

# EECS151 : Introduction to Digital Design and ICs

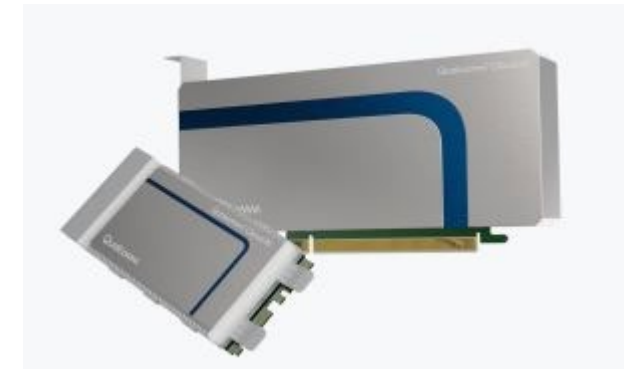
## Lecture 13 – CMOS Logic

**Bora Nikolić**



### Qualcomm Takes on Nvidia for MLPerf Inference Title

October 1, 2021, EETimes, Sally Ward-Foxton - The latest round of MLPerf AI inference benchmark scores are in. Nvidia has dominated both MLPerf training and inference results since the beginning, but in this round Qualcomm appears to be close on Nvidia's tail when it comes to data center/edge server inference.



Qualcomm Cloud AI100 PCIe and M.2 cards (Source: Qualcomm)

EETimes

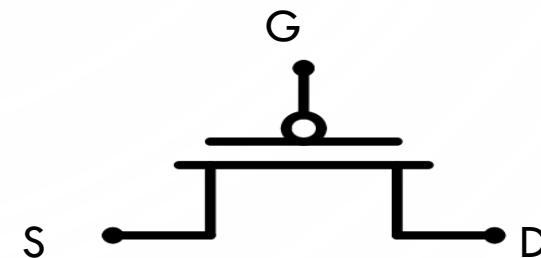
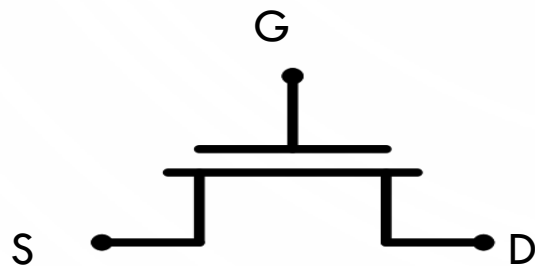
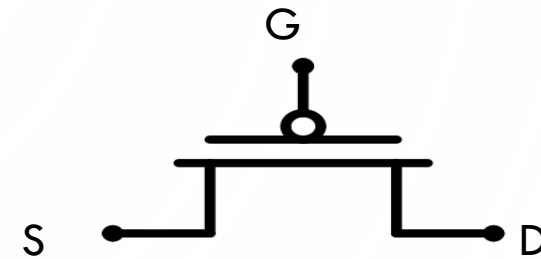
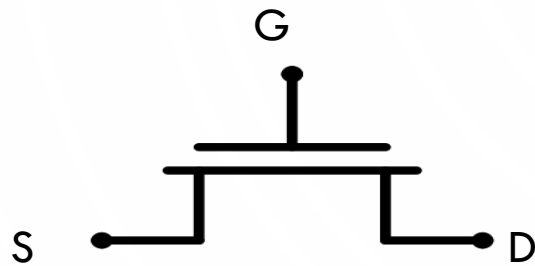
# Review

- CMOS process is used for producing chips
  - Planar bulk process used up to 28nm node
  - finFET, FDSOI used below the 22nm node
- Switch-level abstraction for MOS transistors



# MOS Switch

# MOS Switch



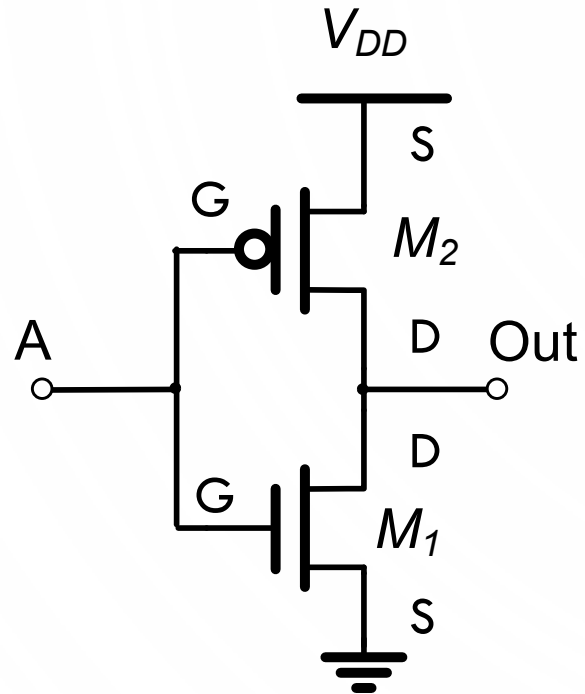


# CMOS Inverter

# CMOS Inverter

- Simple DC behavior

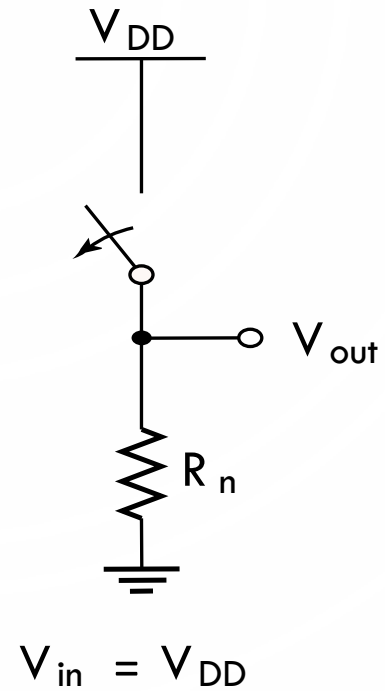
- Schematic



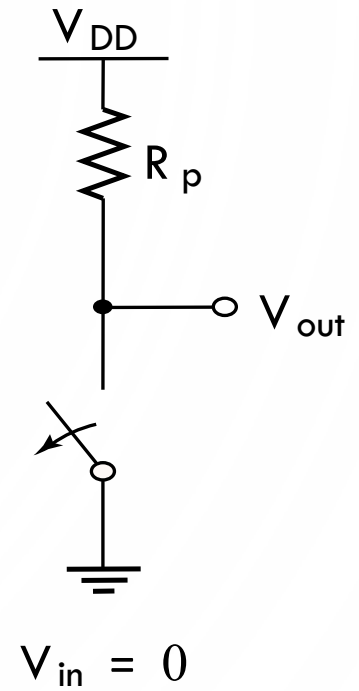
$W_p/L$

$W_n/L$

- Switch model



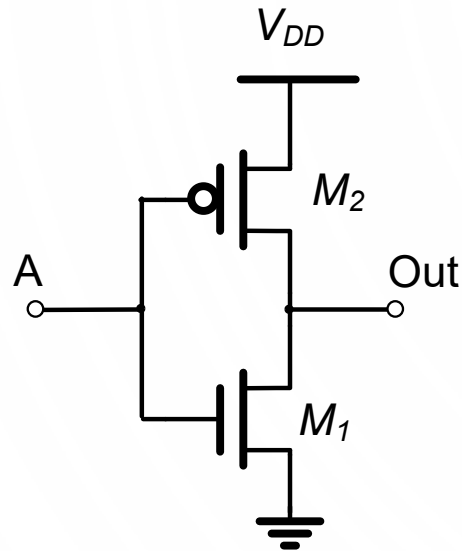
$V_{in} = V_{DD}$



$V_{in} = 0$

$$\begin{aligned} V_{OL} &= 0 \\ V_{OH} &= V_{DD} \end{aligned}$$

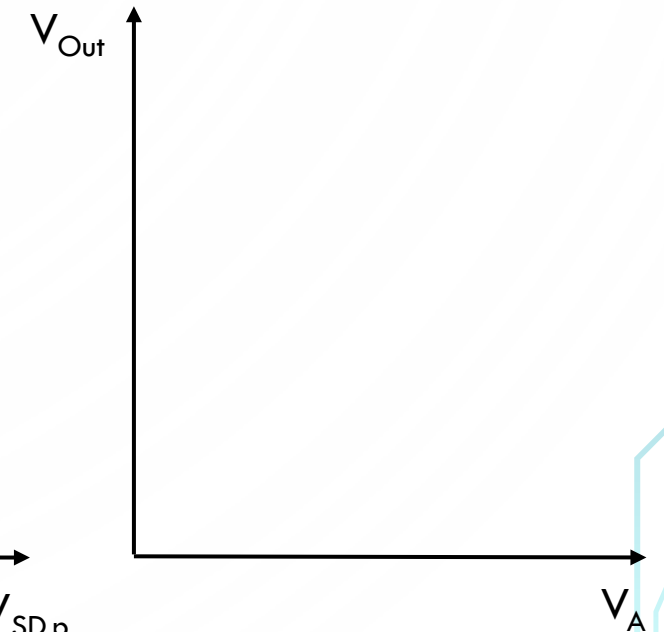
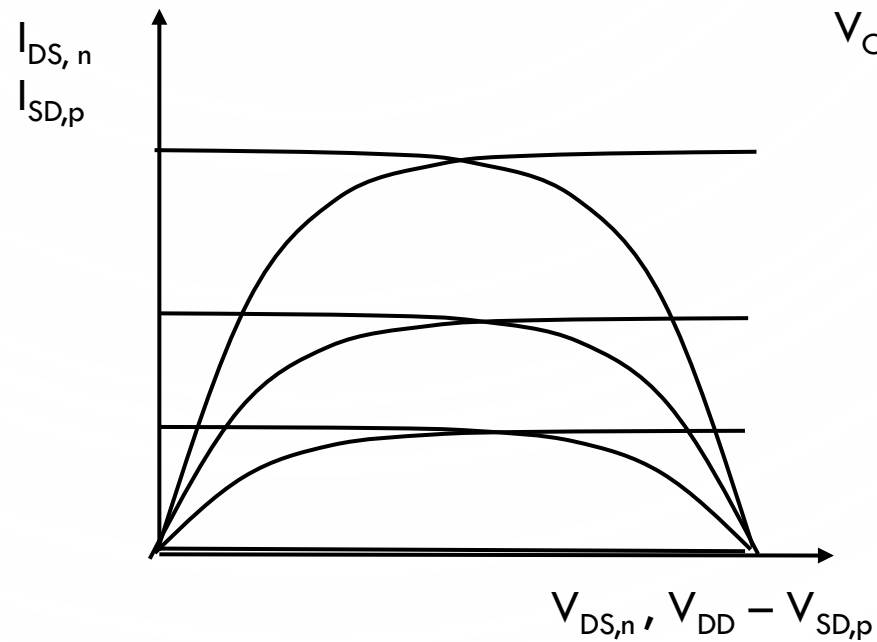
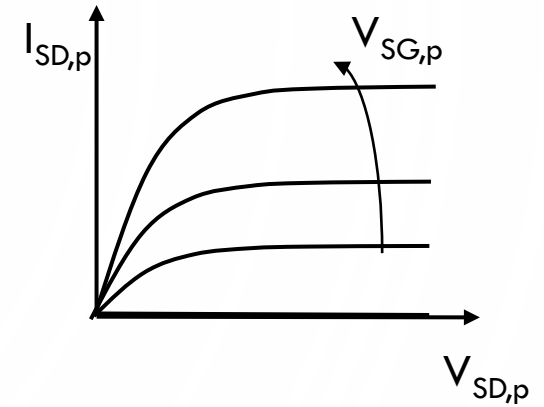
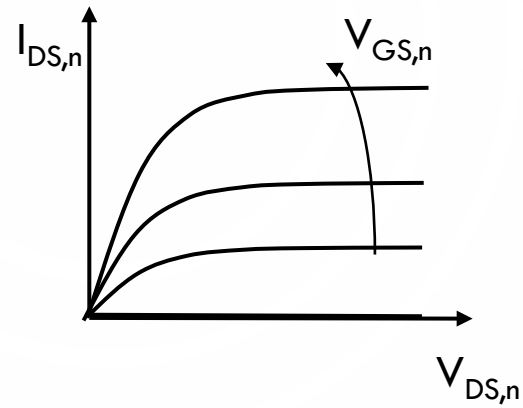
# Voltage Transfer Characteristic (VTC)



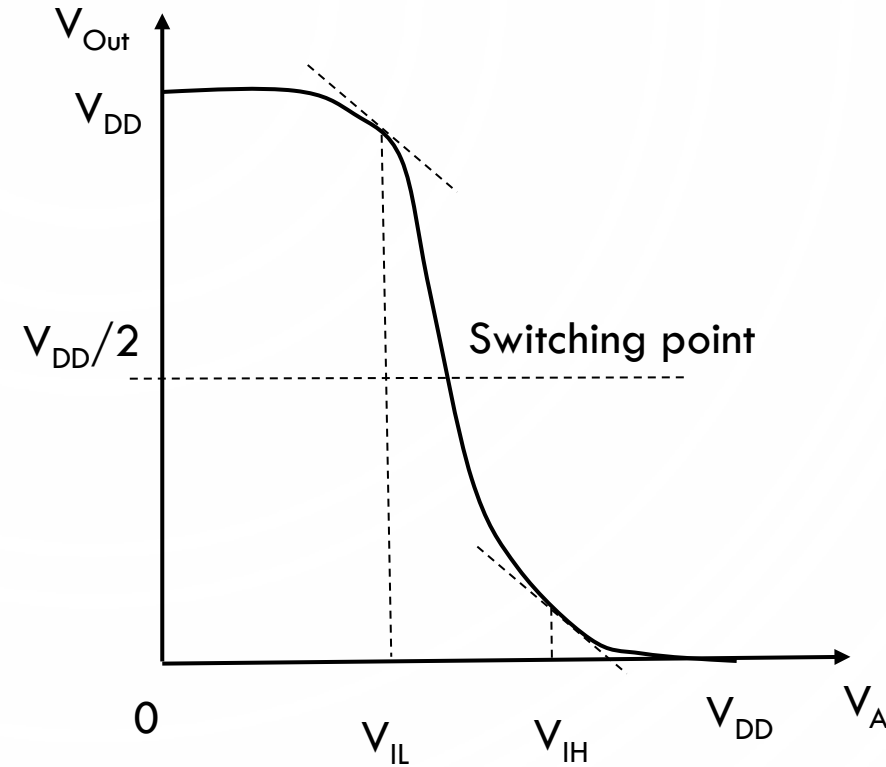
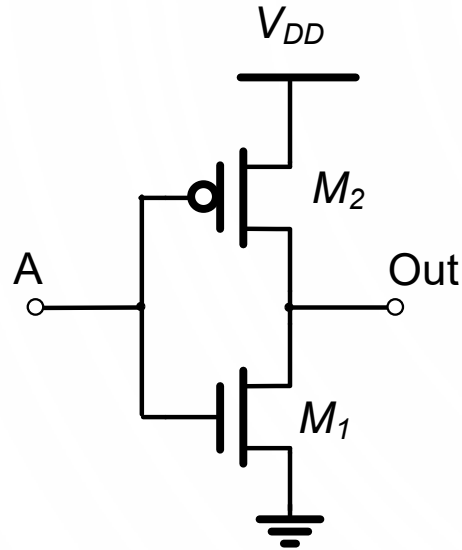
$$V_A = V_{GS,n} = V_{DD} - V_{SG,p}$$

$$I_{DS,n} = I_{SD,p}$$

$$V_{Out} = V_{DS,n} = V_{DD} - V_{SD,p}$$



# Voltage Transfer Characteristic (VTC)



- Can we change switching point ( $V_A$  for which  $V_{out} = V_{DD}/2$ )?



# Digital Circuits

- One logic representation

$$\text{Out} = \bar{A}$$

Truth table

A	Out
0	1
1	0

- Multiple libraries

- Layouts

- Number of metal 'tracks'
- More tracks, faster, but larger
- Less tracks – more compact, but slower

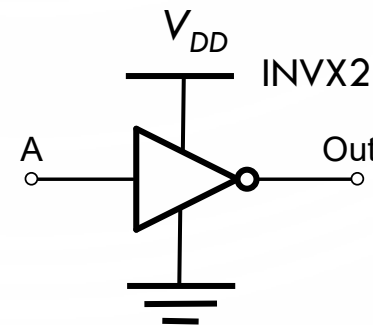
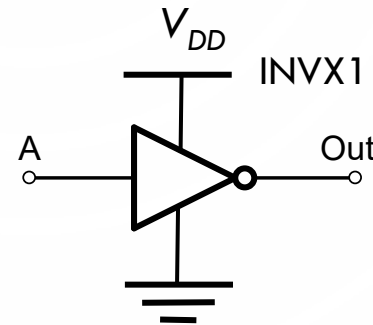
- Transistor thresholds ( $V_{Th}$ ) (for each track height):

- Regular (RVT)
- Low (LVT)
  - Faster, higher power
  - Slower, lower power
- High (HVT)

- Transistor lengths

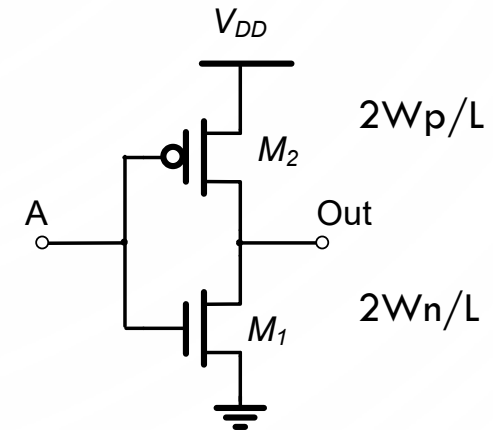
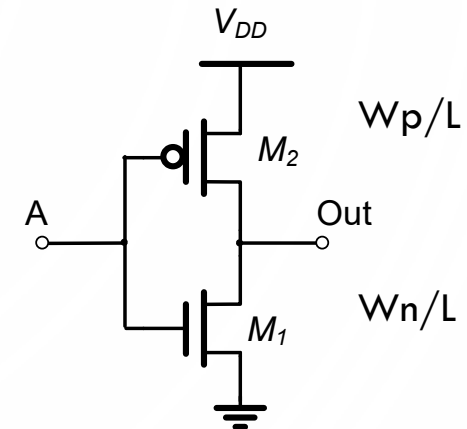
- Multiple gate sizes within a library

- Symbol

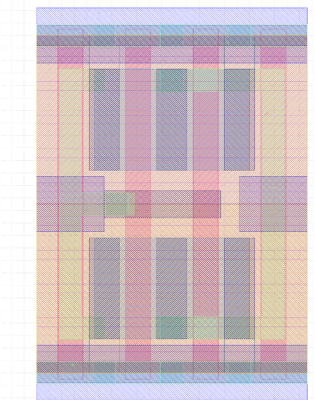
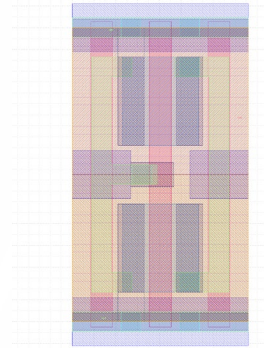


INVX3,  
INVX4,...

- Schematic



- Layout



# Administrivia

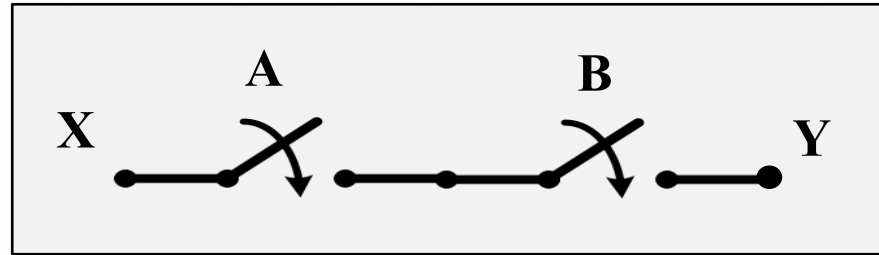
- Homework 5 due this week
- Lab 6 (last) this week
- Projects start next week



# CMOS Logic

# Building logic from switches

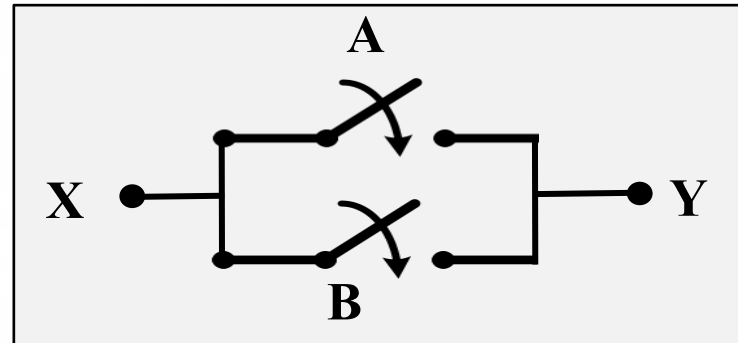
**Series**



**AND**

**$Y = X$  if  $A$  AND  $B$**

**Parallel**



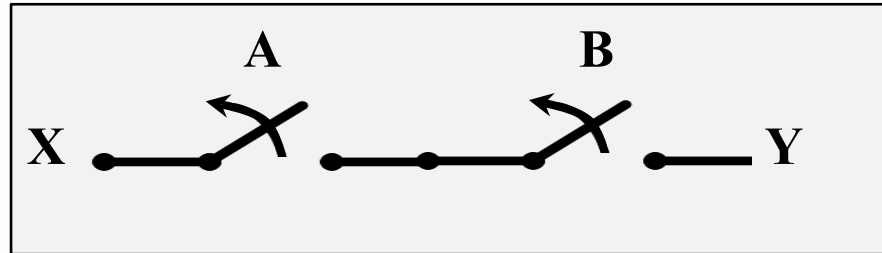
**OR**

**$Y = X$  if  $A$  OR  $B$**

(output undefined if condition not true)

# Logic using inverting switches

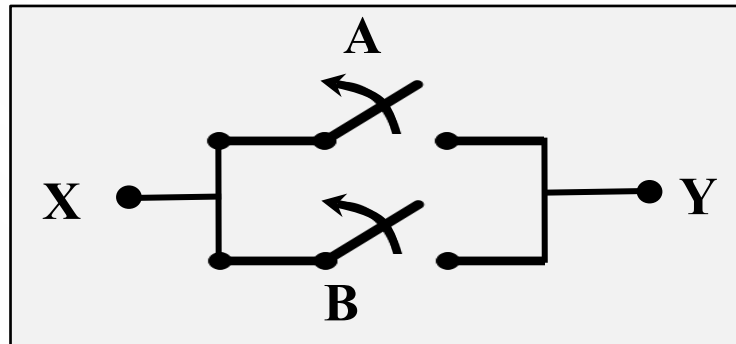
Series



NOR

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

Parallel

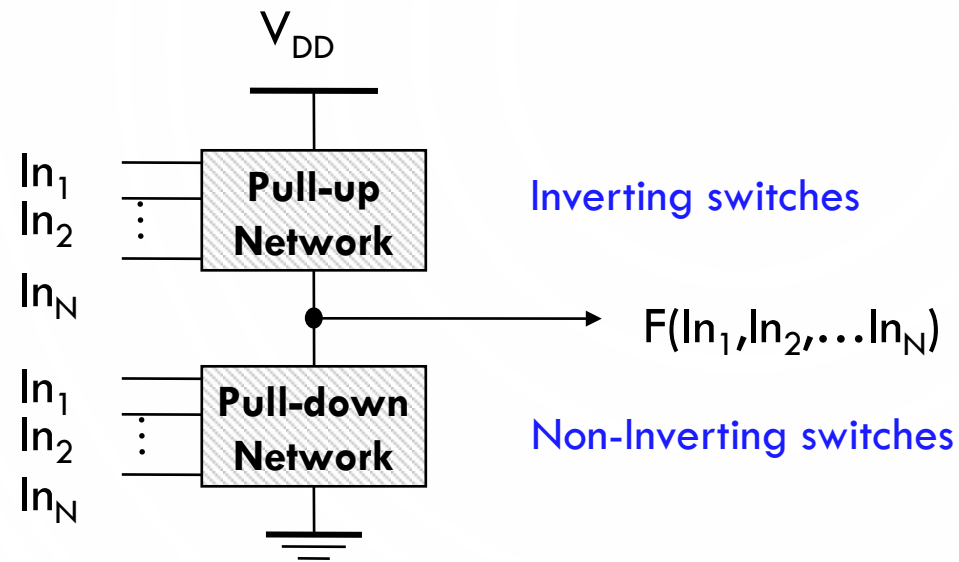


NAND

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

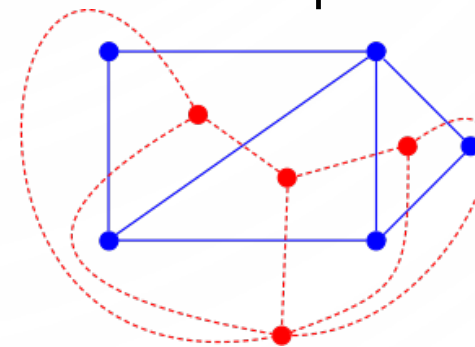
(output undefined if condition not true)

# Static Complementary CMOS



PUN and PDN are **dual** logic networks  
PUN and PDN functions are **complementary**

Dual Graphs



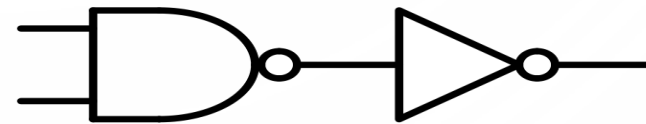
# Complementary CMOS Logic Style

- PUN is the **dual** to PDN  
(can be shown using DeMorgan's Theorems)

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

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

- Static CMOS gates are always inverting

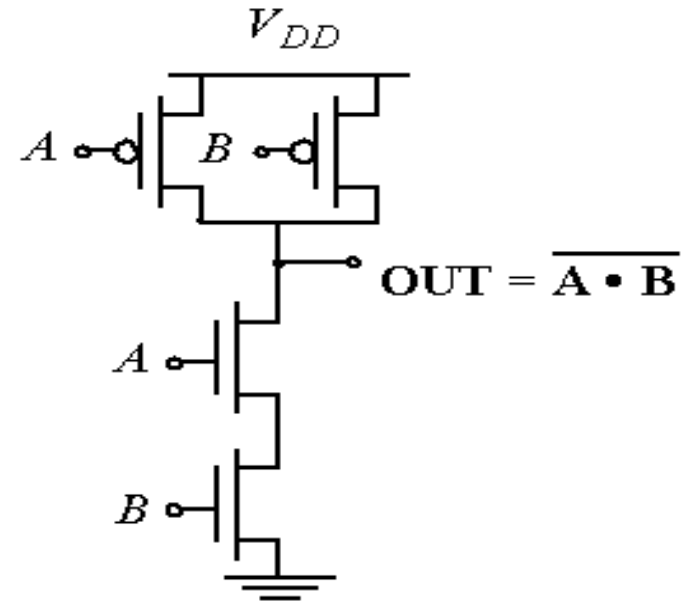


AND = NAND + INV

# 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**



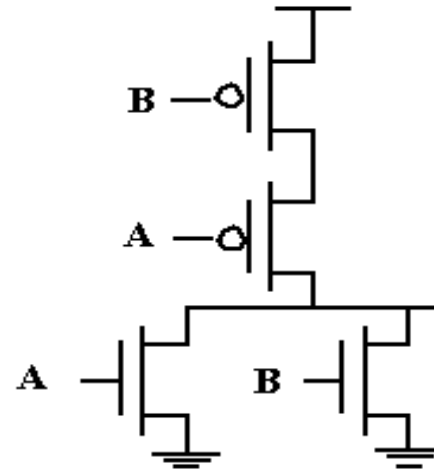
- PDN:  $G = AB \Rightarrow$  Conduction to GND
- PUN:  $F = \overline{A} + \overline{B} = \overline{AB} \Rightarrow$  Conduction to  $V_{DD}$
- $\overline{G(l_n_1, l_n_2, l_n_3, \dots)} \equiv \overline{F(l_n_1, l_n_2, l_n_3, \dots)}$



# Example Gate: NOR

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

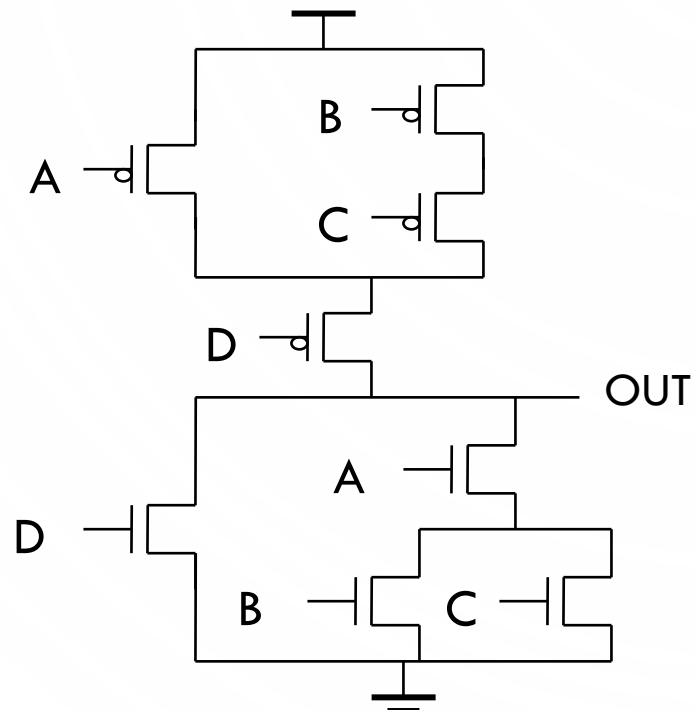
**Truth Table of a 2 input NOR gate**



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

# Complex CMOS Gate

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



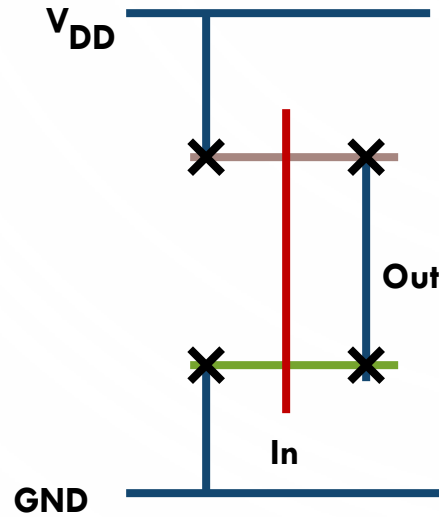
- Note: In scaled processes max #inputs is 3-4
- Max stack height is 2 or 3

# Stick Diagrams

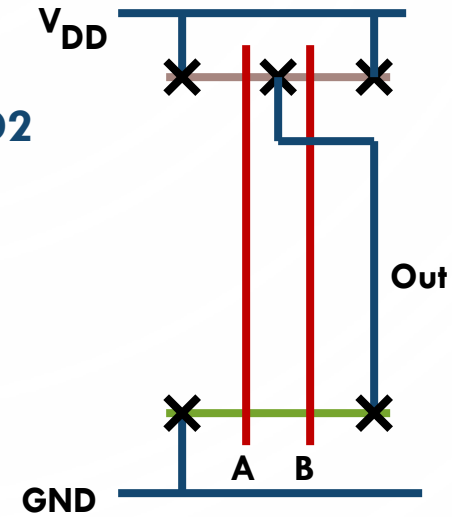
Contains no dimensions

Represents relative positions of transistors

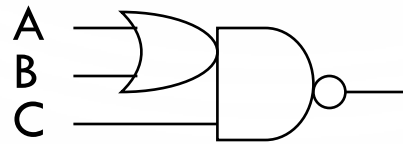
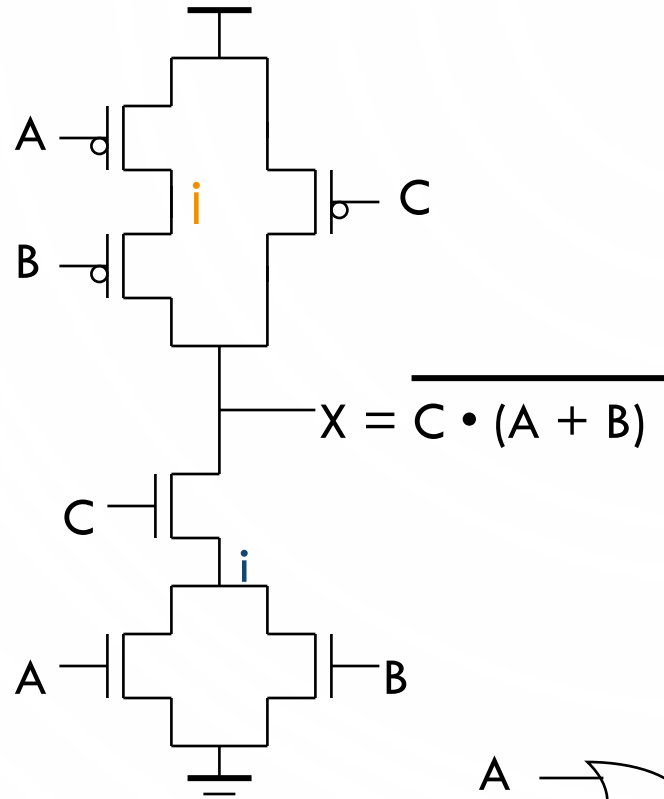
Inverter



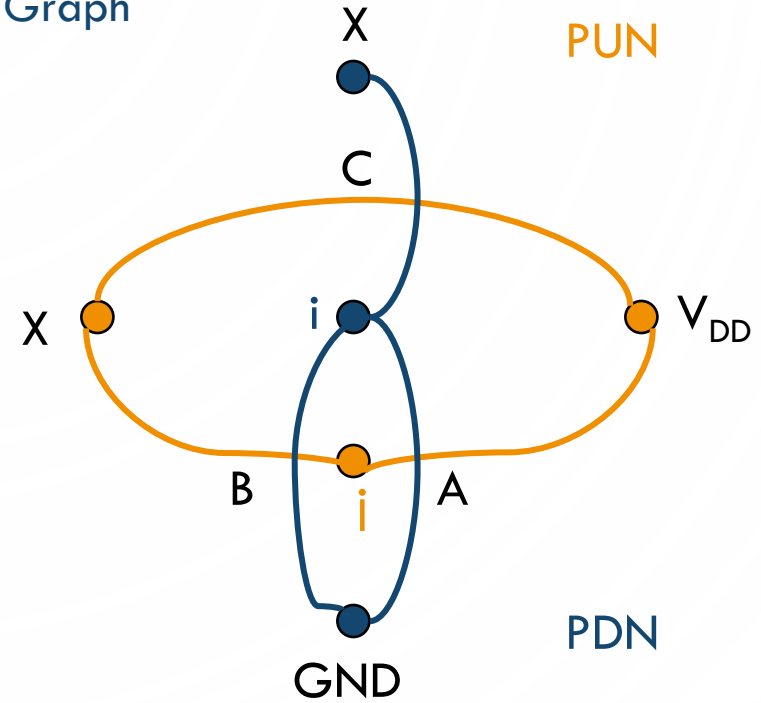
NAND2



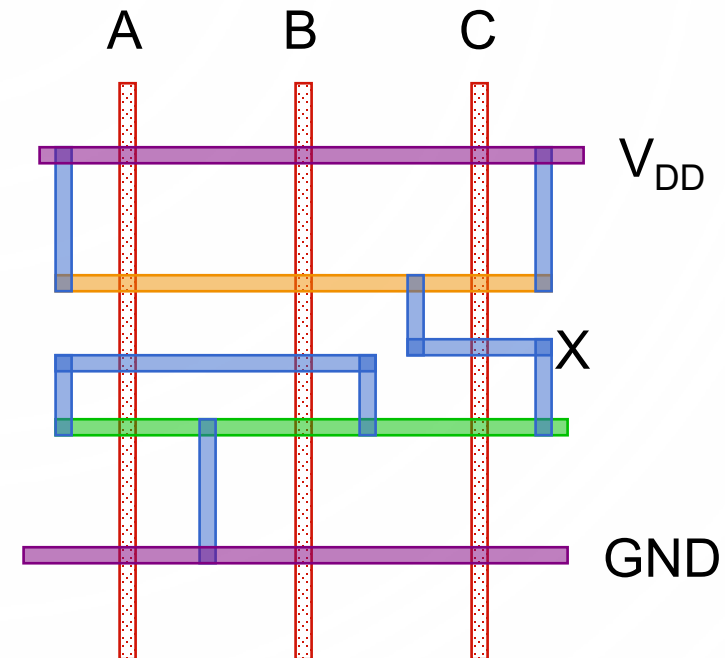
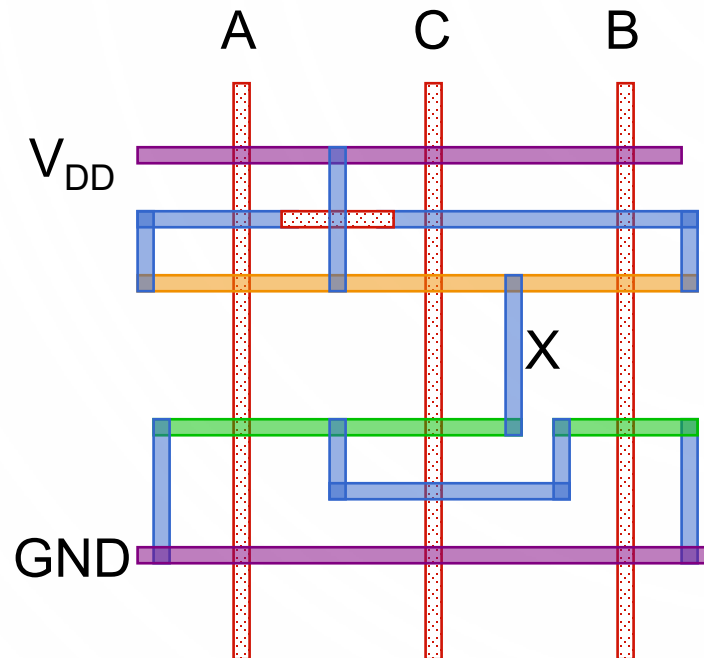
# Stick Diagrams



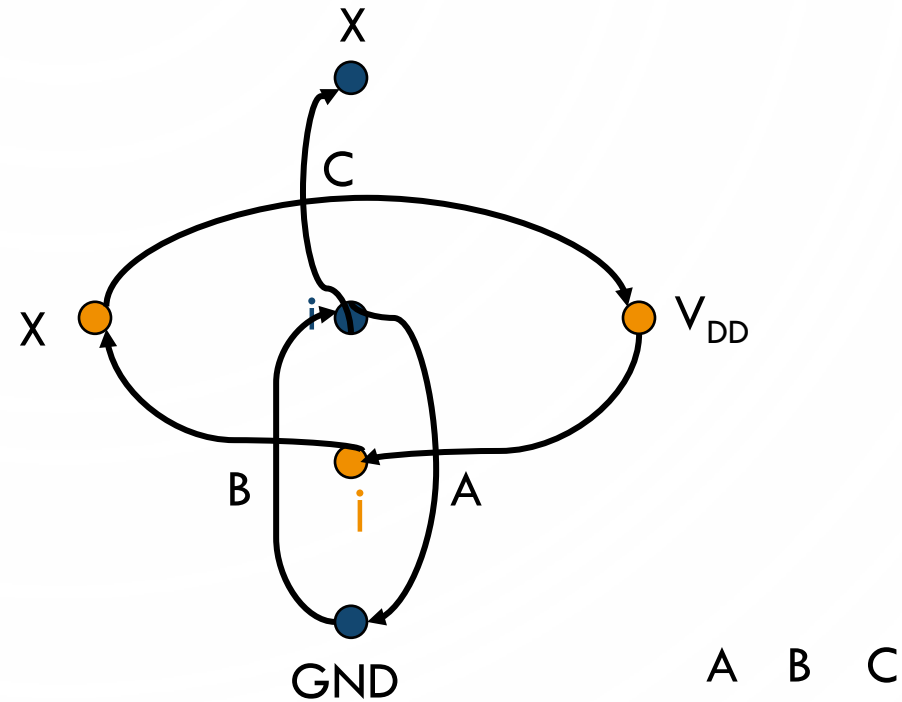
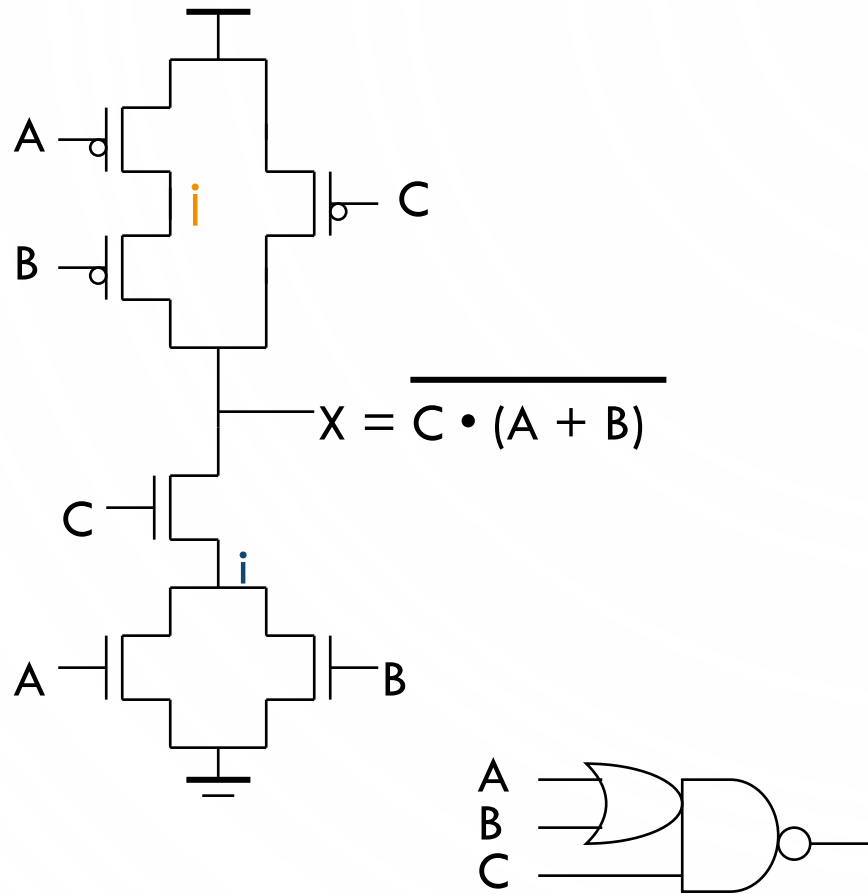
Logic Graph



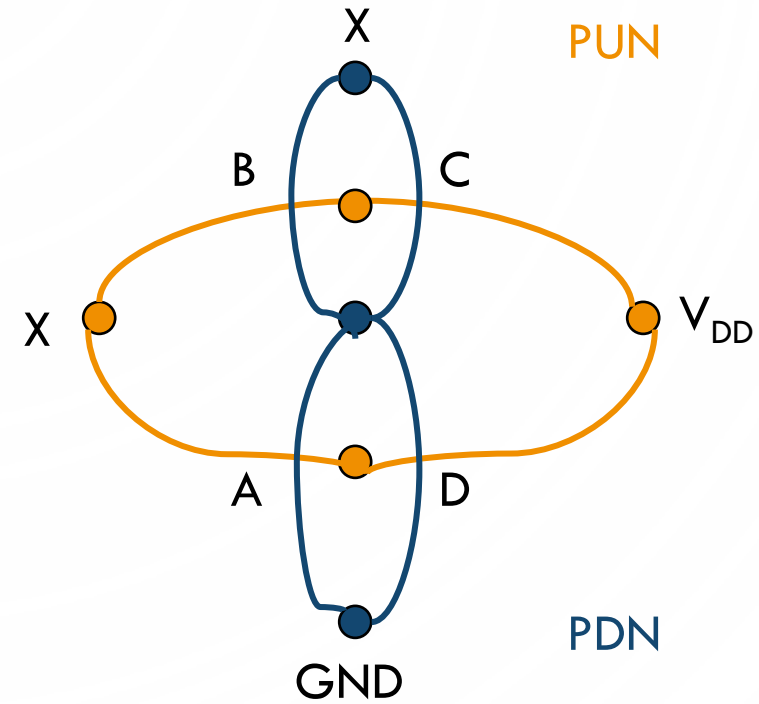
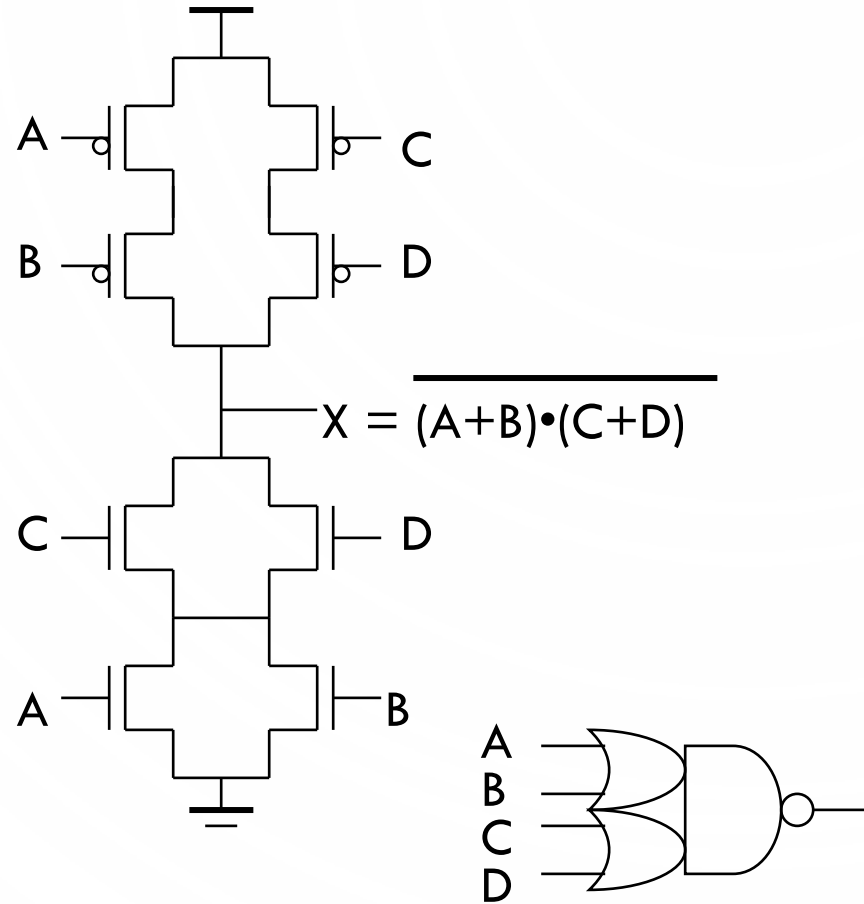
# Two Versions of $C \cdot (A + B)$



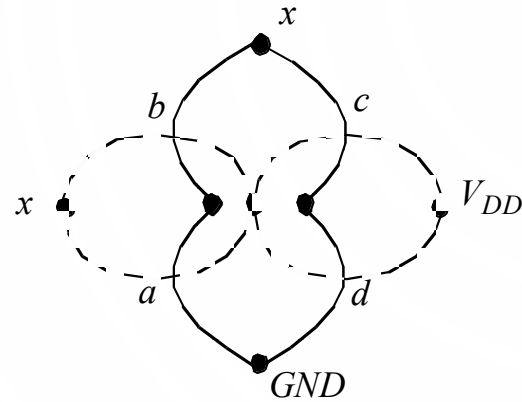
# Consistent Euler Path



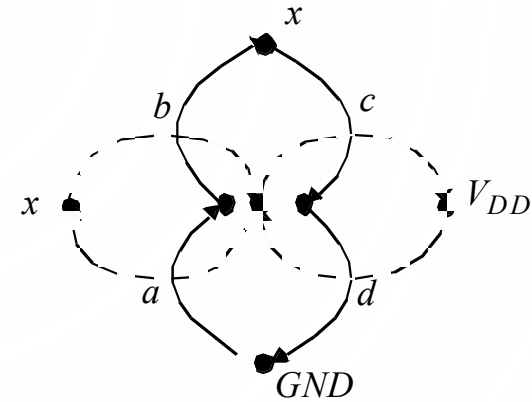
# OAI22 Logic Graph



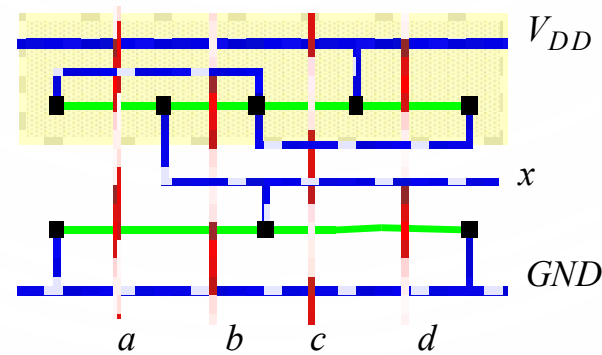
Example:  $x = \overline{ab+cd}$



(a) Logic graphs for  $\overline{ab+cd}$



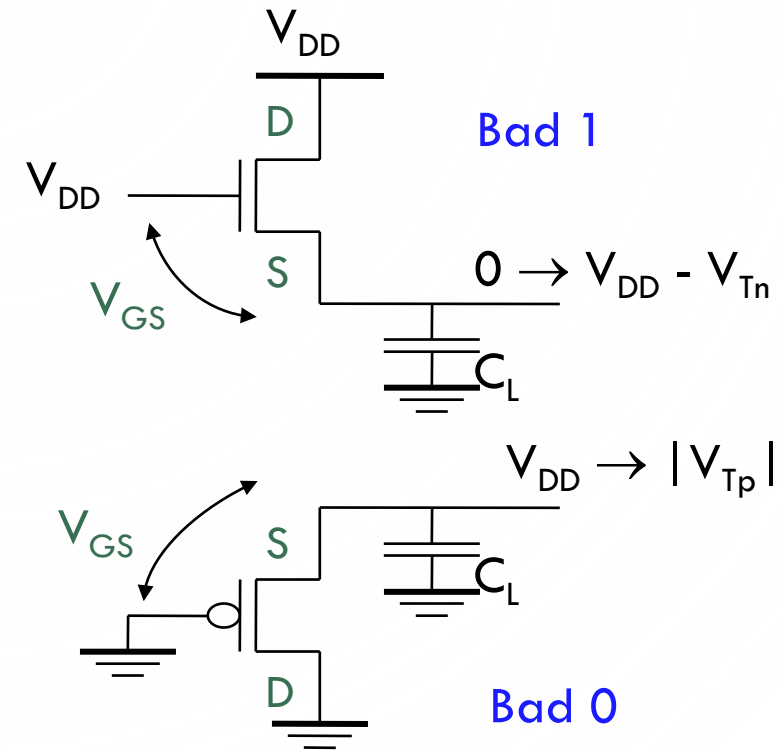
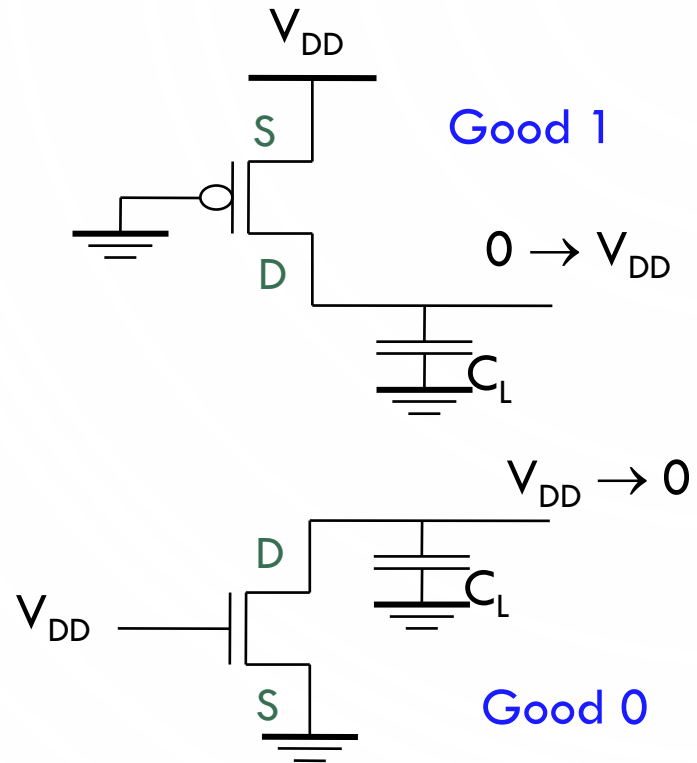
(b) Euler Paths  $\{a\ b\ c\ d\}$



(c) stick diagram for ordering  $\{a\ b\ c\ d\}$

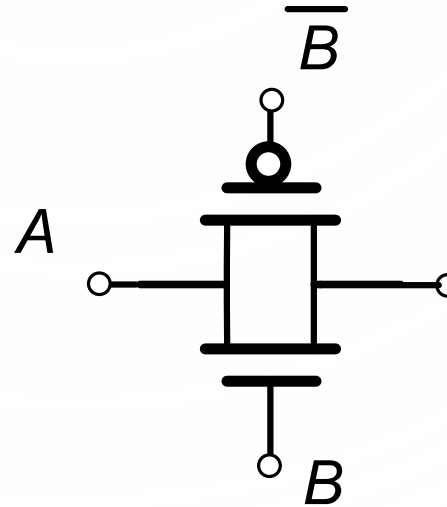


# Switch Limitations



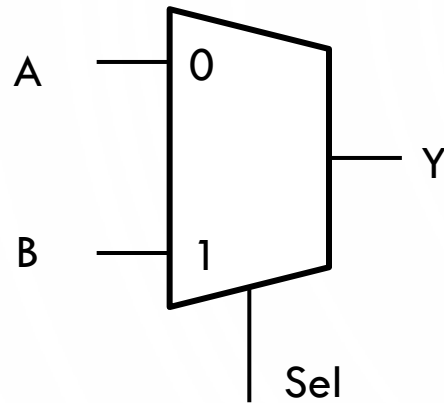
# Transmission Gate

- Transmission gates are the way to build “switches” in CMOS.
- In general, both transistor types are needed:
  - nFET to pass zeros.
  - pFET to pass ones.
- The transmission gate is ‘non-isolating’.

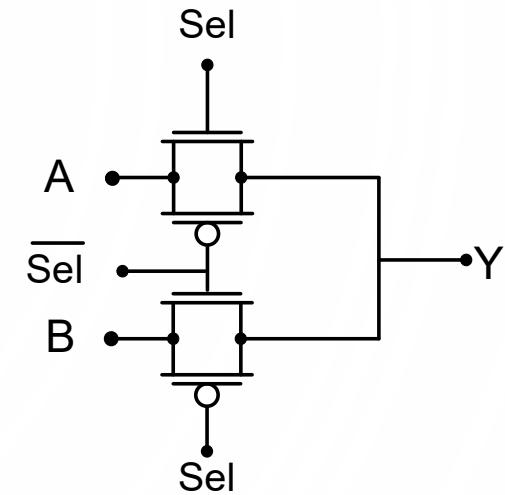


# Transmission-Gate Multiplexer

- Implementation



Sel	Y
0	A
1	B



```
module comb(input a, b, sel,
            output reg y);
    always @(*) begin
        case (sel)
            1b'0: y <= a;
            1b'1: y <= b;
        endcase
    end
endmodule
```

# CMOS Multiplexer

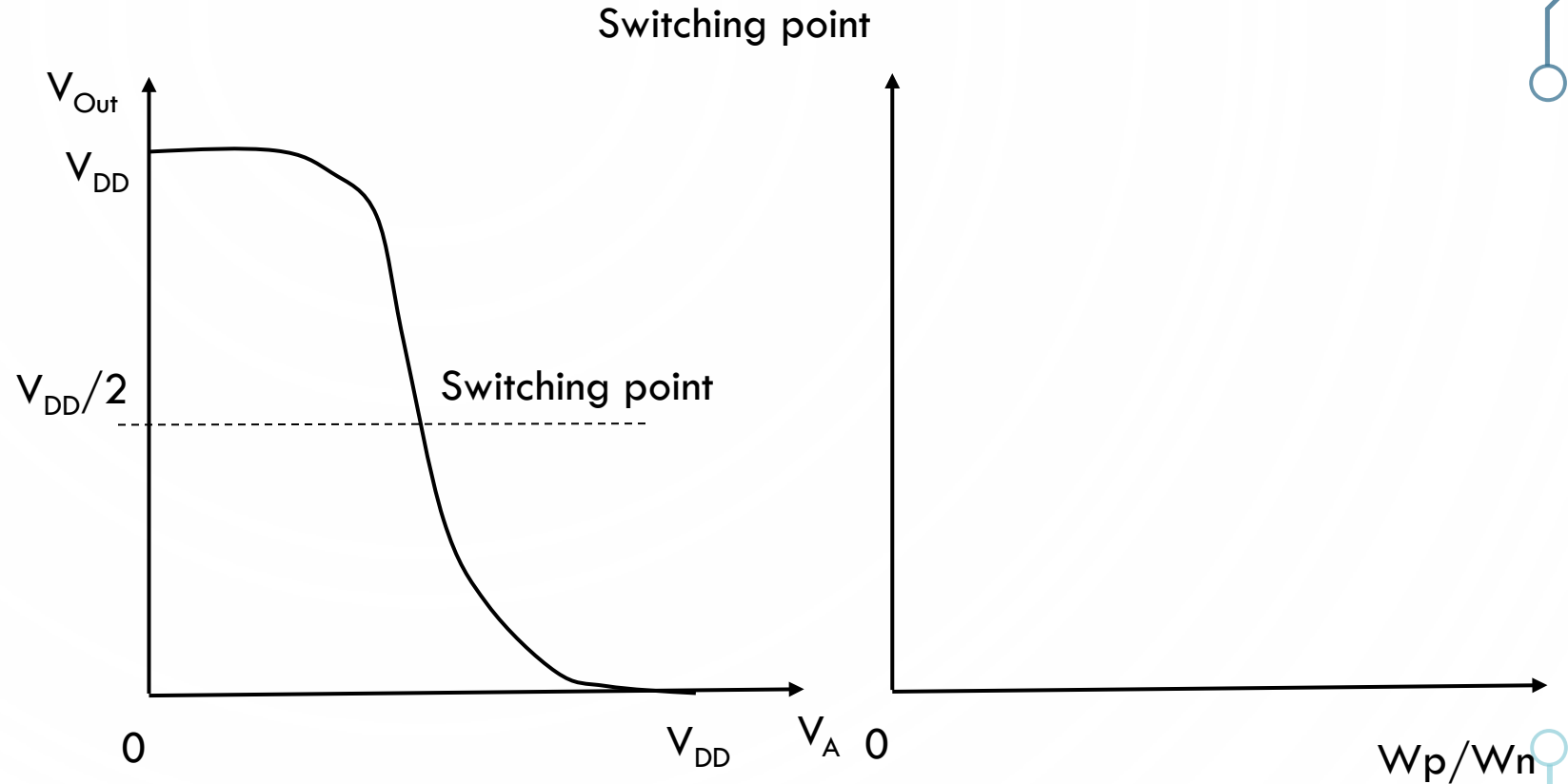
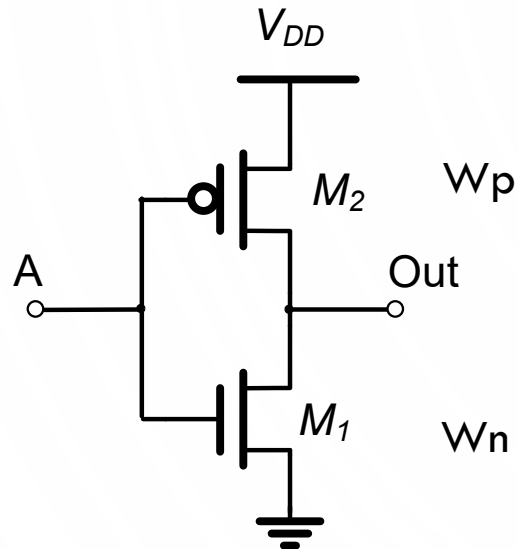
Sel	Y
0	A
1	B



# CMOS Sizing

# Transistor Sizing

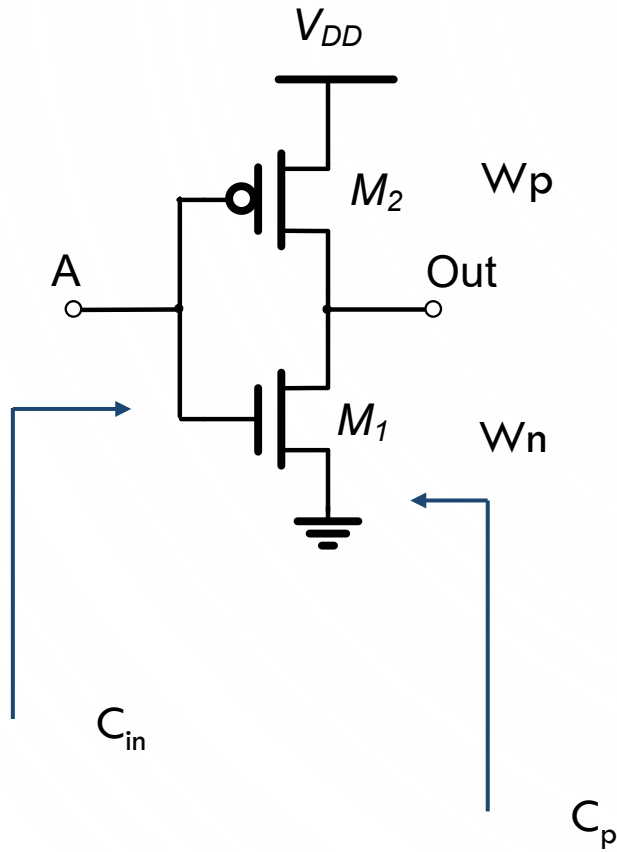
- Optimal  $W_p/W_n$



- In the past,  $W_p > W_n$  (see Rabaey, 2<sup>nd</sup> ed)
- In modern processes (finFET),  $W_p = W_n$

# Gate Sizing

- Doubling the gate size (by doubling  $W_s$ ):



- Doubles  $C_{in}$
- Halves equivalent gate resistance
- Doubles  $C_p$

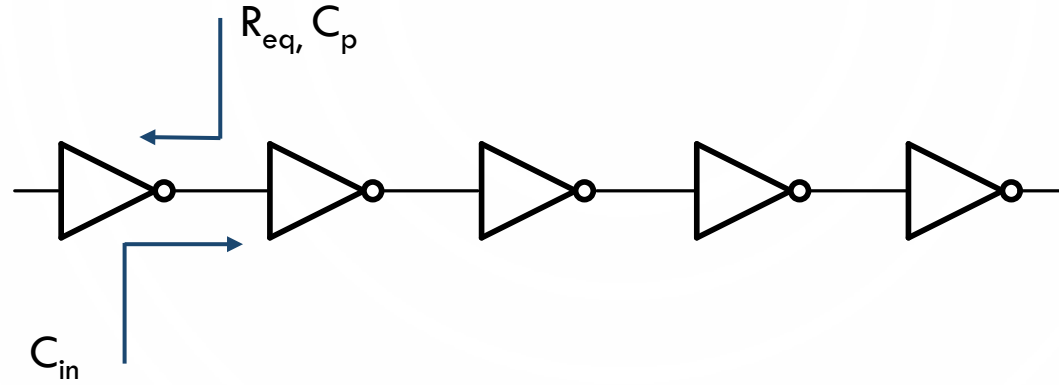


# CMOS Delay

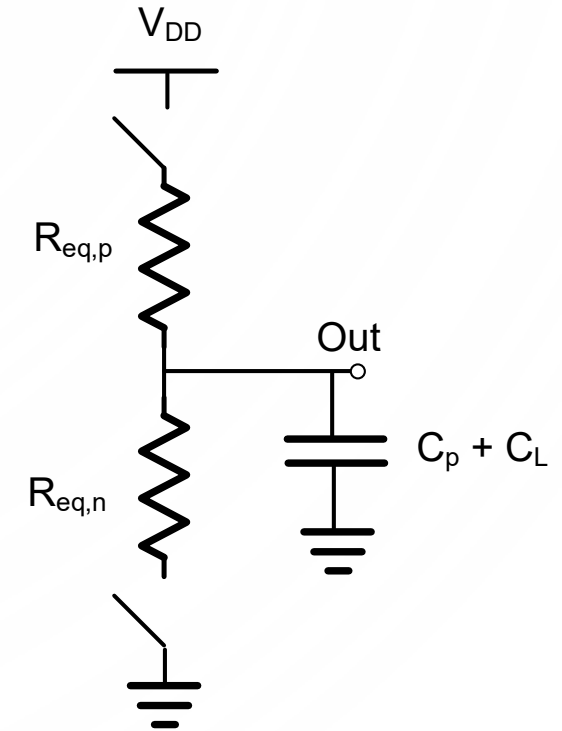
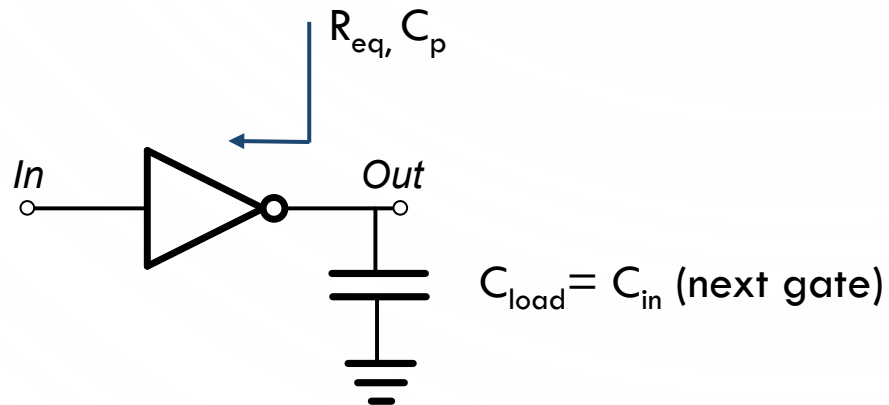


# Inverter Delay

- How to time this?

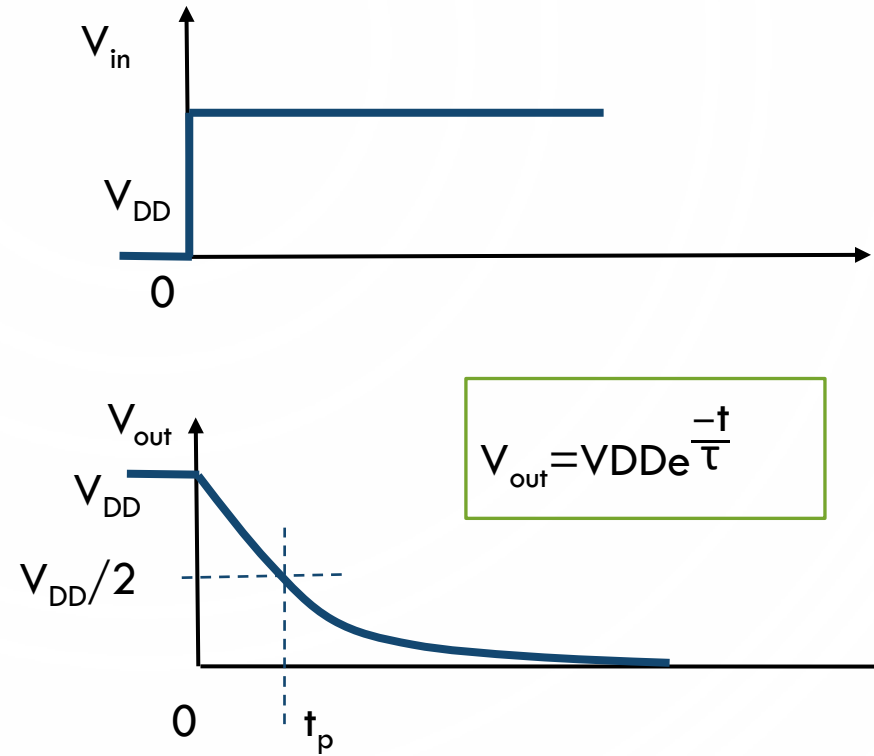
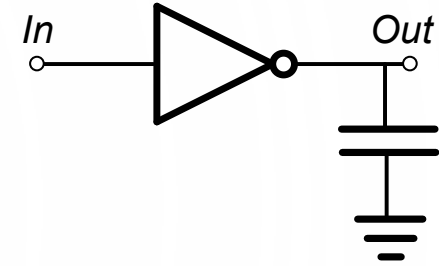
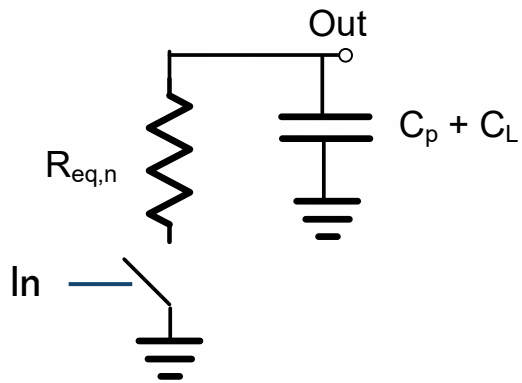
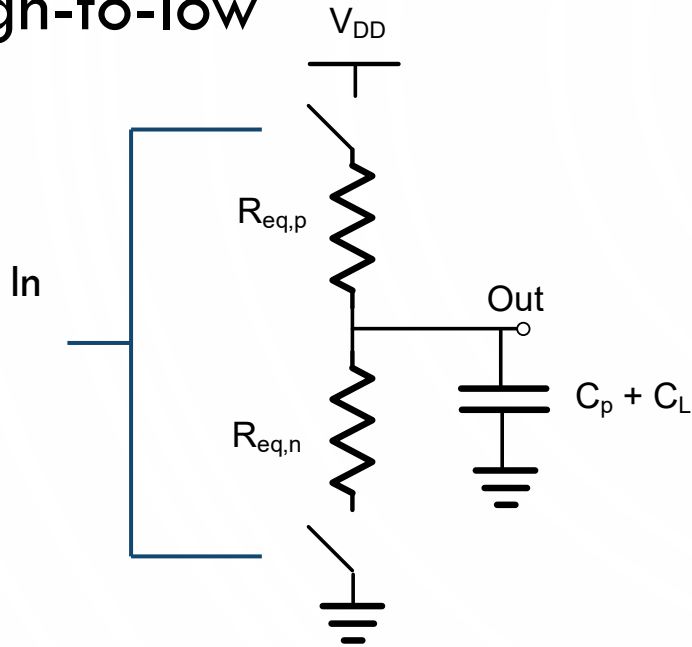


- Each gate has an  $R_{eq}$  and drives  $C_{in}$  of the next gate



# Inverter Delay

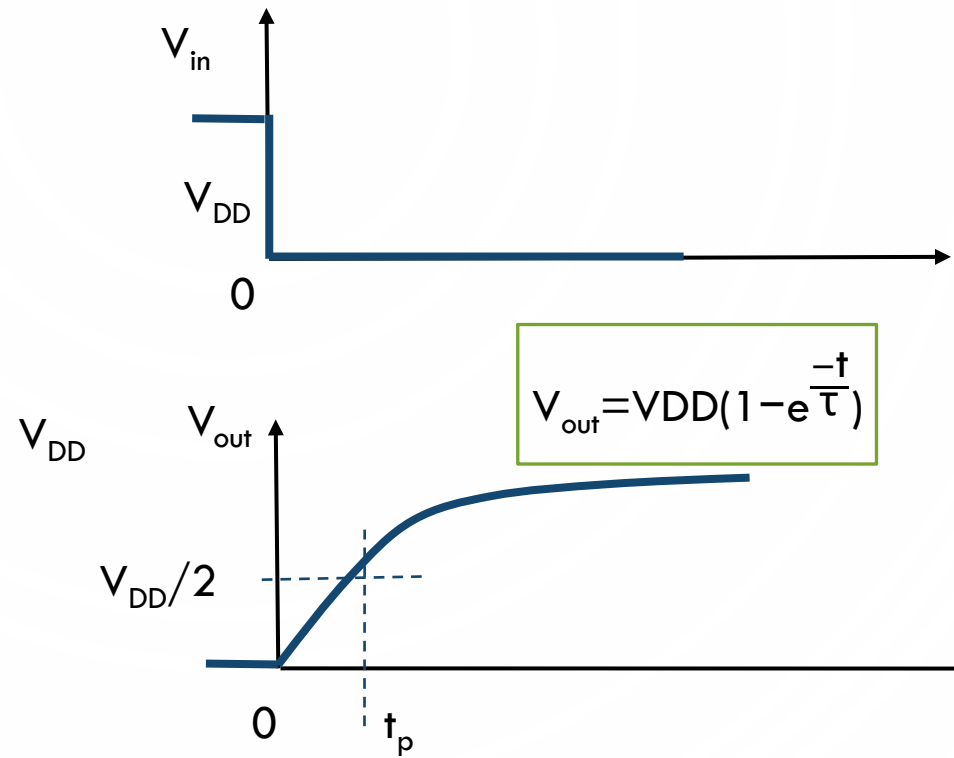
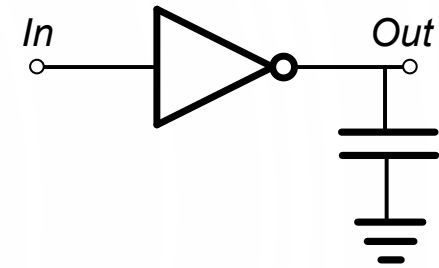
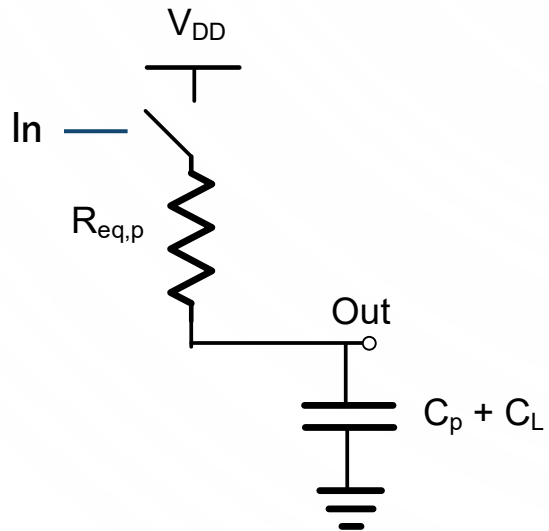
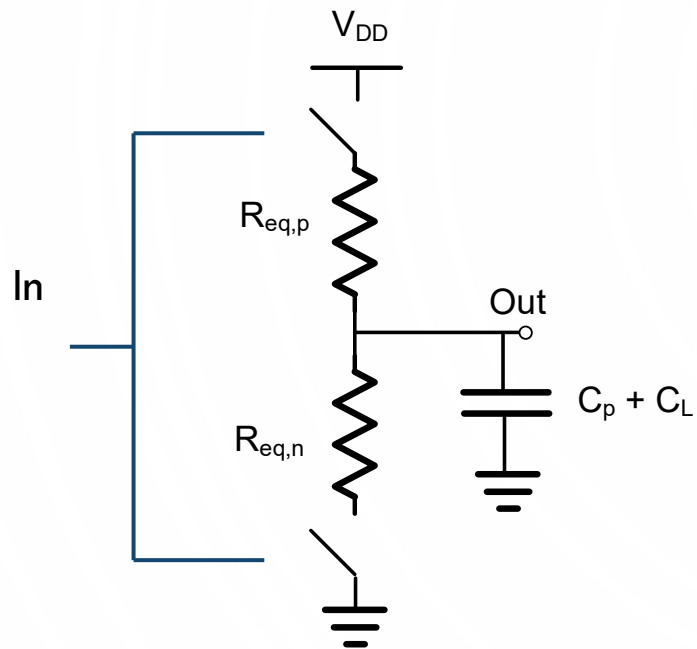
- High-to-low



$$t_{p,HL} = (\ln 2)\tau = 0.7 \text{ Req}_n(C_p + C_L)$$

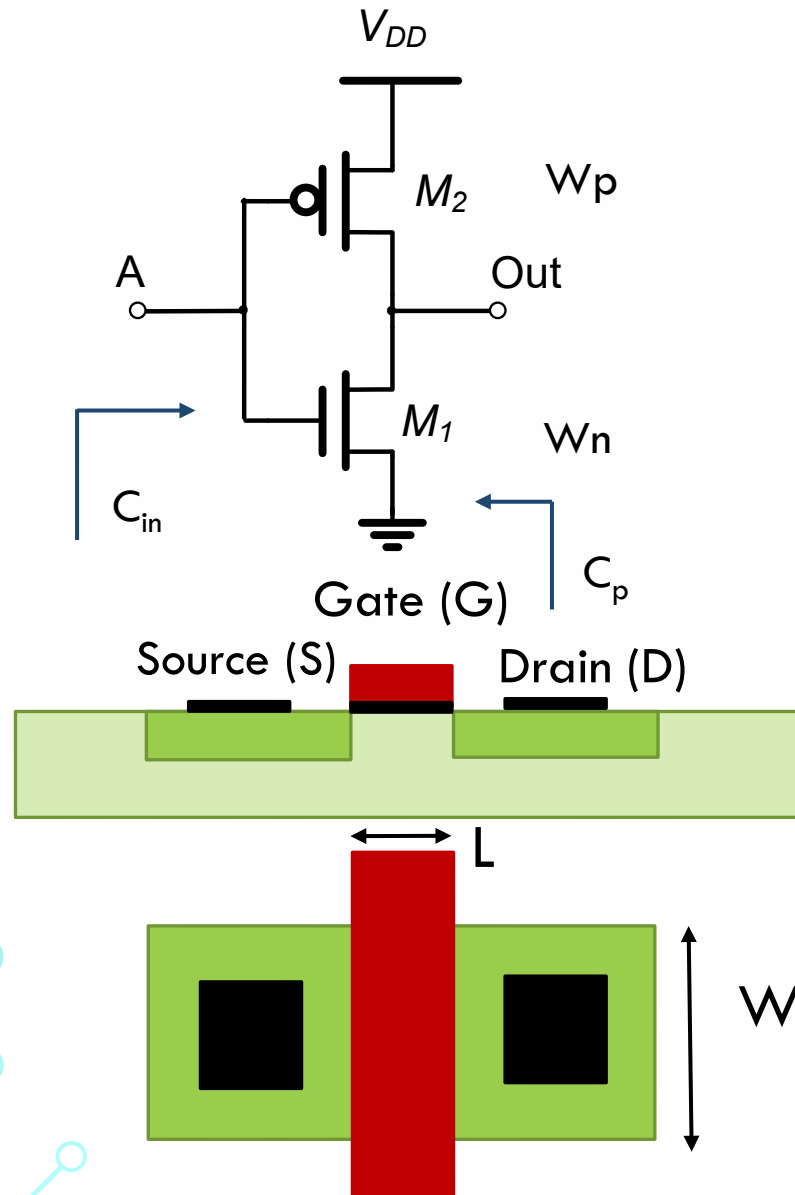
$$\tau = \text{Req}_n(C_p + C_L)$$

# Inverter Delay



$$t_{p,LH} = (\ln 2)\tau = 0.7 R_{eq,p}(C_p + C_L)$$

# Capacitances



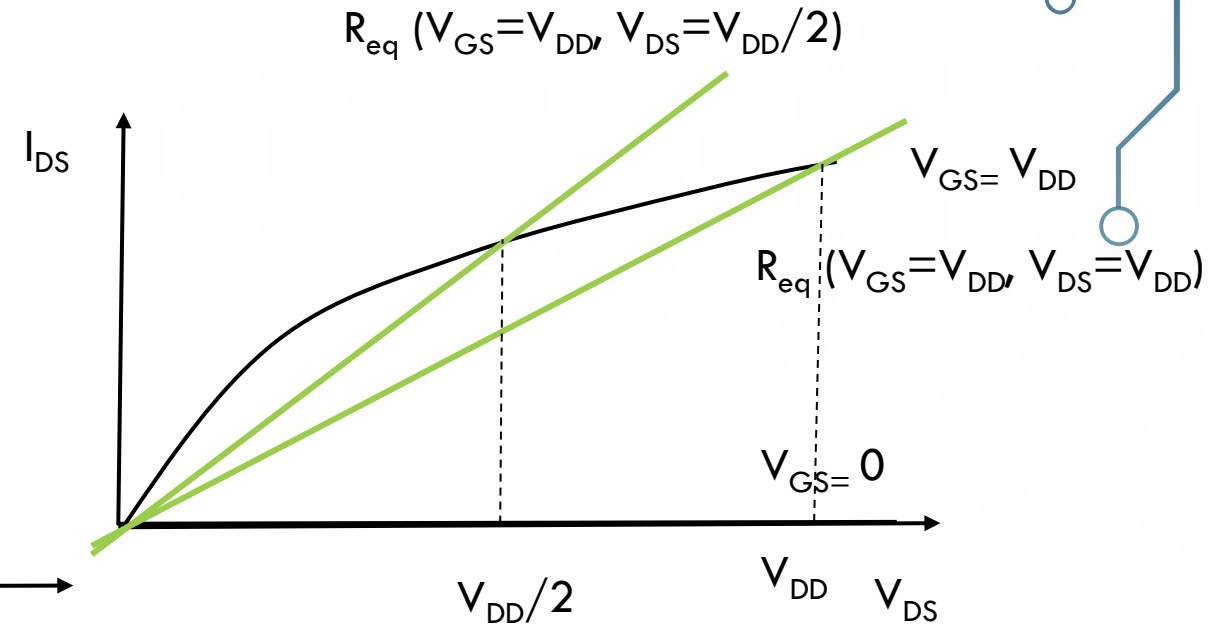
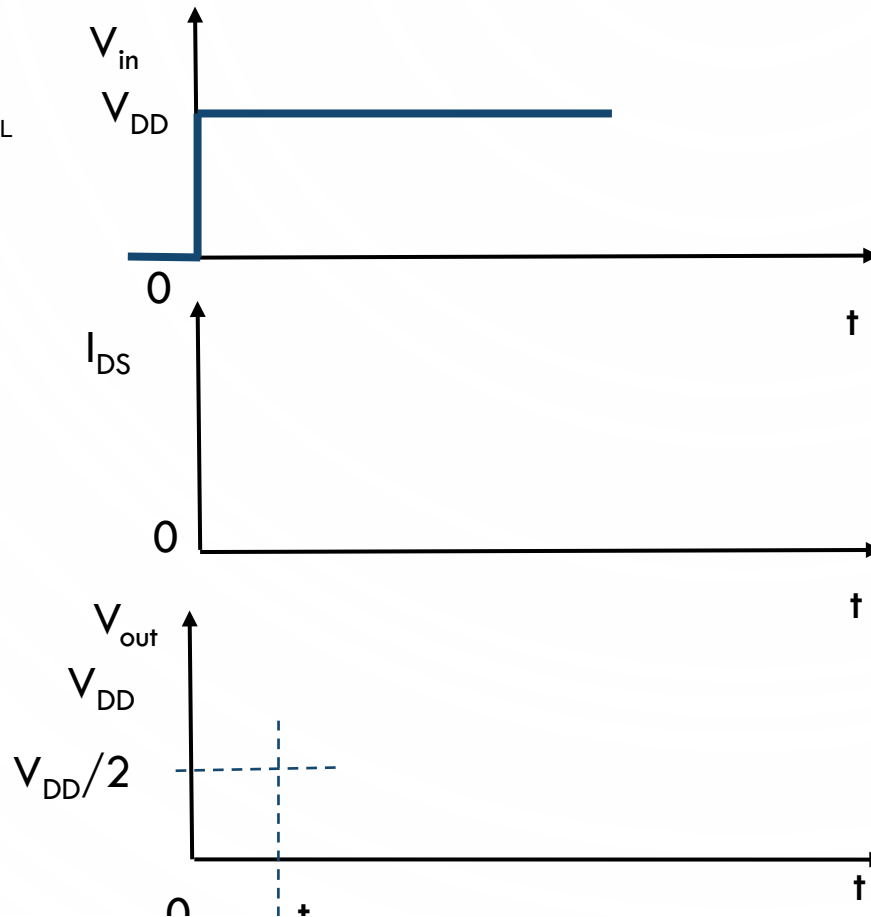
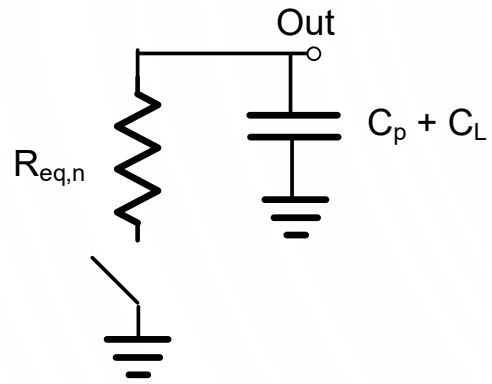
- $C_{in}$  is largely set by the gate cap
  - $\sim WL$
  - $2xW = 2xC_{in}$
  - It is non-linear, but we will ignore that

- $C_p$  is largely set by the drain cap
  - $\sim W$  (drain area/perimeter)
  - $2xW = 2xC_p$

$$C_p = \gamma C_{in}$$

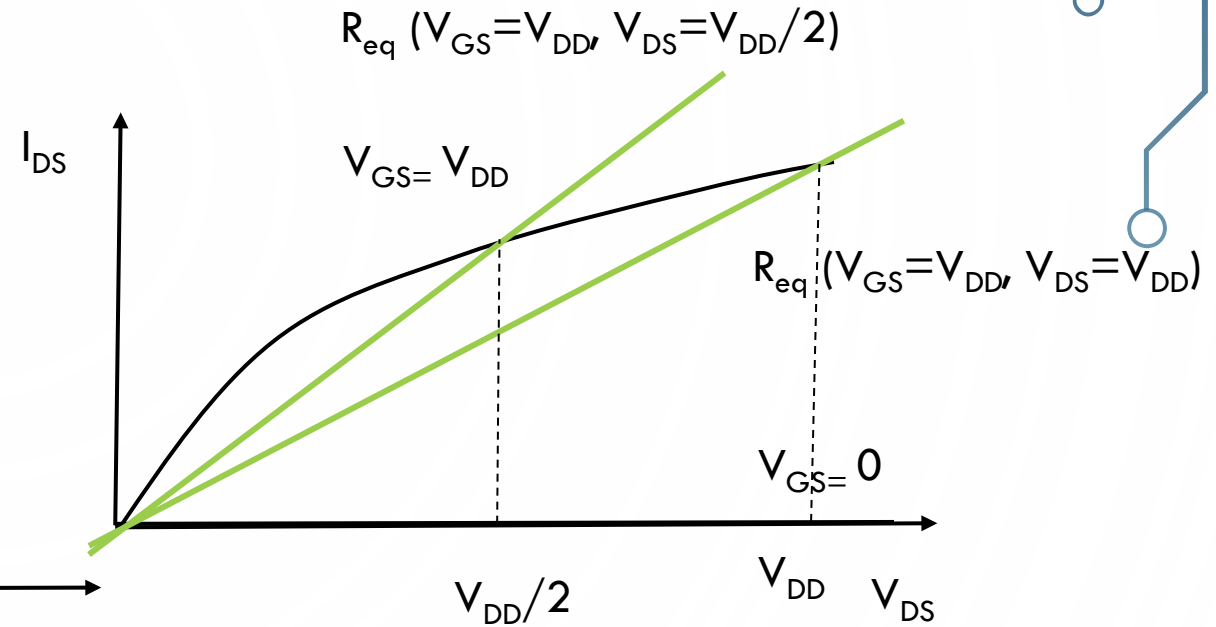
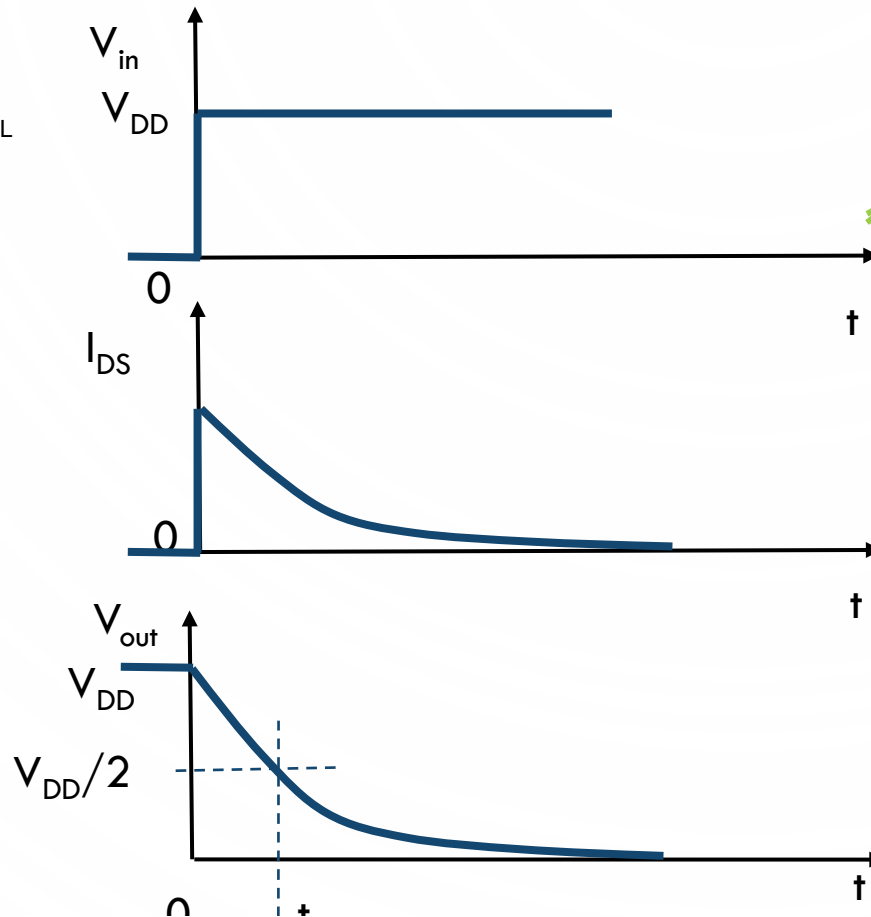
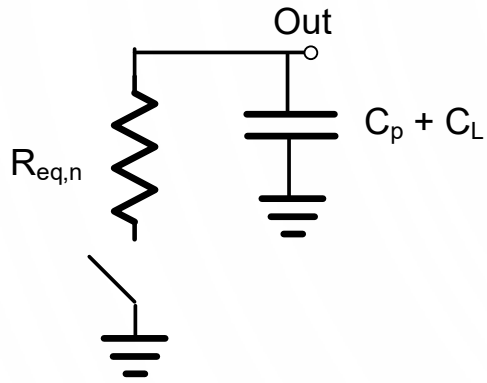
# Equivalent Resistances

- Transistor  $I_{DS}-V_{DS}$  trajectory
- Averaging produces  $R_{eq}$



# Equivalent Resistances

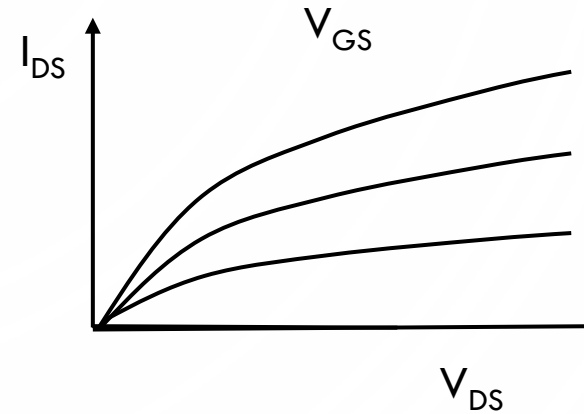
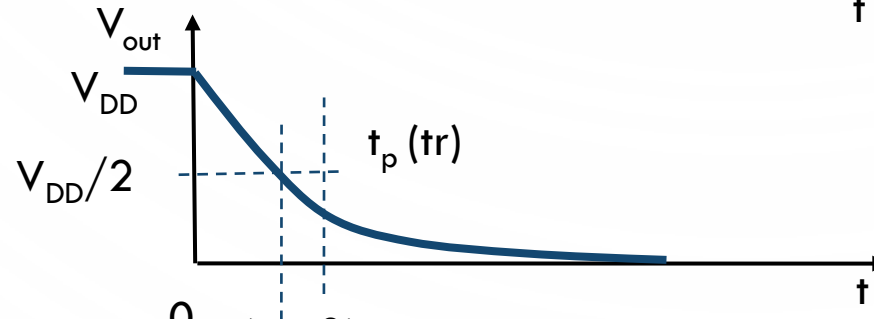
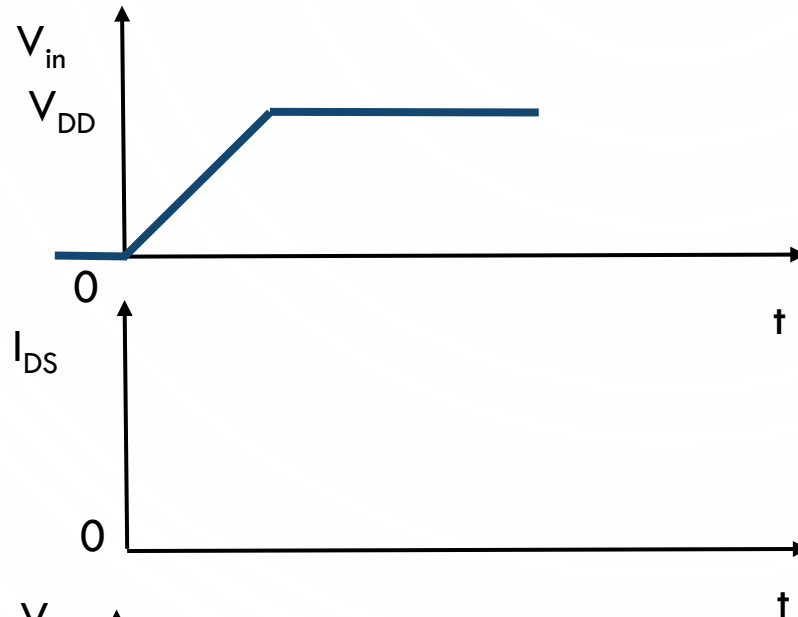
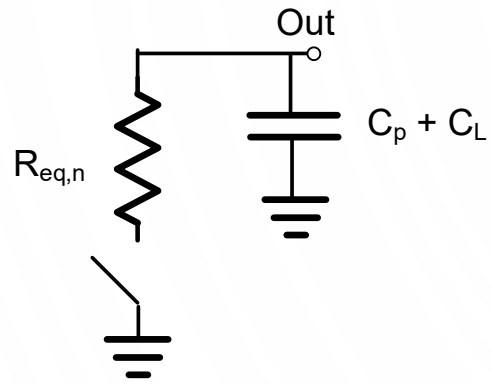
- Transistor  $I_{DS}-V_{DS}$  trajectory
- Averaging produces  $R_{eq}$



$$R_{eq} = (R_{eq,start} + R_{eq,mid})/2$$

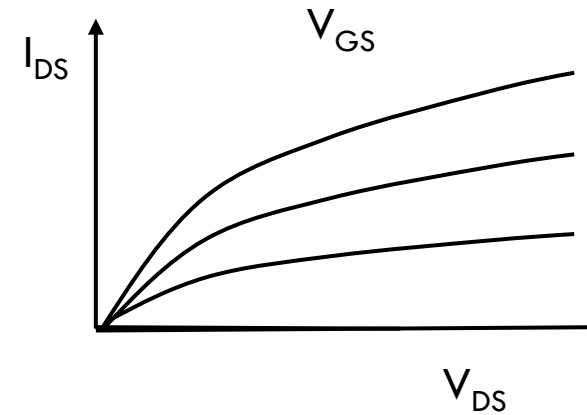
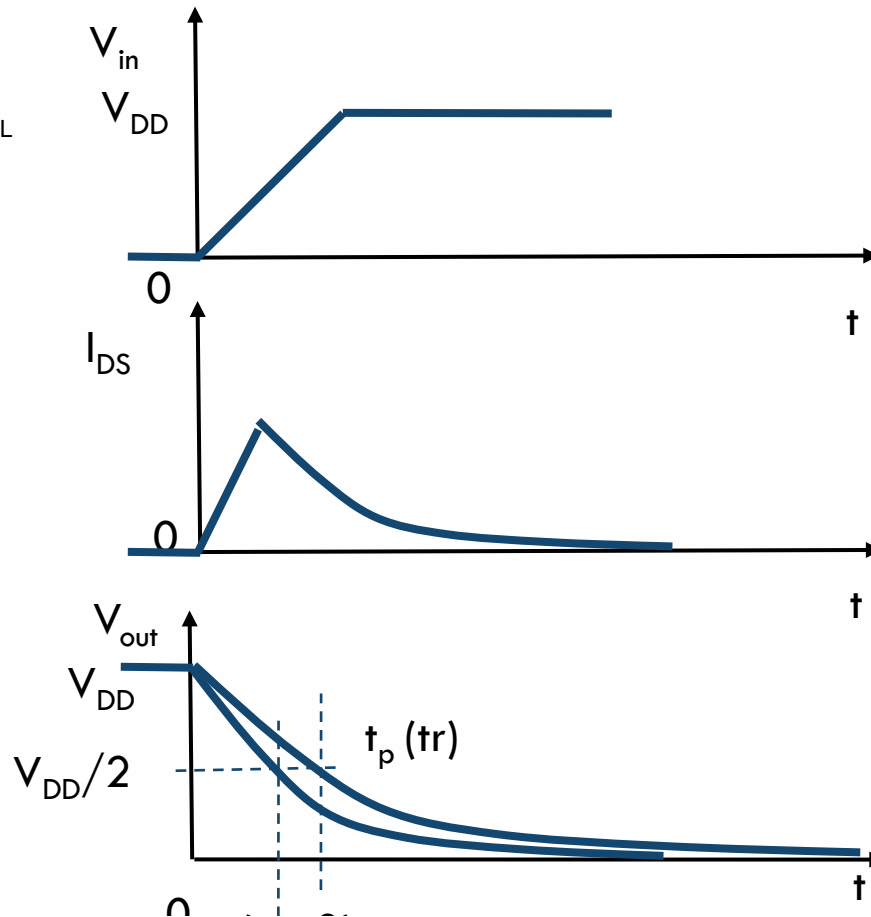
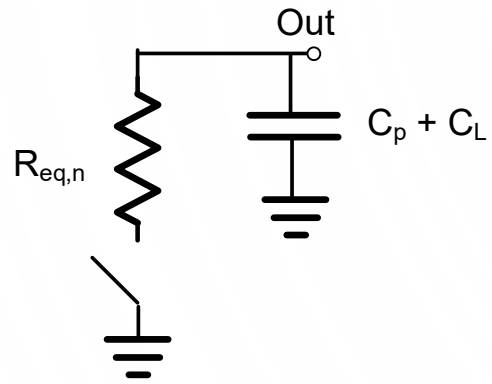
# Impact of Rise/Fall times

- Impacts the  $I_{DS}$ - $V_{DS}$  trajectory



# Impact of Rise/Fall times

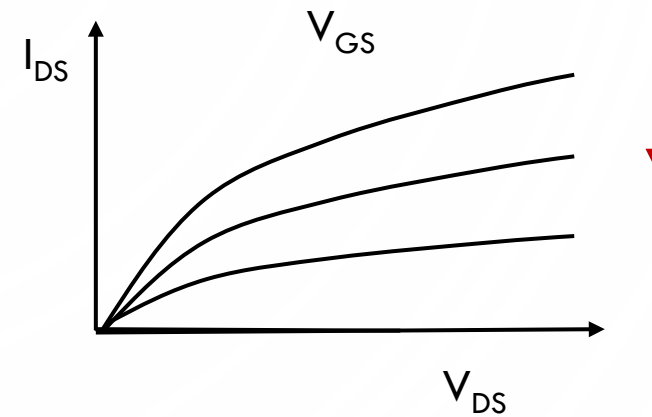
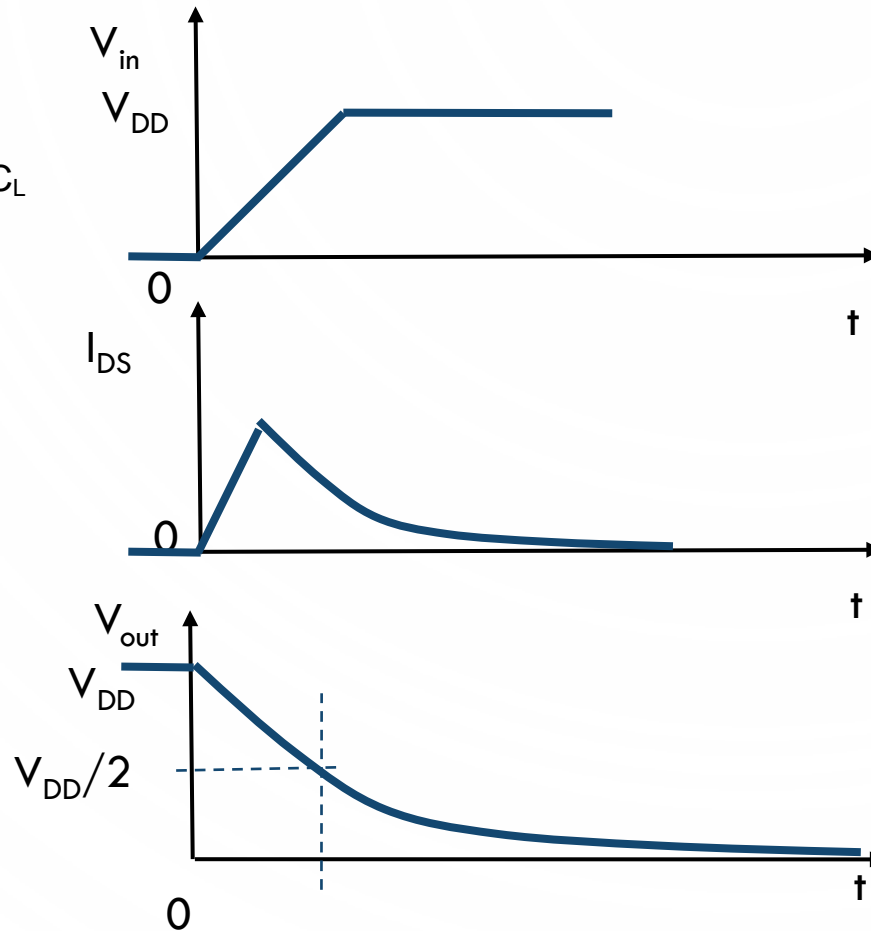
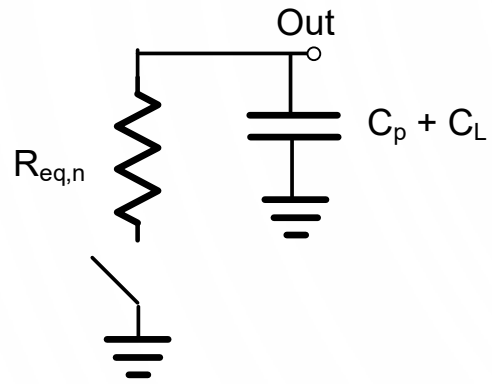
- Impacts the  $I_{DS}$ - $V_{DS}$  trajectory





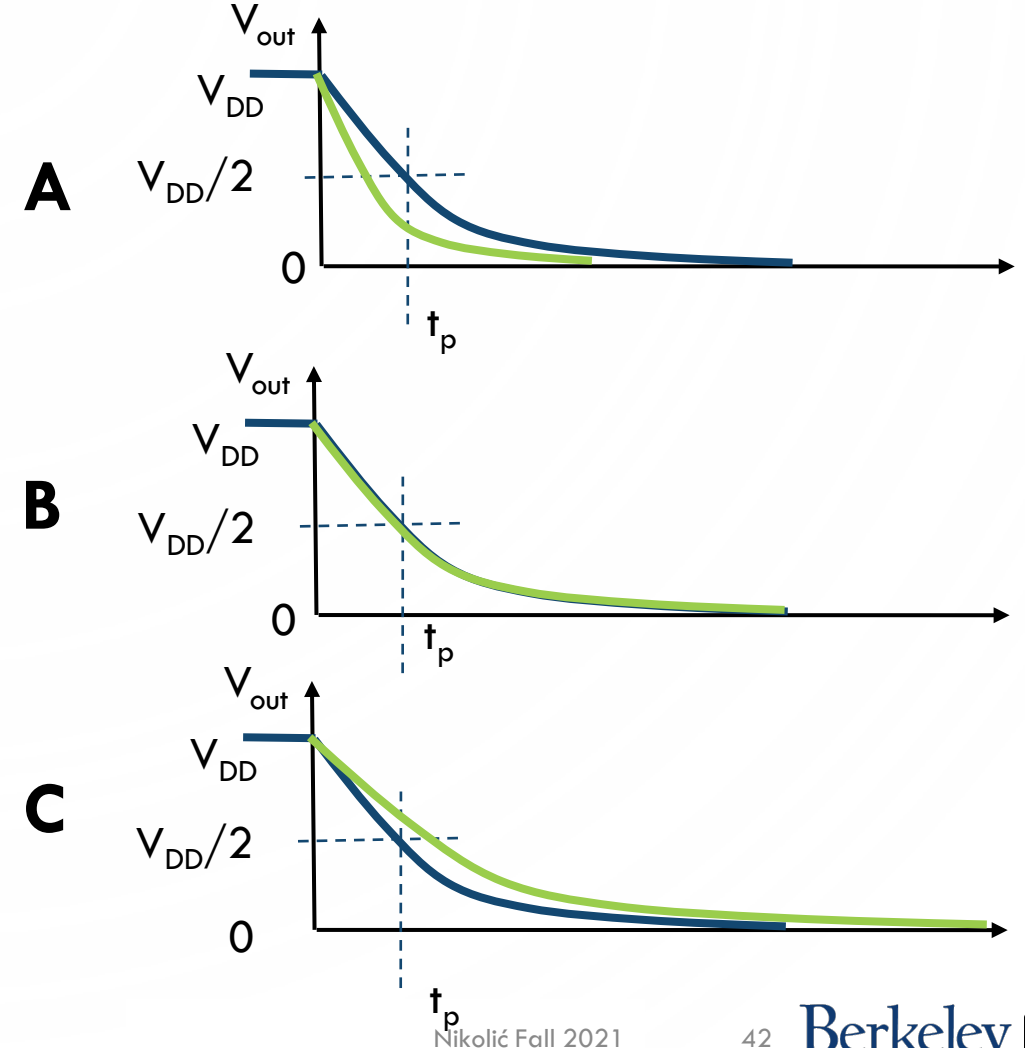
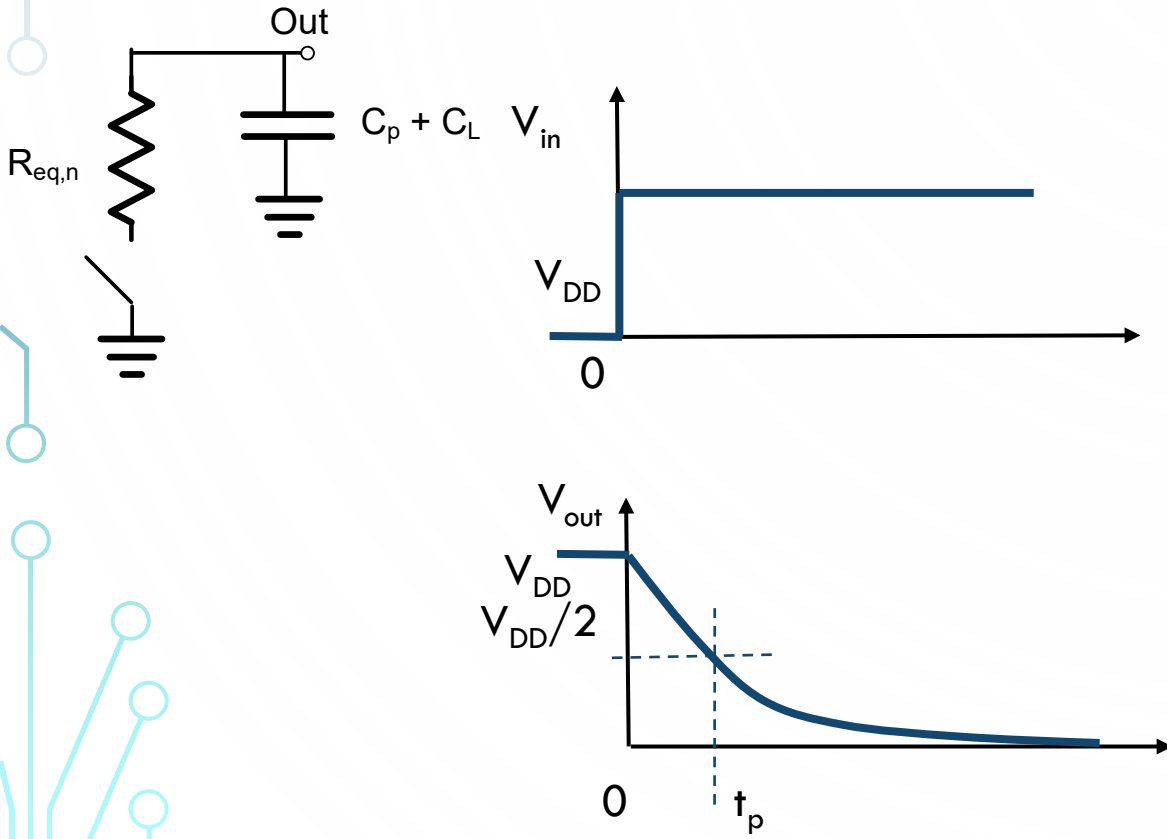
# Impact of Supply Voltage

- Lowering  $V_{DD}$ , slows down the circuit



# Quiz: Inverter Delay

- If we double the load capacitance, assuming the default  $V_{out}$  shown in blue, which of the following waveforms shows the new  $V_{out}$ ?



## Summary

- CMOS allows for convenient switch level abstraction
- CMOS pull-up and pull-down networks are complementary
  - Graph models for CMOS gates
- Transistor sizing affects gate performance
- Delay is a linear function of  $R$  and  $C$