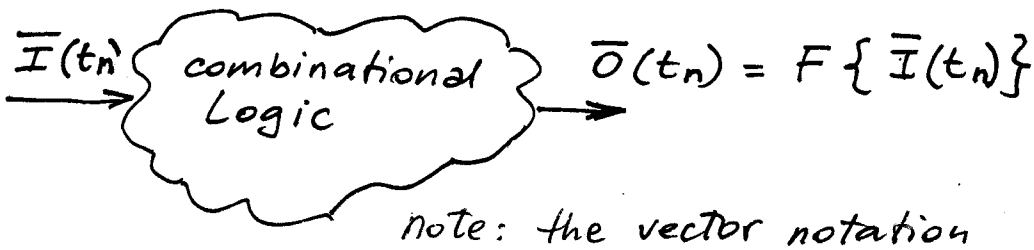
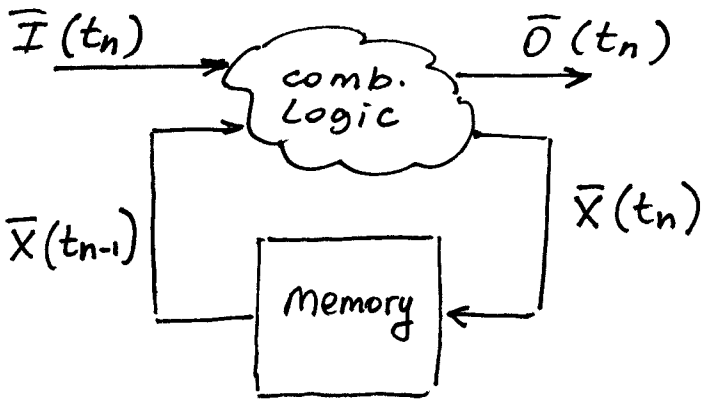
CMOS sequential logic

So far we have seen only circuits where the outputs at a given time are function of the inputs applied at that time (we have no "memory" of what happened in the past). Those kind of circuits are called combinationel.



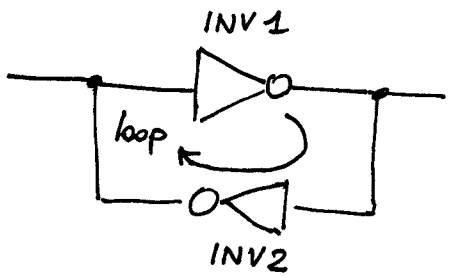
note: the vector notation is just to say that we can have several inputs and outputs (the # of inputs and the # of outputs can be different)

Our goal is now to build a more general class of circuits where the outputs at a given time are function not only of the input variables applied at that time, but as well function of the values taken at some time in the past by certain variables. Those kind of circuits are called sequential.



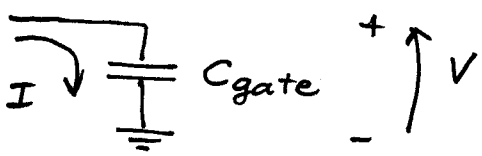
those variables are usually called state variables $\bar{X}$

In order to build a circuit able to store previously applied input variables we need to establish a feedback between the output and the input. Feeding back (combinational loop) the output into the input, we keep "refreshing" the value, so basically we build a mechanism to "remember".



basic "memory" element

through the loop we keep re-generating whatever value is stored on the gate capacitance of INV1



$$I(t) = C_{gate} \cdot \frac{dV(t)}{dt}$$

$$I(t_n) = C_{gate} \cdot \frac{V(t_n) - V(t_{n-1})}{t_n - t_{n-1}}$$

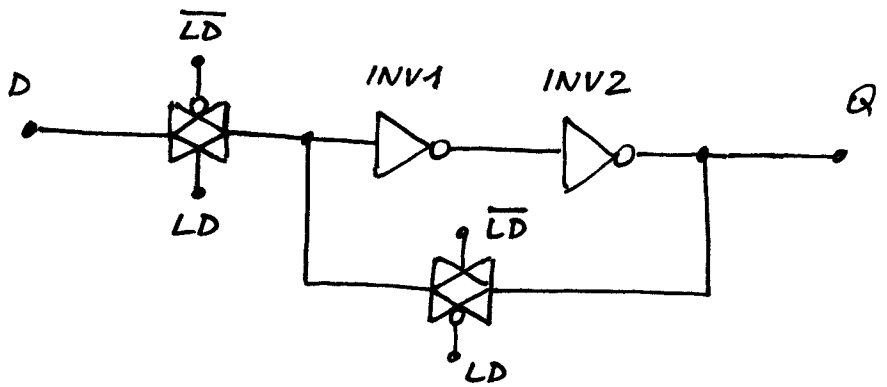
($V(t_n)$: present; $V(t_{n-1})$: past)

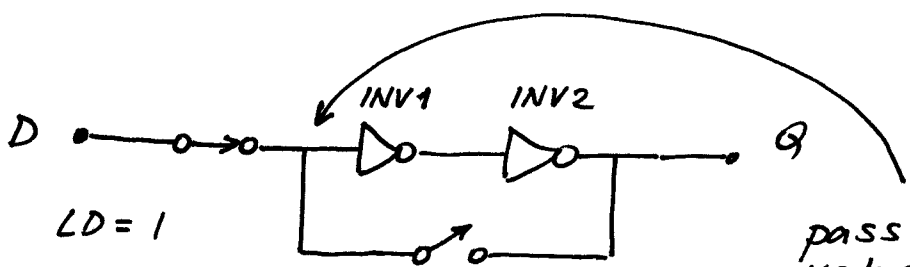
Combining Transmission gates and combinational gates, it is possible to build latches and flip-flops.

latch = level sensitive storage element

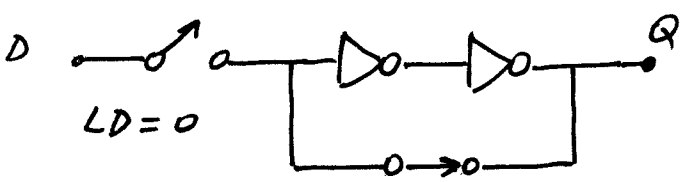
flip-flop = edge-triggered storage element

Data Latch (D-Latch)

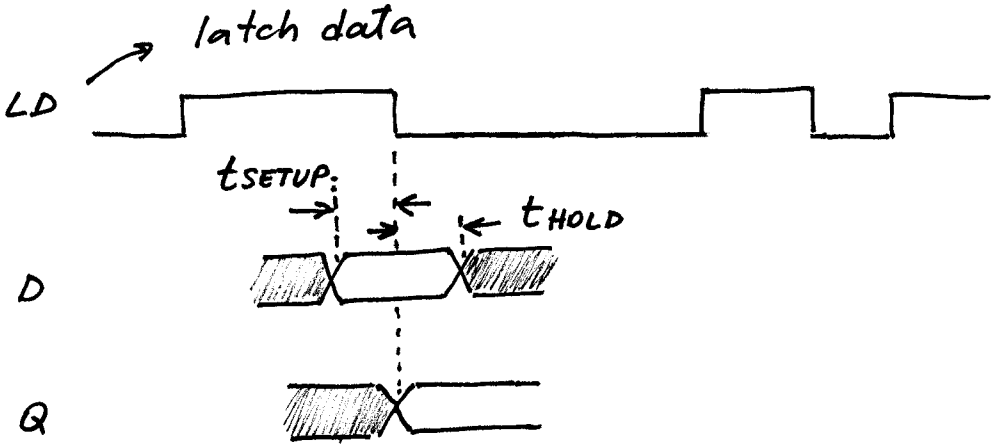




pass the value D that we want to store, on the gate capacitance of INV1

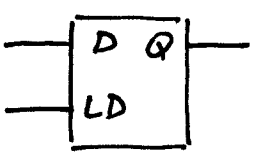


Keep regenerating the value stored on Cgate of INV1.



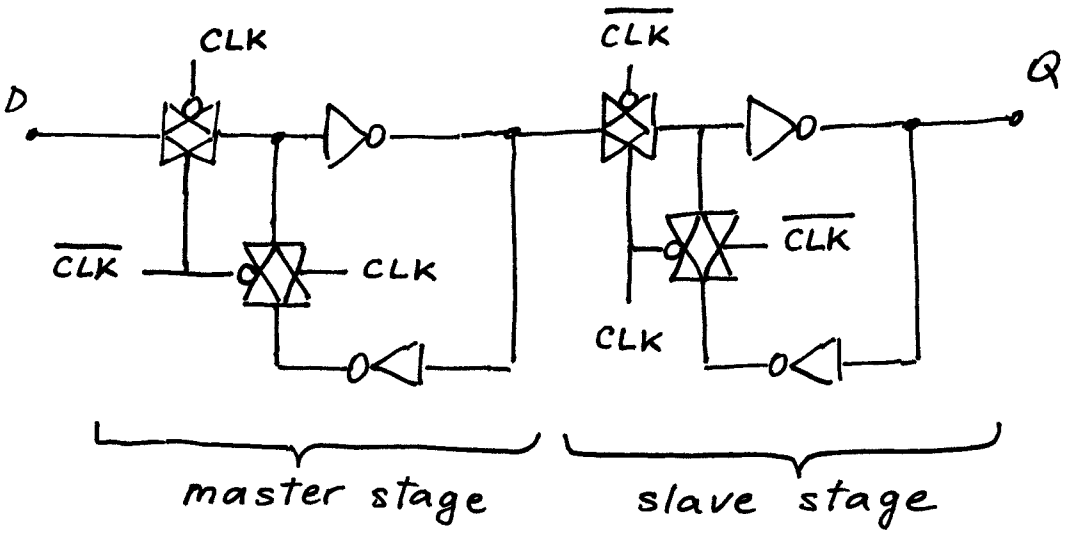
The data D must be stable a little (t_{SETUP}) before the negative edge of LD so that the input switch has the time to close and the loop switch to open.

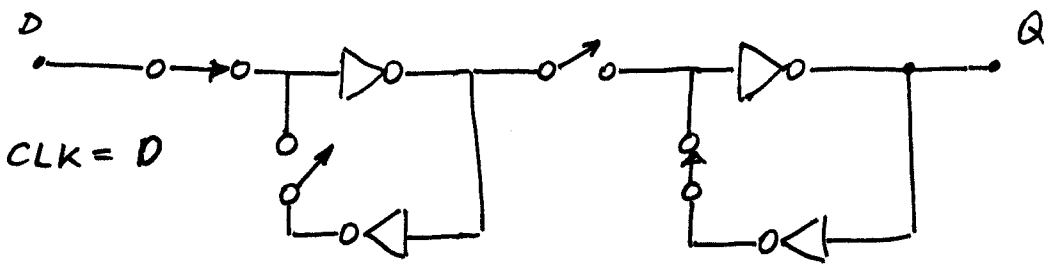
Furthermore D must be stable a little after (t_{HOLD}) the negative edge of LD So that the input switch has the time to open and the loop switch to close.

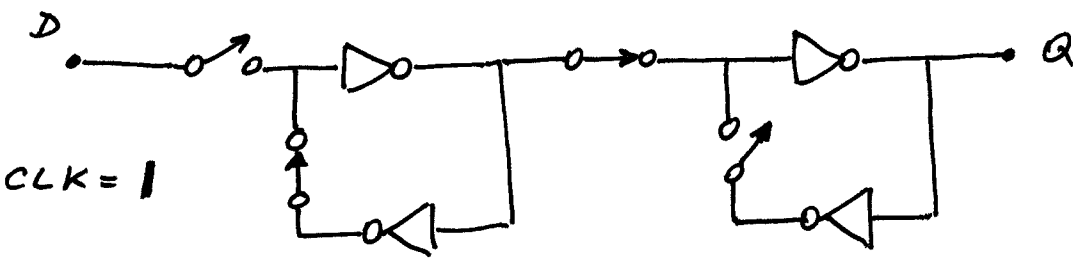


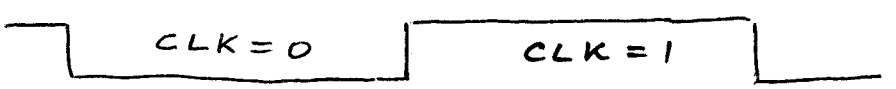
D-latch symbol

D - Flip Flop


CLK
$\overline{CLK}$
D
Q
$\overline{CLK}$
CLK
$\overline{CLK}$
CLK
master stage
slave stage


D
Q
CLK = D


D
Q
CLK = 1


CLK = 0
CLK = 1

While CLK=0 the master stage receive the D input, while the slave stage holds the previous value.
At the transition CLK = 0 → 1, the master captures in the value, and starts to transfer it to the output Q (through the slave stage).
While CLK = 1 the master holds the value captured at the transition 0→1 and the slave transfers it to the output.
When CLK transitions from 1 to 0 the slave captures the value previously held by the master, and the master starts to take in the input D.

In short the D-flip flop samples its input D at the positive edge (rising edge 0→1) of the clock CLK