Timing

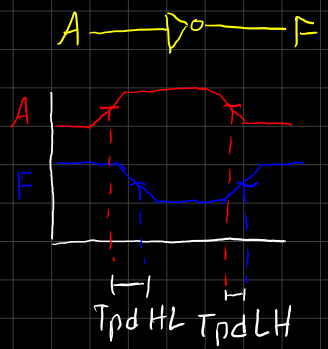
All gates have some delay before the output responds to the input.

Called Propogation delay or Tpd

Propogation delay when going Low → High is TpdLH

Propogation delay when going High → Low is TpdHL

Example Circuit:

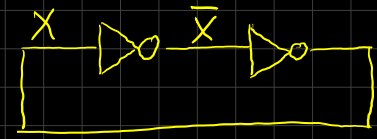


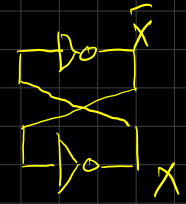
Combinational circuits can only respond to the current state of inputs

However, they cannot do anything in response to what has happened in the pasted.

They cannot store data

We need a way to store data so circuits can change behavior based on what has happened in the past.



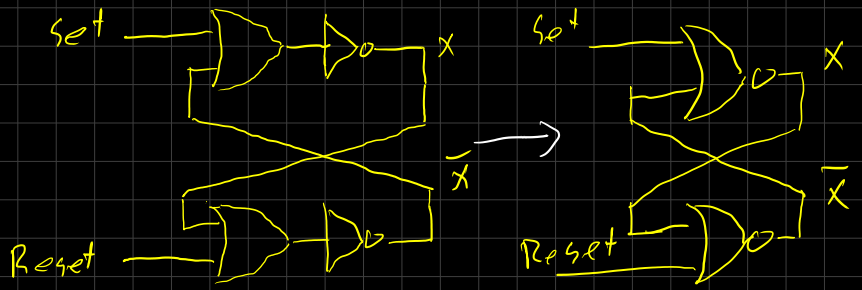


Same circuit drawn different by

If we wire two not gates together, they will become stable and hold a value of X forever. We have no way to tell what that value of x will be.

If we momentarily short the x wire to V_{cc}, we can set the value of x to 1, and it will stay there.

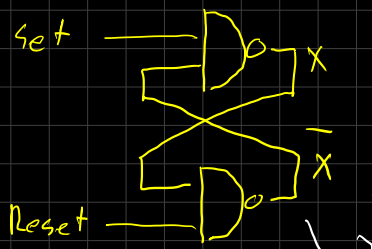
If we momentarily short the x wire to ground, we can set x to 0, and it will stay there.



By adding some OR gates, we can set and reset x with digital signals

Set	Reset	x	$\bar{x}$
0	0	Hold	Hold
0	1	0	1
1	0	1	0
1	1	Invalid State	

But we only have NAND gates, not NOR gates

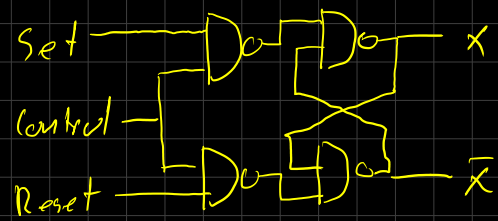


Set	Reset	x	$\bar{x}$
1	1	Hold	Hold
0	1	1	0
1	0	0	1
0	0	Invalid State	

↳ SR Latch (Set-Reset Latch)

Same as with NOR gates, but with inverted Set and Reset

But what if we want to control when we can set and reset?

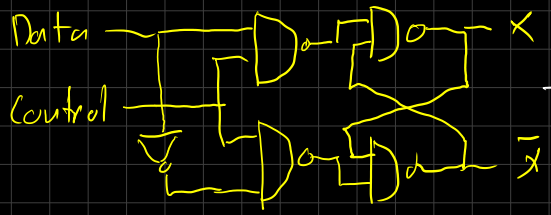


→ SR Latch w/ control (Set-Reset Latch with control)

Set and reset will only take effect when control is 1. We can also set and reset with 1 by using NAND instead of AND gates

Set (don't care)	Reset	Control	x	$\bar{x}$
x	x	0	Hold	
0	0	1	Hold	
1	0	1	1	0
0	1	1	0	1
1	1	1	Invalid state	

What if we do not want the invalid state to be a thing?



→ D Latch (Data Latch)

When control is 1, x gets set to whatever Data is. Otherwise, it holds its previous value.

Data	Control	x	$\bar{x}$
0	0	Hold	
1	1	1	0