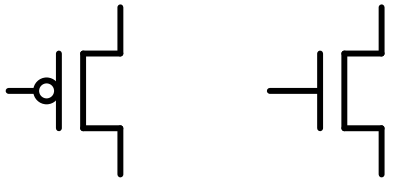
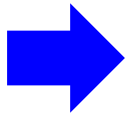
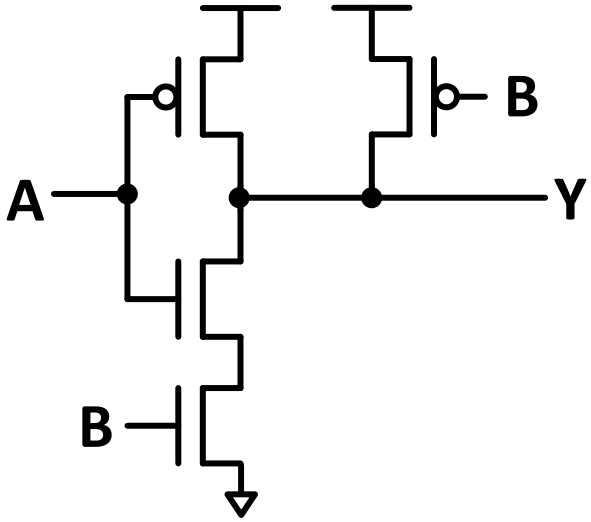
CMOS NAND2;
Any Other Way to NAND2 Implementation?



A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0





Logic Circuit Families (1)

Hanwool Jeong

hwjeong@kw.ac.kr

Contents

- Pseudo nMOS
- Cascode Voltage Switch Logic
- Dynamic Circuit and Domino Logic
- Pass-Transistor Circuits

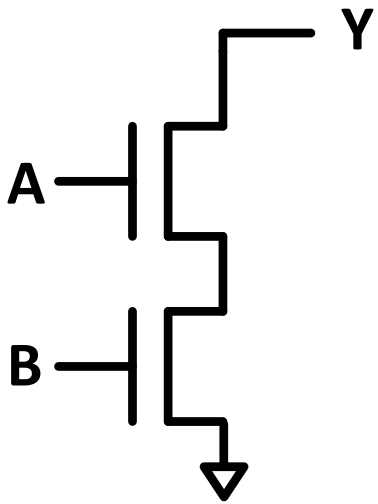
Pseudo nMOS

- ✓ Constructing logic only with pull-down network

Look into How We Form A Logic!

- Y = $\overline{AB}$
- When you form pull-down network, what was your thought?

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

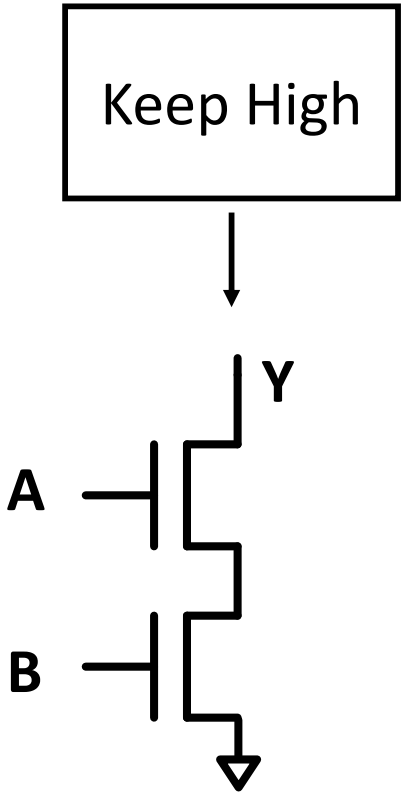


- All the required information is already included in the pull-down network.

Then How About This?

- Can really Y be decreased by AB = 1?

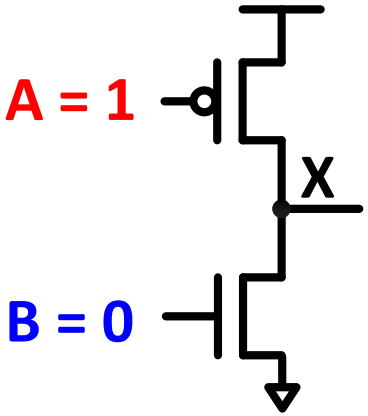
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



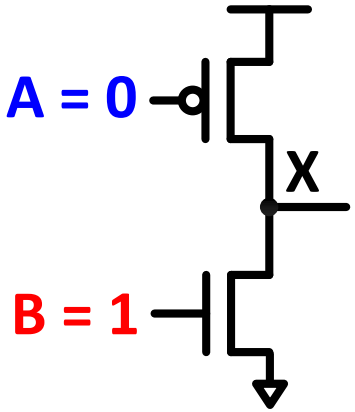
You Remember Contention?

- How is X determined?

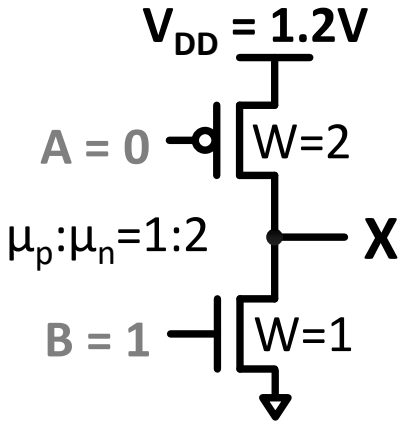
➔ **Note that the current thru pMOS and nMOS (KCL) must be same.**

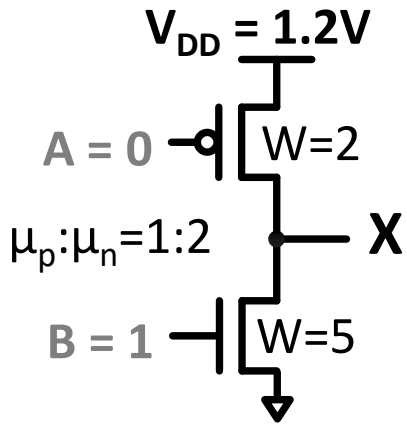


Floating



Contention



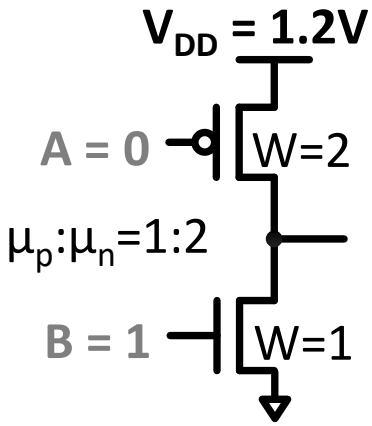


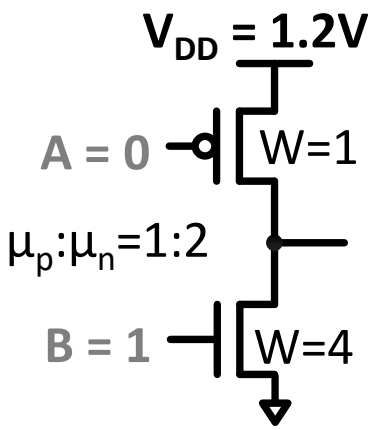
$I_D = \mu C_{ox}(W/L)(V_{GS}-V_{th})^2$ **$(1+\lambda V_{DS})$**

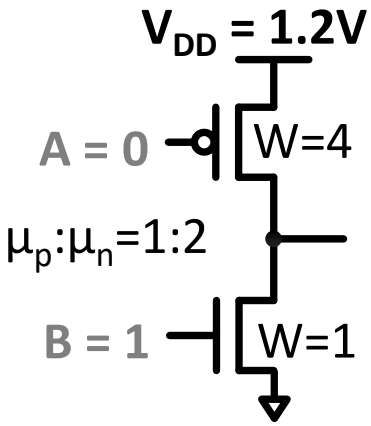
- ✓ $\mu C_{ox}(W/L)$ is same
- ✓ Vov = VGS-Vth is same
- ✓ Current is same for any X?

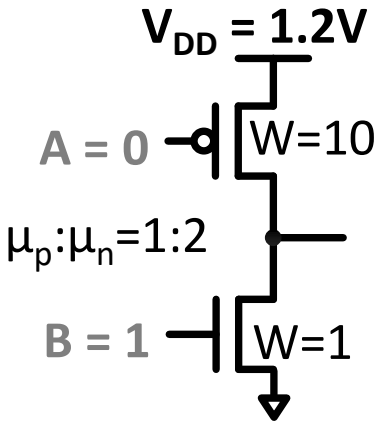
Output by Contention Circuit

- How is X determined? ➔ Dependent on P-N Ratio.



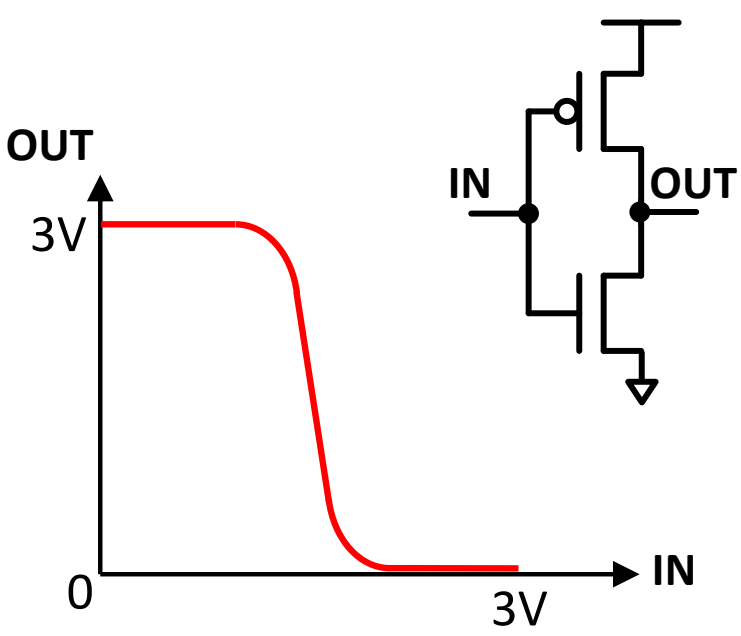


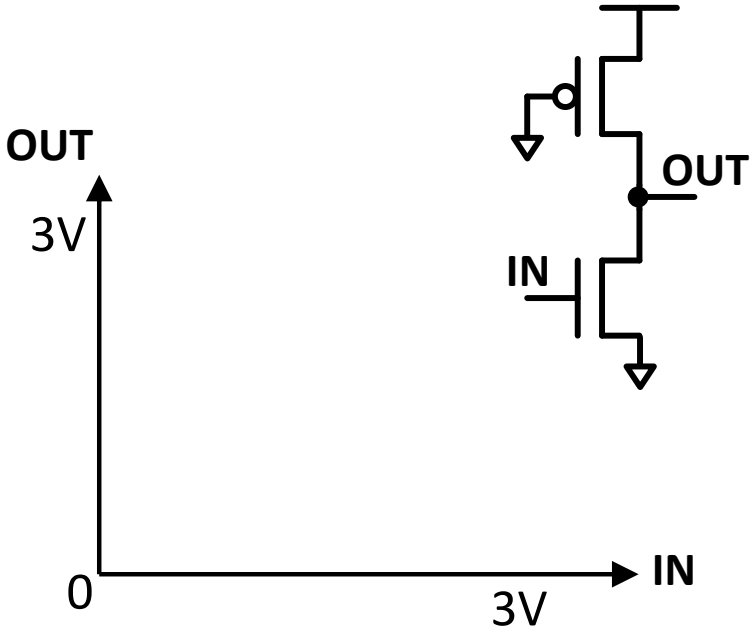




Voltage Transfer Curve (VTC)

- Recall that input vs. output **at the steady state can be represented through VTC**

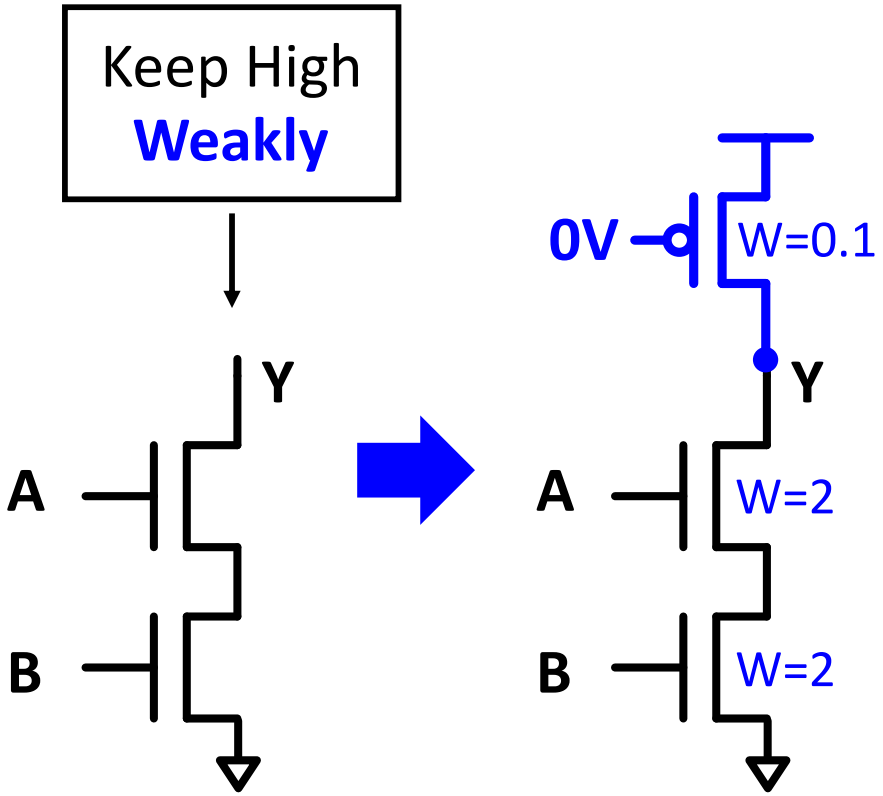




Back to NAND2 Design

- Now, Y can be lowered sufficiently ➔ Related to "**Noise Margin**"
- No problem for rising performance?

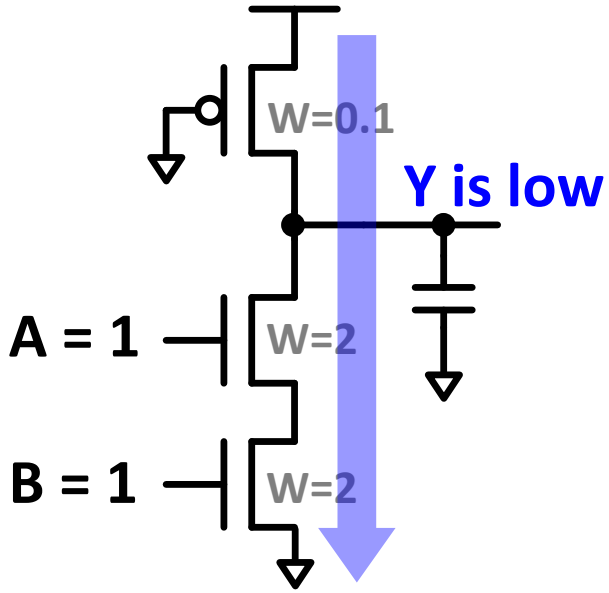
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



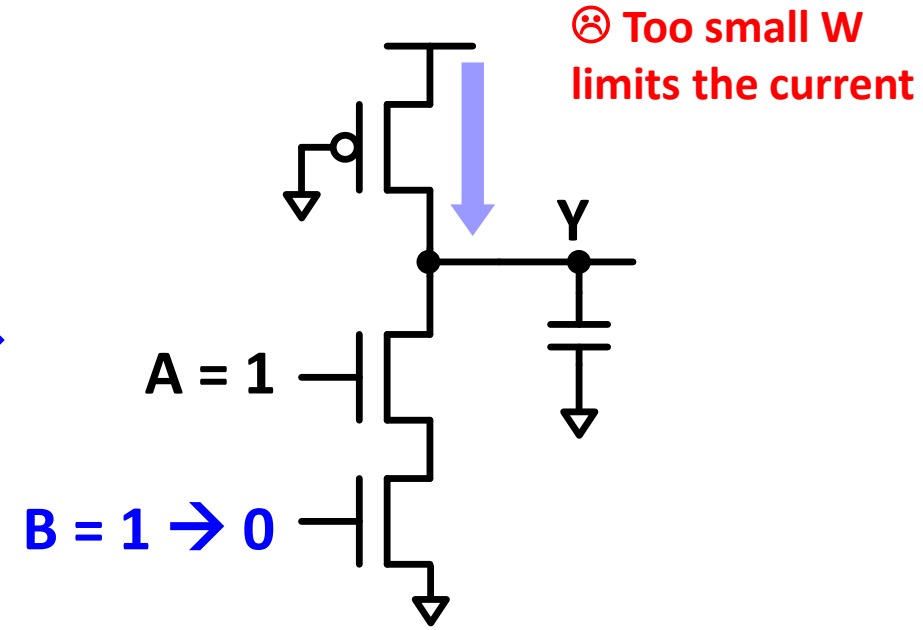
Rising Procedure

- Y : 0 → 1

When Y is low



When Y rises



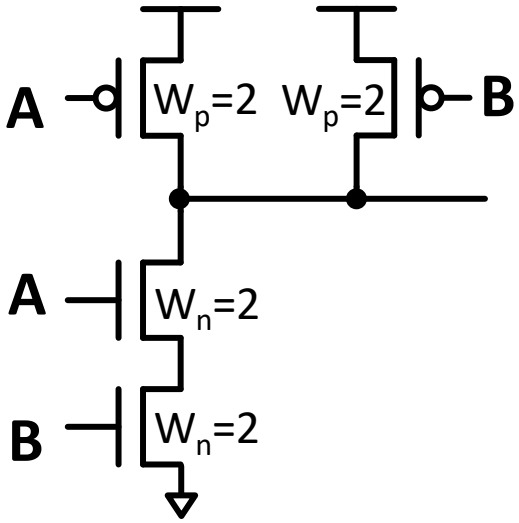
Designing Ratioed Circuit

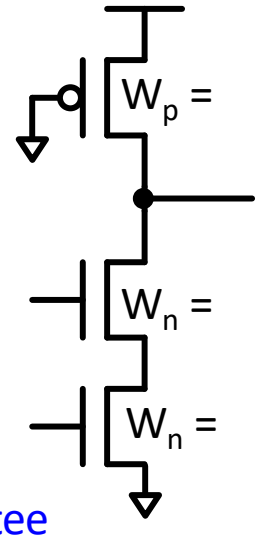
- Sizing constraint
  - Too strong pMOS : low output level is degraded (+Falling speed)
  - Too weak pMOS : Rising speed is degraded

➔ **We need a compromise between noise margin and speed.**

➔ For example, pMOS width is selected to have ¼ strength of pull-down network.

- For simplicity, we can start with inverter logic.

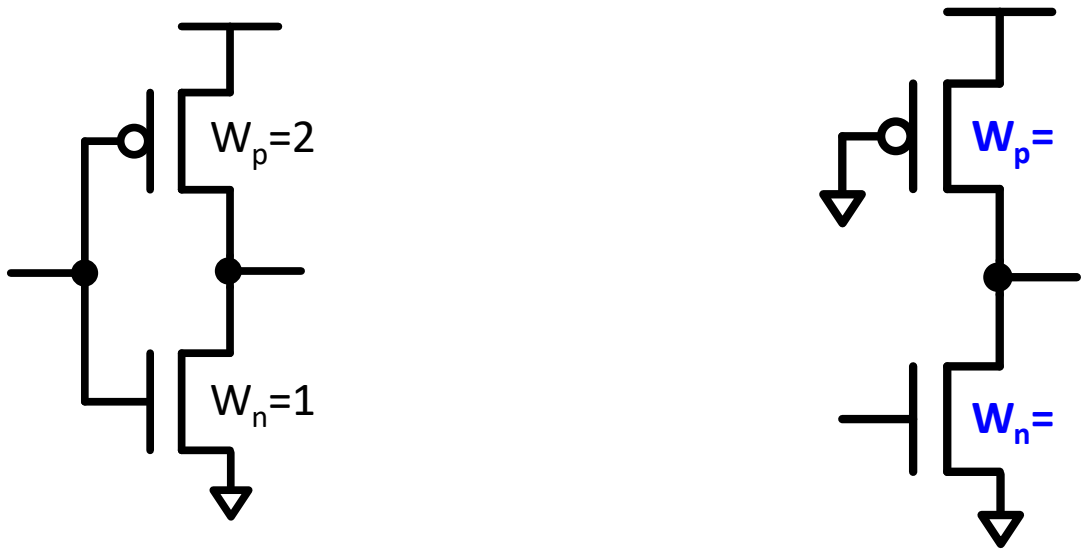




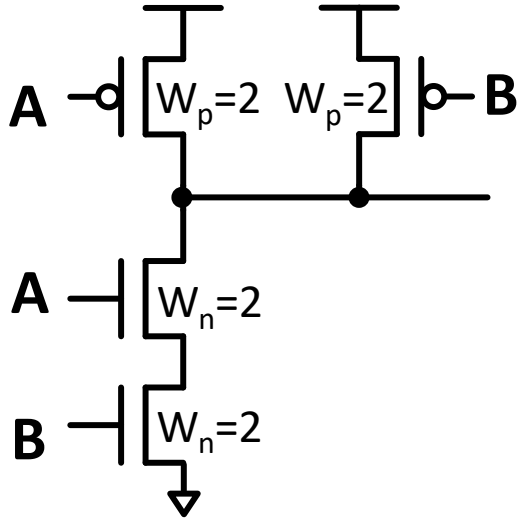
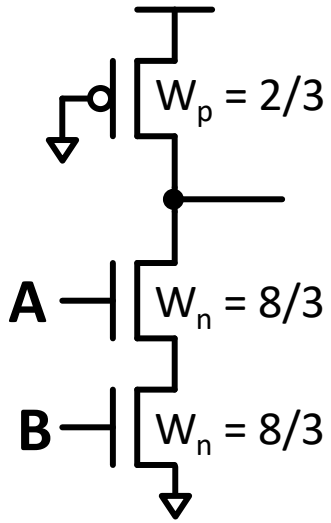
How to determine
W_p and W_n to guarantee
the same pull-down strength?

Example; Inverter Logic

- Assume $\mu_p:\mu_n = 1:2$, and
- Make the pull-up network strength is ¼ of pull-down network strength for noise margin.
- How to determine W_p and W_n to guarantee the same **effective** pull-down network strength?

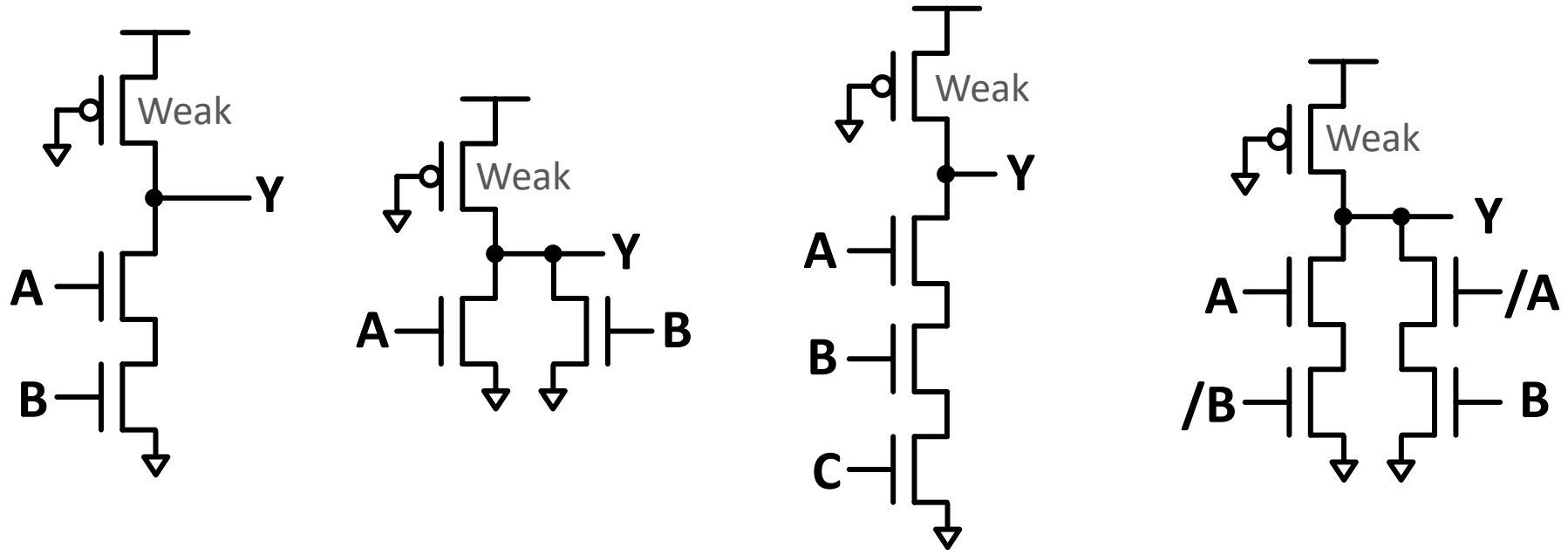


Speed Comparison; NAND2

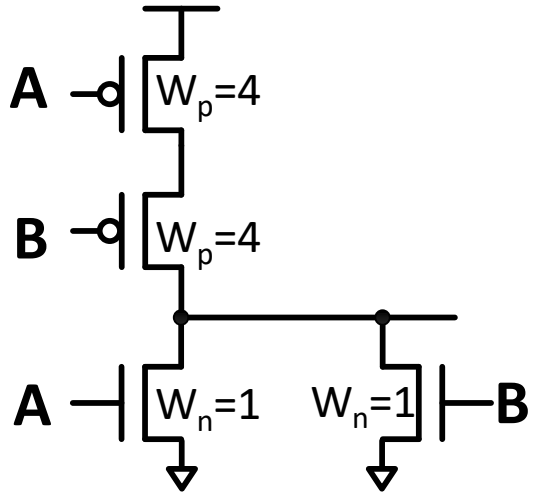
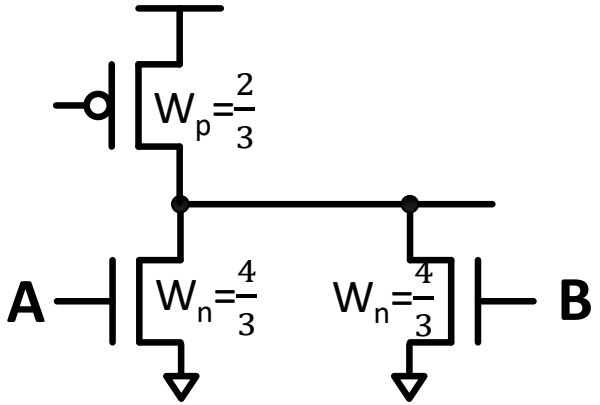
		CMOS NAND2	What we designed
Transistor-level Circuit			
Parasitic Delay	Rise	**R×6C**	**3R×($^{10}/_{3}$C) = 10RC**
	Fall		**R×($^{10}/_{3}$C) = 3.34RC**
Input Capacitance		**4C**	**$^{8}/_{3}$C**

Pseudo nMOS Logic

- You can also design an inverter, NAND3, NOR2, XOR2,.. Etc.



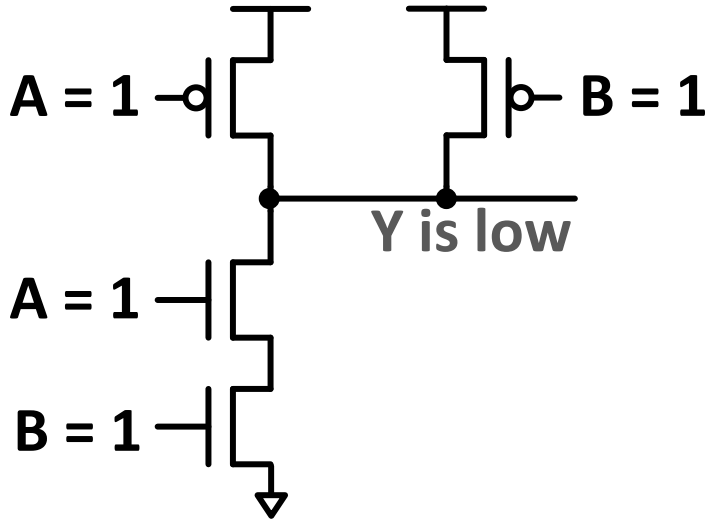
Speed Comparison; NOR2

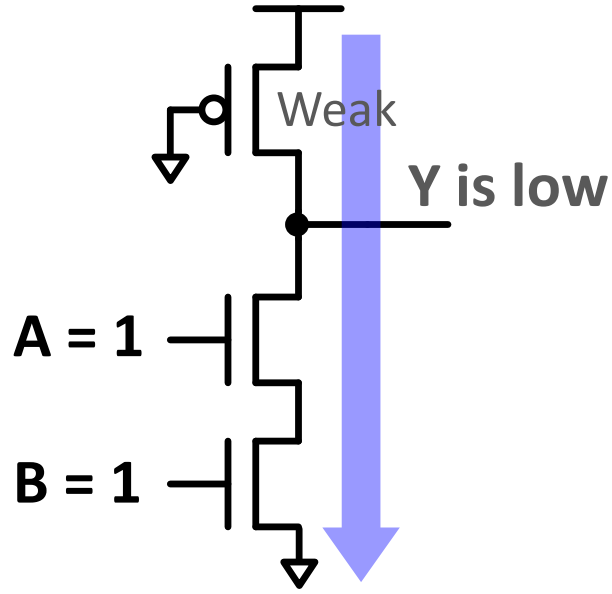
		CMOS NOR2	Pseudo nMOS NOR2
Transistor-level Circuit			
Parasitic Delay	Rise	**R×6C**	**3R×($^{10}/_{3}$C) = 10RC**
	Fall		**R×($^{10}/_{3}$C) = 3.34RC**
Input Capacitance		**4C**	**$^{4}/_{3}$C**

Problem of Pseudo NMOS
Static On-current for Pull-Down Enabled

- Static power consumption would be exceedingly large

➔ **Non-practical**



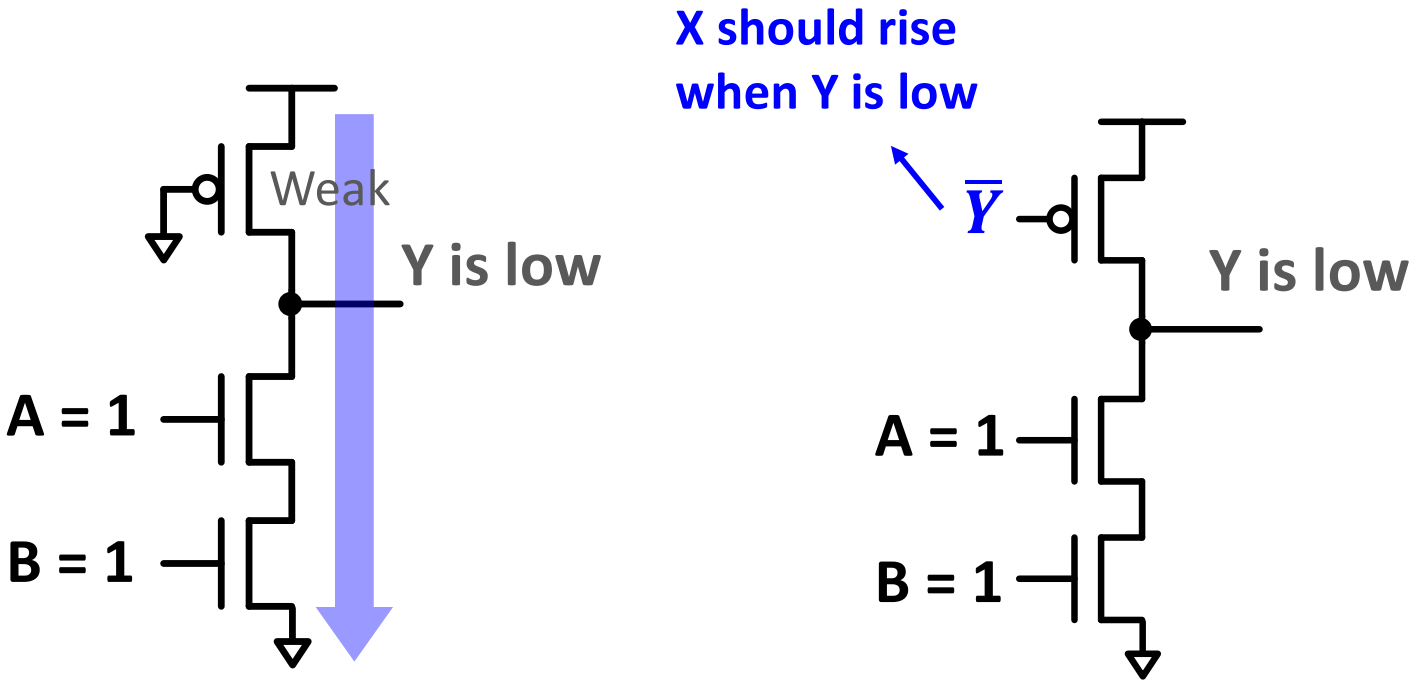


Cascode Voltage Switch Logic

- ✓ Constructing logic only with pull-down network

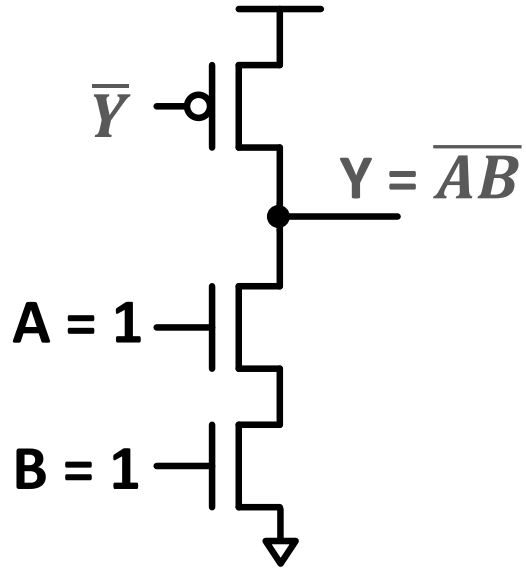
Revisit Pseudo nMOS NAND2

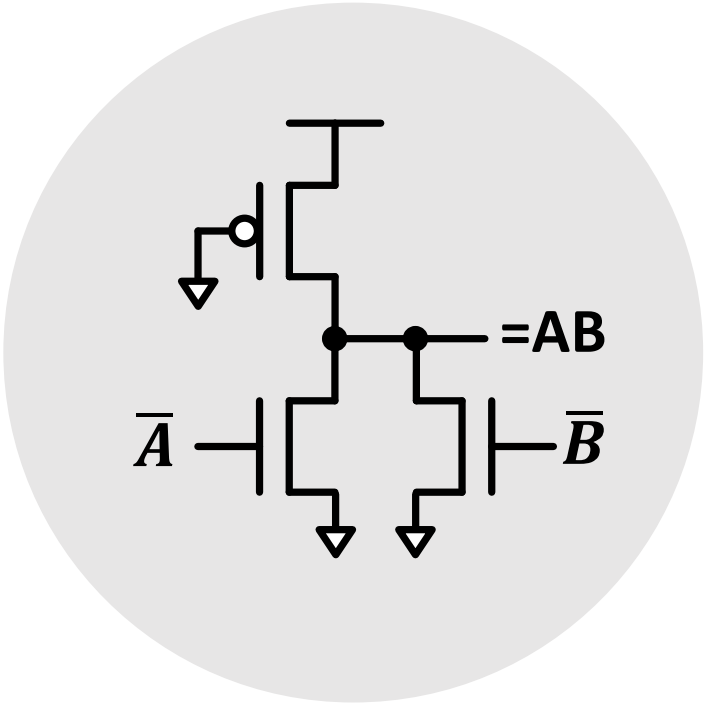
- Is there any way to block pull-up when Y is to be low by A=1 and B=1?



Generating Complementary Output

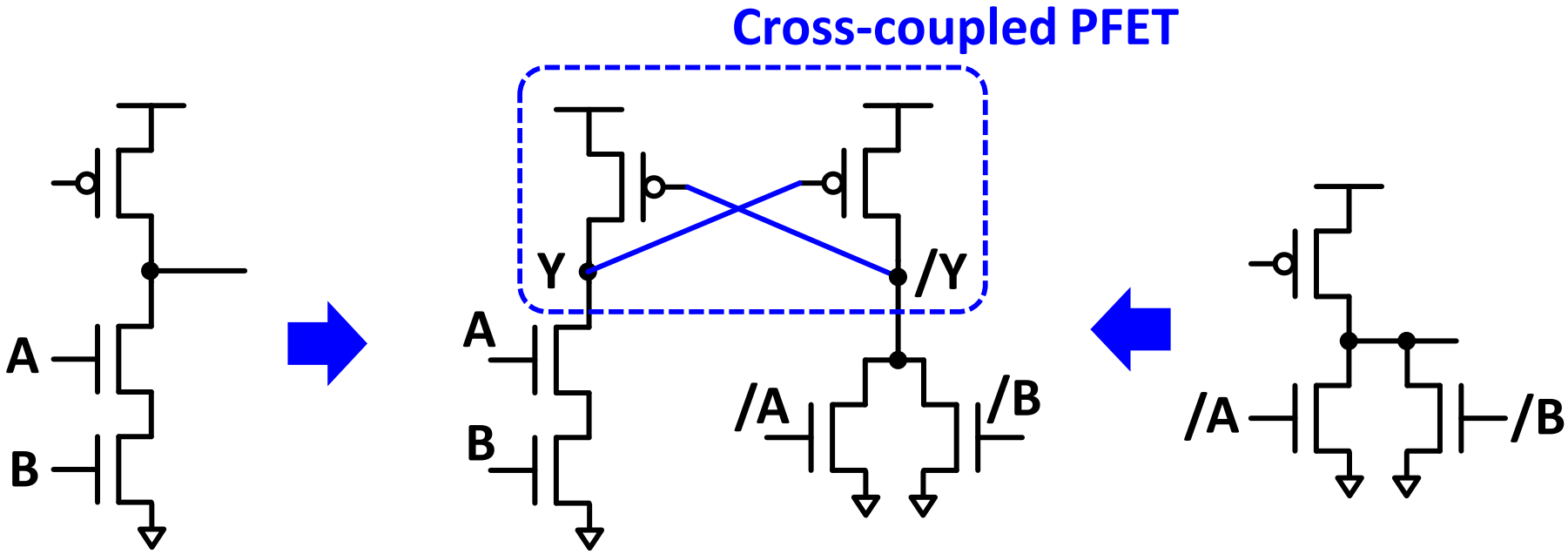
- How can we generate $\overline{Y}$? = **AB?**





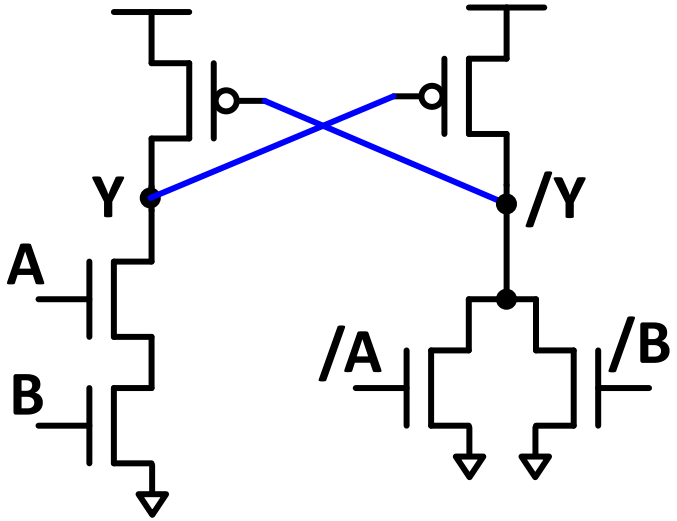
Combining Two Pseudo NMOS

- We call this type logic as **cascade voltage switch logic (CVSL)**.

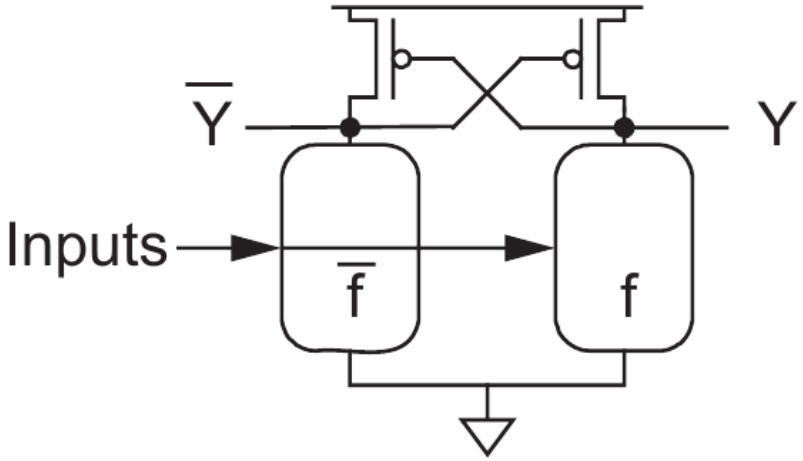


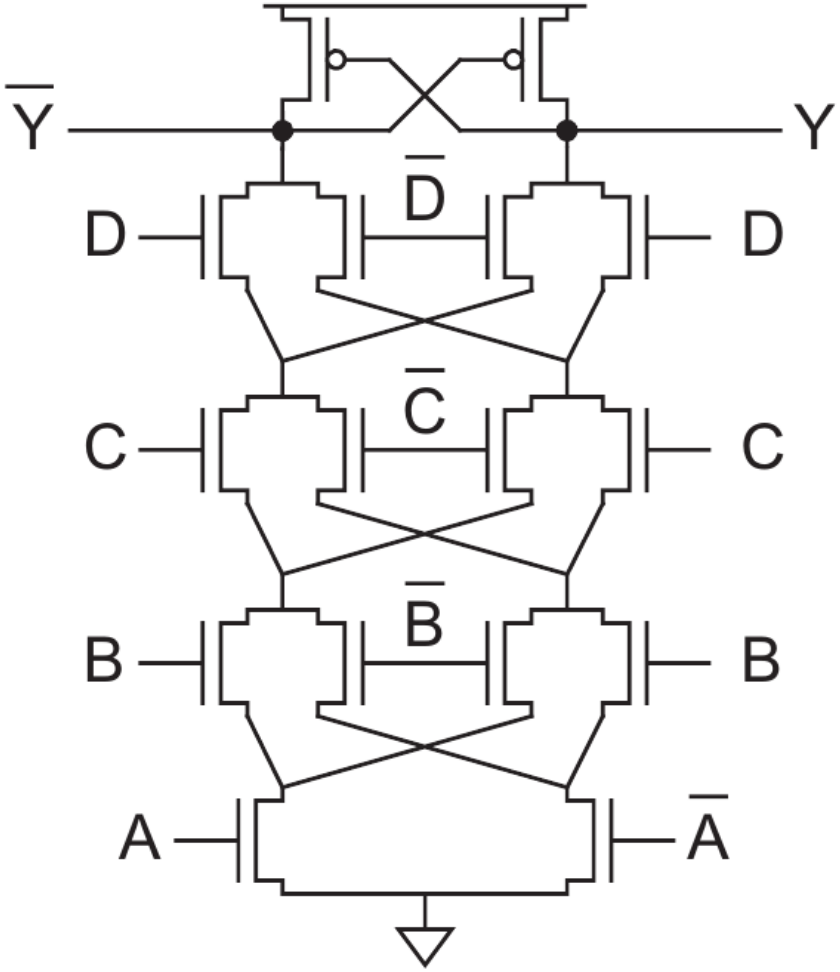
Operation

- A=1, B=1 PDN at the leftside is enabled
- Otherwise, PDN at the rightside is enabled
- Unlike pseudo-nMOS, the feedback tends to turn off the pMOS, so the outputs will settle eventually to a legal logic level.



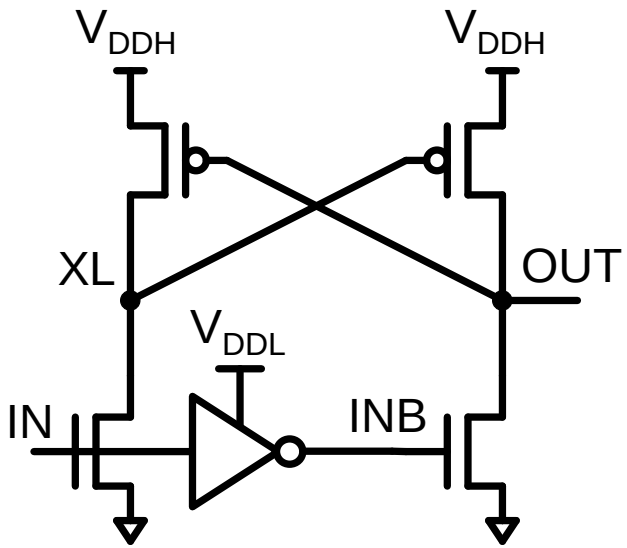
CVSL Logics



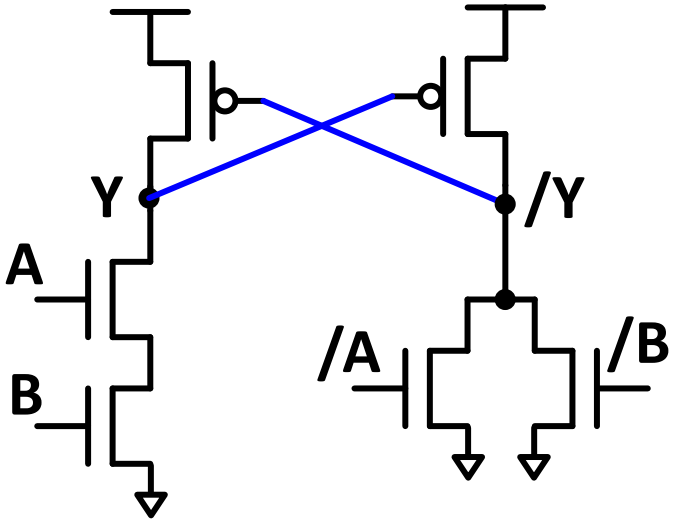


How Does Cross-coupled FET Operate?

- You also saw this structure in the level shifter
- It operates in the positive feedback manner.



Level Shifter



CVSL