# Lecture-7

* Several drawbacks in TTL/ECL logic:-
  * Static power consumption due to finite base-emitter current
  * Voltage swing @ the output is not rail-to-rail
  * Area consumed is higher compared to CMOS for the same logic implementation.

## Static CMOS logic Gates:

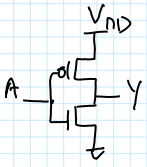
CMOS → Complementary MOS: the implemented gate uses both NMOS & PMOS transistors

Static: Unlike dynamic logic (which we will cover later in the course), in static CMOS, at any point in time, the gate output is connected to either $V_{DD}$ or GND through a low resistance path.

* A quick example for an inverter based on static & dynamic logic


— × — × —

* Revisit the static CMOS inverter.

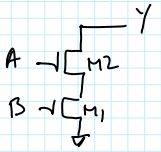



* If we blindly saw the circuit from NMOS side, it looks like the output is complement of the input.

  [i.e If A = 1, Y = 0 since MOS is ON]


So it looks like we can design any logic simply looking from the NMOS side

* Consider the example below with 2 series NMOS gates.




* The output Y will be zero if both MOSFETs M1 & M2 are ON

  i.e If A = 1 & B = 1, Y = 0

* This looks a lot like a NAND Gate!! $Y = \overline{A \cdot B}$

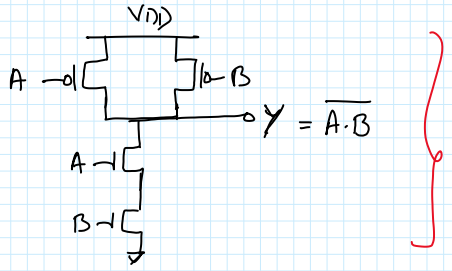
* i.e we compute the logic from NMOS side (AND logic in this case) & invert the output Y to obtain the complete logic

* However, we know that in static CMOS logic both NMOS & PMOS cannot be ON together, by definition

* However, we know that in static CMOS logic both NMOS & PMOS cannot be ON together, by definition

⇒ If the NMOS are in series, the corresponding PMOS must be in parallel.

i.e




} is a Complete Static CMOS NAND Gate

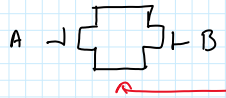
* ∴ To design any logic in static CMOS :-

(a) Implement the logic on the NMOS side & complement the output Y.

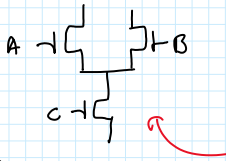
(b) Add PMOS in connections opposite to NMOS [i.e if NMOS is in series, PMOS will be in parallel & vice versa]

Eg: Implement $Y = \overline{(A+B)C + D}$

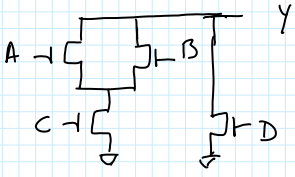
→ First implement $(A+B)$ i.e NMOS in parallel




→ Next implement $(A+B)\cdot C$ i.e additional NMOS in series with

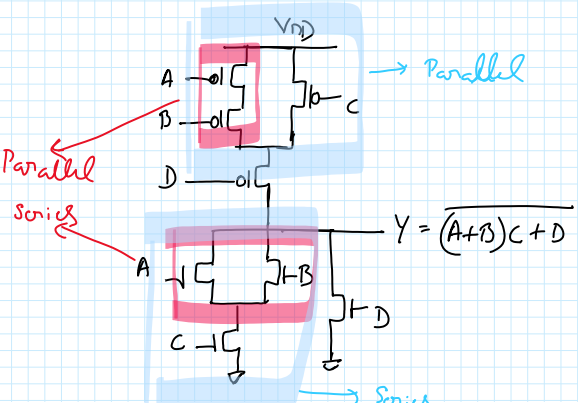



→ Finally implement $(A+B).C + D$ i.e an additional parallel NMOS with

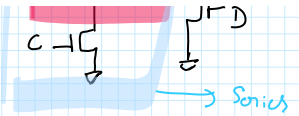



Y (effectively $Y = \overline{(A+B)\cdot C + D}$)

→ Now add all PMOS transistors



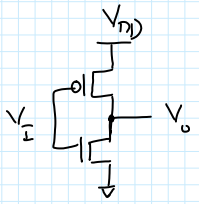

STATIC CMOS with "N" inputs Need "2N" MOSFETs




Need 2N MOSFETs

— x — x —

* Consider the Inverter & its Voltage transfer curve (VTC)




If $\frac{\mu_n}{\mu_p} = 2$ & $V_{TN} = |V_{Tp}|$

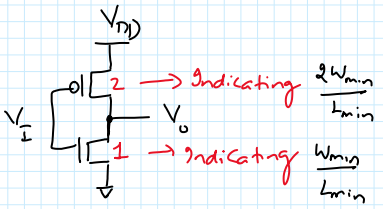
For $t_r = t_f$ $\beta = 1$

$$\Rightarrow \frac{W_p}{W_N} = \frac{2}{1}$$

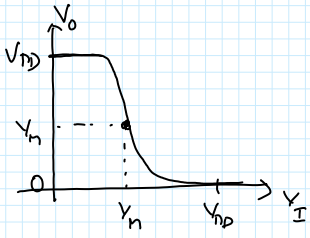
i.e NMOS Aspect ratio : $\frac{W_{min}}{L_{min}}$

PMOS Aspect ratio : $\frac{2.W_{min}}{L_{min}}$

* For quick abstraction, we take all MOS lengths as $L_{min}$ & represent the circuit

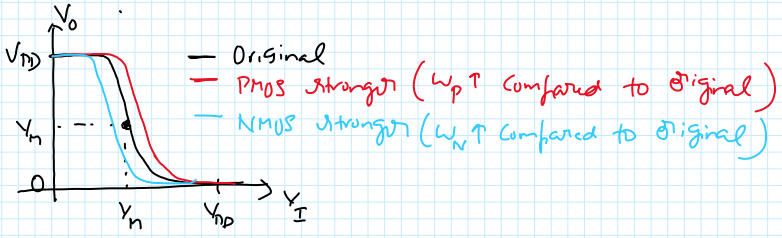



VTC




* Now assume the PMOS strength (i.e Width of PMOS) increases from 2 to 2.5.

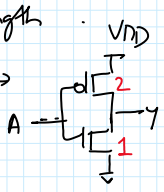
∴ Intuitively, in-and-around $V_m$, both PMOS & NMOS are fighting for the o/p voltage However, since the PMOS strength has increased (and PMOS is connected to $V_{DD}$), the output tends to stay near $V_{DD}$ for a longer time than before.



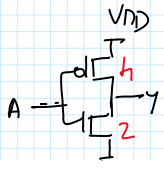

— x — x —

* We define logic gates with strength.

For example an INVX1 ⟶



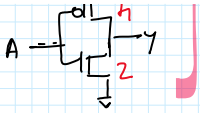

an INVX2 ⟶




i.e strength of individual MOS increases (W↑)

∴ $t_r$ & $t_f$ will reduce with higher strengths for the same cap load

[∴ $t_r = t_f \propto R_{on} \propto \frac{1}{W}$]

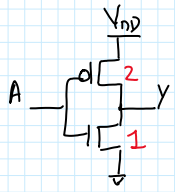


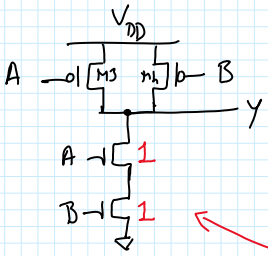

$\therefore t_r = t_f \propto R_{on} \propto \frac{1}{W}$

* A symmetric INVx1 will be taken as a reference (i.e $t_r$ & $t_f$ of INVx1) to define the strength of other logic gates

——— x ——— x ——— x ———

* Now consider an INVx1 and a NANDx1 side by side. Since both are x1 strengths, the worst-case $t_r$ and $t_f$ of the NAND gate cannot exceed beyond that of INVx1 !!



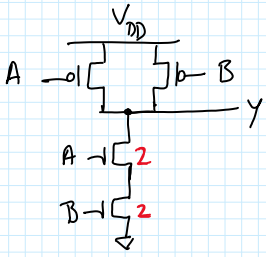




* $t_f$ : The inverter offers $R_{ON,N}$ resistance

The NAND gate offers $(R_{ON,N1} + R_{ON,N1})$ resistance with the sizing shown

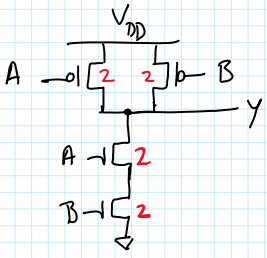
. To get the same $t_f$ as inverter, the NMOS ON resistance has to be halved i.e W has to be doubled $\left(R_{ON} \propto 1/W\right)$

. For X1 strength NAND -




* $t_r$ :- The inverter offers $R_{on,P}$ resistance

Let's start with an assumption that the PMOS devices are sized exactly same as Inverter




The NAND gate can offer two different effective resistance depending on input

(a) If A=0, B=1 or A=1, B=0 } at least ONE PMOS is ON and the effective resistance will be $R_{ON,P}$

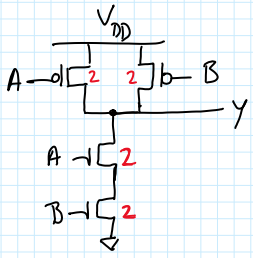
(b) If A=0 & B=0 → both PMOS is ON (and in parallel) and the effective resistance will be $R_{on,P}/2$

will be $R_{onp}/2$

* Notice that from $t_r$ perspective, case (a) is the worst-case (highest delay) & case (b) is the best-case (smallest delay)

* We will design the gate for the worst-case scenario (i.e. if A=0, B=1 or A=1, B=0, the $t_r$ will be same as that of INVx1)

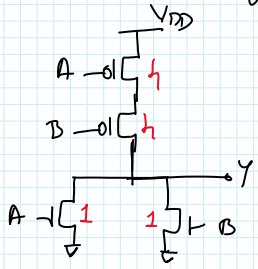
NANDx1 ⟶




[Remember if $\mu_n/\mu_p$ changes, sizing will change]

— x — x —

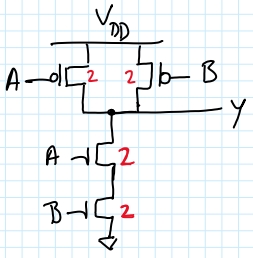
* Similarly for a NOR gate, the sizing will be




work it out

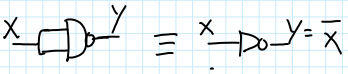
— x —

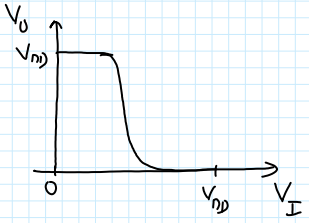
* Now, let's look at the VTC of the NANDx1



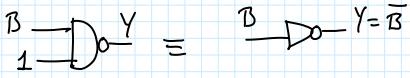

* Let's consider the condition A=B

The case is similar to an inverter!!



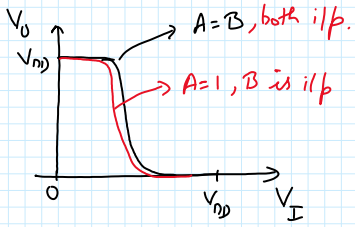

* Now consider the case A=1 & B takes input




* Again, the VTC must be similar to an inverter.!!

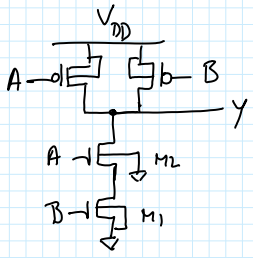
However, A=1 means, one of the PMOS is always OFF

∴ Effectively, the PMOS strength is reduced by half & therefore VTC curve must shift left.




* The same argument holds good if B=1 and A is the input.

* However, remember, in bulk CMOS technology, all NMOS bulks are grounded.




For a MOS, $V_{TH} = V_{TH0} + \gamma\left[\sqrt{|2\phi_F| + V_{SB}} - \sqrt{2\phi_F}\right]$

$V_{SB}$ ↓ Source to bulk voltage

∴ $M_2$ will have a higher $V_{TH}$ compared to $M_1$ and hence the VTC curve will not overlap for the cases

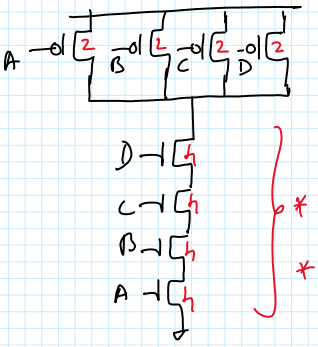
(i) A=1 & B is input

(ii) B=1 & A is input

---

* As the input count of the gate (Fan-in) increases, static CMOS logic will have an impact

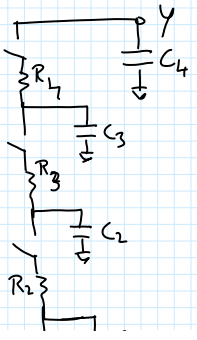
Consider the example of 4 input NAND X1 gate



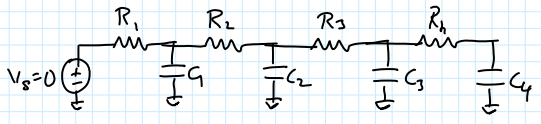

* All NMOS's will have area 4 times that of the NMOS in the inverter

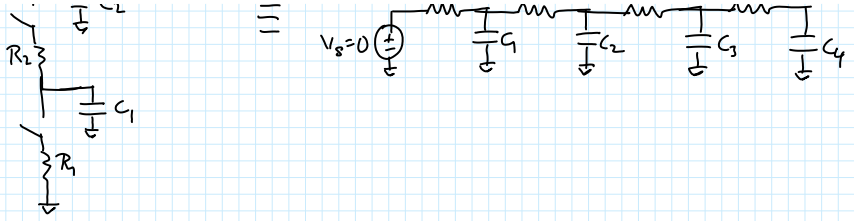
* The capacitors (diffusion caps in Lecture-6) between drain-to-bulk & source-to-bulk will increase

* Effectively




≡

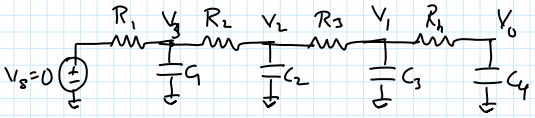





* Elmore Delay (quick first order way to calculate delay between 2 nodes)
  - ↳ If network has one input node
  - ↳ Network does not have resistive loops →
  - ↳ Network has all capacitors w.r.t ground

Then delay from input node $V_s$ to node "i" is (For a N/w with "N" nodes)

$$\tau_i = \sum_{k=1}^{N} R_{ik} C_k$$




* $\tau$ from $V_s$ to $V_3$ is $R_1C_1$
* $\tau$ from $V_s$ to $V_2$ is $R_1C_1 + (R_1+R_2)C_2$
* $\tau$ from $V_s$ to $V_1$ is $R_1C_1 + (R_1+R_2)C_2 + (R_1+R_2+R_3)C_3$
* $\tau$ from $V_s$ to $V_0$ is $R_1C_1 + (R_1+R_2)C_2 + (R_1+R_2+R_3)C_3 + (R_1+R_2+R_3+R_4)C_4$

* In our case $R_1 = R_2 = R_3 = R_4 = R_{ON}/4$ [$R_{ON}$ is resistance of NMOS in INVX1]

$$\therefore \tau = \frac{R_{ON}}{4}\left[C_1 + 2C_2 + 3C_3 + 4C_4\right]$$

* $C_4$ is the load and in the absence of $C_1, C_2, C_3$, $\tau \rightarrow \frac{R_{ON}}{4}[4 \times C_4]$ → same as INVX1
  - An additional delay is contributed.

⇒ In static CMOS logic
- ↳ As Fan-IN ↑
  - ↳ W of either PMOS or NMOS ↑ (NOR vs NAND)
    - ↳ Parasitic Cap ↑
      - ↳ Delay ↑