# The Digital Abstraction

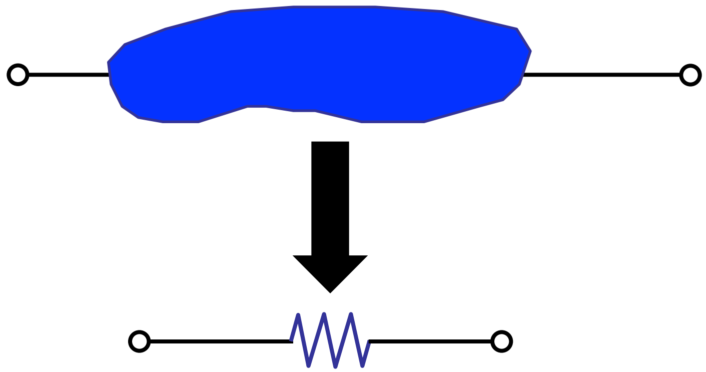
Lecture 4
September 18th, 2018

Jae W. Lee (jaewlee@snu.ac.kr)
Computer Science and Engineering
Seoul National University

***Slide credits**: Prof. Anant Agarwal at MIT*

# Review

- **Discretize matter by agreeing to observe the lumped matter discipline (LMD)**



**Lumped Circuit Abstraction**

- **Analysis tool kit: KVL/KCL, node method, superposition, Thévenin, Norton**
  - Remember superposition, Thévenin, Norton apply only for linear circuits

# Today

- Discretize value ⟶ Digital abstraction

- Interestingly, we will see shortly that the tools learned in the previous three lectures are sufficient to analyze simple digital circuits

# Outline

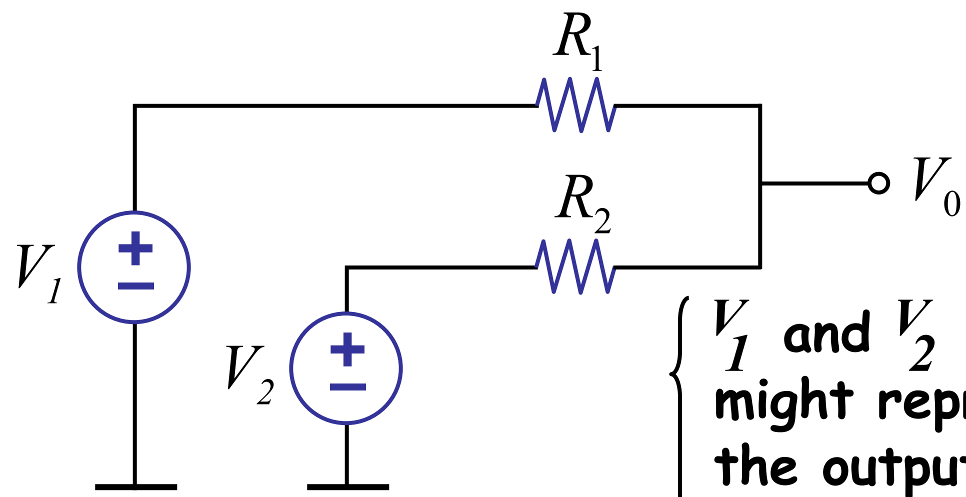
**Textbook: Ch. 5.1-5.3**

- **Voltage Levels and Static Discipline**
- Boolean Logic
- Combinational Gates

# But first, why digital?

- In the past …

Analog signal processing




$V_1$ and $V_2$ might represent the outputs of two sensors, for example.

By superposition,

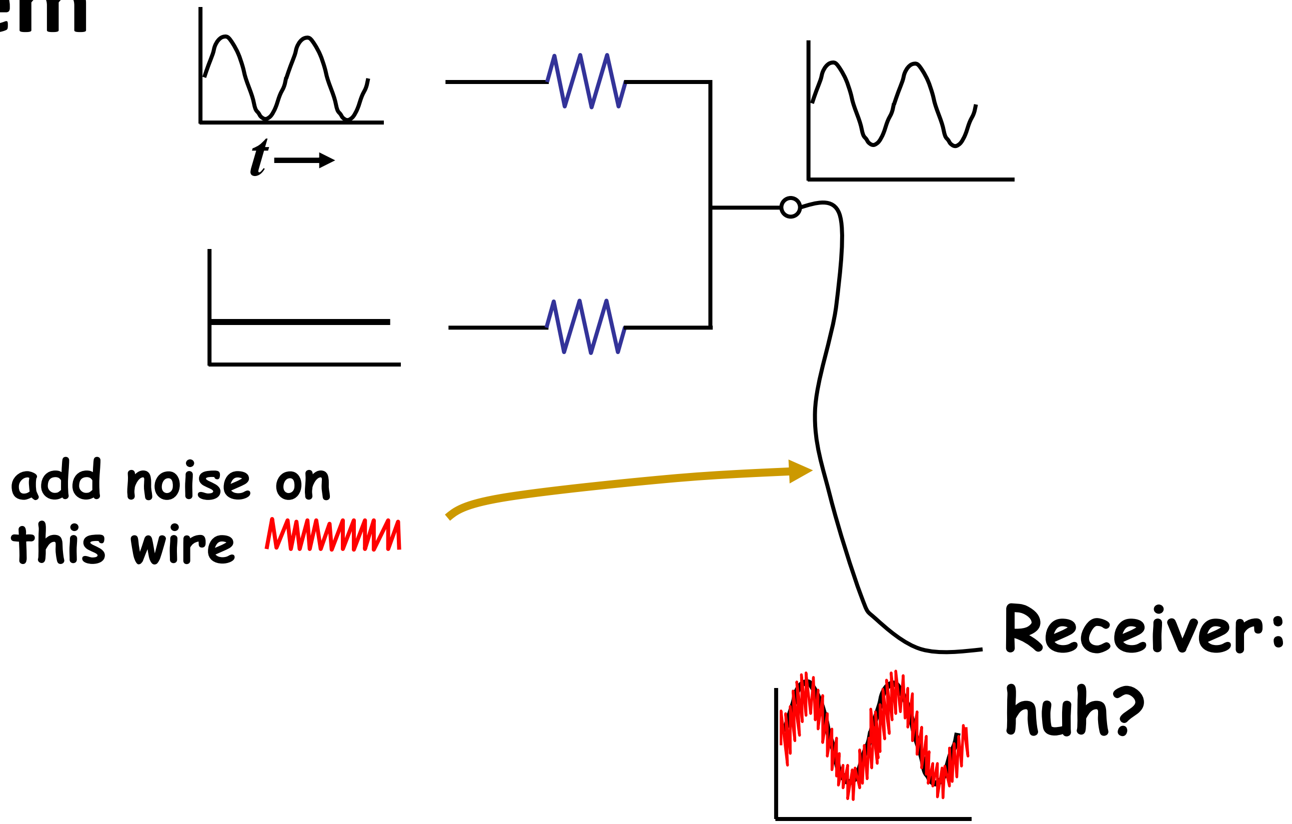
$$V_0 = \frac{R_2}{R_1 + R_2} V_1 + \frac{R_1}{R_1 + R_2} V_2$$

If $R_1 = R_2$,

$$V_0 = \frac{V_1 + V_2}{2}$$

The above is an "adder" circuit.

# Noise Problem




... noise hampers our ability to distinguish between small differences in value — e.g. between 3.1V and 3.2V.

# Value Discretization

Restrict values to be one of two

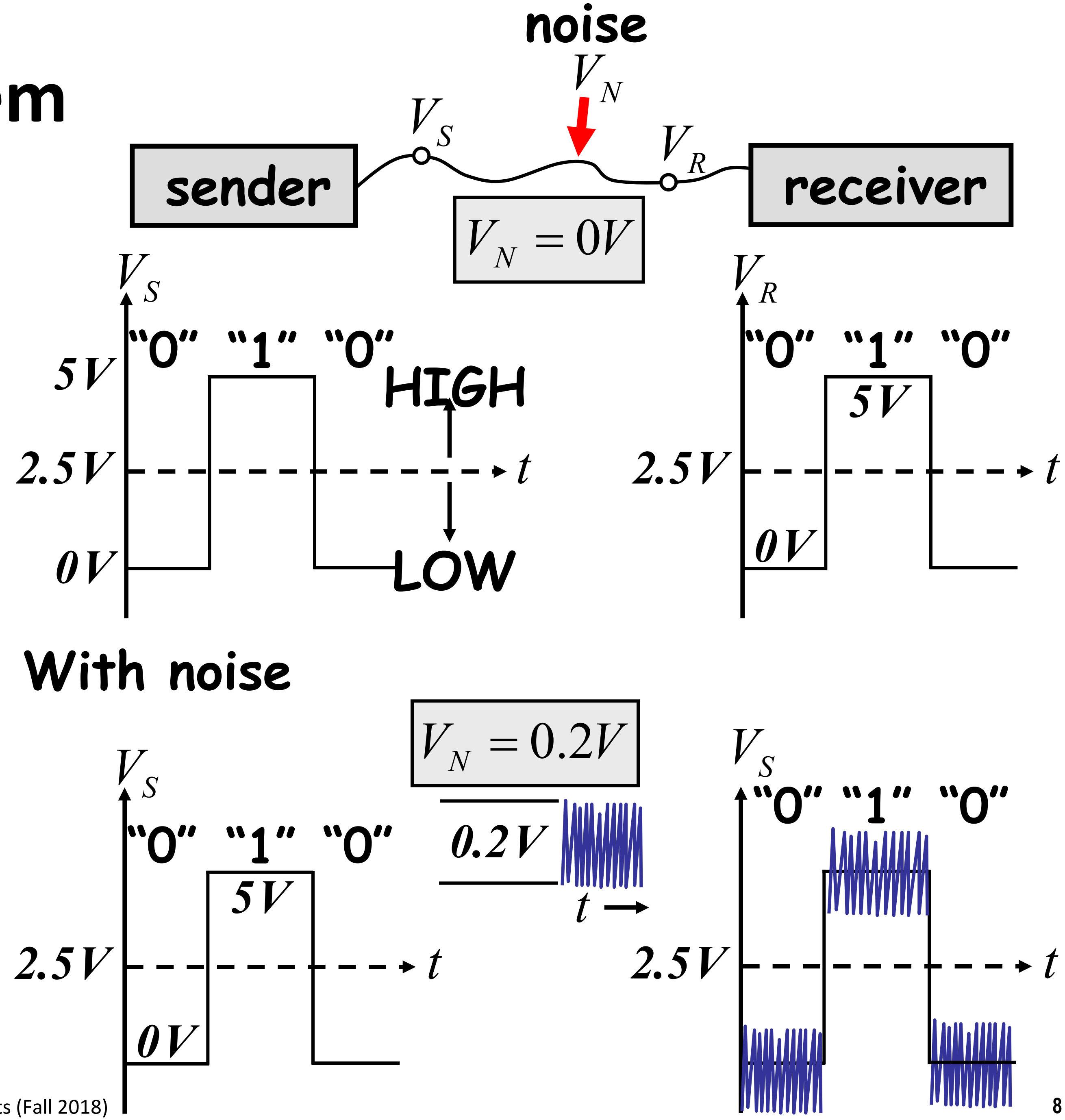
| | |
|---|---|
| HIGH | LOW |
| 5V | 0V |
| TRUE | FALSE |
| 1 | 0 |

...like two digits 0 and 1

**Why is this discretization useful?**

(Remember, numbers larger than 1 can be represented using multiple binary digits and coding, much like using multiple decimal digits to represent numbers greater than 9. E.g., the binary number 101 has decimal value 5.)

# Digital System

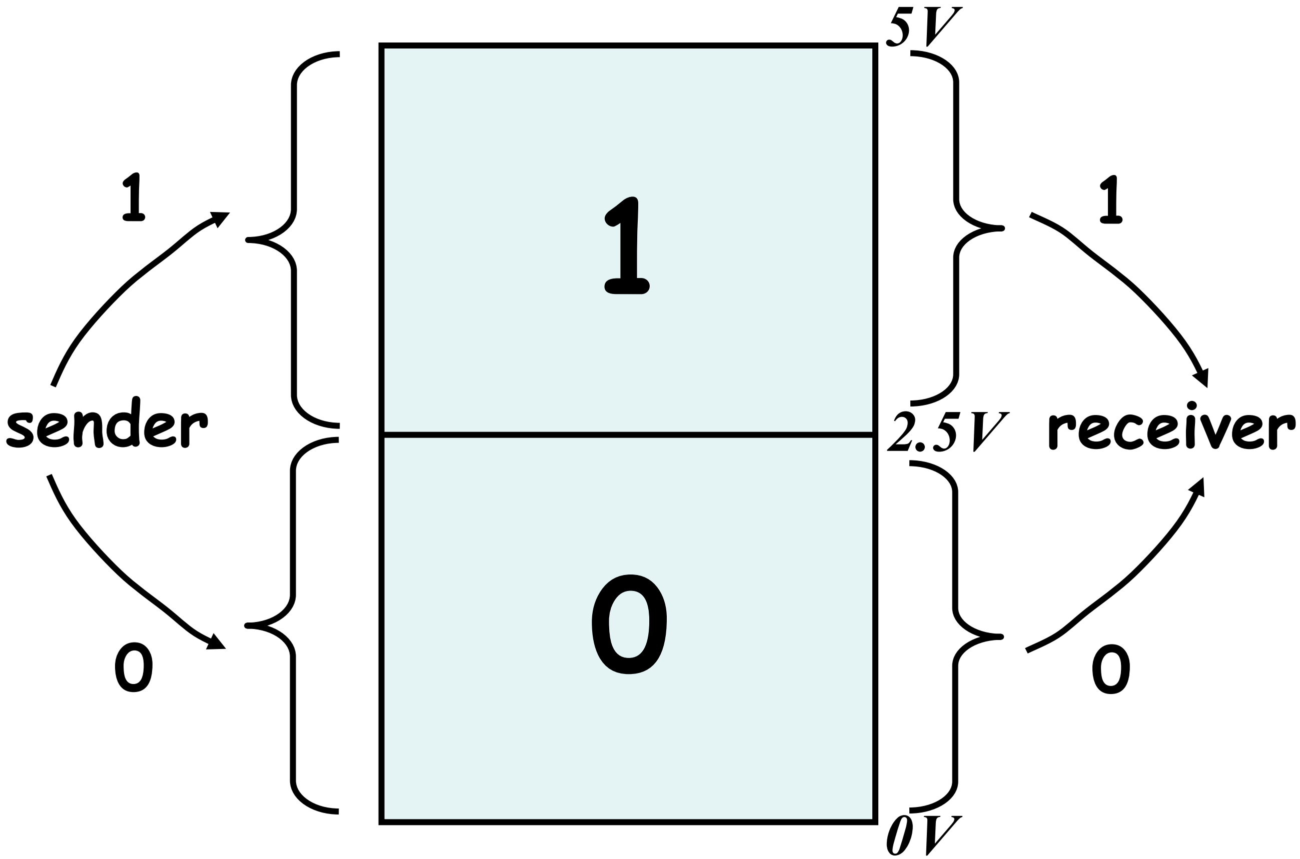


# Digital System

**Better noise immunity**
**Lots of "noise margin"**

**For "1": noise margin** $5V \ \ to\ 2.5V = 2.5V$
**For "0": noise margin** $0V \ \ to\ 2.5V = 2.5V$

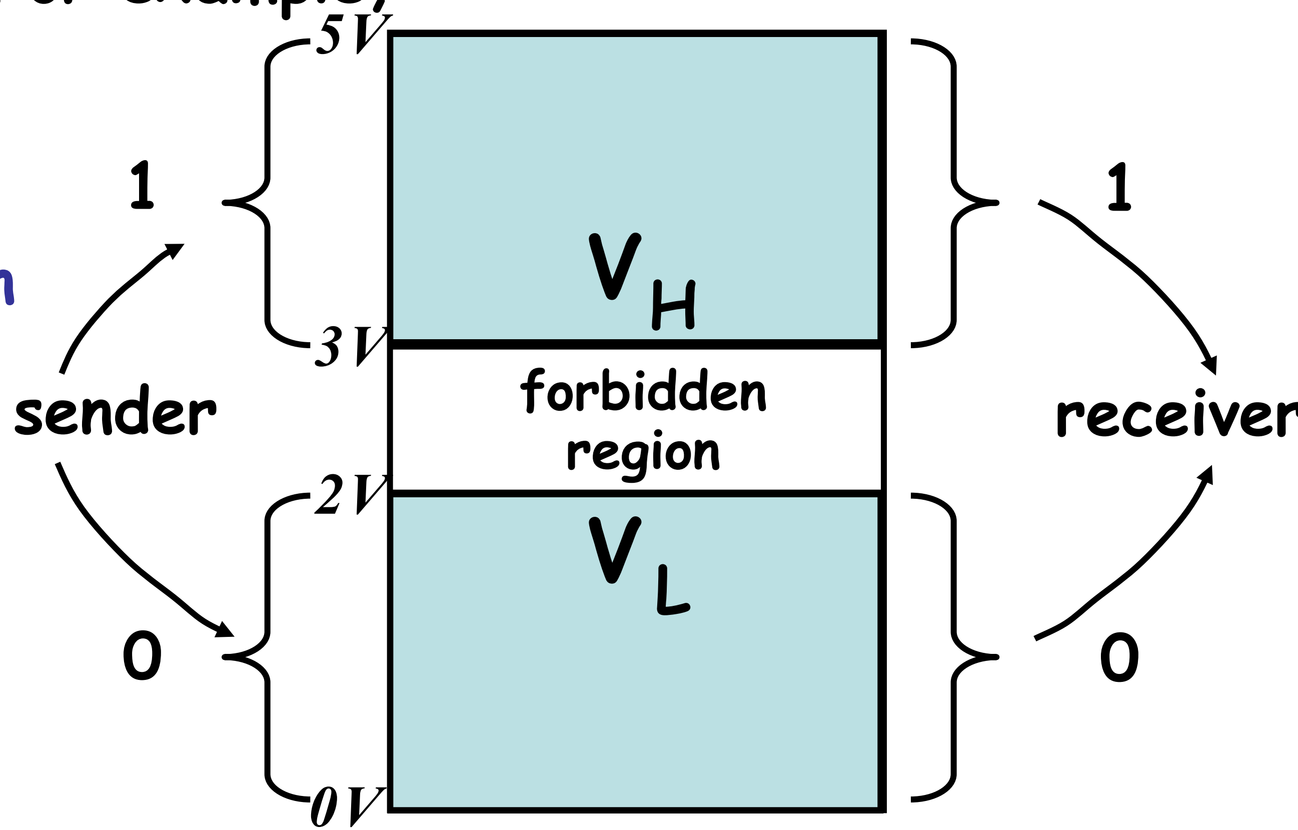
# Voltage Thresholds and Logic Values



# But, but, but … What about 2.5V?

Hmmm… create “no man’s land” or forbidden region

For example,



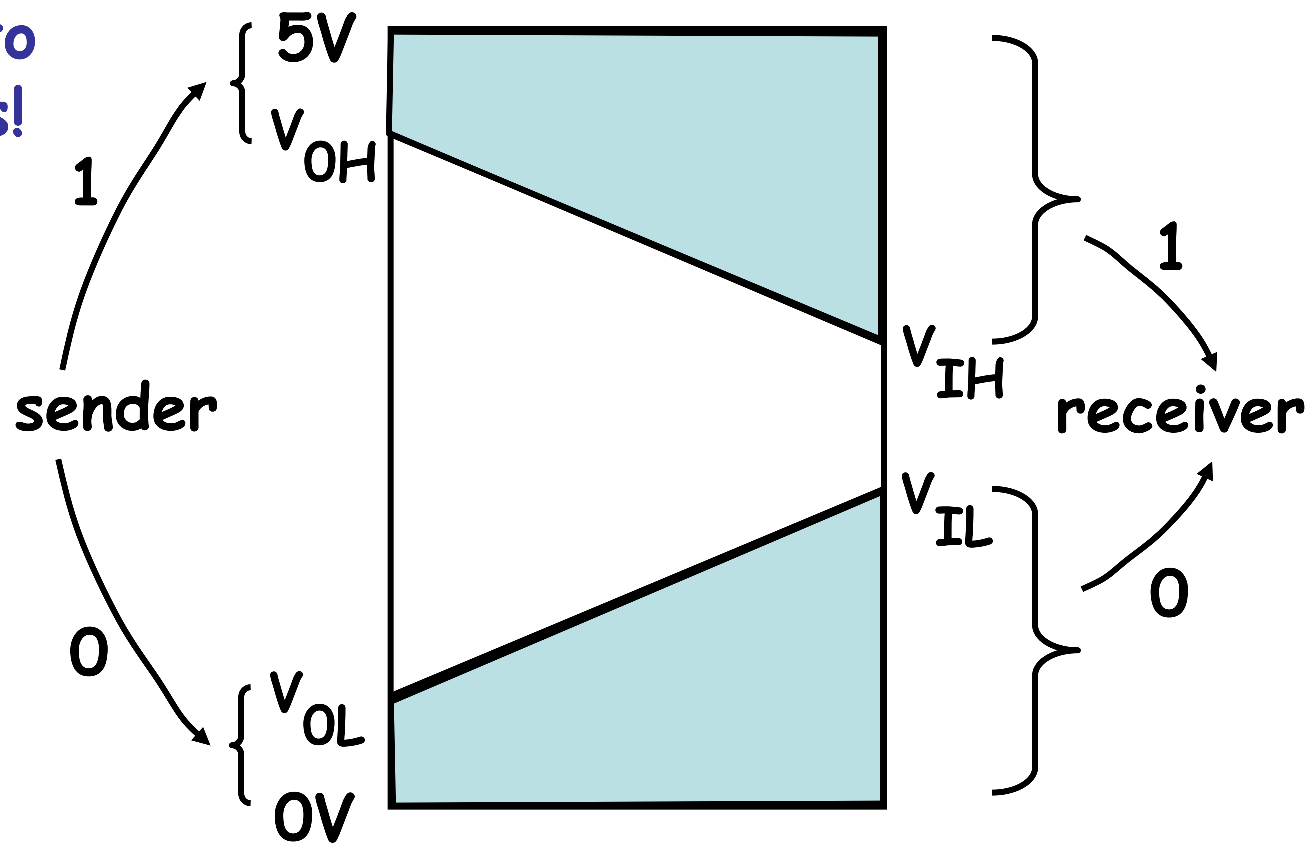

“1” $\Rightarrow$ $V_H$ $\rightarrow$ 5V

“0” $\Rightarrow$ 0V $\rightarrow$ $V_L$

# But, but, but … Where's the noise margin?

## What if the sender sent 1: $V_H$ ?

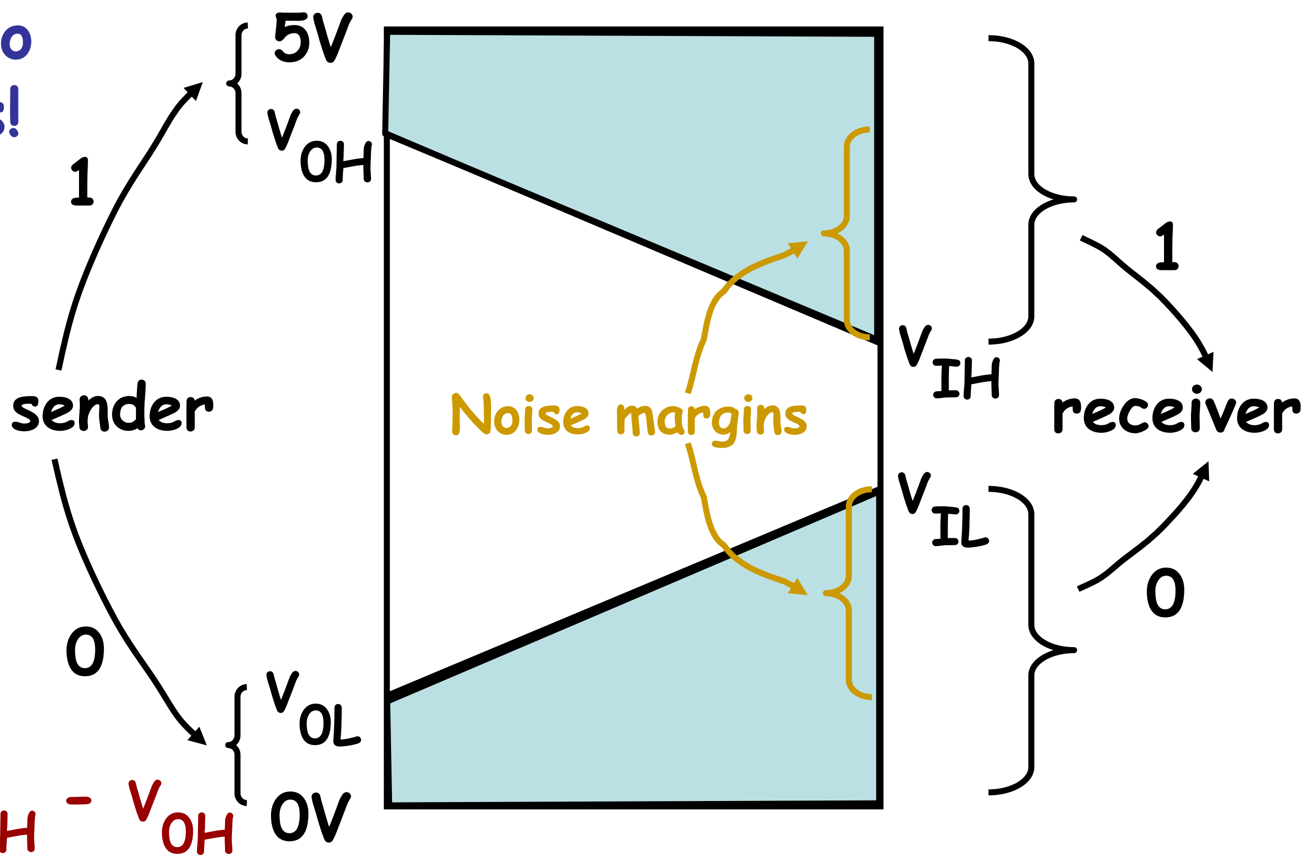
Hold the sender to tougher standards!



# But, but, but … Where's the noise margin?

## What if the sender sent 1: $V_H$ ?

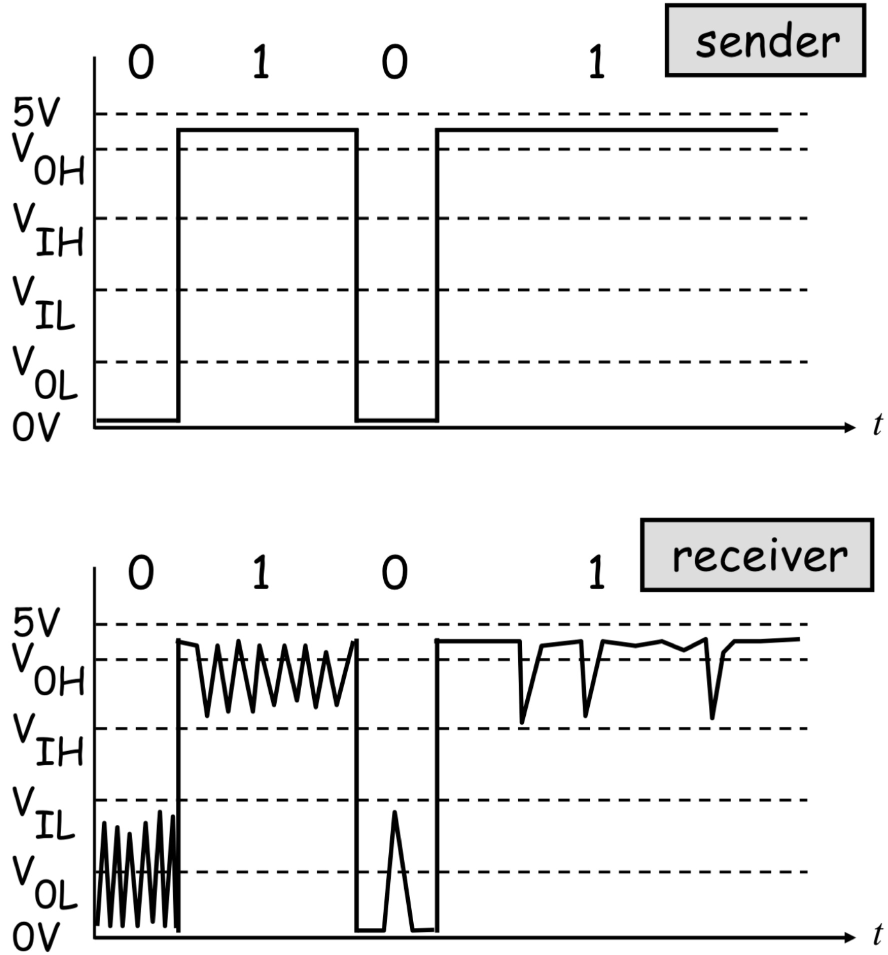
Hold the sender to tougher standards!




"1" noise margin: $V_{IH}$ - $V_{OH}$

"0" noise margin: $V_{IL}$ - $V_{OL}$

# Static Discipline




Digital systems follow **static discipline**: if inputs to the digital system meet valid input thresholds, then the system guarantees its outputs will meet valid output thresholds.

# Outline

**Textbook: Ch. 5.1-5.3**

- Voltage Levels and Static Discipline
- Boolean Logic
- Combinational Gates

# Processing digital signals

Recall, we have only two values —

1,0 ⇨ Map naturally to logic: T, F

⇨ Can also represent numbers

# Processing digital signals

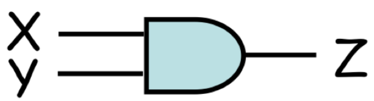
- **Boolean Logic**

⇨ If X is true and Y is true
Then Z is true else Z is false.

⇨ Z = X AND Y

$$\underbrace{Z = X \cdot Y}_{\text{Boolean equation}}$$

X, Y, Z
are digital signals
"0" , "1"

⇨ 


AND gate

⇨ Truth table representation:

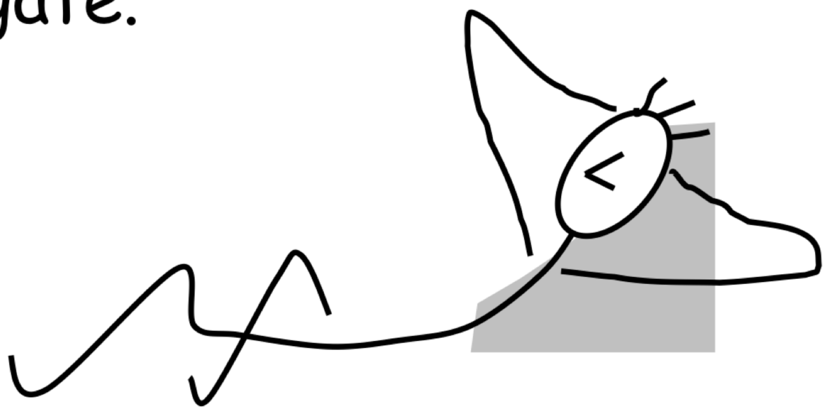
| X | Y | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Enumerate all input combinations

# Combinational gate abstraction

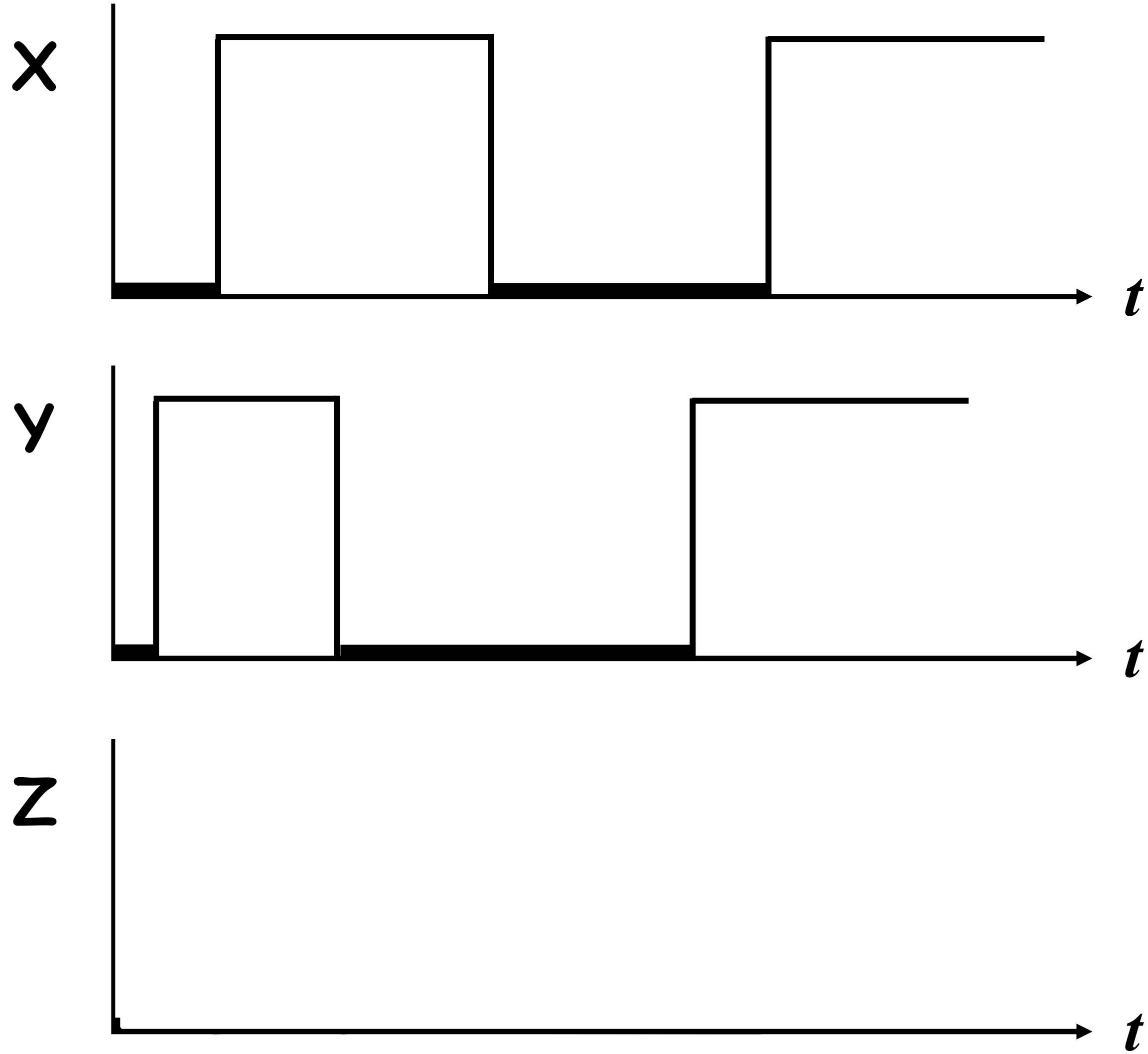
- **Adheres to static discipline**
- **Outputs are a function of inputs alone.**

Digital logic designers do not have to care about what is inside a gate.



# Combinational gate abstraction

- **Example:**
  **Z = X•Y (AND gate)**



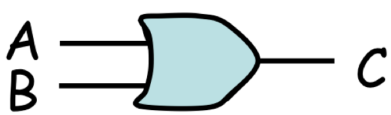
# Combinational gate abstraction

- **Another example of gate: OR gate**

If (**A** is true) OR (**B** is true)
then **C** is true
else **C** is false

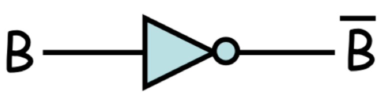
⇨ $C = A + B$ Boolean equation

OR

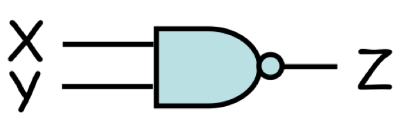
⇨ 


OR gate

More gates




Inverter




NAND

$Z = \overline{X \cdot Y}$

# Combinational gate abstraction

- **Boolean identities**

$$X \cdot 1 = X$$
$$X \cdot 0 = X$$
$$X + 1 = 1$$
$$X + 0 = X$$
$$\overline{1} = 0$$
$$\overline{0} = 1$$
$$AB + AC = A \cdot (B + C)$$

- **Digital circuits: $Z = A + \overline{B \bullet C}$**