Nile University
جـامعـة النيـل

School of
Information Technology
and Computer Science

CSCI101

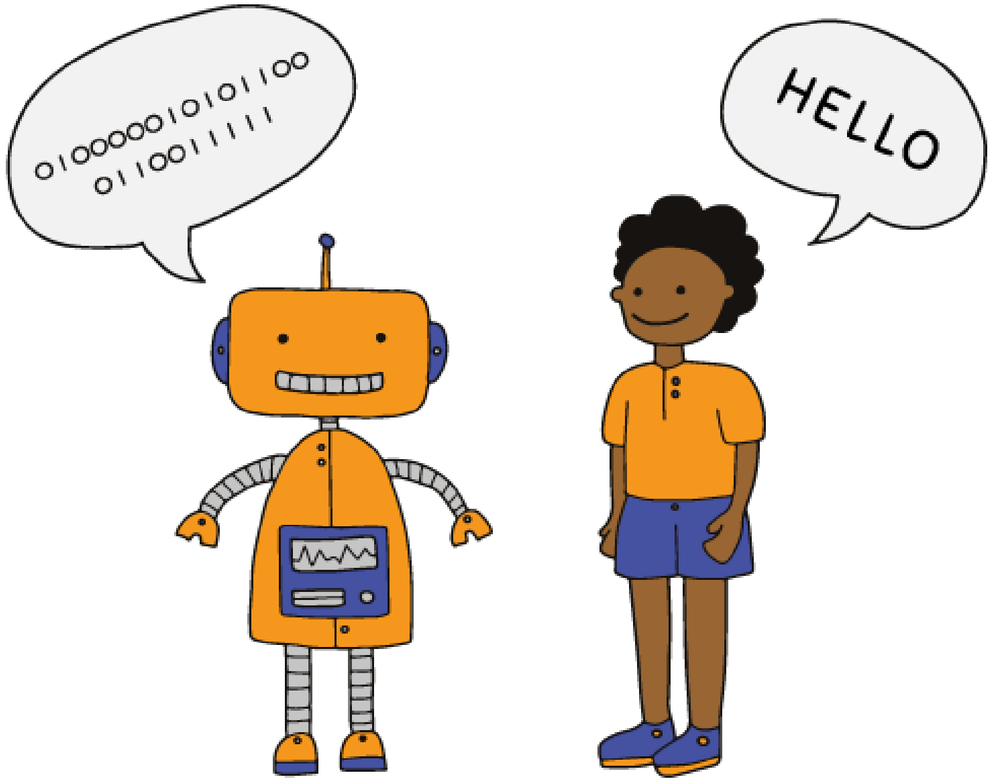
Logic Gates & Numbering

TUTORIAL 2 - WEEK 2

CSCI101 Team

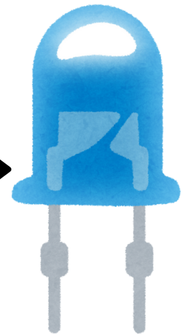
Nile University

Machine Language

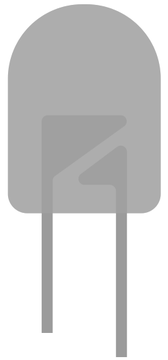


What Are Bits?

1

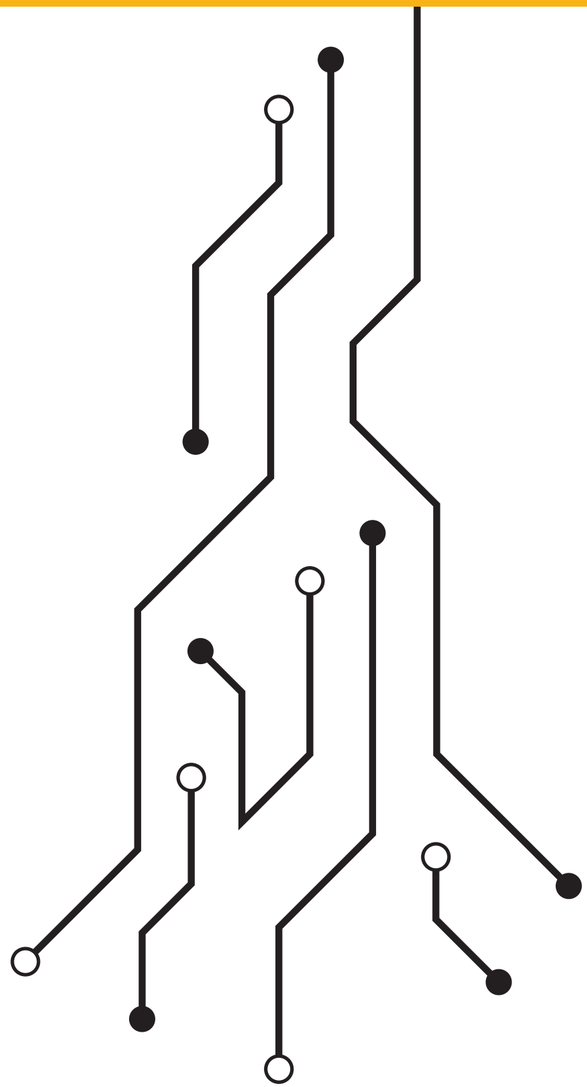


0

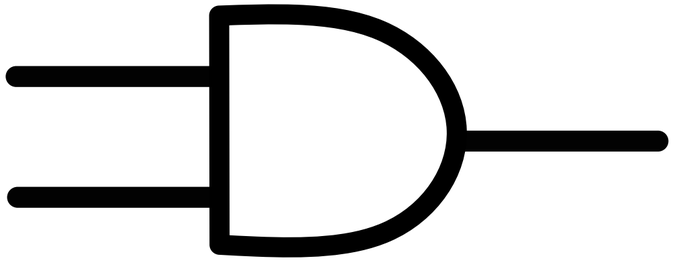


How Does A Computer Work?

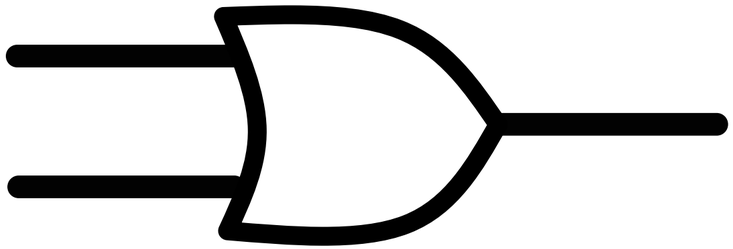
- Logic Gates are the fundamental building blocks of digital circuits.
- Logical functions
- Each gate has one or more input signals and produces an output signal based on a certain logic.



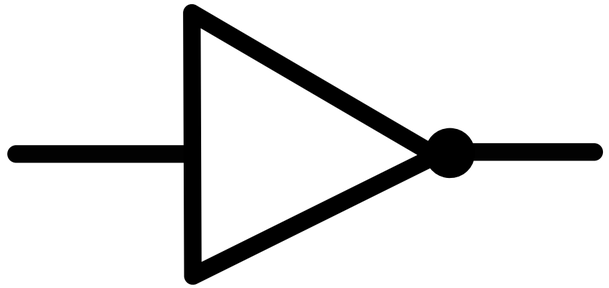
Basic Logic Gates



- AND Gate

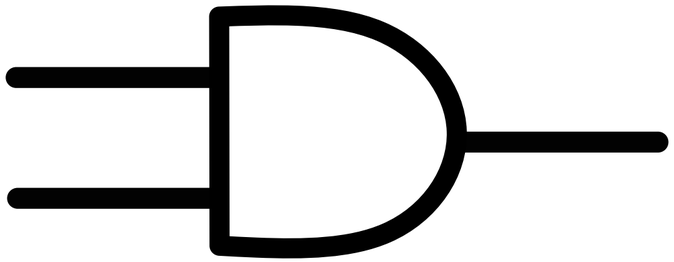


- OR Gate



- NOT Gate

‘AND’ Gate

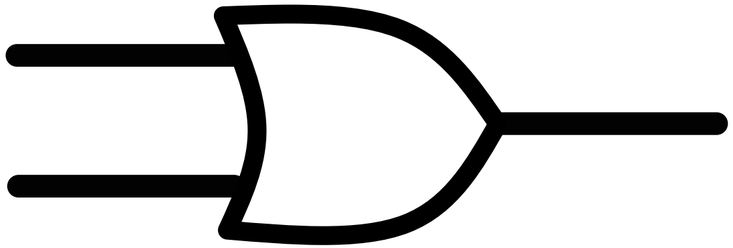


- **AND Gate**

Z = A . B

Input		Output
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

‘OR’ Gate

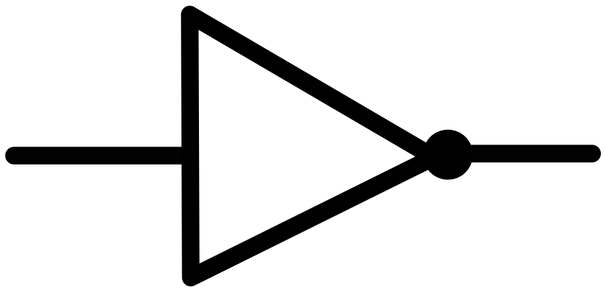


- **OR Gate**

Z = A + B

Input		Output
A	B	A AND B
0	0	0
0	1	1
1	0	1
1	1	1

‘NOT’ Gate

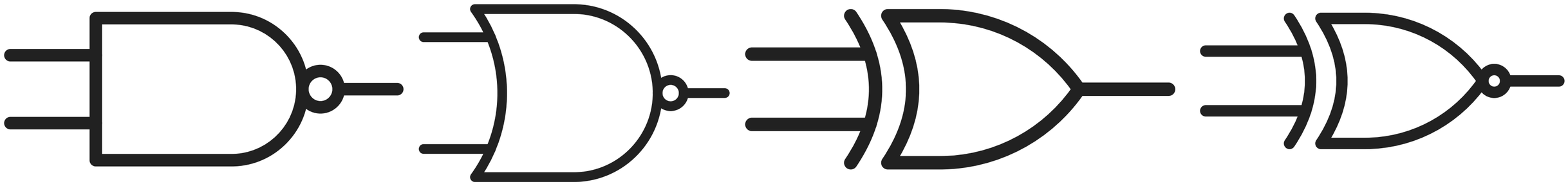


- NOT Gate

$$Z = A$$

Input	Output
A	NOT A
0	1
1	a

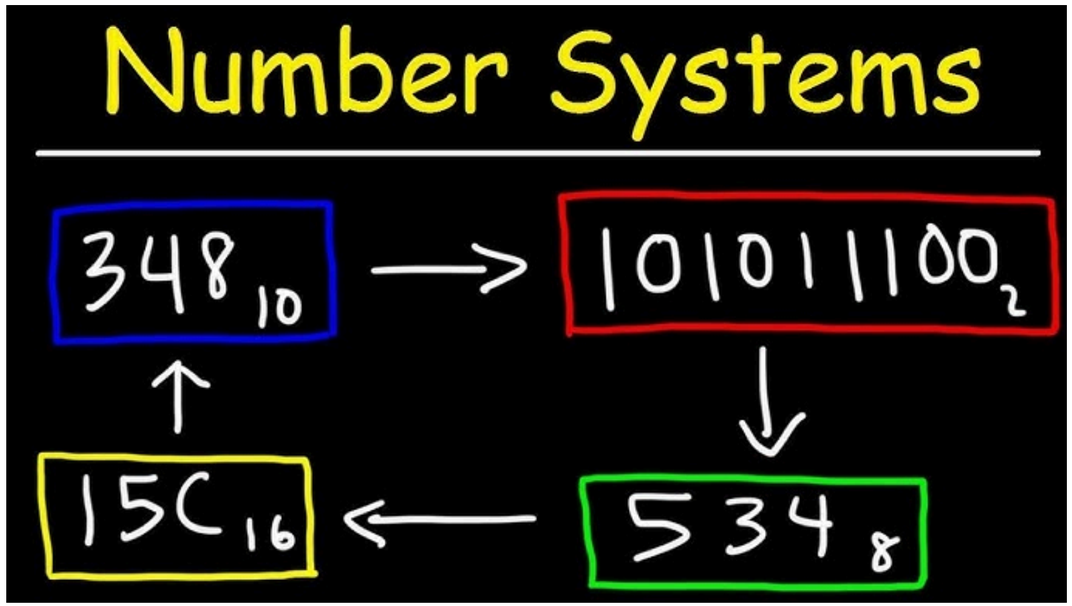
Derived Logic Gates



- NAND Gate
- NOR Gate
- XOR Gate
- XNOR Gate

Numbering Systems

- There are multiple numbering systems that we use for different reasons. Some of the systems are:
  - Decimal
  - Binary
  - Octal
  - Hexadecimal



Decimal Numbering

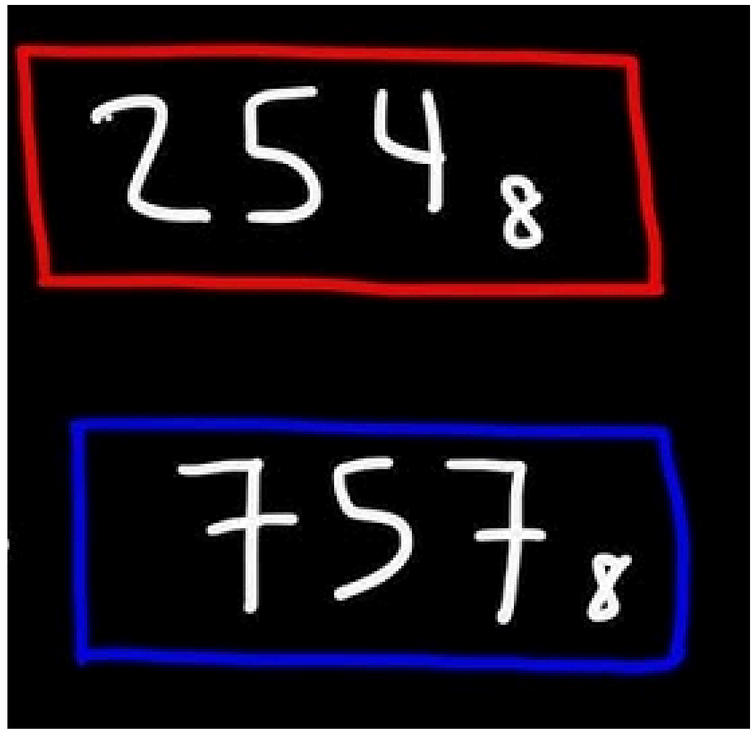
- The Decimal numbering system consists of the numbers we've seen our entire lives, from 0 to 9
- Since this system consists of 10 number (0 to 9), it is known as BASE 10

Binary Numbering

- The Binary Numbering system consists of only 0s and 1s
- Different combinations of these 0s and 1s have different meanings and values!
- Since the binary system has 2 numbers only (0 and 1), it is known as BASE 2

Octal Numbering

- From its name, Octal = Eight, consists of 8 numbers from 0 to 7
- Octal is known as BASE 8 number system

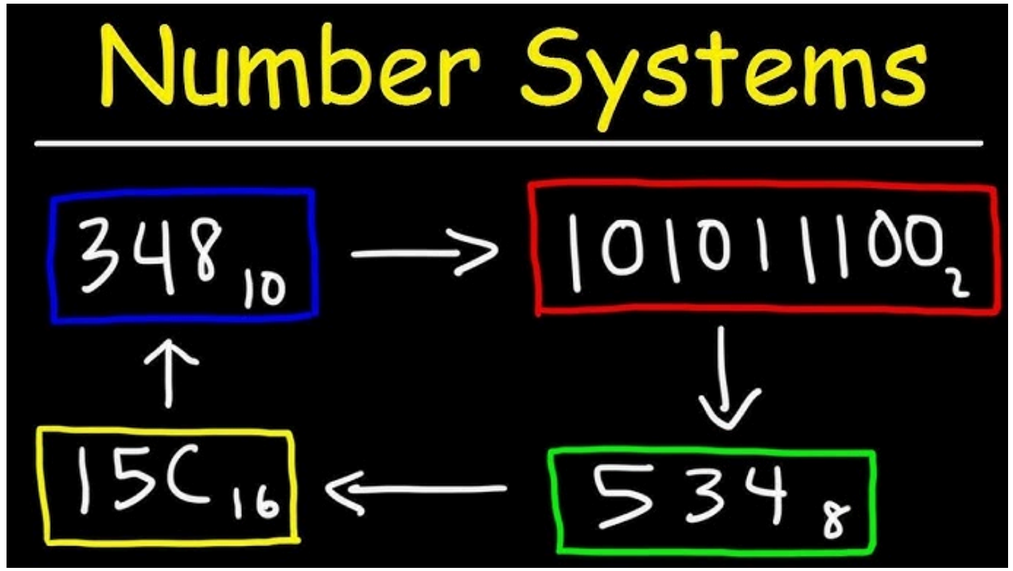


Hexadecimal Numbering

- Hexadecimal is slightly different from the previous system, since it consists of both digits AND letters.
- Hexadecimal consists of 16 symbols:
  - 10 digits (from 0 to 9)
  - 6 letters (from A to F)
- Known as BASE 16

Conversions Between Numbering Systems

- **Each numbering system has a different purpose.**
  - **To be able to use all the systems for any reason, we must learn how to convert between them and move from one system to the other!**



Decimal to Binary Conversion

Decimal	Binary
8	1000
12	1100
145	1001 0001

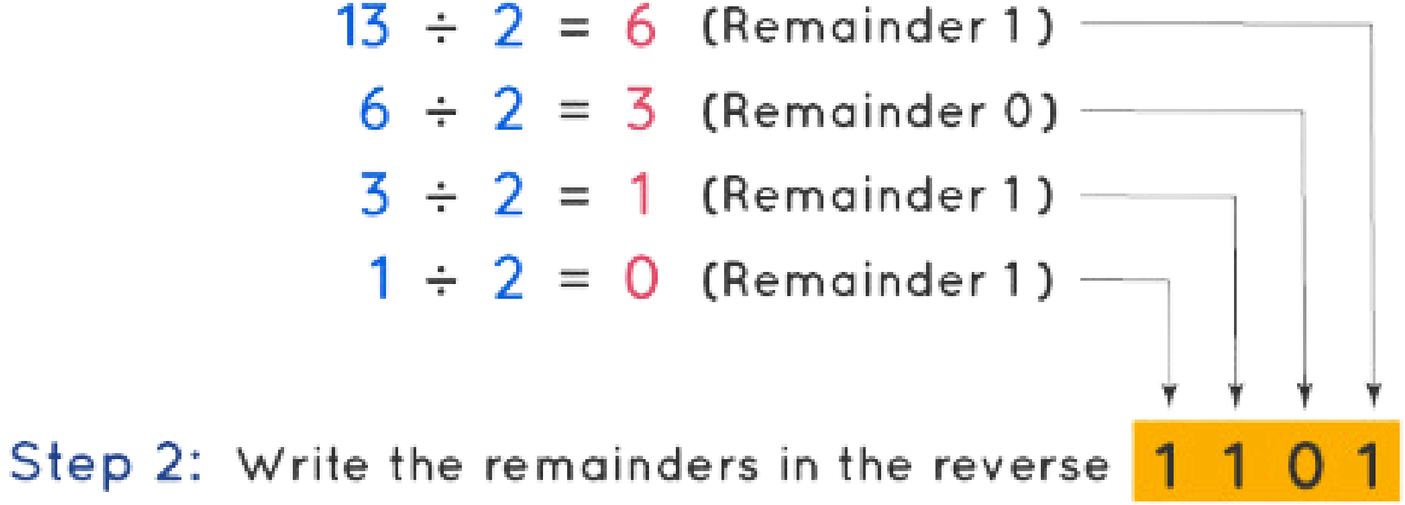
Decimal to Binary Conversion

- There are TWO methods of converting the Decimal numbering system to the Binary numbering system
  - Remainder Method
  - Weights Method

Decimal to Binary Conversion: Remainder Method

The remainder method focuses on using division to find the binary value of the number

Step 1: Divide the given number 13 repeatedly by 2 until you get '0' as the quotient



Step 2: Write the remainders in the reverse order

$\therefore 13_{10} = 1101_2$

(Decimal) (Binary)

Decimal to Binary Conversion: Weights Method

Each position in the binary system has what we call a "weight"

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Decimal to Binary Conversion: Weights Method

Lets convert 75 to binary using the weights

128	64	32	16	8	4	2	1

Since 75 lies between 128 and 64, we will place a 1 at 64 as follows:

0	1						

The remaining will be 75-64 = 11

Decimal to Binary Conversion: Weights Method

Lets convert 75 to binary using the weights cont.

128	64	32	16	8	4	2	1

Since 11 is between 16 and 8, we will place a 1 at the smaller value, 8, and add 0s before it

0	1	0	0	1			

The remaining will be 11-8 = 3

Decimal to Binary Conversion: Weights Method

Lets convert 75 to binary using the weights cont.

128	64	32	16	8	4	2	1

The remaining 3 lies between 4 and 2, which means we will place the 1 at 2

0	1	0	0	1	0	1	

The remaining will be 3-2 = 1

Decimal to Binary Conversion: Weights Method

Lets convert 75 to binary using the weights cont.

128	64	32	16	8	4	2	1

Since the remaining value is 1, we will place 1 in that position

0	1	0	0	1	0	1	1

The remaining will be 1-1 = 0

Decimal to Binary Conversion: Weights Method

Solution:

$$75_{10} = 01001011_{2}$$

Binary Numbering

8 → 1000

12 → 1100

CSCI101

Binary to Hexadecimal Conversion

Hexa	8	9	A	B	C	D	E	F
Binary	1000	1001	1010	1011	1100	1101	1110	1111

- Hexadecimal numbers are base-16, using digits 0-9 and letters A-F.
- Each hexadecimal digit represents four binary digits (bits).
- Hexadecimal is often used in programming to simplify binary code representation.

Binary to Hexadecimal Conversion

1011 1010

11 10

B A

Octal Numbering

- Octal numbers are base-8, using digits 0-7.
- Each octal digit represents three binary digits (bits).
- Octal is commonly used in computing for compact binary representation.

Binary to Octal Conversion

$$= (1010111100)_2$$

$$= (001\ 010\ 111\ 100)_2$$

$$= (1\ 2\ 7\ 4)_8$$

$$= (1274)_8$$

Numbering Systems Conversion: Example 1

Convert $(2A7)_{16}$ into Binary

Numbering Systems Conversion: Example 2

Convert $(1011101)_2$ into Decimal

Tutorial Task

TASK 1

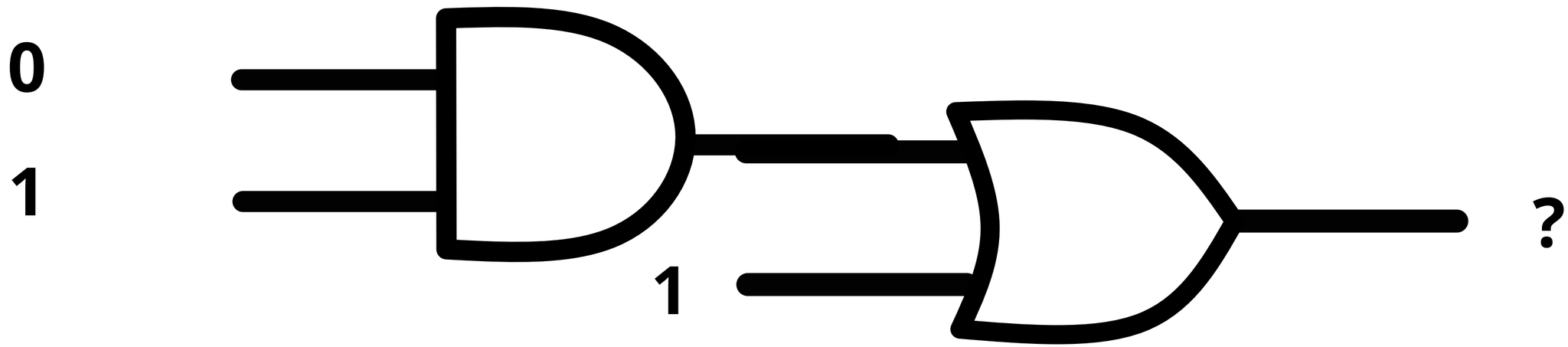
Convert $(FF)_{16}$ into Octal on Paper

TASK 2

Convert $(2F)_{16}$ into its octal equivalent.

Tutorial Task

TASK 3



Calculate the output of the gate.

THANK YOU

Any Questions?

YOU CAN ALSO CHECK

https://www.youtube.com/watch?v=Vq04T0Cwj5Y

Resources:

https://www.cuemath.com/numbers/decimal-to-binary/

https://www.log2base2.com/number-system/how-to-convert-hexadecimal-to-octal-with-example.html