Lecture21:
NMOS inverter (1)

Sung-Min Hong (smhong@gist.ac.kr)

Semiconductor Device Simulation Lab.
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology

GIST Lecture on June 1, 2020 (Internal use only)

Gain & input impedance (1/2)

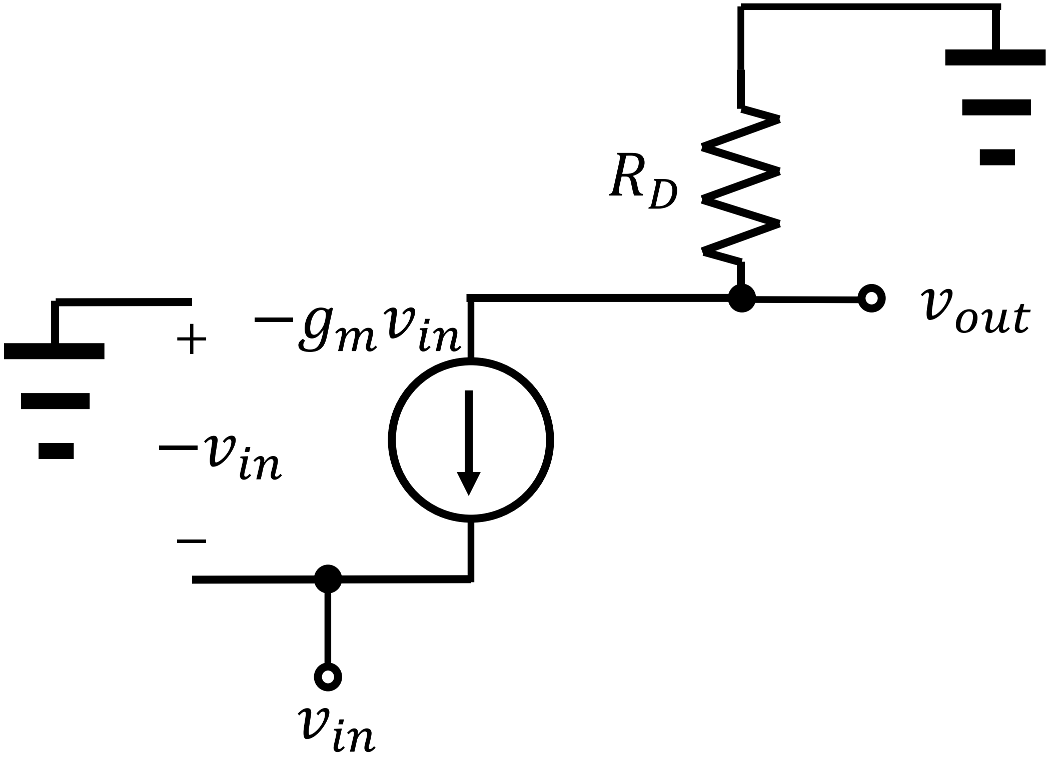
- Neglect the output resistance, r_O.
  - Voltage gain

$$A_v = +g_m R_D$$

  - Input impedance

$$R_{in} = \frac{1}{g_m}$$

It's small!



Gain & input impedance (2/2)

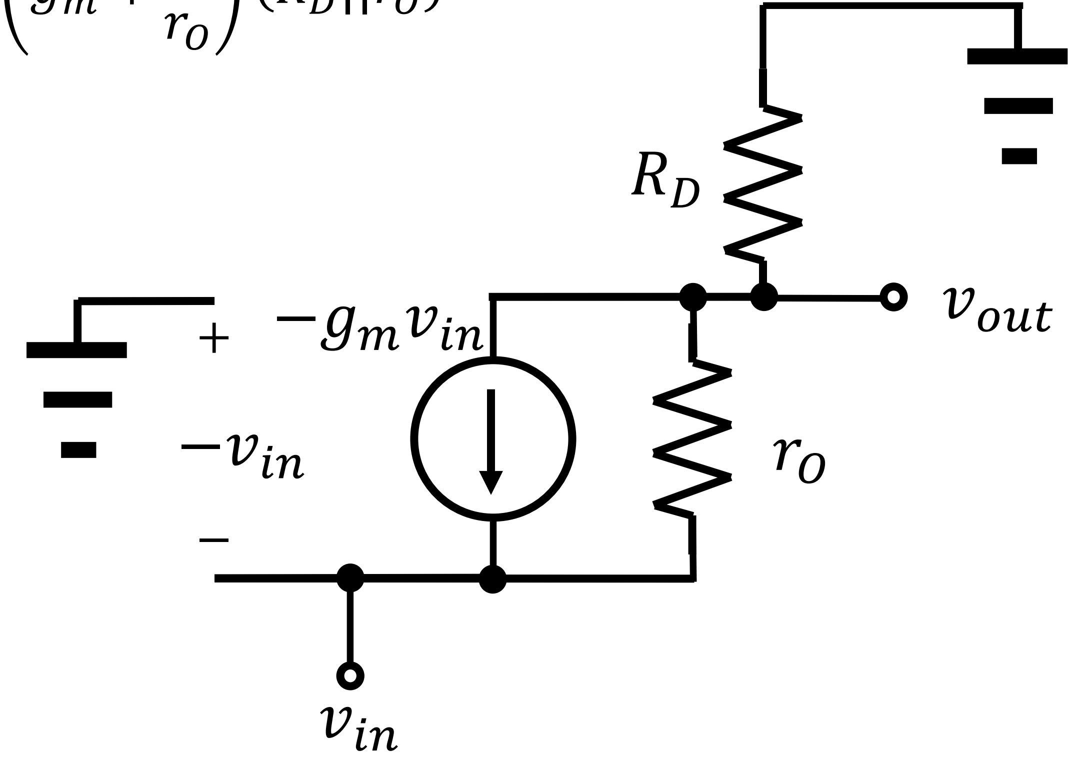
- Consider the output resistance, r_O.
  - Voltage gain

$$A_v = +\left(g_m + \frac{1}{r_O}\right)(R_D || r_O)$$

  - Input impedance

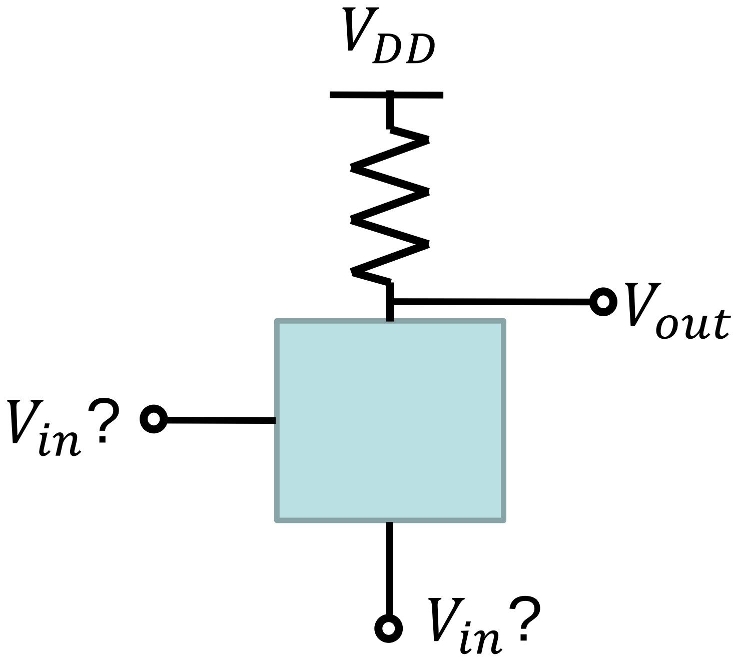
$$R_{in} = \frac{r_O + R_D}{g_m r_O + 1}$$

It's small!

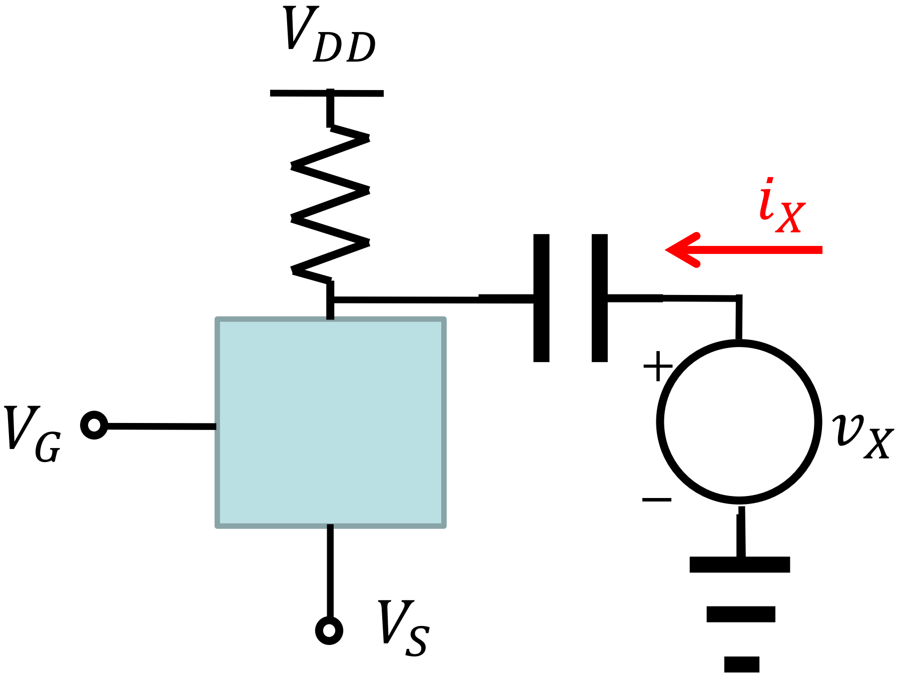


Output impedance

- Without a finite source resistance, $R_{out} = r_O \,||R_D$



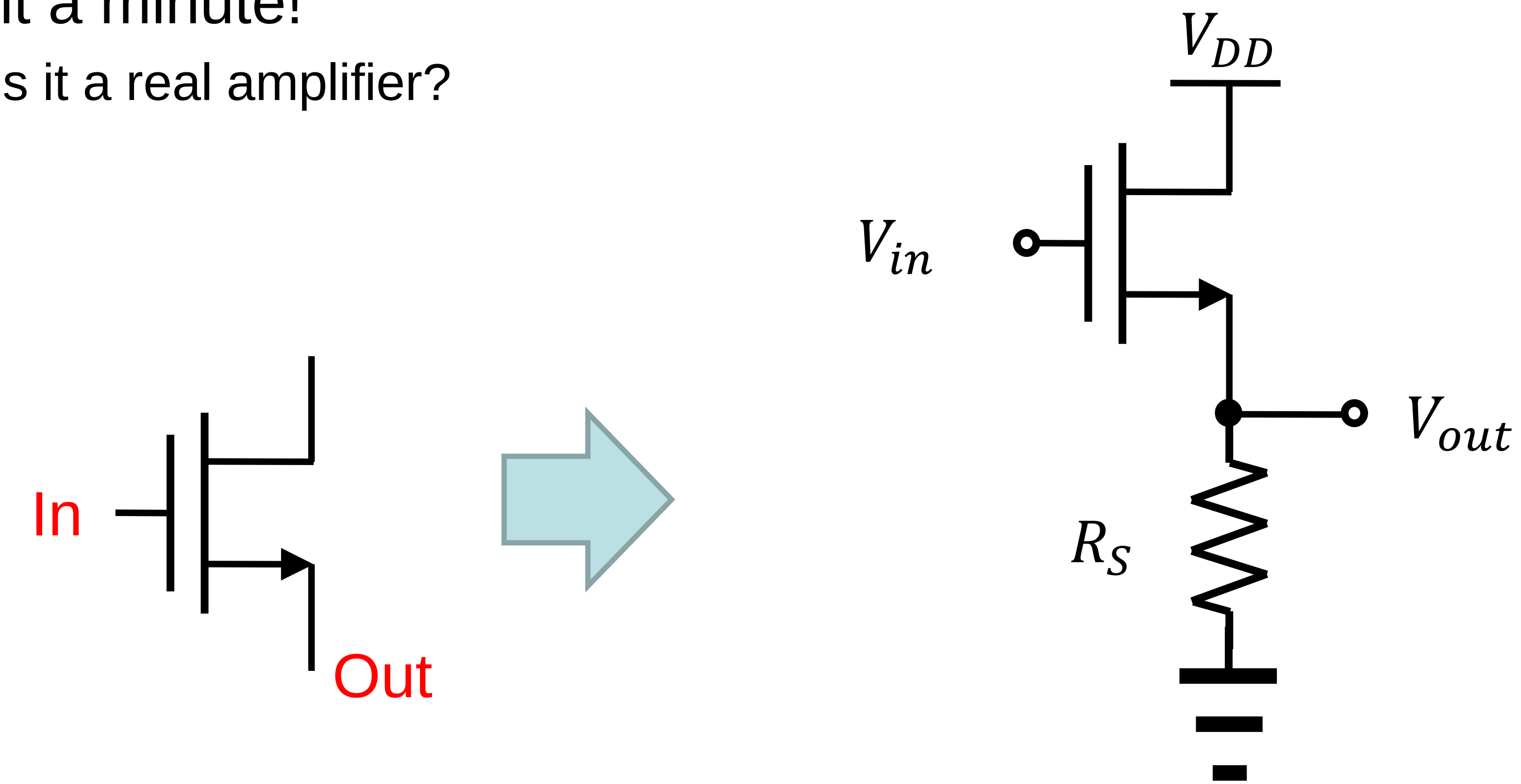
Generic form of CS and CG stages



Setting for calculating R_{out}

Source follower

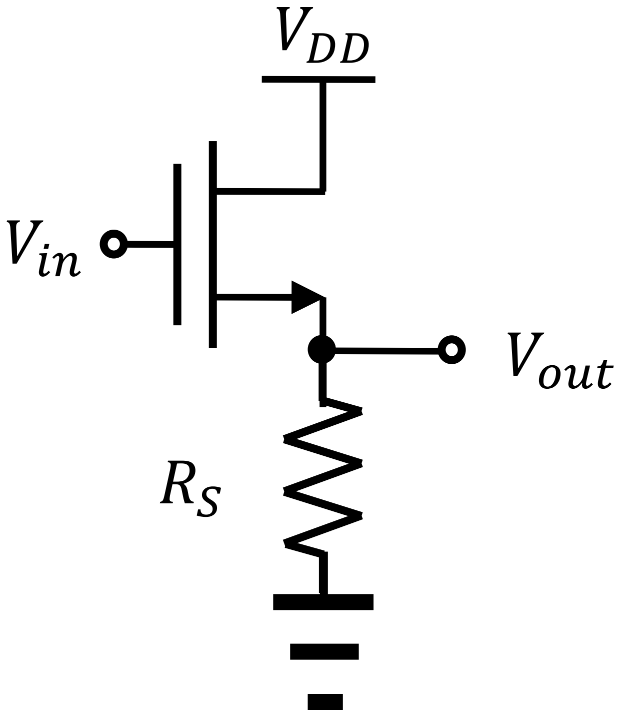
- Also called the “common-drain” stage
  - The drain is ac grounded.
- Wait a minute!
  - Is it a real amplifier?



Its core

- Gain is less than 1?? (Neglecting r_O)

$$A_v = +\frac{g_m R_S}{1 + g_m R_S}$$



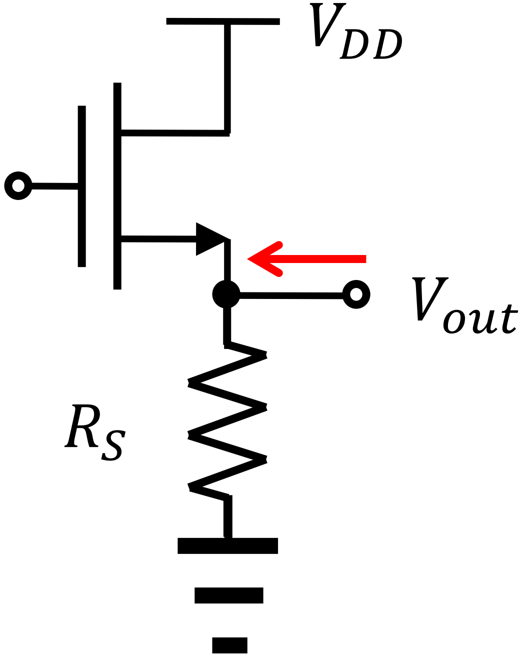
You should be able to draw the small-signal model.

Useless?

- Calculate the input and output impedances.
  - Since the gate is the input terminal, the input impedance is very high at low frequencies.
  - How about the output impedance?

$$R_{out} = \frac{1}{g_m}||r_O||R_S$$

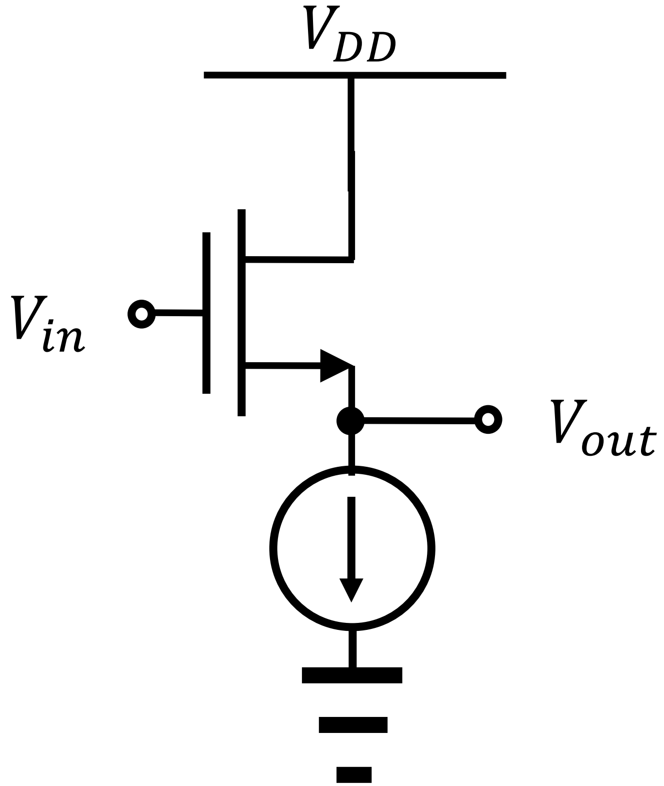
  - It is relatively low.



- High input imp., low output imp.
  - They can serve as good "buffers."

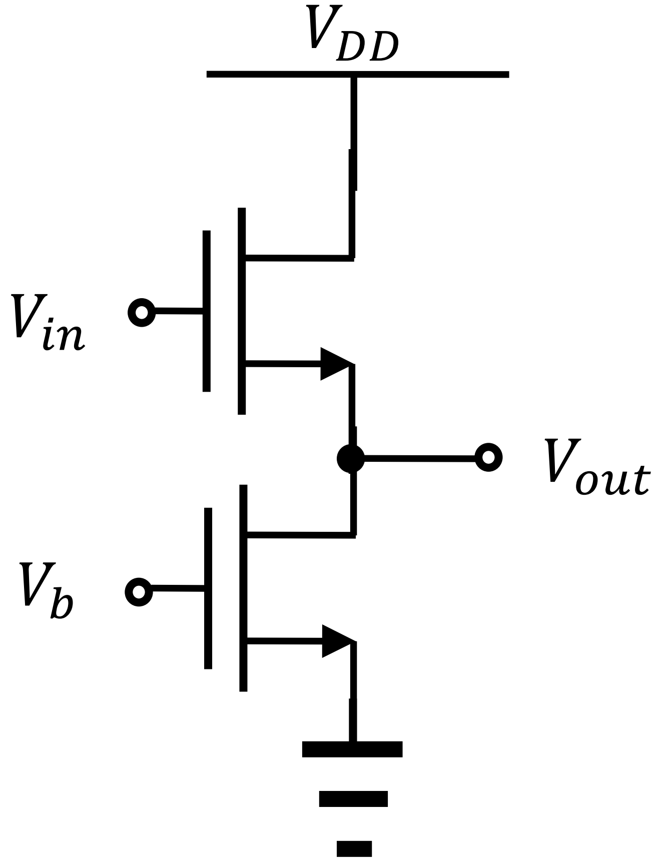
Razavi, example 17.36

- In integrated circuits, the follower is typically realized as shown below. Determine the voltage gain if the current source is ideal. Neglect the channel-length modulation.



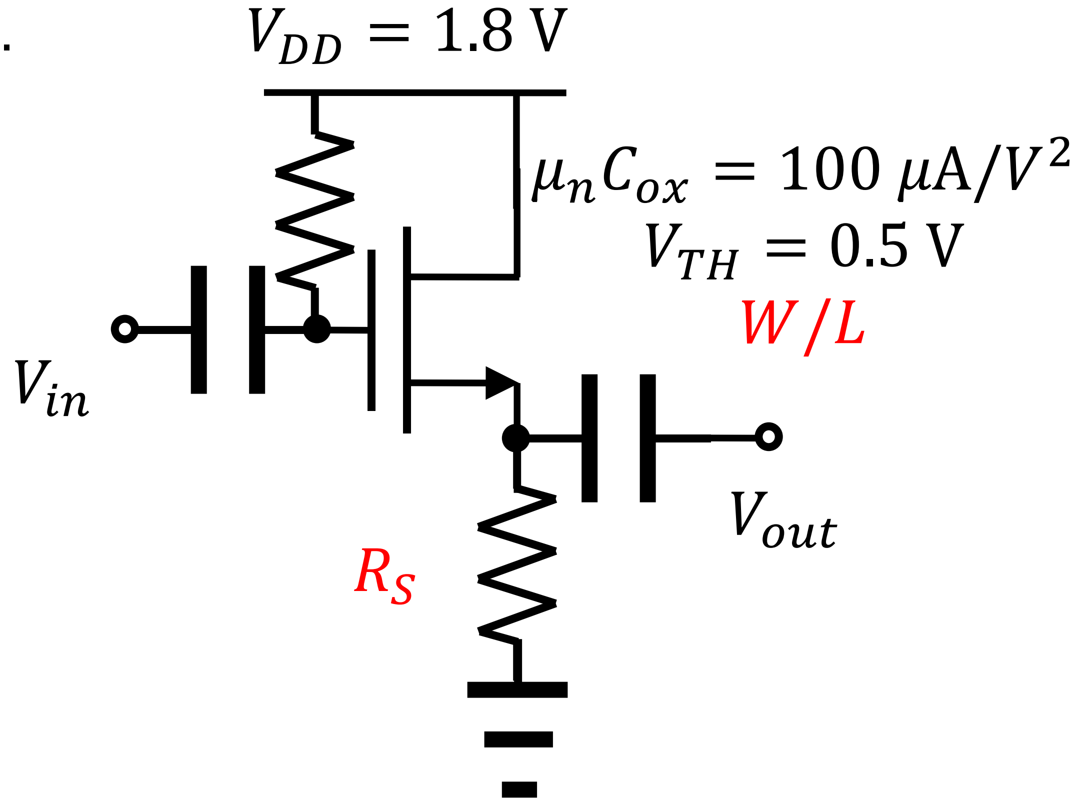
Razavi, example 17.37

- A source follower is realized as shown below. Calculate the voltage gain of the circuit.



Razavi, example 17.39

- Design the source follower.
  - Determine W/L and R_S.
  - The DC drain current is 1 mA.
  - The voltage gain is 0.8.



Why digital?

- You know the answer.
  - And you know what it actually is. (Binary)
- Today, we will consider the following questions:
  - How can we treat the arithmetic operations (Addition, subtraction, multiplication, …)
  - What is the elemental operation?
  - Then, what are the essential circuits to build such a system?
  - (It will be a short review on *Digital Design*.)
- Inverter and NAND gates

Addition

- Once you can add two numbers, x and y, you can do
  - Addition, $x + y$ (of course)
  - Subtraction, $x - y = x + (-y)$
    - A simple example) 4-digit binary numbers, $a = 0110$ and $b = 0011$.
    - The 1's complement of b is 1100.
    - The 2's complement of b is 1101.
    - Sum of 0110 and 1101 is 10011.
    - Discarding the end carry gives us the correct answer, 0011.
  - Multiplication, $x \times y$

Even in addition,

- You can recognize that
  - Addition of two 1-bit binary numbers is the core operation!
  - There are only four possible cases!

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= \boxed{1}0 \longleftarrow \text{Carry} \end{aligned}$$

Inclusion of carry-bit

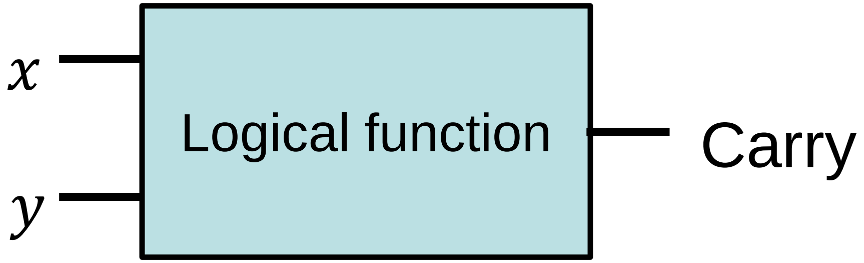
- We introduce a separate bit for representing the carry.
  - Carry-bit & sum-bit

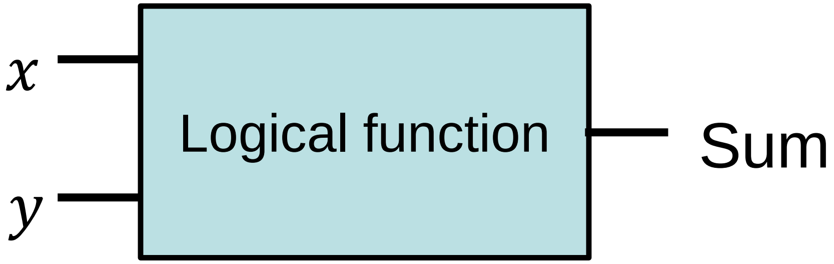
$$
\begin{aligned}
0 + 0 &= 00 \\
0 + 1 &= 01 \\
1 + 0 &= 01 \\
1 + 1 &= 10
\end{aligned}
$$

Carry

Sum

  - Treat them separately!





Relation btw x, y, and sum

- Concentrate on the sum-bit.
  - A table can be made.
  - It is called a **truth table**.

x	y	sum
0	0	0
0	1	1
1	0	1
1	1	0

  - Yes, it is the exclusive OR, $x \text{ XOR } y$.

Relation btw x, y, and carry

- Concentrate on the carry-bit.
  - A table can be made, again.

x	y	carry
0	0	0
0	1	0
1	0	0
1	1	1

  - Yes, it is the AND operation, $x \text{ AND } y$

After all,

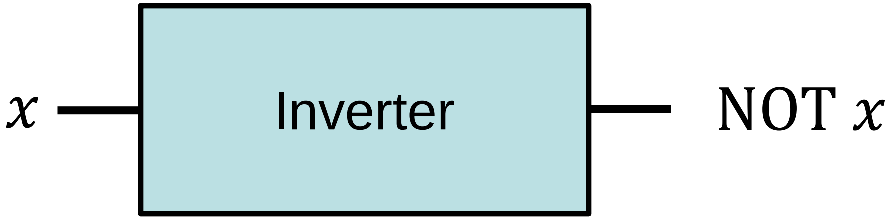
- As much as we have AND, OR, and NOT gates, we can implement any Boolean function.
  - For example,

$$x \text{ XOR } y = \big(x \text{ AND } (\text{NOT } y)\big) \text{ OR } \big((\text{NOT } x) \text{ AND } y\big)$$

  - *With NAND, NOR, and NOT gates, we can, too.*

Inverter and NAND

- NOR can be implemented similarly.

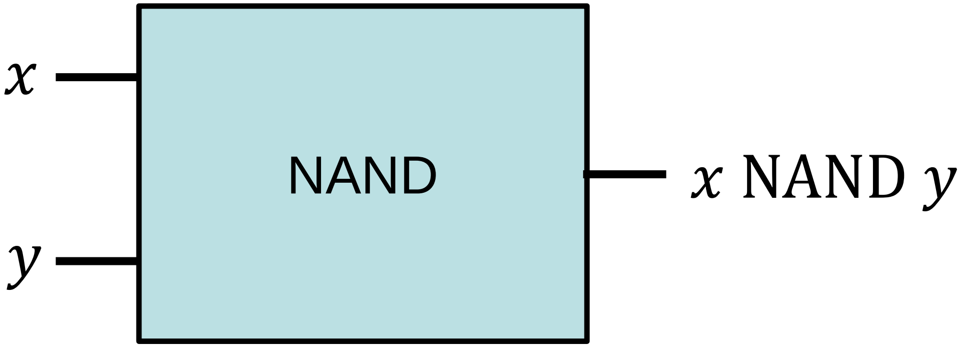


x

Inverter

NOT x

x	NOT
0	1
1	0



x

y

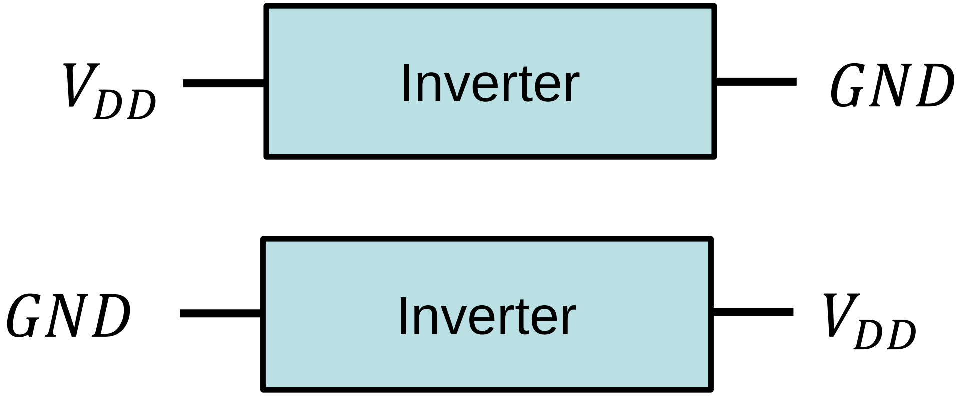
NAND

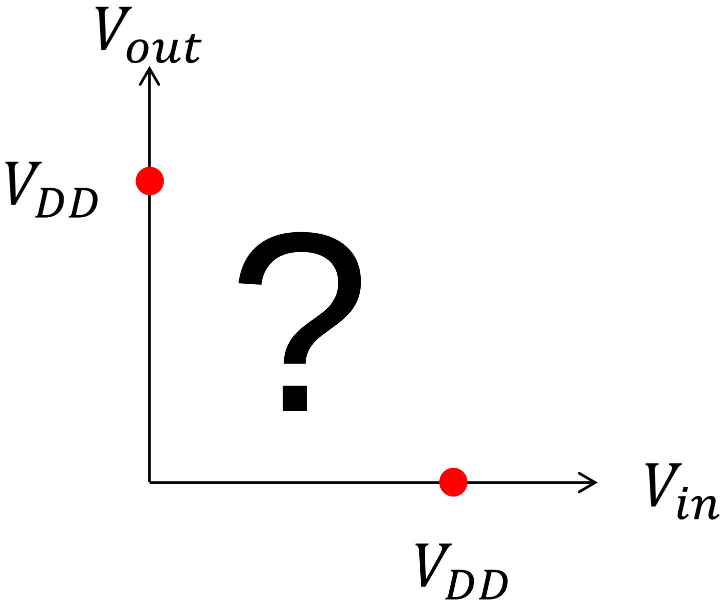
x NAND y

x	y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

In circuit,

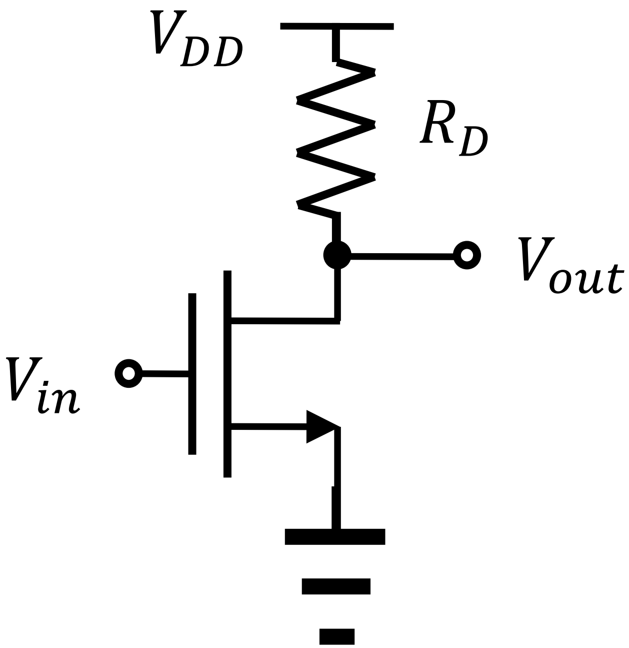
- How can we represent 0 and 1?
  - V_{DD} is assigned to the logical value, 1.
  - GND is assigned to the logical value, 0.





NMOS inverter

- How can we have an output, 0?
  - Only when the input is high. *You have seen it before!*



Vin	Vout
0	1
1	0

Voltage transfer

- When $V_{in} < V_{TH}$, trivially, $V_{out} = V_{DD}$.
- When V_{in} is slightly larger than V_{TH}, the NMOSFET is in the saturation region.

$$V_{out} = V_{DD} - I_D R_D$$

$$V_{out} = V_{DD} - \frac{1}{2}\mu_n C_{ox}\frac{W}{L}R_D(V_{in} - V_{TH})^2$$

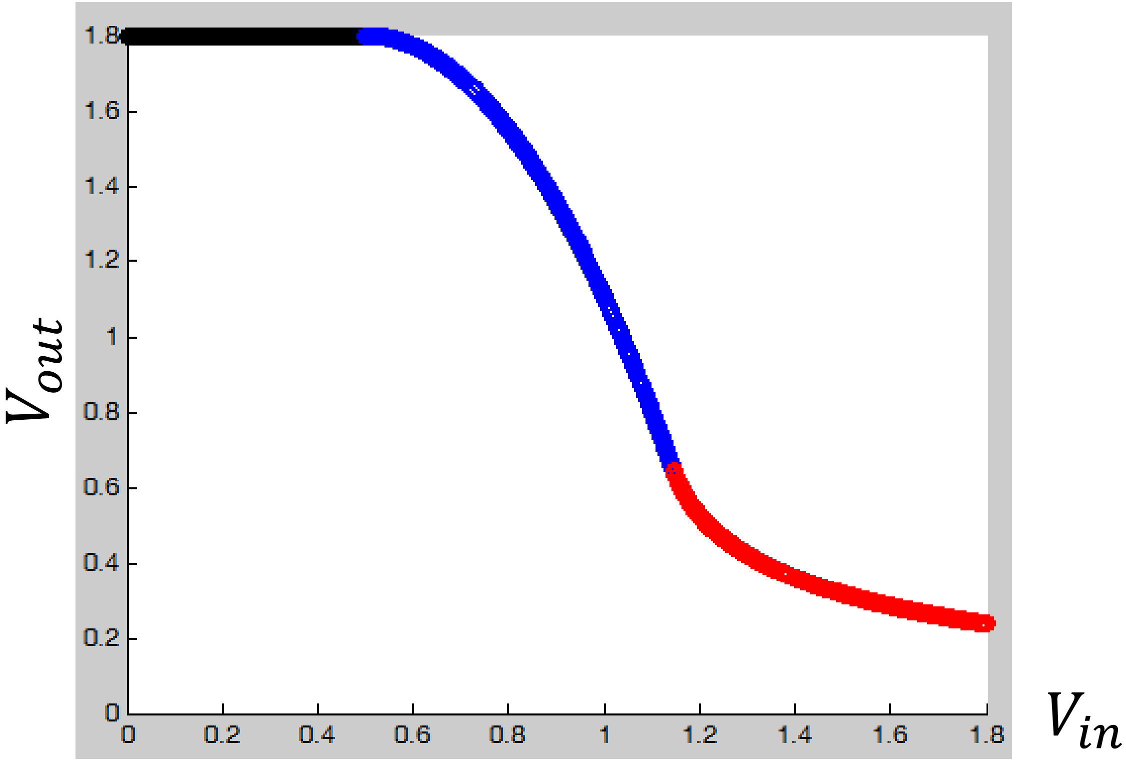
- When V_{in} is further increased, the NMOSFET is in the triode region.

$$V_{out} = V_{DD} - \frac{1}{2}\mu_n C_{ox}\frac{W}{L}R_D[2(V_{in} - V_{TH})V_{out} - V_{out}^2]$$

Draw it! (1/2)

- Parameters in Example 17. 14 (Razavi) w/o modification.

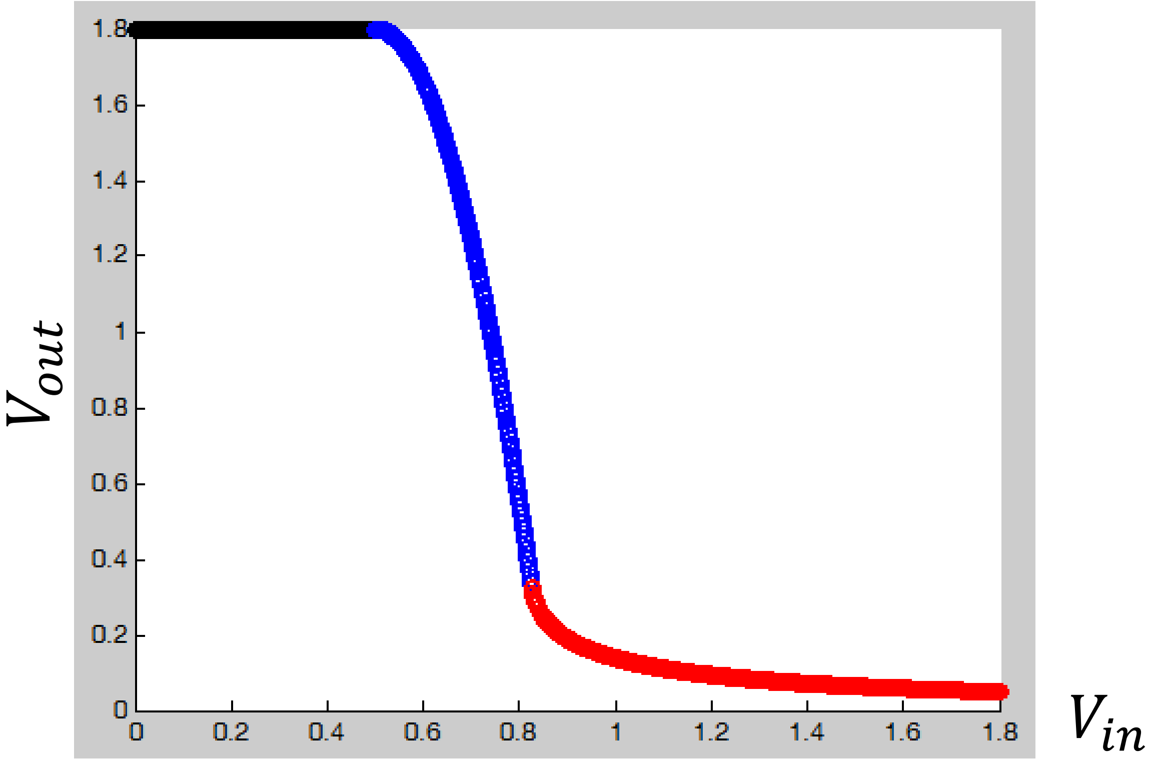
$\mu_n C_{ox} = 100\ \mu\text{A/V}^2$, $V_{TH} = 0.5\ \text{V}$, $\frac{W}{L} = \frac{10}{0.18}$, $R_D = 1k\Omega$ and $V_{DD} = 1.8\ \text{V}$



Draw it! (2/2)

- With a wider NMOSFET

$\mu_n C_{ox} = 100\ \mu\text{A/V}^2$, $V_{TH} = 0.5$ V, $\frac{W}{L} = \frac{50}{0.18}$, $R_D = 1k\Omega$ and $V_{DD} = 1.8$ V



GIST Lecture on June 1, 2020 (Internal use only)