Latches and Flip-Flops

- So far we have only considered "*combinational circuits*". Recall that in combinational circuits the "*output*" depends only on the "*current inputs*". These circuits have no memory of past inputs.

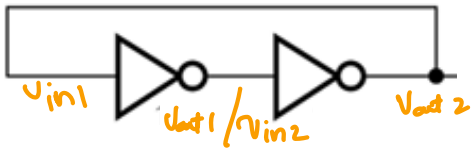
- For example, if you apply "0" and "1" to the input of an AND gate, the output will be "0". If you then change the inputs to "1" and "1", the output will be "1". This is obtained independent of what the input/output were in the previous case. So we say, the output depends only on the current input values.

- In "*sequential circuits*" however, the "*output*" depends on the "*current and past inputs*". An example of a sequential circuit is a counter. At a given time, the output of a counter is dependent on its past input.

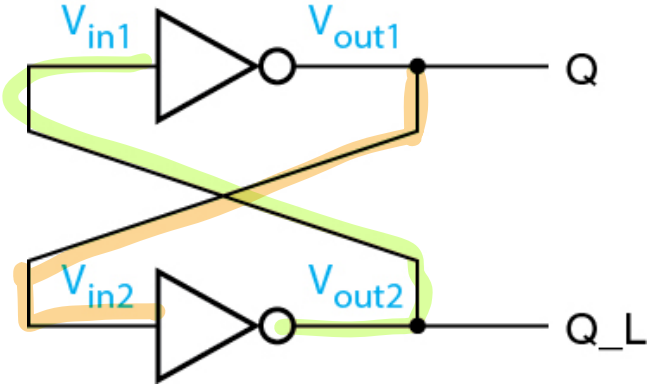
- Sequential circuits require blocks to store information of past states.

Bistable Elements

- The simplest sequential circuit consists of a pair of inverters forming a loop.
- This circuit has no inputs.

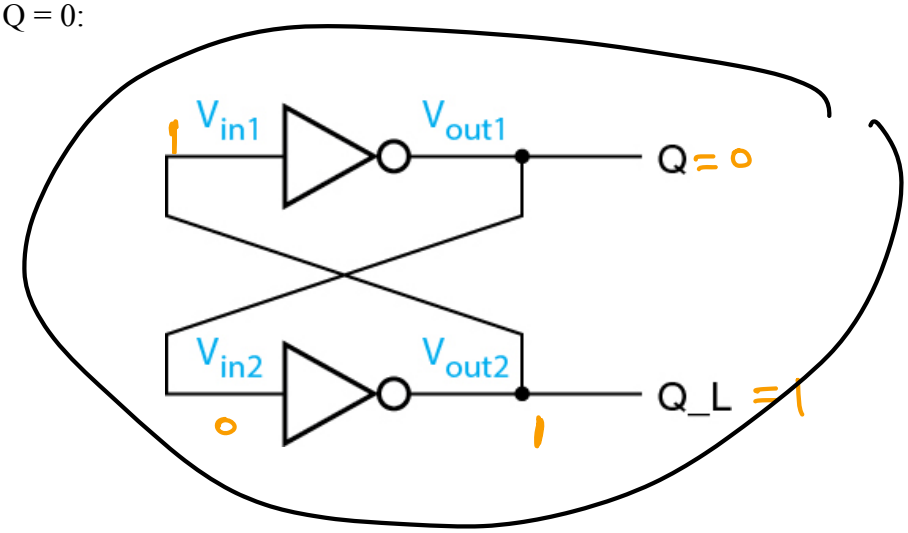


- This circuit can also be represented as below (twisted representation):

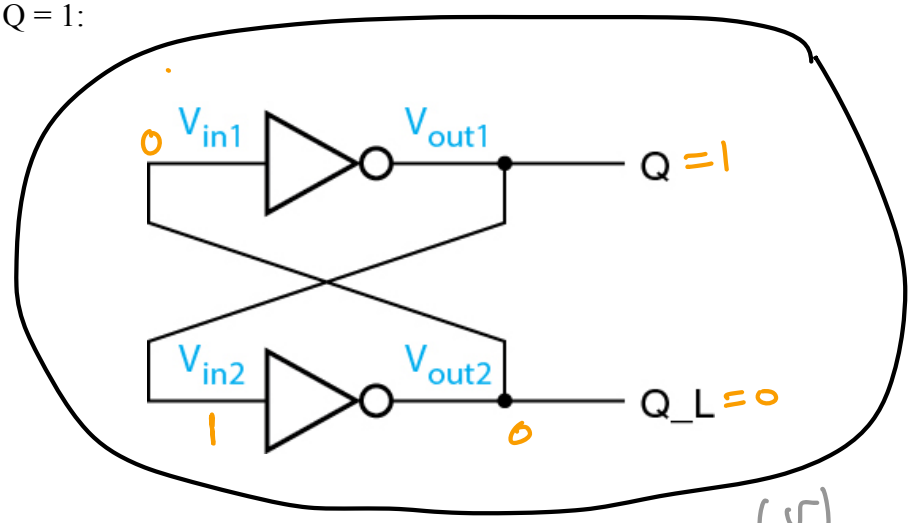


- The circuit has two states. These states can be represented by one variable, say “Q”.

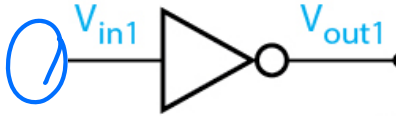
Assume Q = 0:

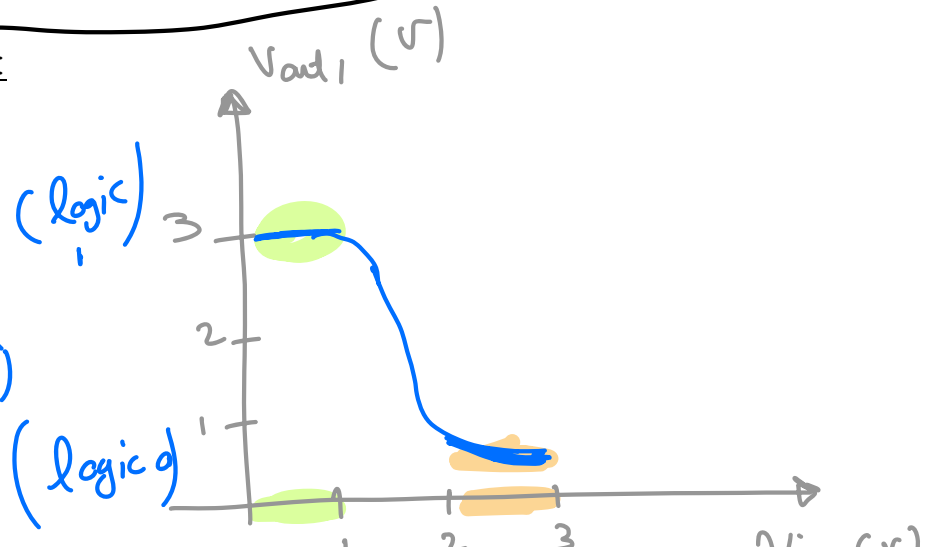


Assume Q = 1:



Analysis of the Bistable Element:



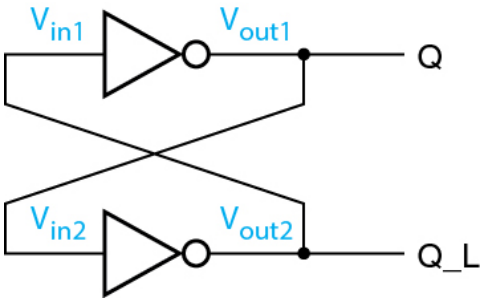


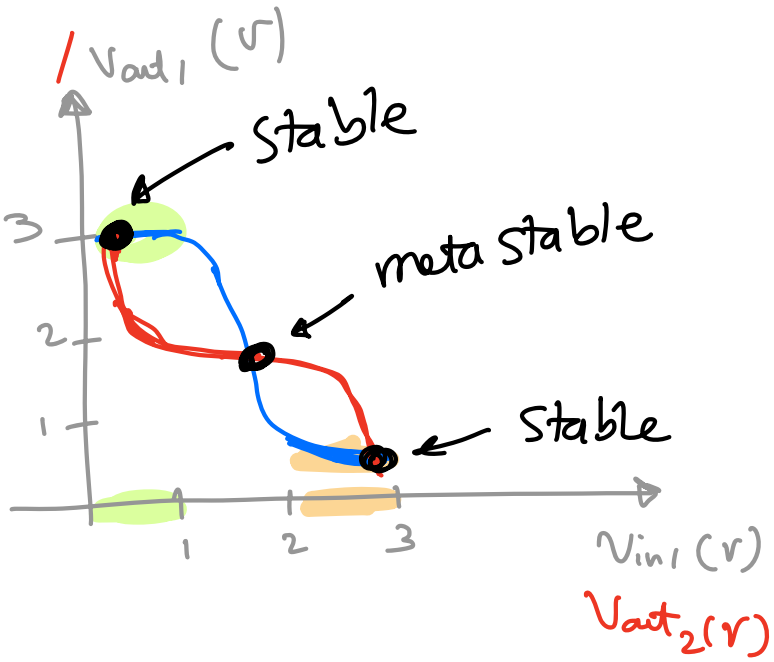
$V_{in1} < 1\,V \Rightarrow V_{out1} \simeq 3V$ (logic "1")

logic "0"

$V_{in1} > 2V \Rightarrow V_{out1} \simeq 0V$ (logic "0")

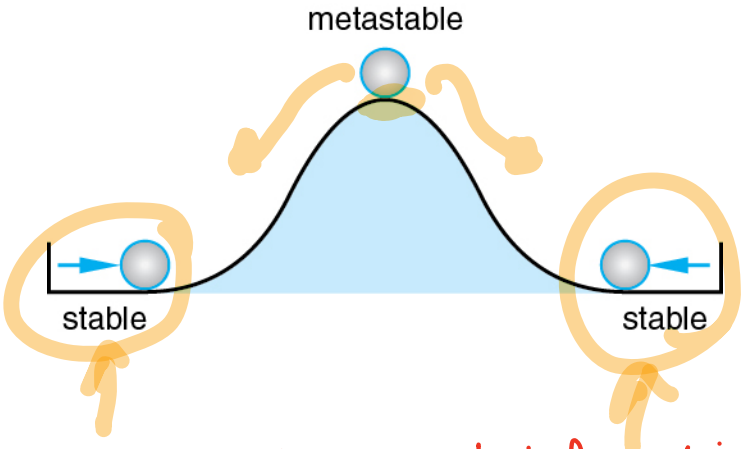
logic "1"





Ball and Hill Analogy:

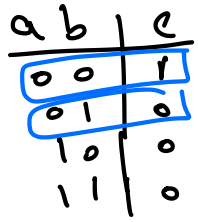
- If the ball sits at the top, it may sit there precariously for a while, until a random force (wind, earthquakes, etc) start it rolling down the hill.



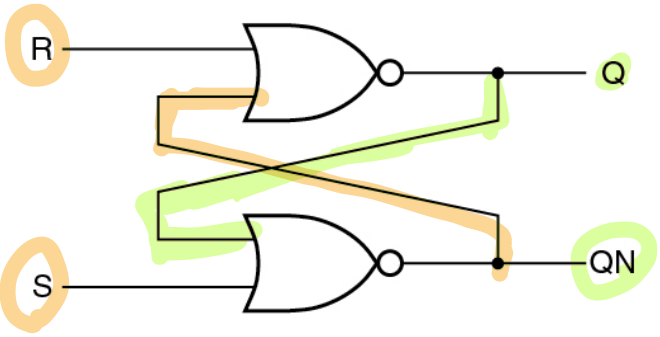
We cannot change the stored information, when the bistable element is powered on.
To be able to change the information, we need extra inputs.

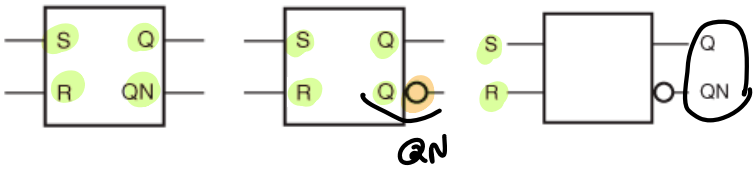
Latches

- Latches are the most basic storage elements.
- Latches do not need a clock signal. The output of the latch can change at anytime (based on the input).
- Latches are referred to as level-sensitive storage elements.
- Examples of latches are : S-R latch, $\overline{S} - \overline{R}$ latch, D latch.



S-R (Set-Reset) Latch





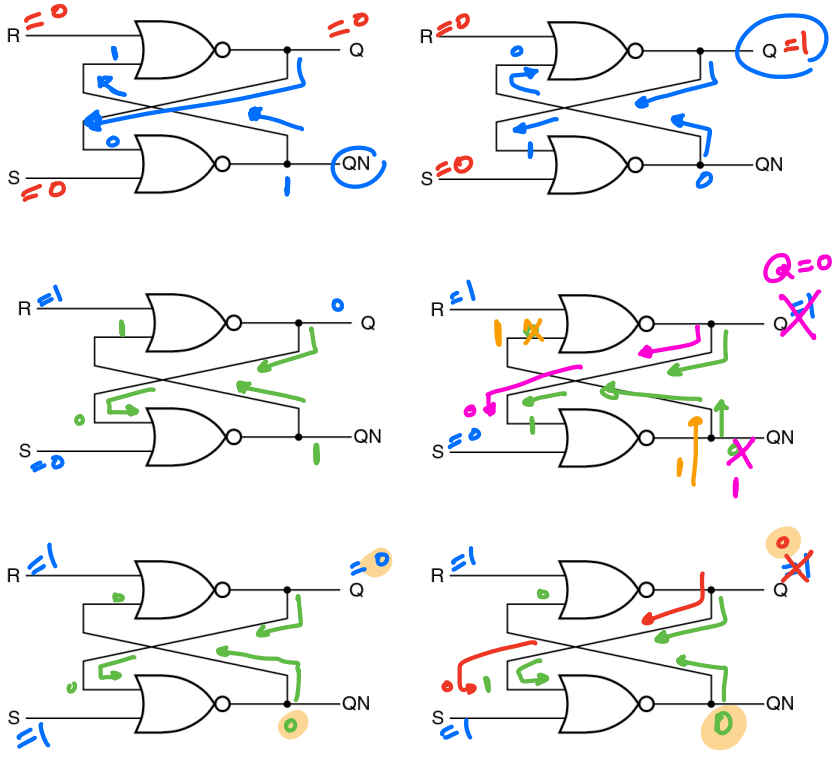
S	R	Q(t)	Q(t+ε)	QN(t+ε)
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0	0
1	1	1	0	0

HOLD

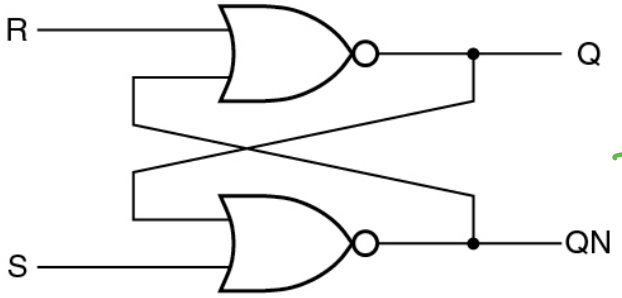
RESET

SET

forbidden

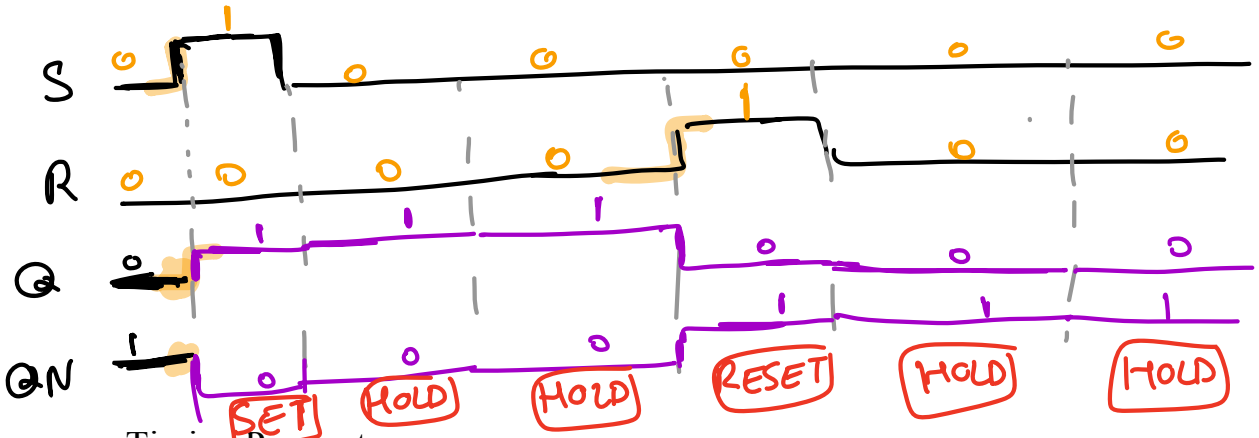


In summary:

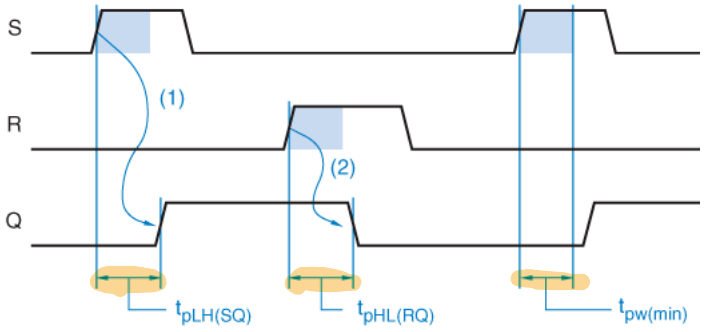


	S	R	Q	QN
HOLD	0	0	last Q	last QN
RESET	0	1	0	1
SET	1	0	1	0
Forbidden	1	1	0	0

Timing Diagram:



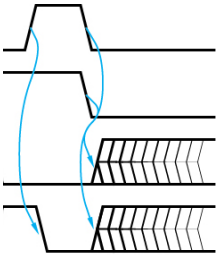
Timing Parameters:



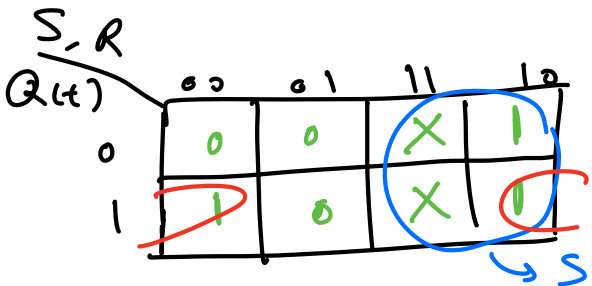
$t_{pw(min)}$: minimum pulse width required so the latch does not go to the metastable state.

$t_{pLH(SQ)}$: propoagation delay for Q to change from low to high with respect to S changing from the low to high.

$t_{pHL(RQ)}$: propoagation delay for Q to change from high to low with respect to R changing from the low to high.



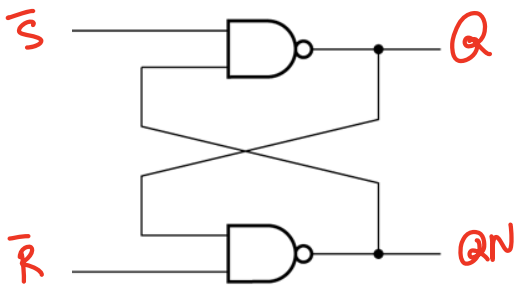
t_{rec} (recovery time): the latch may go to metastable state if both inputs S and R are negated simoultaneously. t_{rec} is the minimmum time needed between negating the inputs S and R so they are not considered as simultaneous.

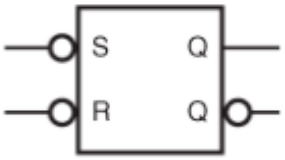


$Q(t+\epsilon) = S + Q(t) \cdot R'$

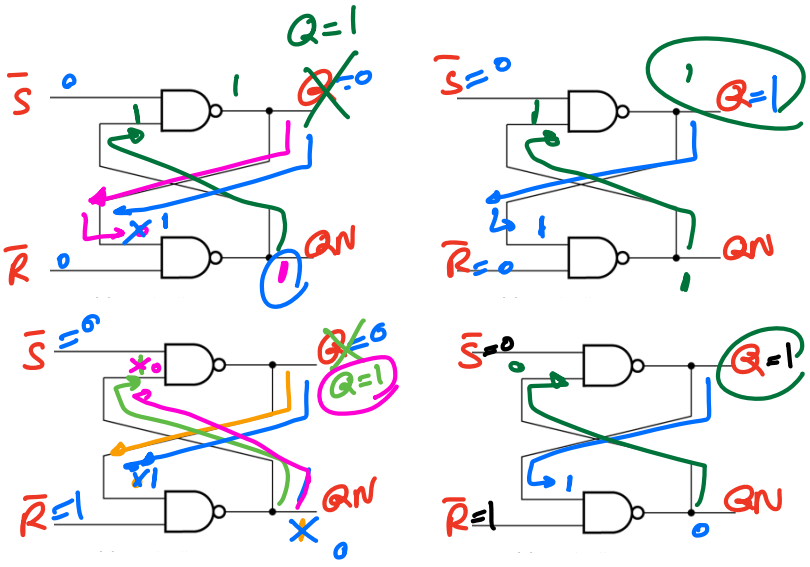
Characteristic equation

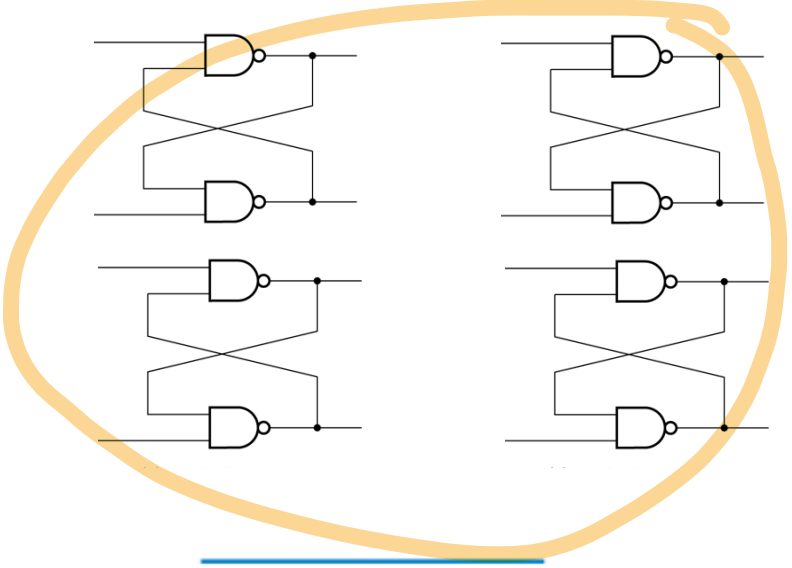
$\bar{S}$ – $\bar{R}$ Latch



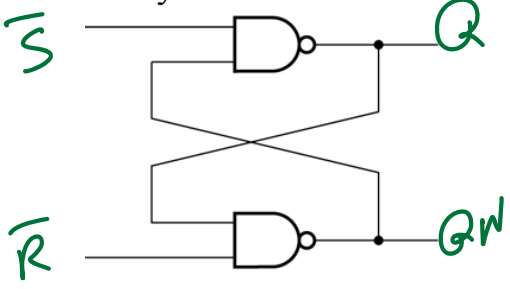


$\bar{S}$	$\bar{R}$	Q(t)	Q(t+ε)	QN(t+ε)	
0	0	0	1	1	Forbidden
0	0	1	1	1	
0	1	0	1	0	SET
0	1	1	1	0	
1	0	0	0	1	RESET
1	0	1	0	1	
1	1	0	0	1	HOLD
1	1	1	1	0	



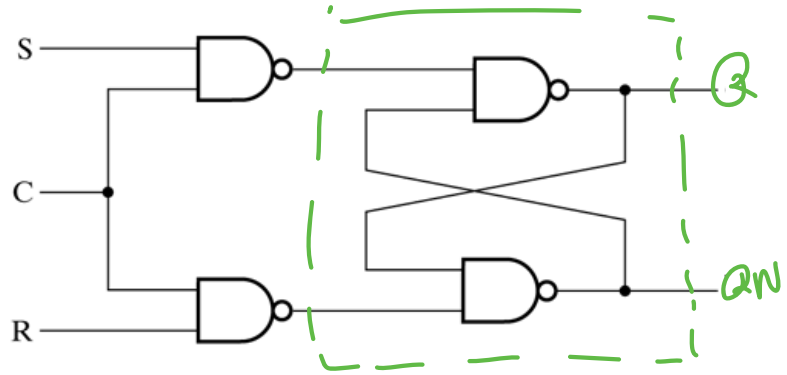


In summary:



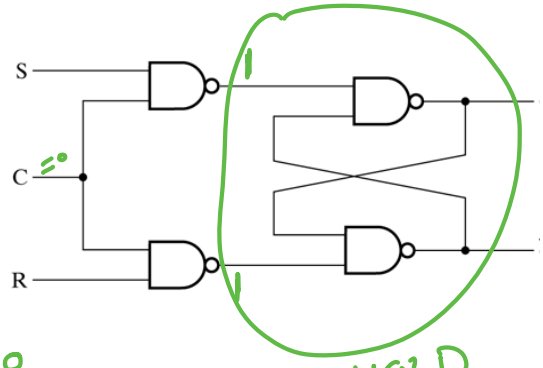
S_L	R_L	Q	QN	
0	0	1	1	Forbidden
0	1	1	0	SET
1	0	0	1	RESET
1	1	last Q	last QN	HOLD

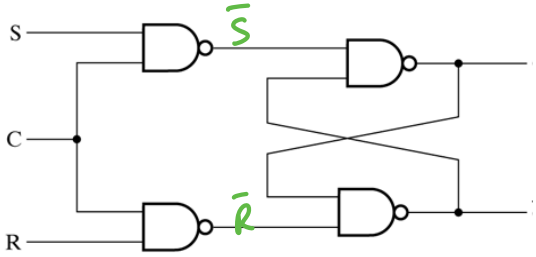
S-R Latch with Additional Control Input (Enable)



$\bar{S}$-$\bar{R}$ Latch

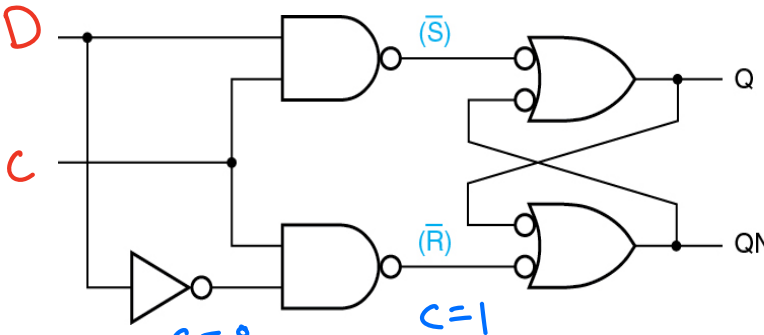
S	R	C	Q	QN
X	X	0	last Q	last QN
0	0	1	last Q	last QN
0	1	1	0	1
1	0	1	1	0
1	1	1	1	1





D (Data) Latch

- Has two inputs: D for input data, and C for enable.
- The forbidden state does ~~not~~ occur in D latches.
- As long as C=1, the latch will change according to the value of input D.

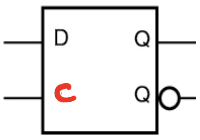


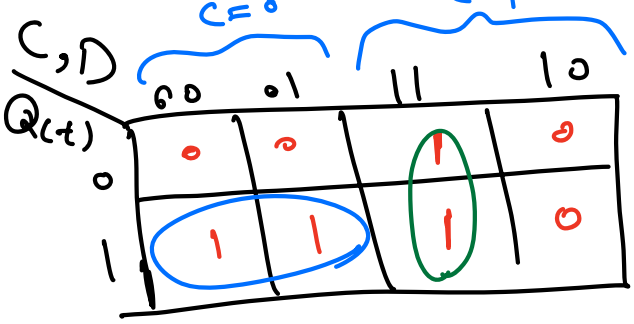
C	D	Q	QN
1	0	0	1
1	1	1	0
0	x	last Q	last QN

Follows

Holds

If C=1, Q follows D
If C=0, Q HOLDS last value

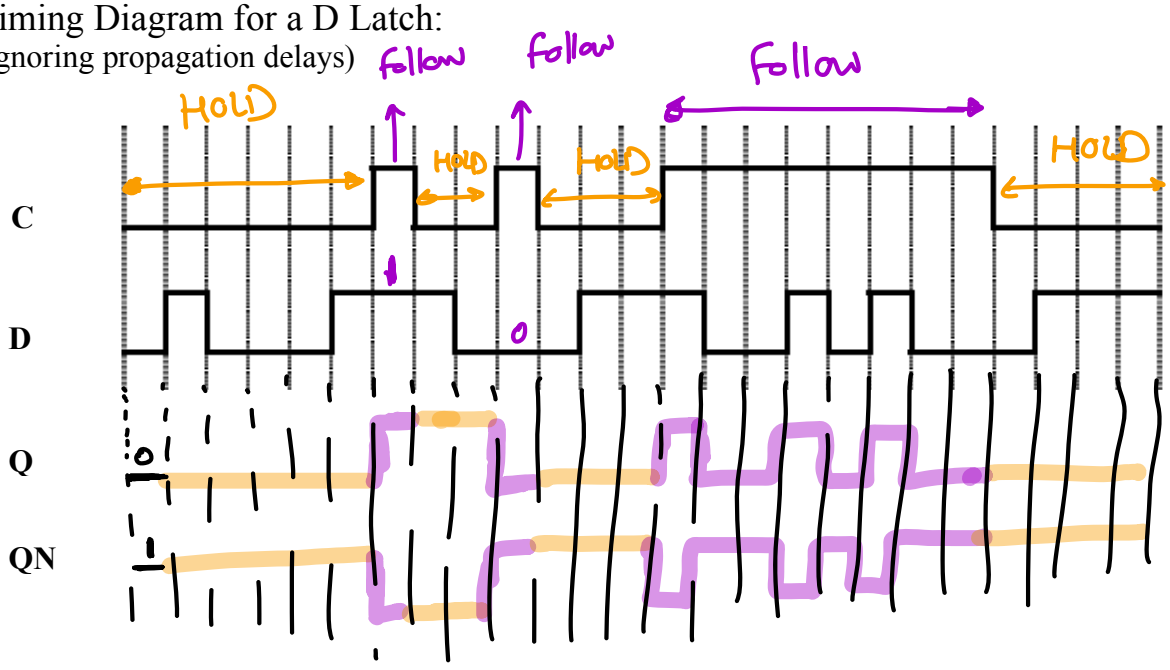




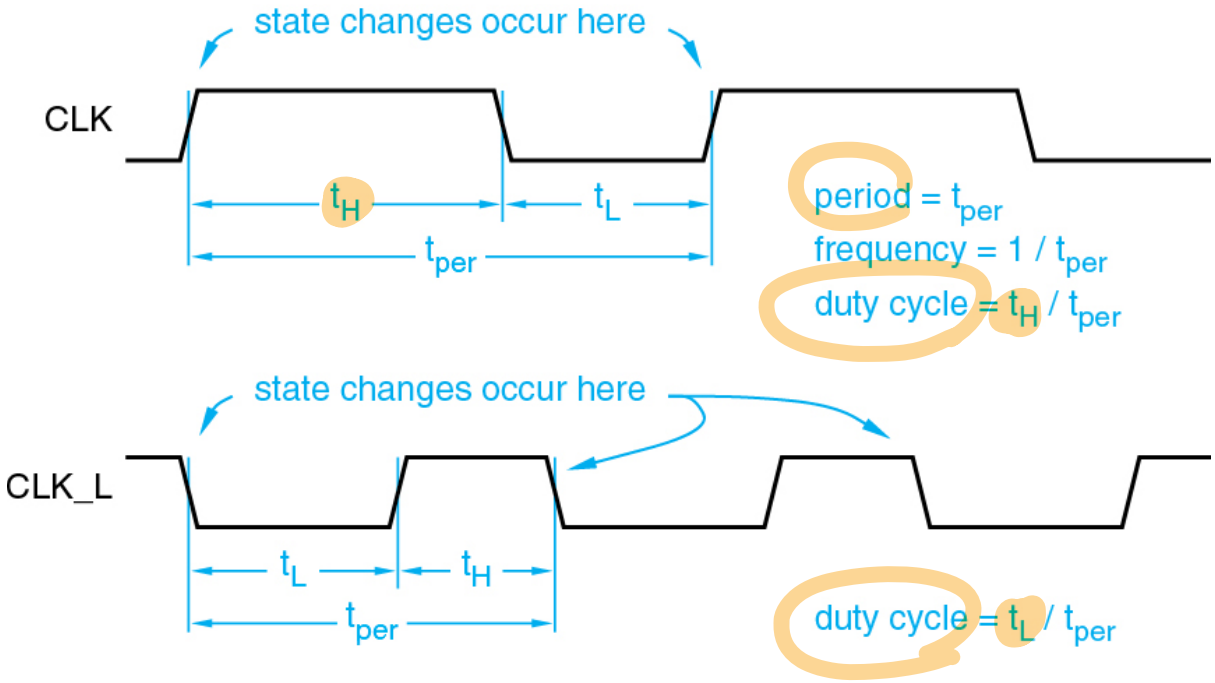
$$Q(t+\varepsilon) = C.D + Q(t).C'$$

Characteristic equation for D latch

Timing Diagram for a D Latch:
(ignoring propagation delays)



Clock Signals

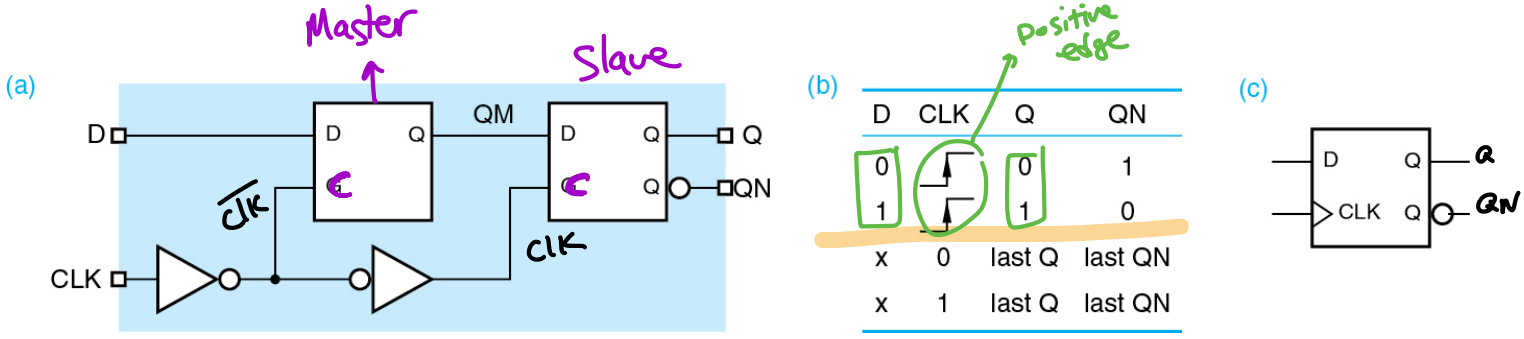


Flip Flops

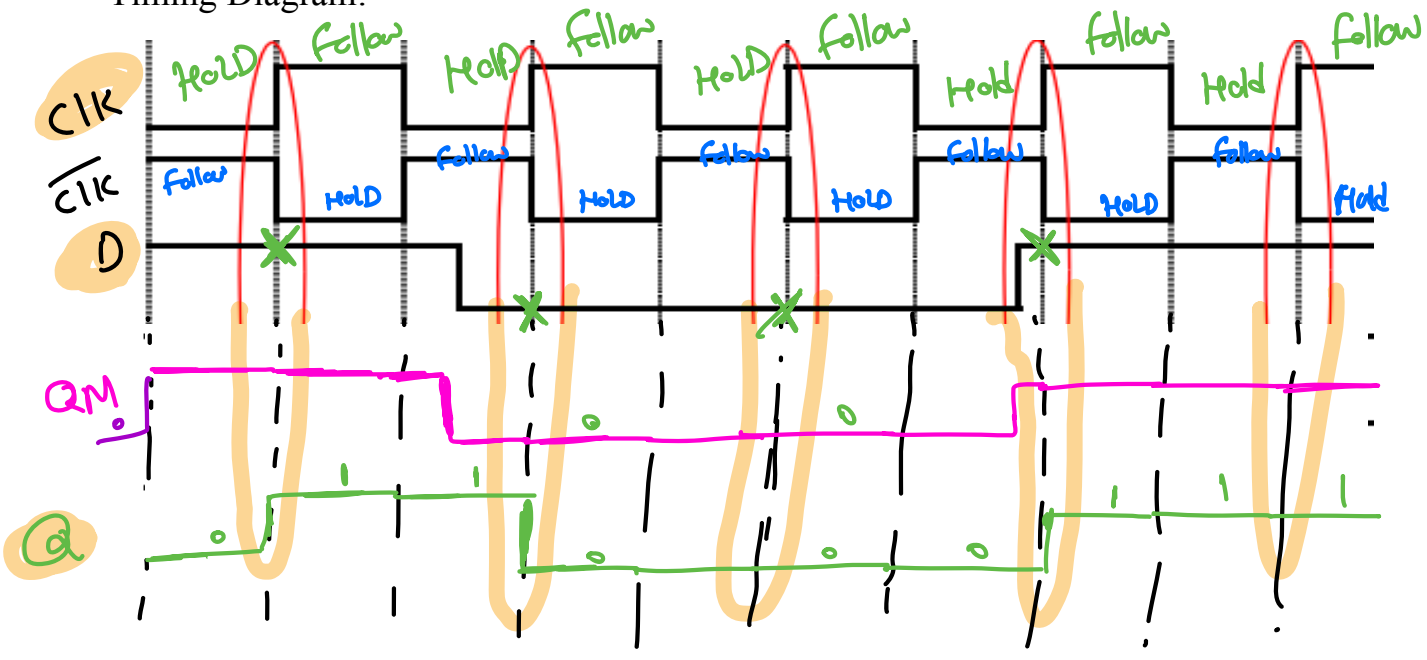
- A flip-flop is a single bit storage unit that works with clock signals.
- Edge-triggered flip-flops can change state only on the edge (either rising or falling) of a clock signal.
- Positive-edge-triggered flip-flops can change state only on the rising edge of a clock signal.
- Negative-edge-triggered flip-flops can change state only on the falling edge of a clock signal.
- Examples of flip-flops are: D flip-flop, T flip-flop, and J-K flip-flop

Positive-Edge-Triggered D Flip-Flop

- Consists of cascading two D latches.
- The first latch (closer to the input) is called the Master latch. The second latch (closer to the output) is called the Slave latch.
- Clock signal is applied to the control input of the D latches. The Master latch receives the inverted version of the clock signal, and the Slave latch receives the clock signal.



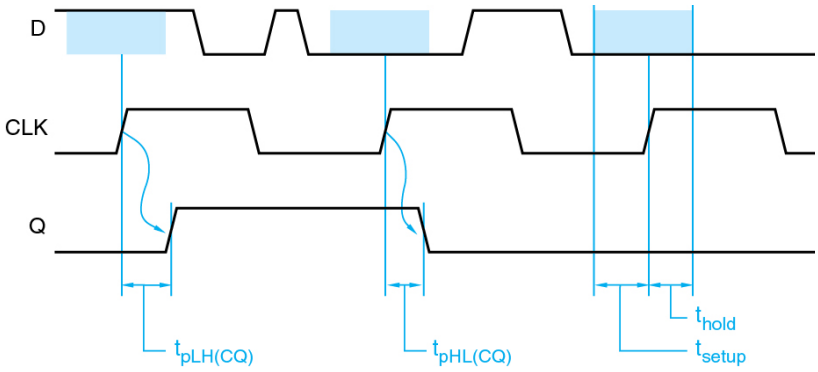
Timing Diagram:



Timing Parameters:

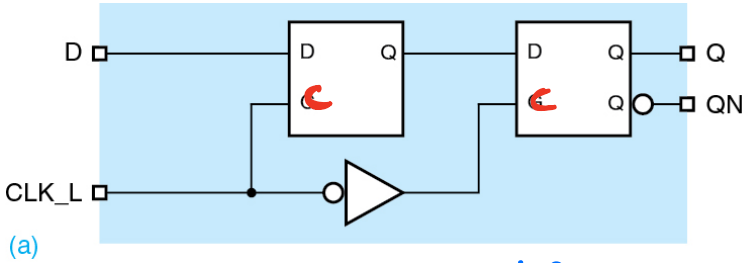
t_{setup} (setup time): minimum time that the input signal must be present prior to a clock edge.

t_{hold} (hold time): minimum time that the input signal mush be kept after the clock edge.



Negative-Edge-Triggered D Flip-Flop

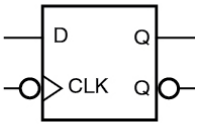
- Consists of cascading two D latches.
- The Master latch receives the clock signal, and the Slave latch receives the inverted version of the clock signal.



(a)

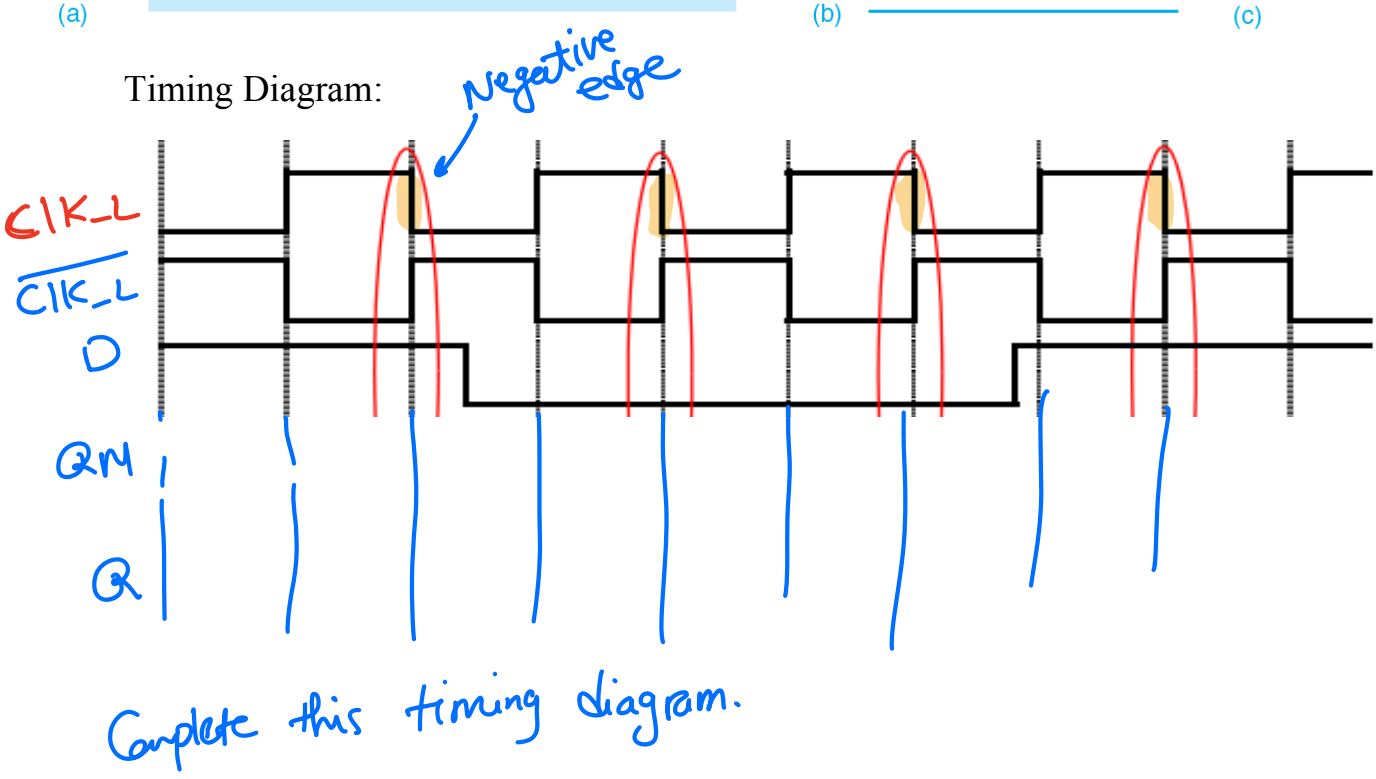
D	CLK_L	Q	QN
0	↓	0	1
1	↓	1	0
x	0	last Q	last QN
x	1	last Q	last QN

(b)



(c)

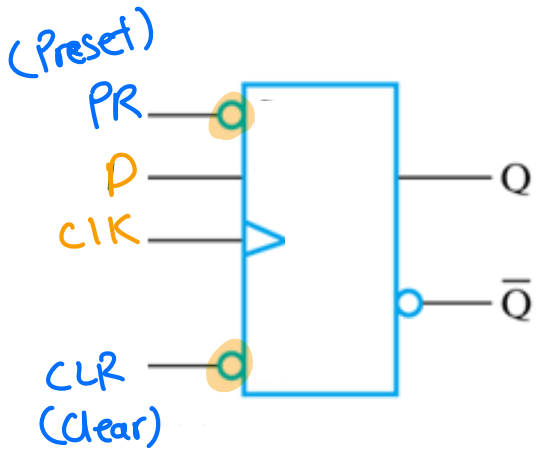
Timing Diagram:



Complete this timing diagram.

Edge-Triggered D Flip-Flop with Asynchronous Inputs

- Asynchronous inputs may be used to force the flip-flop to a particular state "*independent*" of the CLK and D inputs.
- These inputs, typically labeled PR (*preset*) and CLR (*clear*), behave like the set and reset inputs on an S-R latch.



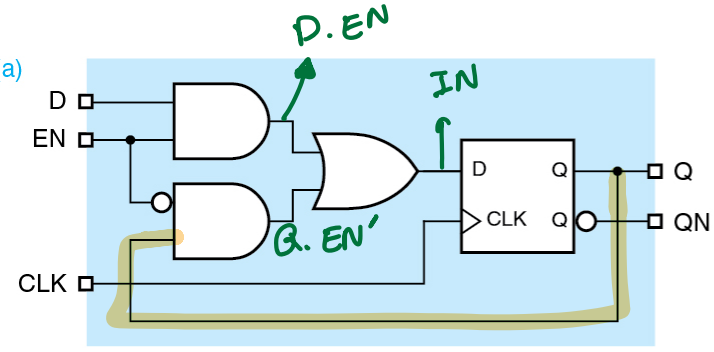
PR	CLR	CLK	D	Q	$\overline{Q}$
0	1	X	X	1	0
1	0	X	X	0	1
0	0	X	X	Undefined	
1	1	↑	0	0	1
1	1	↑	1	1	0

at the positive edge of the clock.

Edge-Triggered D Flip-Flop with Enable Inputs

- Adding tan "Enable" input makes it possible to hold the last value stored, rather than load a new value, at the clock edge.

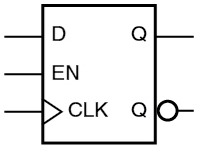
(a)



(b)

D	EN	CLK	Q	QN
0	1	↑	0	1
1	1	↑	1	0
x	0	↑	last Q	last QN
x	x	0	last Q	last QN
x	x	1	last Q	last QN

(c)



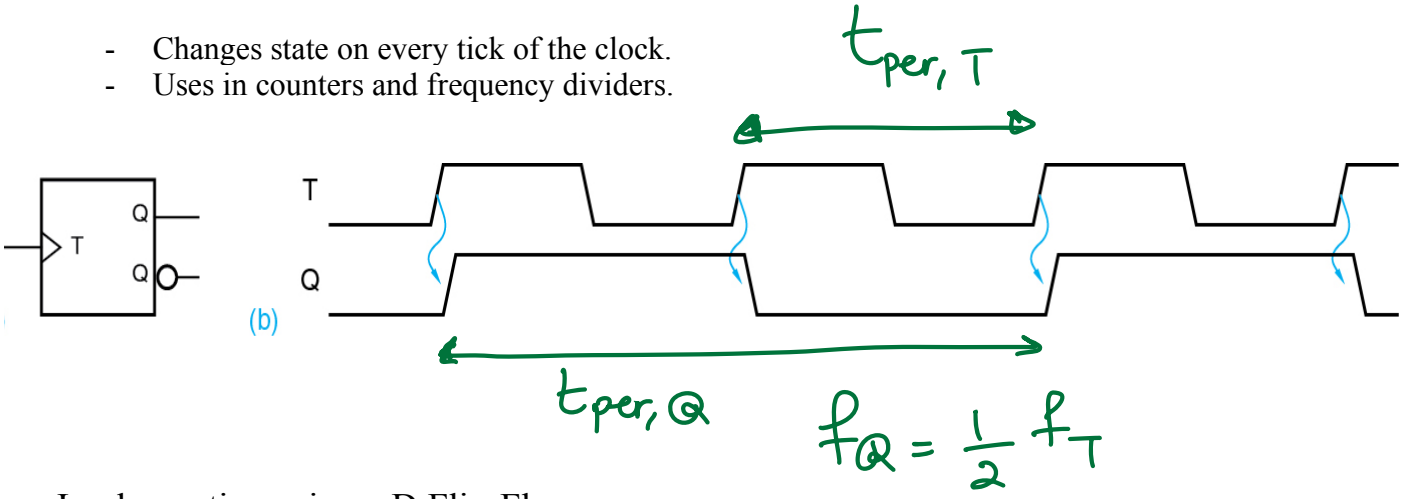
$IN = [D.EN] + [Q.EN']$

When EN=1, IN=D (Normal operation of D FF)

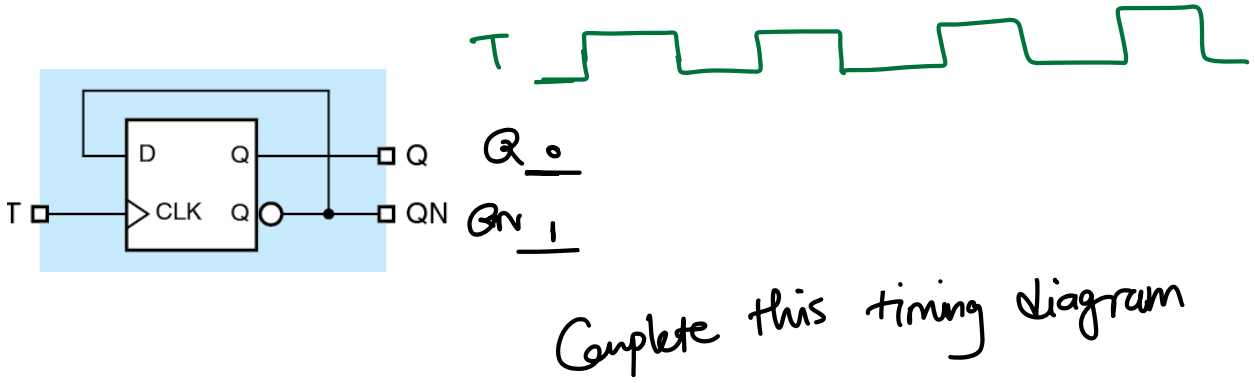
When EN=0, IN=Q (HOLDS the last value)

T (Toggle) Flip-Flop

- Changes state on every tick of the clock.
- Uses in counters and frequency dividers.

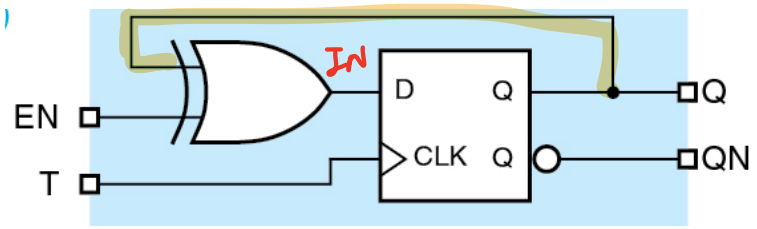


Implementing using a D Flip-Flop:

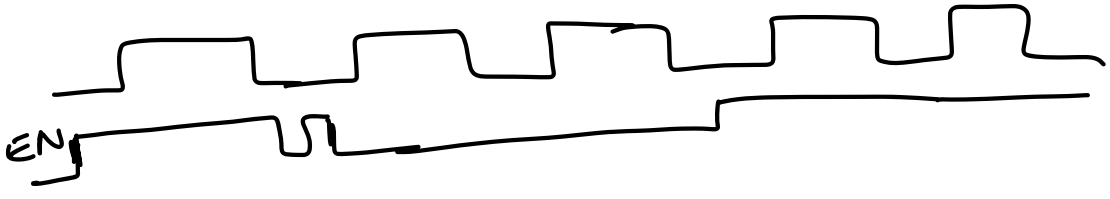


T Flip-Flop with Enable Input

- The flip-flop changes state at the triggering edge of the clock only if the EN signal is asserted.



$IN = EN \oplus Q \Rightarrow$ If EN=0, IN=Q; If EN=1, IN=QN



Q 0