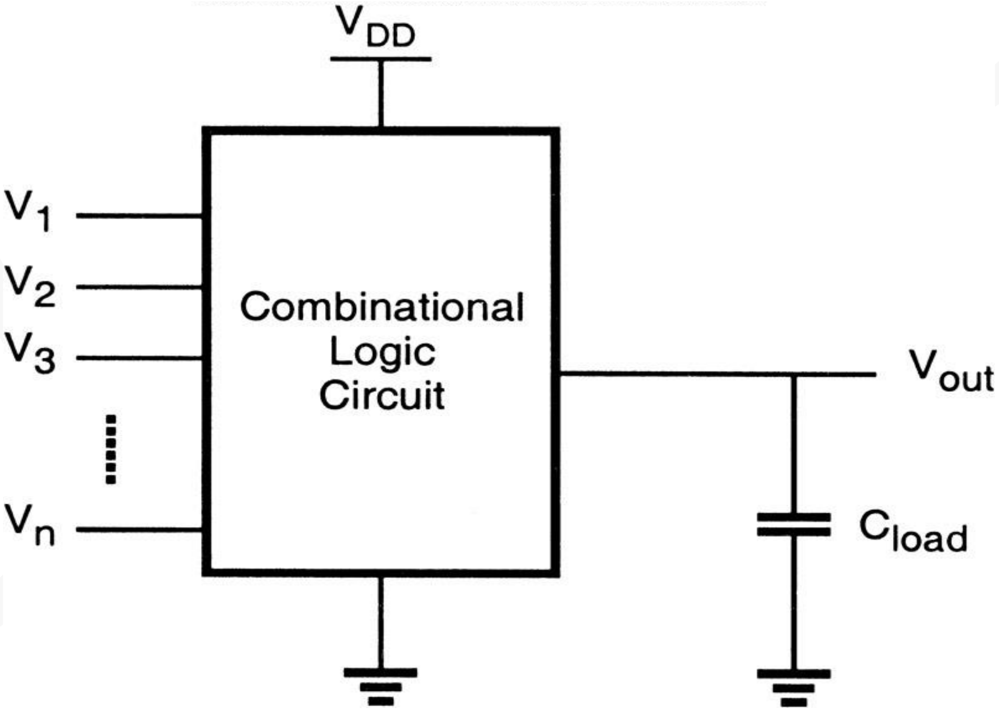
University of Southern California

Viterbi School of Engineering

MOS Logic Design Basics

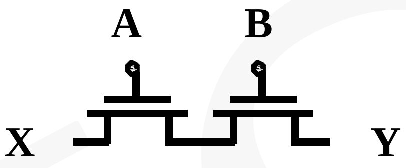
Combinational Logic

- A combinational logic cell, logic circuit or gate is generally a multiple input, single output system that performs a Boolean function
- In the positive logic convention, logic 1 is shown by high voltage V_{DD} and logic 0 by low voltage of zero

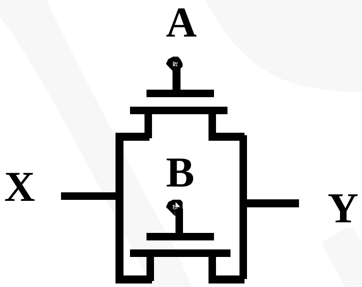


NMOS Transistors in Series/Parallel Connection

- Transistors can be thought as a switch controlled by its gate signal
- NMOS switch closes when switch control input is high



Y = X if A and B

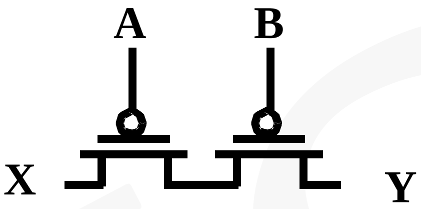


Y = X if A OR B

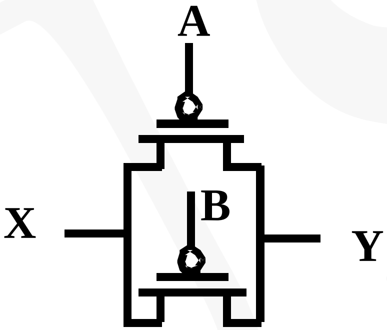
NMOS Transistors pass a “strong” 0 but a “weak” 1

PMOS Transistors in Series/Parallel Connection

- PMOS switch closes when switch control input is low



Y = X if $\bar{A}$ AND $\bar{B}$ = $\overline{A + B}$

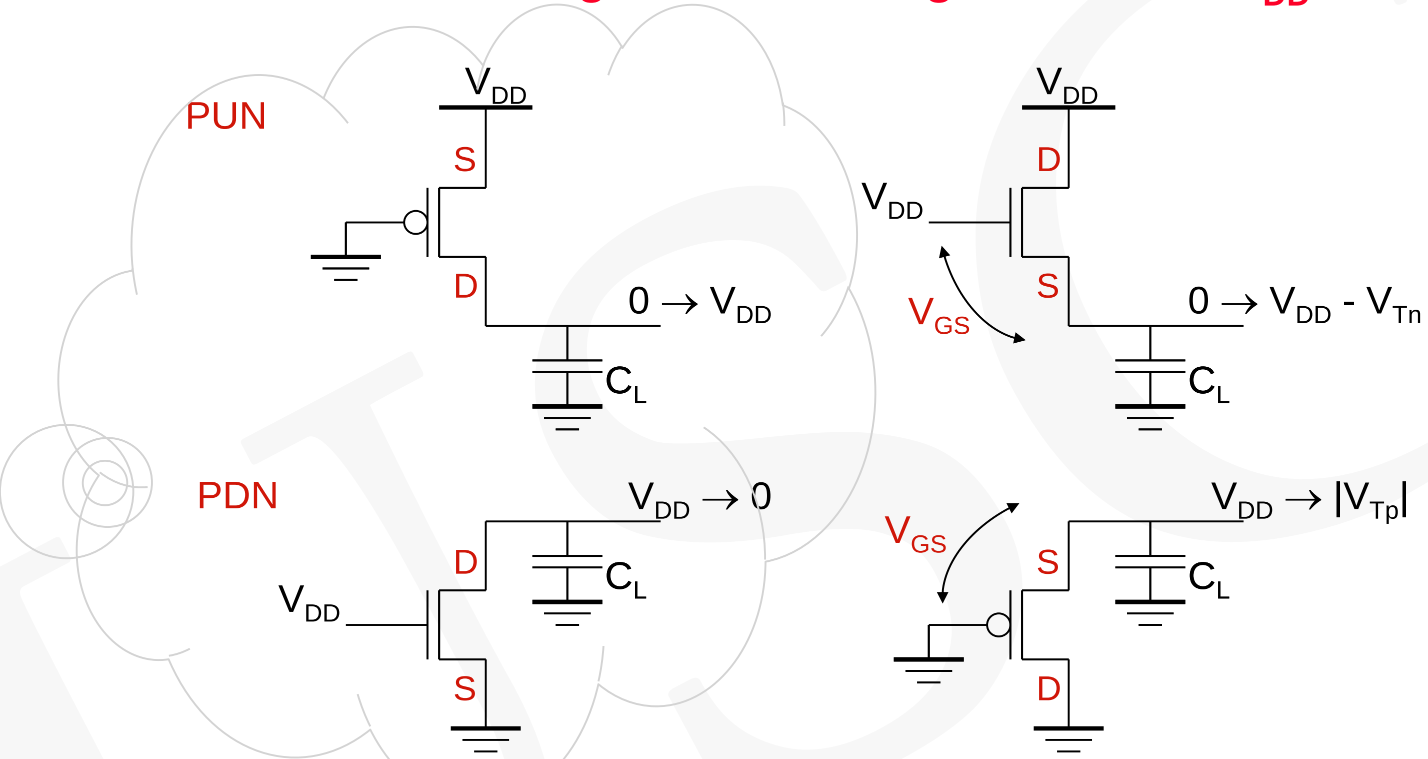


Y = X if $\bar{A}$ OR $\bar{B}$ = $\overline{AB}$

PMOS Transistors pass a “strong” 1 but a “weak” 0

Threshold Drops

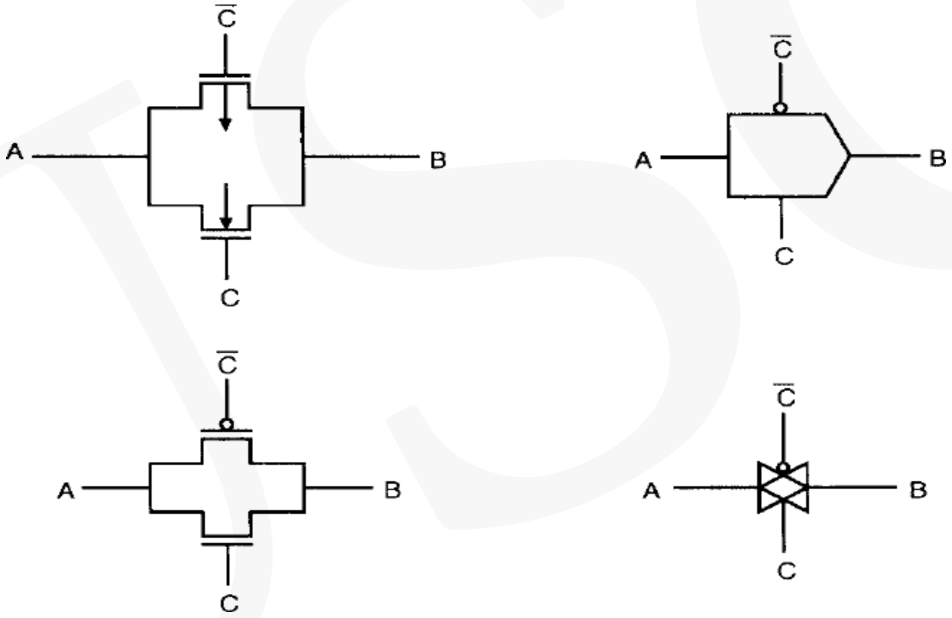
- NMOS: A device which is good in shorting its drain to the ground
- PMOS: A device which is good in shorting its drain to V_{DD}



- NMOS passes a strong zero and a weak one
- PMOS passes a strong one and a weak zero
- Note: NMOS gate = 0, PMOS gate = 1 are in high Impedance

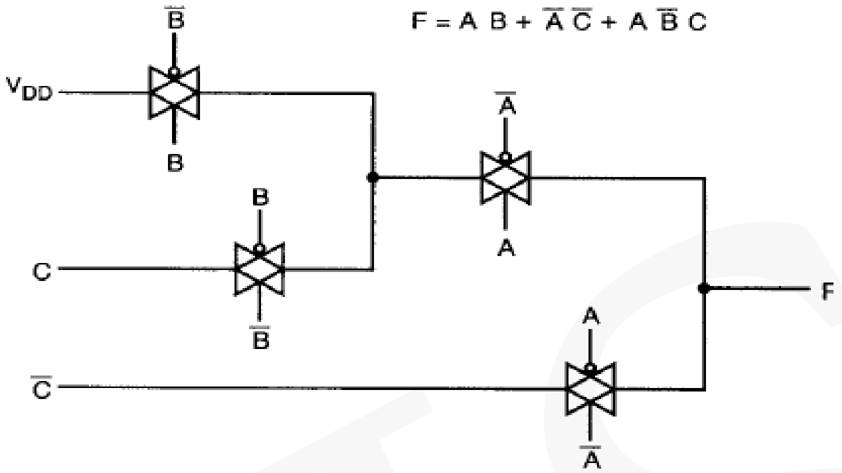
CMOS Transmission Gates (Pass Gates)

- The CMOS transmission gate (TG) consists of one NMOS and one PMOS transistor, connected in parallel
- The CMOS TG operates as a bidirectional switch between nodes A and B, which is controlled by signal C

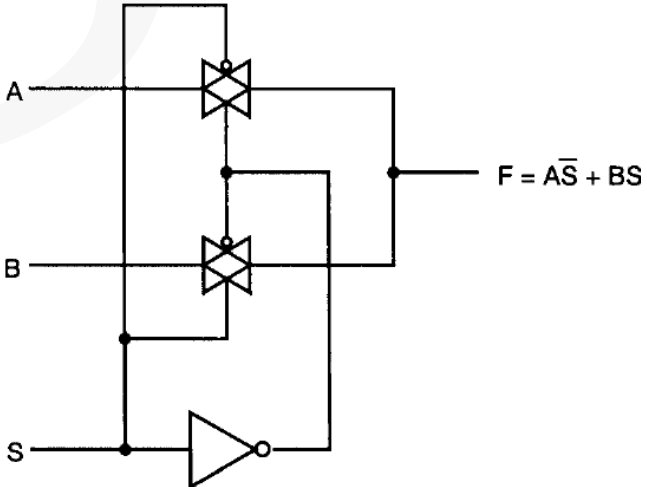


Four different representations of the CMOS transmission gate

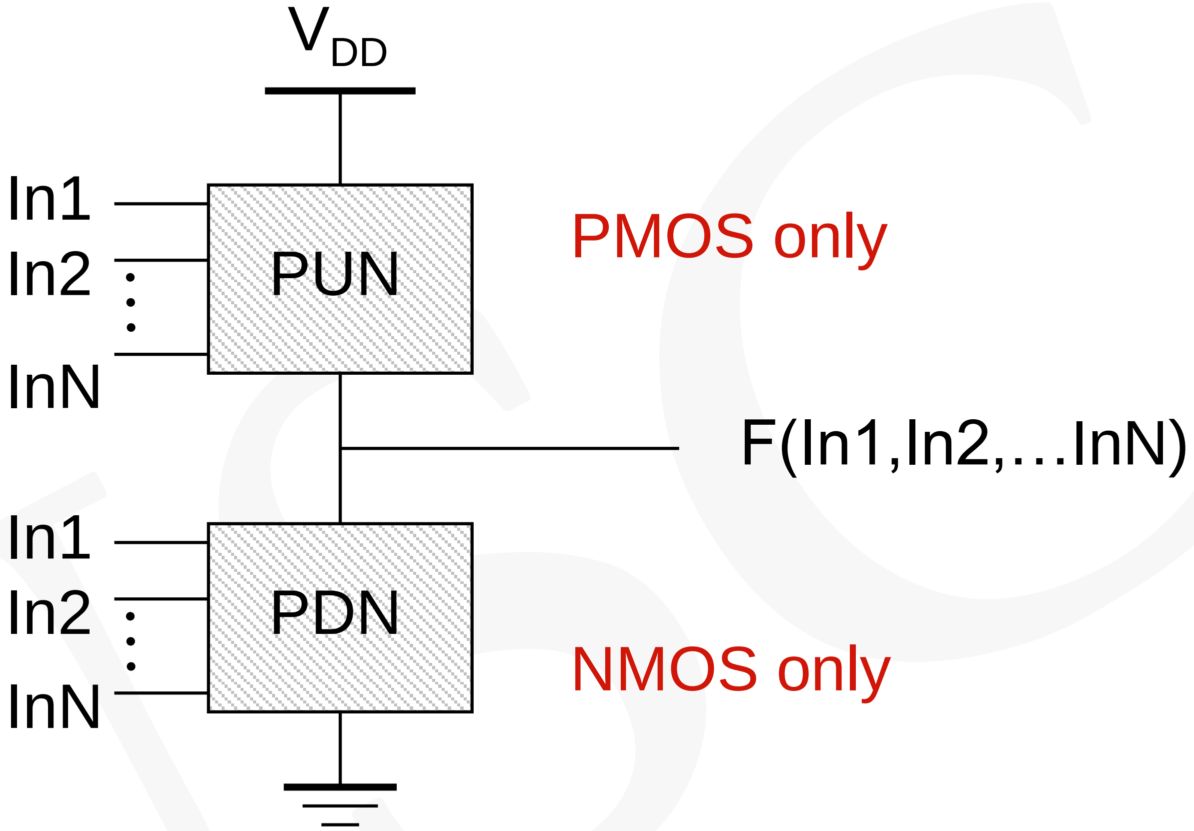
CMOS Transmission Gates – Examples



Two-Input Multiplexer



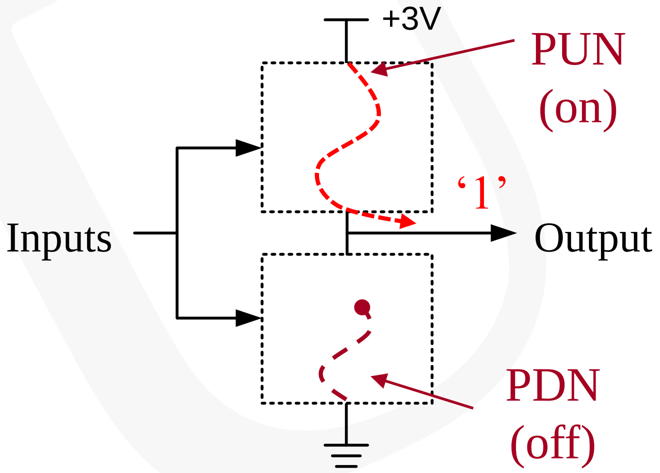
Complementary MOS (CMOS)

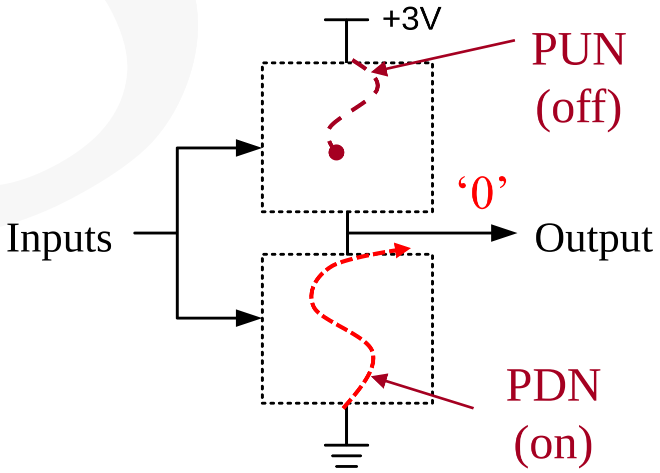


- PUN and PDN are dual Logic Networks

Pull-up and Pull-down Networks

- For the gate to output a ‘1’
  - Some path of PMOS transistors from V_{DD} to output will be on
  - We call the PMOS transistors the **Pull-Up Network (PUN)**
- For the gate to output a ‘0’
  - Some path of NMOS transistors from GND to output will be on
  - We call the NMOS transistors the **Pull-Down Network (PDN)**

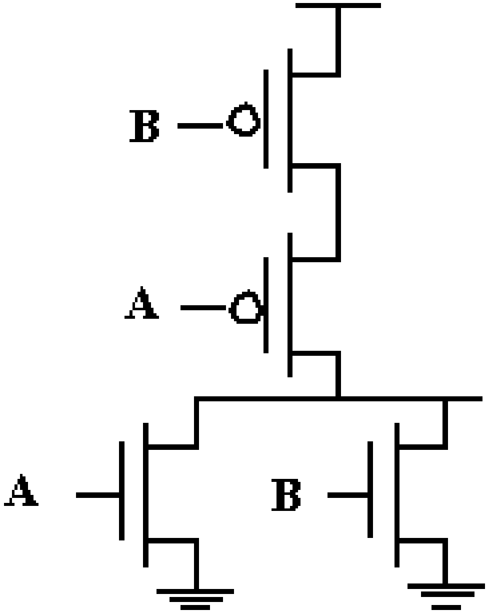




Example Gate: NOR

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

Truth Table of a 2 input NOR gate

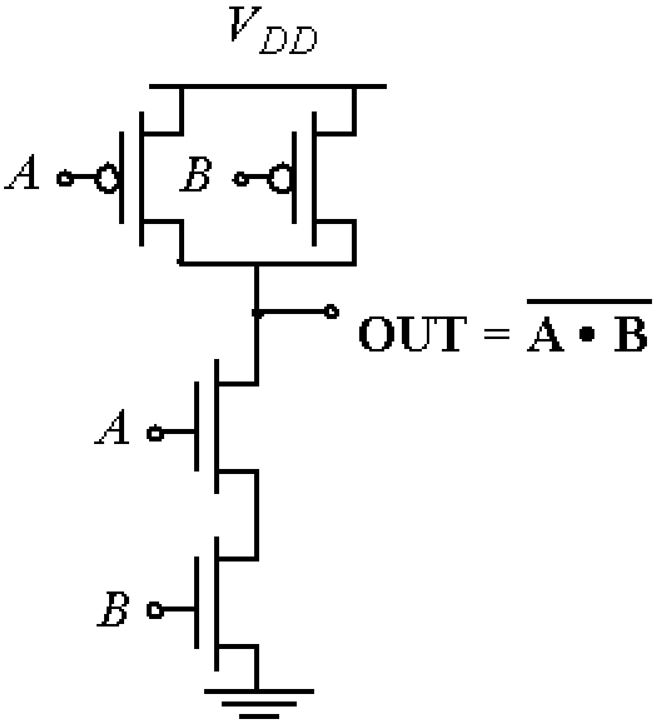


$$\mathrm{OUT} = \overline{A + B}$$

Example Gate: NAND

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

Truth Table of a 2 input NAND gate

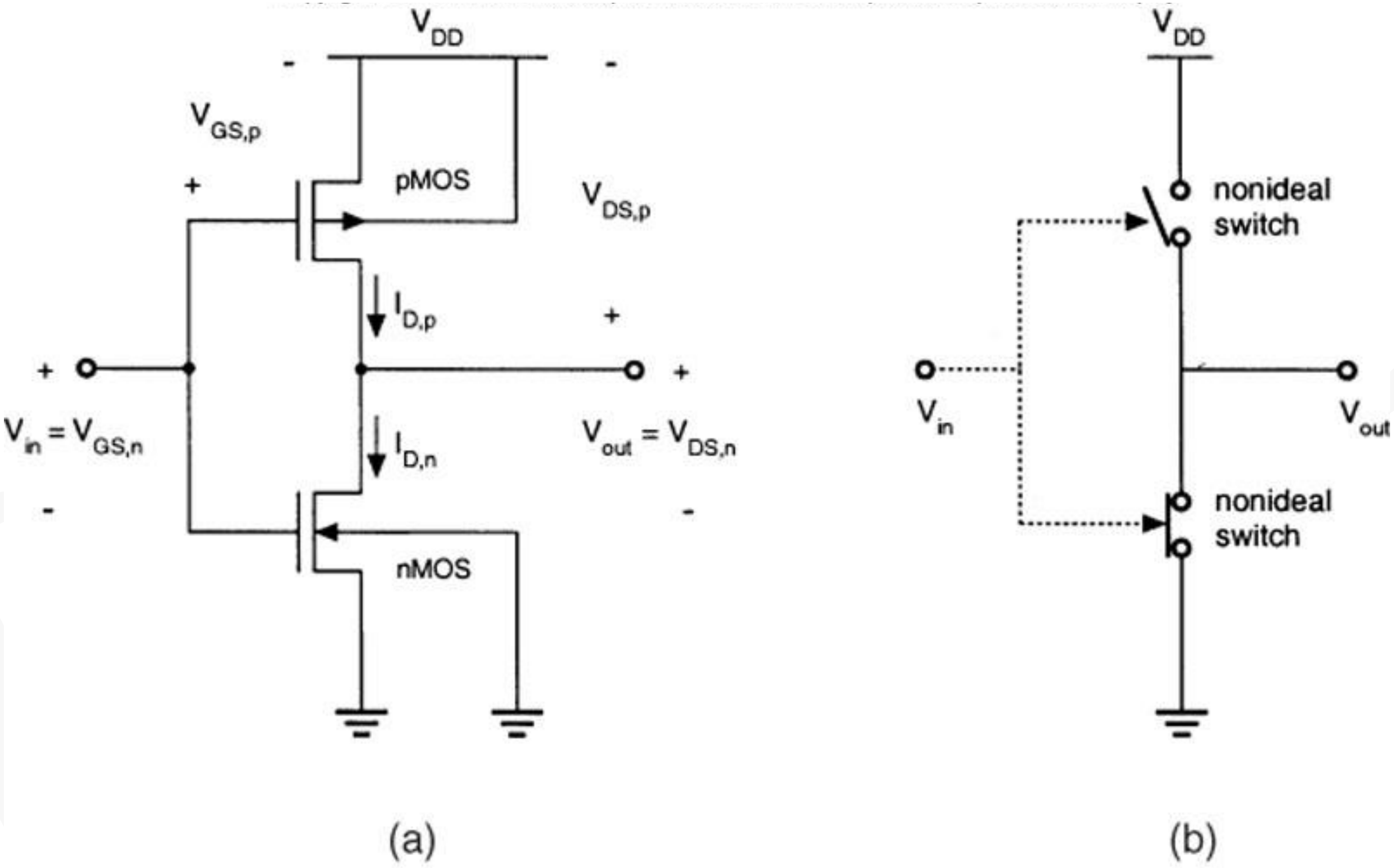


PDN: G = A B $\Rightarrow$ Conduction to GND

PUN: F= $\bar{A} + \bar{B} = \overline{AB} \Rightarrow$ Conduction to V_{DD}

$$\overline{G(In_1, In_2, In_3, \ldots)} \equiv F(\overline{In_1}, \overline{In_2}, \overline{In_3}, \ldots)$$

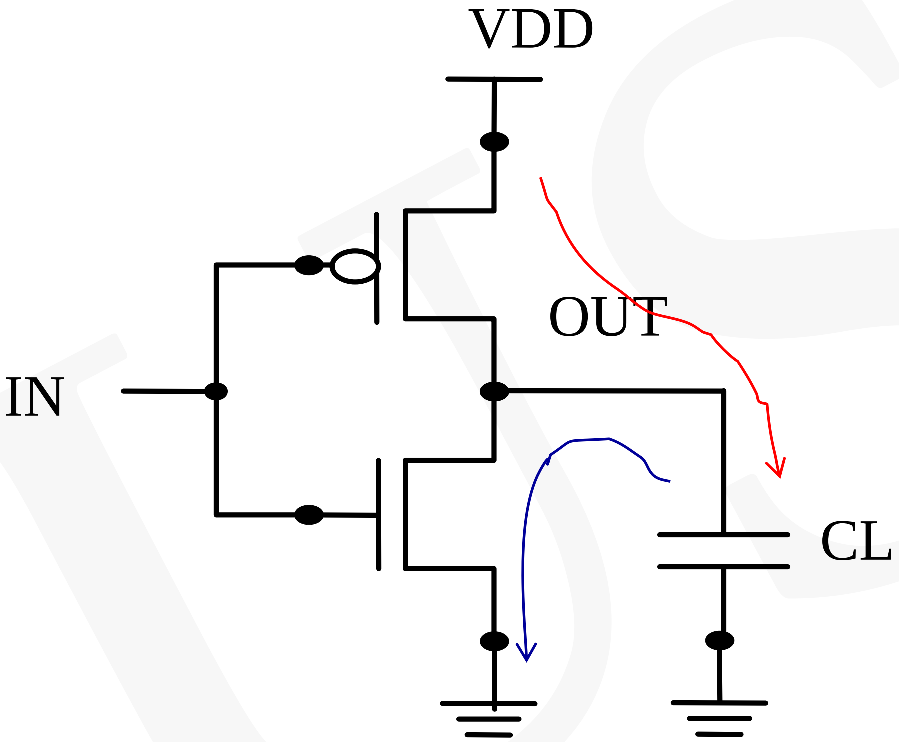
Sizing Background – Inverter



Sizing – PMOS is Slower than NMOS!

$$I_{DS-pMOS} = \frac{\mu_p C_{ox}}{2} \frac{W_p}{L}\left[2(V_{GS} - V_{T0p})V_{DS} - V_{DS}^2\right]$$

$$I_{DS-nMOS} = \frac{\mu_n C_{ox}}{2} \frac{W_n}{L}\left[2(V_{GS} - V_{T0n})V_{DS} - V_{DS}^2\right]$$



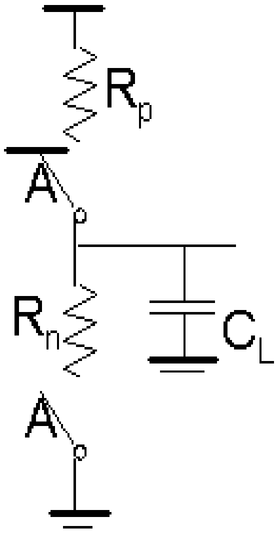
Notes:

- $\frac{1}{R} \propto \frac{\mu W}{L}$
- $\mu_p \approx \frac{\mu_n}{2.5}$
- Sizing required to balance fall and rise time delays
- For all transistors channel is min-sized to L

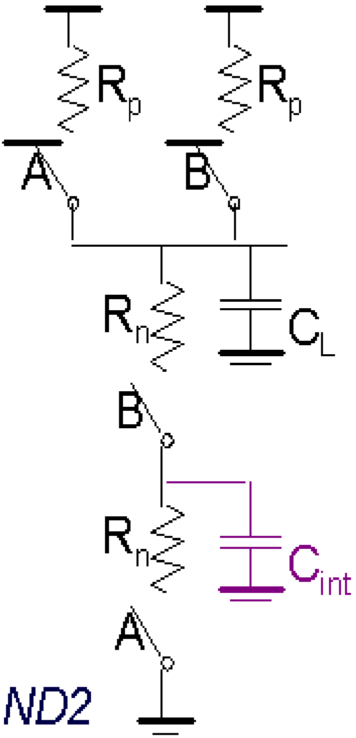
Switch Delay Model

- W_p can be considered 2 to 3 times as big as W_n
- For example for inverter and assuming $\mu_p \approx \frac{\mu_n}{2}$

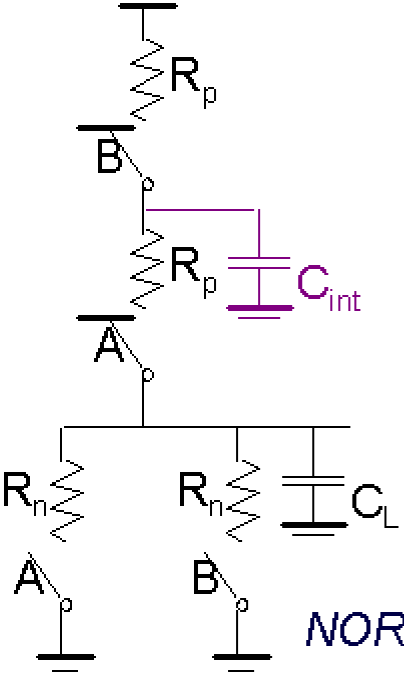
$$\frac{1}{R} \propto \frac{\mu W}{L} \Rightarrow \frac{R_p}{R_n} \propto \frac{\mu_n W_n}{\mu_p W_p} = \frac{2W_n}{W_p} \Rightarrow W_p = 2W_n$$



INV



NAND2



NOR2

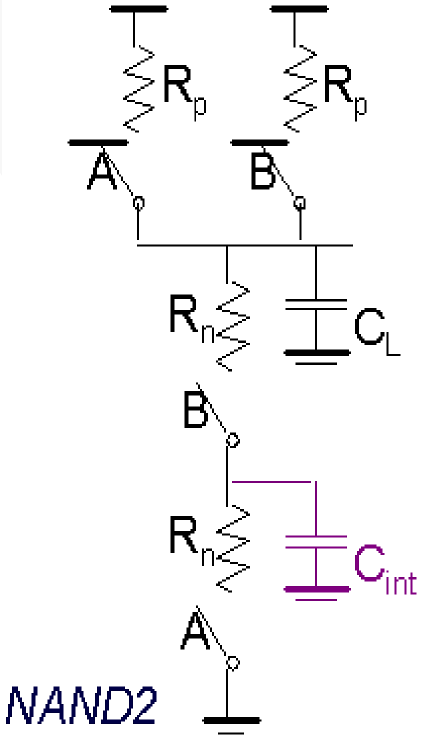
Input Pattern Effect on Delay

- Delay is dependent on the pattern of inputs
  - Low to high transition
    - both inputs go low
      - delay is $R_p/2\ C_L$
  - one input goes low
      - delay is $R_p\ C_L$

$\Rightarrow$ Worst case: $R_p\ C_L$

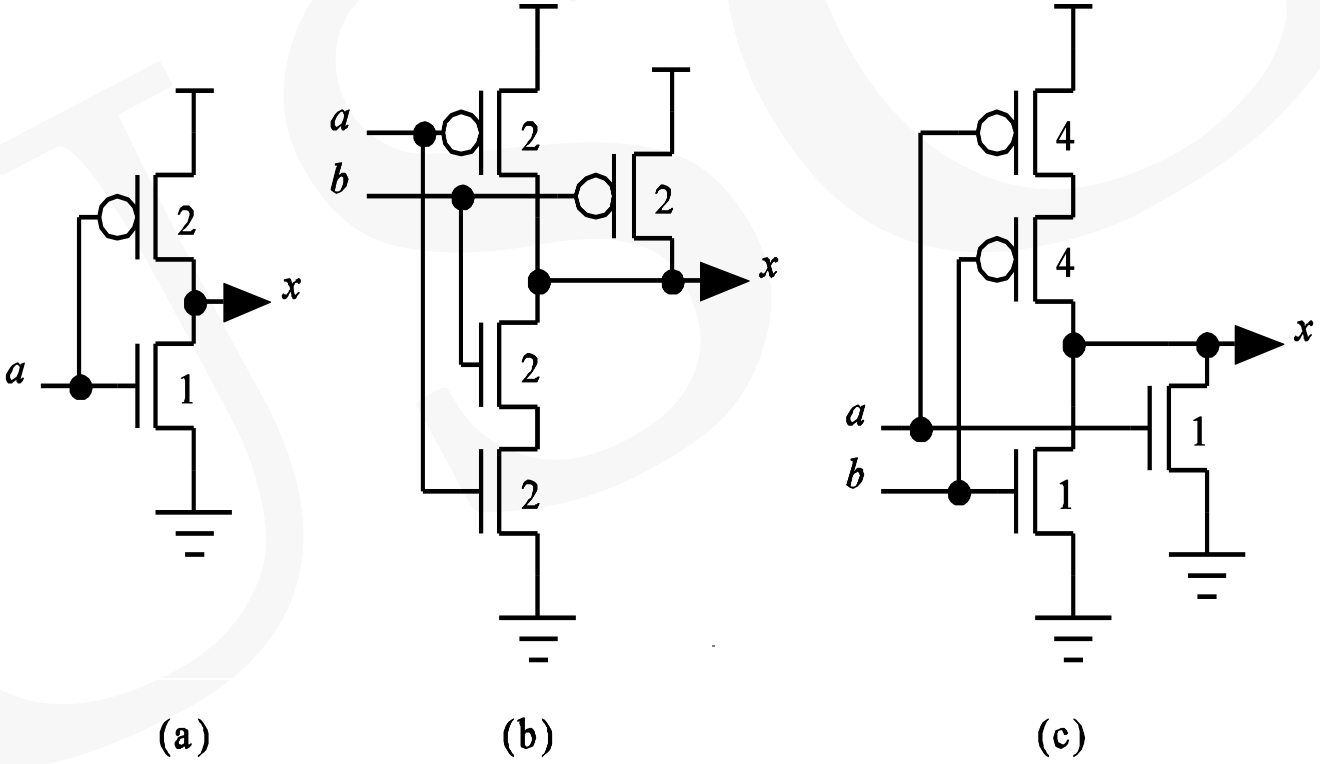
- High to Low transition
  - both inputs go high
    - delay is $2R_n\ C_L$

$R_p=2R_n \rightarrow W_p= W_n$

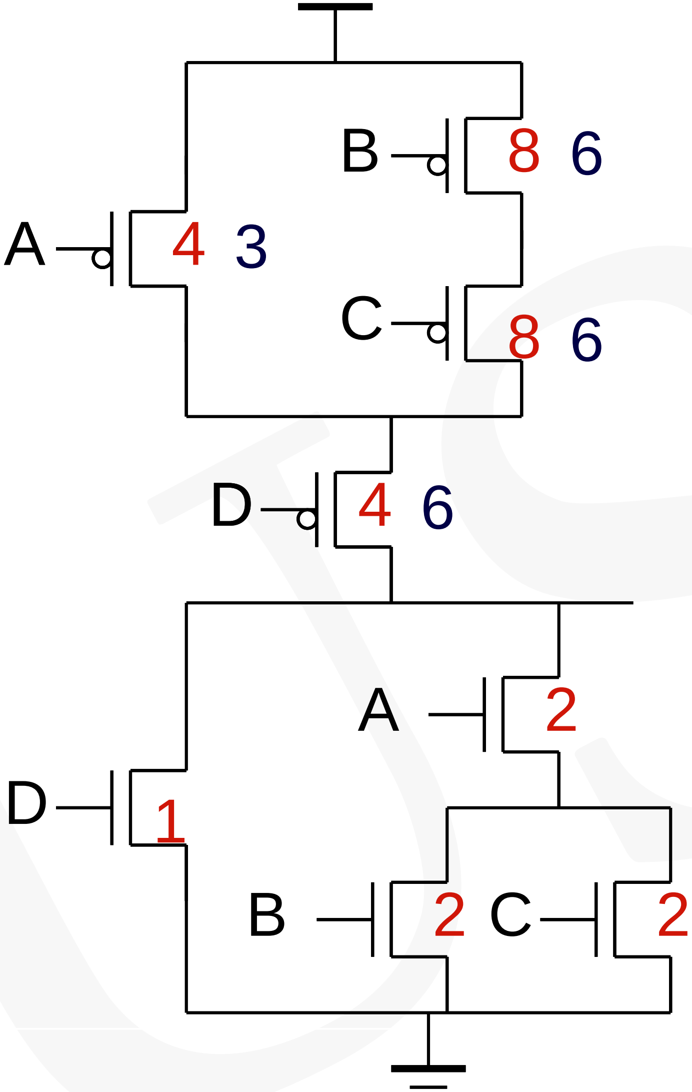


Sizing – Simple CMOS Gates

- Sizing of complex cells to have the same output current (hence same delay) of an inverter with a certain size:
- The ratio of the $\{W/L\}_{PUN}$/ $\{W/L\}_{PDN}$ should be two (or higher) to make up for slow PMOS



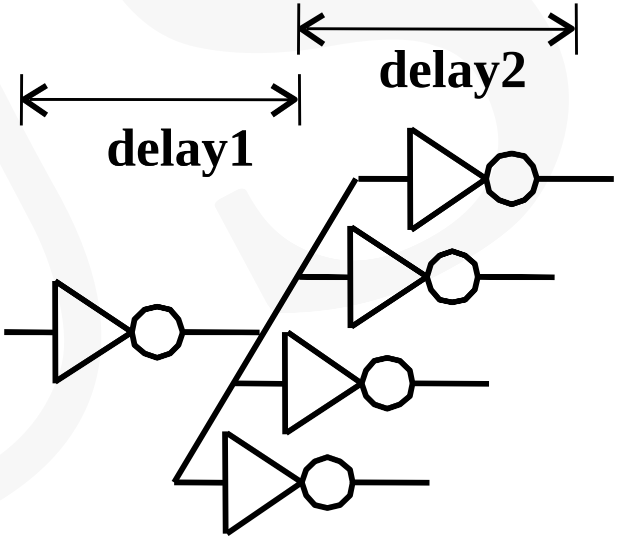
Sizing – Complex CMOS Gates



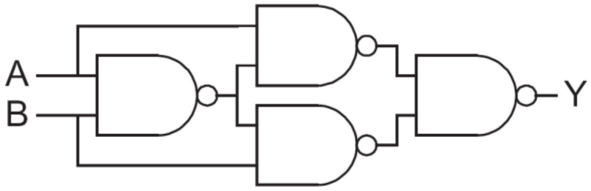
$$\text{OUT} = \overline{D + A \bullet (B + C)}$$

Stage Sizing

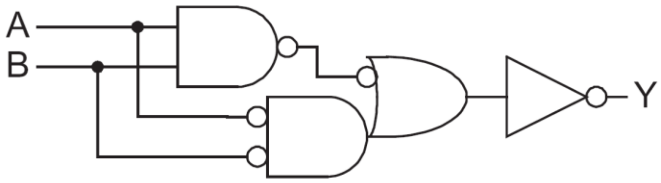
- To drive a capacitive load logical effort (covered later in the course) can be used to calculate the best stage sizing for the gates in each stage
- Number of stages also important (Logic effort can be used to determine the optimal number)



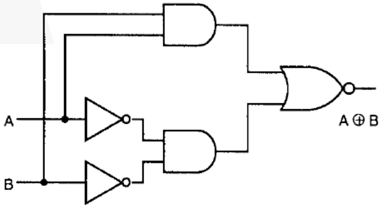
XOR Design

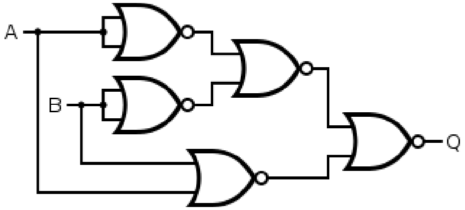


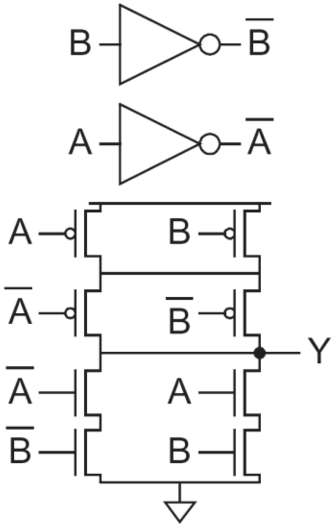
(a)



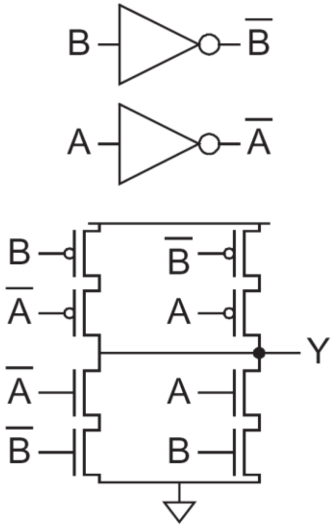
(b)



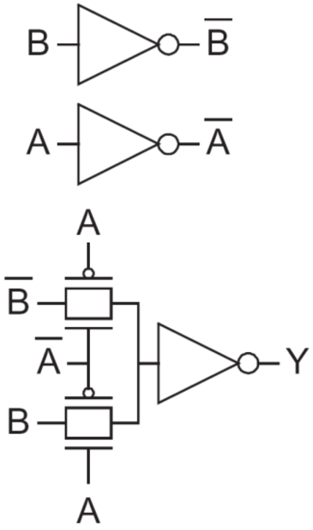




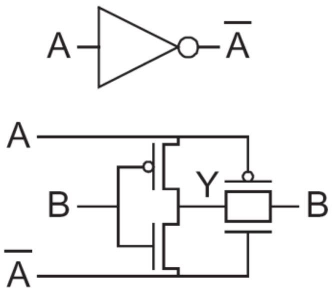
(c)



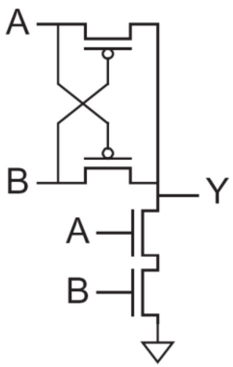
(d)



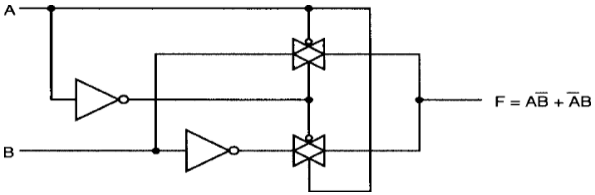
(e)



(f)



(g)

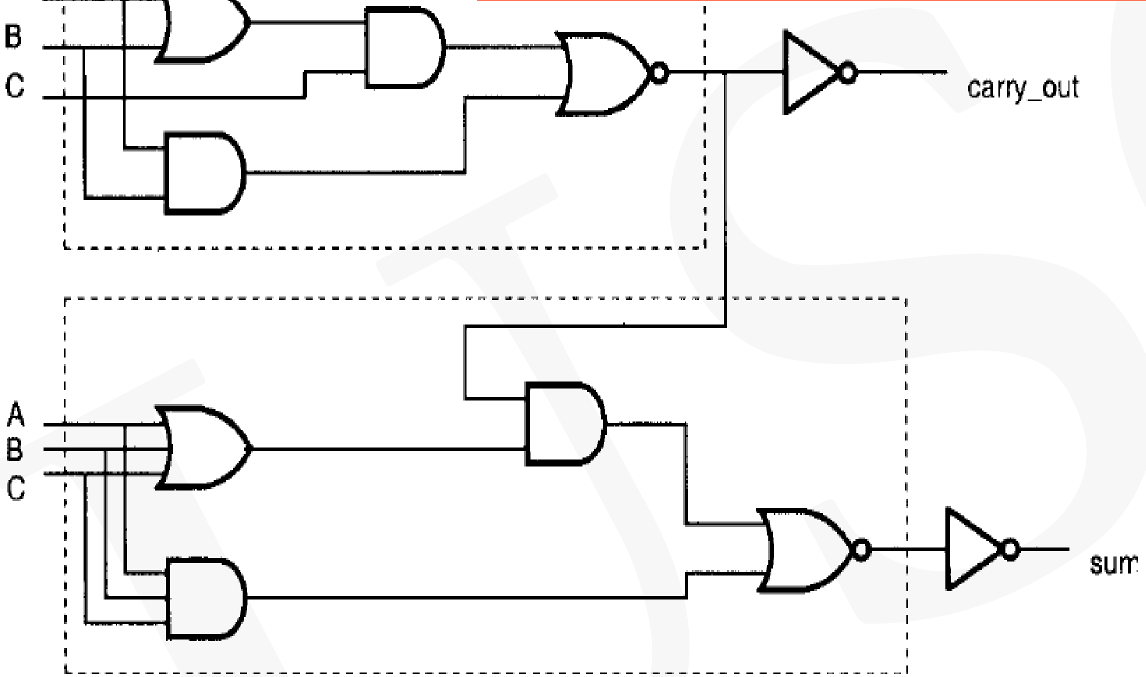


CMOS Full-Adder Circuit

- The sum and carry_out signals of the full adder are defined as:

$$sum = A \oplus B \oplus C = ABC + A\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}\overline{B}C$$

$$c_{out} = AB + AC + BC$$



Note that we use the carry_out signal to generate the sum output:

$$sum = ABC + (A + B + C)\overline{c_{out}}$$

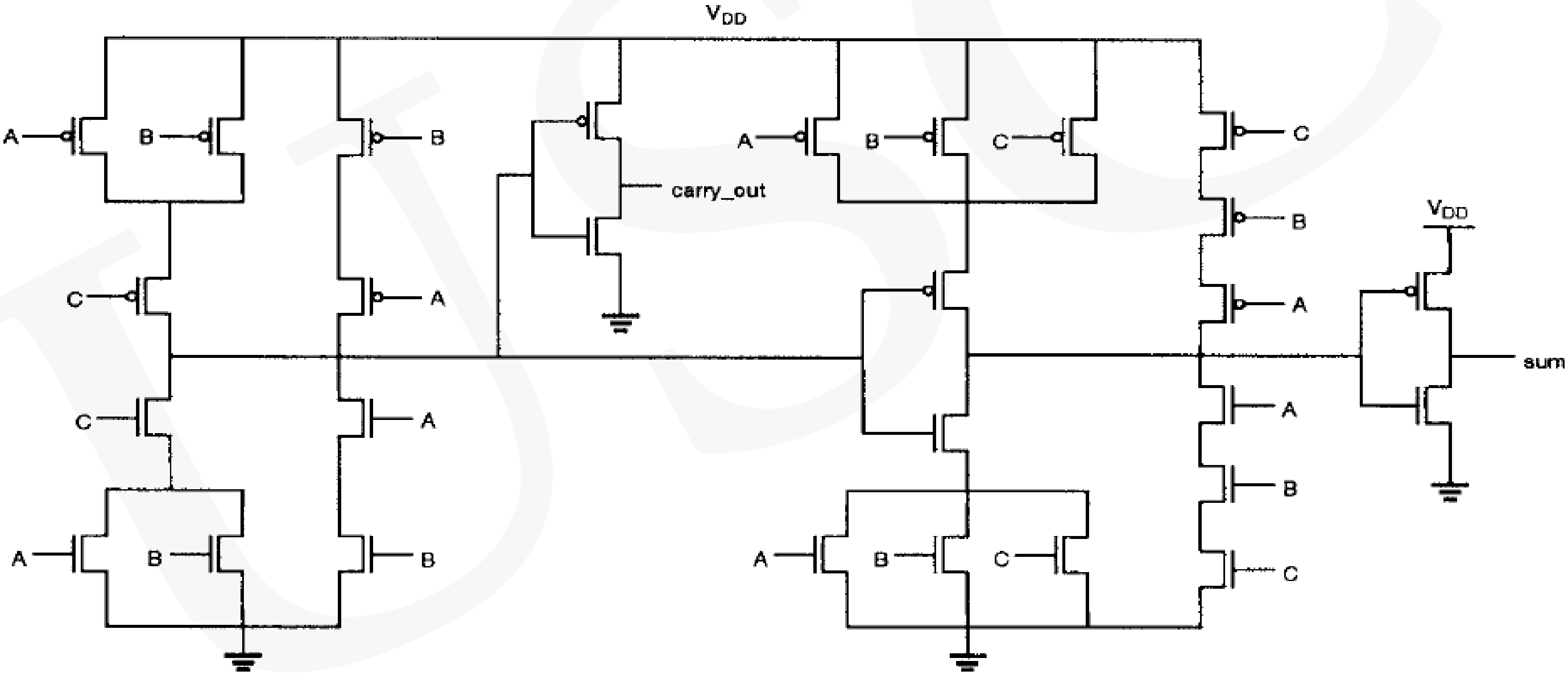
$$\overline{c_{out}} = \overline{A}\overline{B} + \overline{A}\overline{C} + \overline{B}\overline{C}$$

CMOS Full-Adder Circuit (Cont.)

$$sum = A \oplus B \oplus C = ABC + A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C$$

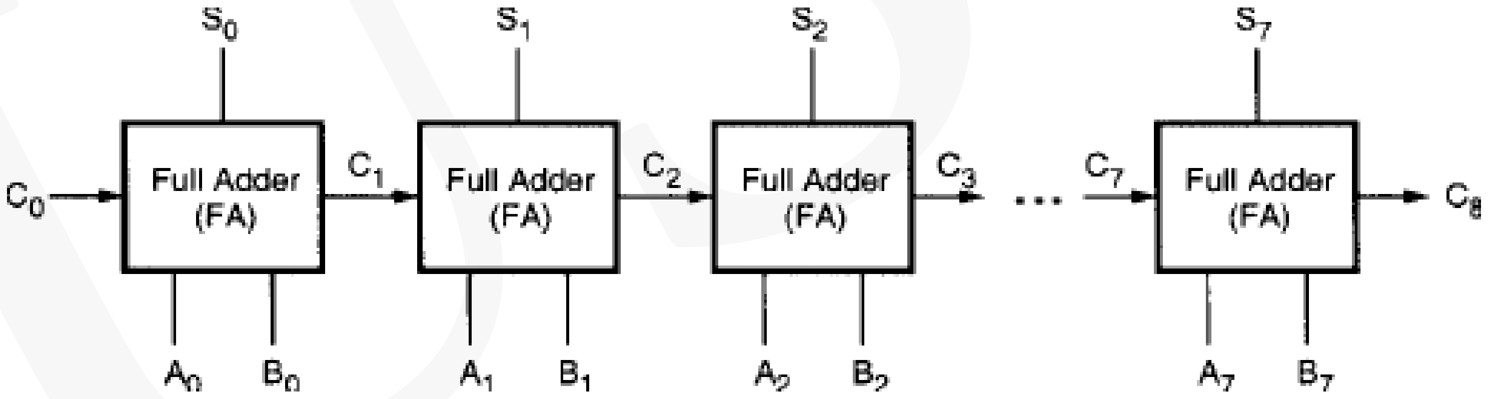
$$c_{out} = AB + AC + BC$$

- **sum and carry_out circuit for a one bit full-adder:**



Ripple Carry Adder (RCA)

- The simplest full-adder circuit can be constructed by a cascade-connection of full adders, where each adder stage performs a two-bit addition, produces the corresponding sum bit, and passes the carry output on to the next state
- The overall speed of RCA is limited by the delay of the carry bits rippling thru the carry chain
- The path from carry_in (C_0) to S_7 is the critical path. The second slowest path is the one from C_0 to the last carry_out (C_8)

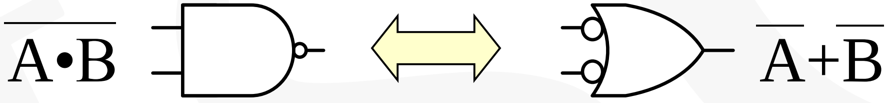


Block diagram of an 8-bit RCA chain consisting of full adders

Logic Design Basics: DeMorgan's Theorem

- Inverting output of an AND gate = inverting inputs of an OR gate
- Inverting output of an OR gate = inverting inputs of an AND gate
- A function's inverse is equivalent to inverting all the inputs and changing AND to OR and vice versa

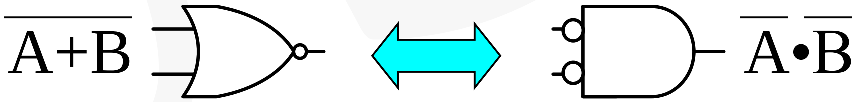
A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0



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

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0



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

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

DeMorgan's Theorem (Cont.)

- DeMorgan is useful to write the dual of the function (useful to design PUN from PDN, because PUN is the dual of PDN)

$$F = (\overline{X}+Y) + \overline{Z} \bullet (Y+W)$$

$$\overline{F} = \overline{(\overline{X}+Y) + \overline{Z} \bullet (Y+W)}$$

$$\overline{F} = \overline{(\overline{X}+Y)} \bullet \overline{(\overline{Z} \bullet (Y+W))}$$

$$\overline{F} = (\overline{\overline{X}} \bullet \overline{Y}) \bullet (\overline{\overline{Z}} + \overline{(Y+W)})$$

$$\overline{F} = (X \bullet \overline{Y}) \bullet (Z + (\overline{Y} \bullet \overline{W}))$$