Sequential circuit and Latches

Sparsh Mittal

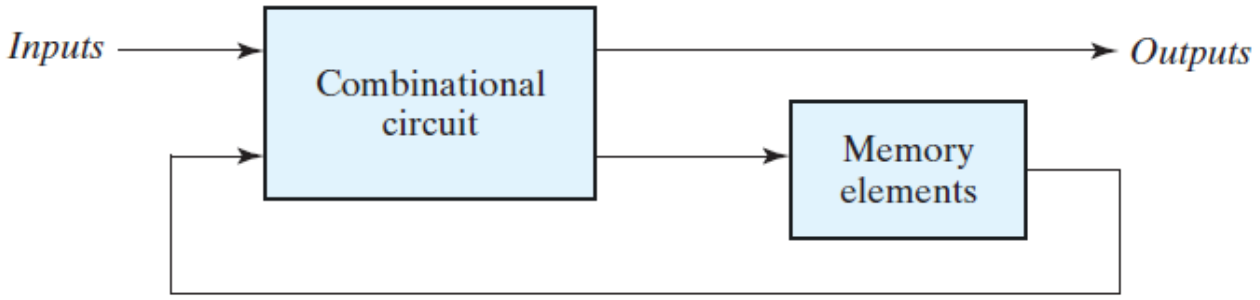
Sequential circuit

It consists of a combinational circuit to which storage elements are connected to form a feedback path.

The binary information stored in storage elements at any given time defines the state of the sequential circuit at that time.

The sequential circuit receives binary information from external inputs that, together with the present state of the storage elements, determine the binary value of the outputs.

These external inputs also determine the condition for changing the state in the storage elements.



A sequential circuit is specified by a time sequence of inputs, outputs, and internal states.

In contrast, the outputs of combinational logic depend only on the present values of the inputs.

Comparison

We have examined designs of circuit elements that can **store information**
Now, we will use these elements to build circuits that **remember** past inputs

Combinational
Only depends on current inputs

Sequential
Opens depending on past inputs

https://www.easykeys.com/228_ESP_Combination_Lock.aspx
https://www.fosmon.com/product/tsa-approved-lock-4-dial-combo

In order for this lock to work, it has to keep track (**remember**) of the past events!

If passcode is **R13-L22-R3**, sequence of **states** to unlock:

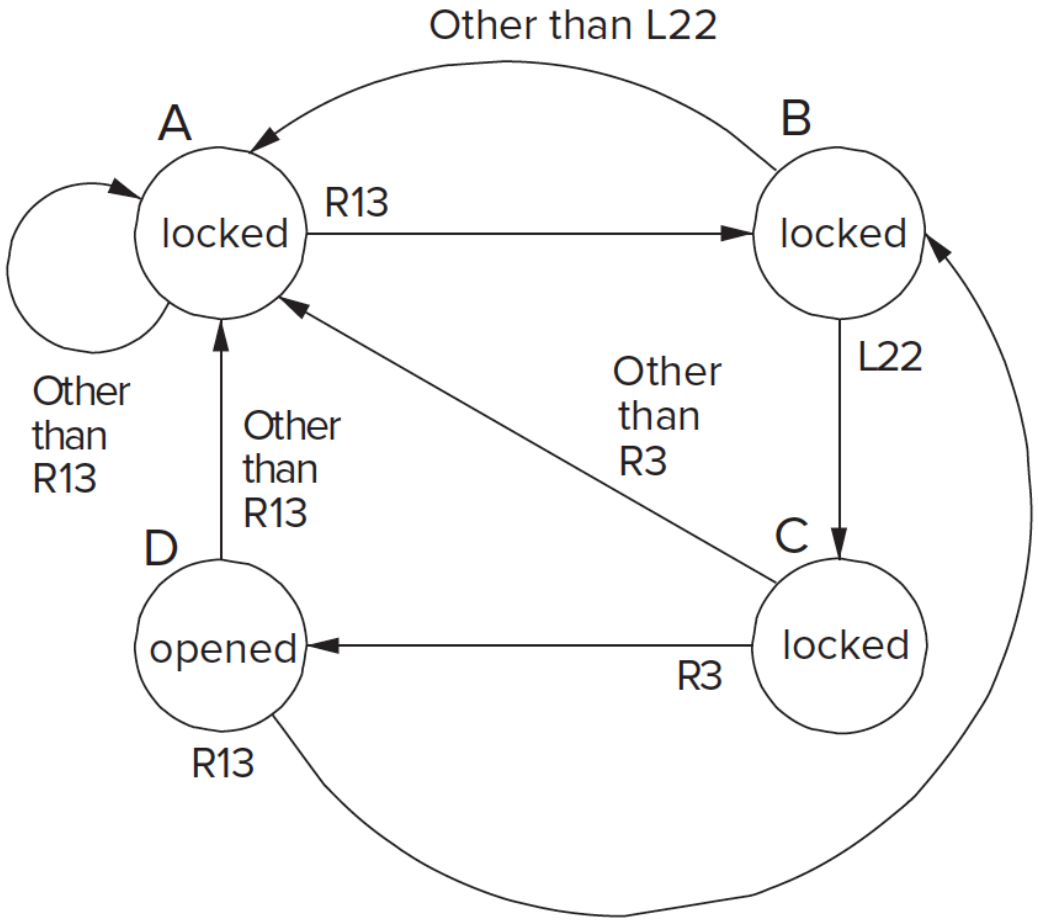
A. The lock is not open (locked), and no relevant operations have been performed
B. Locked but user has completed R13
C. Locked but user has completed R13-L22
D. Unlocked: user has completed R13-L22-R3

The **state** of a system is a snapshot of all relevant elements of the system at the moment of the snapshot

To open the lock,

If anything else happens (e.g., L5), lock **returns** to state A

State Diagram of Our Sequential Lock



Completely describes the operation of the sequential lock

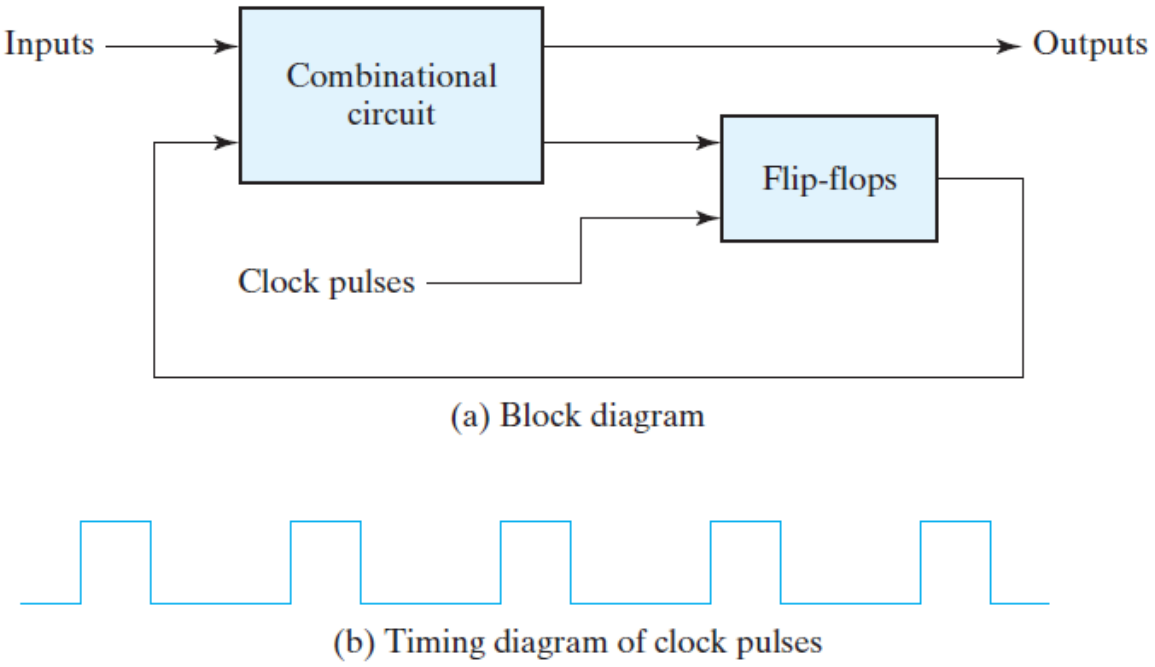
Image source: Patt and Patel, "Introduction to Computing Systems", 2nd ed., page 76.

Synchronous and *asynchronous* sequential circuit

A *synchronous* sequential circuit is a system whose behavior can be defined from the knowledge of its signals at discrete instants of time.

The behavior of an *asynchronous* sequential circuit depends upon the input signals at any instant of time *and* the order in which the inputs change.

Synchronization is achieved by a clock pulses.



(a) Block diagram

(b) Timing diagram of clock pulses

Synchronous clocked sequential circuit

Latches

Latches are the basic circuits from which all flip-flops are constructed.

Although latches are useful for storing binary information and for the design of asynchronous sequential circuits, they are not practical for use as storage elements in synchronous sequential circuits.

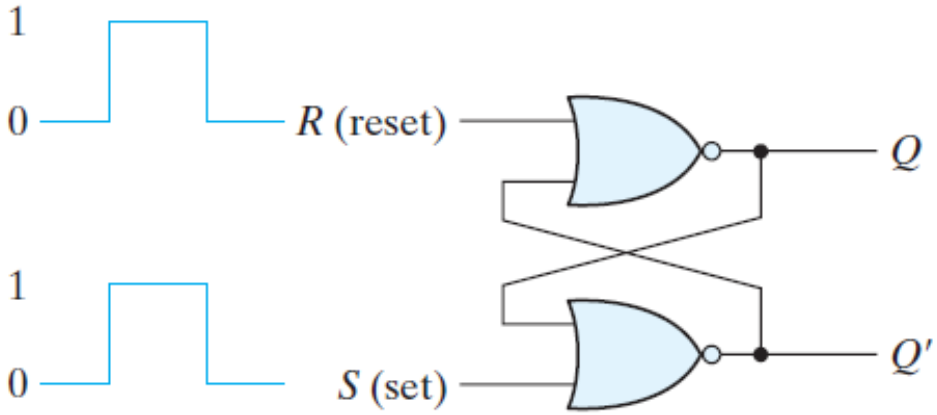
SR Latch

Designed with two cross-coupled NOR gates or two cross-coupled NAND gates.

The latch has two useful states. When output $Q = 1$ and $Q' = 0$, the latch is said to be in the *set state*.

When $Q = 0$ and $Q' = 1$, it is in the *reset state*.

Setting both inputs to 1 is forbidden



(a) Logic diagram

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

(b) Function table

Working

Under normal conditions, both inputs of the latch remain at 0 unless state has to be changed.

The application of a momentary 1 to the *S* input causes the latch to go to the set state.

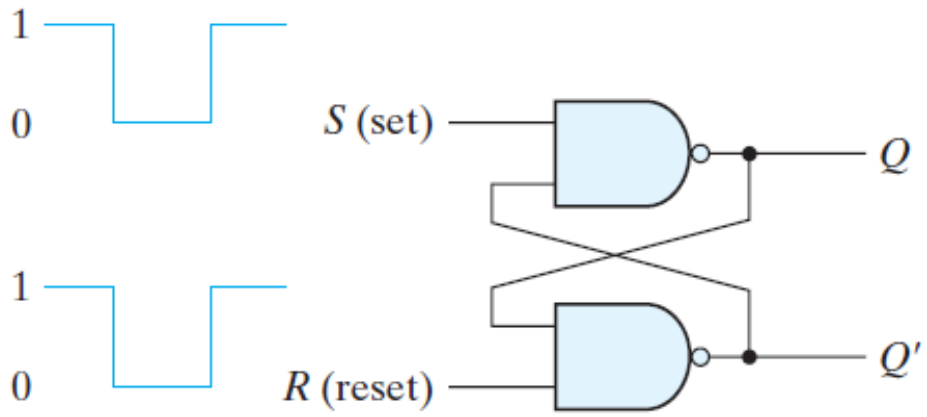
The *S* input must go back to 0 before any other changes take place, to avoid the occurrence of an undefined next state that results from forbidden input condition.

After both inputs return to 0, it is then possible to shift to the reset state by momentary applying a 1 to the *R* input.

The 1 can then be removed from *R*, so the circuit remains in the reset state.

When both inputs *S* and *R* are equal to 0, the latch can be in either the set or the reset state, depending on which input was most recently a 1.

SR latch with NAND gates



(a) Logic diagram

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

(b) Function table

NAND vs NOR latch

Input signals for the NAND require the complement of those values used for the NOR latch.

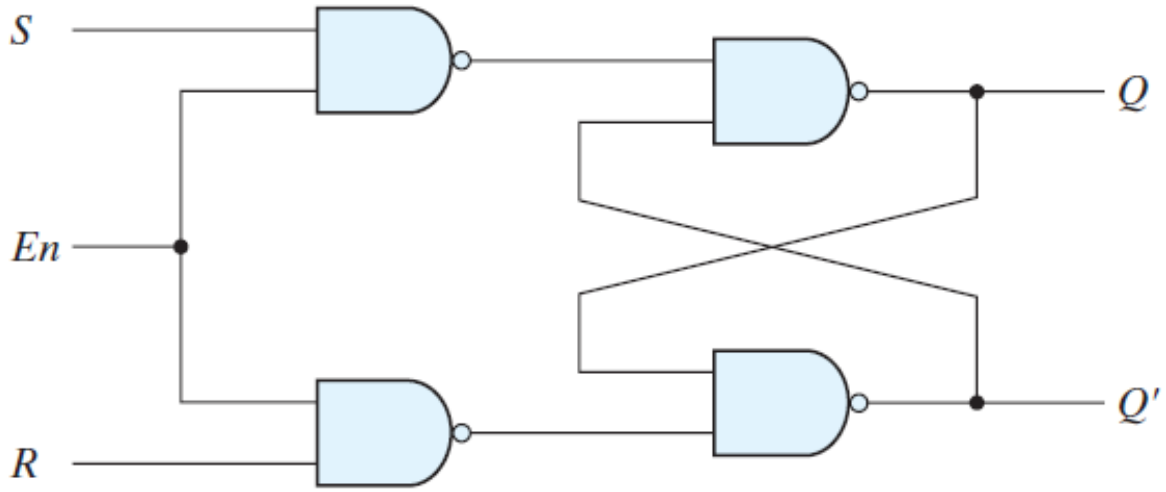
Because the NAND latch requires a 0 signal to change its state, it is sometimes referred to as an *S'R'* latch.

SR latch with control input

Control input decides *when* the state of the latch can be changed.

It disables the circuit by applying 0 to *En*, so that the state of the output does not change regardless of the values of *S* and *R*.

When *En* = 1 and both the *S* and *R* inputs are equal to 0, the state of the circuit does not change.



(a) Logic diagram

En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table

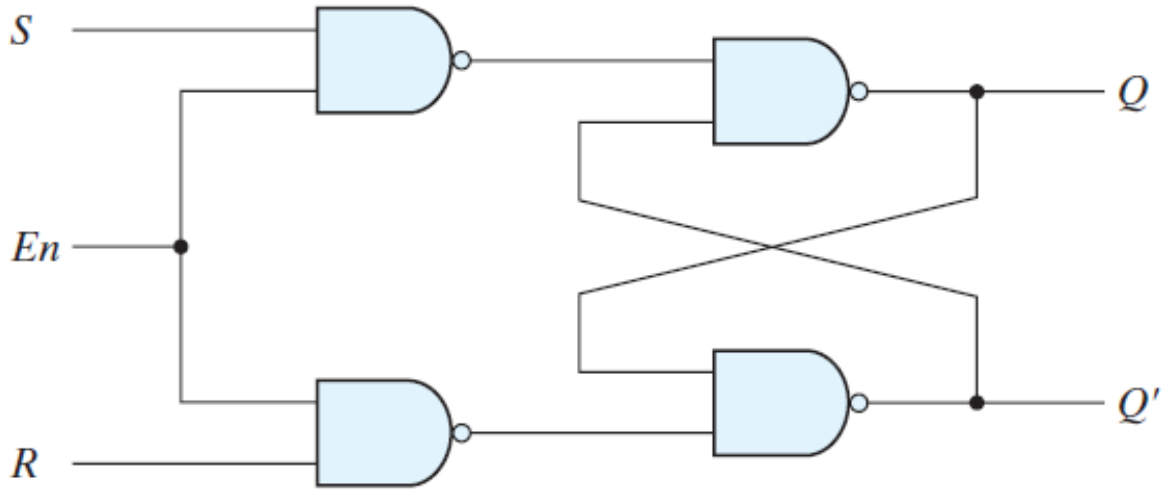
SR latch with control input

An indeterminate condition occurs when all three inputs are equal to 1.

This condition places 0's on both inputs of the basic *SR* latch, which puts it in the undefined state.

When the enable input goes back to 0, one cannot conclusively determine the next state, because it depends on whether the *S* or *R* input goes to 0 first.

This circuit becomes difficult to manage, and it is seldom used in practice.



(a) Logic diagram

En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table

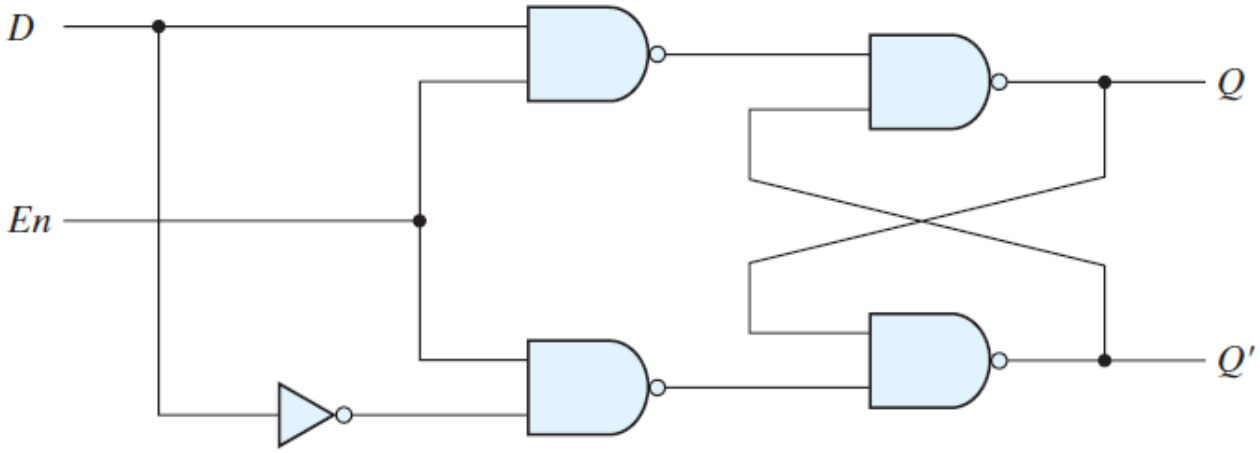
D Latch (Transparent Latch)

D latch ensures that inputs S and R are never equal to 1 at same time.

The D input is sampled when $En = 1$.

If $D = 1$, the Q output goes to 1, placing the circuit in the set state.

If $D = 0$, output Q goes to 0, placing the circuit in the reset state.



(a) Logic diagram

En	*D*	Next state of *Q*
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state

(b) Function table

When the enable input signal is de-asserted, the binary information that was present at the data input at the time the transition occurred is retained (i.e., stored) at the Q output until the enable input is asserted again.

Graphic symbols for latches