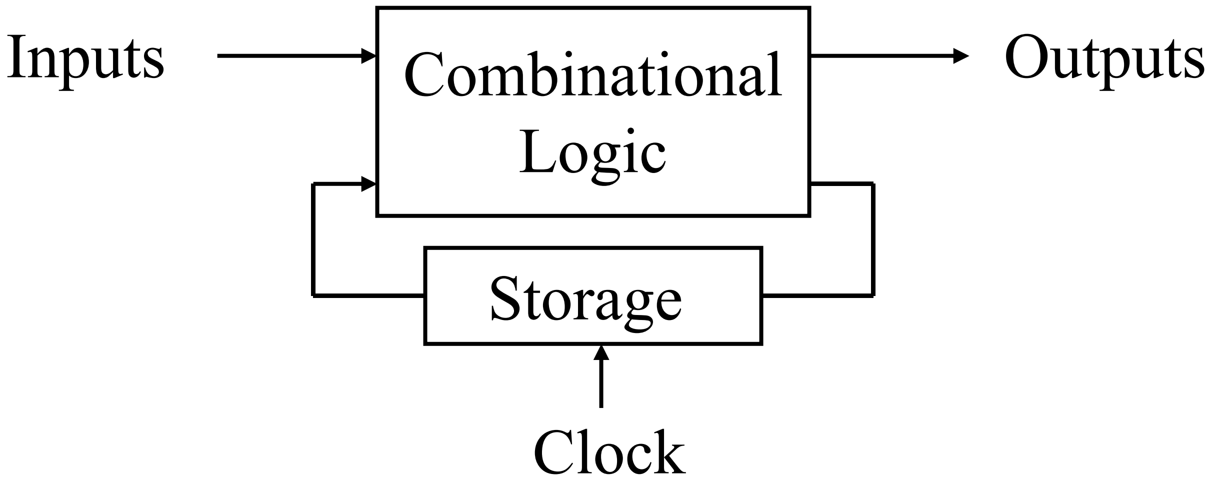
Latches and Flip-Flops

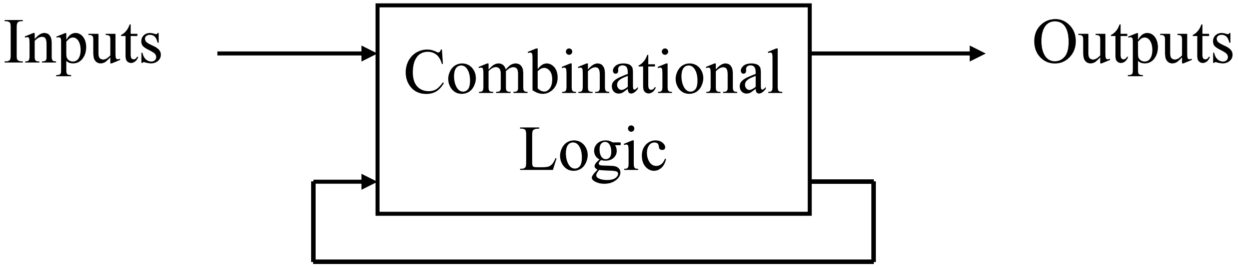
- Asynchronous Sequential Circuits
- Stable and Unstable States
- Latches
- Flip-Flops

Synchronous Sequential Circuits



Changes in the state and changes at the outputs, in response to changes at the inputs, happen at fixed points in time. These are controlled by the rising or falling edges of a free-running clock.

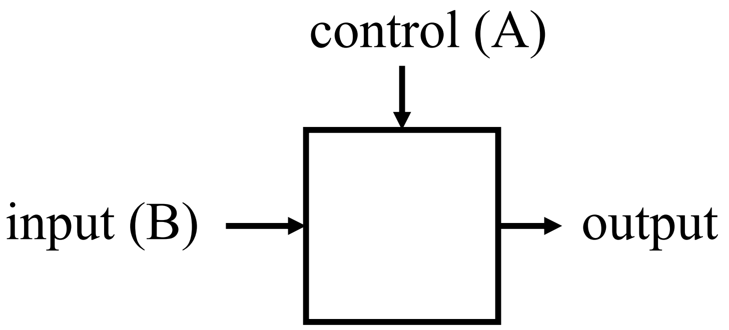
Asynchronous Sequential Circuits



The operation of an asynchronous circuit is not controlled by an external timing mechanism. It is basically a combinational circuit with feedback.

As soon as changes are made at the inputs, after a propagation delay, they take effect at the outputs. This is known as a free-running circuit.

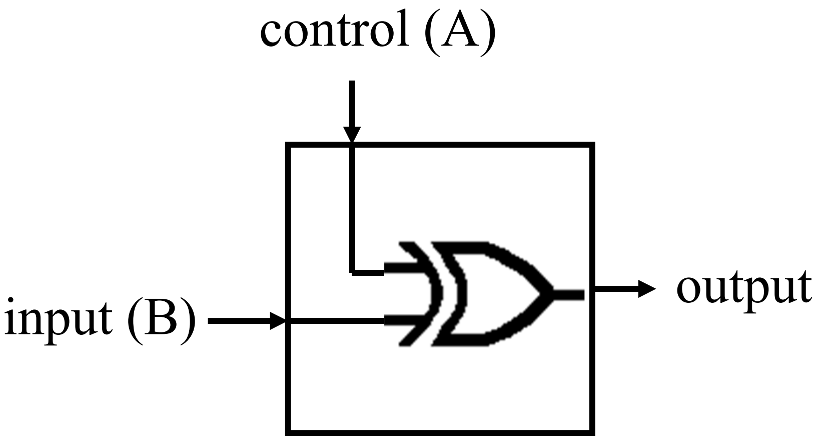
Controlled Inverter



when control (A) = **0**, output = input

when control (A) = **1**, output = $\overline{\text{input}}$

A	B	output
0	0	0
0	1	1
1	0	1
1	1	0



$$\mathbf{0 \oplus B = B}$$

$$\mathbf{1 \oplus B = \overline{B}}$$

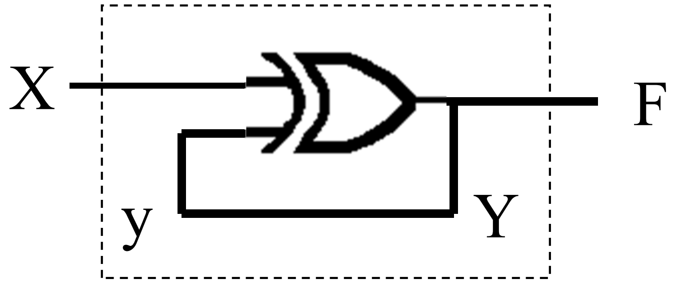
Stable and Unstable States

The circuit shown contains an XOR gate with its output fed back to one of its inputs. This forms an asynchronous sequential circuit.

For the circuit to be stable, the output generated by the inputs must result in the same input conditions.

i.e. $Y = y$

If this is not the case, the circuit will be unstable. It will oscillate as each new output is fed back to the input, changing the state of the output again. The speed of oscillation is determined by the propagation delay of the feedback path.



Stable when $X = 0, y = 0$

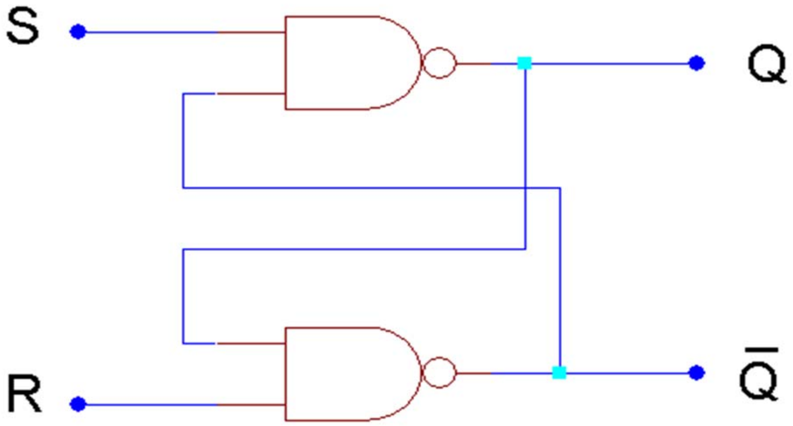
$X = 0, y = 1$

Unstable when $X = 1, y = 0$

$X = 1, y = 1$

Active low SR Latch

A latch is a bistable storage device which uses feedback. It can be formed using two NAND gates in a cross-coupled arrangement as shown.

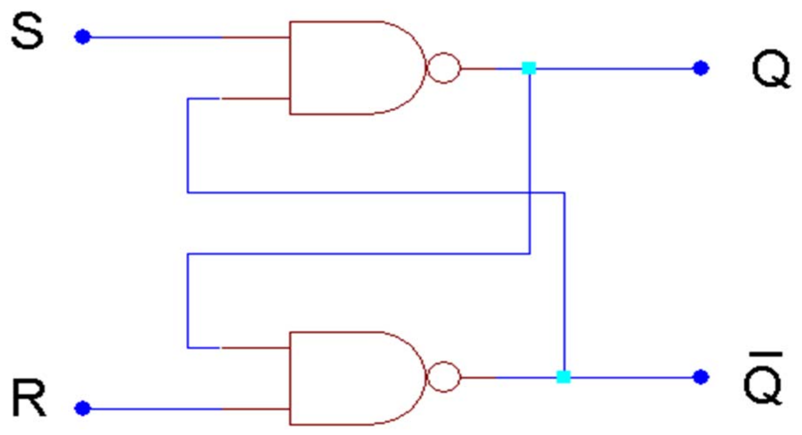


The inputs are labelled **S** for set and **R** for reset. For normal operation, **Q** and $\overline{\textbf{Q}}$ are defined to be the complement of each other.

Activating **S** will set **Q** to **1**

Activating **R** will reset **Q** to **0**

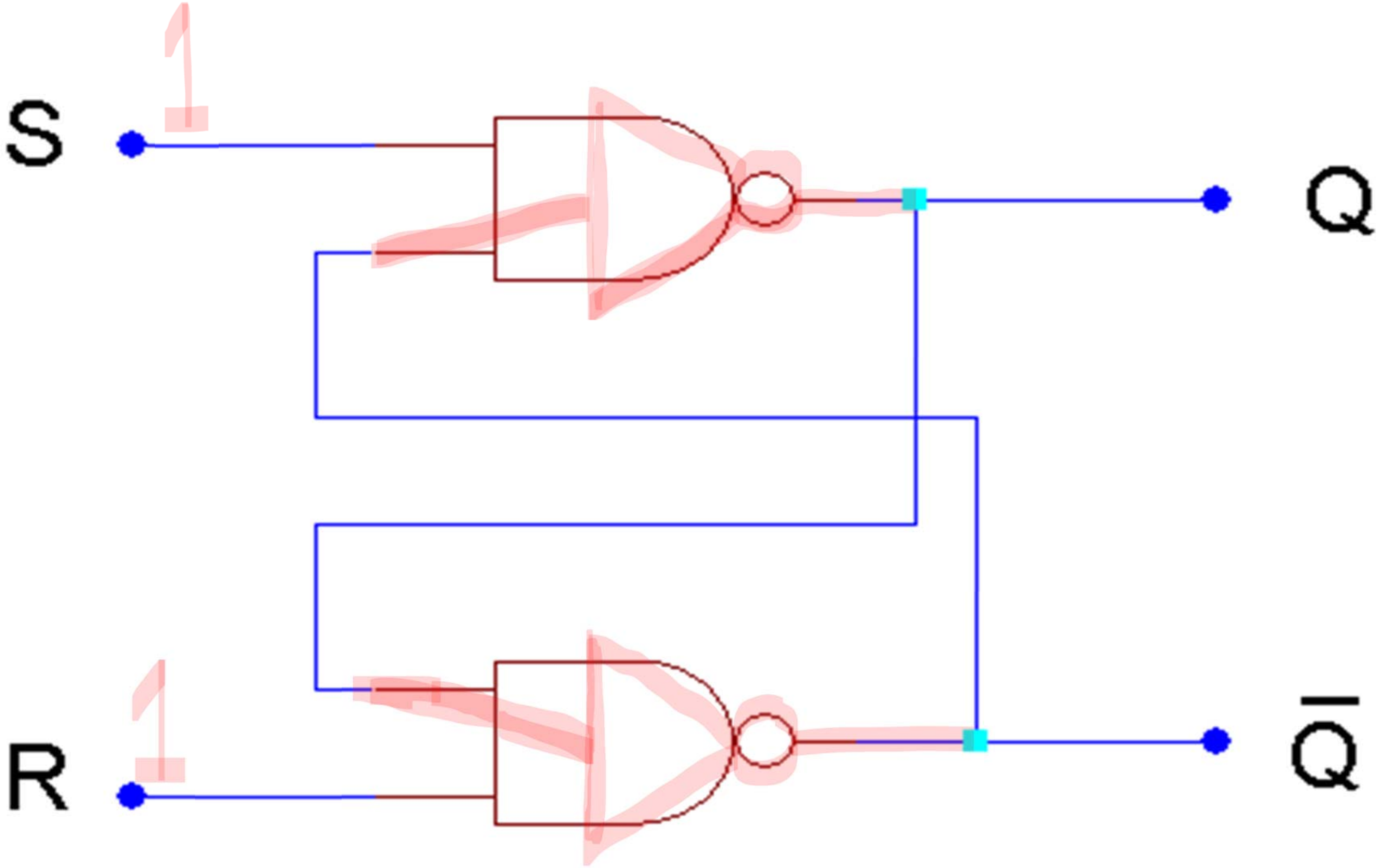
Active low SR Latch



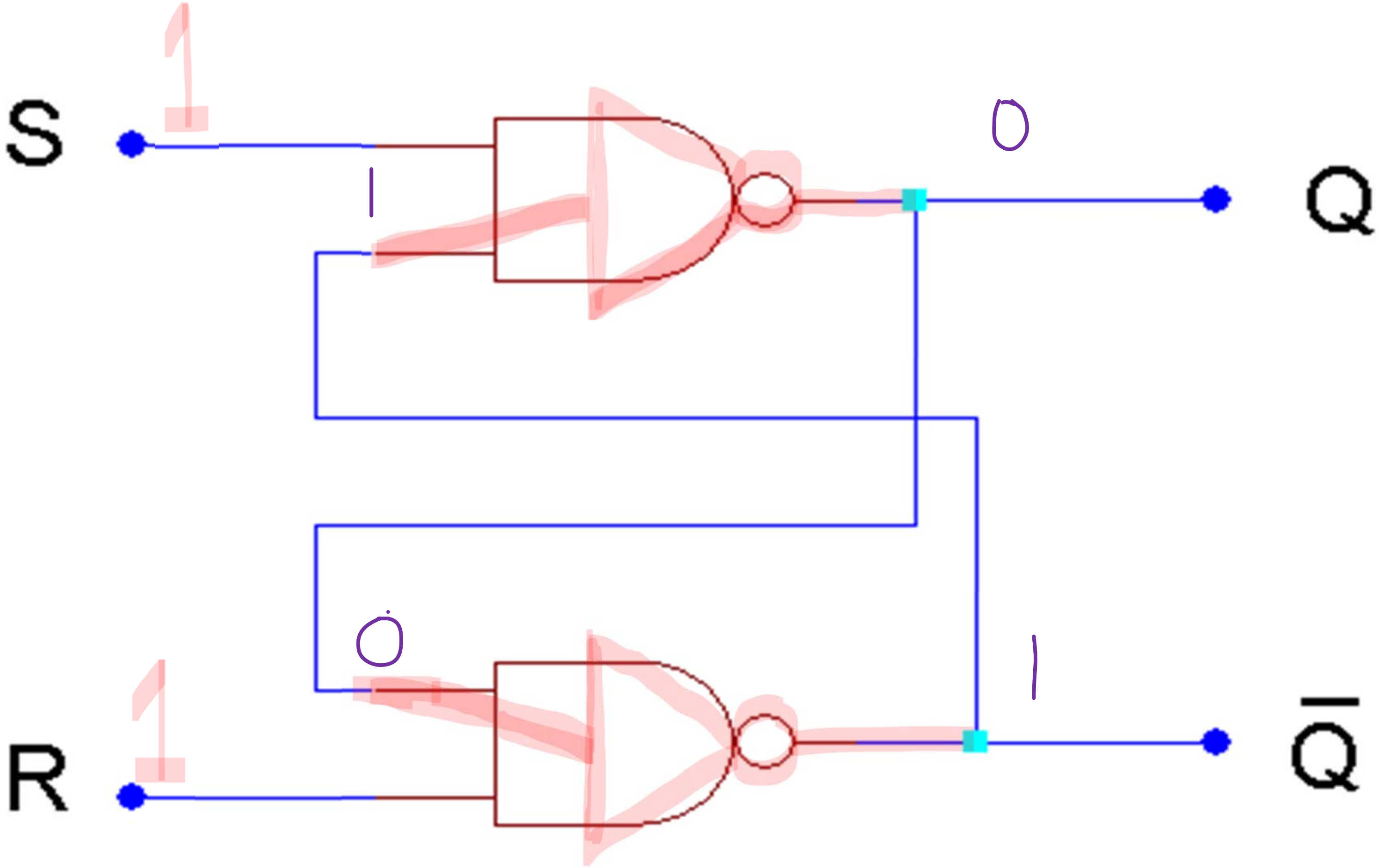
X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

Remember that a **0** input drives a NAND gate output to **1** and that a **1** input to a NAND gate will cause its other input to be inverted.

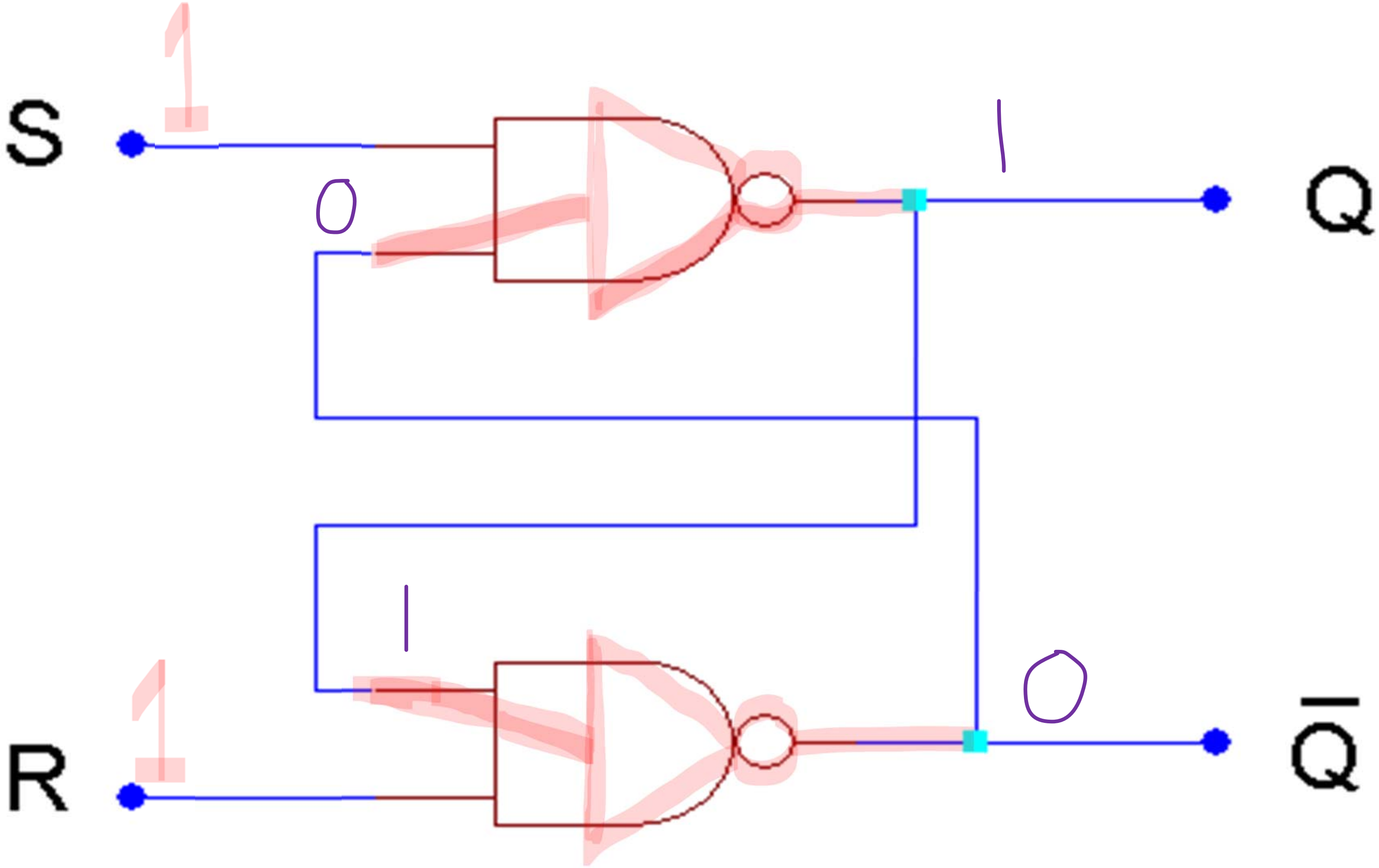
Under normal conditions, both inputs to the latch remain at **1**. This is a stable state.



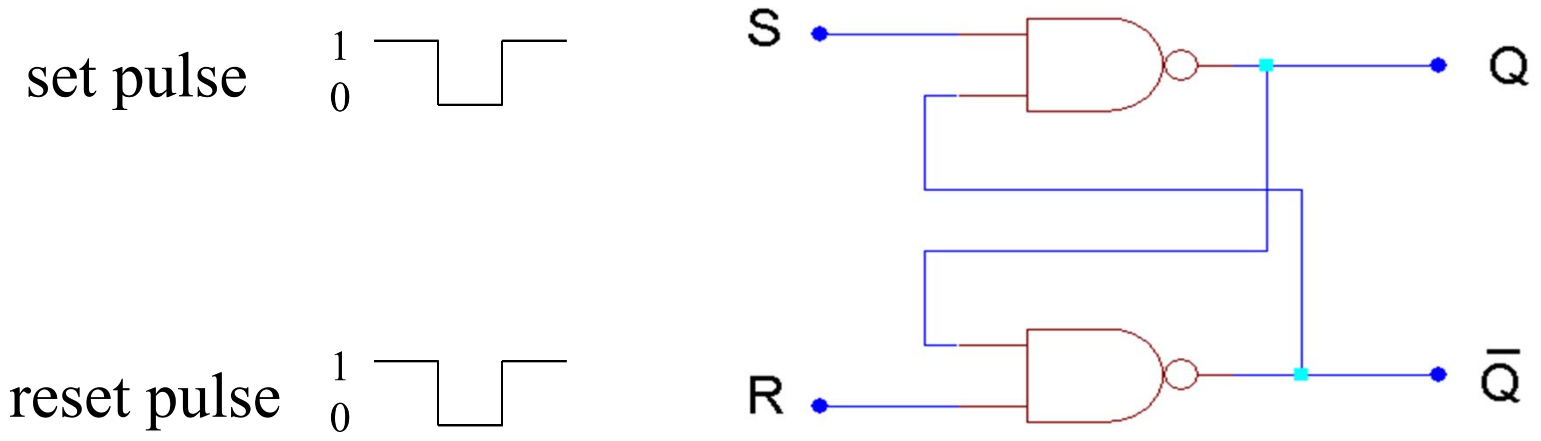
Under normal conditions, both inputs to the latch remain at **1**. This is a stable state.



Under normal conditions, both inputs to the latch remain at **1**. This is a stable state.

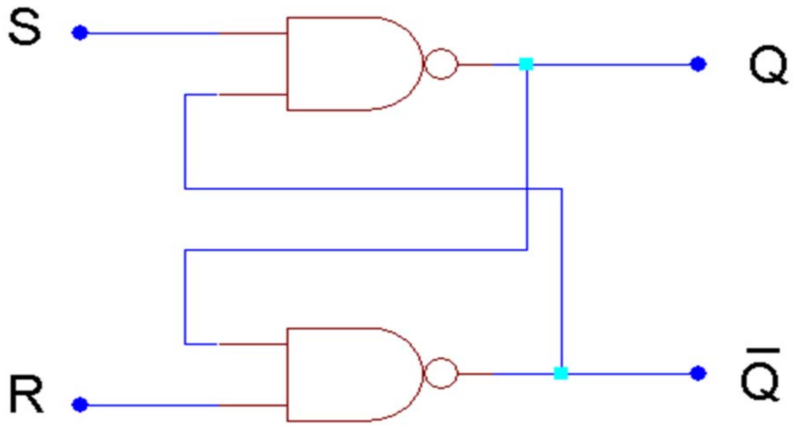


Under normal conditions, both inputs to the latch remain at **1**. This is a stable state. A momentary low pulse applied to the **S** input makes **Q** = **1** and $\overline{\mathbf{Q}}$ = **0**, the SET state. A momentary low pulse applied to the R input makes **Q** = **0** and $\overline{\mathbf{Q}}$ = **1**, the reset state.



Set and reset pulses must not occur at the same time. The inputs should return to '**R** = **S** = **1**' before the next pulse.

If both inputs are equal to **0** at the same time, then an undefined state with both outputs at **1** occurs. When the inputs return to **1**, a race condition will occur which gives an unpredictable next state.



S R	Q
0 0	not allowed
0 1	1
1 0	0
1 1	unchanged

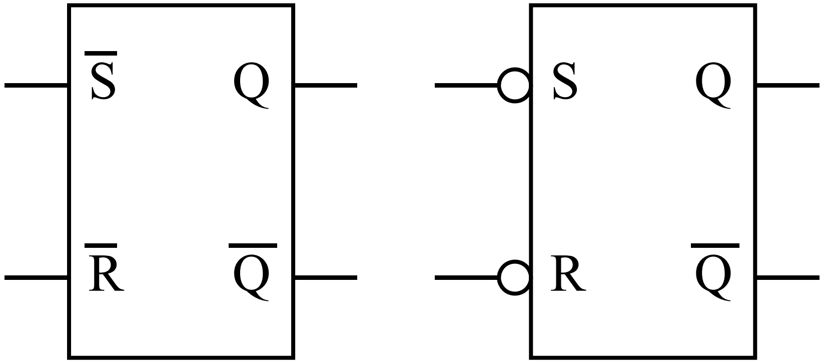
This is an active low SR latch or S’R’ latch. Let **q** be the previous state of **Q**. This gives the characteristic equation shown.

$$Q = \overline{S \,.\, \overline{(R \,.\, q\,)}}$$

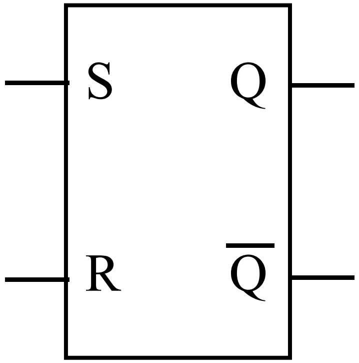
$$= \overline{S} + R \,.\, q$$

and $\overline{S}.\overline{R} = 0$

S’R’ latch symbols

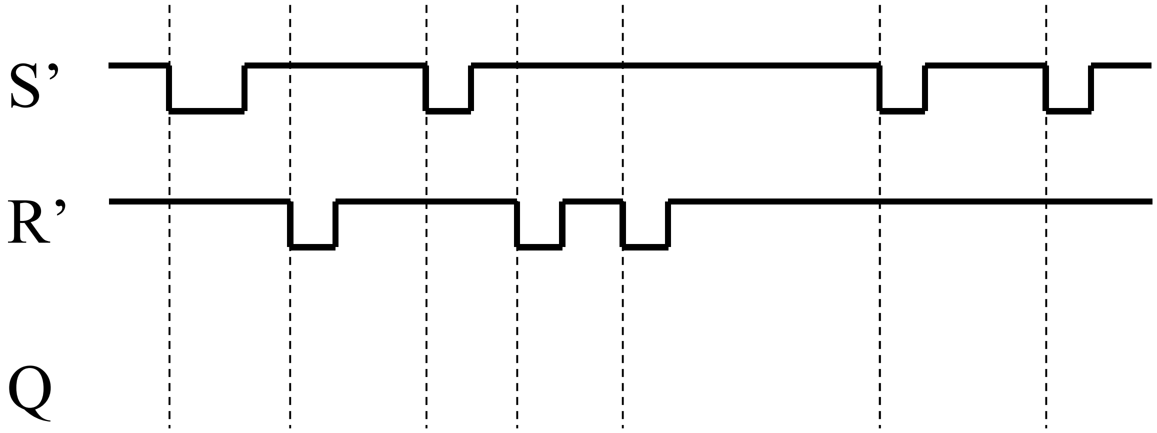


SR latch symbol

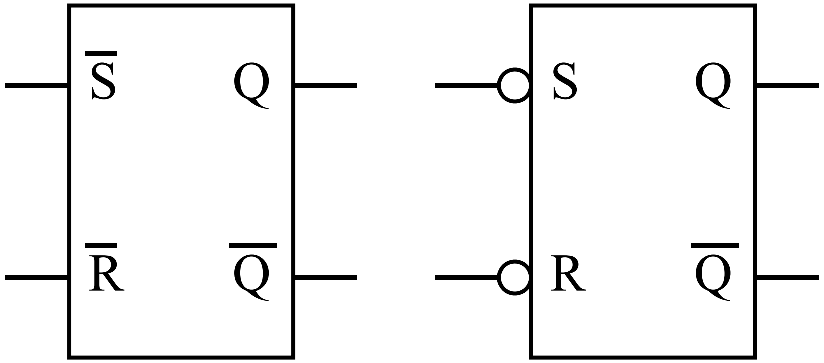


Formed using cross-coupled NOR gates.

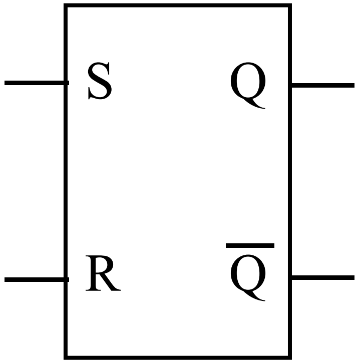
S’R’ latch timing diagram



S’R’ latch symbols

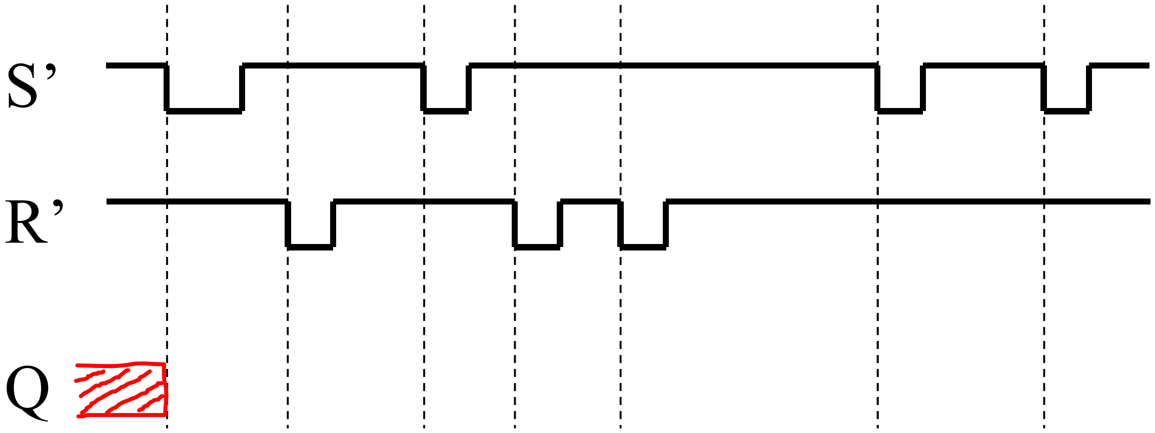


SR latch symbol

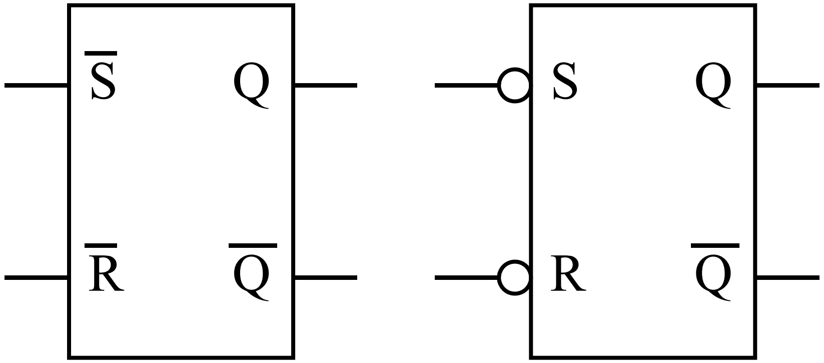


Formed using cross-coupled NOR gates.

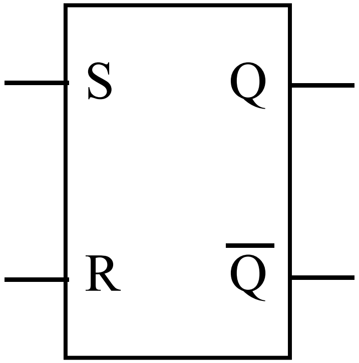
S’R’ latch timing diagram



S’R’ latch symbols

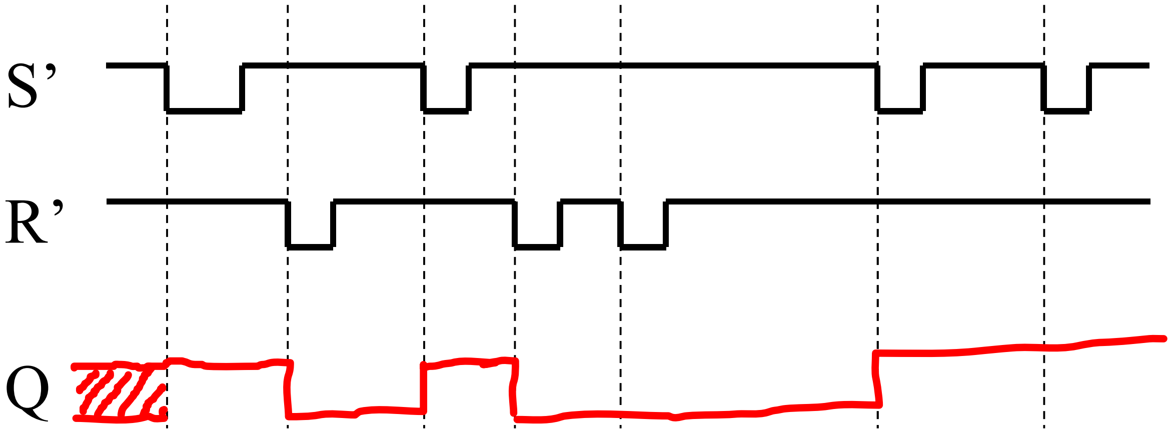


SR latch symbol

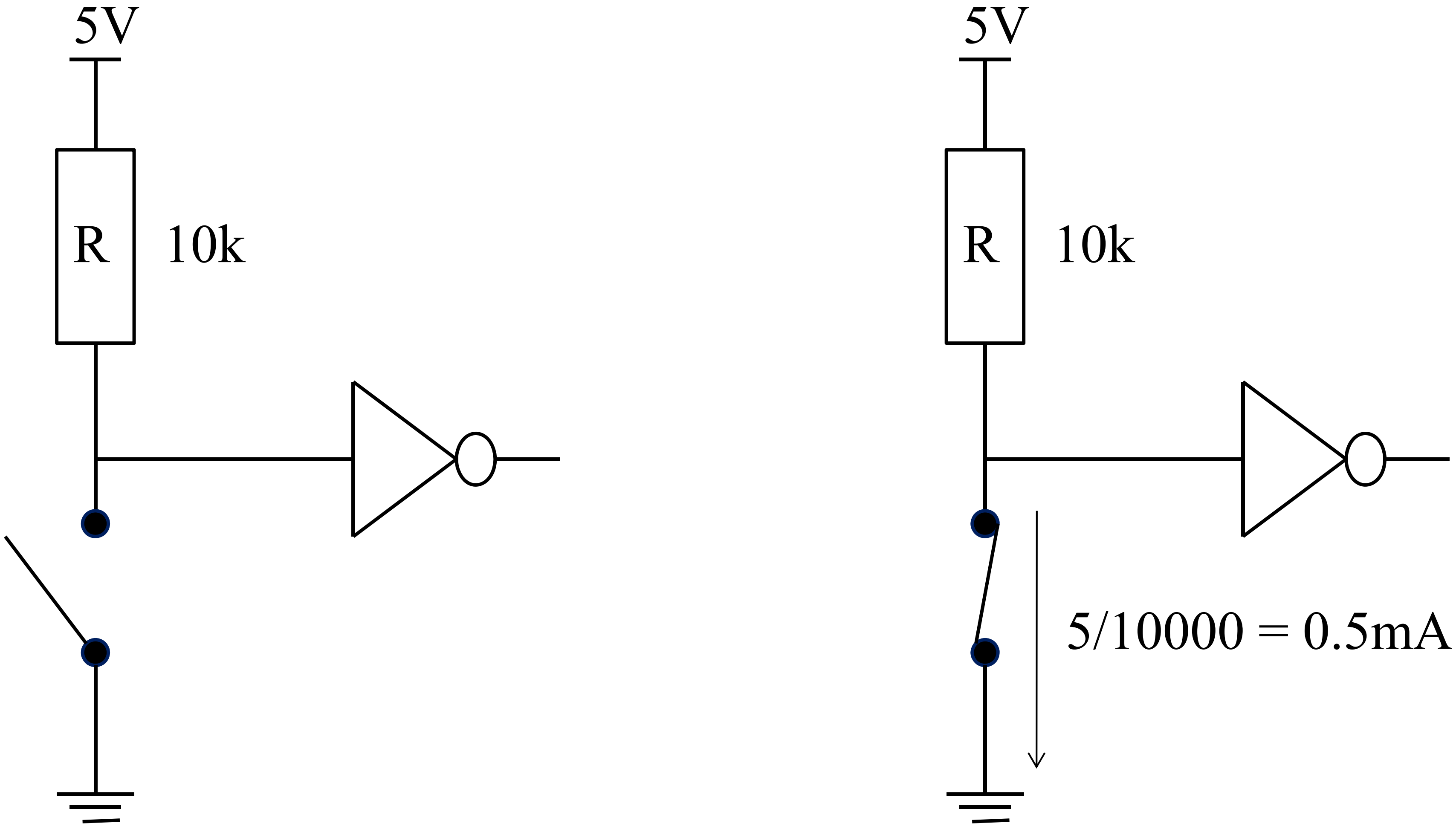


Formed using cross-coupled NOR gates.

S’R’ latch timing diagram

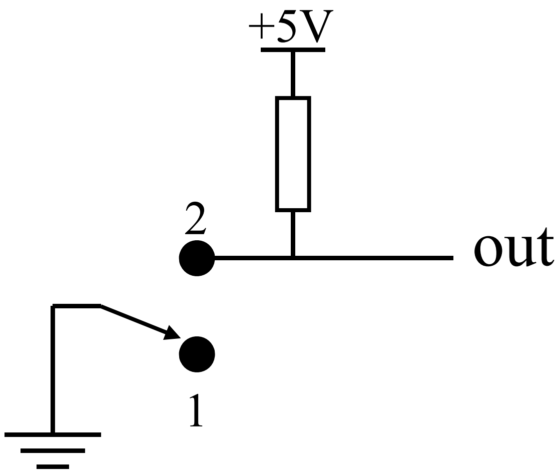


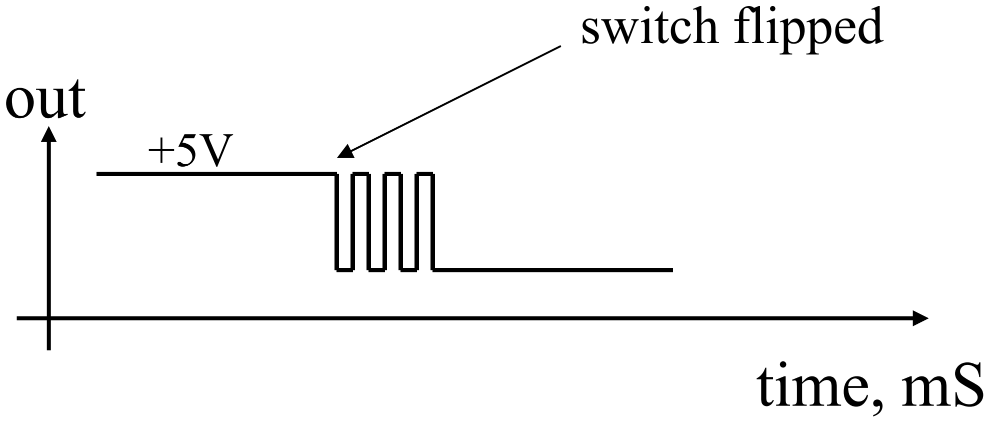
Pull Up Resistor



Switch Contact Bounce

SR latches can be used to eliminate the contact bounce in mechanical switches. Consider the following switch.

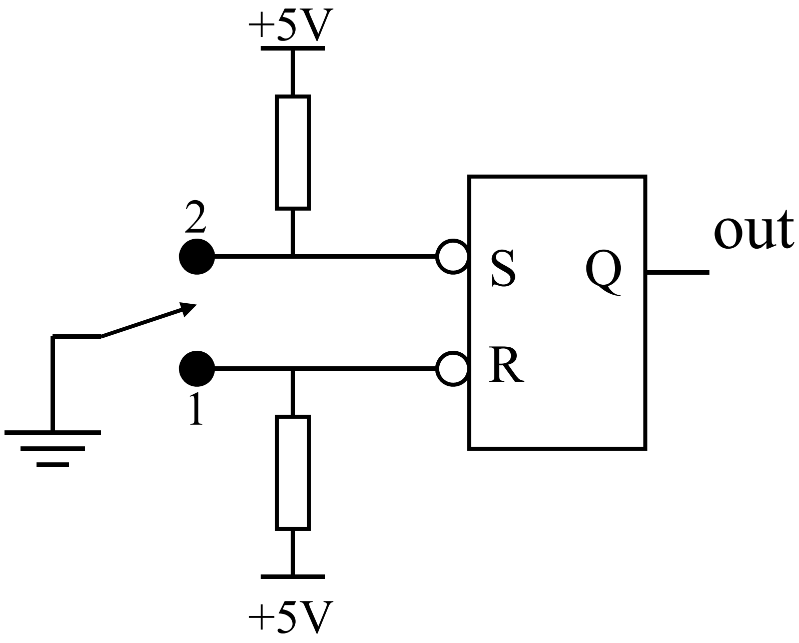




When the switch pin makes contact with position 1 or 2, it will physically vibrate or bounce a few times before settling. These voltage spikes can cause problems in a digital system.

Switch Debouncing

An SR latch can be used to eliminate contact bounce as shown.



The switches common terminal is connected to ground and its other two contacts are pulled up to +5V.

An active-low SR latch is connected to these switch contacts. The invalid 00 state can never occur.

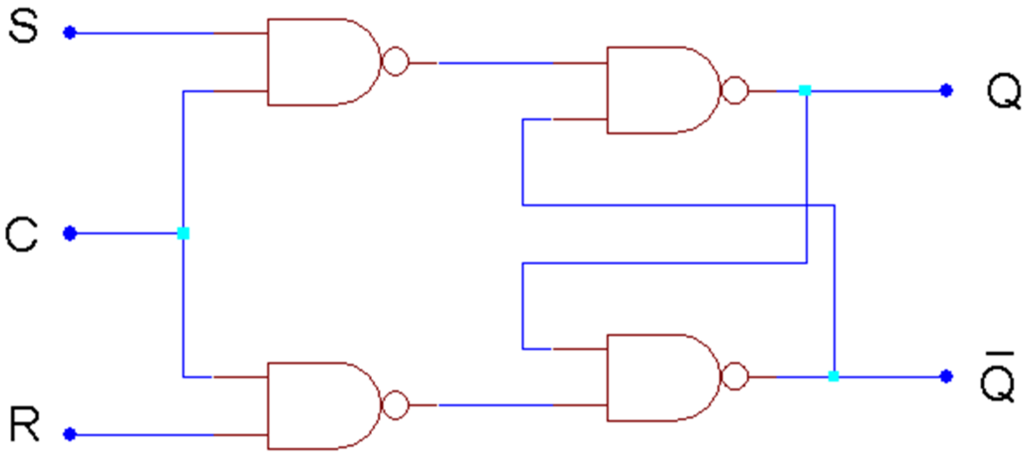
With the switch between contacts, SR = 11, the no change state.

Switching to position 1 resets ‘out’ to 0V. Any input bounces back to 5V results in the no change state, so out remains at 0v.

Switching to position 2 sets ‘out’ to 5V. Any input bounces back to 5V results in the no change state, so out remains at 5V.

Gated SR Latch

The operation of the basic SR latch can be modified by adding a control input to determine when the state of the latch can be changed.



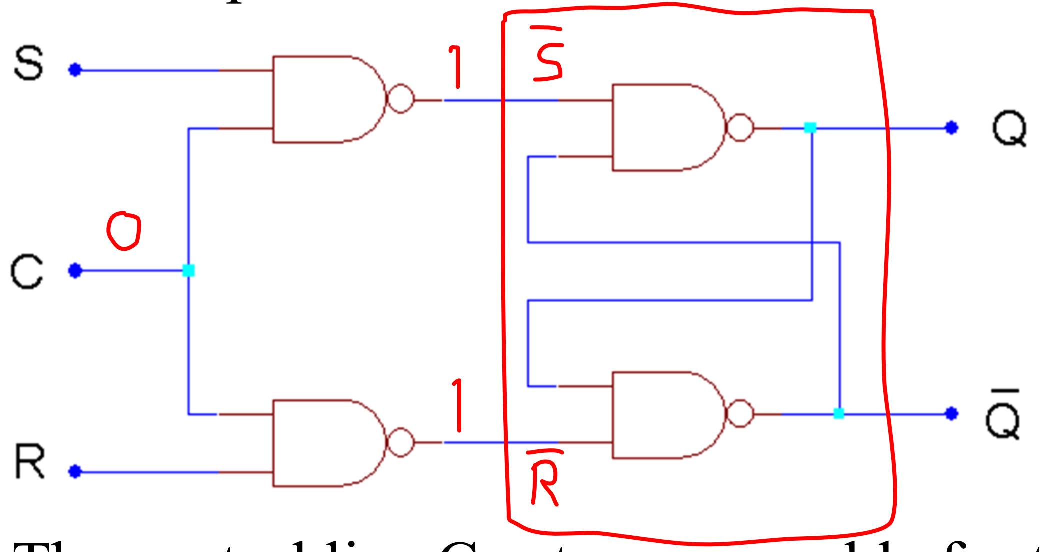
C	S	R	Q
0	x	x	unchanged
1	0	0	unchanged
1	0	1	0
1	1	0	1
1	1	1	not allowed

The control line C acts as an enable for the other two inputs. When C is **0**, the outputs of the first NAND gates stay at **1**, resulting in the ‘unchanged state’ on the SR latch.

When C is **1**, the SR inputs will be inverted by the first NAND gates, then applied to the latch resulting in the function table shown.

Gated SR Latch

The operation of the basic SR latch can be modified by adding a control input to determine when the state of the latch can be changed.



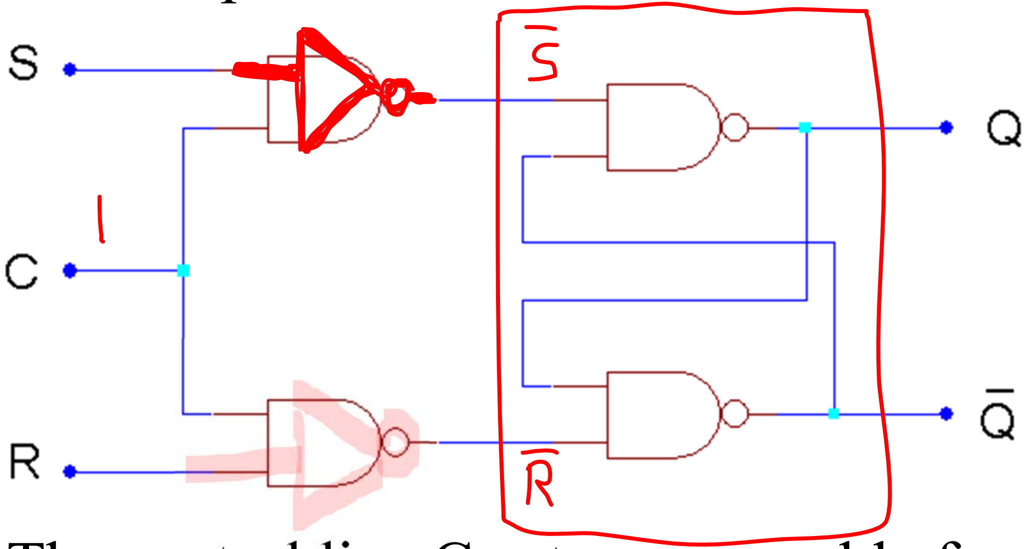
C	S	R	Q
0	x	x	unchanged
1	0	0	unchanged
1	0	1	0
1	1	0	1
1	1	1	not allowed

The control line C acts as an enable for the other two inputs. When C is **0**, the outputs of the first NAND gates stay at **1**, resulting in the ‘unchanged state’ on the SR latch.

When C is **1**, the SR inputs will be inverted by the first NAND gates, then applied to the latch resulting in the function table shown.

Gated SR Latch

The operation of the basic SR latch can be modified by adding a control input to determine when the state of the latch can be changed.



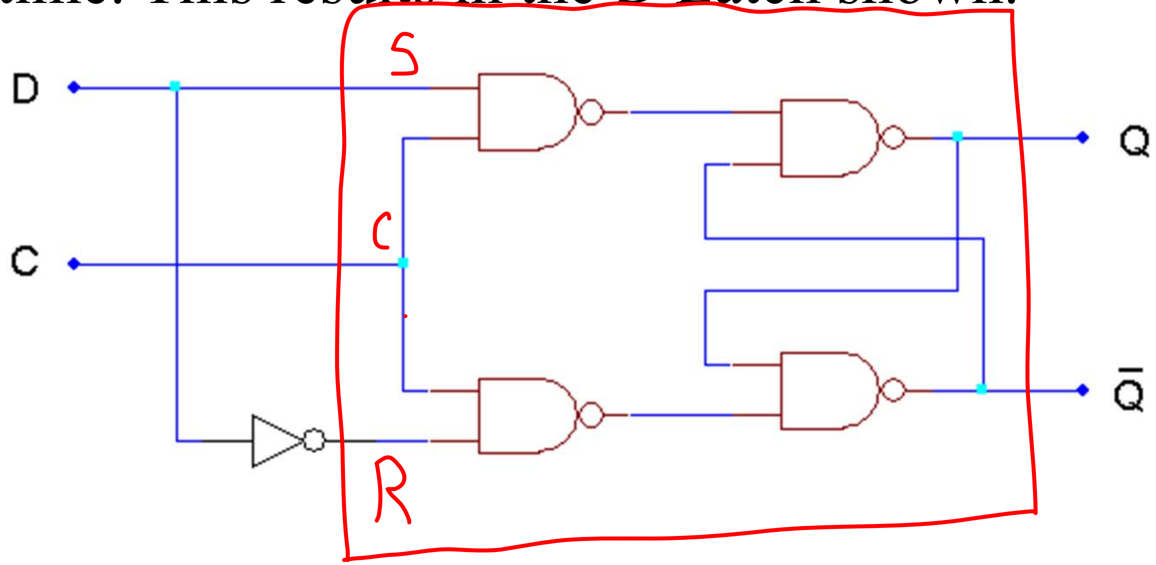
C	S	R	Q
0	x	x	unchanged
1	0	0	unchanged
1	0	1	0
1	1	0	1
1	1	1	not allowed

The control line C acts as an enable for the other two inputs. When C is **0**, the outputs of the first NAND gates stay at **1**, resulting in the 'unchanged state' on the SR latch.

When C is **1**, the SR inputs will be inverted by the first NAND gates, then applied to the latch resulting in the function table shown.

D Latch

The undesirable indeterminate state of the SR latch can be eliminated by ensuring that the S and R inputs are never equal to ‘1’ at the same time. This results in the D Latch shown.



C	D	Q
0	x	unchanged
1	0	0
1	1	1

When C is low, the SR latch has inputs **11**, the unchanged state.

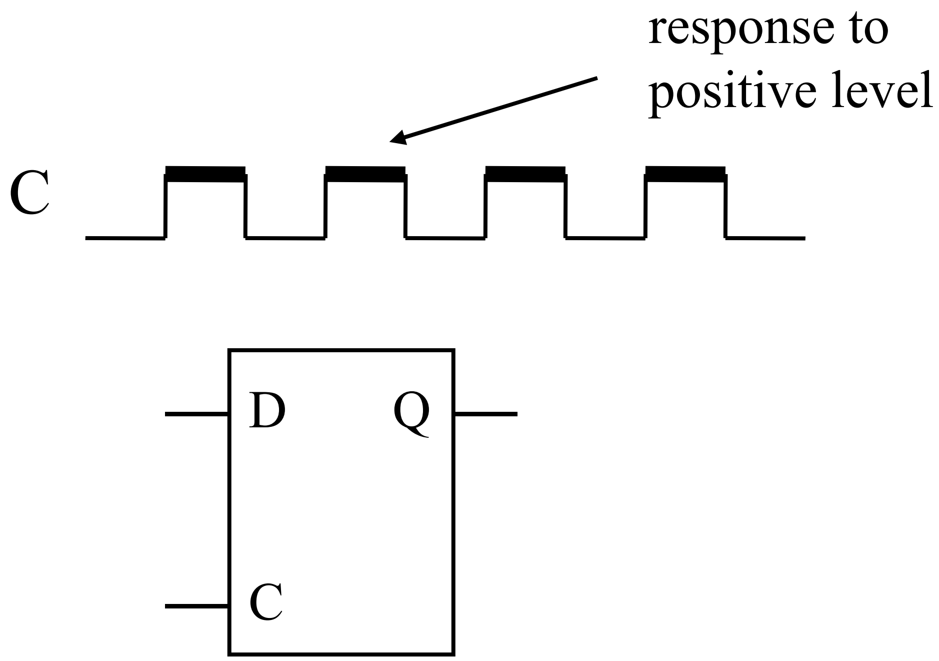
When C is high, the D input is sampled:

D = ‘0’ gives Q = ‘0’ D = ‘1’ gives Q = ‘1’

Level Sensitive Operation

The operation of the D Latch is **level sensitive**.

As long as the control input is high, the latch will respond to any changes in D. It is sometimes known as a transparent latch.

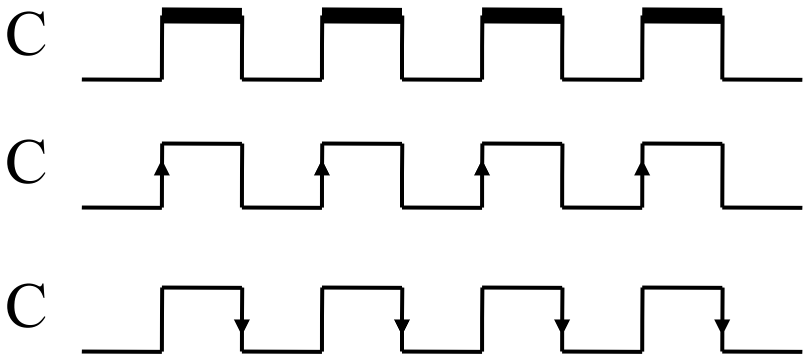


When the control input goes low, the binary information present on the data input is retained at the Q output until the control input goes high again.

The D Latch is sometimes used as temporary storage at interfaces.

Edge Triggering

Level sensitive devices have a serious problem if they are used in sequential circuits. When the clock pulse changes to its active level, changes will propagate through the latch. These changes will propagate through subsequent latches resulting in incorrect states. For correct operation, the flip-flops should only change state at a clock transition.



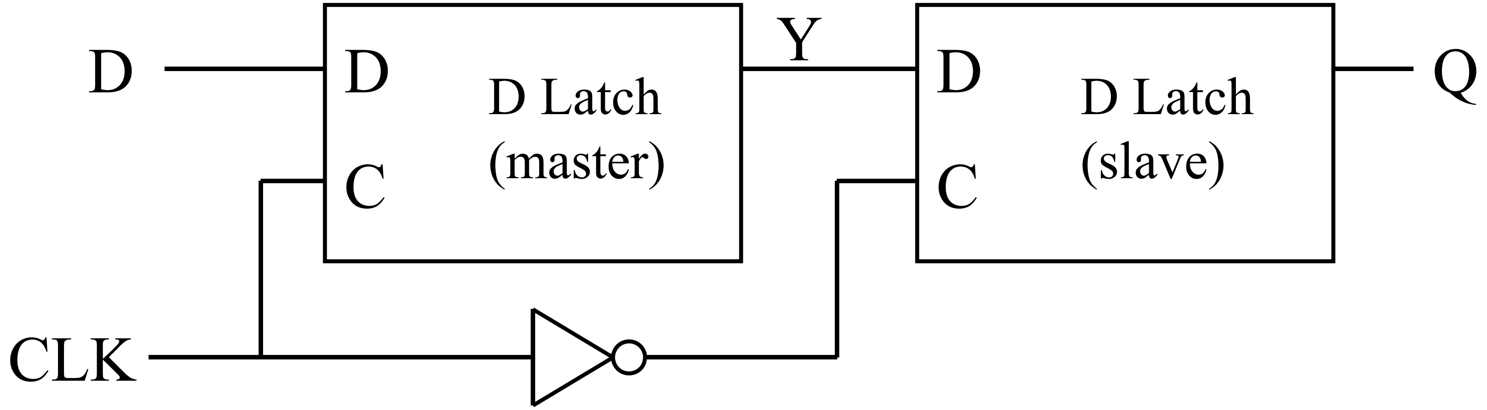
Level sensitive

Positive edge triggered

Negative edge triggered

Edge Triggered D Flip-Flop

A D-Type flip-flop can be constructed with two D Latches and an inverter connected in a master-slave arrangement.



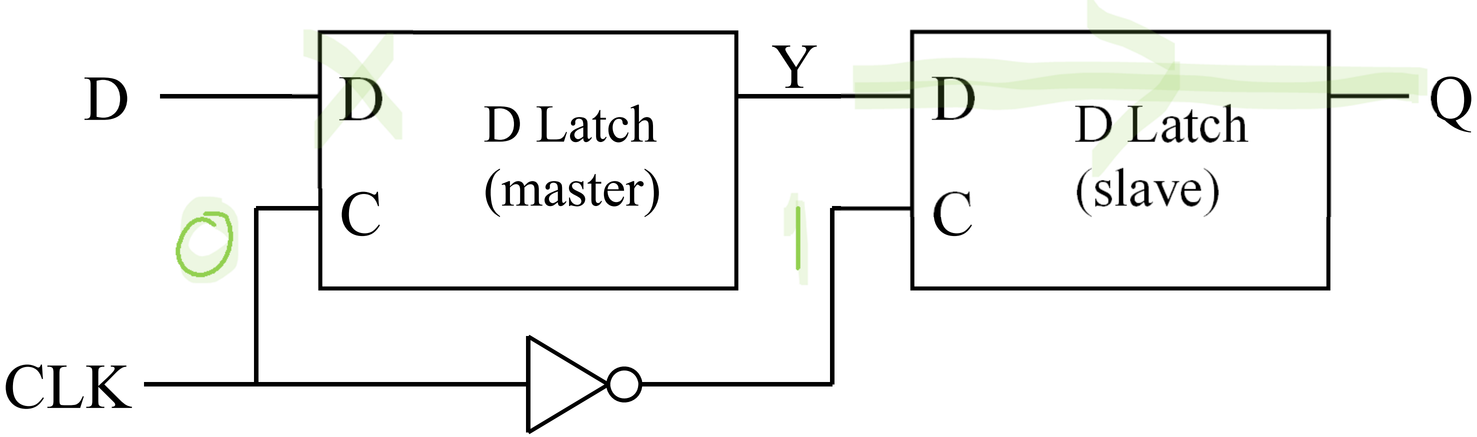
When CLK is ‘0’, the master latch is disabled, and the slave is enabled taking the value of Y from the master.

When CLK changes to ‘1’, the slave latch is disabled, holding the value of Y on its Q output. The master latch is enabled and data on D passes through to Y but can not effect the Q output.

The output can only change on the falling edge of the clock.

Edge Triggered D Flip-Flop

A D-Type flip-flop can be constructed with two D Latches and an inverter connected in a master-slave arrangement.



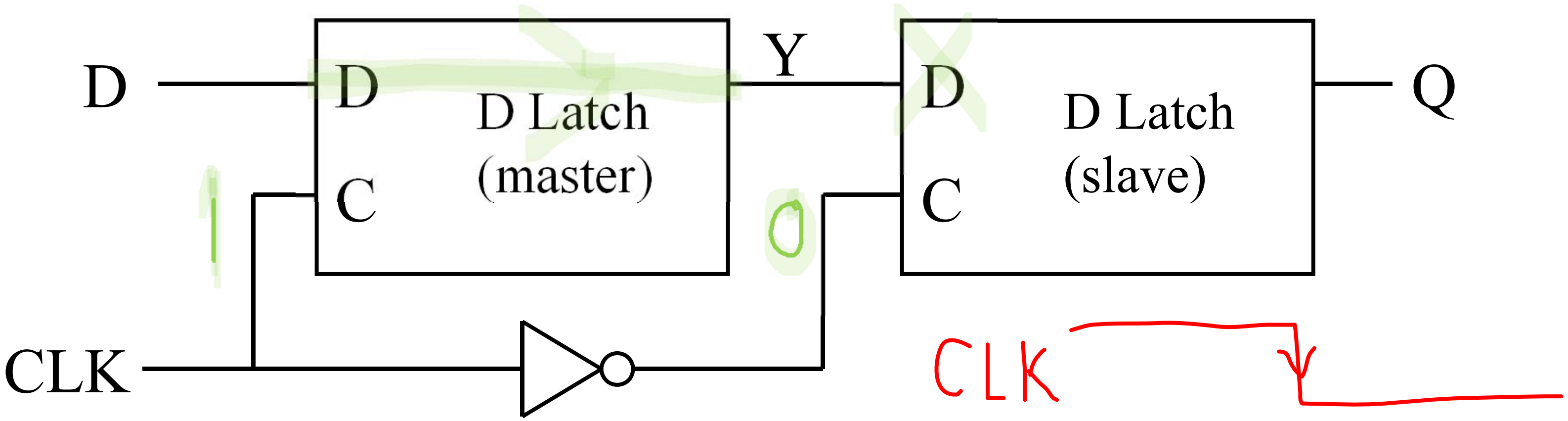
When CLK is ‘0’, the master latch is disabled, and the slave is enabled taking the value of Y from the master.

When CLK changes to ‘1’, the slave latch is disabled, holding the value of Y on its Q output. The master latch is enabled and data on D passes through to Y but can not effect the Q output.

The output can only change on the falling edge of the clock.

Edge Triggered D Flip-Flop

A D-Type flip-flop can be constructed with two D Latches and an inverter connected in a master-slave arrangement.



When CLK is ‘0’, the master latch is disabled, and the slave is enabled taking the value of Y from the master.

When CLK changes to ‘1’, the slave latch is disabled, holding the value of Y on its Q output. The master latch is enabled and data on D passes through to Y but can not effect the Q output.

The output can only change on the falling edge of the clock.

Summary

- Asynchronous circuits can exist in stable and unstable states.
- Latches can be formed from a cross-coupled arrangement of NAND or NOR gates.
- Latches can be used to debounce mechanical switches.
- Latches are level sensitive.
- An edge triggered flip-flop can be formed from two latches in a master-slave configuration.