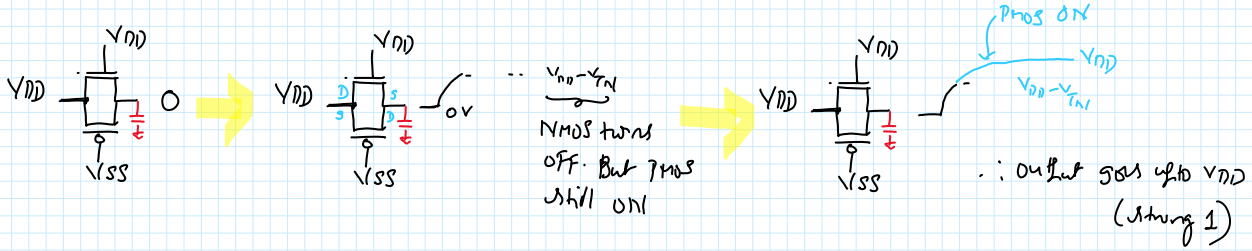
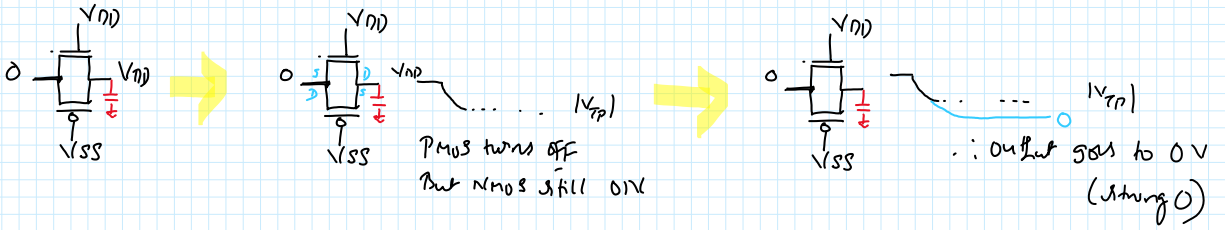
Lecture-10

* Recall that NMOS passes strong '0' and weak '1' (vice versa for PMOS)

* Consider a combination of NMOS and PMOS:-

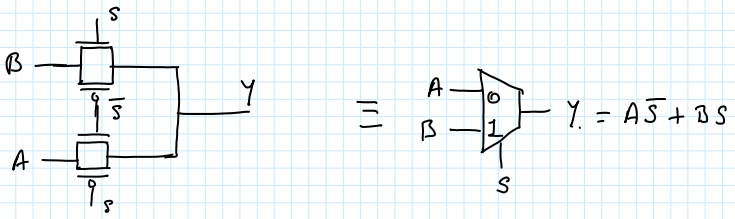




* This is termed as a "Transmission Gate" (TG) and a TG passes a strong 1 & strong 0.

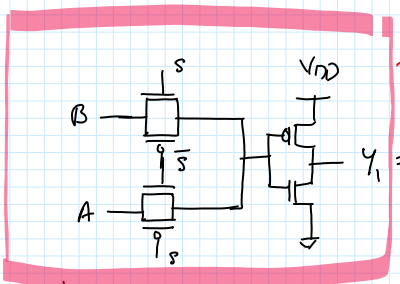
* A TG acts as a "switch" (without inversion unlike an inverter)

↳ It is easy to implement a 2 to 1 MUX using 2 switches (TG) :-



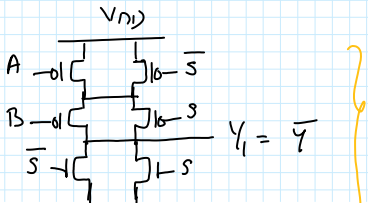
* This is not a static gate since by definition, the output node 'Y' is not tied to VDD or VSS through a low resistance path (i.e ON PMOS or ON NMOS).

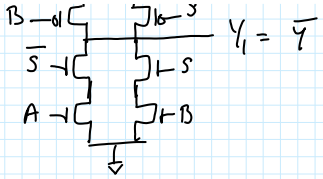
* To convert this into a static gate, we can simply add an inverter at the output.



→ we will come back to this circuit

* A similar logic using our earlier CMOS implementation -

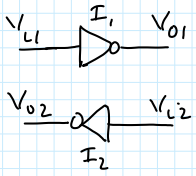


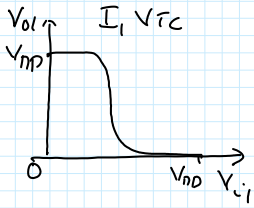


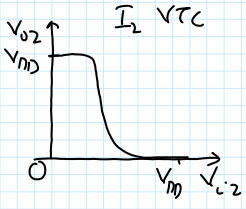
8 MOS

× ×

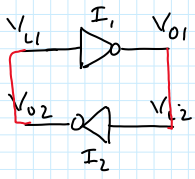
* Consider two inverters I_1 & I_2 & their VTC







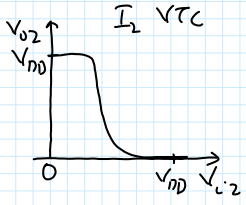
* We make a "back-to-back" inverter connection -



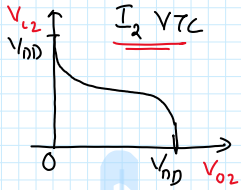
i.e $V_{i2} = V_{01}$ and $V_{i1} = V_{02}$

* To get the operating point of this new circuit, we need to find intersection on VTC curves.

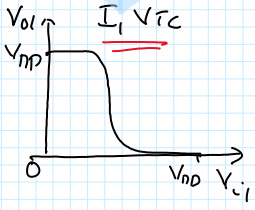
* First lets transform the VTC curve of I_2 (swap axes)

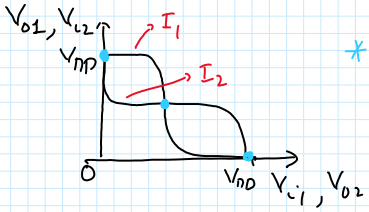


transform



Combine the two to find the operating points





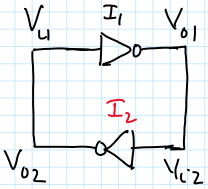
* only three possible operating points

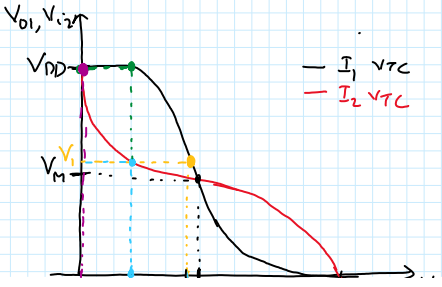
a) $V_{01} = V_{DD}$, $V_{02} = 0$

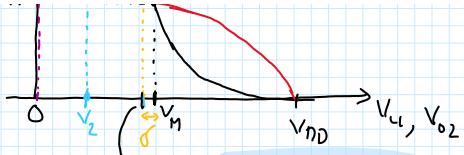
b) $V_{01} = 0$, $V_{02} = V_{DD}$

c) $V_{01} = V_{02} = V_M$.

* Now Consider the following :- at Power ON, $V_{i1} = V_{01} = V_{i2} = V_{02} = V_M$







Now there is a small perturbation @ V_{i1}, i.e $V_{i1} = V_M - \delta$ (δ is a very very small voltage)

Follow the following -

a) $V_{i1} = V_M - \delta$ is applied

↳ $V_{o1} = V_1$ (for this new input) [Yellow traces]

b) V_1 will be the new input to I_2 ($V_{i2} = V_1$)

↳ $V_{o2} = V_2$ (for this new input) [Blue traces]

c) V_2 will be the new input to I_1 ($V_{i1} = V_2$)

↳ $V_{o1} = V_{DD}$ (for this new input) [Green traces]

d) V_{DD} will be the new input to I_2 ($V_{i2} = V_{DD}$)

↳ $V_{o2} = 0$ (for this new input) [Purple traces]

⇒ Given that the back to back inverter happens to be biased at V_M,

↳ A small $-\delta$ change @ V_{i1} drives V_{i1} to 0 and V_{i2} node to V_{DD}

↳ Similarly small $+\delta$ change @ V_{i1} will drive V_{i1} to V_{DD} and V_{i2} node to 0

* In reality a small circuit noise will act like this δ voltage and take the output nodes to V_{DD} or 0

∴ In reality (practical scenarios), the back to back inverter has only 2 states

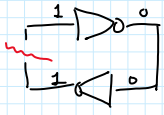
→ either V_{DD} or 0

* Once the output nodes have reached V_{DD} or 0, any small δ change will not have any impact since in this region (in the VTC), the slope is always < 1 and no noise is amplified (recall our discussion on NM, slope $= \pm 1$ points).

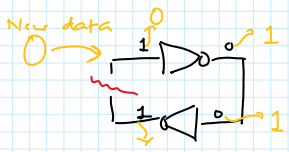
∴ A back-to-back inverter can act like a 1-bit memory

↳ To change the data to be stored in this 1-bit memory cell we need to

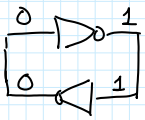
(a) open the back to back connection



(b) Apply new data to be stored

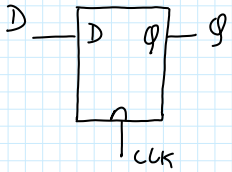


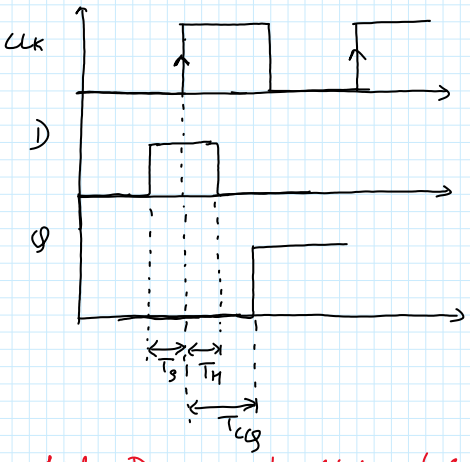
(c) Close the connection, to preserve the state



— × — × — × — × —

* 1-bit register (D FF) has following timing characteristic





T_S Setup time : Time for which D must be stable before the clock edge

T_H Hold time : Time for which D must be held after the clock edge.

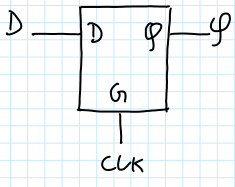
T_{CQ} : CLK-to-Q delay : Delay from clock edge to output changing state.

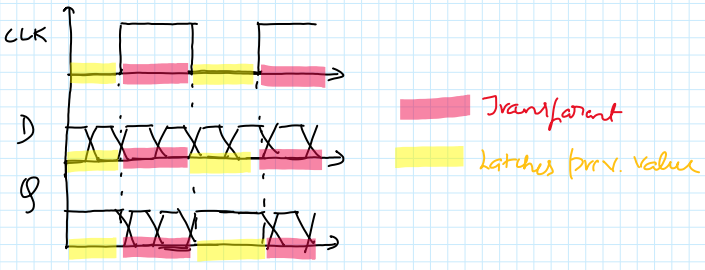
* Similar definitions for both +ve & -ve edge triggered DFF

— × — × —

Latch :

* +Ve Latch (G is gate input) : Latch is transparent (o/p = i/p) when G is high.



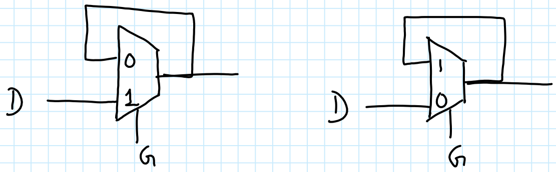


* -Ve latch is transparent when G is low.

— × — × —

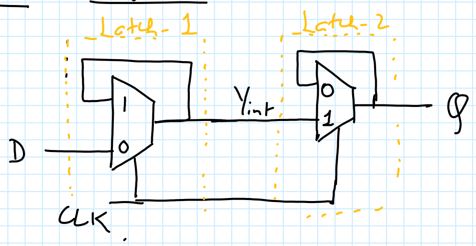
* Now follow the sequence of implementations :-

a) Latches using MUX :-



+ve Latch —— −ve Latch

b) Flip Flop using Latches :-



* +ve edge triggered flip flop.

* Latch - 1 : master

Latch - 2 : slave

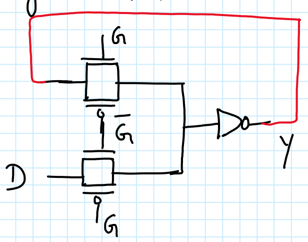
* CLK = 0 → Latch - 1 : transparent ($Y_{int} = D$)
  ↳ Latch - 2 . Holds prev state ($Q = D^-$)

— Clock edge b/t these two

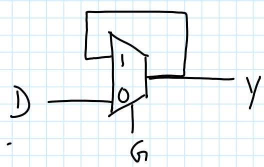
* CLK = 1 → Latch - 1 . Holds current state ($Y_{int} = D$)
  ↳ Latch - 2 : transparent ($Q = D$)

Flip Flop at the gate level (evolution) :-

1) Latch using MUX :-



≠



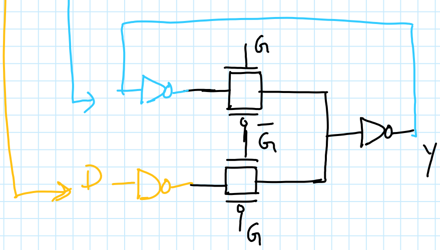
−ve Latch

* G = 0 , $Y = \overline{D}$ —— BUT WE NEED $Y = D$

* G = 1 , we need to preserve prev. state (i.e need 1-bit memory)
  ↳ we already discussed, 1-bit memory → 2 back to back inverters

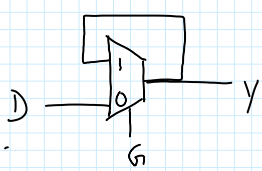
2)

Solved —— Solved



−ve latch

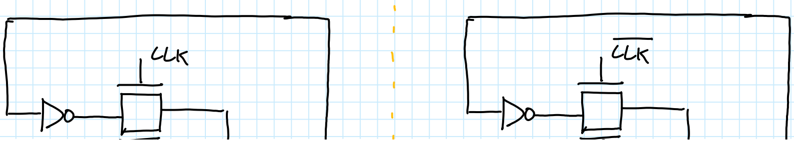
≡

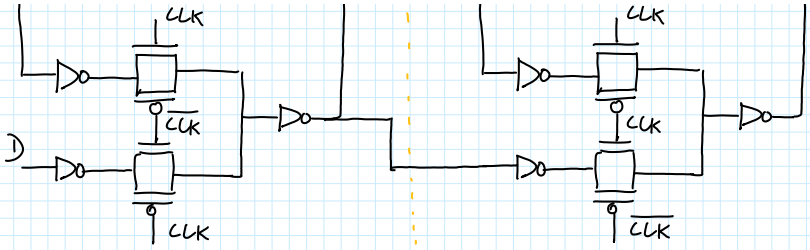


−ve Latch

3) +ve edge triggered FF (−ve Latch followed by +ve Latch)

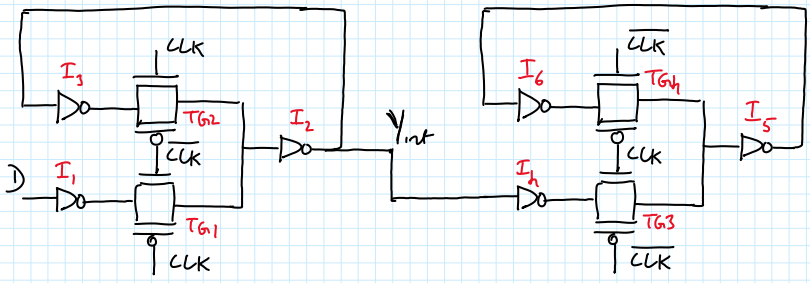
Master −ve Latch —— slave +ve Latch

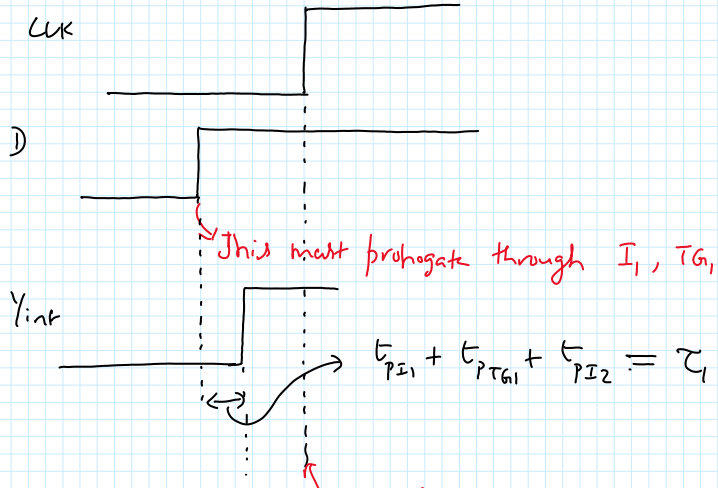




— × — × —

* Now consider finite propagation delay of inverters and transmission gates.

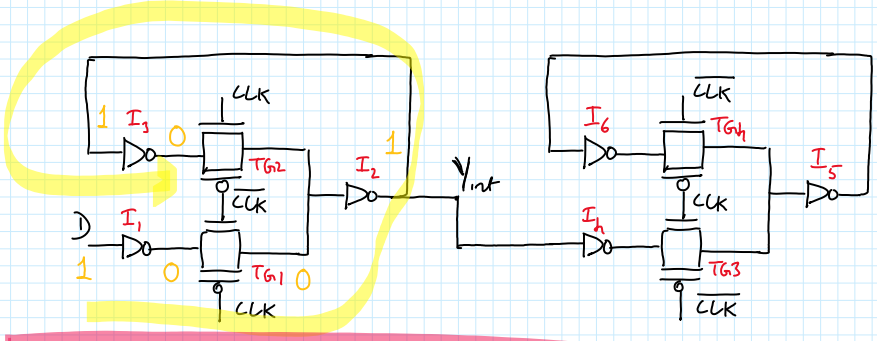




⇒ D must arrive atleast τ_1 before the clock edge !!! (Setup-time ??)

* Note that before making CLK high TG2 must see same logic states at either sides !!

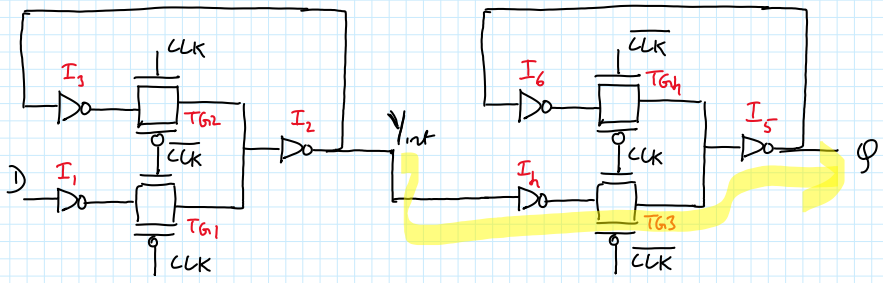
i.e the D input must propagate through I_1, TG_1, I_2 and I_3 BEFORE the CLK is made high !!



$\therefore T_s = t_{pI_1} + t_{pTG_1} + t_{pI_2} + t_{pI_3}$ SETUP TIME of DFF

* Notice that when CLK = 0, TG1 is OFF and the D-input is already latched. Hence no additional time is required to hold the data 'D' after CLK going high.

$\Rightarrow T_h = 0$



* Consider the CLK going from low to high. $\Rightarrow Y_{int}$ will reach Q as latch-2 is transparent

$\Rightarrow$ From the CLK edge, Y_{int} has to travel through I_4, TG3 and I_5 to reach Q. (T_{cq})

* However note that I_3 and I_4 are similar and

(a) the delay associated for Y_{int} to reach I_3 output = delay associated for Y_{int} to reach I_4 output.

(b) Since the delay of I_3 is already in the setup time, the T_{cq} should not include I_4 delay (i.e data has already reached output of I_4 before clock edge because of setup time restriction)

$\Rightarrow$ $T_{cq} = t_{PTG3} + t_{PI5}$.