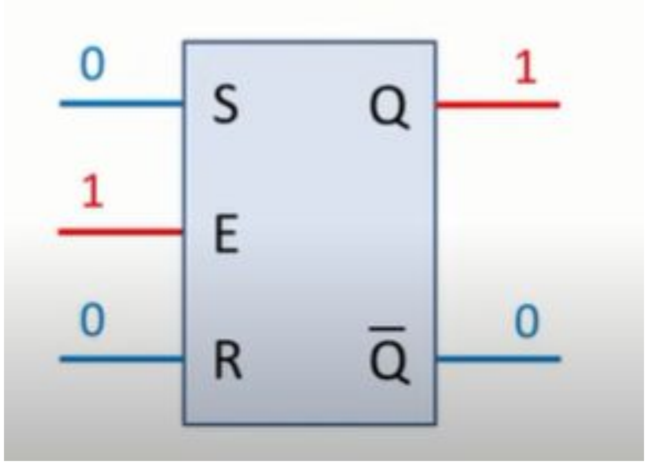


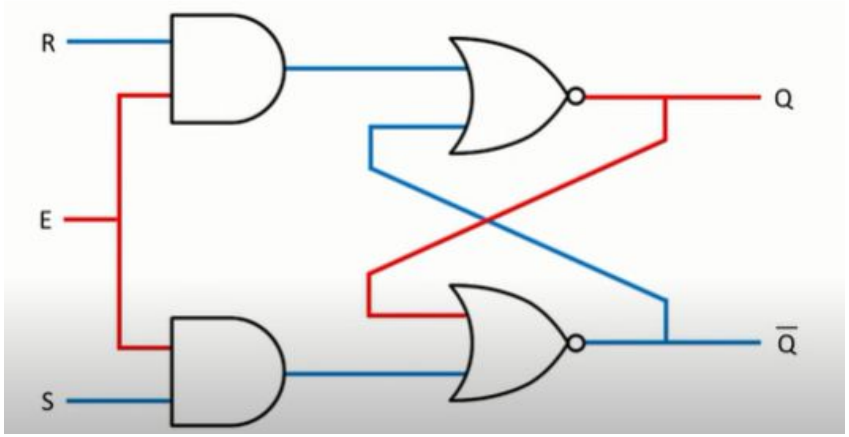
# Summary

## Gated-SR- Latch

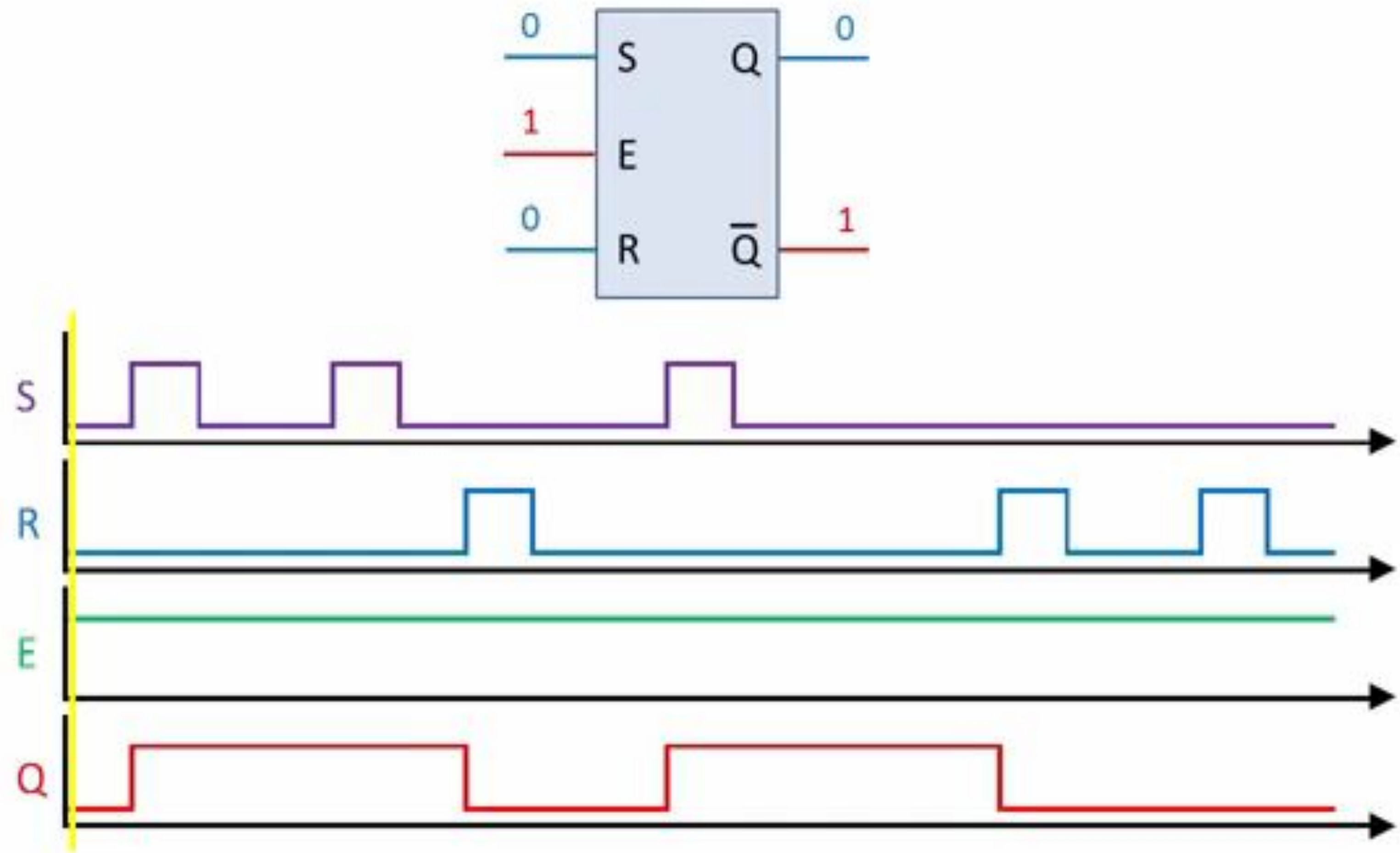
A gated latch is a variation on the basic latch.

The gated latch has an additional input, called enable (*EN*) that must be HIGH in order for the latch to respond to the *S* and *R* inputs.

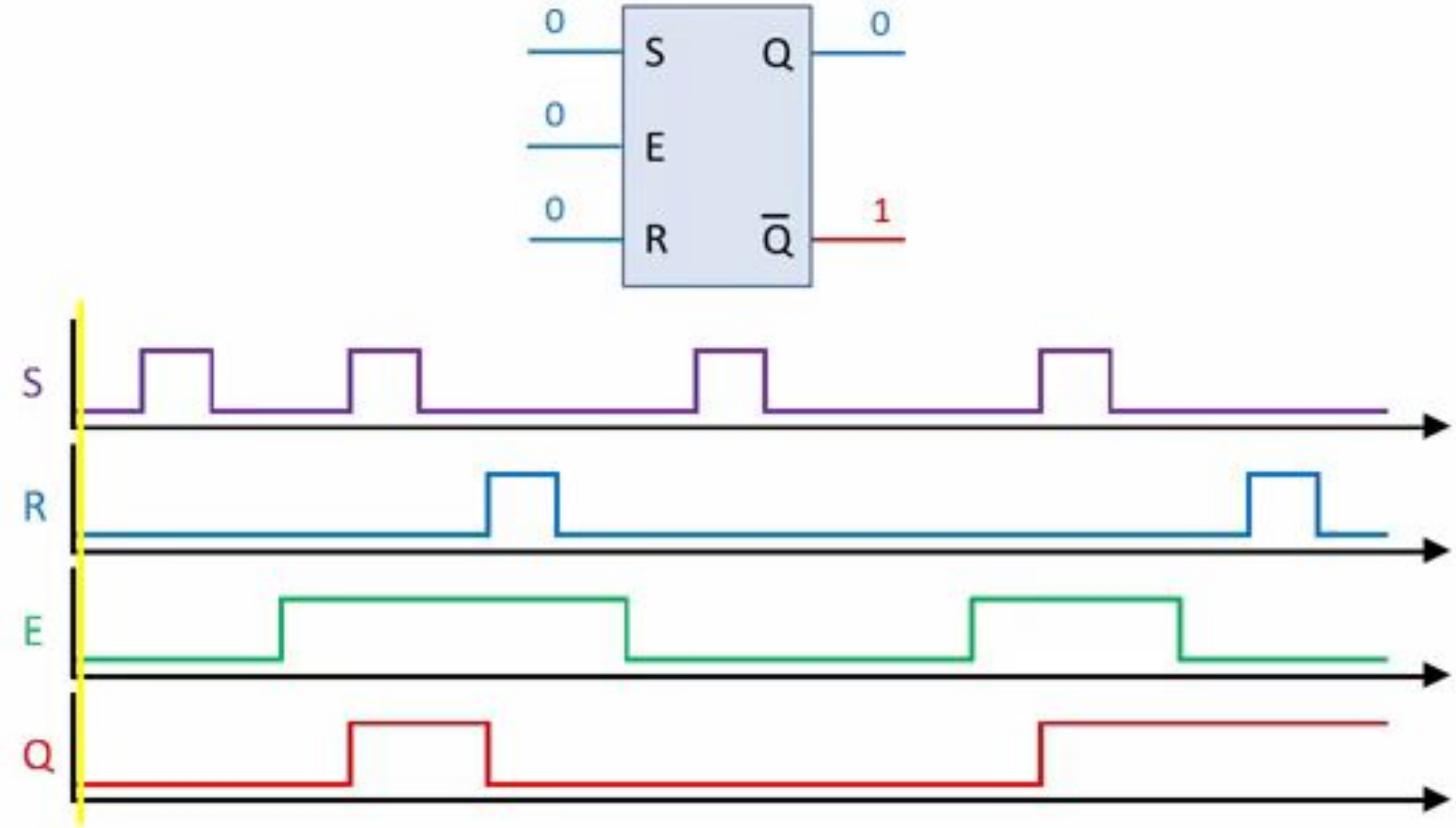




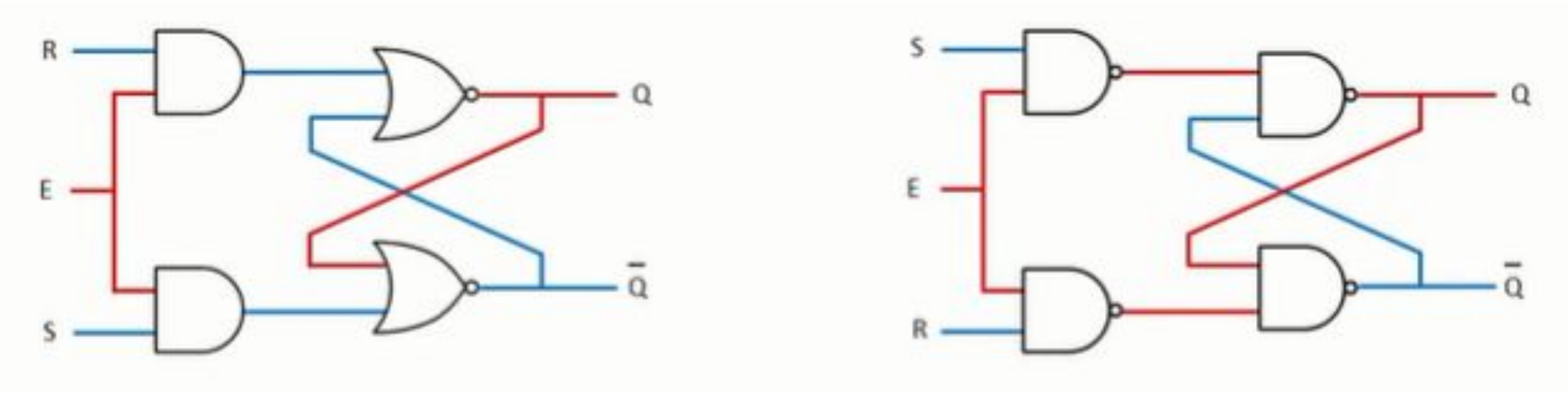
# Gated-SR- Latch



# Gated-SR- Latch

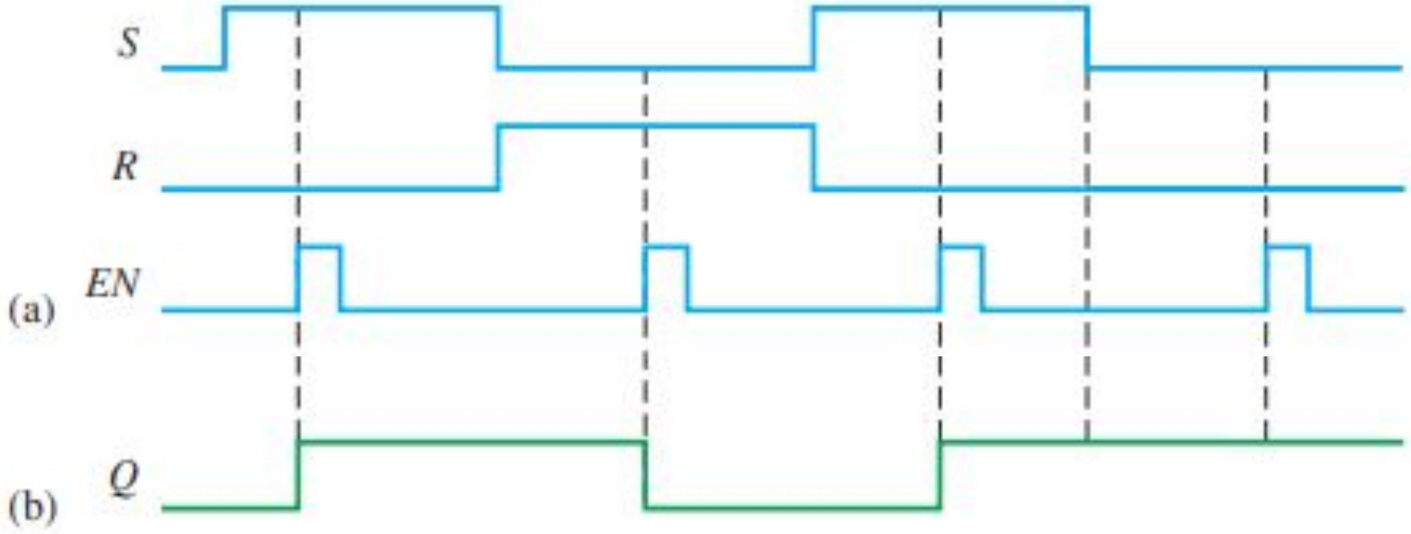


# Gated-SR- Latch



# EXERCISE

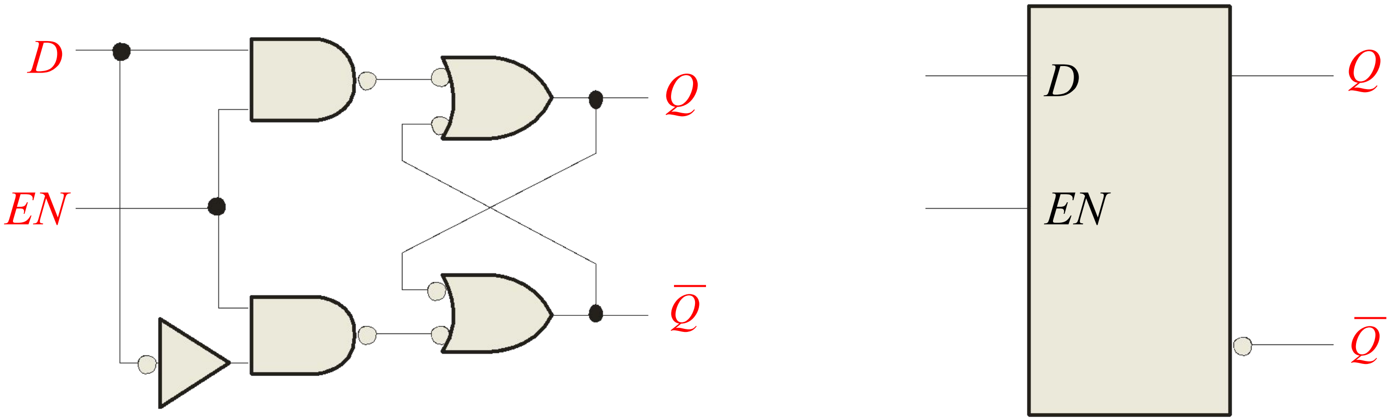
- Determine the Q output waveform if the inputs shown in Figure 7–9(a) are applied to a gated S-R latch that is initially RESET.



# Summary

## D-Latches

The *D* latch is an variation of the *S*-*R* latch but combines the *S* and *R* inputs into a single *D* input as shown:



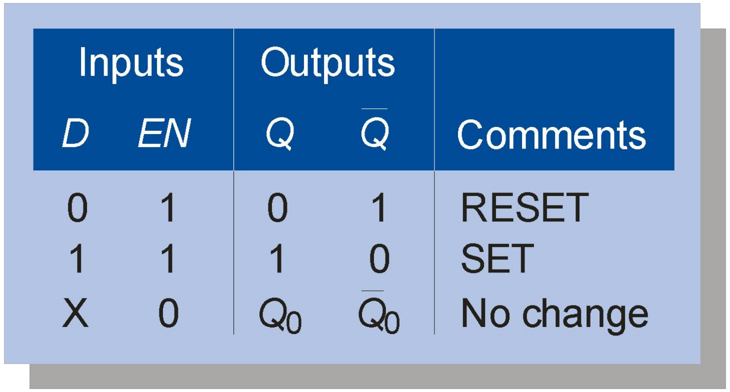
A simple rule for the *D* latch is:

*Q* follows *D* when the Enable is active.

# Summary

## Latches

The truth table for the *D* latch summarizes its operation. If *EN* is LOW, then there is no change in the output and it is latched.



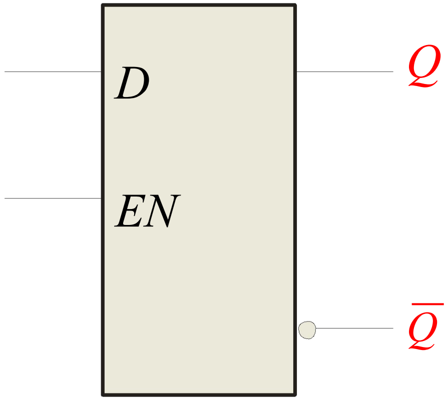
| Inputs | | Outputs | | |
|---|---|---|---|---|
| *D* | *EN* | *Q* | $\overline{Q}$ | Comments |
| 0 | 1 | 0 | 1 | RESET |
| 1 | 1 | 1 | 0 | SET |
| X | 0 | $Q_0$ | $\overline{Q}_0$ | No change |

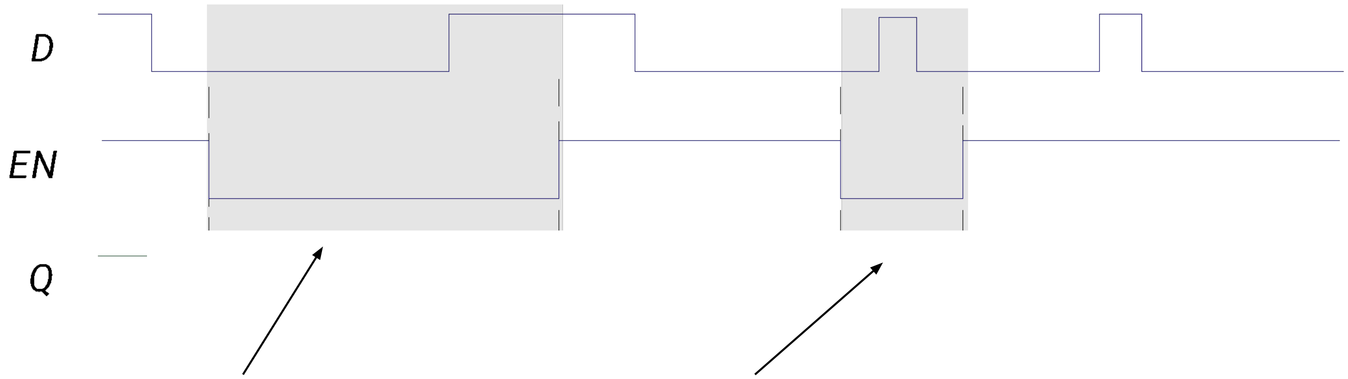
# Summary

## Latches

**Example**

Determine the *Q* output for the *D* latch, given the inputs shown.





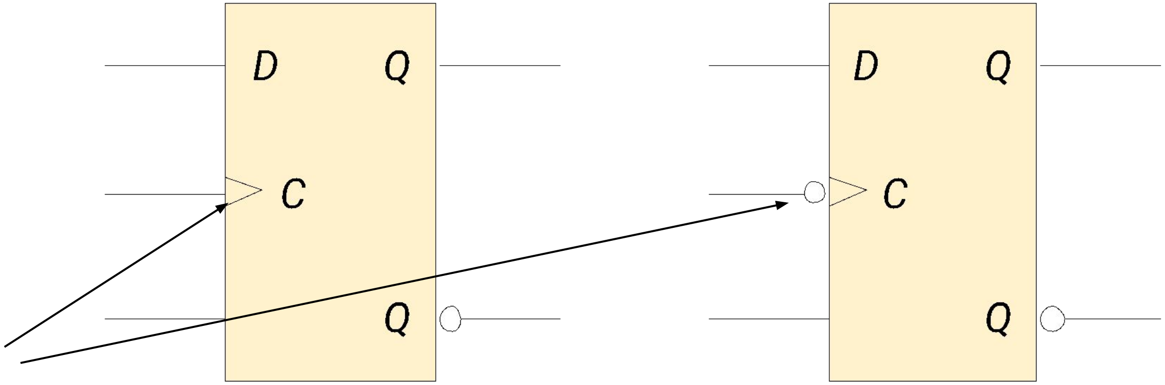
Notice that the Enable is not active during these times, so the output is latched.

# Summary

## Flip-flops

A flip-flop differs from a latch in the manner it changes states. A flip-flop is a clocked device, in which only the clock edge determines when a new bit is entered.

The active edge can be positive or negative.



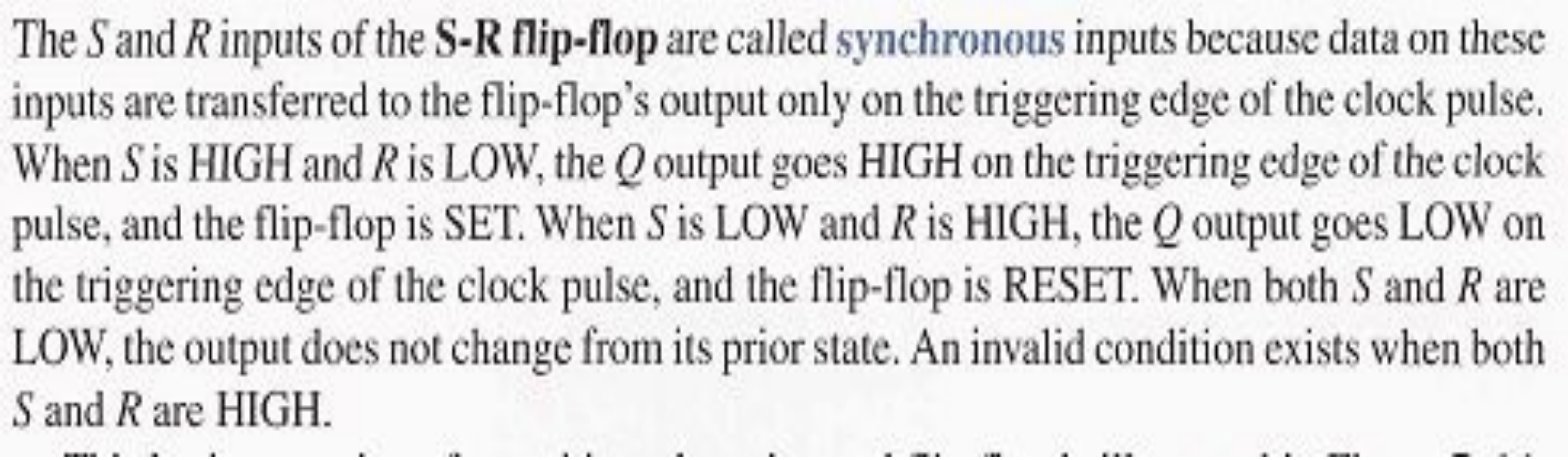
Dynamic input indicator

(a) Positive edge-triggered

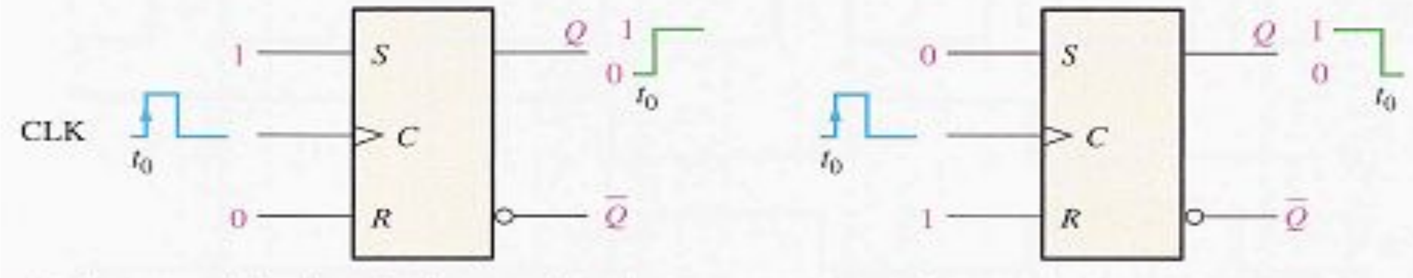
(b) Negative edge-triggerec

# Summary

## S-R Flip-flops



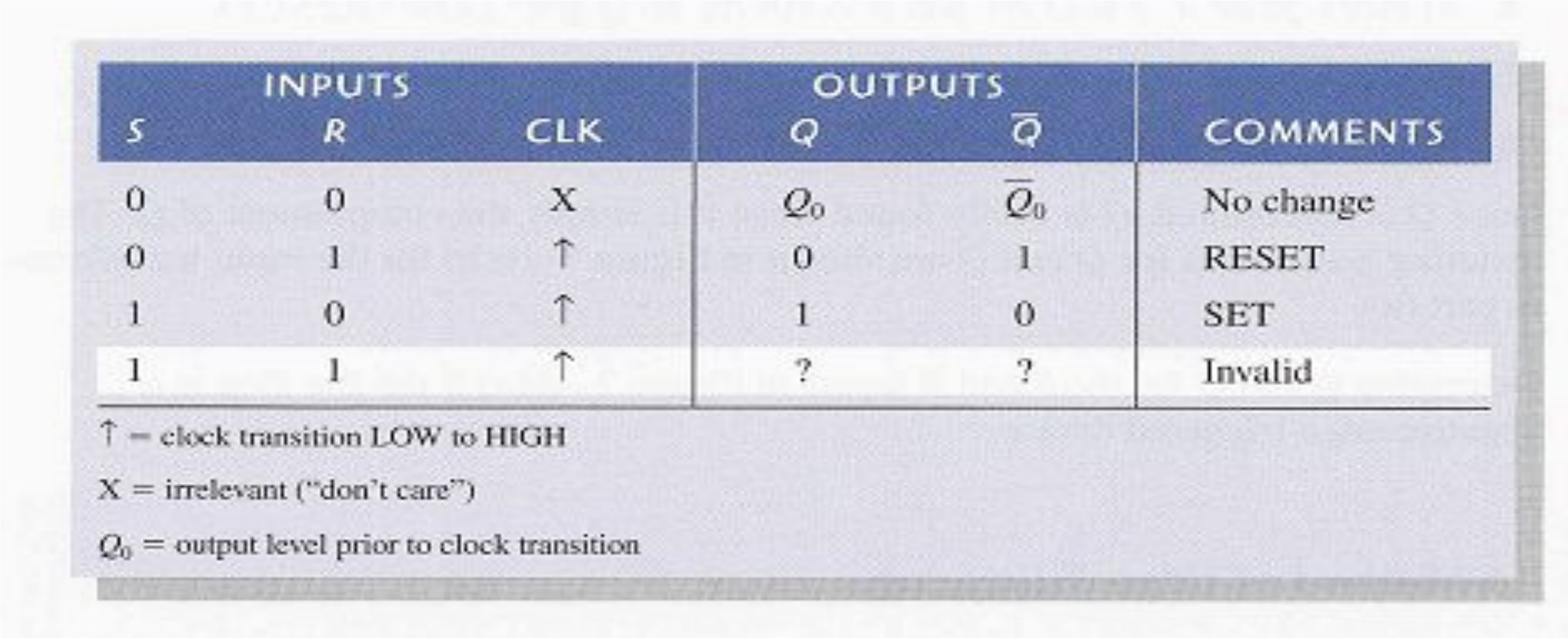
The $S$ and $R$ inputs of the **S-R flip-flop** are called synchronous inputs because data on these inputs are transferred to the flip-flop's output only on the triggering edge of the clock pulse. When $S$ is HIGH and $R$ is LOW, the $Q$ output goes HIGH on the triggering edge of the clock pulse, and the flip-flop is SET. When $S$ is LOW and $R$ is HIGH, the $Q$ output goes LOW on the triggering edge of the clock pulse, and the flip-flop is RESET. When both $S$ and $R$ are LOW, the output does not change from its prior state. An invalid condition exists when both $S$ and $R$ are HIGH.



(a) $S = 1$, $R = 0$ flip-flop SETS on positive clock edge. (If already SET, it remains SET.)

(b) $S = 0$, $R = 1$ flip-flop RESETS on positive clock edge. (If already RESET, it remains RESET.)

# Summary



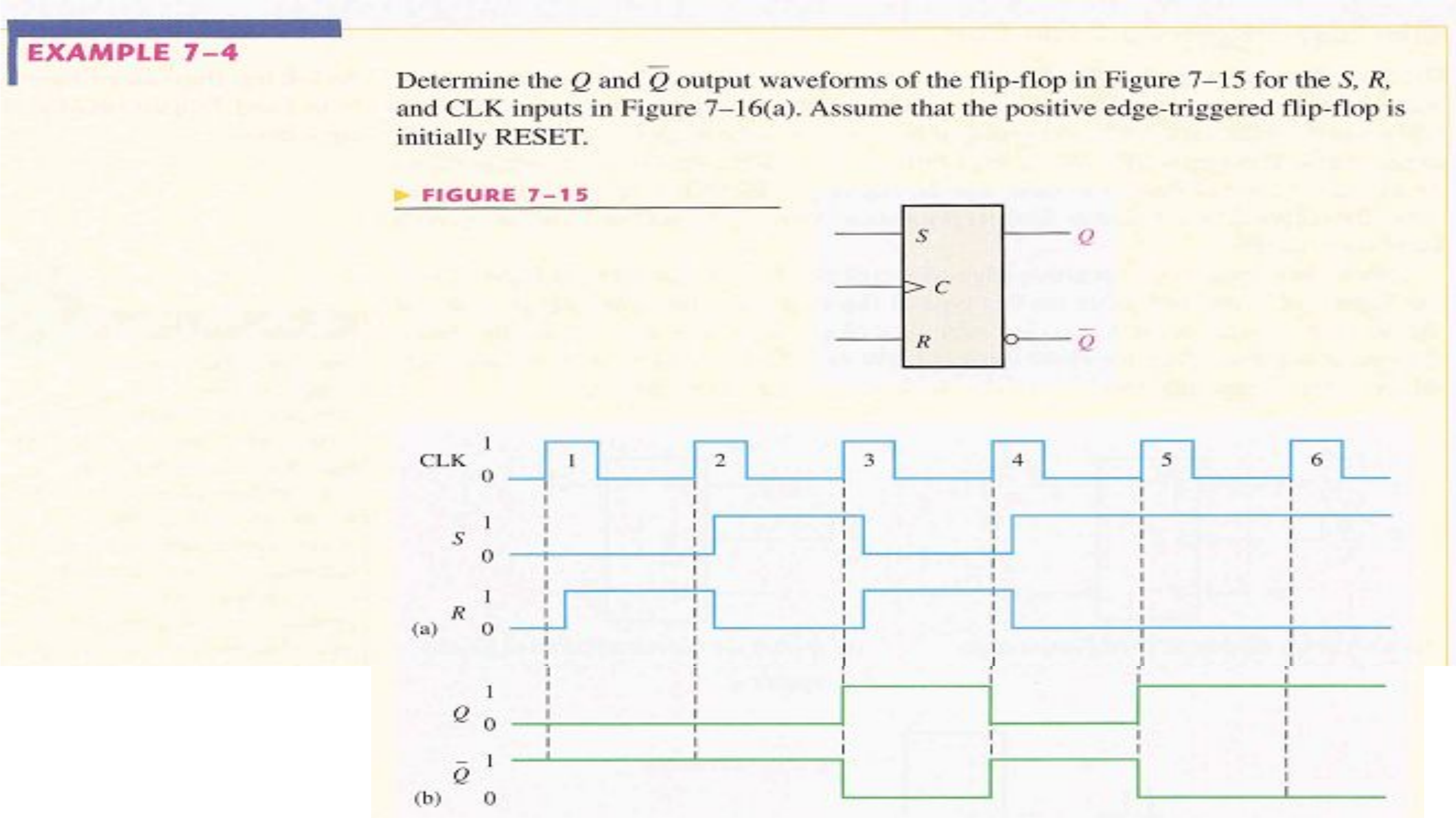
| INPUTS | | | OUTPUTS | | COMMENTS |
|---|---|---|---|---|---|
| S | R | CLK | Q | $\overline{Q}$ | |
| 0 | 0 | X | $Q_0$ | $\overline{Q}_0$ | No change |
| 0 | 1 | ↑ | 0 | 1 | RESET |
| 1 | 0 | ↑ | 1 | 0 | SET |
| 1 | 1 | ↑ | ? | ? | Invalid |

↑ = clock transition LOW to HIGH

X = irrelevant ("don't care")

$Q_0$ = output level prior to clock transition

# Summary



**EXAMPLE 7–4**

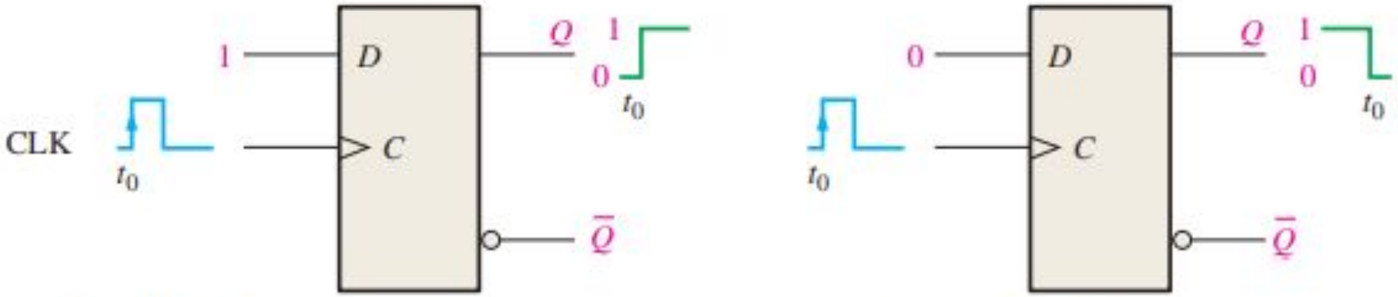
Determine the $Q$ and $\overline{Q}$ output waveforms of the flip-flop in Figure 7–15 for the $S$, $R$, and CLK inputs in Figure 7–16(a). Assume that the positive edge-triggered flip-flop is initially RESET.

▶ FIGURE 7–15

# Summary

## D-Flip-flops

The D input of the D flip-flop is a synchronous input because data on the input are transferred to the flip-flop's output only on the triggering edge of the clock pulse. When D is HIGH, the Q output goes HIGH on the triggering edge of the clock pulse, and the flip-flop



(a) $D = 1$ flip-flop SETS on positive clock edge. (If already SET, it remains SET.)

(b) $D = 0$ flip-flop RESETS on positive clock edge. (If already RESET, it remains RESET.)

# Summary

## D-Flip-flops

The truth table for a positive-edge triggered D flip-flop shows an up arrow to remind you that it is sensitive to its *D* input only on the rising edge of the clock; otherwise it is latched. The truth table for a negative-edge triggered D flip-flop is identical except for the direction of the arrow.

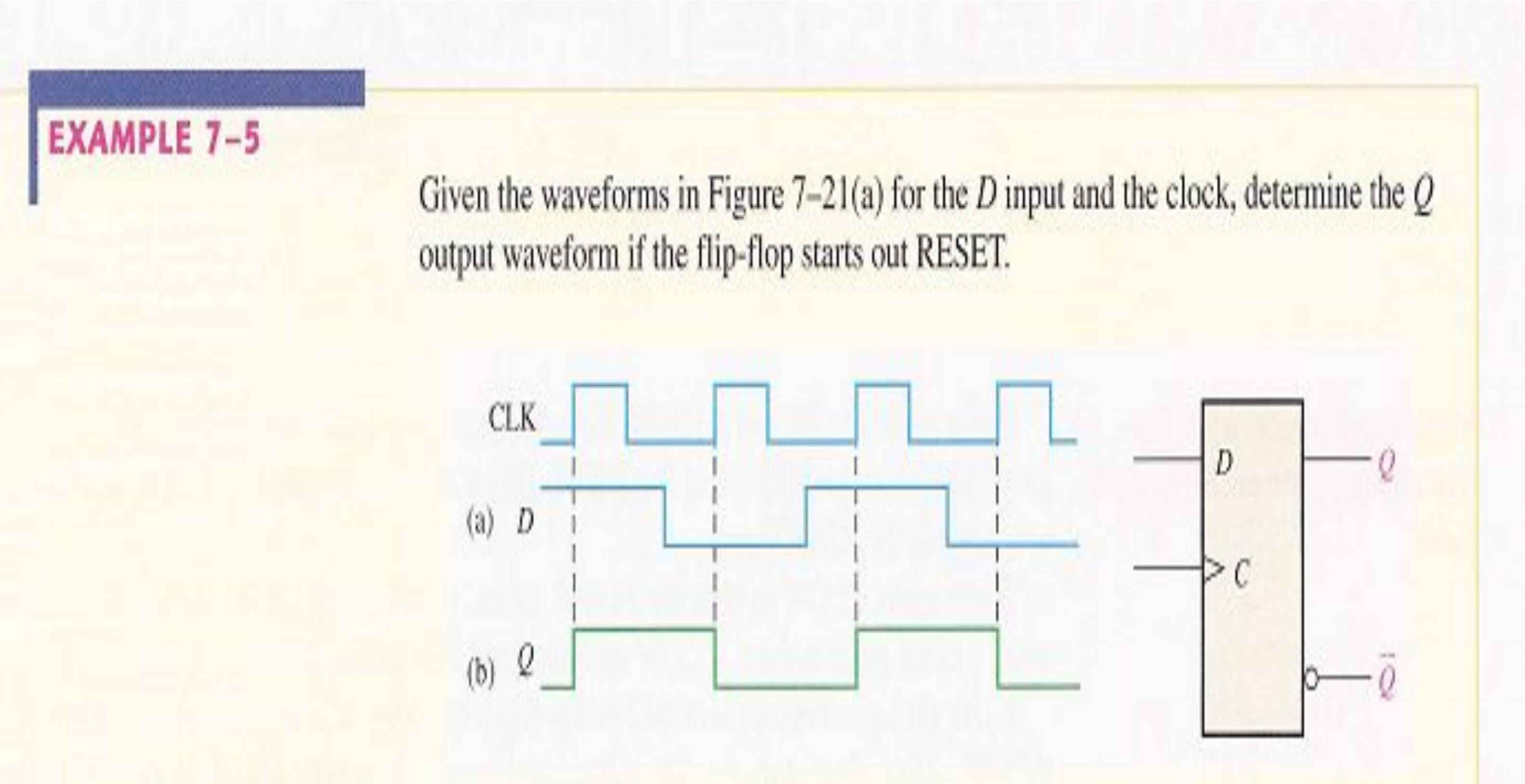
| Inputs | | Outputs | | |
|---|---|---|---|---|
| *D* | CLK | Q | $\overline{Q}$ | Comments |
| 1 | ↑ | 1 | 0 | SET |
| 0 | ↑ | 0 | 1 | RESET |

(a) Positive-edge triggered

| Inputs | | Outputs | | |
|---|---|---|---|---|
| *D* | CLK | Q | $\overline{Q}$ | Comments |
| 1 | ↓ | 1 | 0 | SET |
| 0 | ↓ | 0 | 1 | RESET |

(b) Negative-edge triggered

# Summary



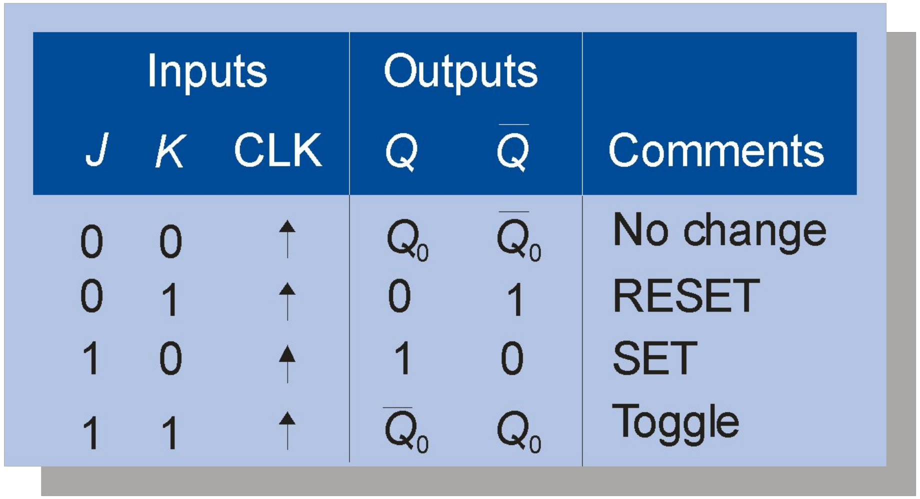
EXAMPLE 7–5

Given the waveforms in Figure 7–21(a) for the *D* input and the clock, determine the *Q* output waveform if the flip-flop starts out RESET.

# Summary

## J-K Flip-flops

The J-K flip-flop is more versatile than the D flip flop. In addition to the clock input, it has two inputs, labeled *J* and *K*. When both *J* and *K* = 1, the output changes states (toggles) on the active clock edge (in this case, the rising edge).



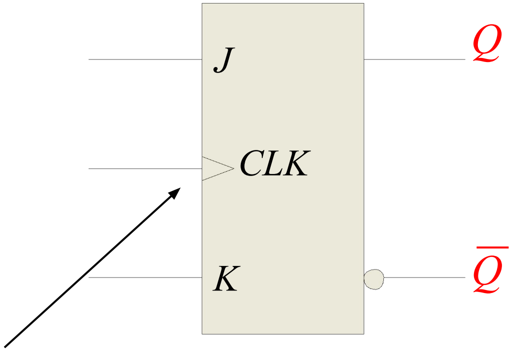
| Inputs | | | Outputs | | |
|---|---|---|---|---|---|
| *J* | *K* | CLK | *Q* | $\overline{Q}$ | Comments |
| 0 | 0 | ↑ | $Q_0$ | $\overline{Q}_0$ | No change |
| 0 | 1 | ↑ | 0 | 1 | RESET |
| 1 | 0 | ↑ | 1 | 0 | SET |
| 1 | 1 | ↑ | $\overline{Q}_0$ | $Q_0$ | Toggle |

# Summary

## Flip-flops

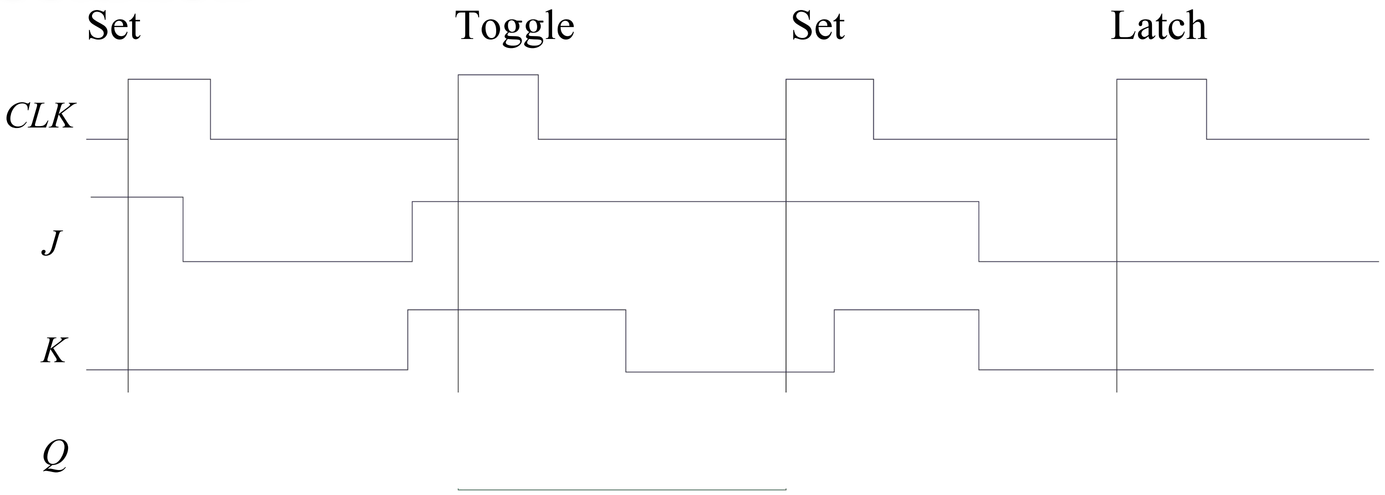
**Example**

Determine the $Q$ output for the $J$-$K$ flip-flop, given the inputs shown.



Notice that the outputs change on the leading edge of the clock.

**Solution**

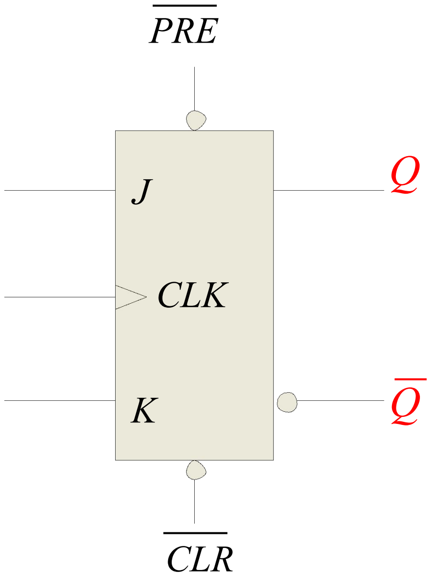


# Summary

## Flip-flops

Synchronous inputs are transferred in the triggering edge of the clock (for example the *D* or *J-K* inputs). Most flip-flops have other inputs that are *asynchronous*, meaning they affect the output independent of the clock.

Two such inputs are normally labeled preset (*PRE*) and clear (*CLR*). These inputs are usually active LOW. A J-K flip flop with active LOW preset and CLR is shown.

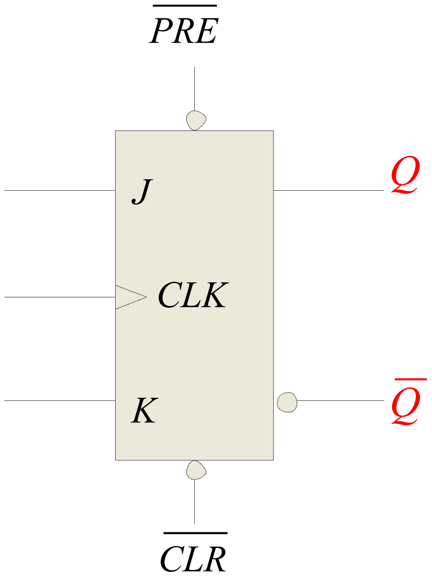


# Summary

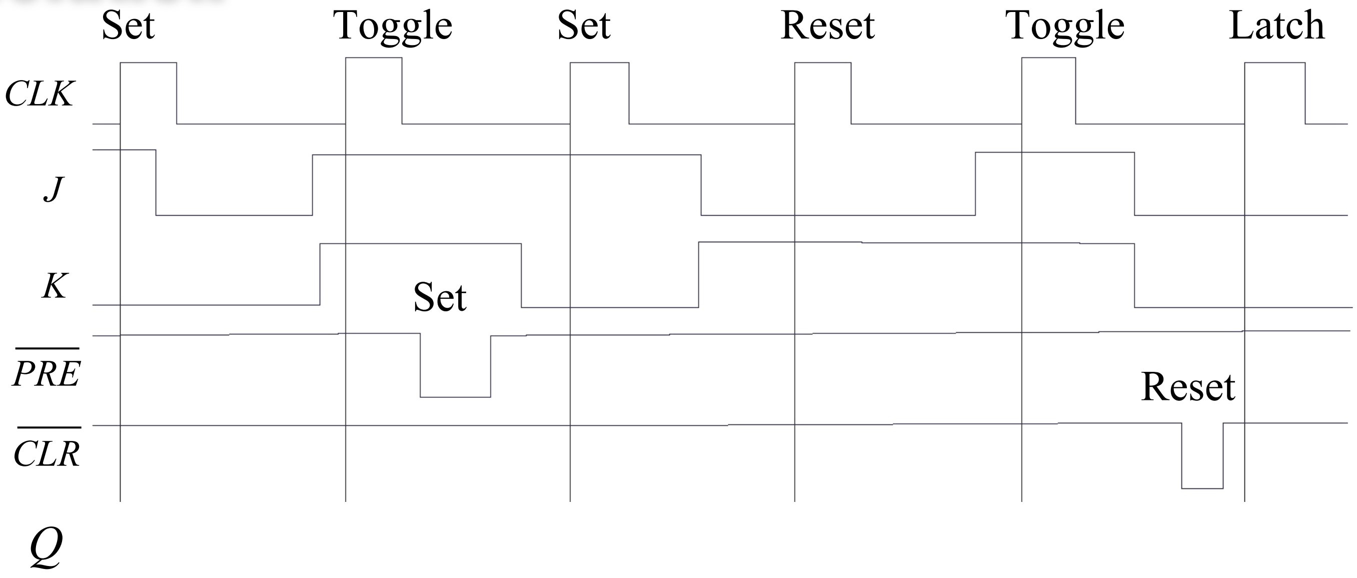
## Flip-flops

### Example

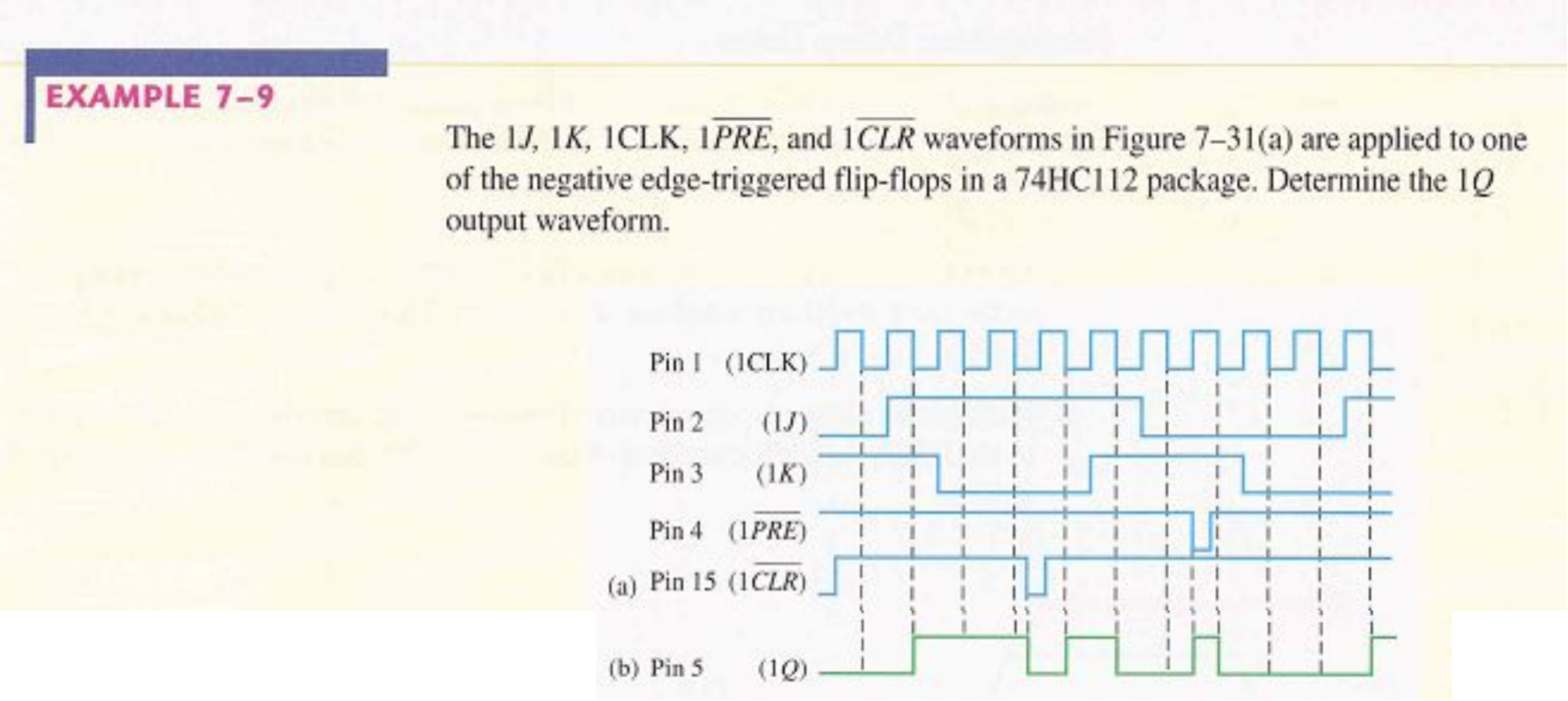
Determine the *Q* output for the *J-K* flip-flop, given the inputs shown.



### Solution



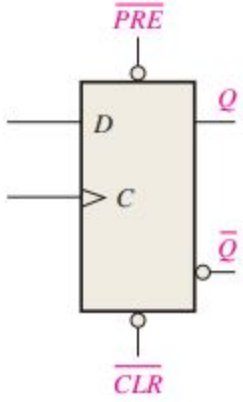
# Summary

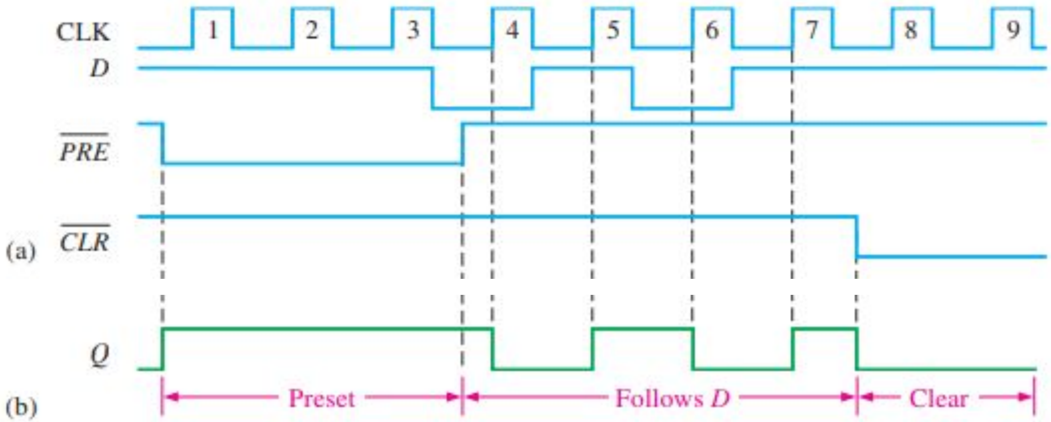


## EXAMPLE 7–9

The 1$J$, 1$K$, 1CLK, 1$\overline{PRE}$, and 1$\overline{CLR}$ waveforms in Figure 7–31(a) are applied to one of the negative edge-triggered flip-flops in a 74HC112 package. Determine the 1$Q$ output waveform.

For the positive edge-triggered D flip-flop with preset and clear inputs in Figure 7–27, determine the Q output for the inputs shown in the timing diagram in part (a) if Q is initially LOW.

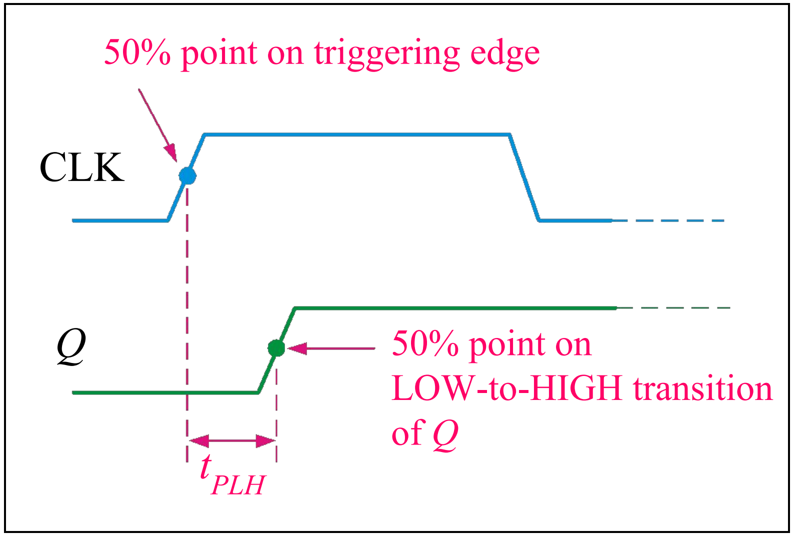




# Summary

## Flip-flop Characteristics

**Propagation delay time** is specified for the rising and falling outputs. It is measured between the 50% level of the clock to the 50% level of the output transition.

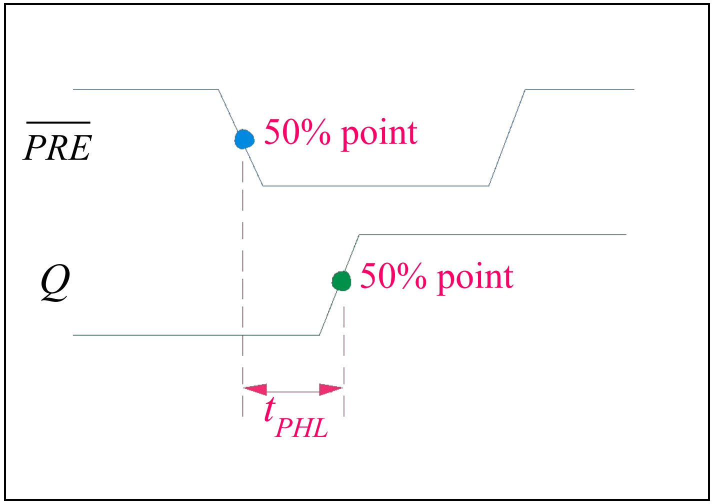

50% point on triggering edge
CLK
Q
50% point on LOW-to-HIGH transition of Q
$t_{PLH}$

The typical propagation delay time for the 74AHC family (CMOS) is 4 ns. Even faster logic is available for specialized applications.

# Summary

## Flip-flop Characteristics

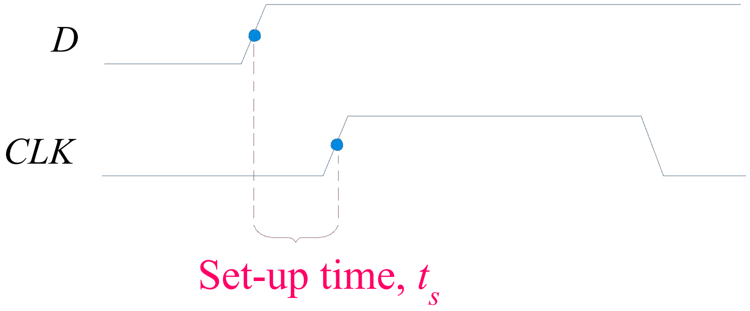
Another **propagation delay time** specification is the time required for an *asynchronous* input to cause a change in the output. Again it is measured from the 50% levels. The 74AHC family has specified delay times under 5 ns.



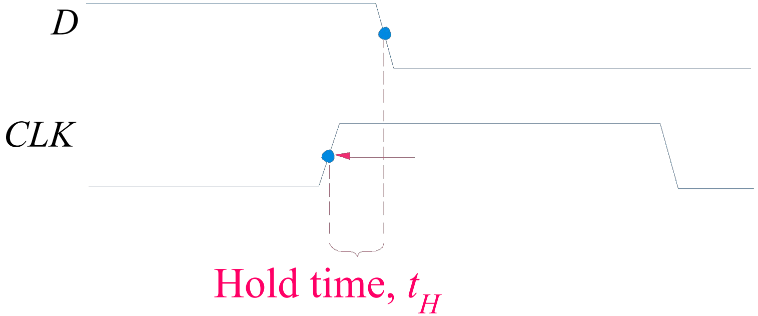
## Flip-flop Characteristics

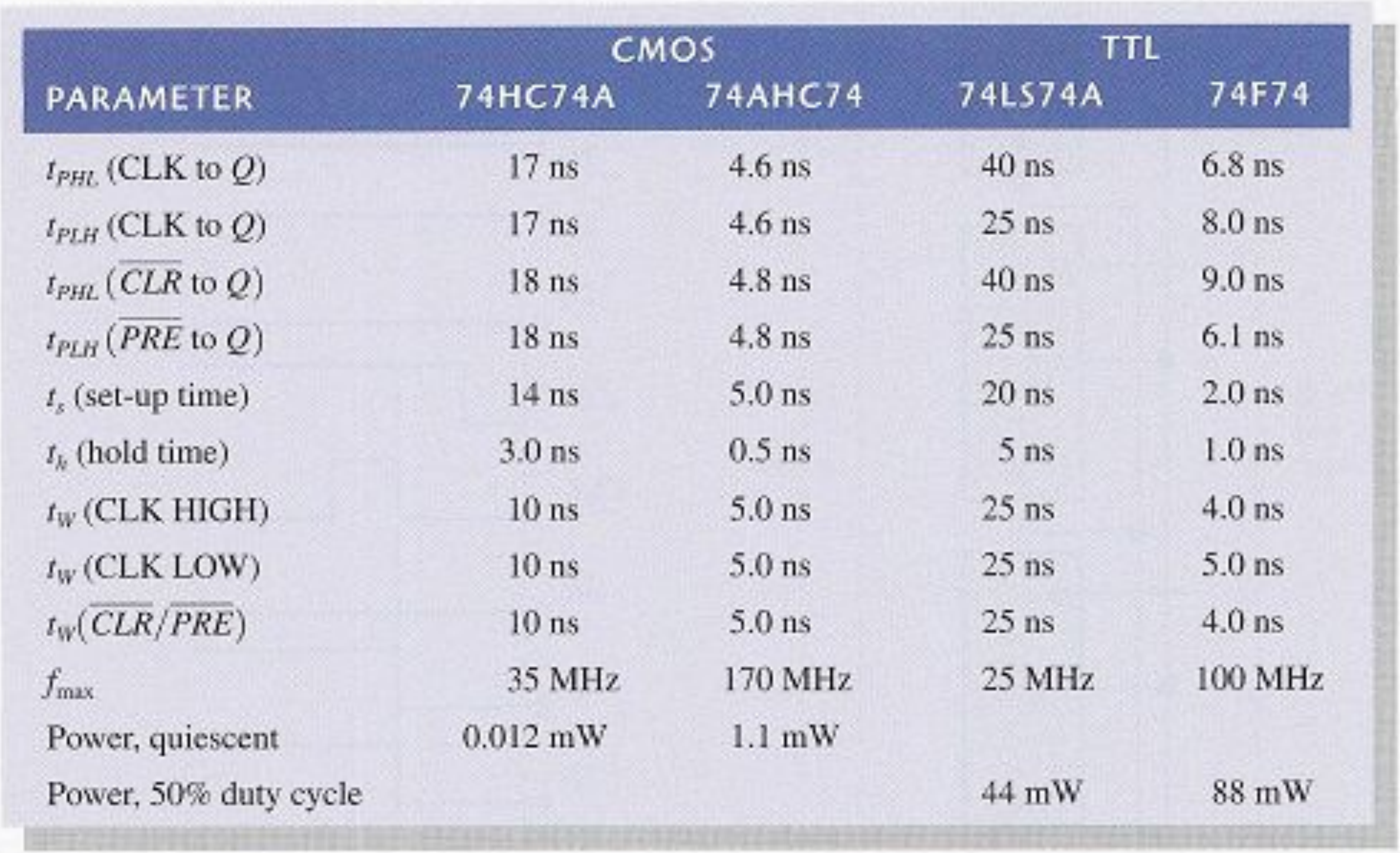
**Set-up time** and **hold time** are times required before and after the clock transition that data must be present to be reliably clocked into the flip-flop.

**Setup time** is the minimum time for the data to be present *before* the clock.



**Hold time** is the minimum time for the data to *remain* after the clock.





| | CMOS | | TTL | |
|---|---|---|---|---|
| PARAMETER | 74HC74A | 74AHC74 | 74LS74A | 74F74 |
| $t_{PHL}$ (CLK to $Q$) | 17 ns | 4.6 ns | 40 ns | 6.8 ns |
| $t_{PLH}$ (CLK to $Q$) | 17 ns | 4.6 ns | 25 ns | 8.0 ns |
| $t_{PHL}$ ($\overline{CLR}$ to $Q$) | 18 ns | 4.8 ns | 40 ns | 9.0 ns |
| $t_{PLH}$ ($\overline{PRE}$ to $Q$) | 18 ns | 4.8 ns | 25 ns | 6.1 ns |
| $t_s$ (set-up time) | 14 ns | 5.0 ns | 20 ns | 2.0 ns |
| $t_h$ (hold time) | 3.0 ns | 0.5 ns | 5 ns | 1.0 ns |
| $t_W$ (CLK HIGH) | 10 ns | 5.0 ns | 25 ns | 4.0 ns |
| $t_W$ (CLK LOW) | 10 ns | 5.0 ns | 25 ns | 5.0 ns |
| $t_W(\overline{CLR}/\overline{PRE})$ | 10 ns | 5.0 ns | 25 ns | 4.0 ns |
| $f_{\max}$ | 35 MHz | 170 MHz | 25 MHz | 100 MHz |
| Power, quiescent | 0.012 mW | 1.1 mW | | |
| Power, 50% duty cycle | | | 44 mW | 88 mW |

## Flip-flop Characteristics

Other specifications include maximum clock frequency, minimum pulse widths for various inputs, and power dissipation. The power dissipation is the product of the supply voltage and the average current required.

A useful comparison between logic families is the **speed-power product** which uses two of the specifications discussed: the average propagation delay and the average power dissipation. The unit is energy.

**Example** What is the speed-power product for 74AHC74A? Use the data from Table 7-5 to determine the answer.

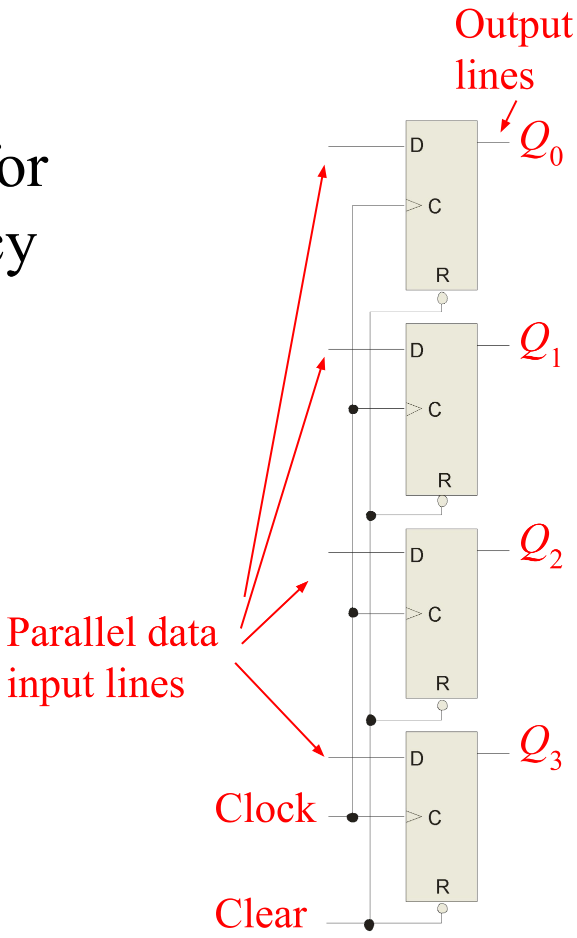
**Solution** From Table 7-5, the average propagation delay is 4.6 ns. The quiescent power dissipated is 1.1 mW. Therefore, the speed-power product is 5 pJ

# Summary

## Flip-flop Applications

Principal flip-flop applications are for temporary data storage, as  frequency dividers, and in counters (which are covered in detail in Chapter 8).

Typically, for **data storage** applications, a group of flip-flops are connected to parallel data lines and clocked together. Data is stored until the next clock pulse.



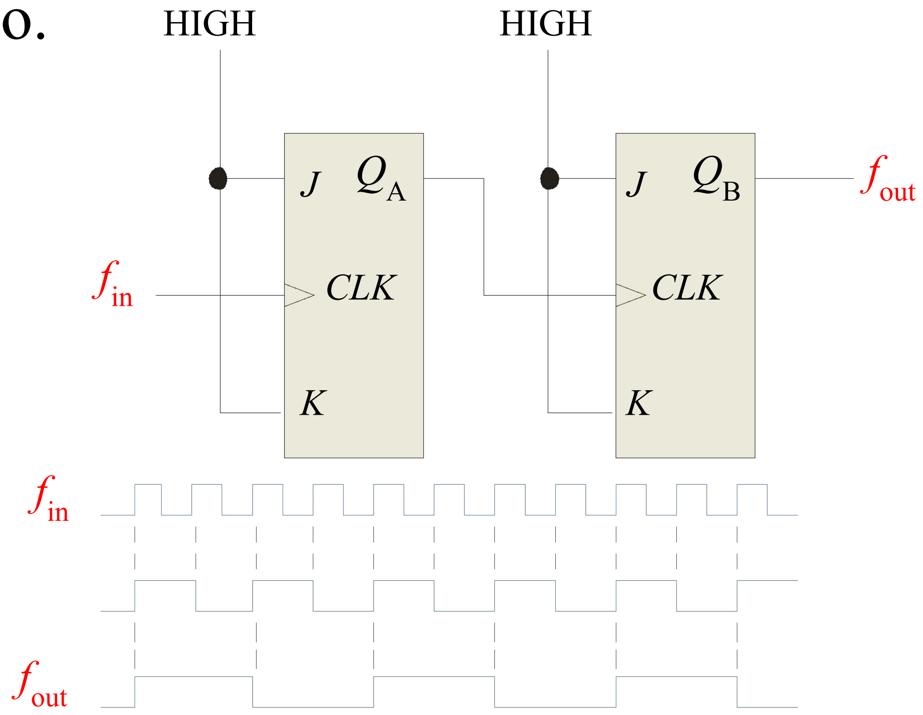
# Summary

## Flip-flop Applications

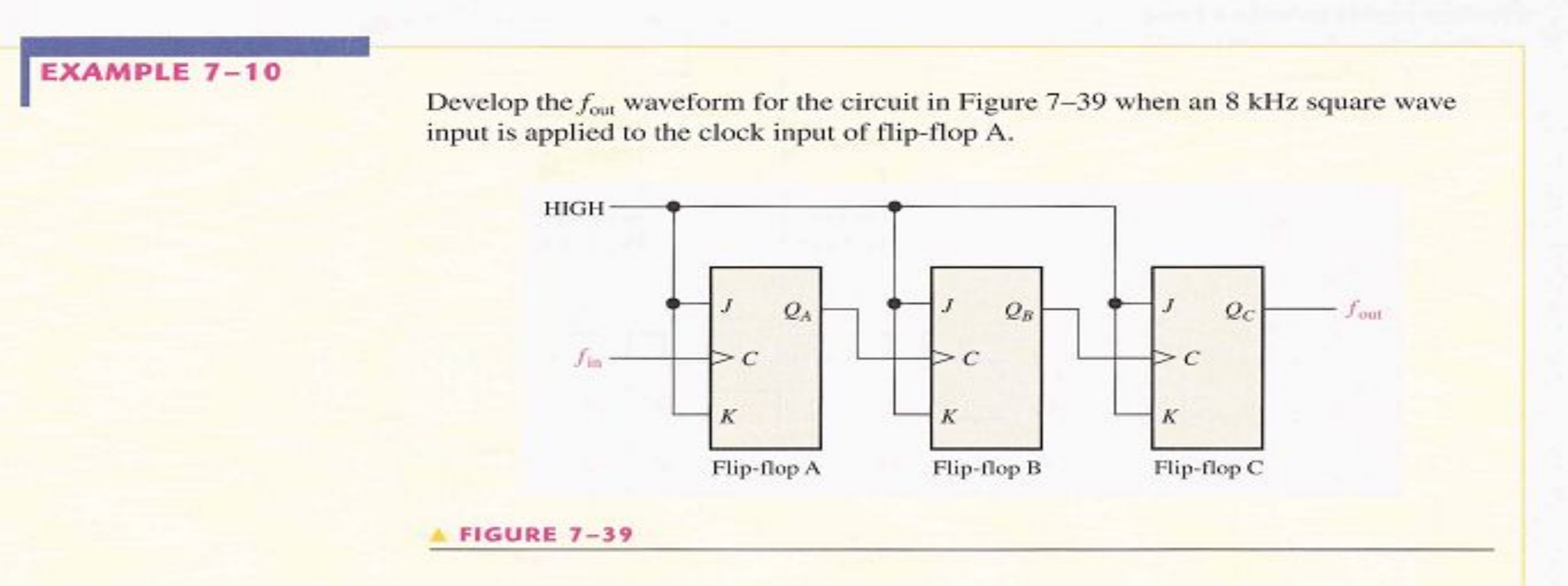
For **frequency division**, it is simple to use a flip-flop in the toggle mode or to chain a series of toggle flip flops to continue to divide by two.

One flip-flop will divide $f_{in}$ by 2, two flip-flops will divide $f_{in}$ by 4 (and so on). A side benefit of frequency division is that the output has an exact 50% duty cycle.

Waveforms:



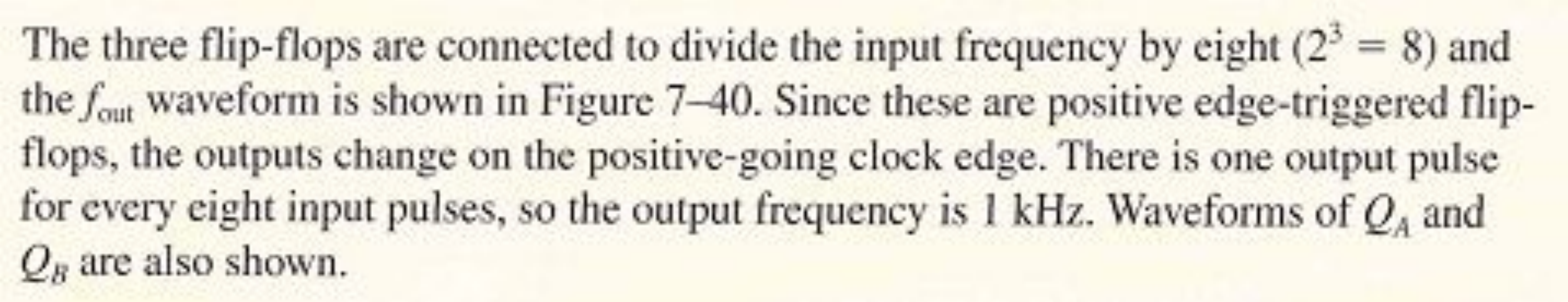
# Summary



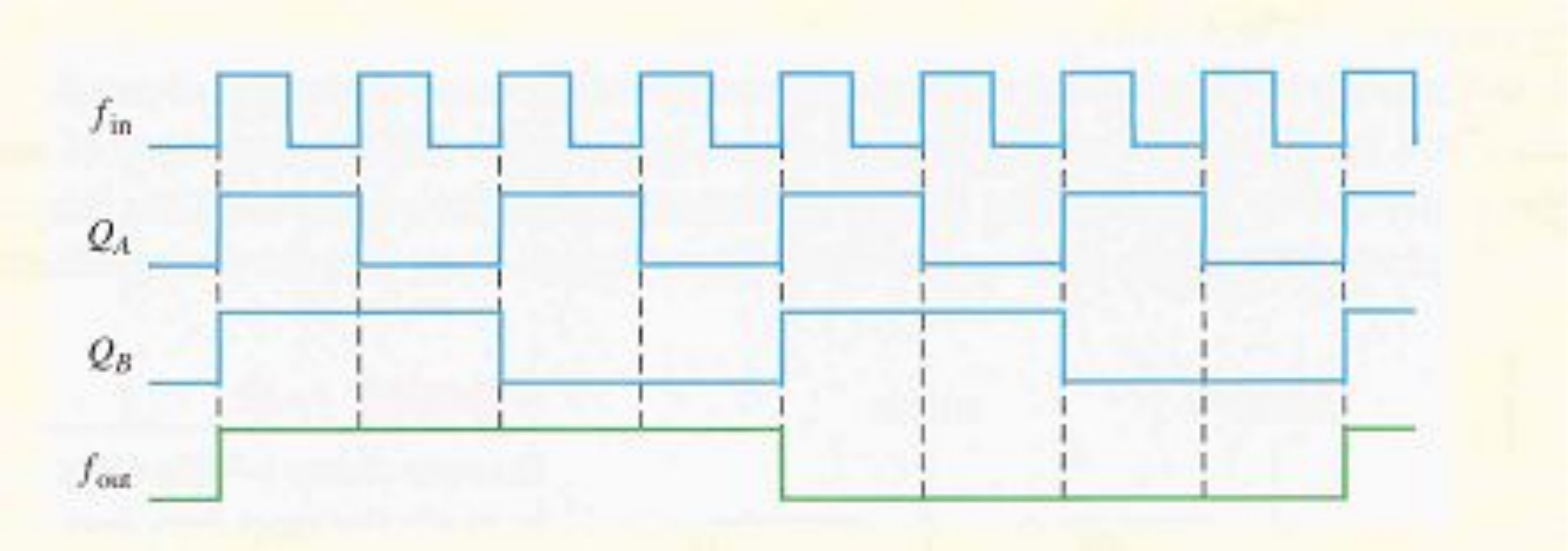
**EXAMPLE 7–10**

Develop the $f_{out}$ waveform for the circuit in Figure 7–39 when an 8 kHz square wave input is applied to the clock input of flip-flop A.

▲ FIGURE 7–39



The three flip-flops are connected to divide the input frequency by eight ($2^3 = 8$) and the $f_{out}$ waveform is shown in Figure 7–40. Since these are positive edge-triggered flip-flops, the outputs change on the positive-going clock edge. There is one output pulse for every eight input pulses, so the output frequency is 1 kHz. Waveforms of $Q_A$ and $Q_B$ are also shown.

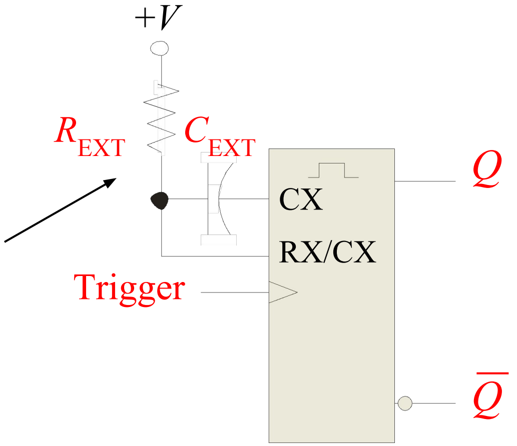

$f_{in}$
$Q_A$
$Q_B$
$f_{out}$

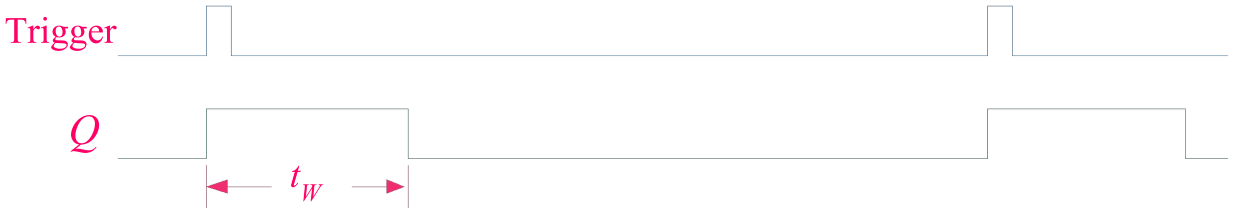
# Summary

## One-Shots

The **one-shot** or **monostable** multivibrator is a device with only one stable state. When triggered, it goes to its unstable state for a predetermined length of time, then returns to its stable state.

For most one-shots, the length of time in the unstable state ($t_W$) is determined by an external *RC* circuit.





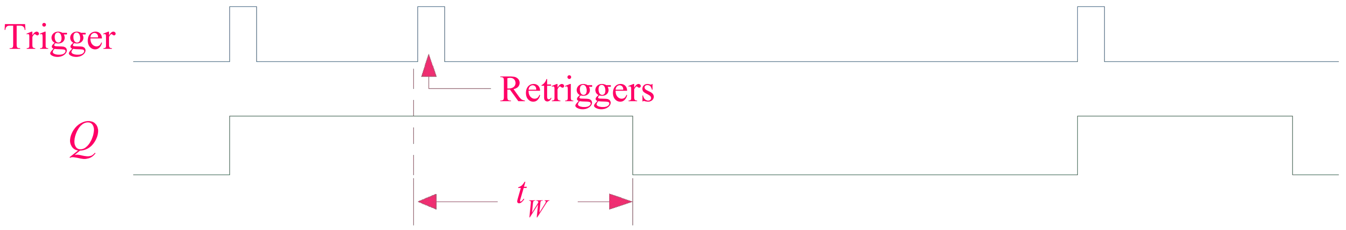
# Summary

## One-Shots

Nonretriggerable one-shots do not respond to any triggers that occur during the unstable state.

Retriggerable one-shots respond to any trigger, even if it occurs in the unstable state. If it occurs during the unstable state, the state is extended by an amount equal to the pulse width.

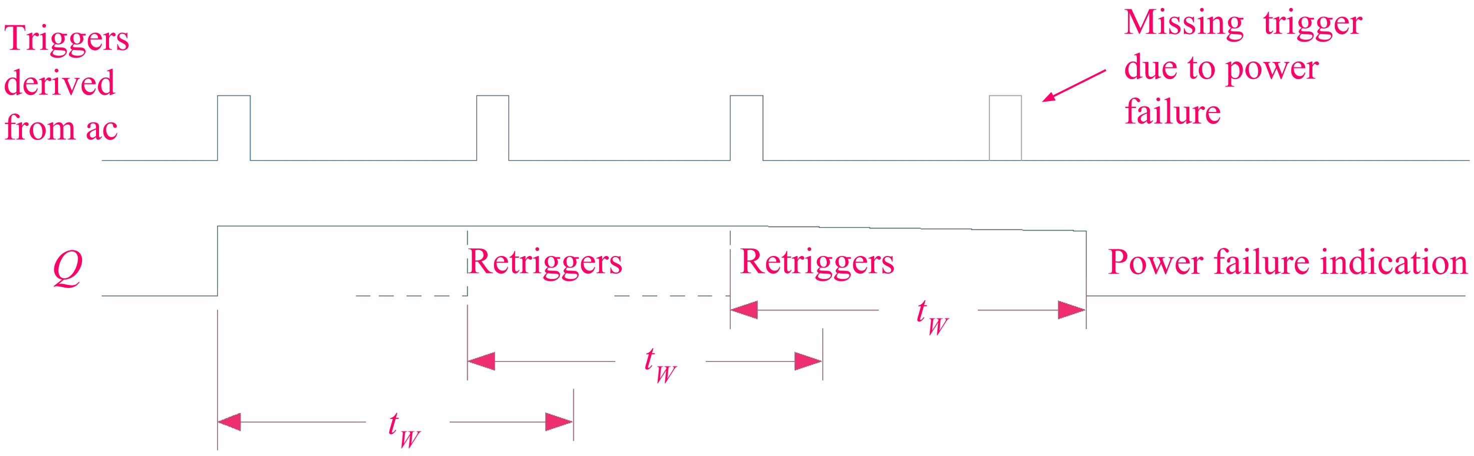
Retriggerable one-shot:



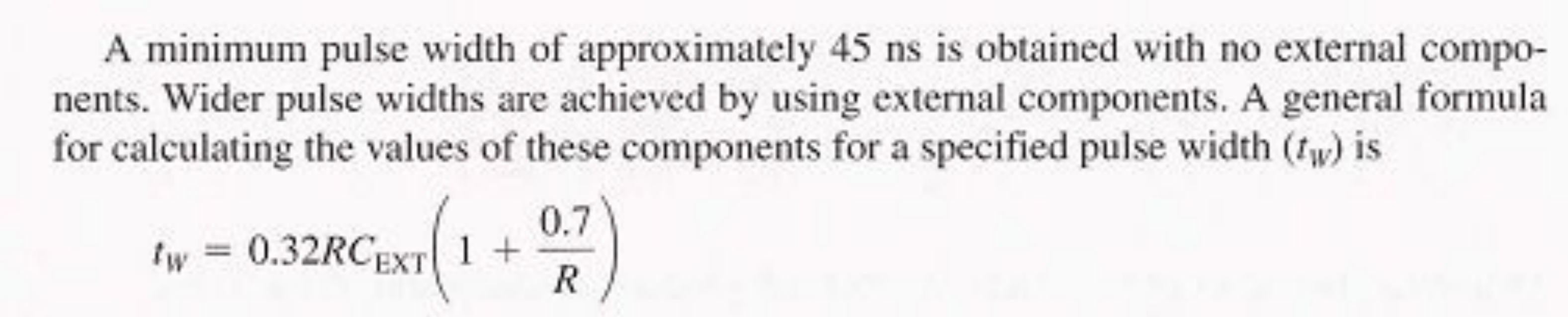
# Summary

## One-Shots

An application for a retriggerable one-shot is a power failure detection circuit. Triggers are derived from the ac power source, and continue to retrigger the one shot. In the event of a power failure, the one-shot is not triggered and an alarm can be initiated.

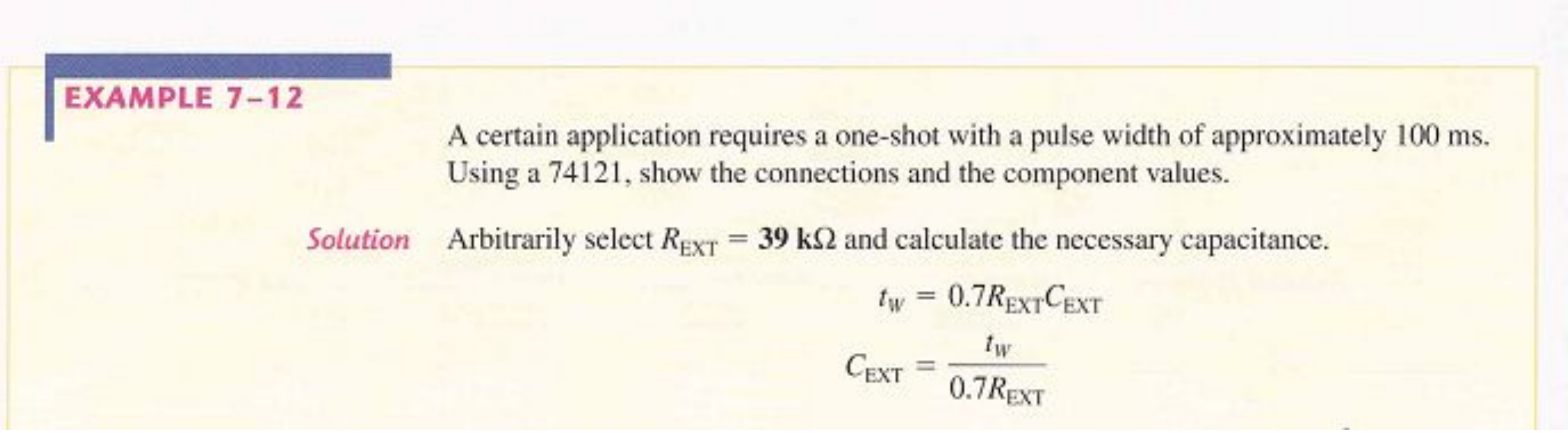


# Summary



A minimum pulse width of approximately 45 ns is obtained with no external components. Wider pulse widths are achieved by using external components. A general formula for calculating the values of these components for a specified pulse width ($t_W$) is

$$t_W = 0.32RC_{EXT}\left(1 + \frac{0.7}{R}\right)$$



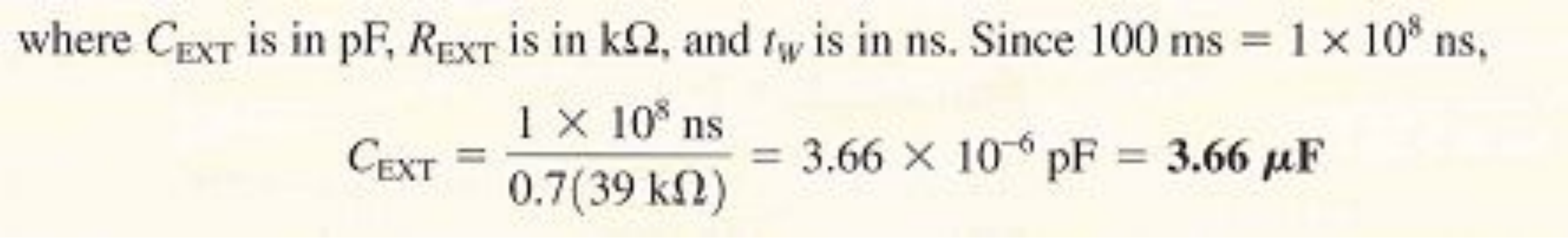
**EXAMPLE 7–12**

A certain application requires a one-shot with a pulse width of approximately 100 ms. Using a 74121, show the connections and the component values.

*Solution* Arbitrarily select $R_{EXT}$ = **39 kΩ** and calculate the necessary capacitance.

$$t_W = 0.7R_{EXT}C_{EXT}$$

$$C_{EXT} = \frac{t_W}{0.7R_{EXT}}$$



where $C_{EXT}$ is in pF, $R_{EXT}$ is in kΩ, and $t_W$ is in ns. Since 100 ms = $1 \times 10^8$ ns,

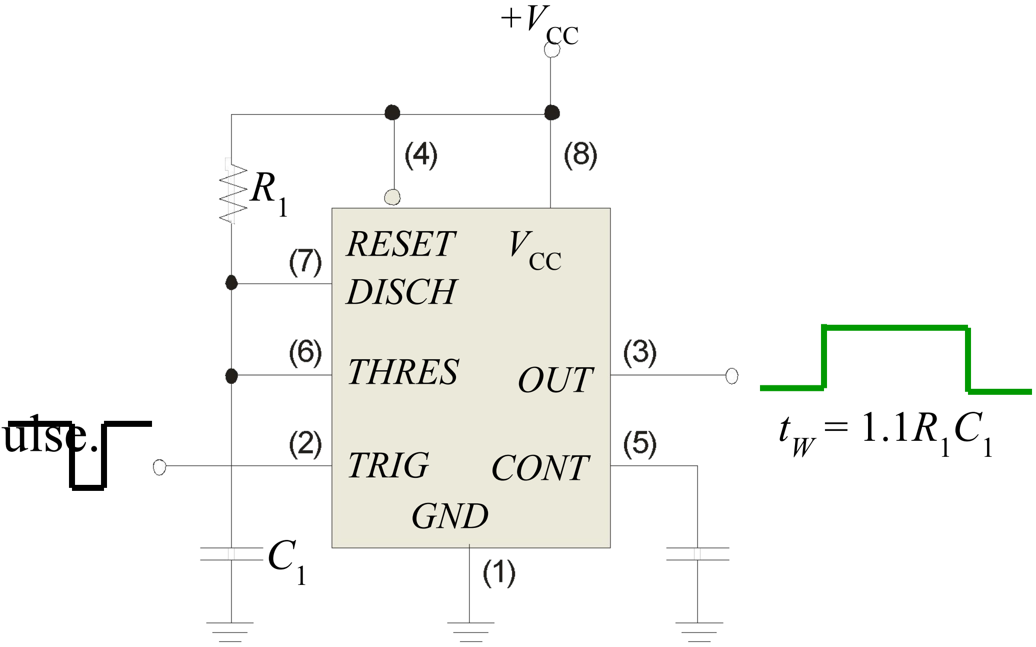
$$C_{EXT} = \frac{1 \times 10^8 \text{ ns}}{0.7(39 \text{ k}\Omega)} = 3.66 \times 10^{-6} \text{ pF} = \mathbf{3.66\ \mu F}$$

# Summary

## The 555 timer

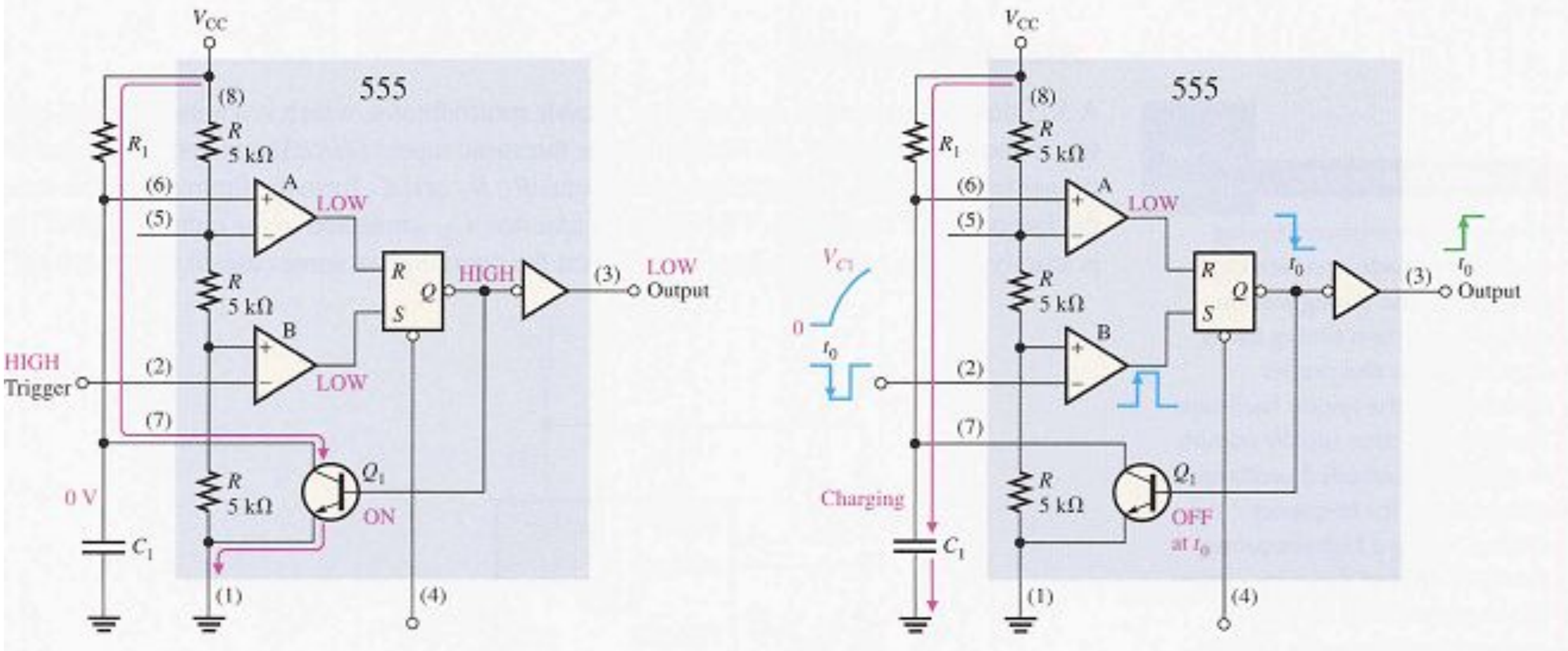
The 555 timer can be configured in various ways, including as a one-shot. A basic one shot is shown. The pulse width is determined by $R_1C_1$ and is approximately $t_W = 1.1R_1C_1$.

The trigger is a negative-going pulse.



# Summary

## The 555 timer Internal Diagram



(a) Prior to triggering. (The current path is indicated by the red arrow.)

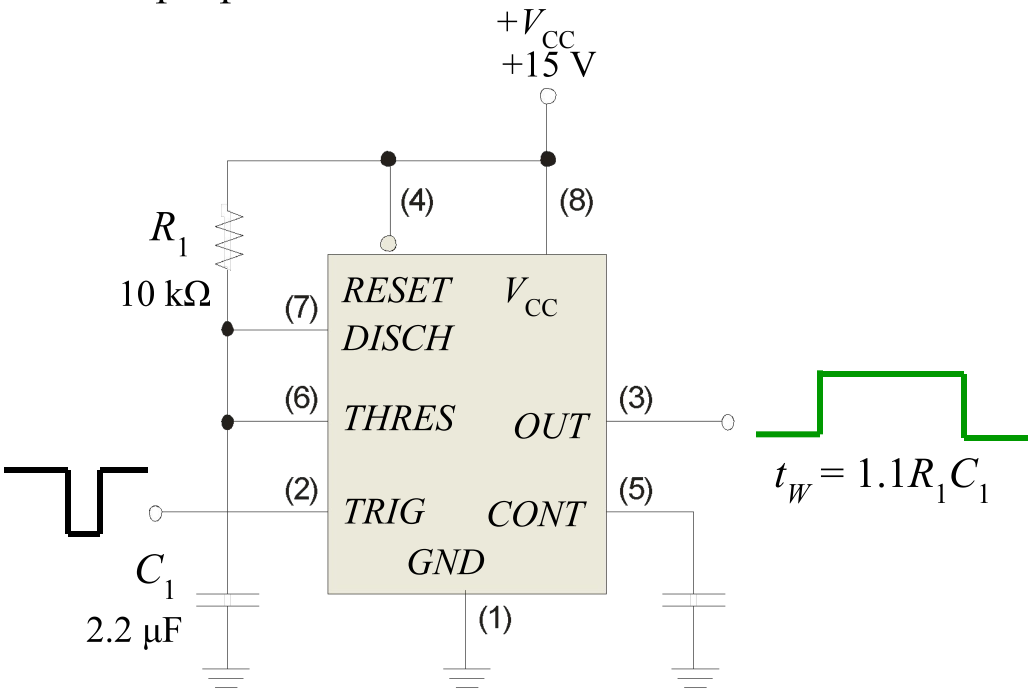
(b) When triggered

# Summary

## The 555 timer

**Example** Determine the pulse width for the circuit shown.

**Solution** $t_W = 1.1R_1C_1 = 1.1(10\ \text{k}\Omega)(2.2\ \mu\text{F}) =$ 24.2 ms



# Summary

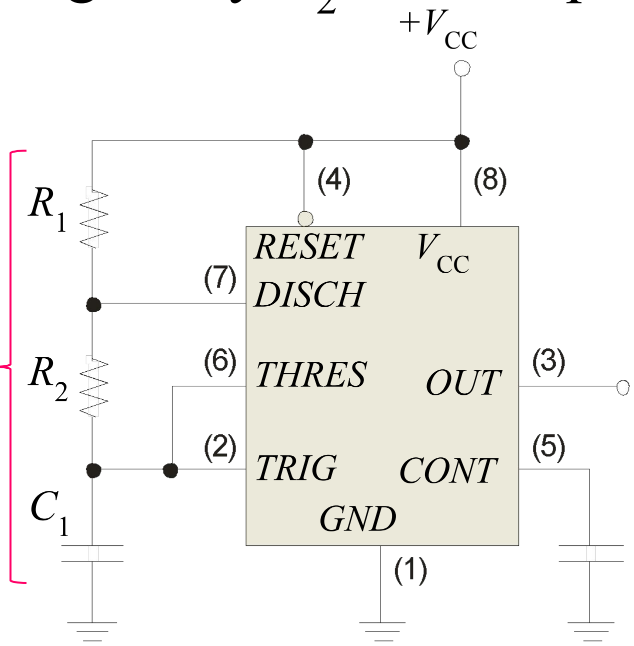
## The 555 timer

The 555 can be configured as a basic astable multivibrator with the circuit shown. In this circuit $C_1$ charges through $R_1$ and $R_2$ and discharges through only $R_2$. The output frequency is given by:

$$f = \frac{1.44}{(R_1 + 2R_2)C_1}$$

The frequency and duty cycle are set by these components.

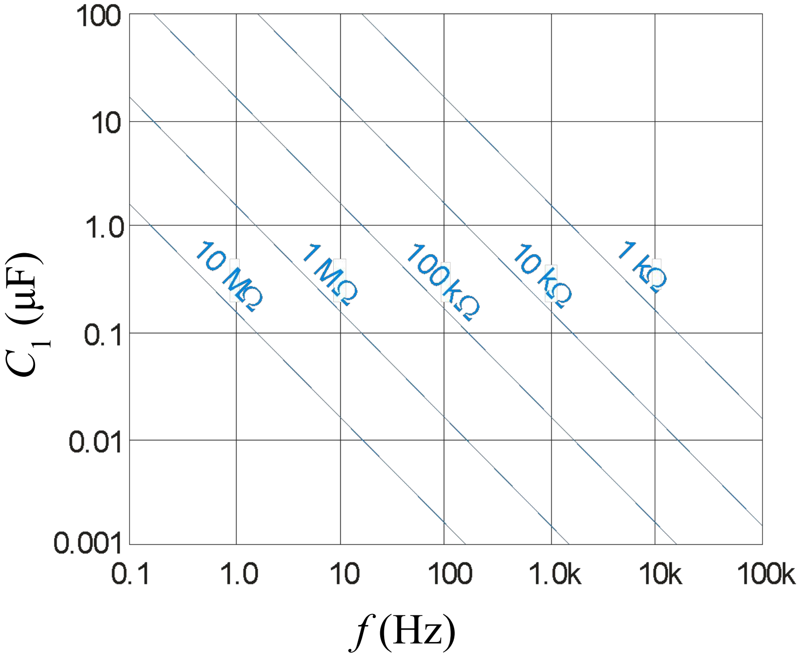
$$\text{Duty cycle} = \left(\frac{R_1 + R_2}{R_1 + 2R_2}\right)100\%$$

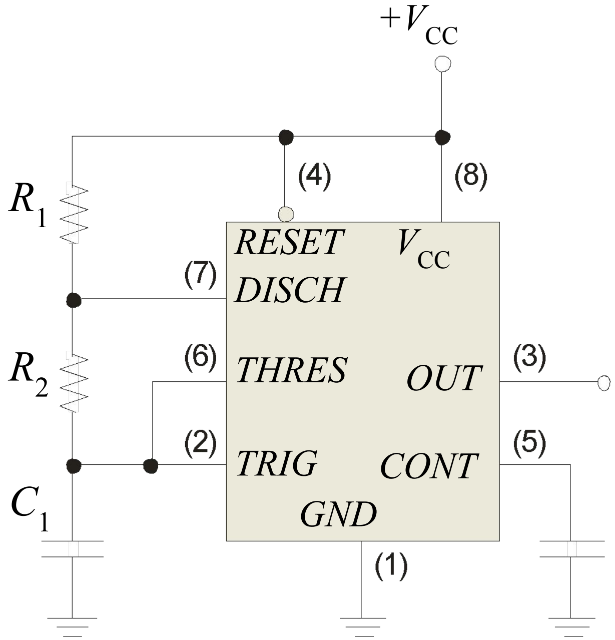


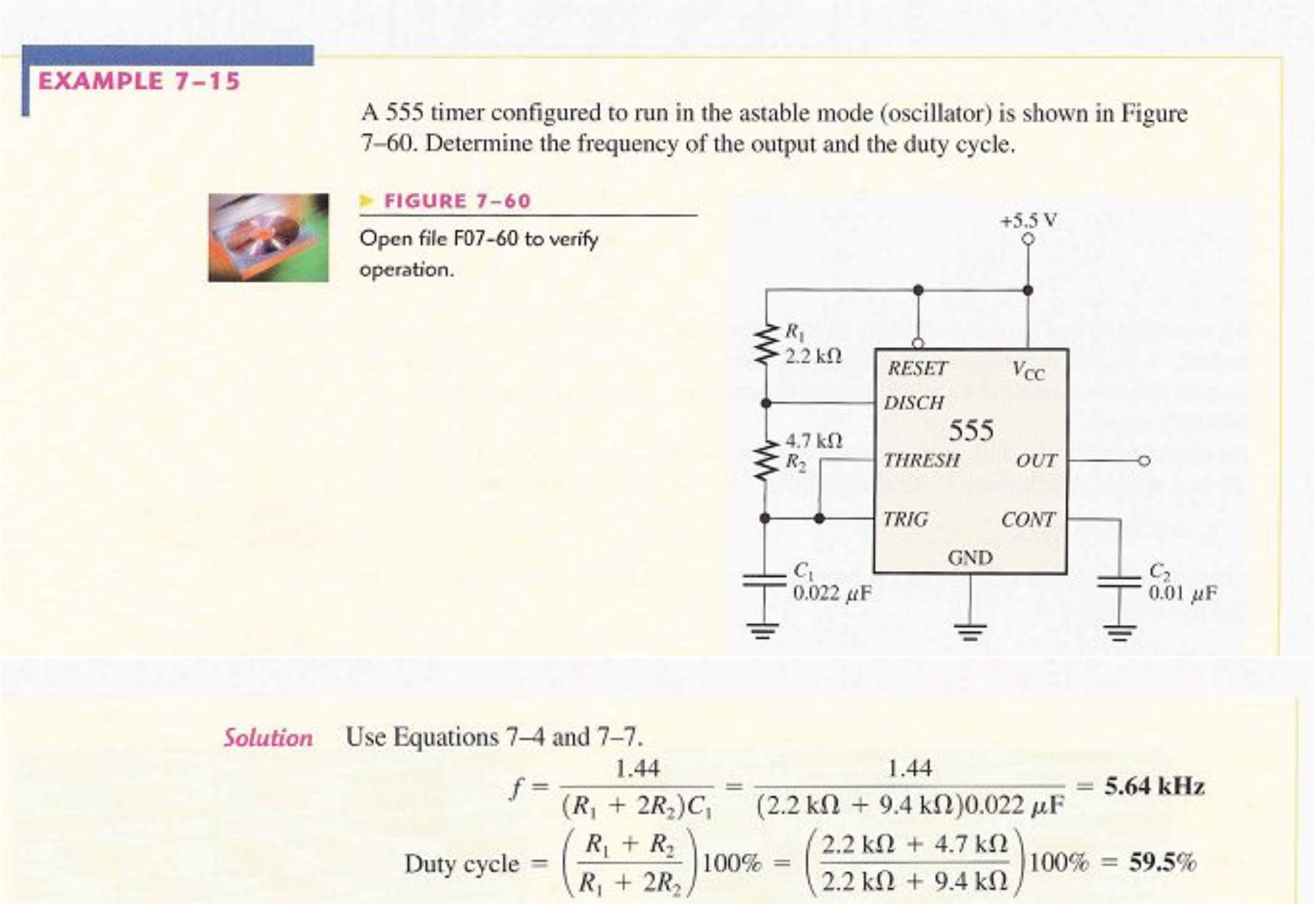
# Summary

## The 555 timer

Given the components, you can read the frequency from the chart. Alternatively, you can use the chart to pick components for a desired frequency.







EXAMPLE 7–15

A 555 timer configured to run in the astable mode (oscillator) is shown in Figure 7–60. Determine the frequency of the output and the duty cycle.

FIGURE 7–60

Open file F07-60 to verify operation.

+5.5 V
R1
2.2 kΩ
4.7 kΩ
R2
RESET
VCC
DISCH
555
THRESH
OUT
TRIG
CONT
GND
C1
0.022 μF
C2
0.01 μF

*Solution* Use Equations 7–4 and 7–7.

$$f = \frac{1.44}{(R_1 + 2R_2)C_1} = \frac{1.44}{(2.2\text{ k}\Omega + 9.4\text{ k}\Omega)0.022\ \mu\text{F}} = \mathbf{5.64\ kHz}$$

$$\text{Duty cycle} = \left(\frac{R_1 + R_2}{R_1 + 2R_2}\right)100\% = \left(\frac{2.2\text{ k}\Omega + 4.7\text{ k}\Omega}{2.2\text{ k}\Omega + 9.4\text{ k}\Omega}\right)100\% = \mathbf{59.5\%}$$

# Selected Key Terms

***Latch*** A bistable digital circuit used for storing a bit.

***Bistable*** Having two stable states. Latches and flip-flops are bistable multivibrators.

***Clock*** A triggering input of a flip-flop.

***D flip-flop*** A type of bistable multivibrator in which the output assumes the state of the *D* input on the triggering edge of a clock pulse.

***J-K flip-flop*** A type of flip-flop that can operate in the SET, RESET, no-change, and toggle modes.

# Selected Key Terms

***Propagation delay time*** The interval of time required after an input signal has been applied for the resulting output signal to change.

***Set-up time*** The time interval required for the input levels to be on a digital circuit.

***Hold time*** The time interval required for the input levels to remain steady to a flip-flop after the triggering edge in order to reliably activate the device.

***Timer*** A circuit that can be used as a one-shot or as an oscillator.