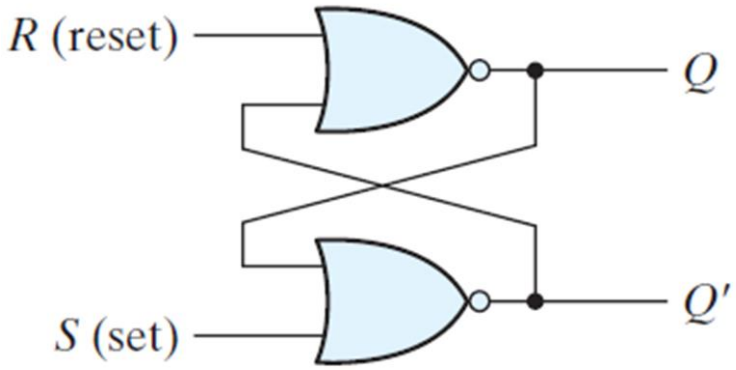
Lecture 17 – Sequential circuits 2

Chapter 5

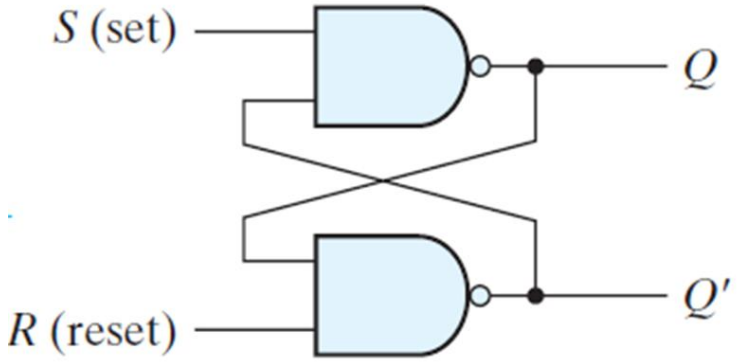
SR Latch

- The (Set – Reset) *SR* latch is a circuit with two cross-coupled NOR gates or two cross-coupled NAND gates, and two inputs labeled *S* and *R*



SR NOR Latch

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)

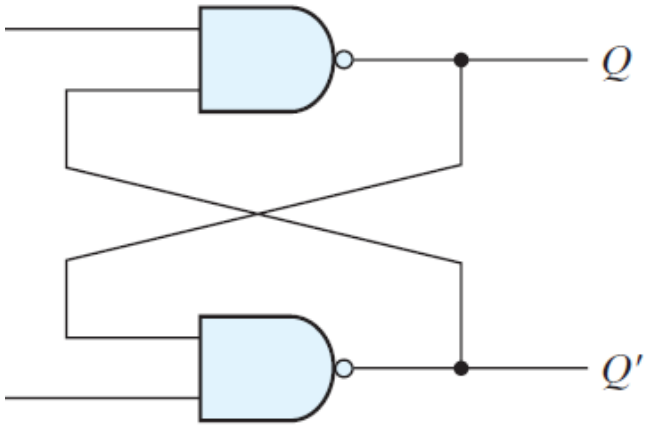


SR NAND Latch (S'R' latch)

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)

Latch with enable

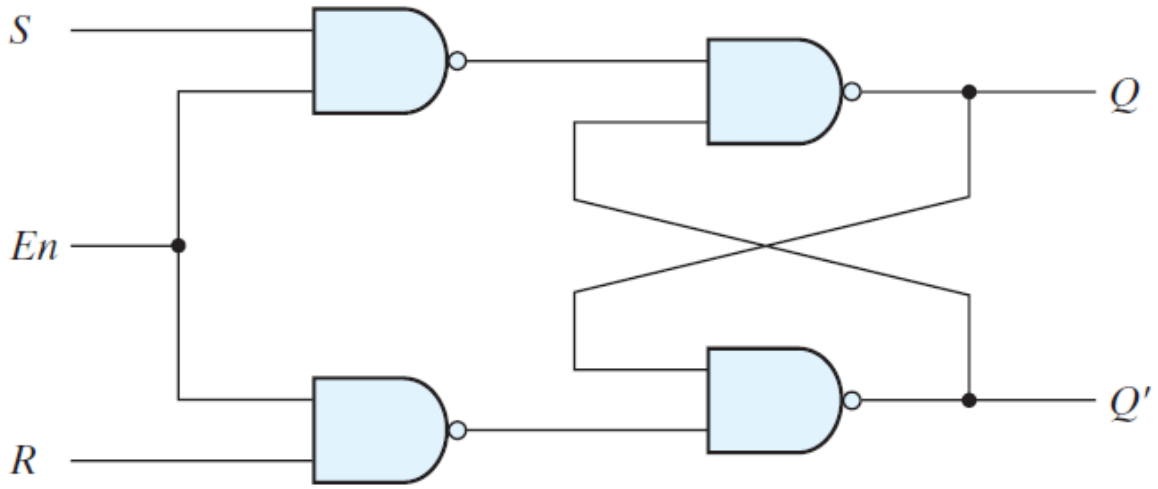
- The operation of the basic *SR* latch can be modified by providing an additional input signal that determines (controls) *when* the state of the latch can be changed by *S* and *R* (*S'* and *R'*)
- It consists of the basic *SR* latch and two additional NAND gates
- The control input *En* acts as an *enable* signal for the other two inputs
- The outputs of the NAND gates stay at the logic-1 level as long as the enable signal remains at 0
- This is the quiescent condition for the *SR* latch



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

Latch with enable

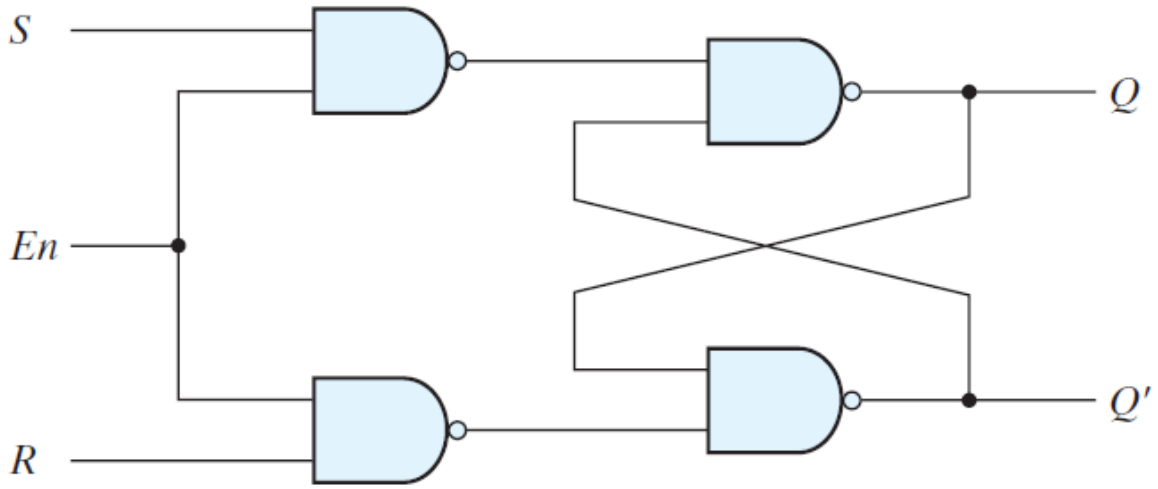
- When the enable input goes to 1, information from the *S* or *R* input is allowed to affect the latch
  - The set state is reached with *S* = 1, *R* = 0, and *En* = 1 (active-high enabled) and reset is reached with *S* = 0, *R* = 1, and *En* = 1
  - In either case, when *En* returns to 0, the circuit remains in its previous stable state
  - Further, when *En* = 1 and both the *S* and *R* inputs are equal to 0, the state of the circuit does not change



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

Latch with enable

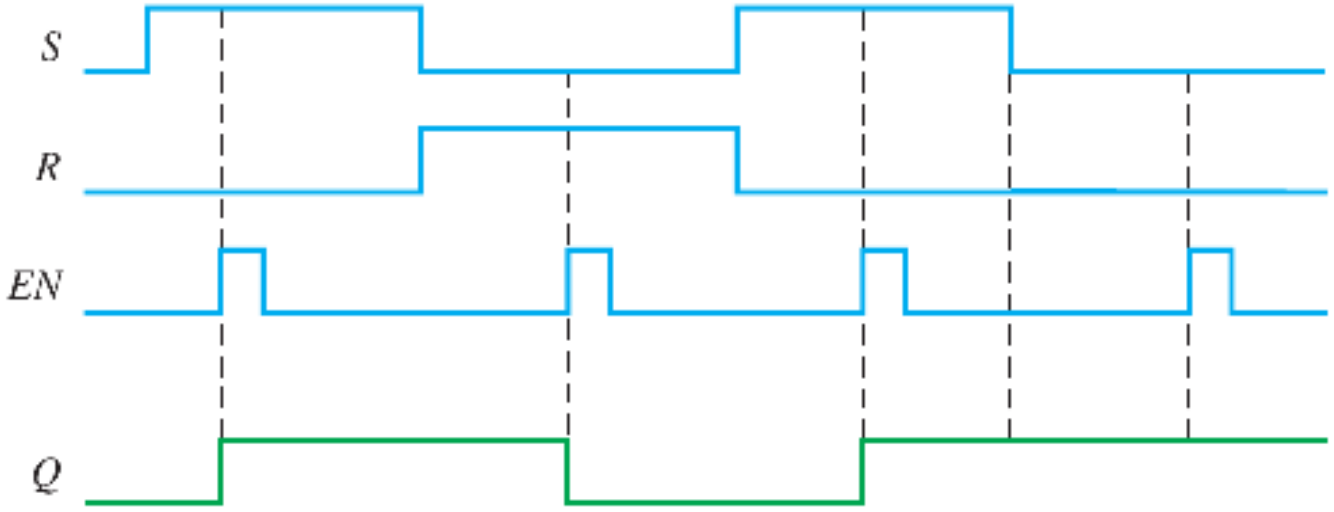
- *An indeterminate condition occurs when all three inputs are equal to 1*
  - This condition places 0's on both inputs of the basic *SR* latch, which puts it in the undefined state
  - When the enable input goes back to 0, one cannot conclusively determine the next state, because it depends on whether the *S* or *R* input goes to 0 first
  - This indeterminate condition makes this circuit difficult to manage, and it is seldom used in practice
- Nevertheless, the *SR* latch is an important circuit because flip-flops are constructed from it

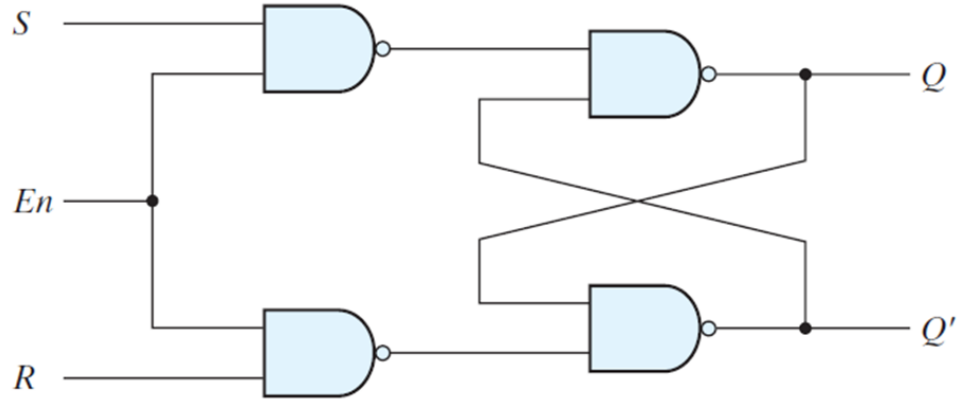


En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

Latch with enable

Eg: Determine the Q output waveform if the inputs shown are applied to a gated S-R latch with enable. Assume the latch is initially RESET.

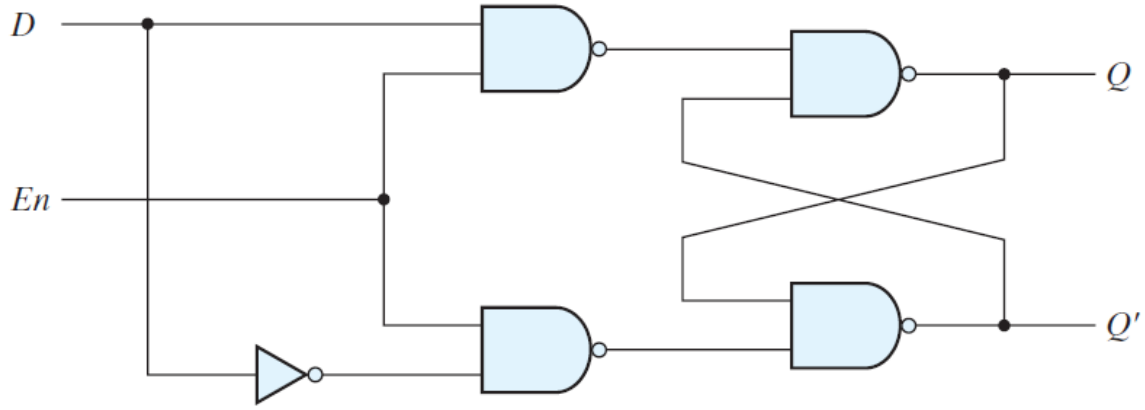




En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

D Latch

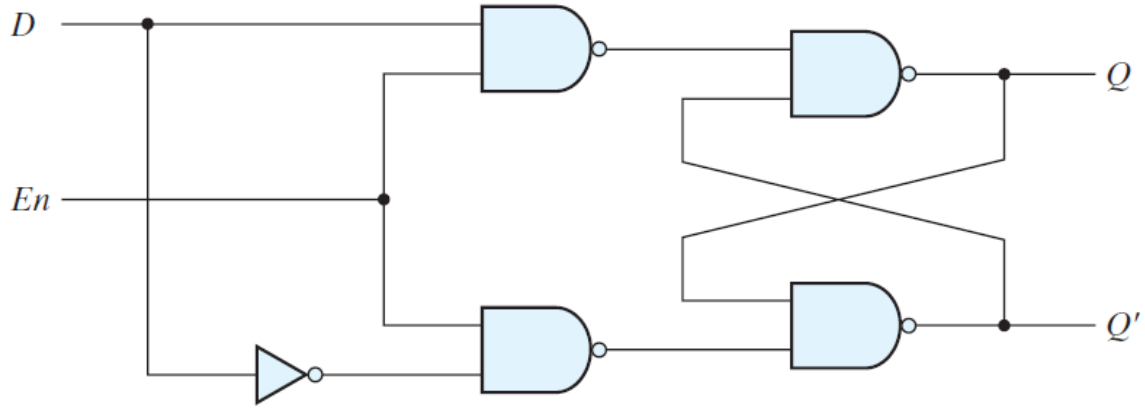
- One way to eliminate the undesirable condition of the indeterminate state in the *SR* latch is to ensure that inputs *S* and *R* are never equal to 1 at the same time
- This is done in the *D* latch
  - The *D* input goes directly to the *S* input, and its complement is applied to the *R* input
  - As long as the enable input is at 0, the cross-coupled *SR* latch has both inputs at the 1 level and the circuit cannot change state regardless of the value of *D*
  - 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



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

D Latch

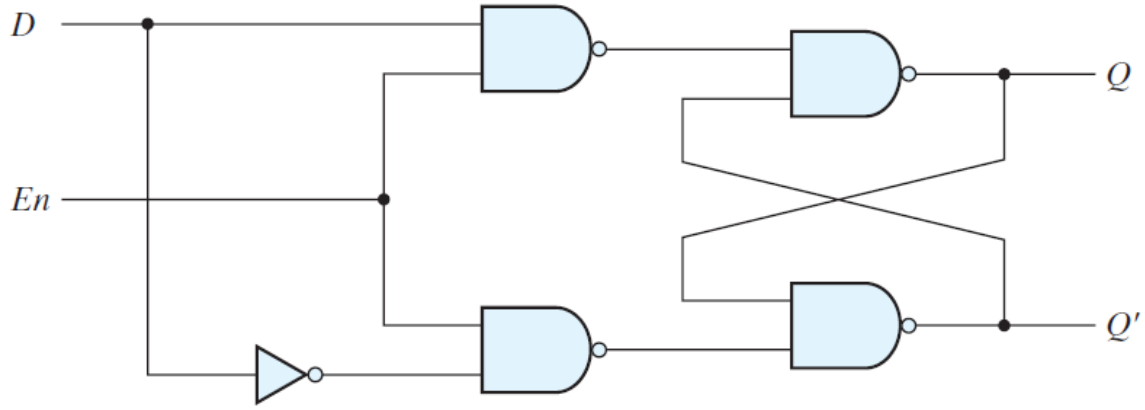
- The *D* latch receives that designation from its ability to hold *data* in its internal storage
- It is suited for use as a temporary storage for binary information between a unit and its environment
- The binary information present at the data input of the *D* latch is transferred to the *Q* output when the enable input is asserted
- The output follows changes in the data input as long as the enable input is asserted
- This situation provides a path from input *D* to the output, and for this reason, the circuit is often called a *transparent* latch



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

D Latch

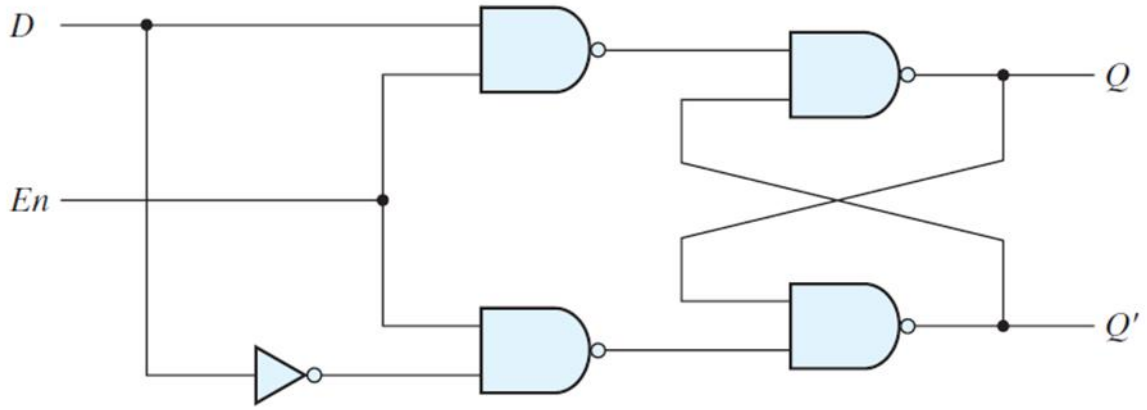
- When the enable input signal is de-asserted, the binary information that was present at the data input at the time the transition occurred is retained (i.e., stored) at the *Q* output until the enable input is asserted again
- Note that an inverter could be placed at the enable input
- Then, depending on the physical circuit, the external enabling signal will be a value of 0 (active low) or 1 (active high)

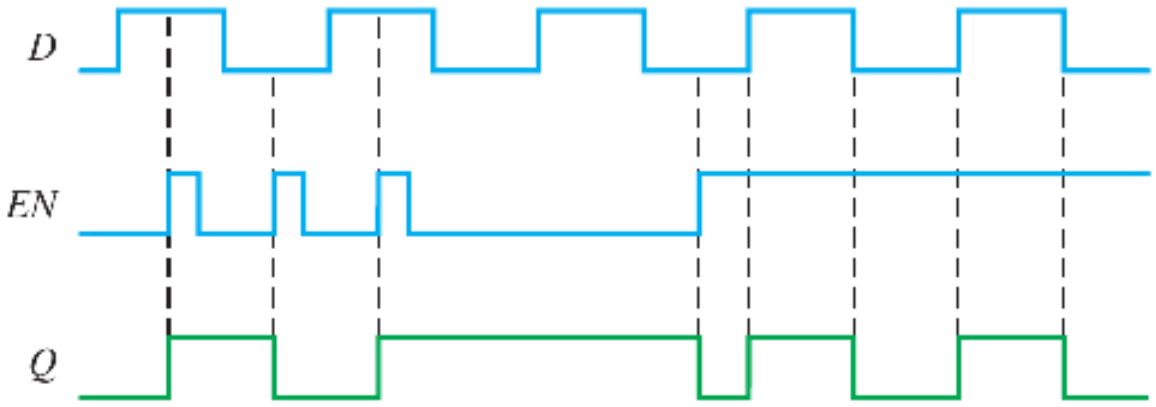


En	*D*	Next state of *Q*
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state

D Latch

Eg: Determine the Q output waveform if the inputs shown in below are applied to a D latch, which is initially RESET.

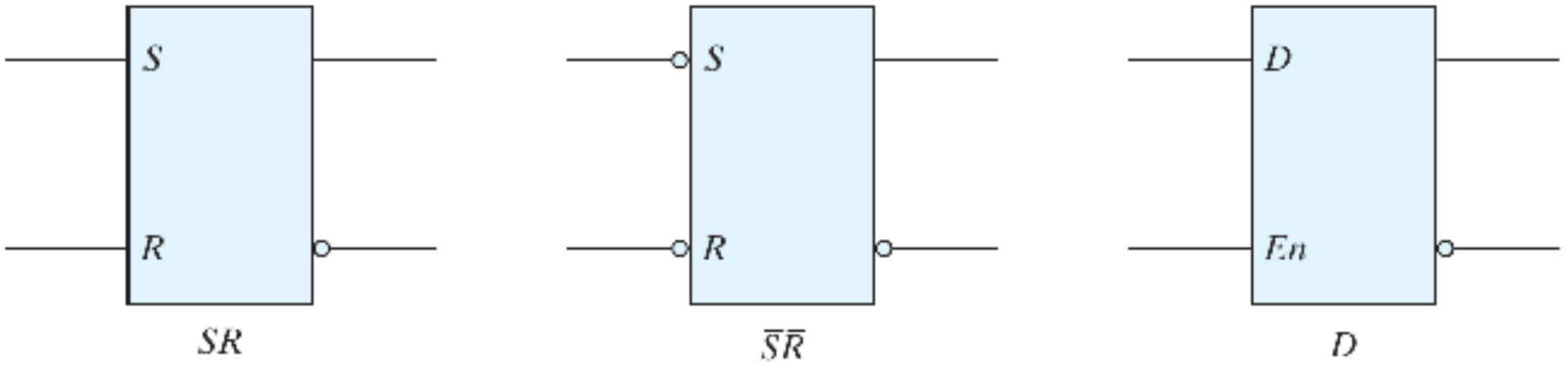




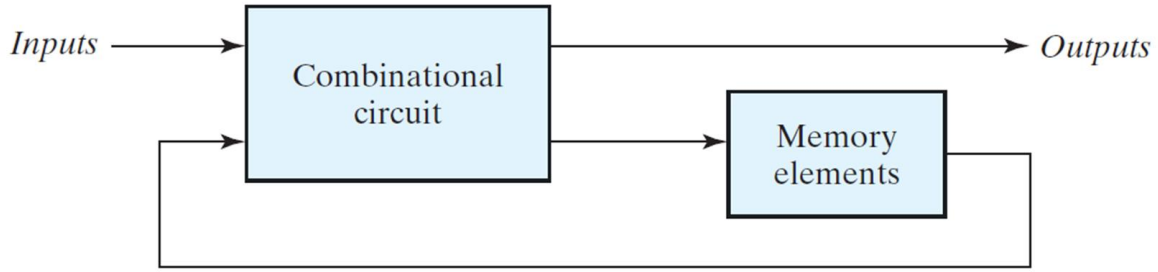
En	*D*	Next state of *Q*
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state

Latches

Graphic symbols for latches

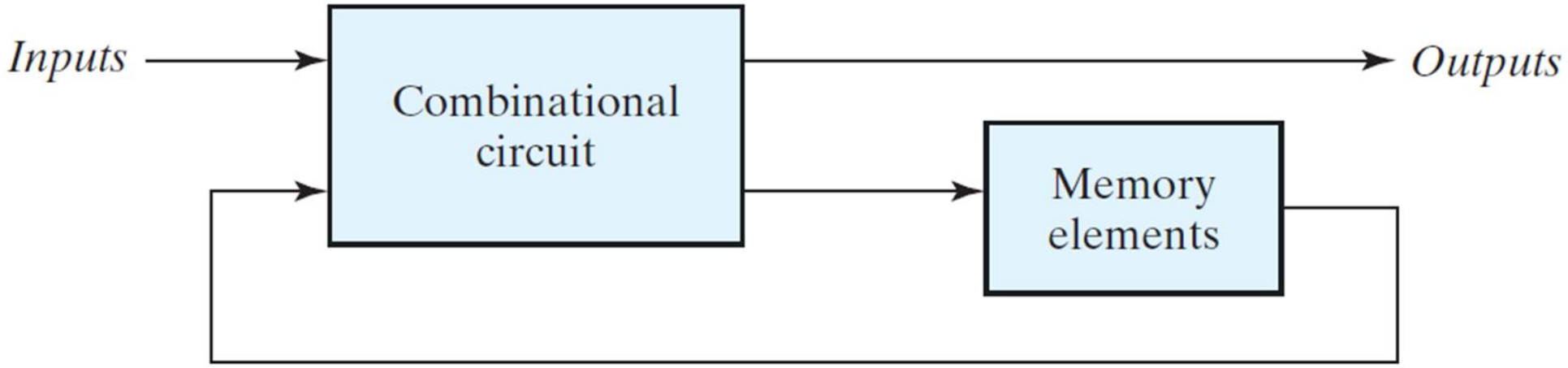


Problem with latches



- A sequential circuit has a feedback path from the outputs of the flip-flops to the input of the combinational circuit
- Consequently, the inputs of the latches are derived in part from the outputs of the same and other latches
- The state transitions of the latches start as soon as the enable/data pulse changes to the logic-1 level
- The new state of a latch appears at the output while the pulse is still active
- This output is connected to the inputs of the latches through the combinational circuit
- If the inputs applied to the latches change again while the enable pulse is still at the logic-1 level, the latches will respond to new values and a new output state may occur

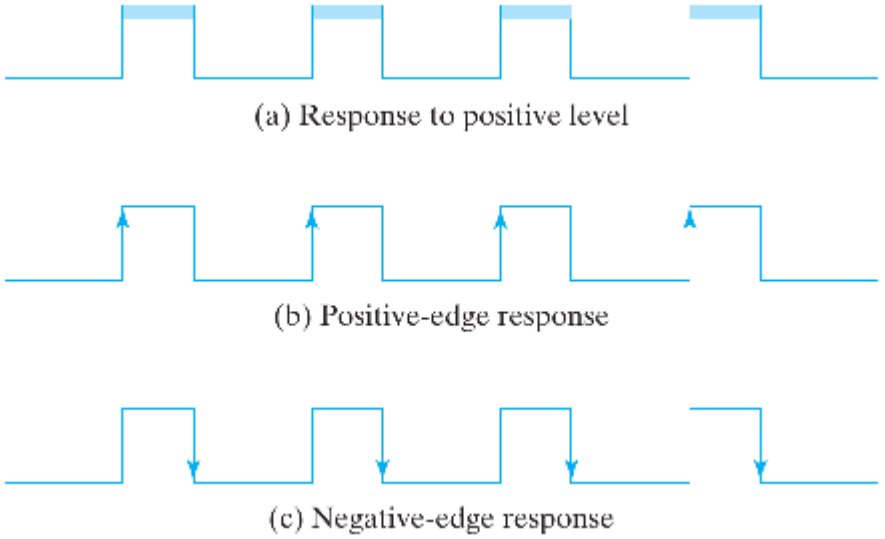
Problem with latches



- If the inputs applied to the latches change again while the enable pulse is still at the logic-1 level, the latches will respond to new values and a new output state may occur
- The result is an unpredictable situation, since the state of the latches may keep changing for as long as the enable pulse stays at the active level
- Because of this unreliable operation, the output of a latch cannot be applied directly or through combinational logic to the input of the same or another latch when all the latches are triggered by a enable signal

Flip flops

- *Flip-flops* are constructed in such a way as to make them operate properly when they are part of a sequential circuit that employs a common clock
- The problem with the latch is that it responds to a change in the *level* of a clock pulse
- A positive level response in the enable input allows changes in the output when the *D* input changes while the clock pulse stays at logic 1
- The key to the proper operation of a flip-flop is to trigger it only during a *signal transition*
- A clock pulse goes through two transitions: positive transition (0 -> 1) and negative transition (1 -> 0)



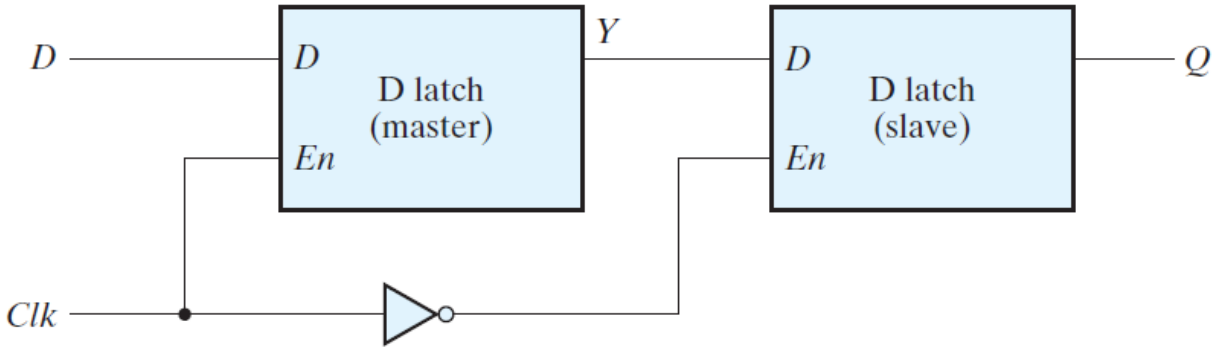
(a) Response to positive level

(b) Positive-edge response

(c) Negative-edge response

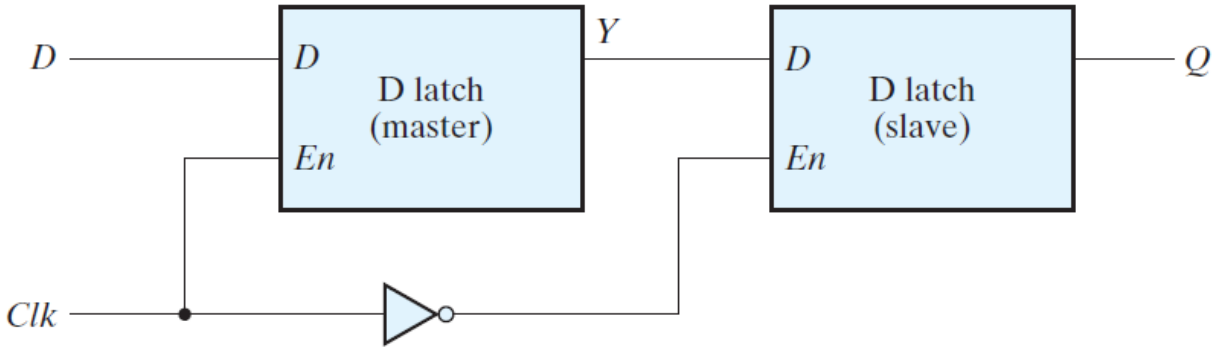
Flip flops

- We can implement flip flops using two latches
- The first latch is called the *master* and the second the *slave*
- The circuit samples the *D* input and changes its output *Q* only at the negative edge of the synchronizing or controlling clock (designated as *Clk*)
  - When the *Clk = 0*, the slave latch is enabled, and its output *Q* is equal to the master output *Y*
  - When the input pulse changes to 1, the data from the external *D* input are transferred to the master

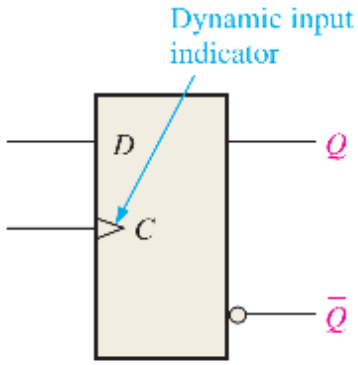


Flip flops

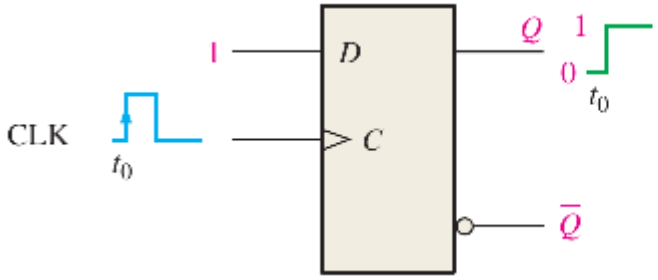
- The slave, however, is disabled as long as the clock remains at 1, because its *enable* input is equal to 0
- Any change in the input changes the master output at *Y*, but cannot affect the slave output
- When the clock pulse returns to 0, the master is disabled and is isolated from the *D* input
- At the same time, the slave is enabled and the value of *Y* is transferred to the output of the flip-flop at *Q*
- Thus, *a change in the output of the flip-flop can be triggered only by and during the transition of the clock from 1 to 0 (-ve edge)*



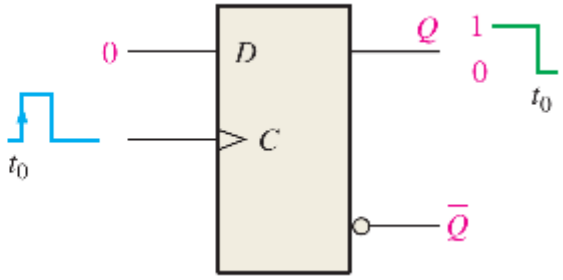
Flip flops



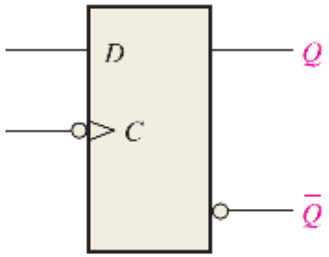
positive edge-triggered flip-flop



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



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



negative edge-triggered flip-flop

The dynamic input indicator means the flip-flop changes state only on the edge of a clock pulse.

Flip flops

Determine the Q and $\overline{Q}$ output waveforms of the positive edge-triggered D flip-flop. Assume the flip-flop is initially RESET.