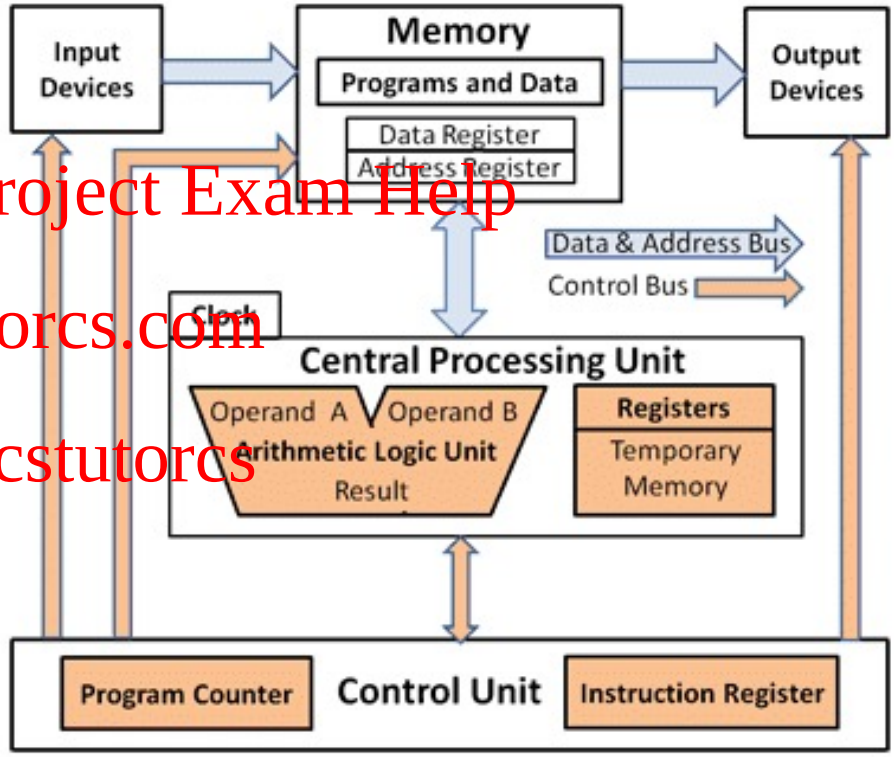
# Transistors, Gate Design and Properties






Brent C. Munsell

Computer Organization

# Agenda

- Announcements (~ 5 mins)
- Poll Everywhere (~15 mins)
- Complete ALU Material (~15 mins)
- Transistors
- Voltage and Logic } (~35 mins)
- Propagation Delay

# Announcements

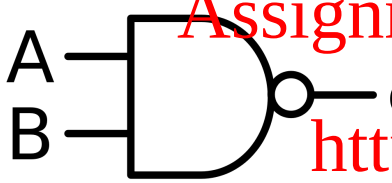
- Lab 1
  - Due 9/15 @ 11:55 pm with no late penalty.
- Quiz 1
  - Great job (mean ~86%, mode 100%)!
  - Release grades and feedback today or tomorrow (talk to Jesse)
- Quiz 2
  - Release this Friday @ 12 pm
  - Due this Sunday @ 11:55 pm
  - Post announcement this Thursday.

# Transistor

**Overview, Design, and Operation**

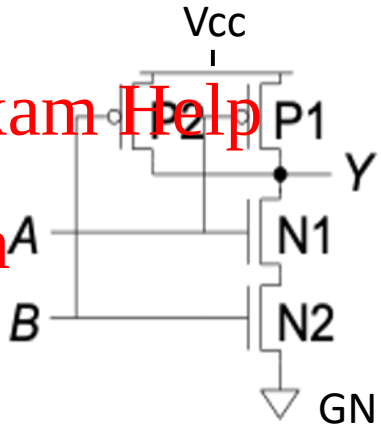
# Overview

## Schematic Diagram




Abstraction for humans

## Transistor




Physical Hardware: Along with power ($V_{CC}$) and ground (GND), four transistors are required to make one two-input NAND gate

# Transistor Size?

Intel Video: Really good ☺


Three billion transistors on a fingertip. Credit: Fritzchens Fritz

# Voltage ($V_{CC}$) and Ground (GND)

Think of a direct current (DC) battery:




(+) voltage DC

- e.g., $V_{CC}$ = 1.5 VDC or 9 VDC (many other values)

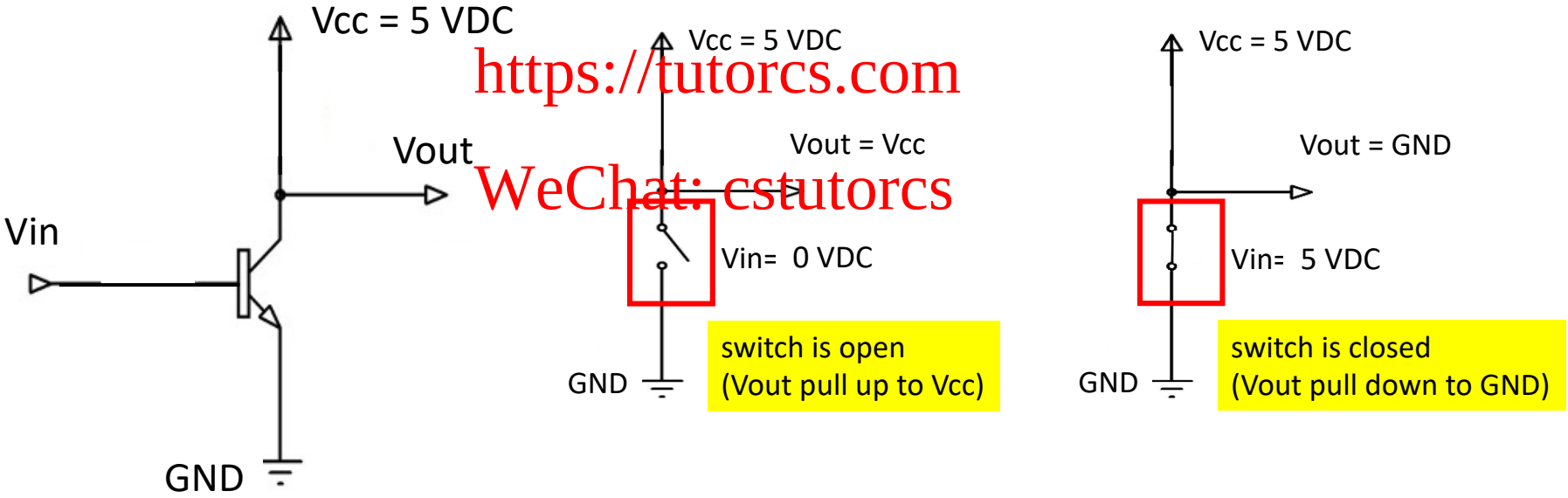
(-) voltage DC

- E.g., GND or 0 VDC

# Basic Design

Conceptually, a DC voltage switch!

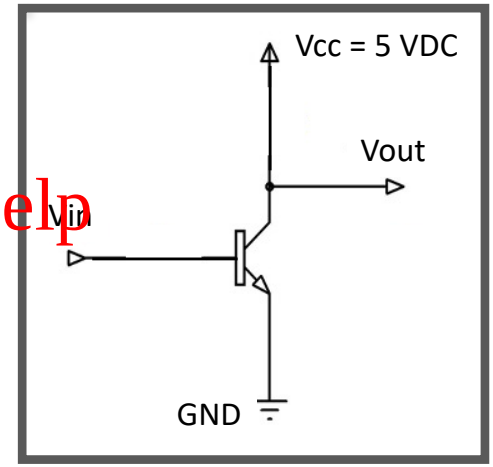


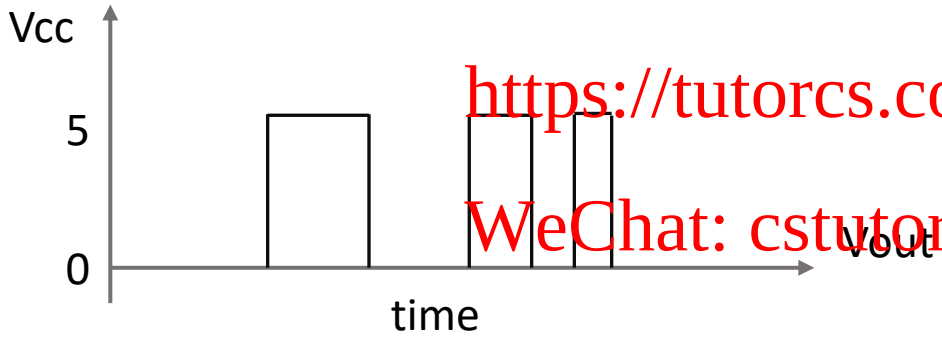


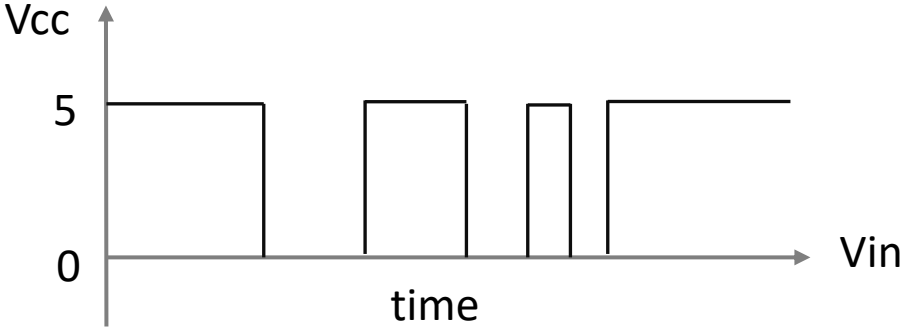
# Basic Operation

Let's work though a simple example






Input (Vin) voltage is "switching" the output (Vout) voltage

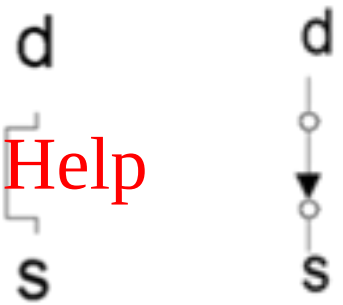
# Metal Oxide Semiconductor (MOS) Transistor



**Complementary Pull-up/down Design**

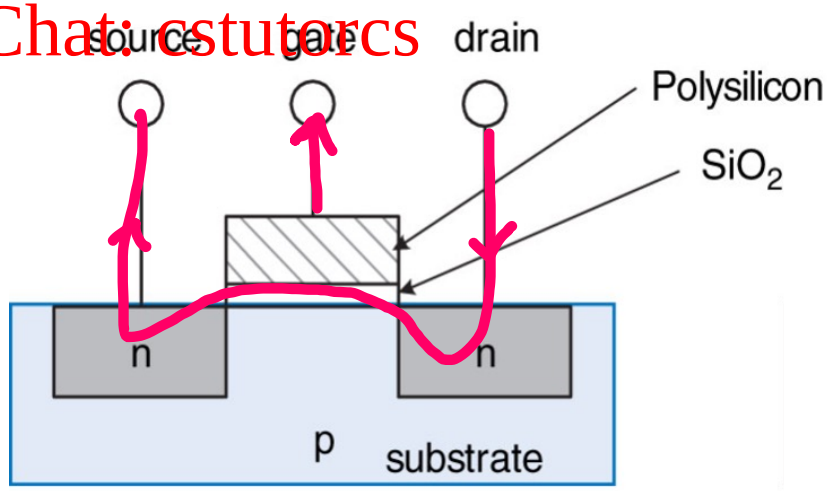
# n-channel Metal Oxide Semiconductor (nMOS) Transistor

Switch is closed when gate (g) is positive VDC value (e.g., 5 VDC)



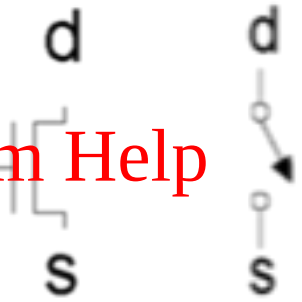



Channel is open, electrons can flow from drain to source



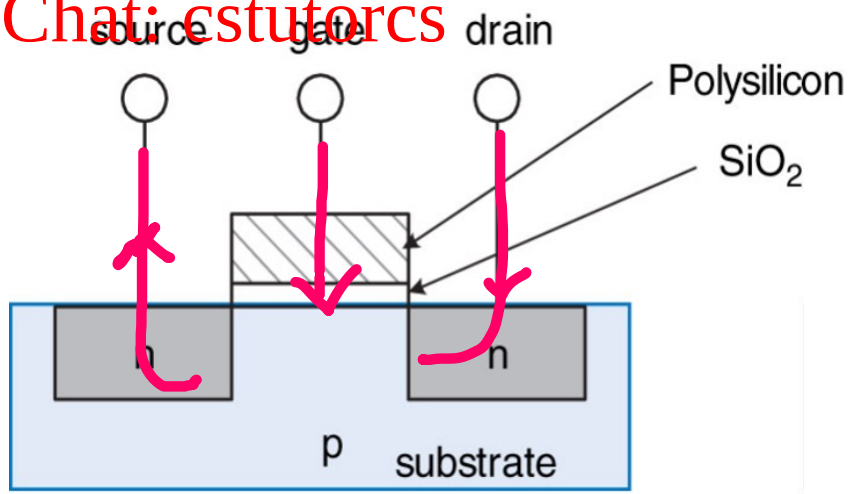
# n-channel Metal Oxide Semiconductor (nMOS) Transistor

Switch is open when gate (g) is GND (e.g., 0 VDC)



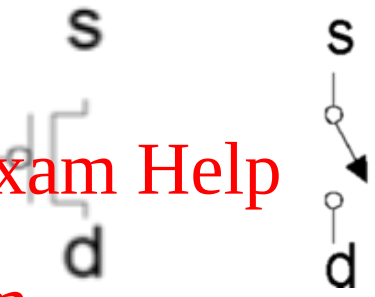



Channel is closed, electrons cannot flow from drain to source

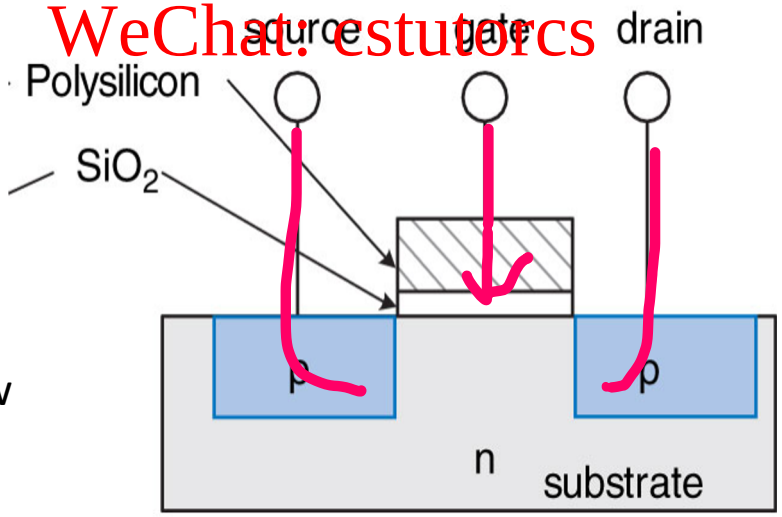


# p-channel Metal Oxide Semiconductor (pMOS) Transistor

Switch is open when gate is positive VDC value (e.g., 5 VDC)



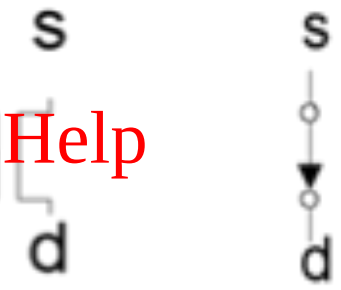



Channel is closed, electrons cannot flow from source to drain

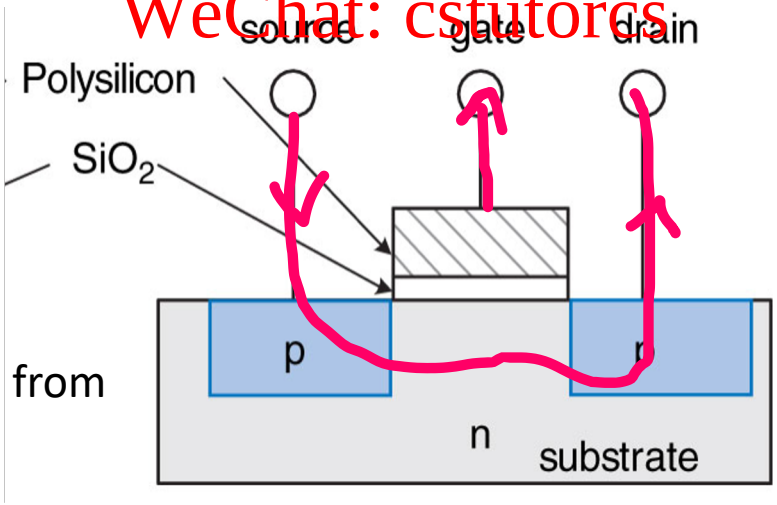
# p-channel Metal Oxide Semiconductor (pMOS) Transistor

Switch is closed when gate is GND (e.g., 0 VDC).



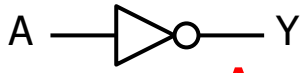



Channel is open, electrons can flow from source to drain.



# NOT Gate MOS Gate Design

Schematic Diagram



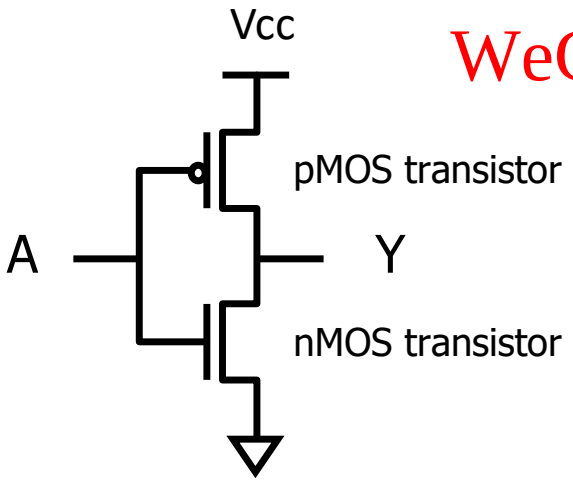

Truth Table

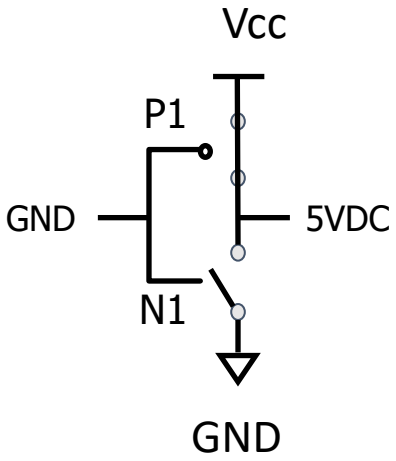
Vcc = 5 VDC

| A | P1 | N1 | Y |
|---|---|---|---|
| GND | 5 VDC | GND | 5 VDC |
| 5 VDC | GND | 5 VDC | GND |

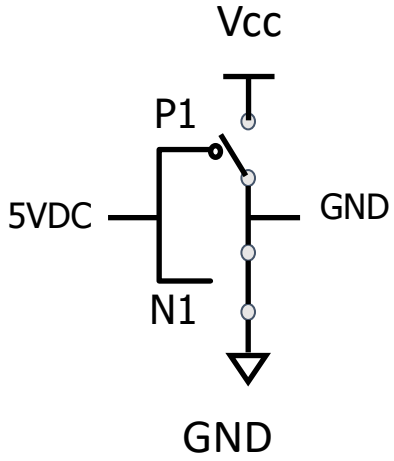


Pull UP

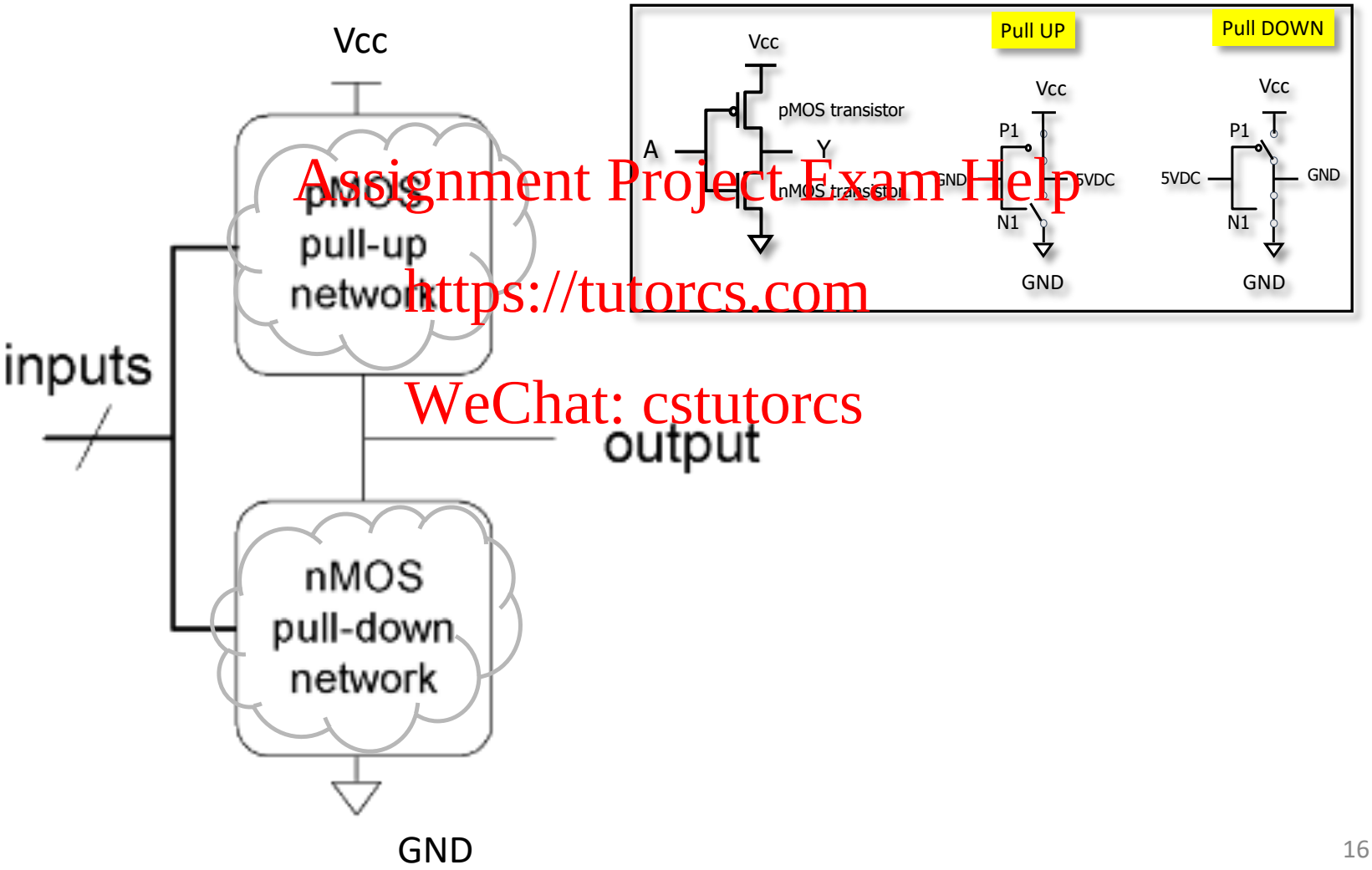


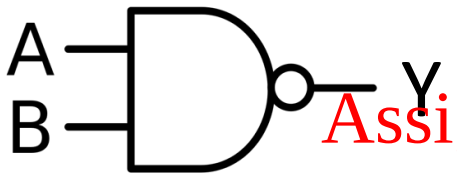

Pull DOWN



# Complementary MOS Design



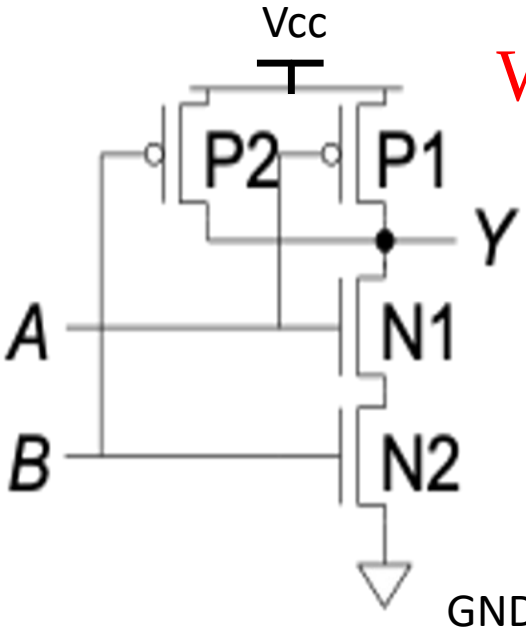
# NAND Gate: MOS Design



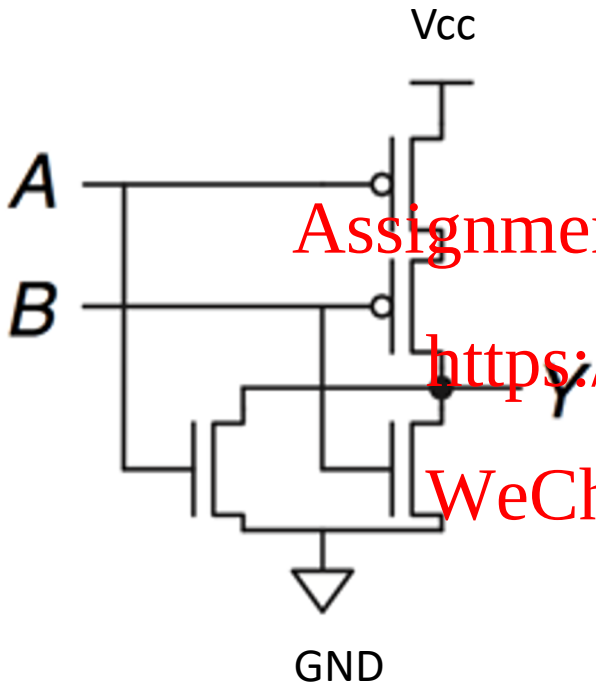

| A | B | P1 | P2 | N1 | N2 | Y |
|---|---|---|---|---|---|---|
| GND | GND | 5VDC | 5VDC | GND | GND | |
| GND | 5VDC | 5VDC | GND | GND | 5VDC | |
| 5VDC | GND | GND | 5VDC | 5VDC | GND | |
| 5VDC | 5VDC | GND | GND | 5VDC | 5VDC | |



Vcc = 5 VDC



# NOR Gate: MOS Design




| A | B | Y |
|---|---|---|
| GND | GND | |
| GND | 5VDC | |
| 5VDC | GND | |
| 5VDC | 5VDC | |

Vcc = 5 VDC

# Relationship between Voltage and Logic

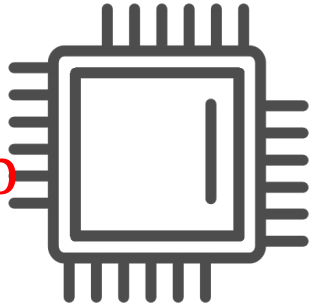


**Digital abstraction**

# Voltage and Logic

Voltage a continuous value

- Has a defined range of values, e.g., 0 to 5 VDC
- And any VDC value between, e.g., 0.1, 0.11, 1.25, 4.52, etc.
- Hardware understands voltage values





Boolean Logic (or Logic for short) is a discrete value that can only be 0 or 1.

- Abstraction that humans understand
- Apply the rules of Boolean algebra
- Simplifies circuit design

# Continuous to Discrete Conversion

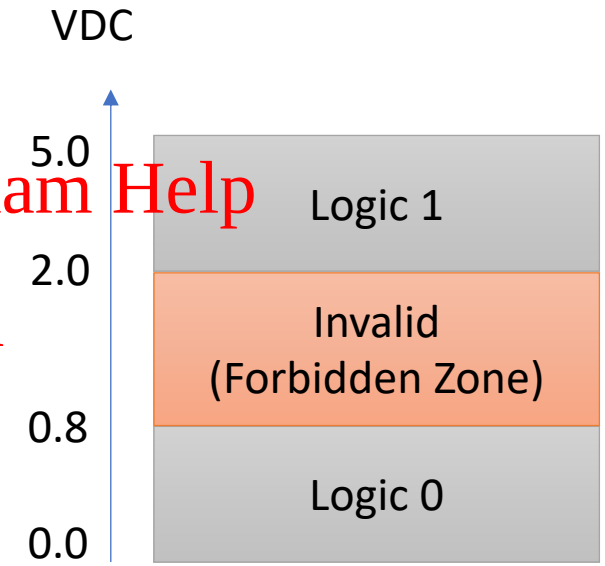
Logic 1

- Voltage range 5 to 2 VDC

Logic 0

- Voltage range 0 to 0.8 VDC






Invalid

- Less than 2 VDC and greater than 0.8 VDC
- Unstable and not reliable.

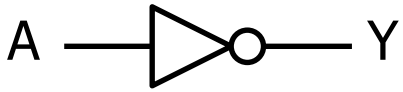
Why is a range of values acceptable? Give an example, not related to voltage?

# Component Properties

**Propagation delay**

# NOT Gate: Closer Inspection

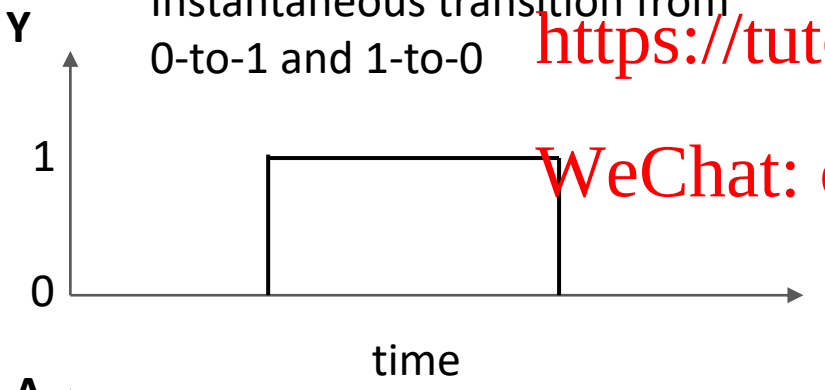
Output (Y) transitions from logic 0-to-1 (or 1-to-0)





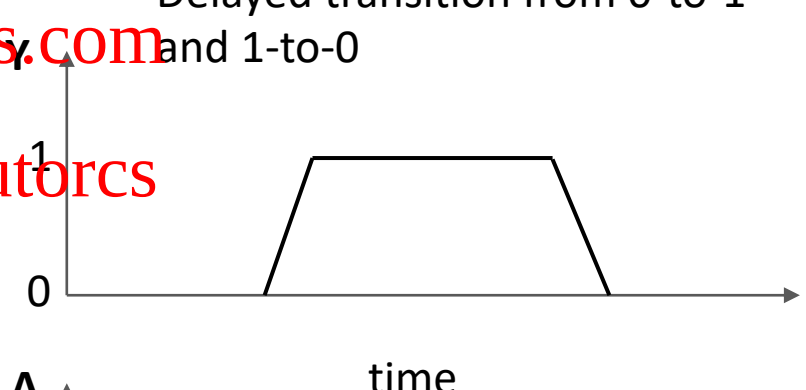
Ideal Property

Instantaneous transition from 0-to-1 and 1-to-0



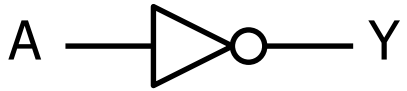

Real Property

Delayed transition from 0-to-1 and 1-to-0

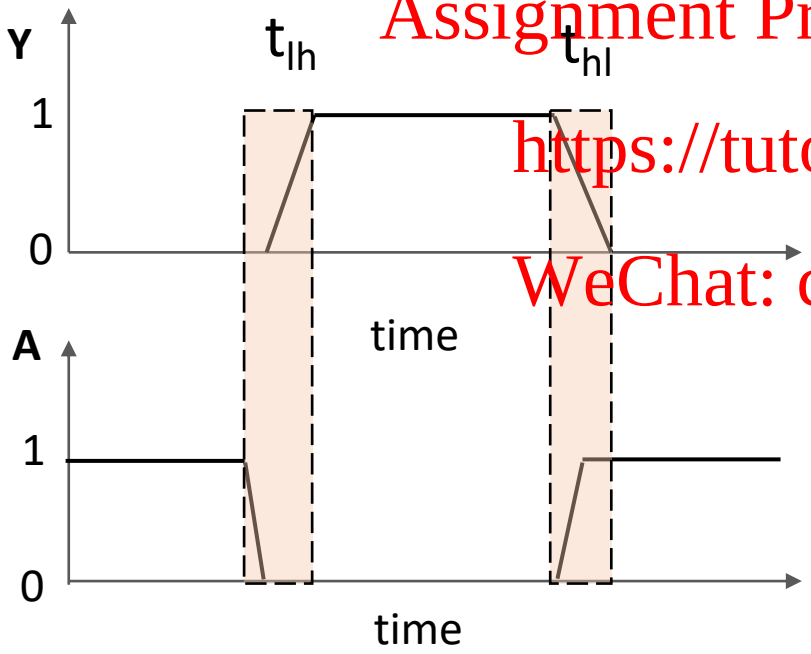


# Gate Delay

Output (Y) transitions from logic 0-to-1 (or 1-to-0)




Real Property




Forbidden zone

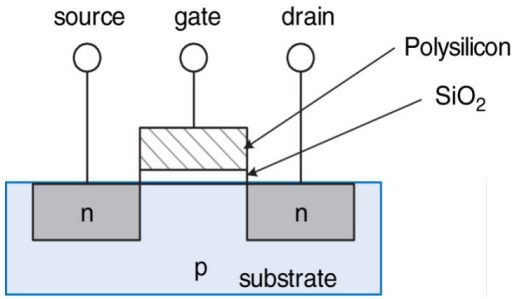
$t_{lh}$ = 0-to-1 (low to high) time delay

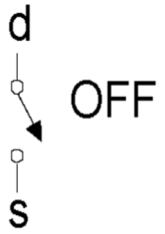
$t_{hl}$ = 1-to-0 (high to low) time delay

The amount of time (in seconds) needed for the output value to change (**propagation delay, $t_d$**)

In this course, we'll assume:

$t_d = t_{lh} = t_{hl}$



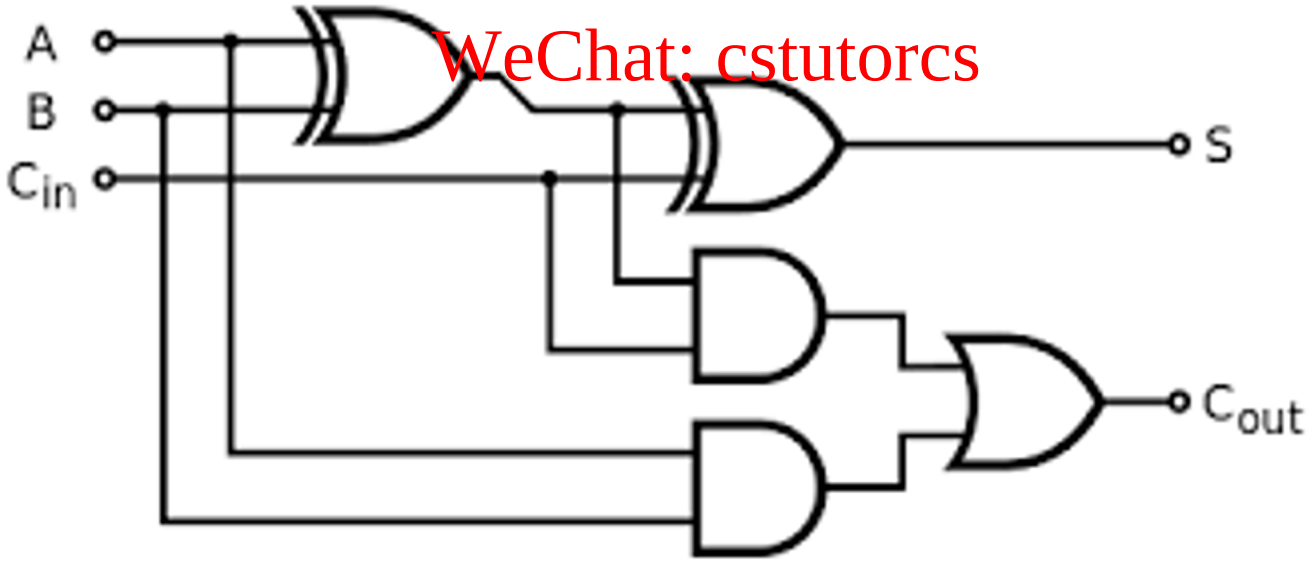



# FA Component Analysis

Assume, propagation delay ($t_d$) for each logic gate is 1 nanosecond (ns).

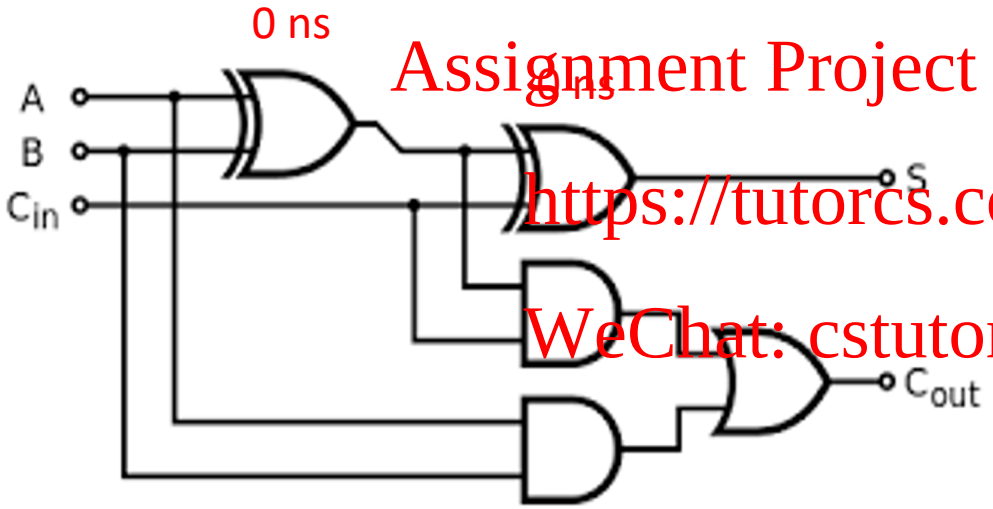
- When the gate output changes 0-to-1 (or 1-to-0) one ns is required for the output to become stable.





# Sum bit analysis

Assume $C_i$, A, B, S are all initially 0

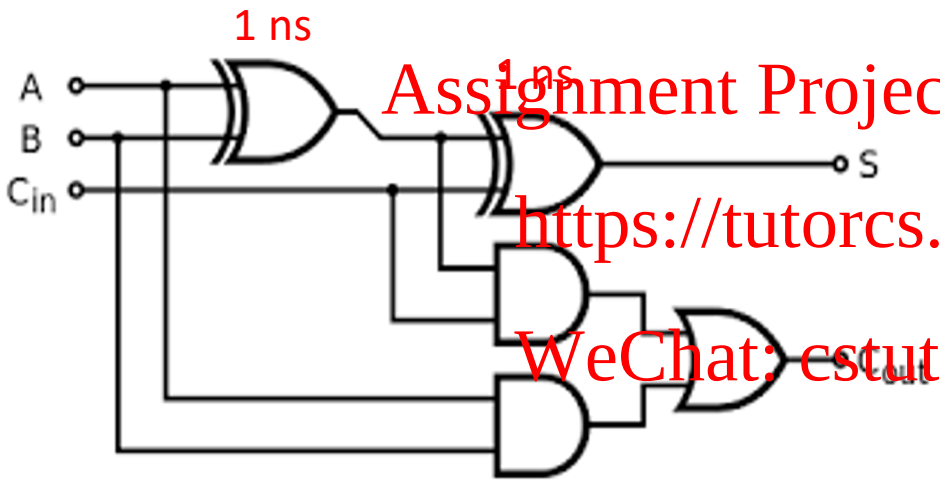





| New Input Value | | | S | |
|---|---|---|---|---|
| $C_i$ | A | B | value | $t_d$ (ns) |
| 0 | 0 | 0 | 0 | 0 |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# Sum bit analysis (continued)

Assume $C_i$, A, B, S are all initially 0

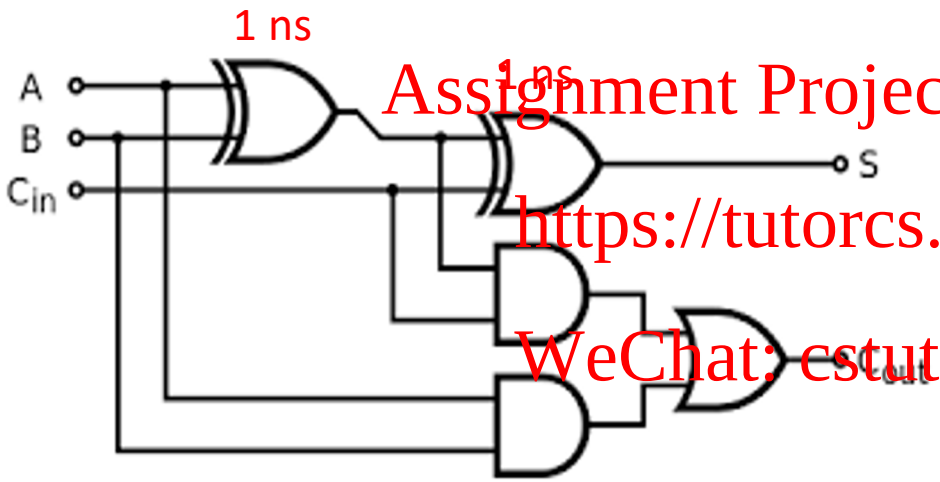





| New Input Value | | | S | |
|---|---|---|---|---|
| $C_i$ | A | B | value | $t_d$ (ns) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 2 |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# Sum bit analysis (continued)

Assume $C_i$, A, B, S are all initially 0

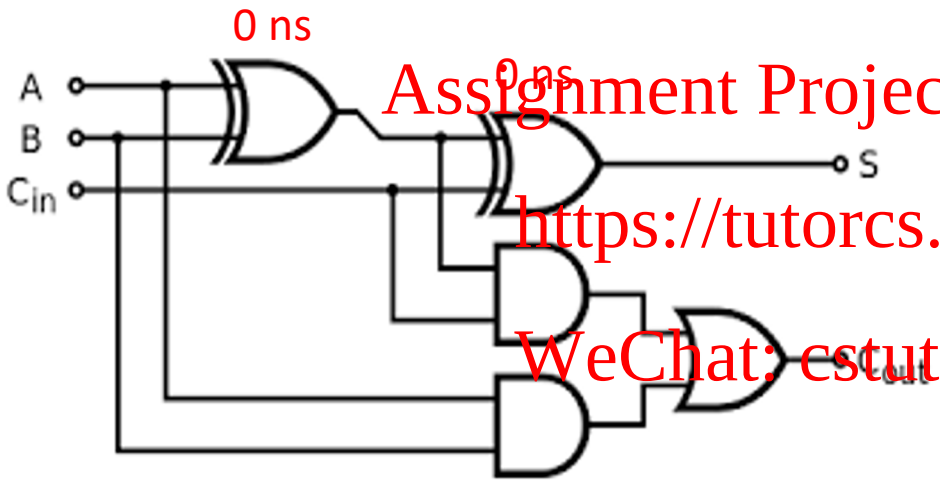





| New Input Value | | | S | |
|---|---|---|---|---|
| $C_i$ | A | B | value | $t_d$ (ns) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 2 |
| 0 | 1 | 0 | 1 | 2 |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# Sum bit analysis (continued)

Assume $C_i$, A, B, S are all initially 0



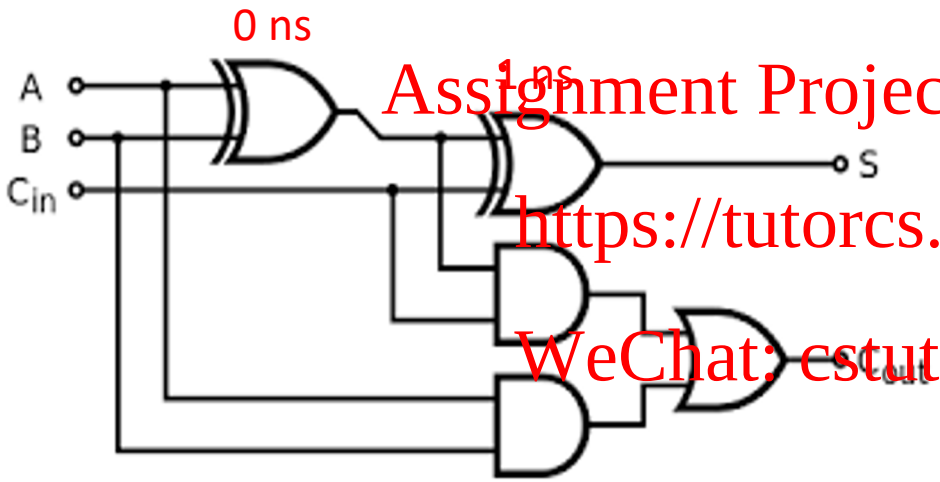



| New Input Value | | | S | |
|---|---|---|---|---|
| $C_i$ | A | B | value | $t_d$ (ns) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 2 |
| 0 | 1 | 0 | 1 | 2 |
| **0** | **1** | **1** | **0** | **0** |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

MOS pull-up and pull-down transistor networks are not changing, so the output does not change (even though the inputs have changed)

# Sum bit analysis (continued)

Assume $C_i$, A, B, S are all initially 0

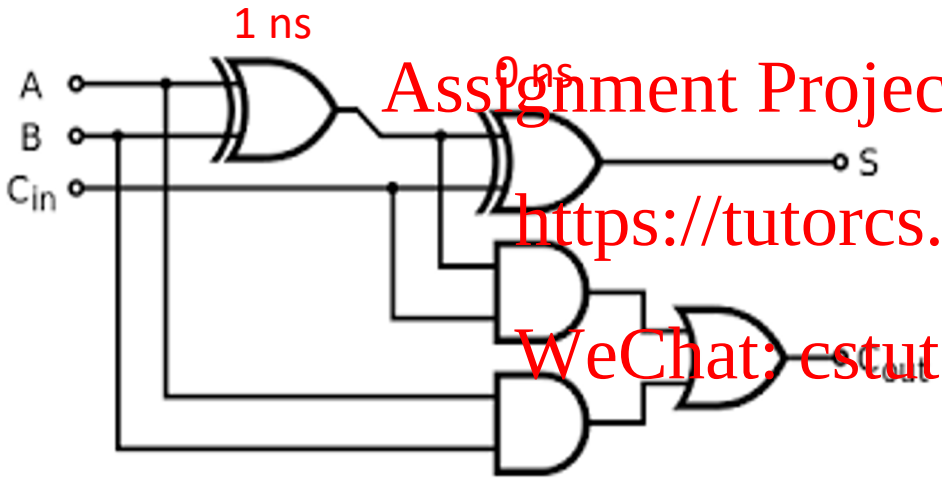





| New Input Value | | | S | |
|---|---|---|---|---|
| $C_i$ | A | B | value | $t_d$ (ns) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 2 |
| 0 | 1 | 0 | 1 | 2 |
| 0 | 1 | 1 | 0 | 0 |
| **1** | **0** | **0** | **1** | **1** |
| | | | | |
| | | | | |
| | | | | |

# Sum bit analysis (continued)

Assume $C_i$, A, B, S are all initially 0



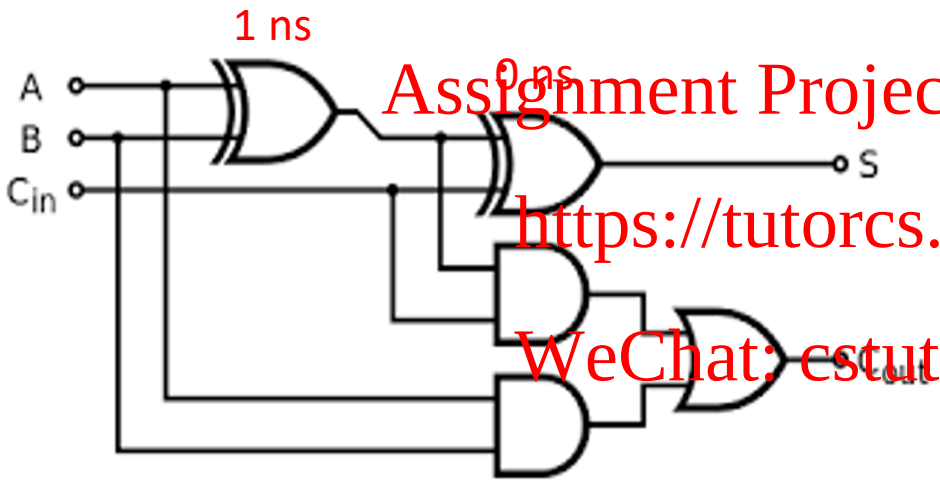



| New Input Value | | | S | |
|---|---|---|---|---|
| $C_i$ | A | B | value | $t_d$ (ns) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 2 |
| 0 | 1 | 0 | 1 | 2 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| **1** | **0** | **1** | **0** | **1** |
| | | | | |
| | | | | |

MOS pull-up and pull-down transistor networks are not changing, so the output is not changing (even though the inputs have changed)

# Sum bit analysis (continued)

Assume $C_i$, A, B, S are all initially 0

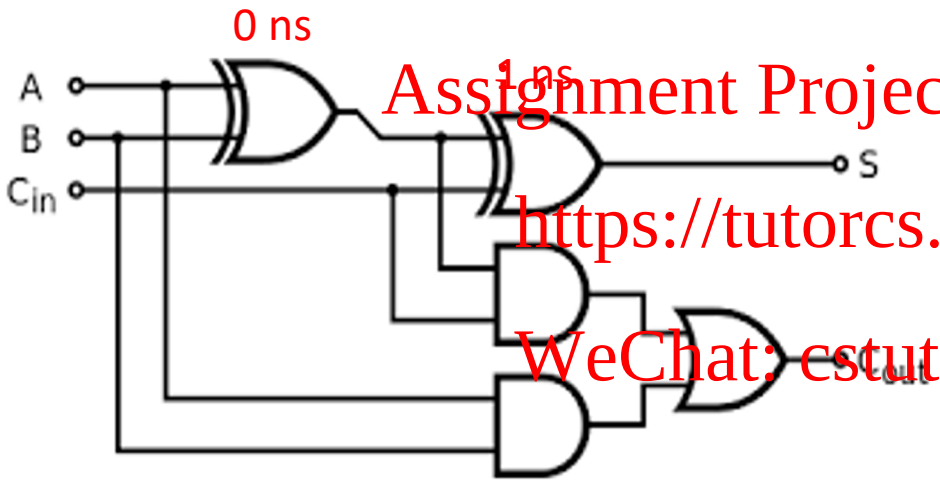





| New Input Value | | | S | |
|---|---|---|---|---|
| $C_i$ | A | B | value | $t_d$ (ns) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 2 |
| 0 | 1 | 0 | 1 | 2 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| **1** | **1** | **0** | **0** | **1** |
| | | | | |

# Sum bit analysis (continued)

Assume $C_i$, A, B, S are all initially 0






| New Input Value | | | S | |
|---|---|---|---|---|
| $C_i$ | A | B | value | $t_d$ (ns) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 2 |
| 0 | 1 | 0 | 1 | 2 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

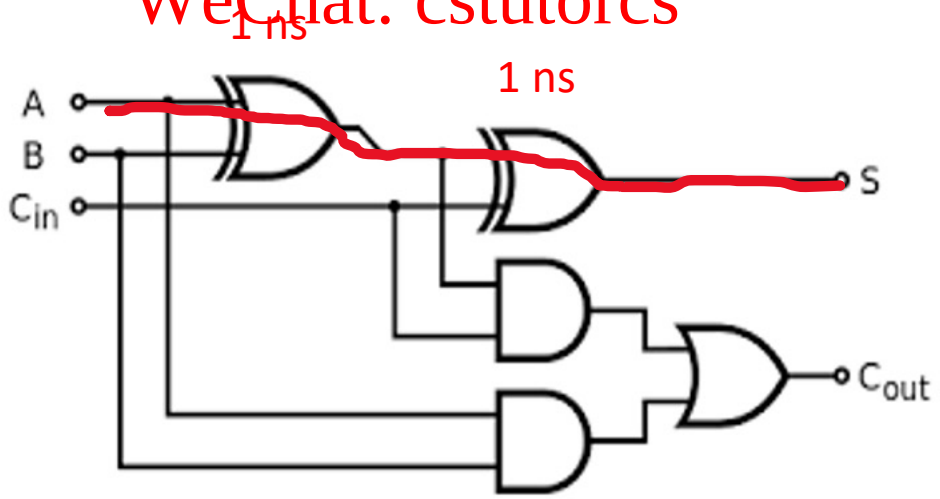
# Sum bit: Worst case analysis

Sum bit is guaranteed to be stable in 2 ns.

In general, follow the longest path from output (S) to inputs and add the gate delays

- Easier than analyzing every possible input combination!

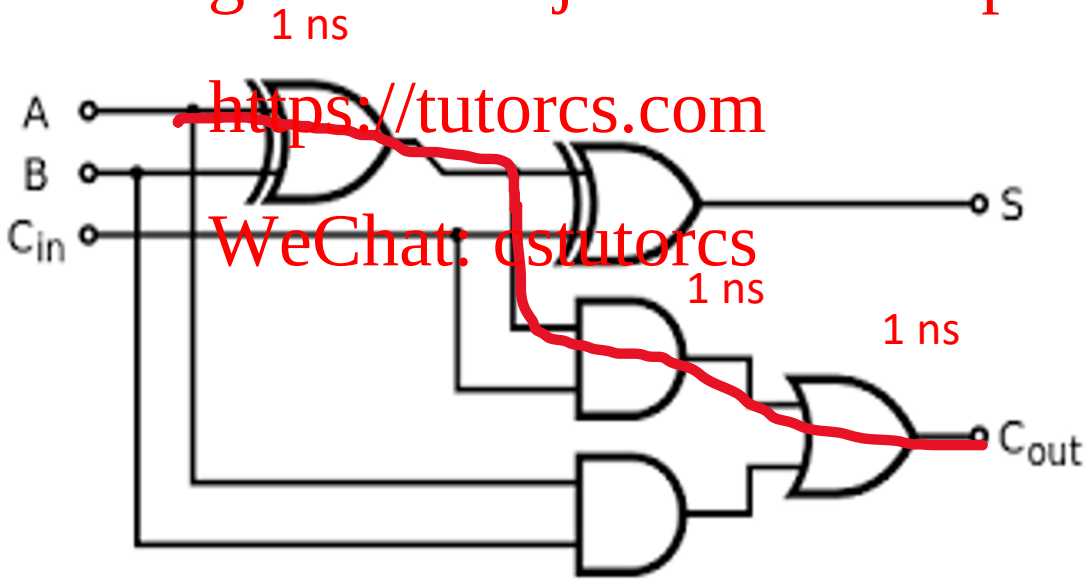




# Carry-out bit: Worst Case Analysis

Carry-out bit is guaranteed to be stable in 3 ns.





# FA Component: Worst Case Analysis

max( Sum $t_d$, Carry-out $t_d$)

- All component outputs will be stable in 3 ns