

# COMPUTER LOGIC AND DIGITAL CIRCUIT DESIGN (PHYS306/COSC330): UNIT 2.1

---

Jordan Hanson

January 25, 2018

Whittier College Department of Physics and Astronomy

## SUMMARY

---

### Reading: DF Chapter 3-4 (Moodle)

1. Logic Gates
  - Circuit diagram
  - Truth table
  - Timing diagram
  - Boolean logic
2. Boolean algebra I
3. IC Circuits, data sheets
4. Boolean algebra II

Homework: Chapter 3, ex. 1-22 (two weeks)

## LOGIC GATES

---

A **logic gate** is a digital circuit component made of transistors, that performs a basic logic function with  $n$  inputs and  $m$  outputs.

We will cover this basic set:

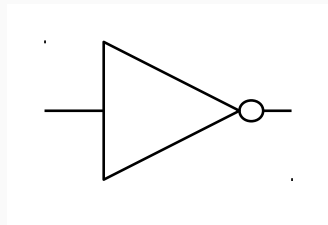
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

Each gate has  $n$  inputs, and  $m$  outputs. We represent the inputs (HIGH/LOW) with  $A, B, \dots$ , and the output usually with  $X$ .

- Circuit diagram
- Truth table
- Timing diagram
- Boolean logic

The **NOT gate**: flips the input from LOW/HIGH to HIGH/LOW.

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 1:** The NOT gate is represented with a triangle and a circle. It has one input and one output.

The **NOT gate**: flips the input from LOW/HIGH to HIGH/LOW.

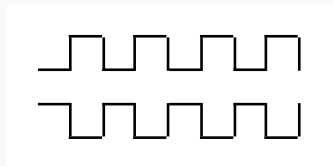
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

IN	OUT
1	0
0	1

**Table 1:** Truth table for NOT.

The **NOT gate**: flips the input from LOW/HIGH to HIGH/LOW.

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 2:** Example timing diagram for the NOT gate. (Top) example signal,  $s$ . (Bottom) NOT  $s$ , or  $\bar{s}$ .



The **NOT** gate: flips the input from LOW/HIGH to HIGH/LOW.

- NOT,  $n = 1, m = 1$

$$\text{NOT}(A) = \bar{A} \quad (1)$$

- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

The **NOT gate**: flips the input from LOW/HIGH to HIGH/LOW.

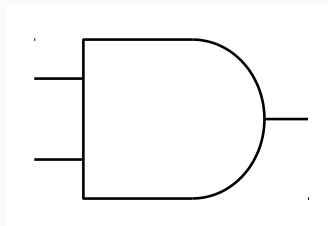
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

## In class exercises:

- Using NOT gates, create a circuit that forms the 2's complement of an 8 bit binary number.
- Once you have this, draw a timing diagram representing the conversion of 142 to the proper 2's complement, followed by the conversion of 56 to the proper 2's complement.

**The AND gate:** requires both inputs to be HIGH, for HIGH output.

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 3:** The circuit representation of an AND gate. Two inputs, one output.

The **AND gate**: requires both inputs to be HIGH, for HIGH output.

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

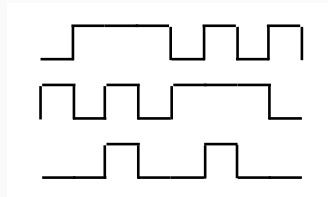
A	B	X
1	1	1
1	0	0
0	1	0
0	0	0

**Table 2:** Truth table for AND.

How many input combinations are there for  $n$ ? Specifically, how many for two-input gates? Three-input?

**The AND gate:** requires both inputs to be HIGH, for HIGH output.

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 4:** Example timing diagram for AND gate. (Top) Input A. (Middle) Input B. (Bottom) AB (A AND B).

**The AND gate:** requires both inputs to be HIGH, for HIGH output.

- NOT,  $n = 1, m = 1$

$$A \text{ AND } B = AB \quad (2)$$

- AND,  $n, m = 1$

- OR,  $n, m = 1$

- NAND,  $n, m = 1$

- NOR,  $n = 2, m = 1$

- XOR,  $n = 2, m = 1$

- XNOR,  $n = 2, m = 1$

**The AND gate:** requires both inputs to be HIGH, for HIGH output.

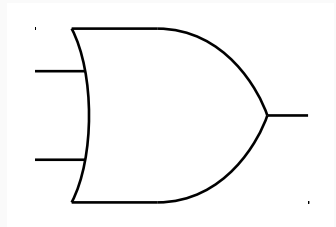
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

## In-class exercises:

- Using AND as “enable:”  
Develop a circuit that flashes a light every time the clock signal is HIGH, but only if an enable signal is also raised to HIGH.
- Derive the truth table for a 3-input AND gate. How many input combinations should there be?

**The OR gate:** requires either input to be HIGH, for HIGH output.

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 5:** The circuit representation of an OR gate. Two inputs, one output.



The **OR gate**: requires either input to be HIGH, for HIGH output.

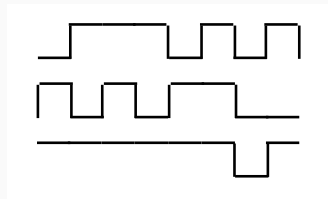
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

A	B	X
1	1	1
1	0	1
0	1	1
0	0	0

**Table 3:** Truth table for OR.

**The OR gate:** requires either input to be HIGH, for HIGH output.

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 6:** Example timing diagram for OR gate. (Top) Input A. (Middle) Input B. (Bottom)  $A + B$  (A OR B).

The OR gate: requires either input to be HIGH, for HIGH output.

- NOT,  $n = 1, m = 1$

$$A \text{ OR } B = A + B \quad (3)$$

- AND,  $n, m = 1$

- OR,  $n, m = 1$

- NAND,  $n, m = 1$

- NOR,  $n = 2, m = 1$

- XOR,  $n = 2, m = 1$

- XNOR,  $n = 2, m = 1$

The OR gate: requires either input to be HIGH, for HIGH output.

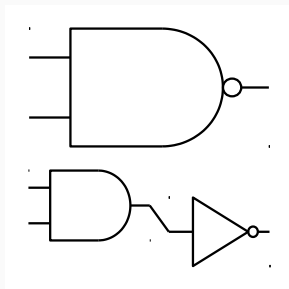
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

## In-class exercises:

- In physics experiments, sometimes we establish a *combinatorial trigger*, sometimes called a coincidence trigger. Suppose the task of three digital channels is to observe a high-energy particle pass through a detector. If there is an observation in a channel, it raises HIGH for a time called a *gate time*.
- Using OR gates, draw a circuit that triggers if **two of the three** channels observes the particle. Draw an example of a timing diagram corresponding to a trigger. (The gate time is up to you, it just means that if a channel is HIGH it stays high for at least one gate).

The **NAND gate**: requires **neither** input to be HIGH for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 7:** The circuit representation of an NAND gate. Two inputs, one output.

The **NAND gate**: requires **neither** input to be HIGH for the output to be HIGH

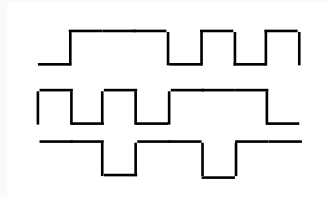
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

A	B	X
1	1	0
1	0	1
0	1	1
0	0	1

**Table 4:** Truth table for NAND.

The **NAND gate**: requires **neither** input to be HIGH for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 8:** Example timing diagram for NAND gate. (Top) Input A. (Middle) Input B. (Bottom)  $\overline{AB}$  (A NAND B or NAND AB).

The **NAND gate**: requires **neither** input to be HIGH for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

$$A \text{ NAND } B = \overline{AB} \quad (4)$$

Note: have you started to pick up the notation here? The bar is like complex conjugation, or the **compliment** of the signal.



The **NAND gate**: requires **neither** input to be HIGH for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

## In-class exercises:

- A certain fluid pump has a display that shows sensor information from two pairs of pipes, pair A and pair B. The sensors are raised HIGH if no fluid is flowing through a pipe, and there is one sensor per pipe.

The **NAND gate**: requires **neither** input to be HIGH for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

### In-class exercises:

- Design a system using NAND gates that produces three output signals: one LOW if neither pipe in pair A has fluid flowing, one LOW if neither pipe in pair B has fluid flowing, and one LOW if neither pair has fluid flowing. Create a timing diagram that shows pair A turning off, then pair B.

The **NAND gate**: requires **neither** input to be HIGH for the output to be HIGH

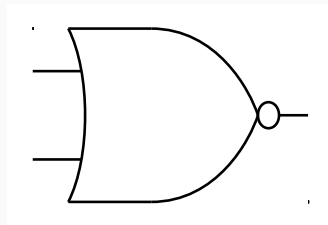
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

## In-class exercises:

- Create an OR gate with NAND and NOT gates. a) 2-input version 3) 3-input version.
- Create an AND gate with NAND and NOT gates. a) 2-input version 3) 3-input version.

The **NOR gate**: requires **both** inputs to be LOW for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 9:** The circuit representation of an NOR gate. Two inputs, one output.

The **NOR gate**: requires **both** inputs to be LOW for the output to be HIGH

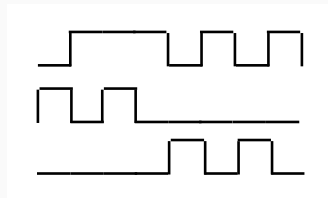
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

A	B	X
1	1	0
1	0	0
0	1	0
0	0	1

**Table 5:** Truth table for NOR.

The **NOR gate**: requires **both** inputs to be LOW for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 10:** Example timing diagram for NOR gate. (Top) Input A. (Middle) Input B. (Bottom)  $\overline{A + B}$  (A NOR B or NOR AB).

The **NOR gate**: requires **both** inputs to be LOW for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

$$A \text{ NOR } B = \overline{A + B} \quad (5)$$

The **NOR gate**: requires **both** inputs to be LOW for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

## In-class exercises:

- Create a NOR gate from a NAND gate and inverters.  
a) 2-input version b) 3-input version. Verify with either a timing diagram or truth table.



The **NOR gate**: requires **both** inputs to be LOW for the output to be HIGH

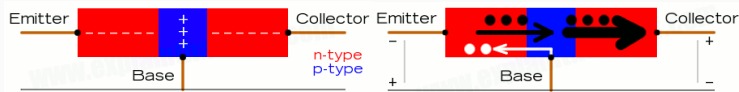
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

In-class exercise (from the text, this one is cool)

- As part of an aircraft's functional monitoring system, a circuit is required to indicate the status of the landing gears prior to landing. A green LED display turns on if all three gears are properly extended when the gear-down switch has been activated in preparation for landing. A red LED display turns on if any of the gears fail to extend properly prior to landing. When a landing gear is extended, its sensor produces a LOW voltage. When a landing gear is retracted, its sensor produces a HIGH voltage. Implement a circuit to meet this requirement.

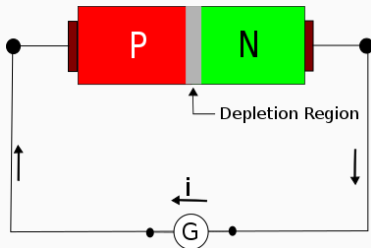
To solve this problem, let's have a quick aside as to how LED's (light emitting diodes) work...

**LED - light emitting diode.** What's a diode? What is a transistor? One kind of transistor is a p-n-p junction, or a n-p-n junction:



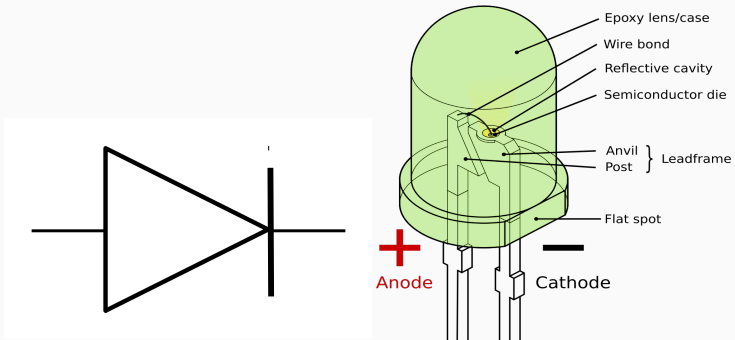
**Figure 11:** One kind of transistor is the sandwich of semiconductors.

**LED - light emitting diode.** What's a diode? Like a transistor, minus one layer. Diodes will only conduct in one direction.

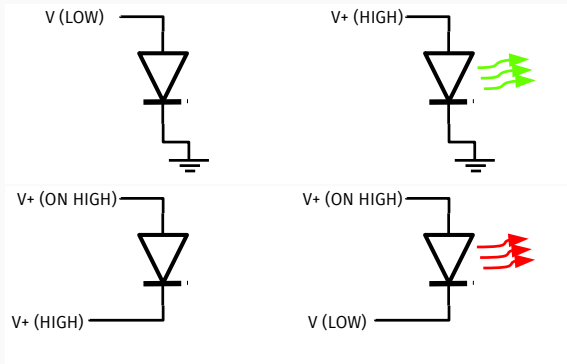


**Figure 12:** A diode is a sandwich of only two layers of semiconductor rather than three.

LED - light emitting diode. What's a diode? Like a transistor, minus one layer. Diodes will only conduct in one direction.



**Figure 13:** (Left) The symbol for a diode. Current flows towards the flat line. (Right) Some diodes emit visible photons when conducting electricity.



**Figure 14:** (Upper left): The green LED is held off by a LOW signal, no voltage gradient. (Upper right) The green LED is activated by a HIGH signal, a voltage gradient. (Lower left): The red LED is held off by a HIGH signal, no voltage gradient. (Lower right): The red LED is activated by a LOW signal, a voltage gradient.

The **NOR gate**: requires **both** inputs to be LOW for the output to be HIGH

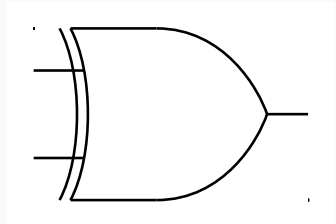
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

So now we can solve this problem:

- As part of an aircraft's functional monitoring system, a circuit is required to indicate the status of the landing gears prior to landing. A green LED display turns on if all three gears are properly extended when the gear-down switch has been activated in preparation for landing. A red LED display turns on if any of the gears fail to extend properly prior to landing. When a landing gear is extended, its sensor produces a LOW voltage. When a landing gear is retracted, its sensor produces a HIGH voltage. Implement a circuit to meet this requirement.

The **XOR gate**: requires **exactly one** input to be HIGH for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 15:** The circuit representation of an XOR gate. Two inputs, one output.

The **XOR gate**: requires **exactly one** input to be HIGH for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

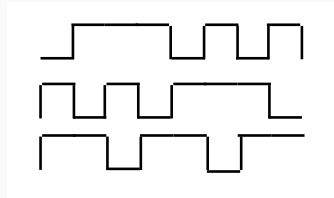
A	B	X
1	1	0
1	0	1
0	1	1
0	0	0

**Table 6:** Truth table for XOR.



The **XOR gate**: requires **exactly one** input to be HIGH for the output to be HIGH

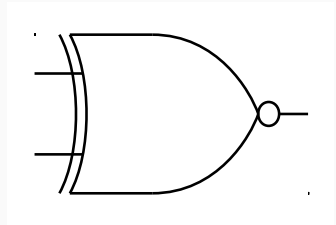
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 16:** Example timing diagram for XOR gate. (Top) Input A. (Middle) Input B. (Bottom) Output

The **XNOR gate**: requires **both** inputs to be HIGH or LOW for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 17:** The circuit representation of an XNOR gate. Two inputs, one output.

The **XNOR gate**: requires **both** inputs to be HIGH or LOW for the output to be HIGH

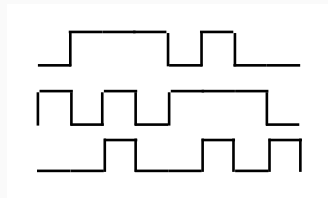
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

A	B	X
1	1	1
1	0	0
0	1	0
0	0	1

**Table 7:** Truth table for XNOR.

The **XNOR** gate: requires **both** inputs to be HIGH or LOW for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 18:** Example timing diagram for XNOR gate. (Top) Input A. (Middle) Input B. (Bottom) Output

The **NOR gate**: requires **both** inputs to be LOW for the output to be HIGH

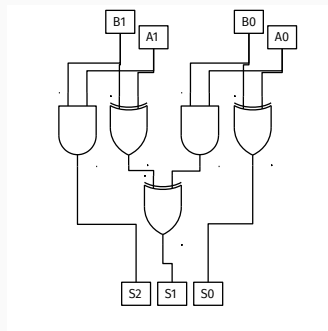
- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$

**In-class exercises:**

- Show that an XOR gate is a two-bit adder, neglecting the carry bit.
- Create an 2-bit adder from XOR and AND gates.

The **NOR gate**: requires **both** inputs to be LOW for the output to be HIGH

- NOT,  $n = 1, m = 1$
- AND,  $n, m = 1$
- OR,  $n, m = 1$
- NAND,  $n, m = 1$
- NOR,  $n = 2, m = 1$
- XOR,  $n = 2, m = 1$
- XNOR,  $n = 2, m = 1$



**Figure 19:** Example of a circuit that adds two-bit digital numbers.

## CONCLUSION

---

### Reading: DF Chapter 3-4 (Moodle)

1. Logic Gates
  - Circuit diagram
  - Truth table
  - Timing diagram
  - Boolean logic
2. Boolean algebra I
3. IC Circuits, data sheets
4. Boolean algebra II

Homework: Chapter 3, ex. 1-22 (two weeks)