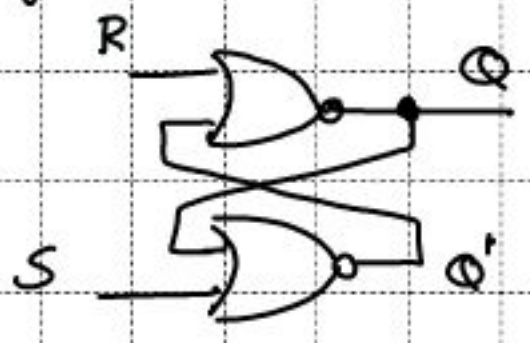
Basic sequential building blocs : 1-bit memory cell

How to infer a D-FF from logic gates?

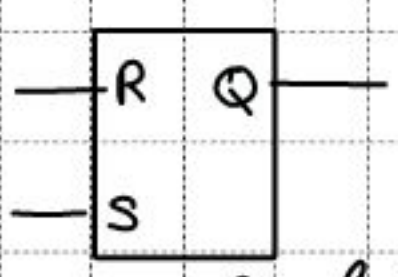
① The first idea is to use a basic RS-latch



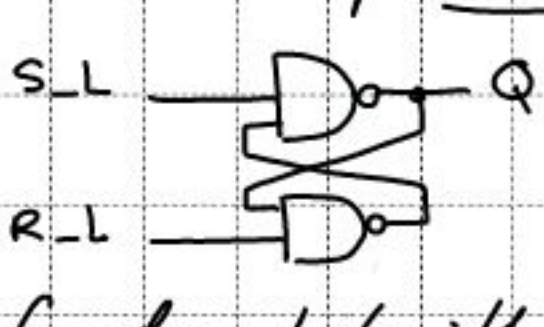
R	S	Q^+ ← next value (future)
0	0	Q current value (present)
0	1	1 Set
1	0	0 Reset
1	1	X (Not defined)

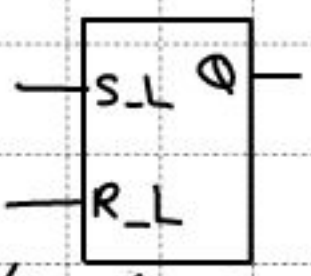
↗ This is the structure that has memory to store a bit.

R → to reset the output → '0'
S → to set the output → '1'

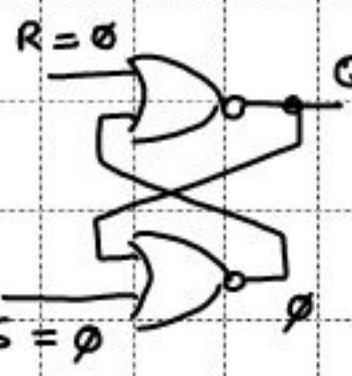


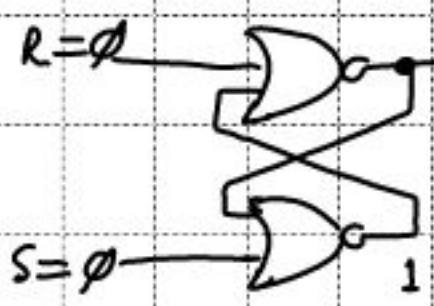
Like the standard chip 74LS279



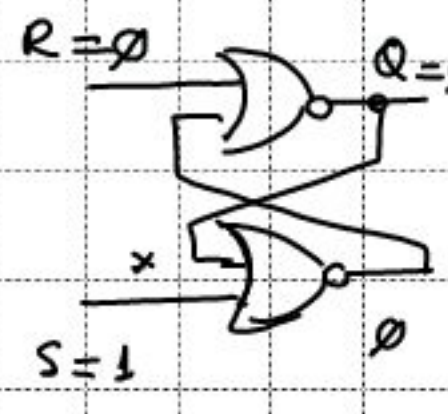


The same function, but with active-low inputs

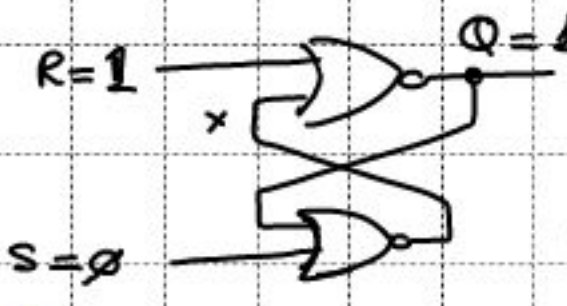




Do nothing
$Q^+ = Q$
keep the Q value forever
"memory"

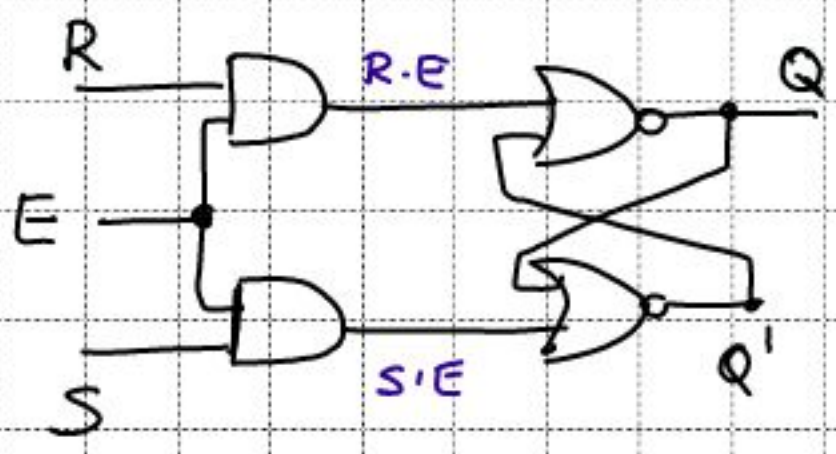


Set $Q^+ \rightarrow$ '1'
write a '1'



Reset $Q^+ \rightarrow 0$
write a '0'

(2) The RS latch with enable input E (gated latch)

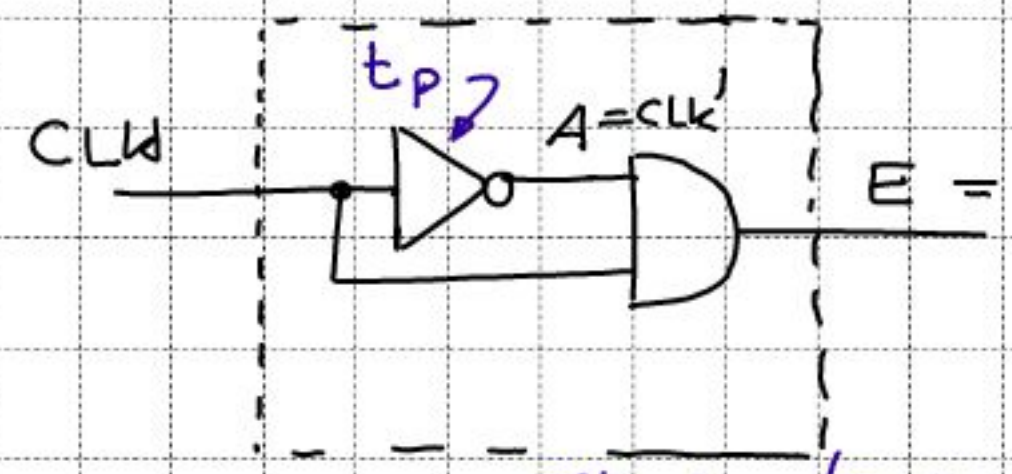


It works in the same way but only when enabled E = 1

E = Ø → disabled and the output saved value cannot be changed.

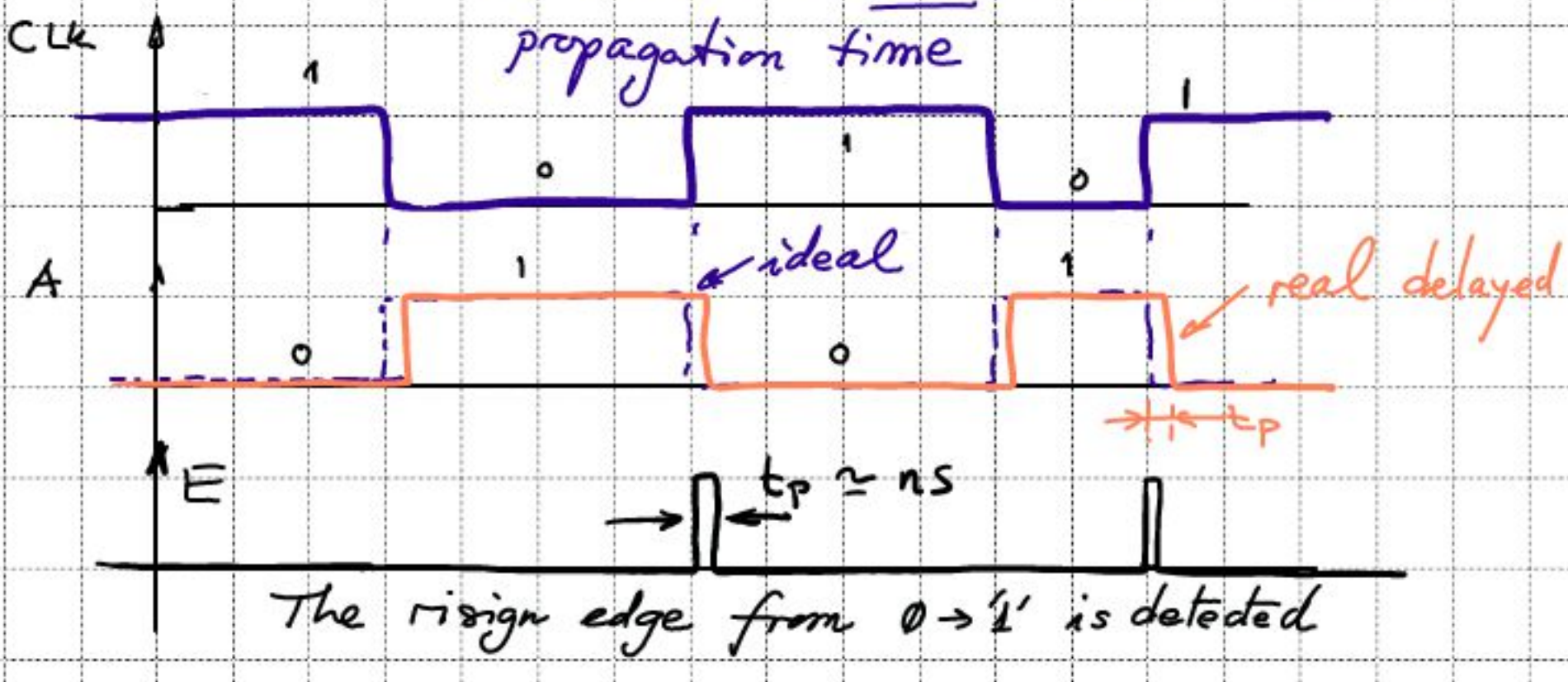
E	R	S	Q^+ (next value after propagating input changes)
Ø	X	X	Q
1	Ø	Ø	Q → do nothing $Q^+ \to Q$
1	Ø	1	1 → set $Q^+ \to$ '1'
1	1	Ø	Ø → Reset $Q^+ \to$ 'Ø'
1	1	1	X (Not defined)

(3) Can we invent an edge (transition detecter)

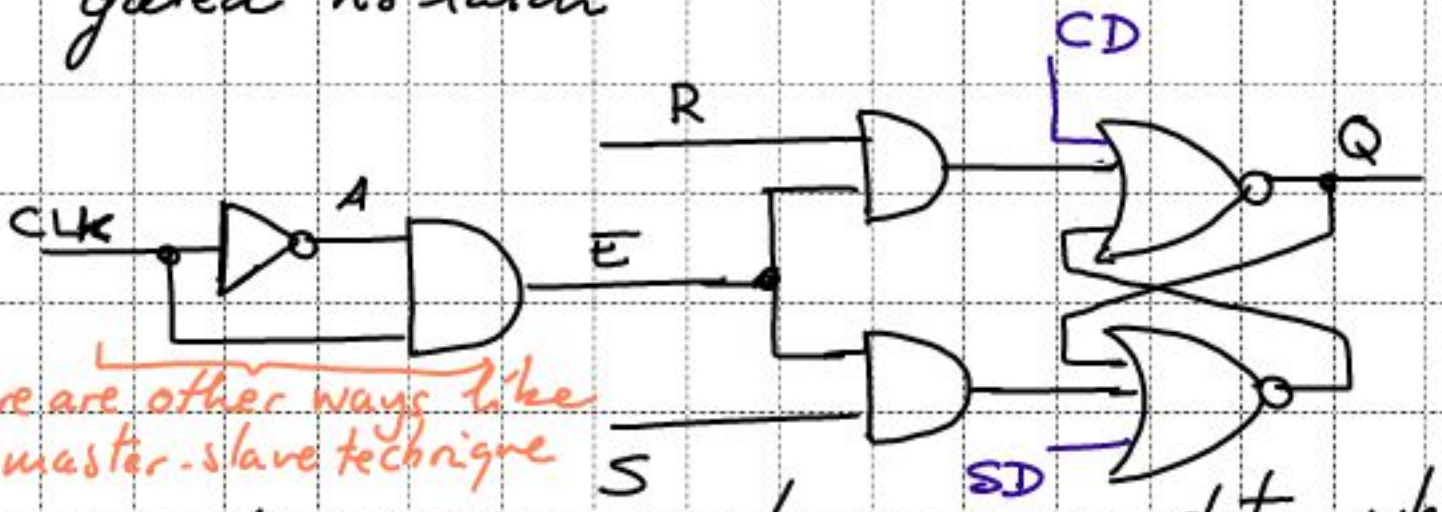


It looks like that the output is always zero

The inverter is <u>real</u> → it has a certain propagation time



④ What if we apply the edge detector circuit to the gated RS latch

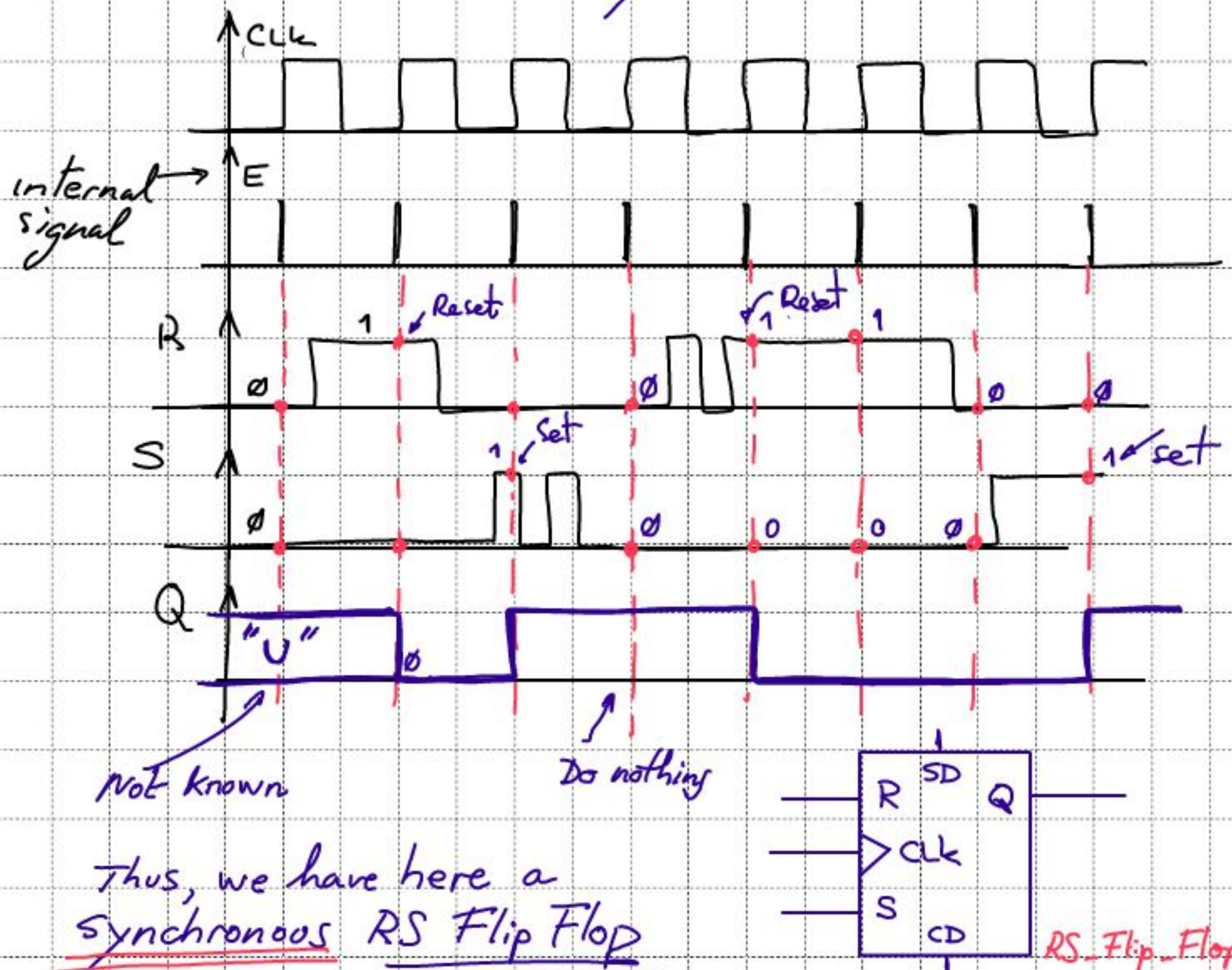


There are other ways like the master-slave technique

⇒ The circuit can only save new data when there is a CLK rising edge

→ Note the new inputs CD (Clear direct) and SD (set direct) which don't have to wait for an enable signal to save a '0' or a '1' when active.

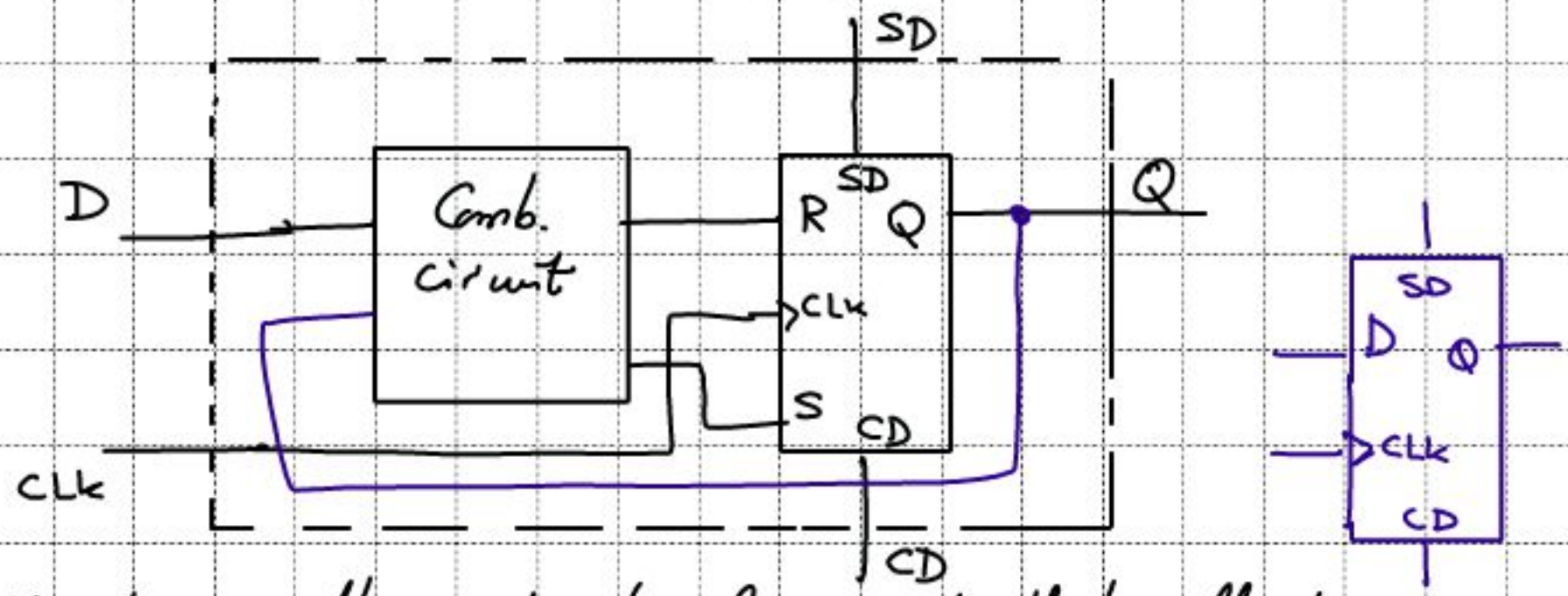
⇒ CD → asynchronous reset
SD → asynchronous set



RS_Flip_Flop

Thus, we have here a Synchronous RS Flip Flop

(5) And finally, if we have an RS Flip Flop, how to implement a D-FF (data flip flop)?



Let's deduce the combinational circuit that will drive the RS flip flop inputs.

This is the truth table required:

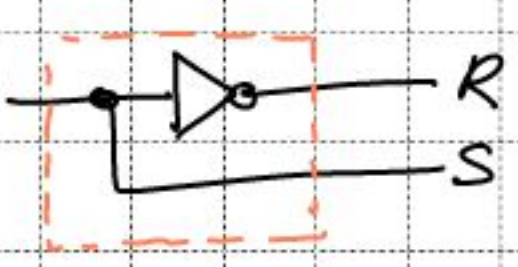
D	Q	R	S	Q^+	
Ø	Ø	x	Ø	Ø	maintain the zero
Ø	1	1	Ø	Ø	Reset
1	Ø	Ø	1	1	Set
1	1	Ø	x	1	maintain the one

This is what we want

This is what we need to apply:

Q	Q^+	RS
Ø	Ø	xØ
Ø	1	Ø1
1	Ø	1Ø
1	1	Øx

Thus the logic equations: $R = D'$
$S = D$



However, instead of using this demonstrative structural design to build a basic device like the D-FF, we'll use a VHDL behavioural definition which is easily understood by any EDA tool.

⟹ Go back to the web and examine the standard VHDL code for the D-FF that will be used in all the applications