Pull-Up and Pull-Down Networks

Mahit Kumar Paul
Assistant Professor, Dept. of CSE
RUET, Rajshahi-6204

mahit.cse@gmail.com
mahit@cse.ruet.ac.bd

Pull-Up and Pull-Down Networks

- A **CMOS** gate is a combination of two networks: the Pull Up Network (PUN) and the Pull Down Network (PDN).

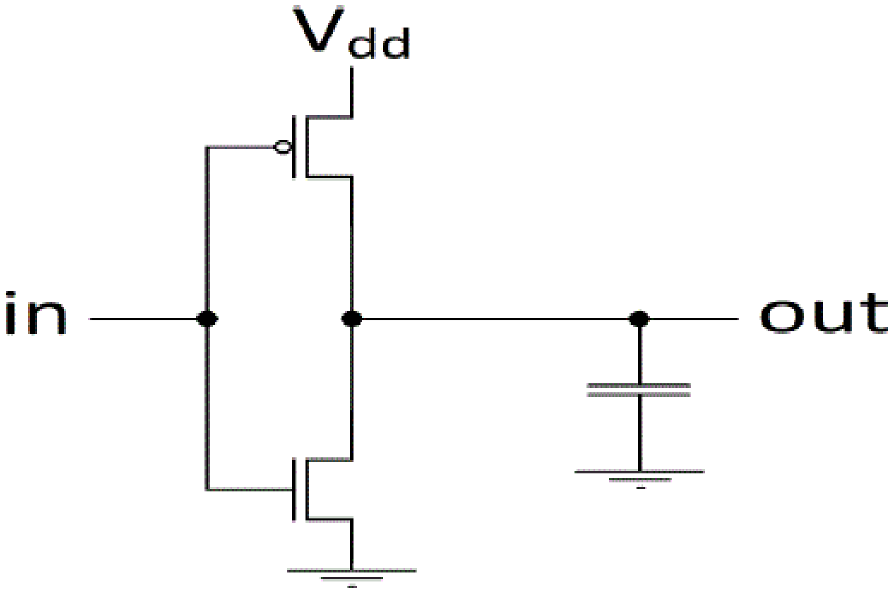
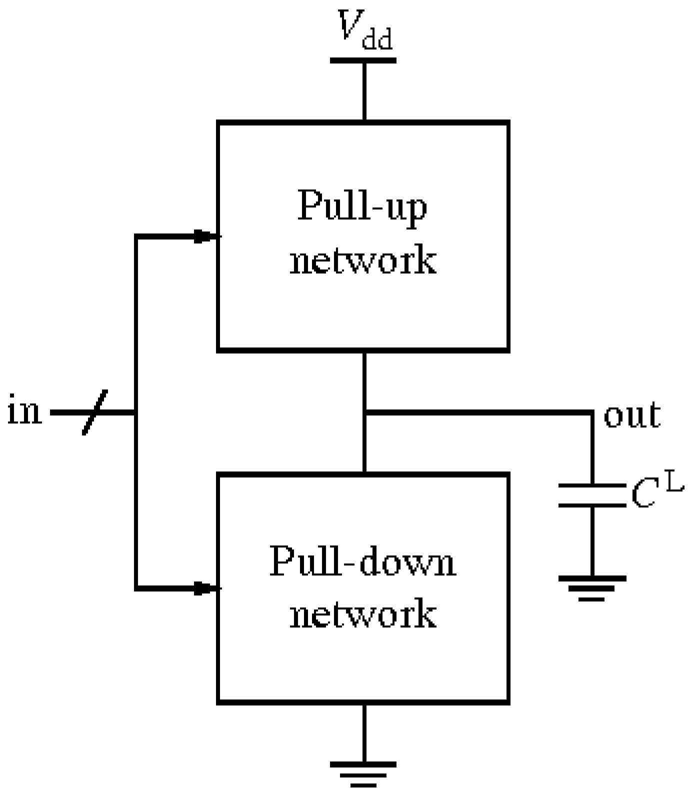


Fig1. CMOS Inverter



Pull-Up and Pull-Down Networks…

- Normally the PDN is consisting of **NMOS** devices whereas PUN is consisting of **PMOS** devices.

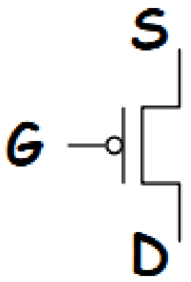
- The function of PUN is to provide a connection between VDD and Vout to pull Vout to logic '1' whereas the function of PDN is to provide connection between GND and Vout to pull Vout to logic '0'.

- Using this CMOS logic, many circuits can be developed. Such as NAND, NOR, XNOR etc.

Pull-Up and Pull-Down Networks…

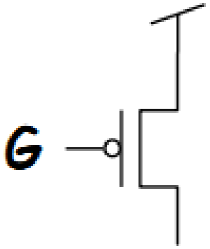
Binary logic values represented by voltages:

"High" = Supply Voltage, "Low" = Ground Voltage

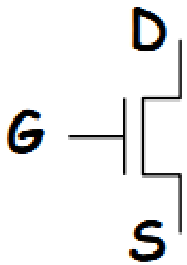


PFET connects S and D when G="low"=0V

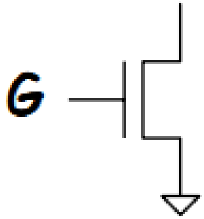
Supply Voltage = V_{DD}



PFET only good at pulling up



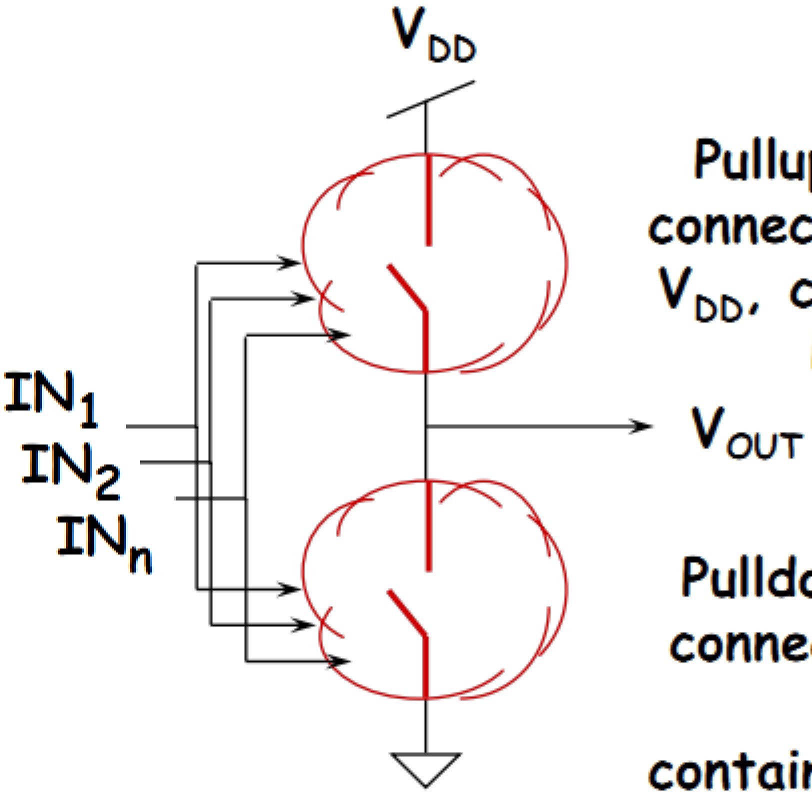
NFET connects D and S when G="high"=V_{DD}



NFET only good at pulling down

Ground = GND = 0V

Pull-Up and Pull-Down Networks…



Pullup network, connects output to V_{DD}, contains only PMOS

Pulldown network, connects output to GND, contains only NMOS

For every set of input logic values, either pullup or pulldown network makes connection to VDD or GND

- If both connected, power rails would be shorted together
- If neither connected, output would float (tristate logic)

Pull-Up and Pull-Down Networks…

Pull up network should connect output to V_{DD} when $f(x_1, x_2, \ldots) = 1$

Pull down network should connect output to GND when $\overline{f}(x_1, x_2, \ldots) = 1$

Because PMOS is conducting with low inputs, useful to write pullup as function of inverted inputs

$$p(\overline{x}1, \overline{x}2, \ldots) = f(x_1, x_2, \ldots)$$

Pull-Up Is Dual of Pull-Down Network

For NAND gate, $f=(\overline{A.B})$

Pulldown $\overline{f} = A.B$

Pullup $p = f = \overline{A.B}$

$= \overline{A}+\overline{B}$

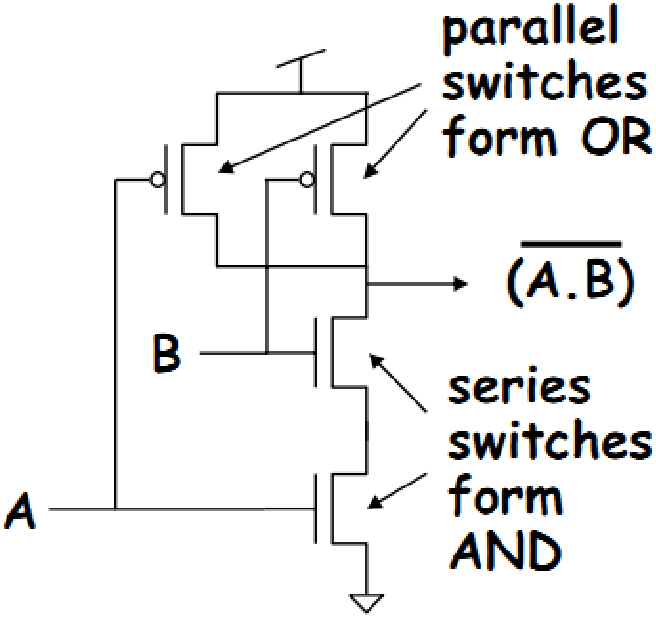
(De Morgan's Laws)

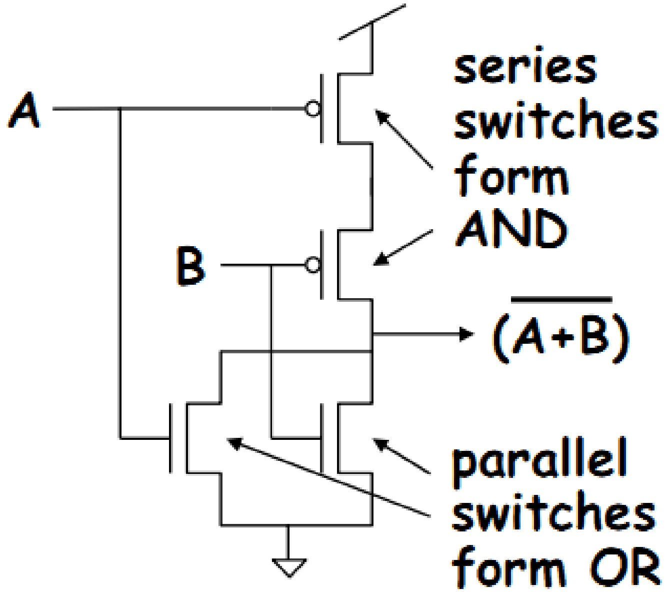
For NOR gate, $f=(\overline{A+B})$

Pulldown $\overline{f} = A+B$

Pullup $p = f = \overline{A+B}$

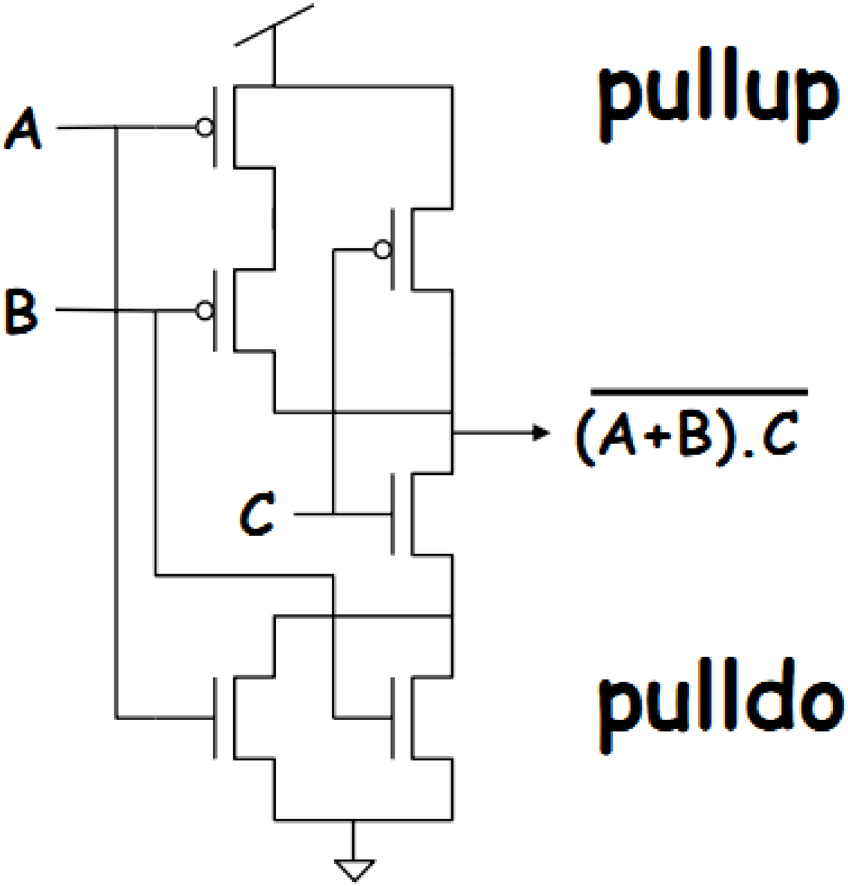
$= \overline{A}.\overline{B}$





More Examples

$$f = \overline{(A+B).C}$$



$$\text{pullup } p = \overline{(A+B).C}$$

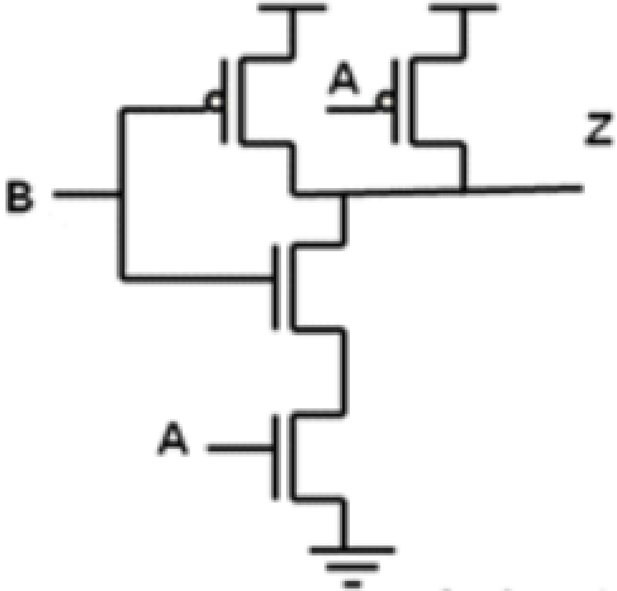
$$= \overline{(A+B)}+\overline{C}$$

$$= (\overline{A}.\overline{B})+\overline{C}$$

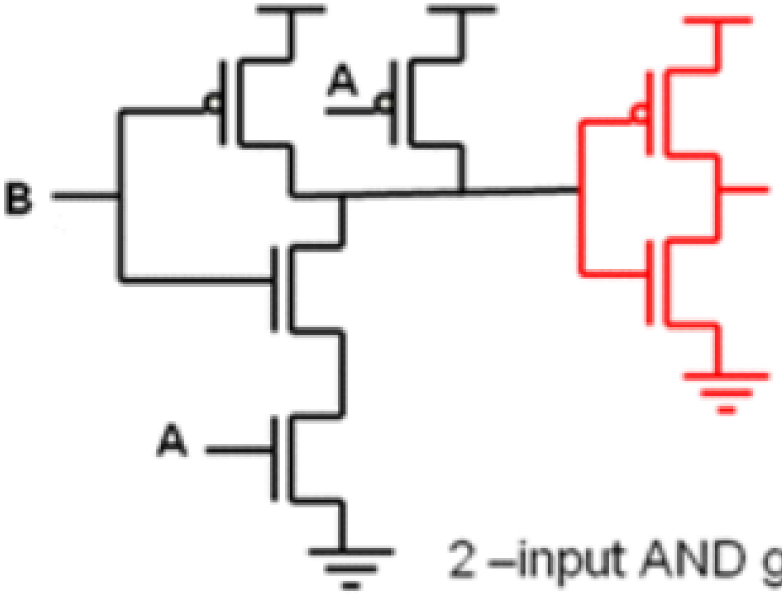
$$\text{pulldown } \overline{f} = (A+B).C$$

More Examples…

CMOS always ought to provide INVERTED outputs like Inverter, NAND, NOR, XNOR etc. Sometimes a non-inverting function is required, in which case it's just as easy to implement it with a final inverter or with a non-inverting function like AND, OR as also shown in the below Fig



2 –input NAND gate

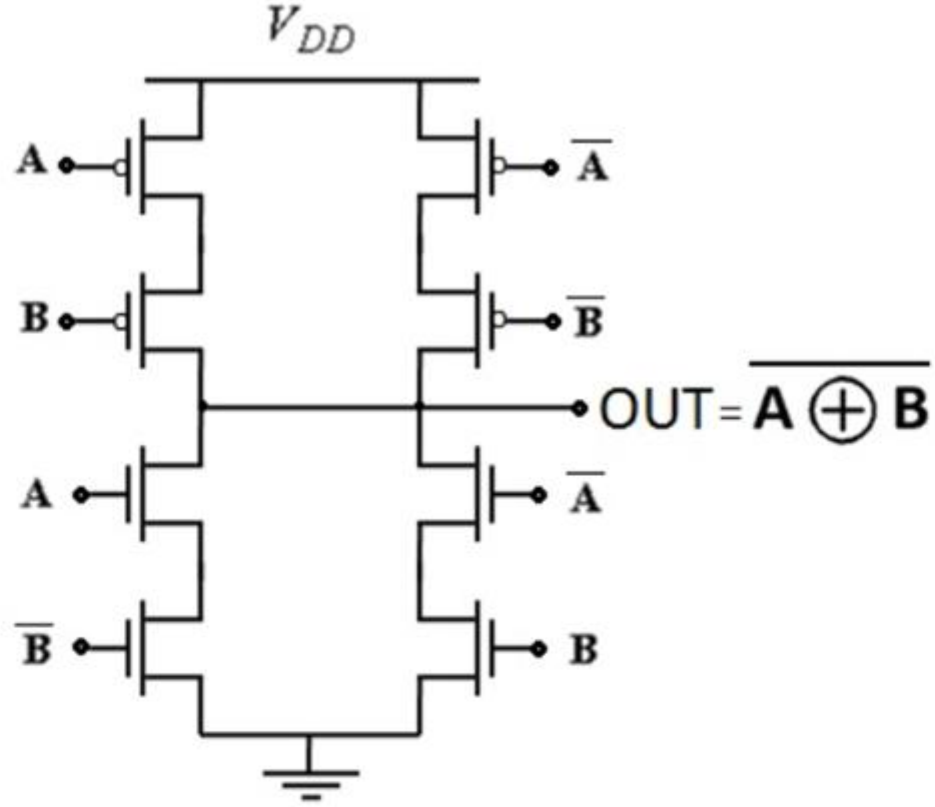


2 –input AND gate

*https://www.google.com/search?q=half+adder+using+cmos&rlz=1C1RLNS_enBD805BD805&source=lnms&tbm=isch&sa=X&ved=0ahUKEwik2dX-oKrhAhWKtY8KHRM1AFIQ_AUIDigB&biw=1280&bih=610#imgrc=7eNPtYD3Vnd11M:

More Examples…

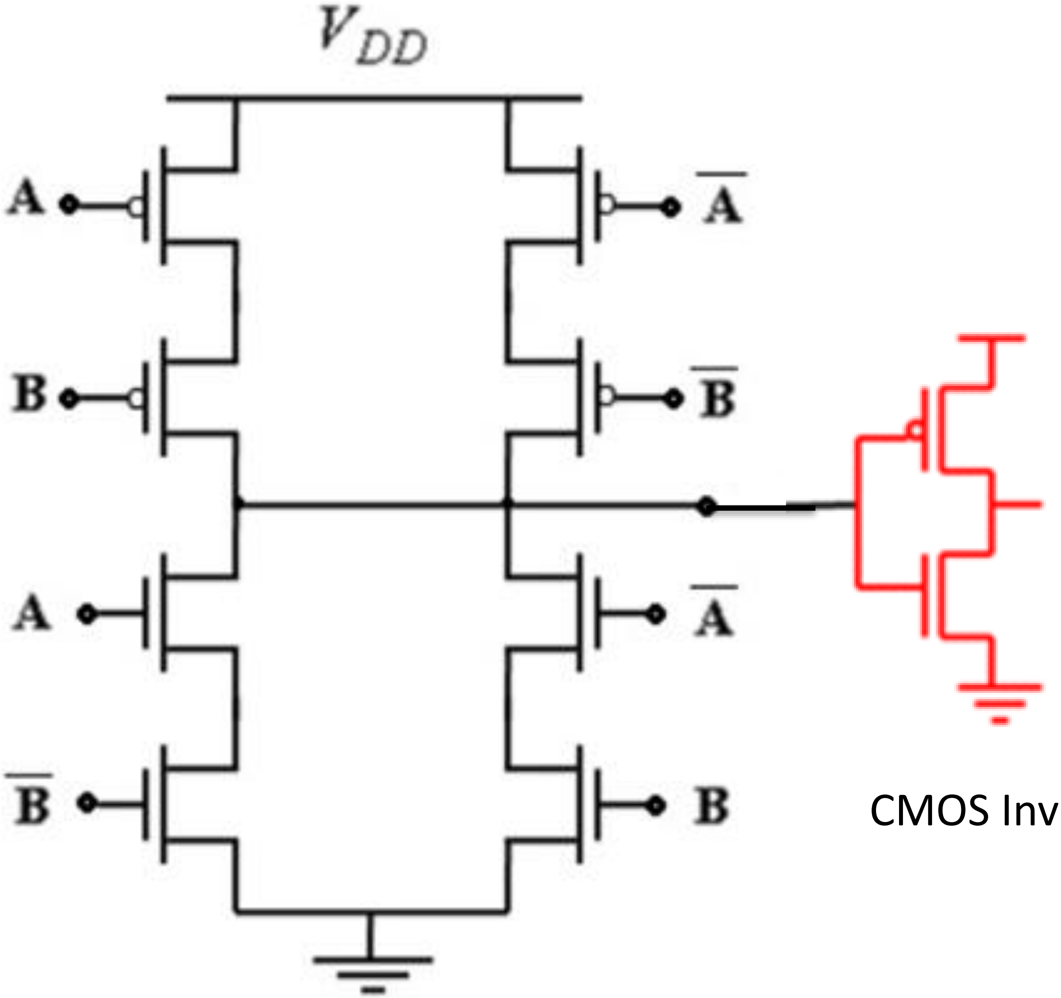
$f = (A'B + AB')'$



2 input XNOR gate

More Examples…

$f = A'B + AB'$



2 input XNOR gate

CMOS Inverter

References

[1] https://www.electronics-tutorial.net/Digital-CMOS-Design/CMOS-Logic-Gates/Pull-up-and-Pull-Down-Networks/

[2]https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-884-complex-digital-systems-spring-2005/lecture-notes/l03_cmos_gates.pdf

Thank You