EECS151 : Introduction to Digital Design and ICs

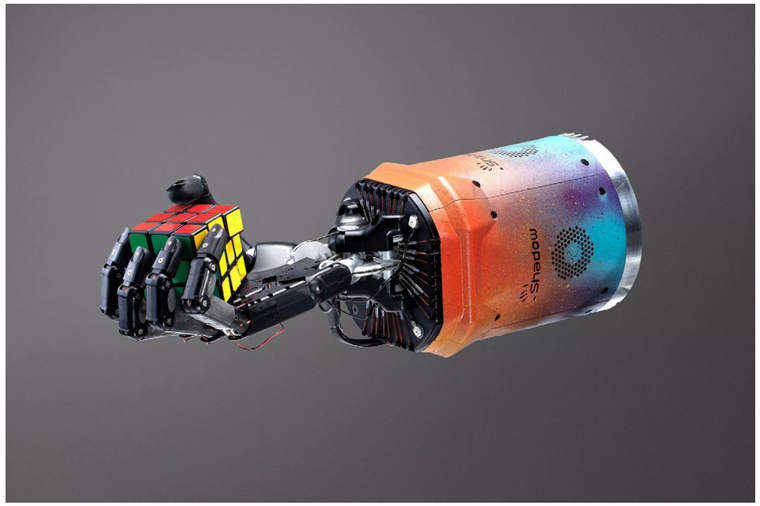
Lecture 12 – Delays

Bora Nikolić and Sophia Shao

OpenAI's AI-powered robot learned how to solve a Rubik's cube one-handed

Oct. 15, 2019, Artificial intelligence research organization OpenAI has achieved a new milestone in its quest to build general purpose, self-learning robots. The group's robotics division says Dactyl, its humanoid robotic hand first developed last year, has learned to solve a Rubik's cube one-handed.

The Verge, Oct. 15, 2019.



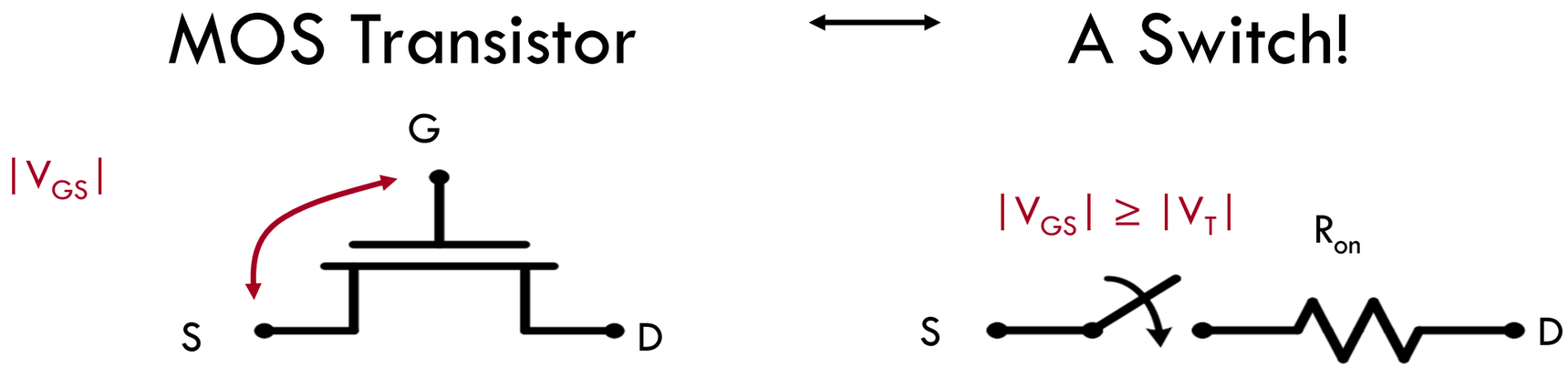
OpenAI

Review

- Modern lithography is reflected in design rules
- FinFET widths are quantized
- MOS transistors can be modeled as resistive switches
  - They also have a capacitance

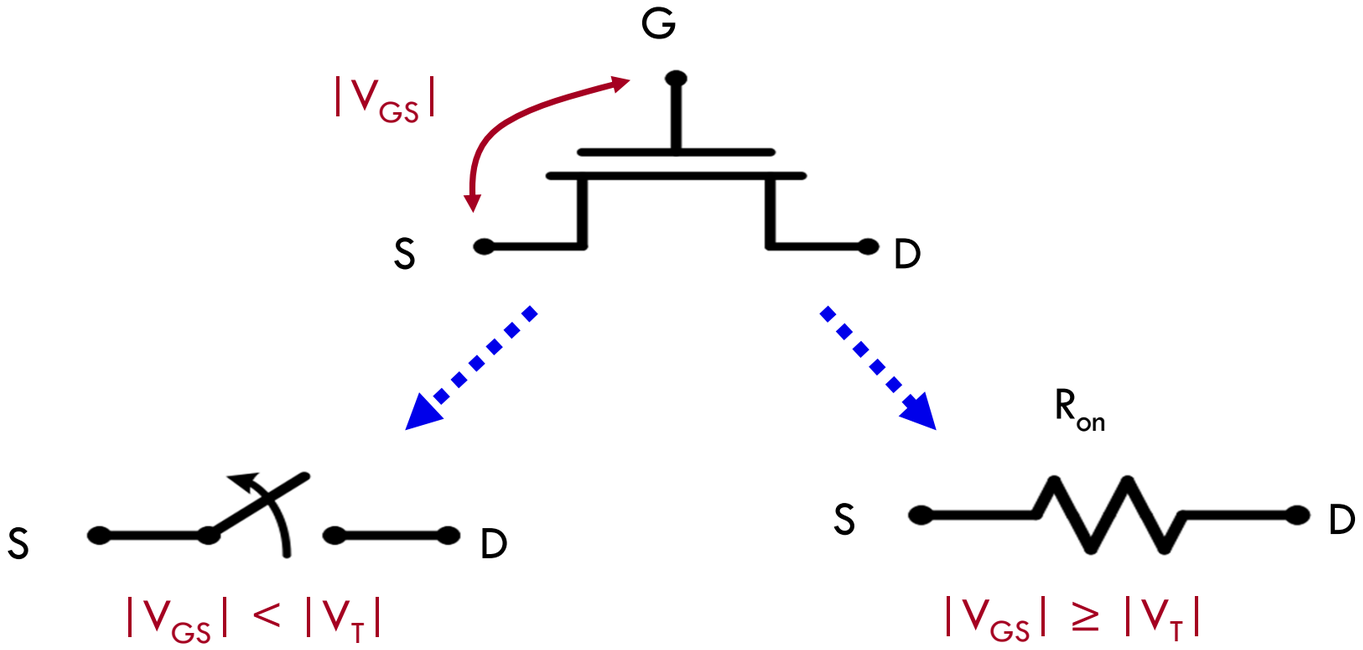
MOS Transistor as a Switch

MOS Transistor as a Resistive Switch

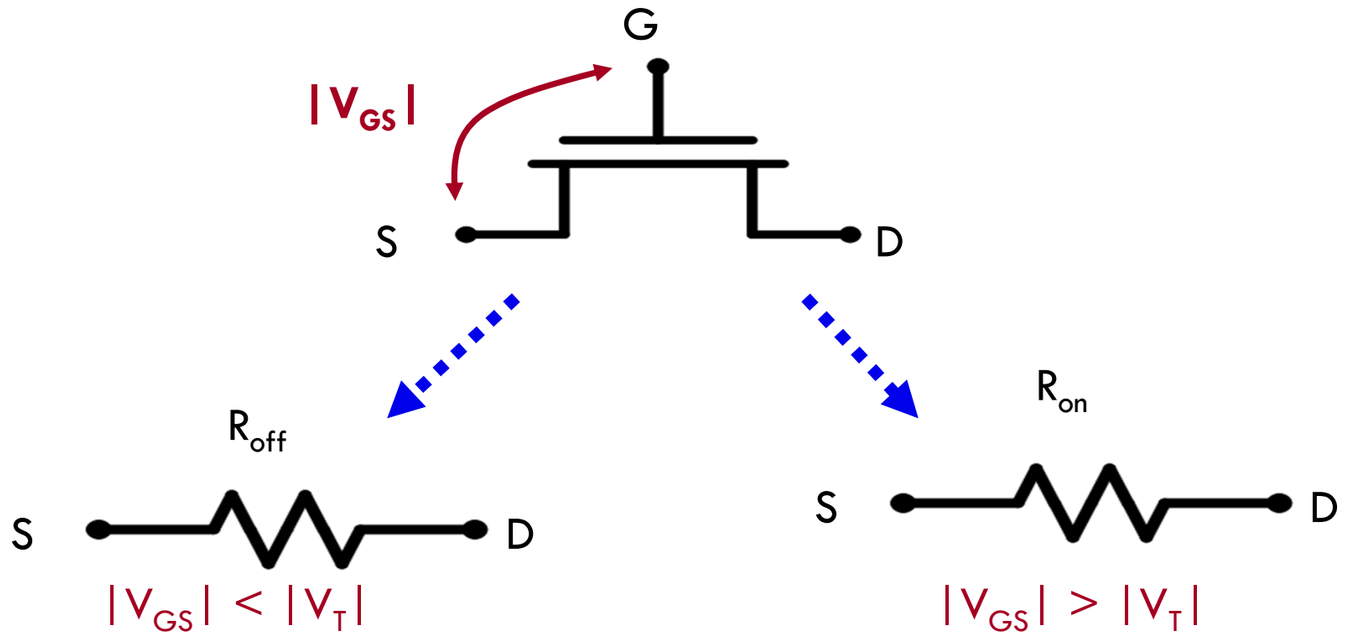


- V_{GS} controls the switch
  - (it also charges the channel capacitor)

ON/OFF Switch Model of MOS Transistor



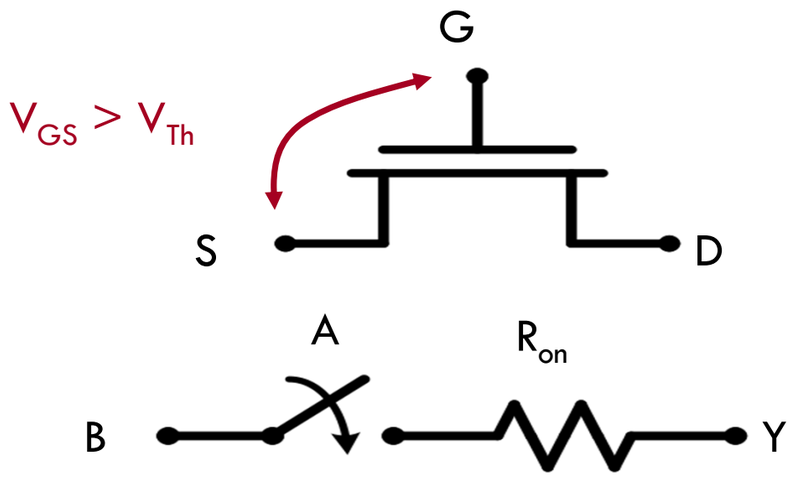
A More Realistic Model



- It is a dimmer!

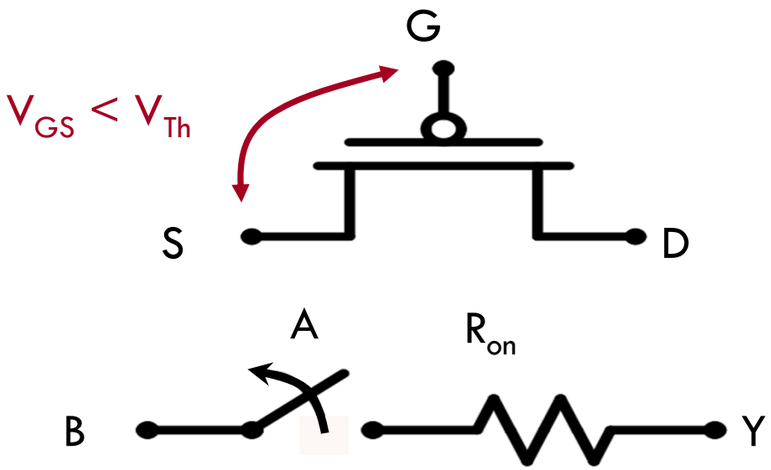
A Logic Perspective

NMOS Transistor



Y = B if A = 1

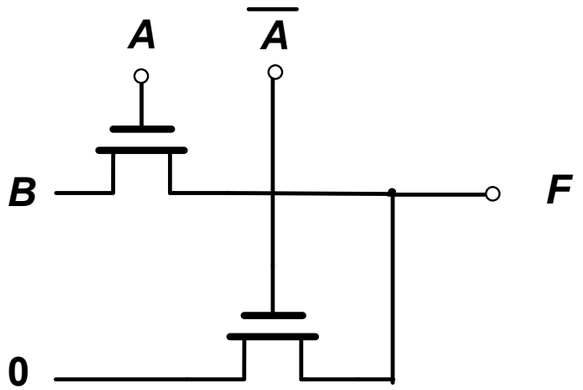
PMOS Transistor



Y = B if A = 0

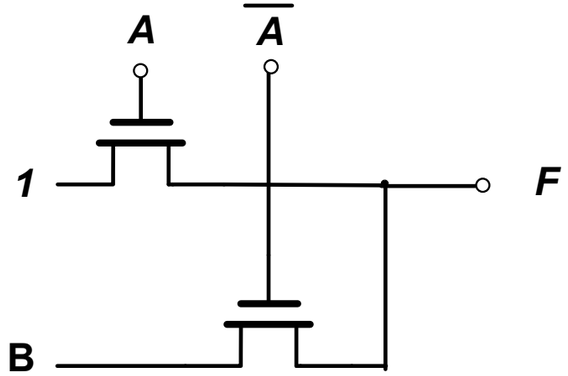
AND and OR

- AND



F = AB
(F = AB + $\overline{A}$·0)

- OR

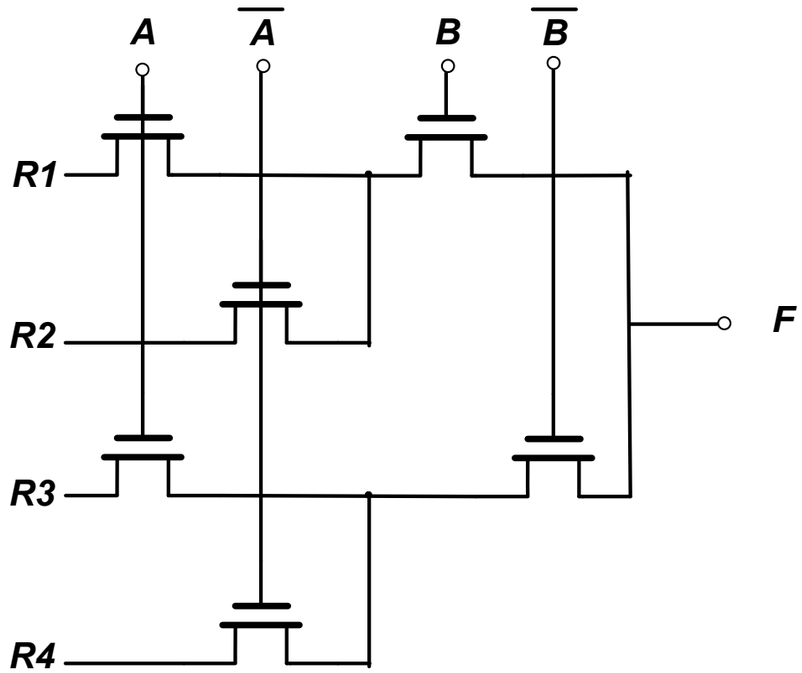


F = A+B
(F = A·1 + $\overline{A}$B)

- Keep in mind – single NMOS/PMOS transistors are imperfect switches!
  - Turns off when $|V_{GS}| = |V_{Th}|$

Peer Instruction

- Switch logic
- Which combination of inputs implements F = AB?

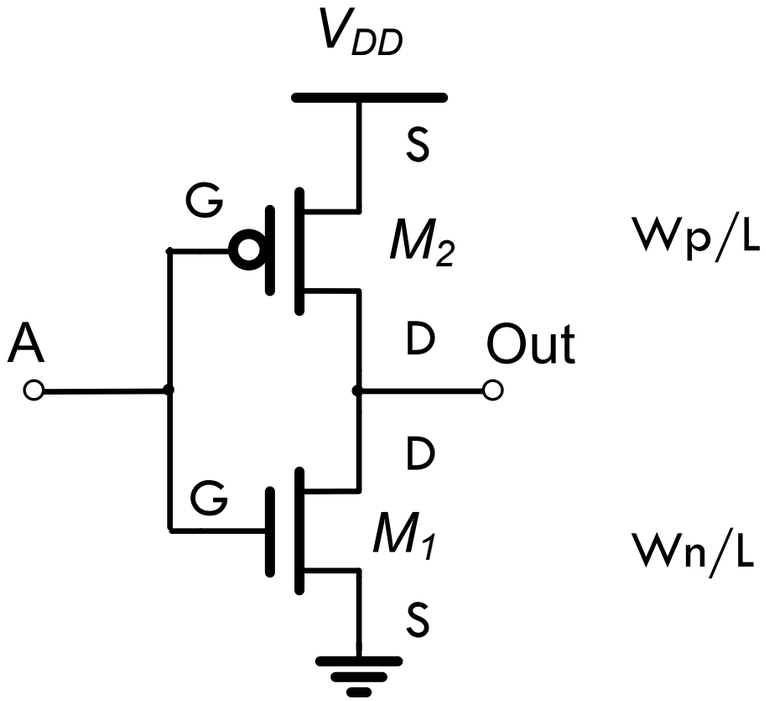


	R1	R2	R3	R4
a)	1	X	X	X
b)	0	X	X	X
c)	1	0	0	0
d)	1	1	1	0
e)	1	1	1	1
f)	None of the above			

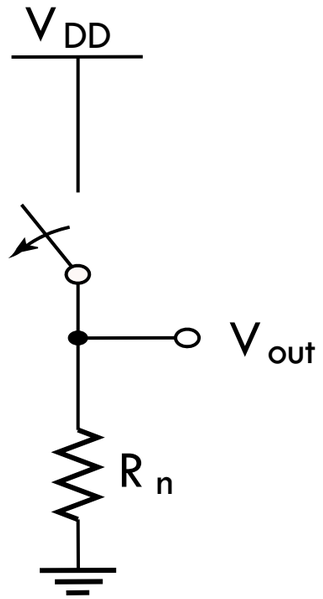
CMOS Inverter

CMOS Inverter

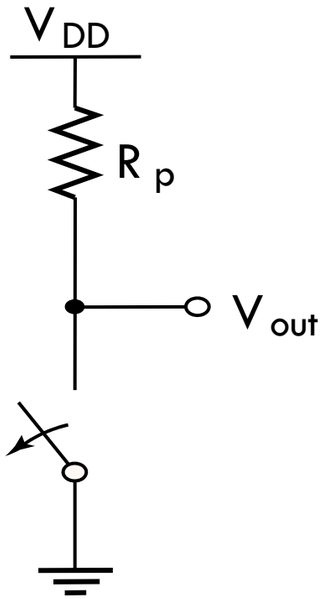
- Simple DC behavior
  - Schematic



- Switch model



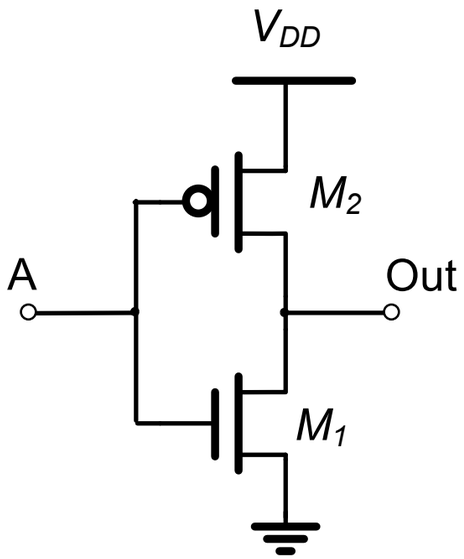
$V_{in} = V_{DD}$



$V_{in} = 0$

$V_{OL} = 0$
$V_{OH} = V_{DD}$

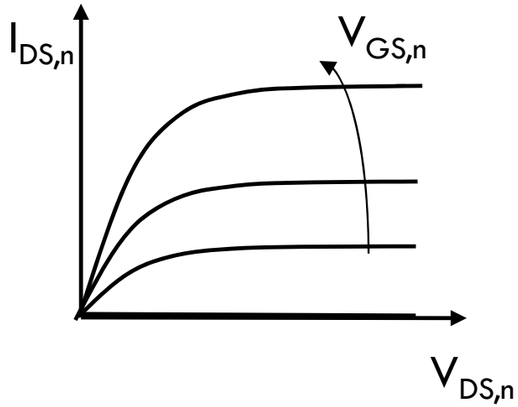
Voltage Transfer Characteristic (VTC)

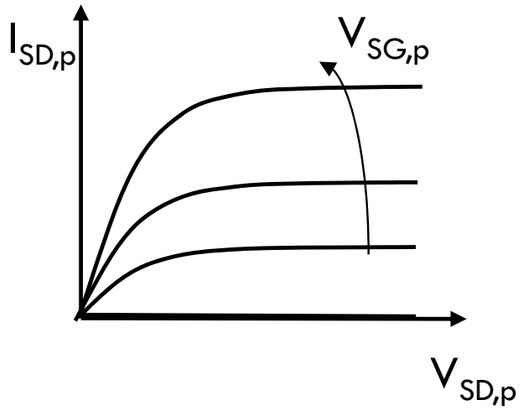


$$V_A = V_{DD} + V_{GSp}$$

$$I_{Dn} = -I_{Dp}$$

$$V_{out} = V_{DD} + V_{DSp}$$

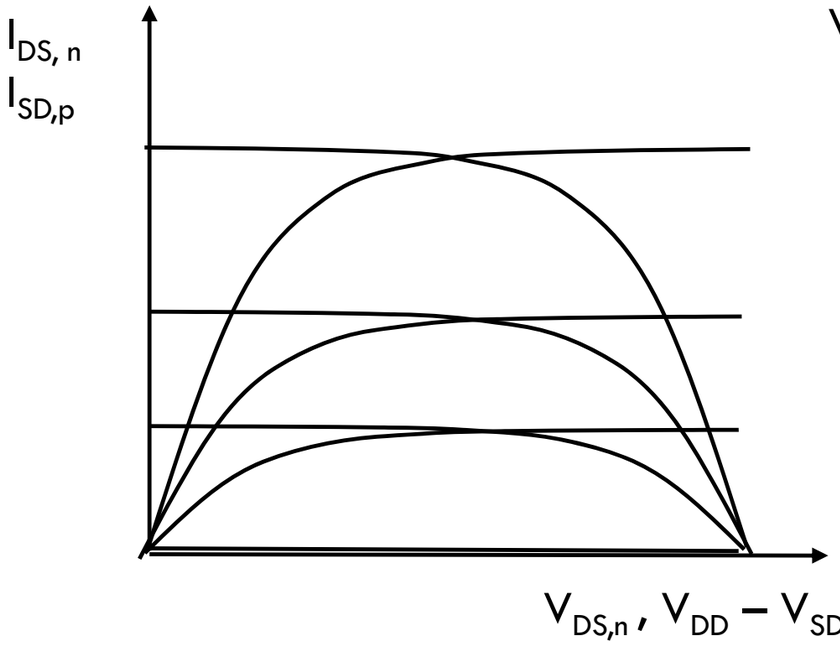


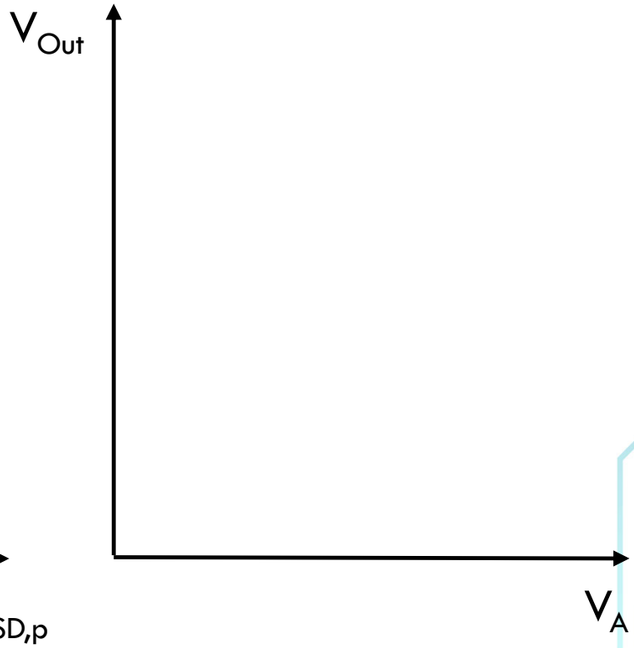


$$V_A = V_{GS,n} = V_{DD} - V_{SG,p}$$

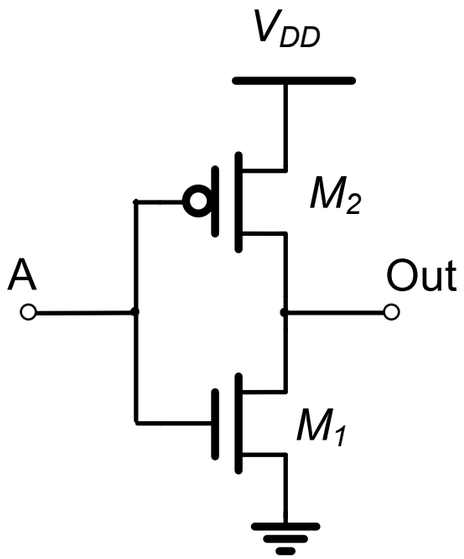
$$I_{DS,n} = I_{SD,p}$$

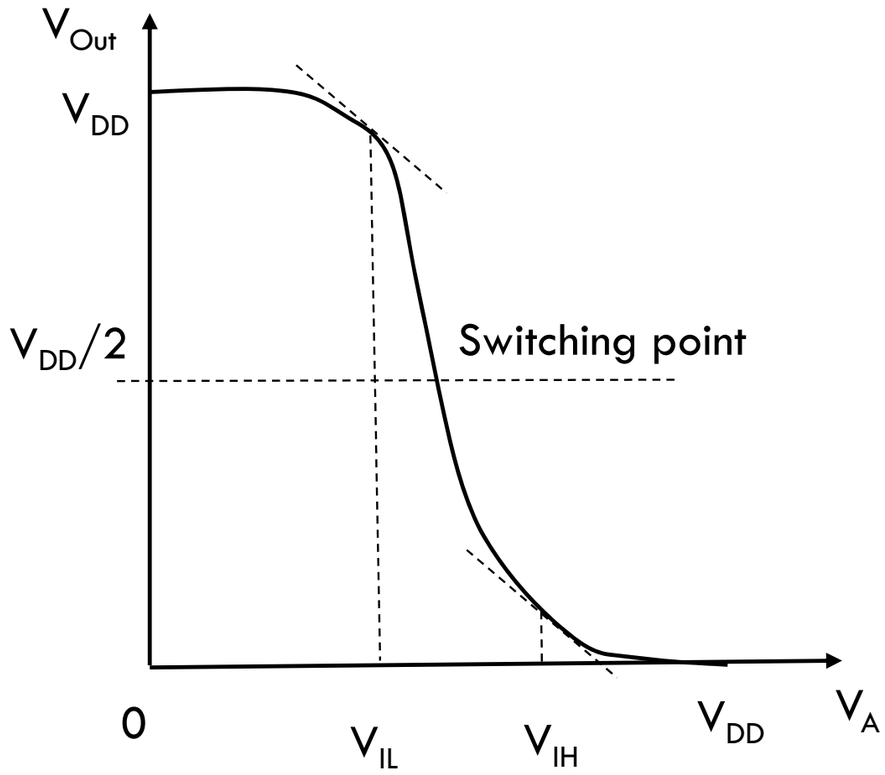
$$VOut = V_{DS,n} = V_{DD} - V_{SD,p}$$





Voltage Transfer Characteristic (VTC)





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

Digital Circuits

- One logic representation

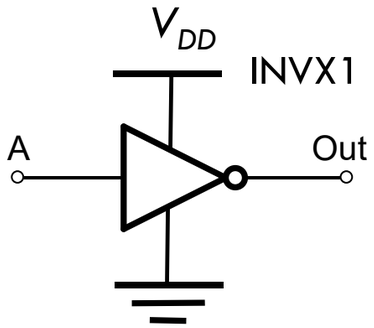
$$Out = \overline{A}$$

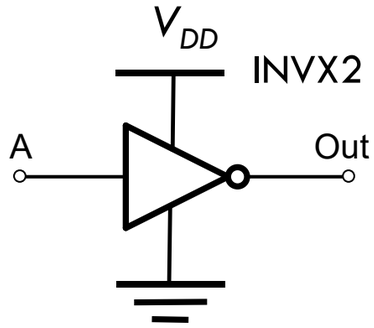
Truth table

A	Out
0	1
1	0

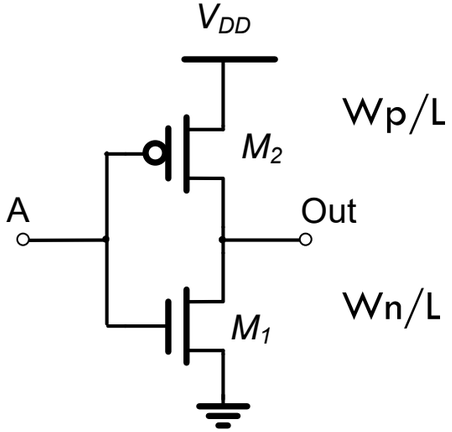
- Multiple libraries
- Layouts
  - Number of metal 'tracks'
  - More tracks, faster, but larger
  - Less tracks – more compact, but slower
- Transistor thresholds (V_{Th}) (for each track height):
  - Regular (RVT)
  - Low (LVT)
    - Faster, higher power
    - Slower, lower power
  - High (HVT)
- Transistor lengths

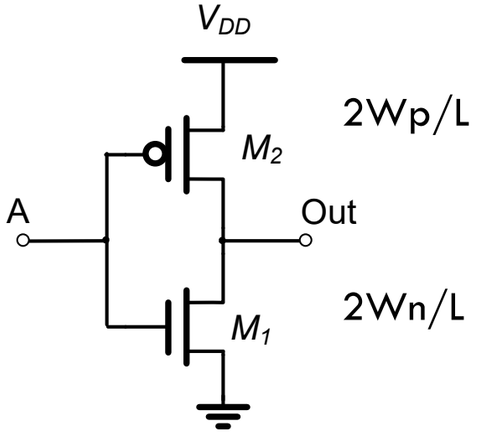
- Multiple gate sizes within a library

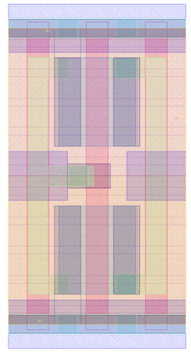


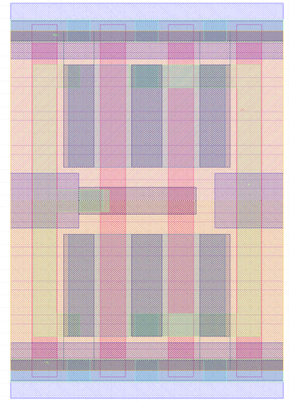


INVX3, INVX4,…









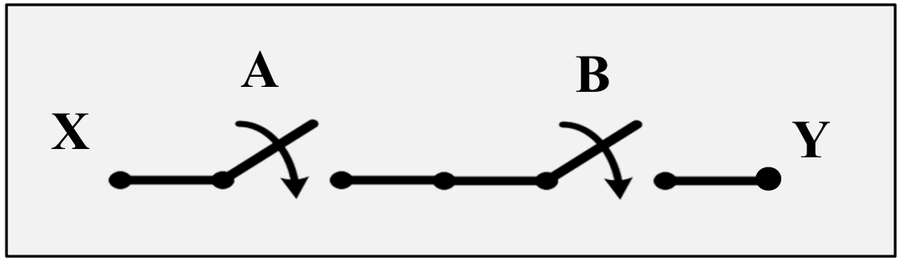
Administrivia

- Use this week to finish labs!
- Midterm 2 in 3 weeks
- Bora is away next week

CMOS Logic

Building logic from switches

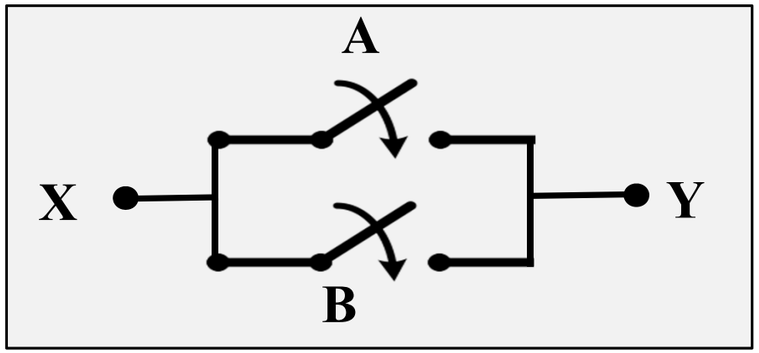
Series



AND

Y = X if A AND B

Parallel



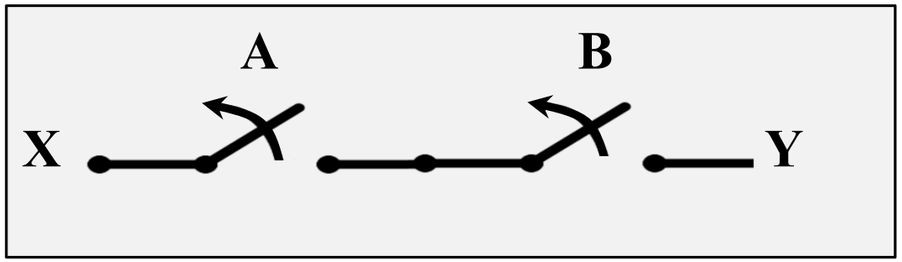
OR

Y = X if A OR B

(output undefined if condition not true)

Logic using inverting switches

Series

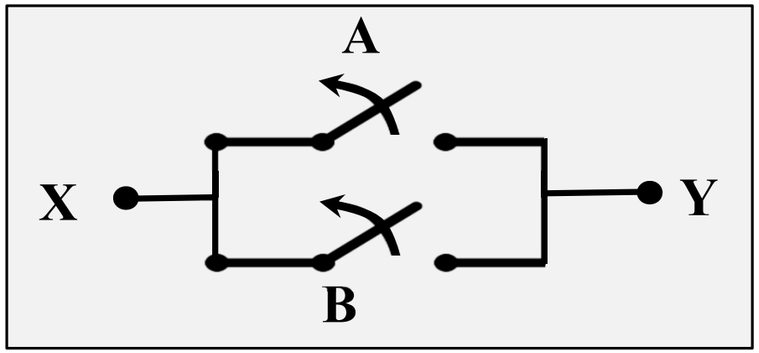


NOR

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

$= \overline{A + B}$

Parallel



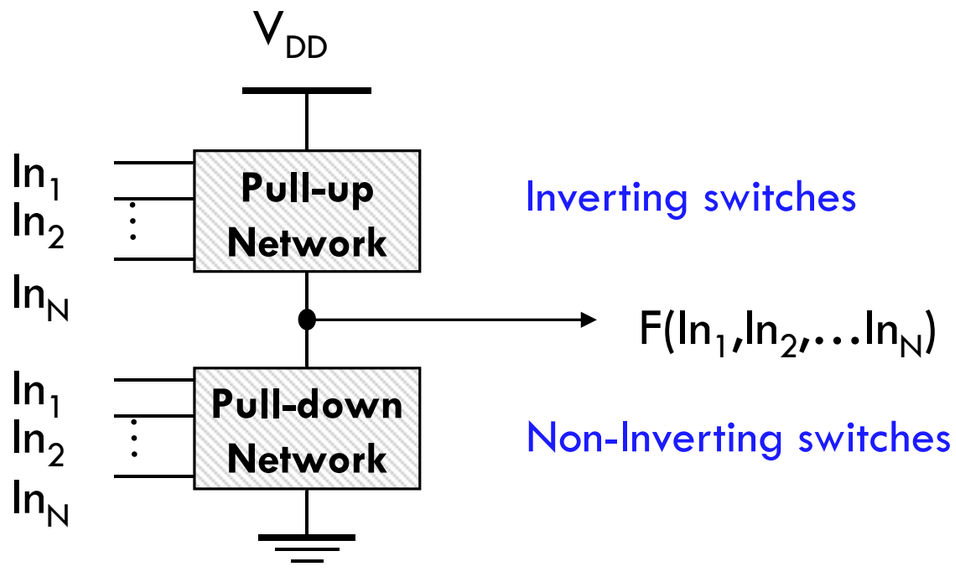
NAND

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

$= \overline{AB}$

(output undefined if condition not true)

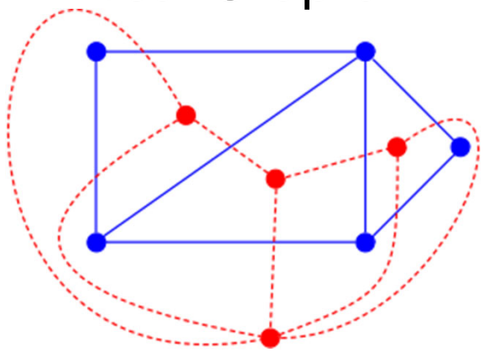
Static Complementary CMOS



PUN and PDN are dual logic networks

PUN and PDN functions are complementary

Dual Graphs



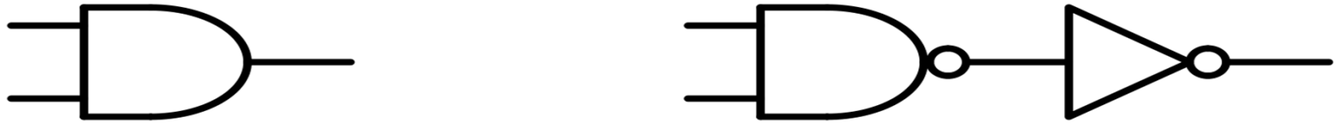
Complementary CMOS Logic Style

- 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

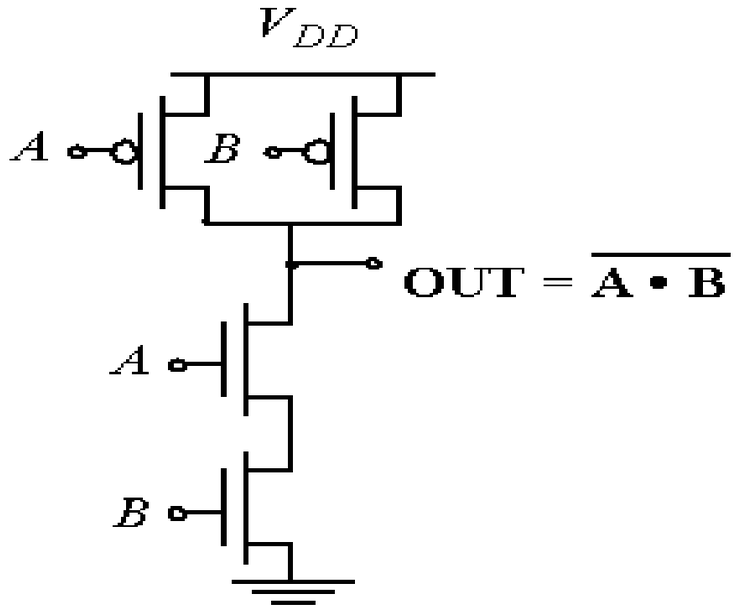


AND = NAND + INV

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

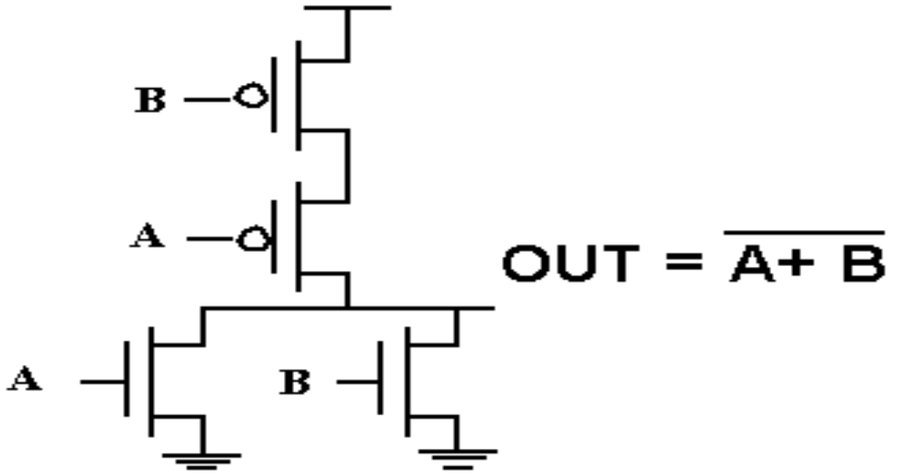


- PDN: G = AB $\Rightarrow$ Conduction to GND
- PUN: F = $\overline{A}$ + $\overline{B}$ = AB $\Rightarrow$ Conduction to V_{DD}
- $\overline{G(In_1, In_2, In_3, \ldots)} \equiv F(\overline{In_1}, \overline{In_2}, \overline{In_3}, \ldots)$

Example Gate: NOR

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

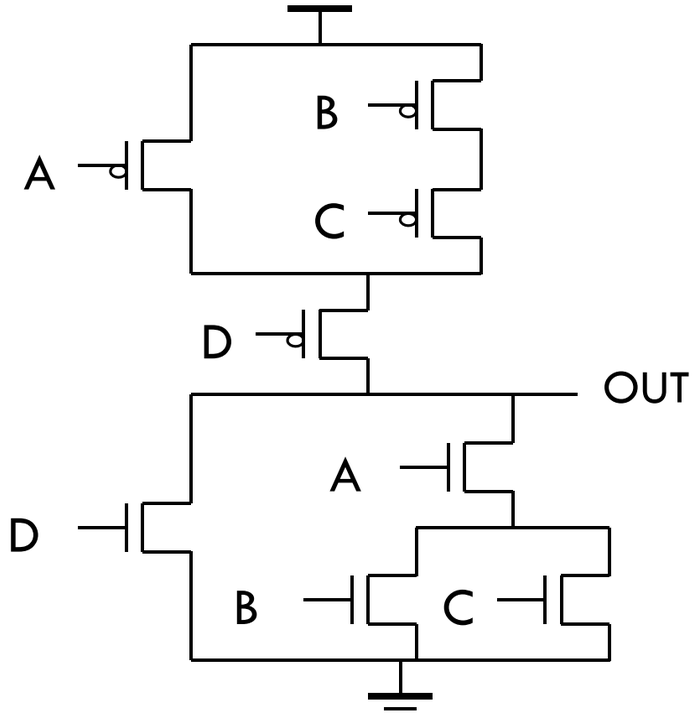
Truth Table of a 2 input NOR gate



Complex CMOS Gate

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

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

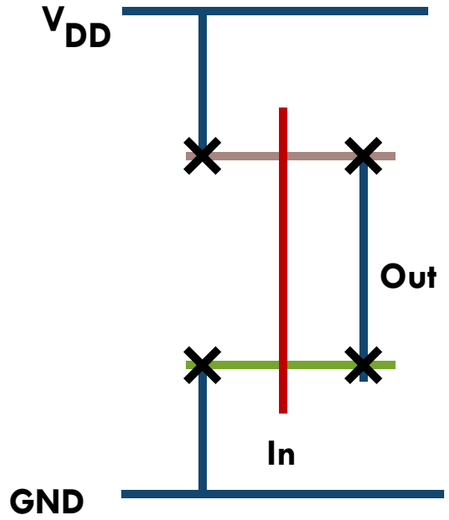


- Note: In scaled processes max #inputs is 3

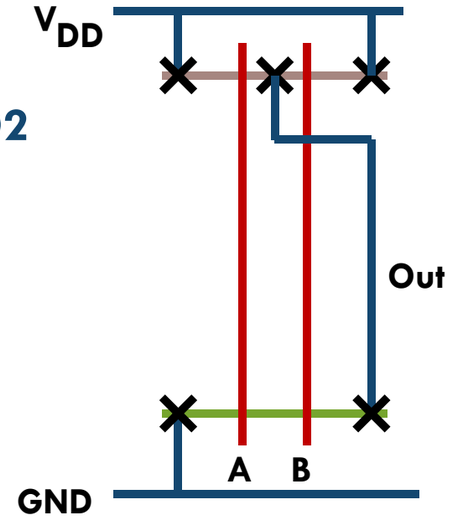
Stick Diagrams

Contains no dimensions
Represents relative positions of transistors

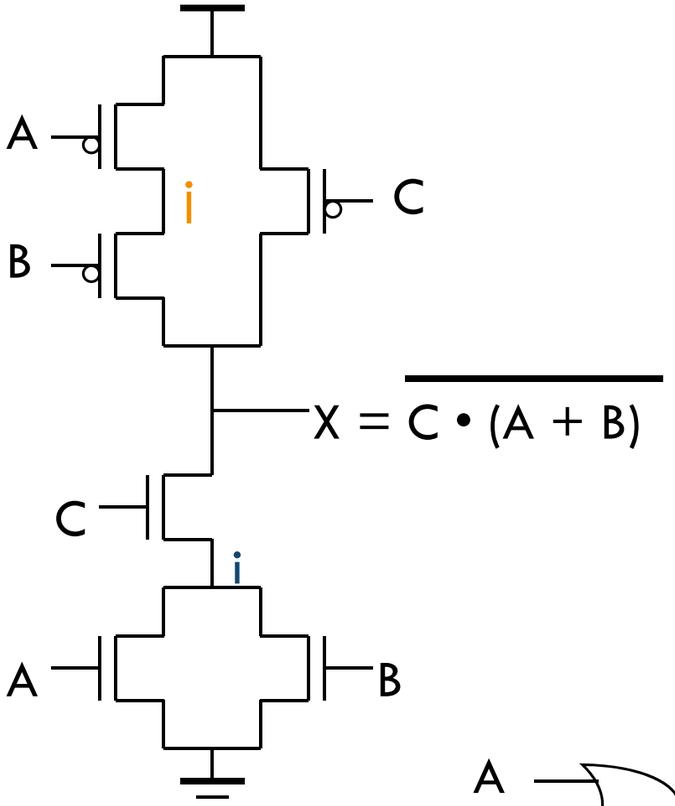
Inverter



NAND2

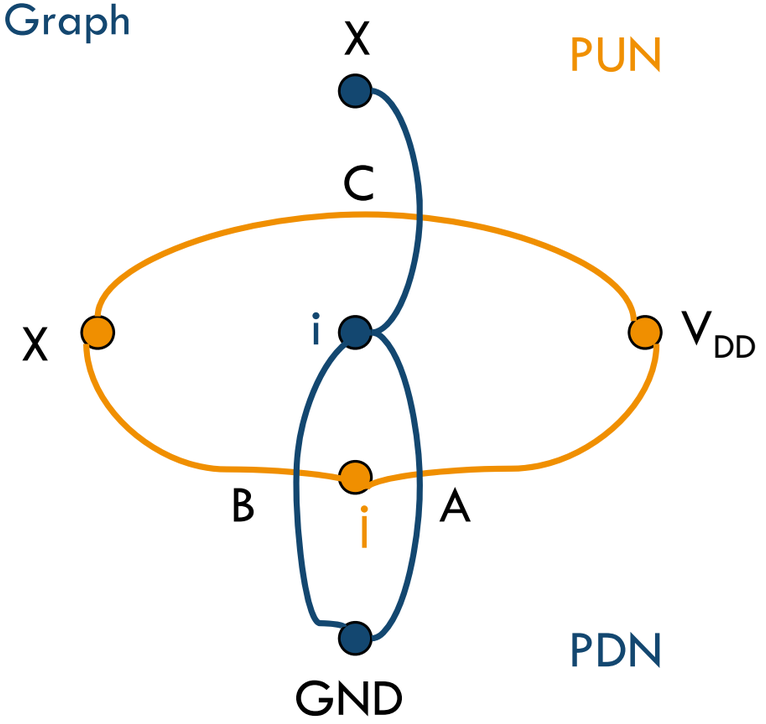


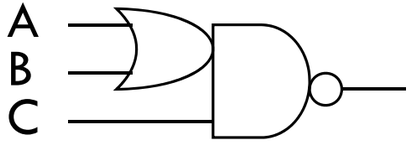
Stick Diagrams



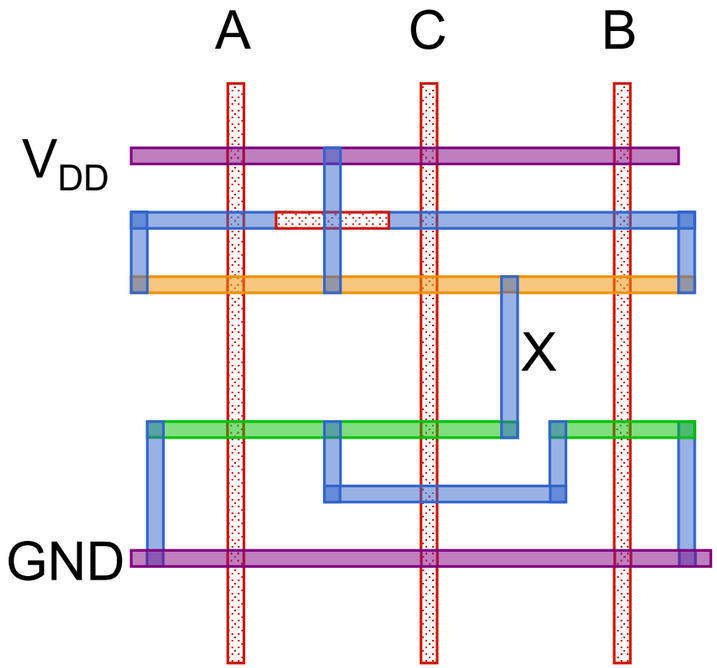
$$X = \overline{C \bullet (A + B)}$$

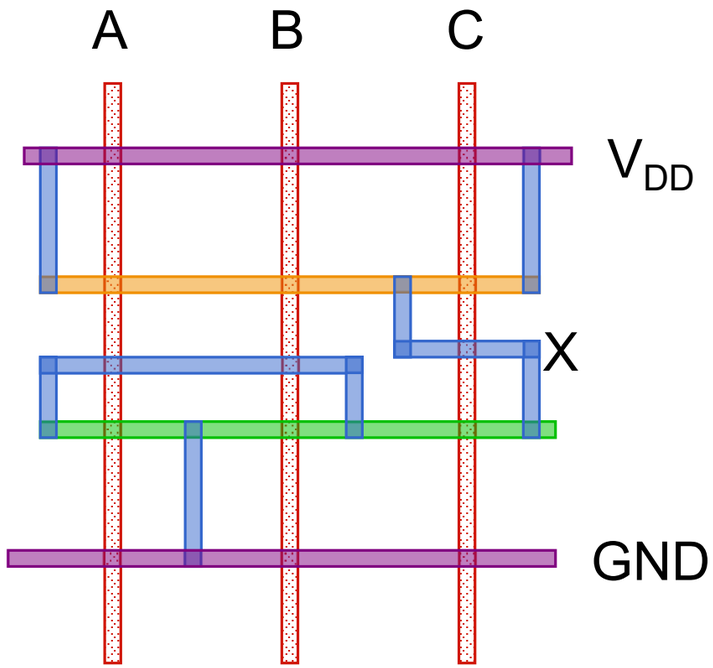
Logic Graph



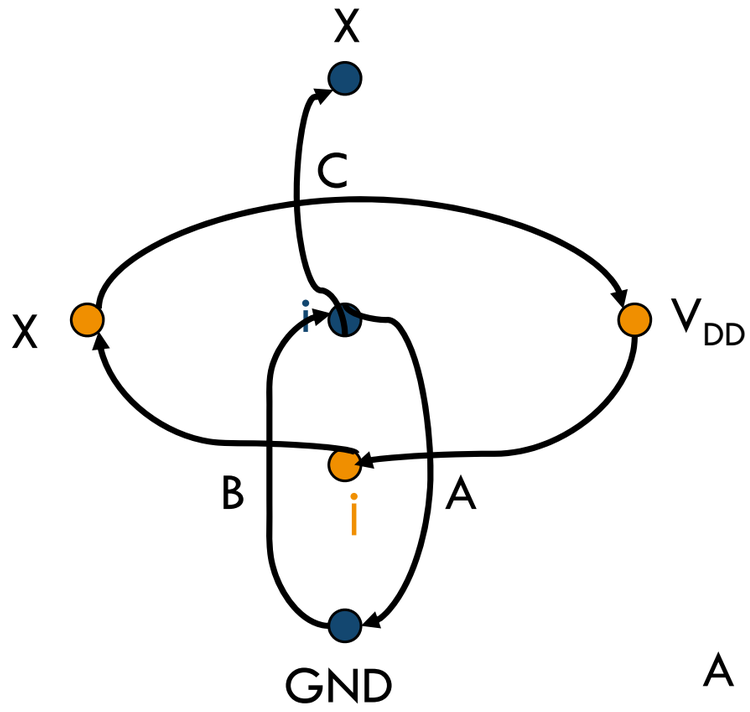


Two Versions of $\overline{C \bullet (A + B)}$



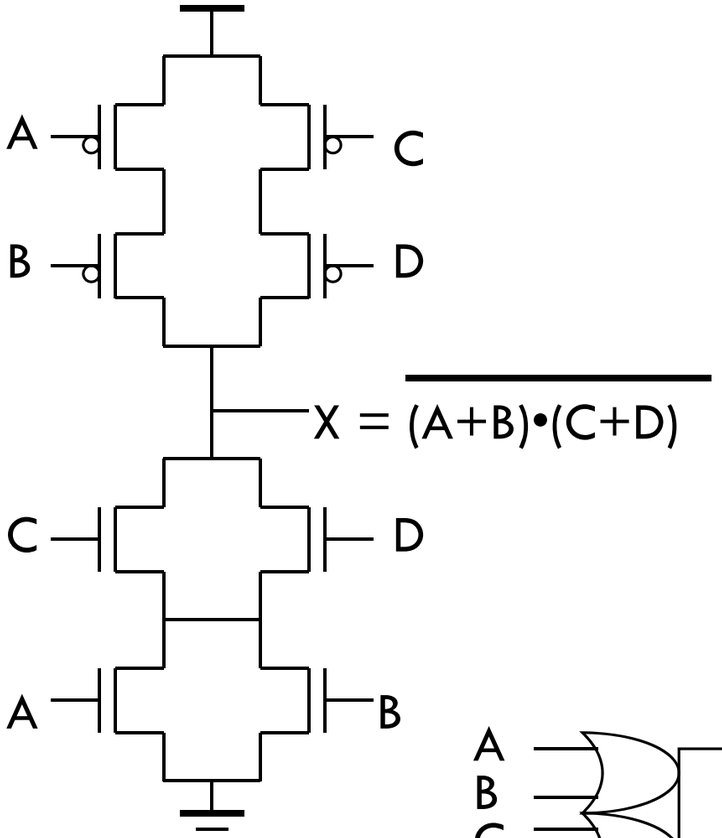


Consistent Euler Path

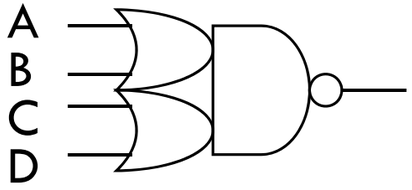


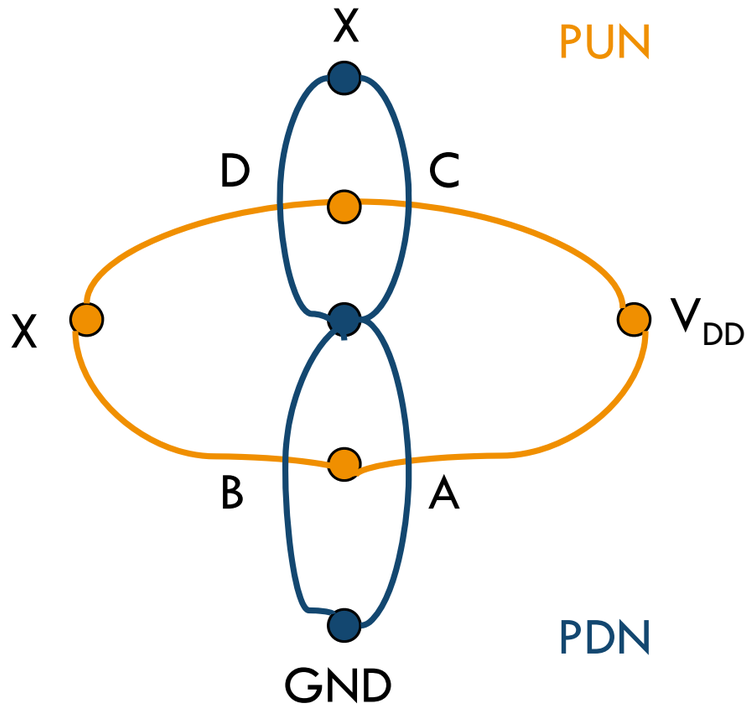
A B C

OAI22 Logic Graph

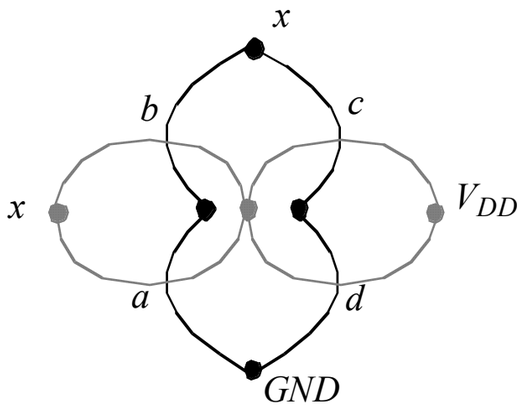


$$X = \overline{(A+B)\bullet(C+D)}$$

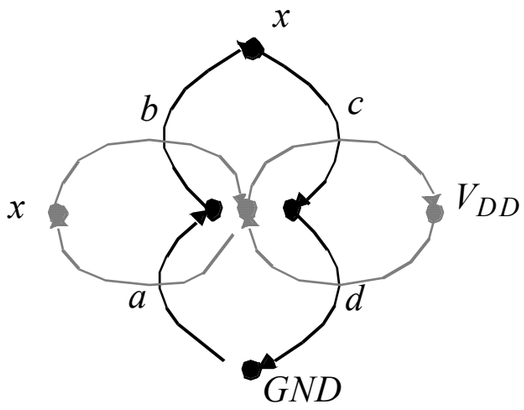




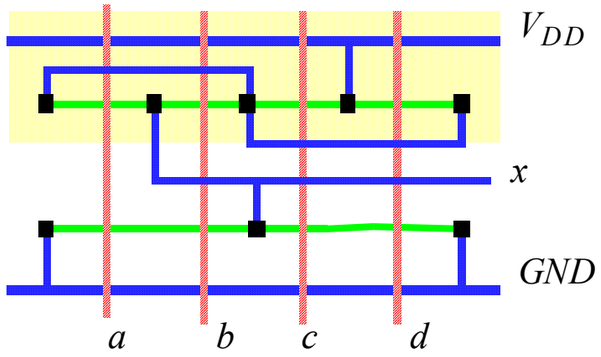
Example: x = $\overline{ab+cd}$



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

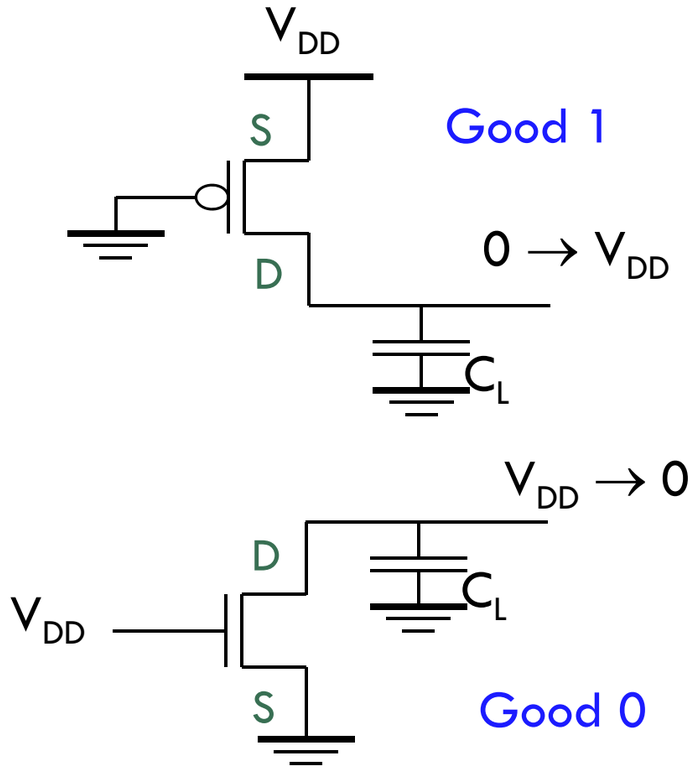


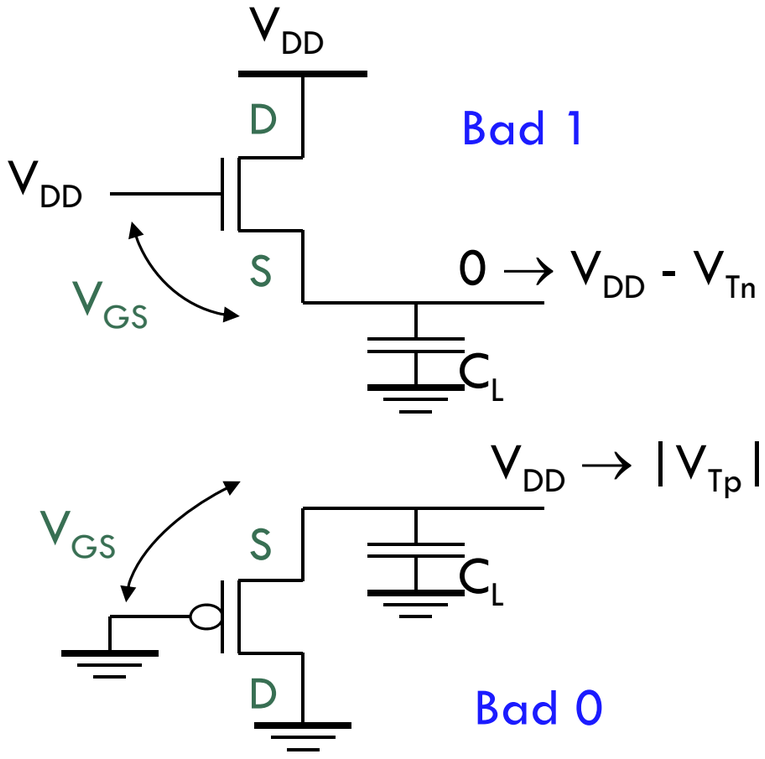
(b) Euler Paths {*a b c d*}



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

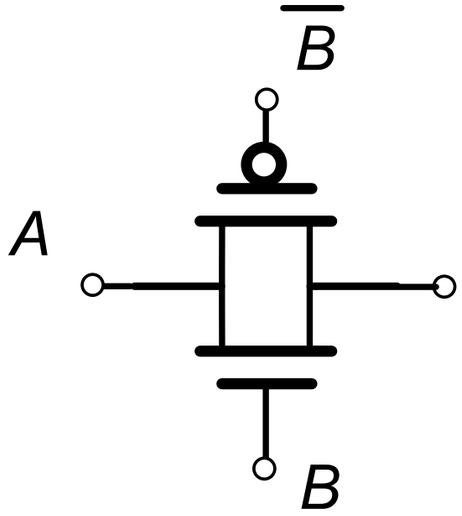
Switch Limitations





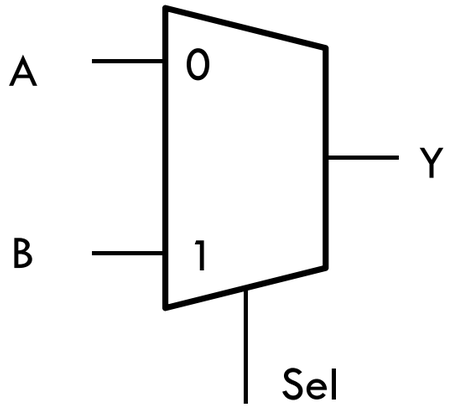
Transmission Gate

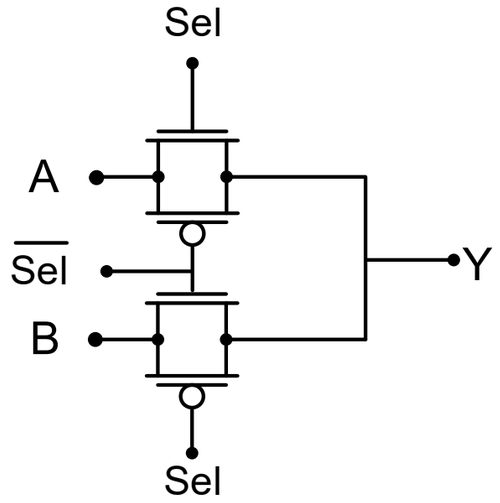
- Transmission gates are the way to build “switches” in CMOS.
- In general, both transistor types are needed:
  - ❑ nFET to pass zeros.
  - ❑ pFET to pass ones.
- The transmission gate is ‘non-isolating’.



Transmission-Gate Multiplexer

- Implementation





Sel	Y
0	A
1	B

```
module comb(input a, b, sel,
    output reg y);
  always @(*) begin
    if (sel) y = b;
    else out = a;
  end
endmodule
```

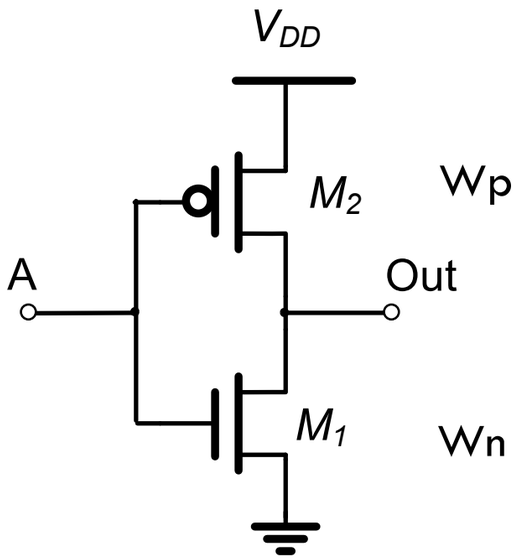
CMOS Multiplexer

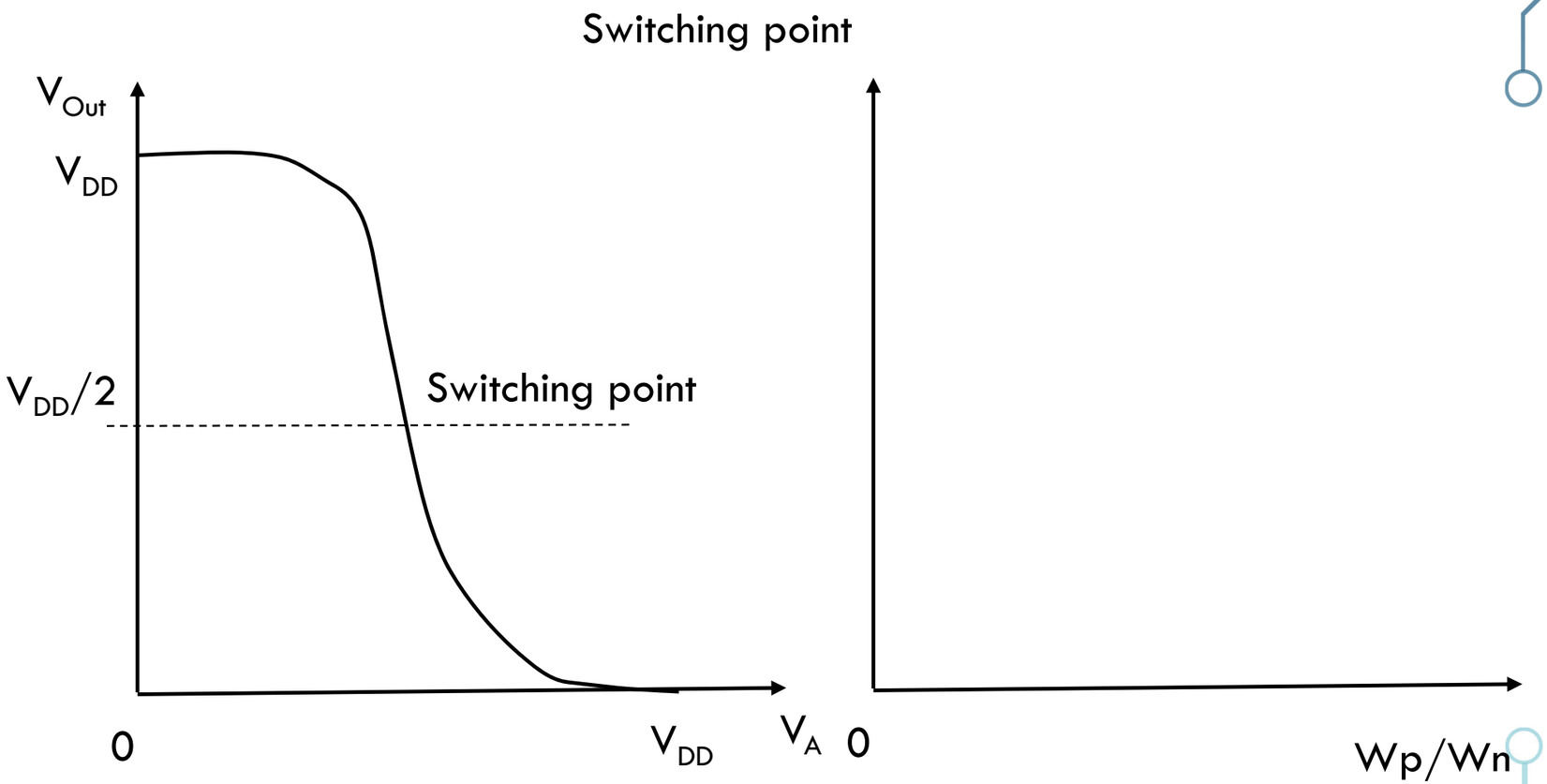
Sel	Y
0	A
1	B

CMOS Sizing

Transistor Sizing

- Optimal Wp/Wn

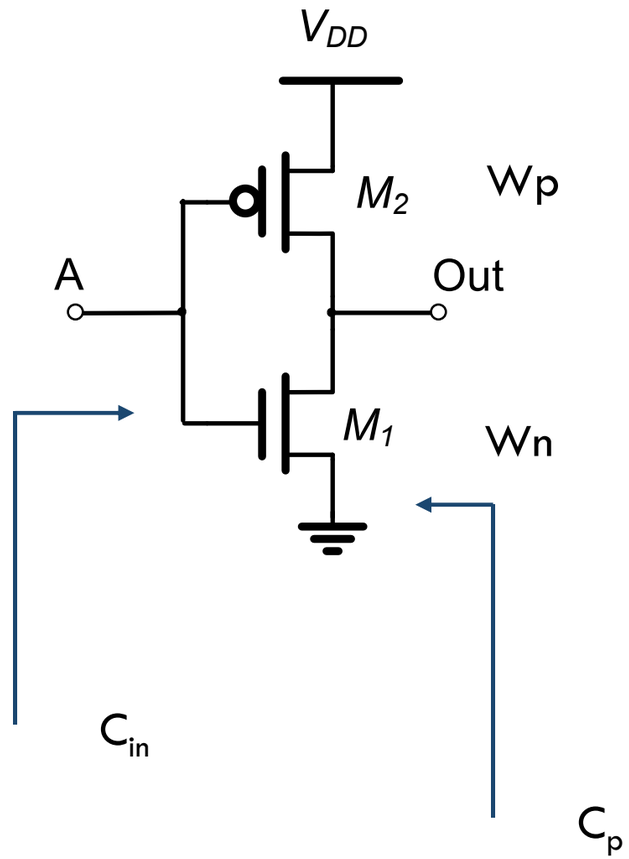




- In the past, Wp > Wn (see Rabaey, 2nd ed)
- In modern processes (finFET), Wp = Wn

Gate Sizing

- Doubling the gate size (by doubling Ws):

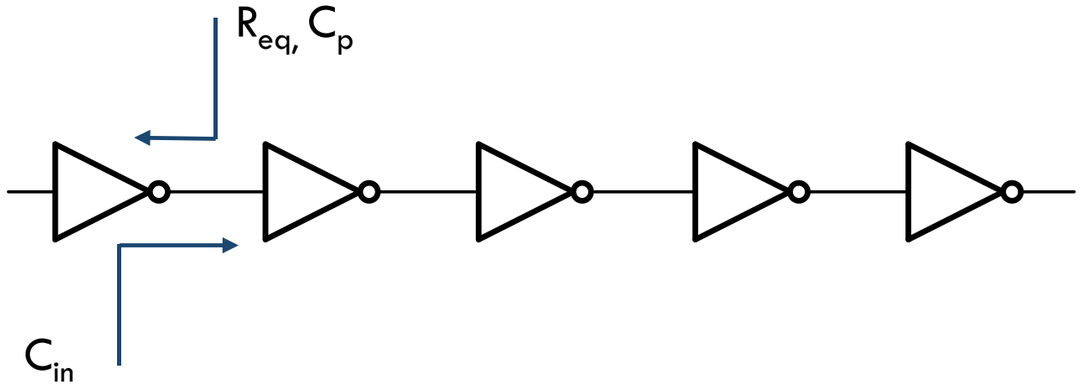


- Doubles C_{in}
- Halves equivalent gate resistance
- Doubles C_p

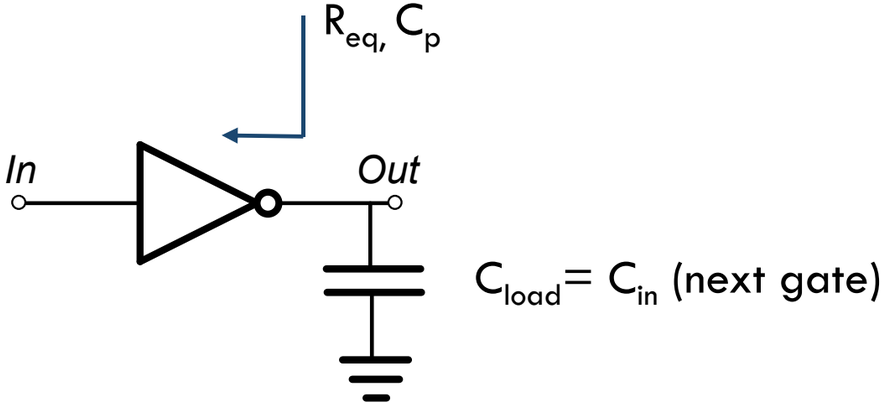
CMOS Delay

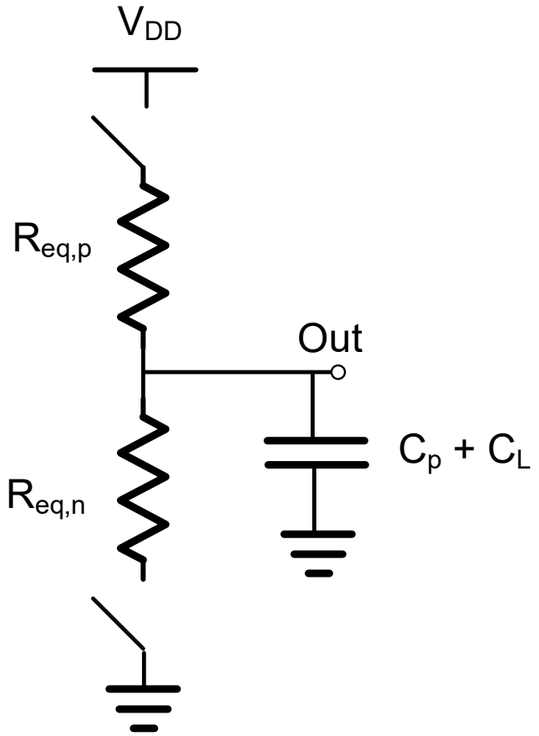
Inverter Delay

- How to time this?



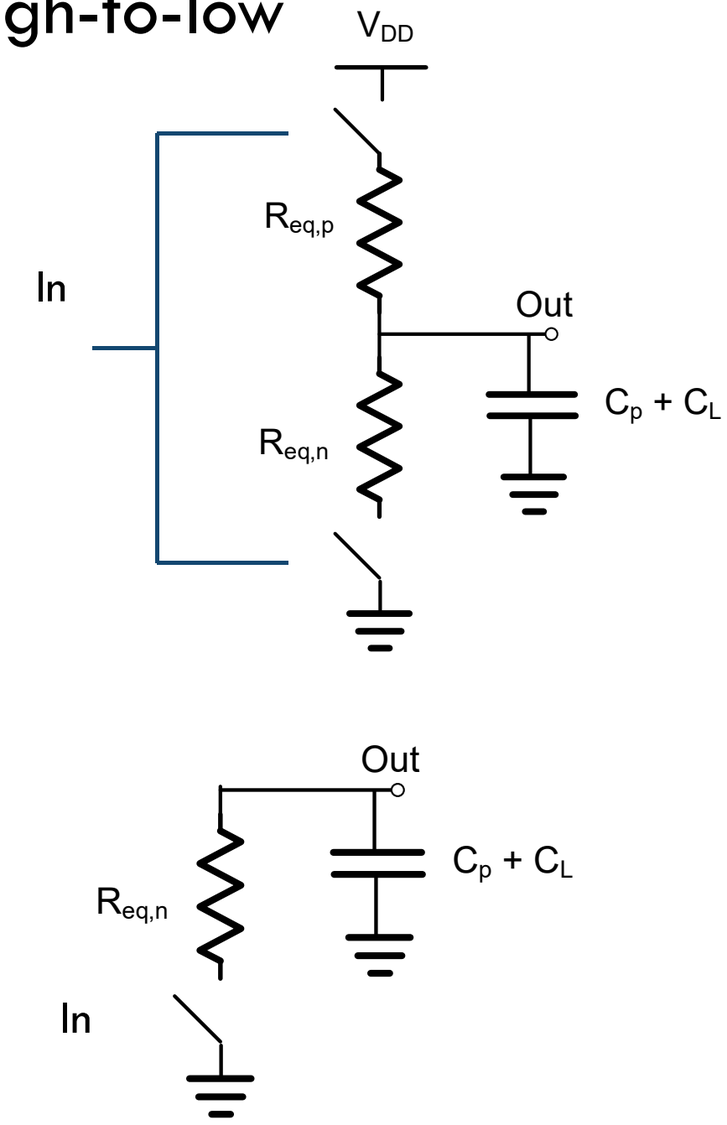
- Each gate has an R_{eq} and drives C_{in} of the next gate

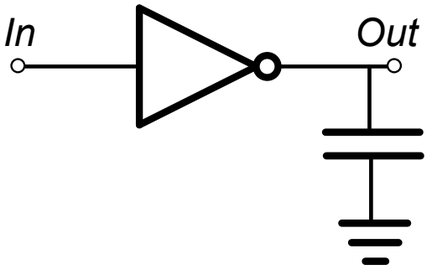


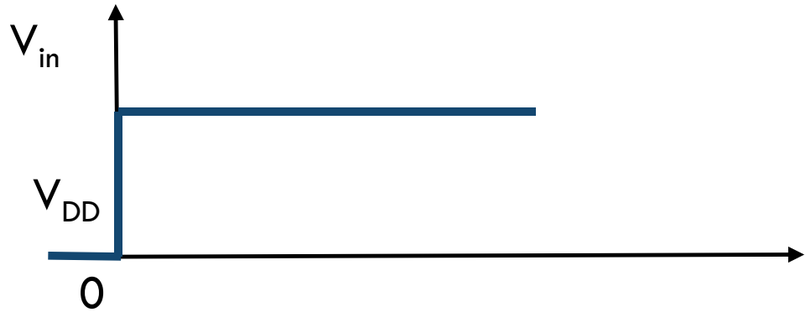


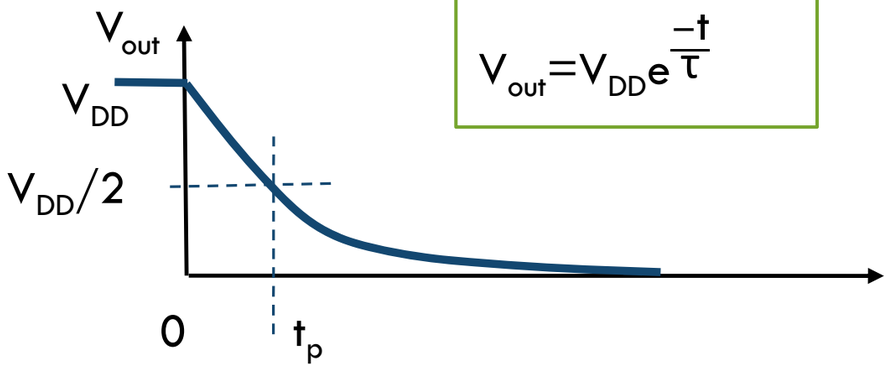
Inverter Delay

- High-to-low





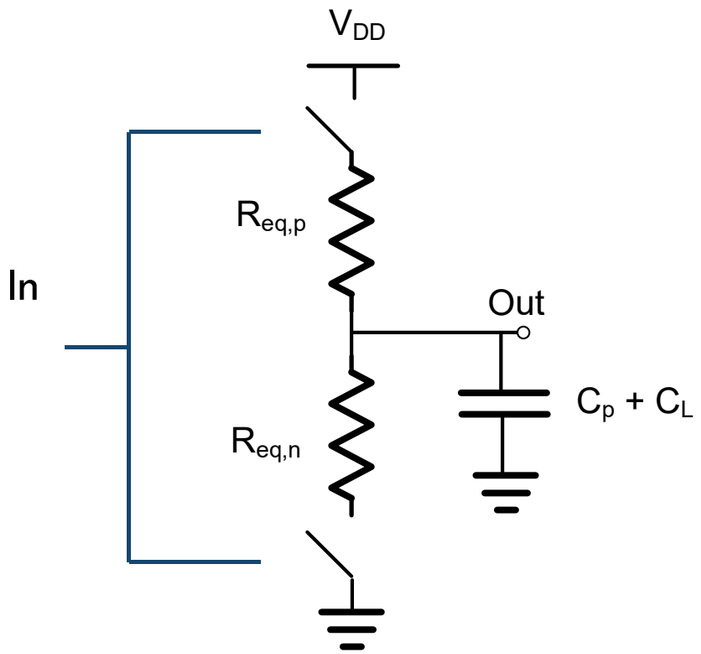


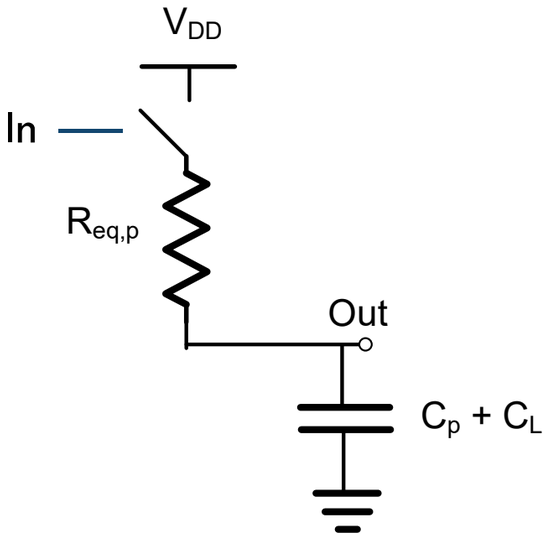


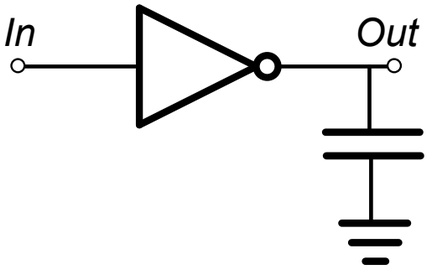
$$V_{out} = V_{DD} e^{\frac{-t}{\tau}}$$

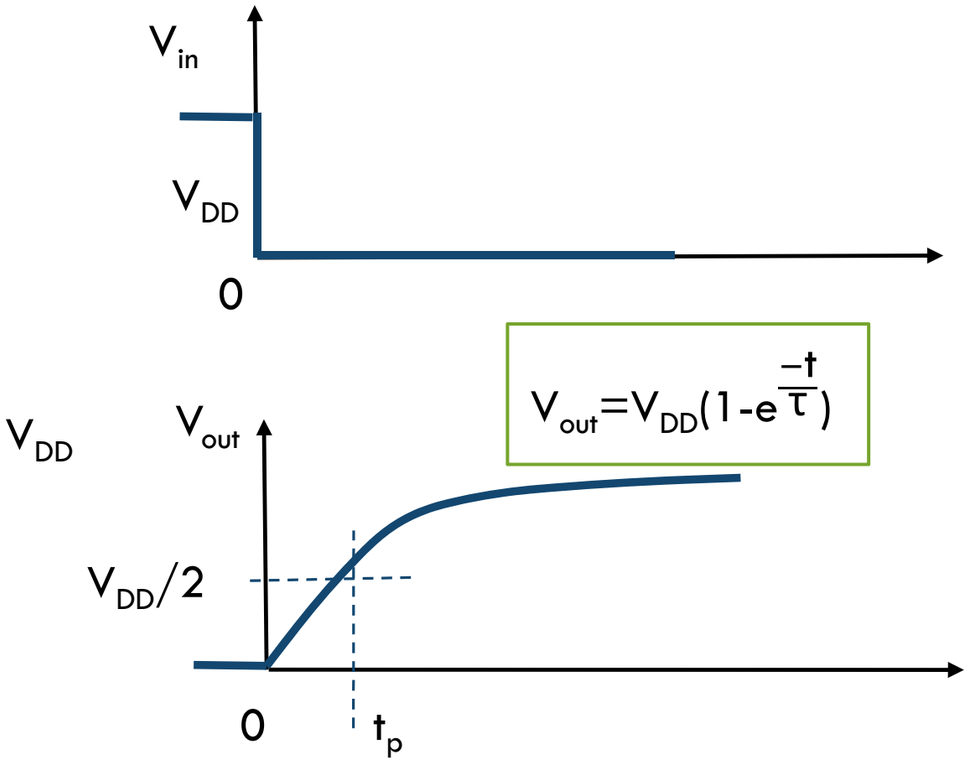
$$t_{p,HL} = (\ln 2)\tau = 0.7\, R_{eq,n}(C_p + C_L)$$

Inverter Delay









$$V_{out} = V_{DD}(1 - e^{\frac{-t}{\tau}})$$

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

Summary

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