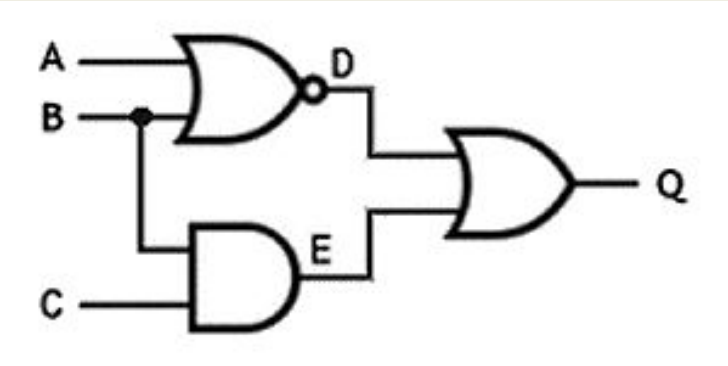
Logic Gates

Group 11:
Gisela Lima, Paul Lewis, Matthew Lyon, Andrew Leaper, Chi K Leung, Jiajie Liu, Adam Lutton

What are logic gates?

- Logic gates are electronic systems that process one or more boolean inputs to create a boolean output.
- There are several different logic gates, each one performs a different function. The seven main ones we will cover are NOT, AND, OR, NAND, NOR, EOR and NEOR.
- Logic gates can be combined to make more complex systems.

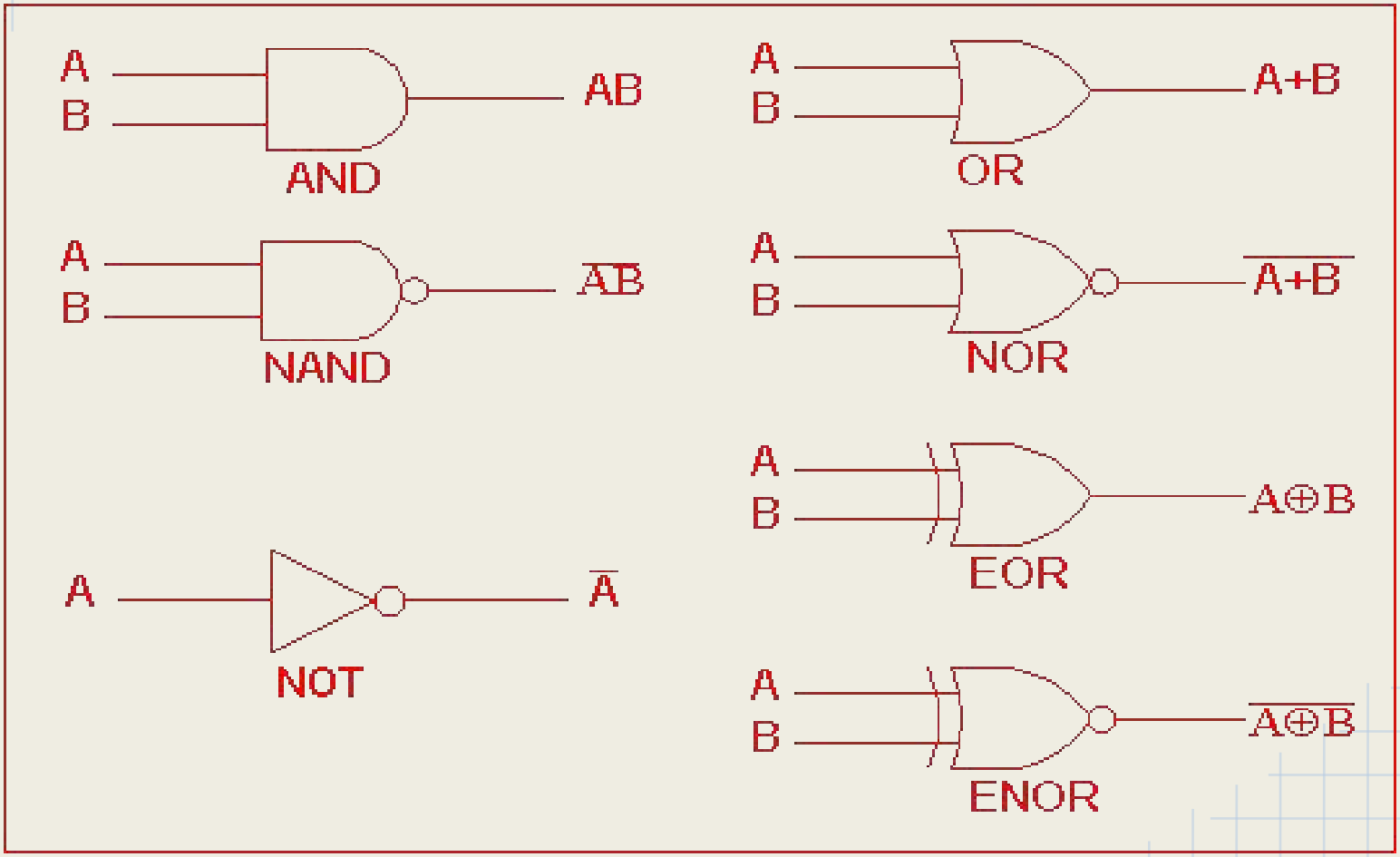


Logic Gates and Symbols

In the next slide we will show a diagram of the seven main logic gates, for each logic gate there is a boolean statement.
Each statement has a corresponding symbol equation and they are:

AND = A.B or AB
OR = A+B
NOT = A
NAND = A.B or AB
NOR = A+B
EOR = A+B
NEOR = A+B

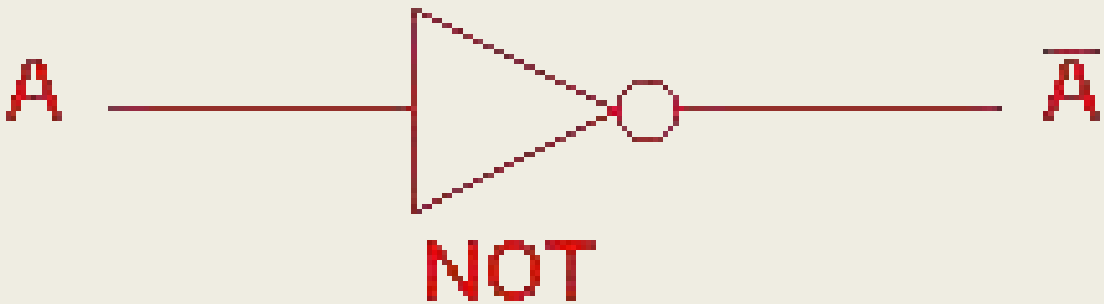
A circuit diagram of the seven main logic gates



NOT

- NOT are logical gates that returns a high output (1), when the signal inputs are low (0).

- This gate can work as long as one input returns low. For example: If we have two inputs; A and B, and B returns a 0, then it is NOT B, or B.



AND

What is an AND gate?

- AND gates is a logical gate that returns a high (1) output if and only if both input signals are high (1).
- It could be said that this gate only works when both inputs have the high value.
- Meaning that they need each other to work, for example you could not have a light bulb on if you don’t have electricity as you need both sources to have light.

by Gisela Lima

Truth Tables

- Truth tables are a very useful method to use for gates, in this case when working with AND gates, truth tables are good as they are based on the idea of using true and false.
- Where value 0 is equal to false and there is no outcome and value 1 is equal to true as there is an outcome given

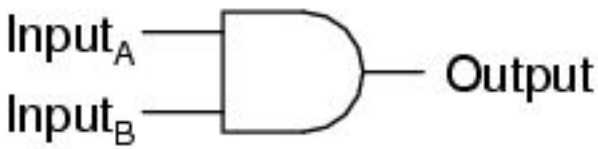
by Gisela Lima

AND

Representing gates using Truth Tables

- To represent gates we can use truth tables, using values 1 and 0, as gates function in a logical manner
- then gives out a result of the inputs in the gates. For example, AND gate truth table which looks like this:

2-input AND gate



A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

By Gisela Lima

OR

A and B represent the input and O represent the output.

Logic function O=A+B

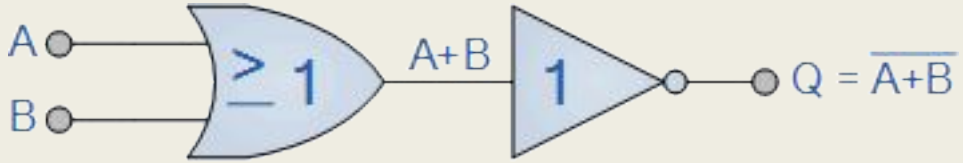
If one or both input is equal 1 which represent true, the output will be 1. the output will only be 0 if both input is equal to 0.

Truth Table for an OR gate		
Input (A)	Input (B)	Output(O)
0	0	0
1	0	1
0	1	1
1	1	1

NOR

NOR Gate

- The output is high only when neither A nor B is high. That is, it is normally high but any kind of non-zero input will take it low.
- The NOR gate and the NAND gate can be said to be universal gates since combinations of them can be used to accomplish any of the basic operations and can thus produce an inverter, an OR gate or an AND gate.
- The non-inverting gates do not have this versatility since they can't produce an invert.
- The **Logic NOR Gate** or Inclusive-NOR gate is a combination of the digital logic OR gate with that of an inverter or NOT gate connected together in series.
- The NOR (Not - OR) gate has an output that is normally at logic level "1" and only goes "LOW" to logic level "0" when **ANY** of its inputs are at logic level "1".



2-input "OR" gate plus a "NOT" gate

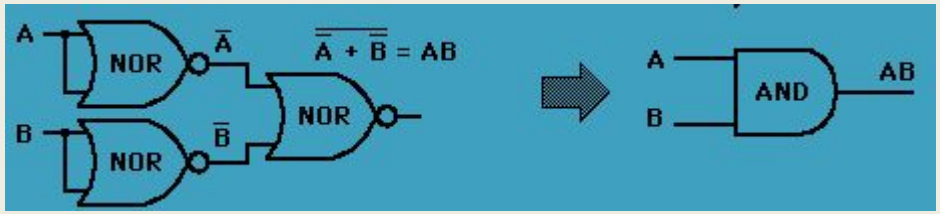
NOR

Make a NOT from a NOR

Connect a single input to both gate input pins. When A is true output will be false, When A is False output will be true.

A	NOT (A OR A)	OUTPUT		A	NOT A	OUTPUT
1	0	0		1	0	0
0	1	1		0	1	1

Make an AND FromNORs

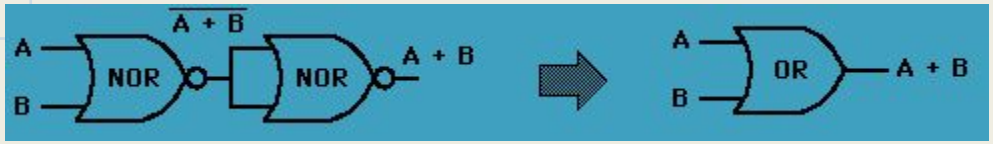


We have 2 of our made not gates connected to a NOR.

A	B	NOT A	NOT B	NOT (NOT A OR NOT B)	OUTPUT		A	B	A AND B	OUTPUT
1	0	0	1	0	0		1	0	0	0
1	1	0	0	1	1		1	1	1	1
0	0	1	1	0	0		0	0	0	0
0	1	1	0	0	0		0	1	0	0

NOR

Make an OR from NORs

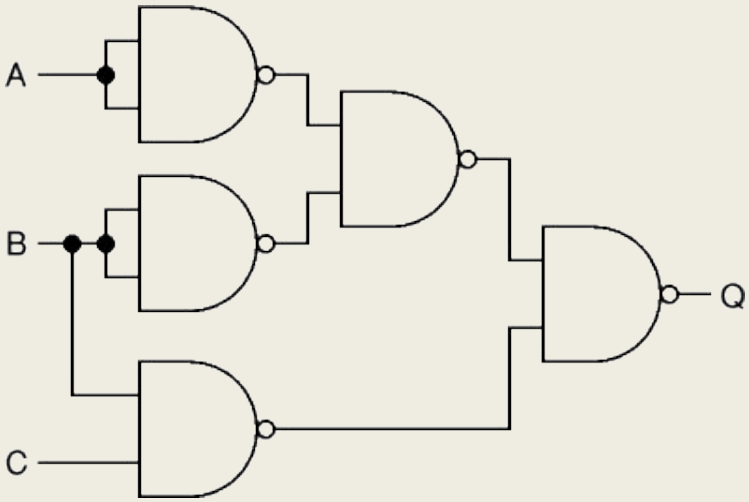


We connect the output of a Nor to both inputs of another NOR.

A	B	NOT (A OR B)	NOT (NOT A OR B)	OUTPUT
1	0	0	1	1
1	1	0	1	1
0	0	0	1	1
0	1	1	0	0

A	B	A OR B	OUTPUT
1	0	1	1
1	1	1	1
0	0	1	1
0	1	0	0

NAND



- The **Logic NAND Gate** is a combination of the digital logic AND gate with that of an inverter or NOT gate connected together in series.
- The NAND (Not - AND) gate has an output that is normally at logic level "1" and only goes "LOW" to logic level "0" when **ALL** of its inputs are at logic level "1".
- The inputs of the gate *must* be connected, either to LOW or to HIGH, and *must not* be left open circuit.
- This is the function of the input switches with their pull-down resistors. To avoid loading the output of the gate, a transistor switch indicator circuit should be used.

EOR

- The EOR Gate also known as Exclusive OR gate, is a Logic Gate that returns a High(1) Output if the inputs are different.

- If the inputs are the same the output will always be Low(0)!

Logic function $O=A \oplus B$

Truth Table for an EOR gate		
Input (A)	Input (B)	Output(O)
0	0	0
1	0	1
0	1	1
1	1	0

NEOR

- The NEOR (Not Exclusive OR)

gate is an exclusive or gate where the output has been inverted.

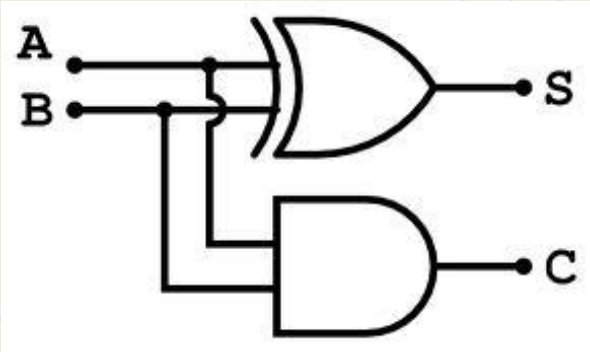
- This means that the output is high only if both inputs are the same, regardless of them being high or low.

Logic function $O=\overline{A \oplus B}$

Truth Table for an NEOR gate		
Input (A)	Input (B)	Output(O)
0	0	1
1	0	0
0	1	0
1	1	1

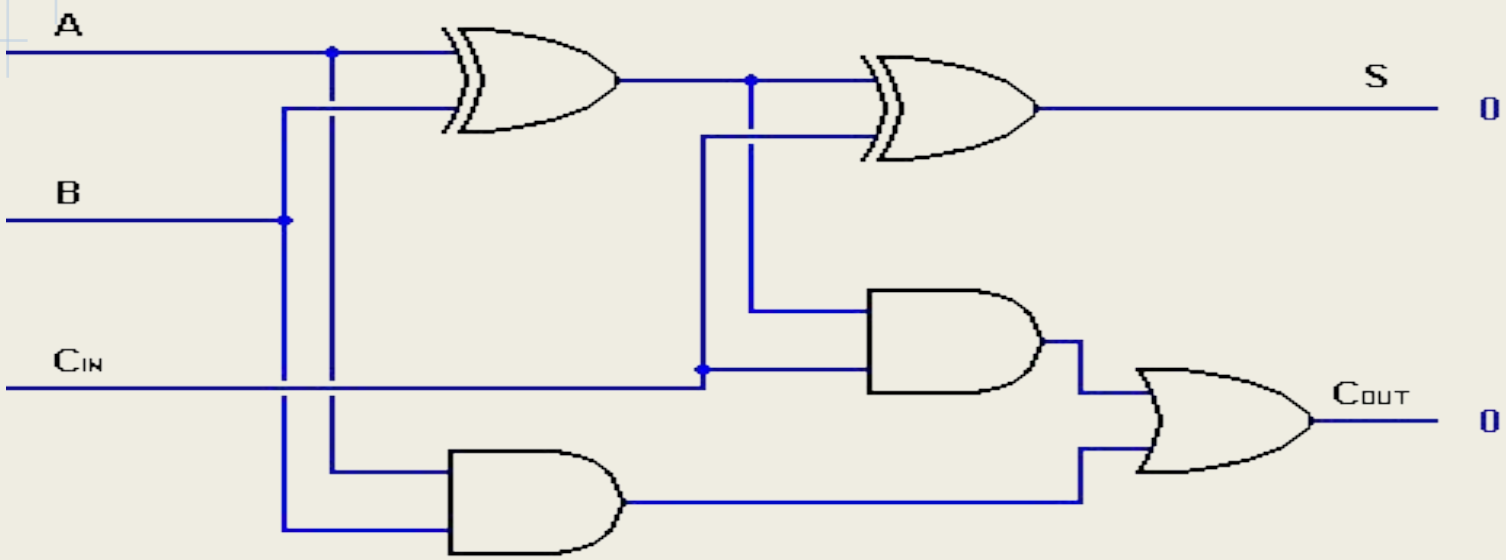
Half- Adder

- Adders are circuits that combine together by using different types of gates.
- To make adders 2 numbers have to be added, these number are the outcome of the gates once they have been joint together.
- For example, if you use 2 inputs A and B,then the sum of these gates will be represent by the bit and carry bit (C) as the output, however if A and B are the input bits, then the sum bit(S) is XOR of A and B then the carry bit of (C) will be AND of A and B.



by Gisela Lima

Full Adder

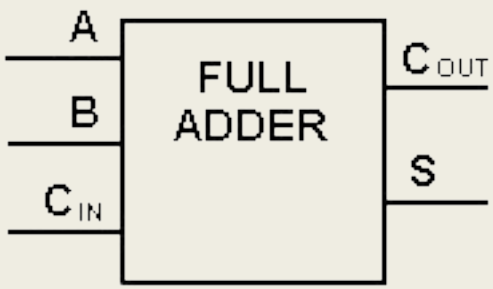


- The full adder is a bit more difficult to explain because it has more inputs than half adder, full adder consists in 3 inputs and outputs.

- The main inputs are A and B and the third input is C(carry), when the full adder logic is designed, which are used to create a byte wide adder and the output of one adder can create an input to the next adder.

by Gisela Lima

Full Adder



- On the other hand, full adders could find half adders very useful, as it can calculate part of the sum of the two inputs A and B and the other half can be used to add the carry input, which then will produce a final value.

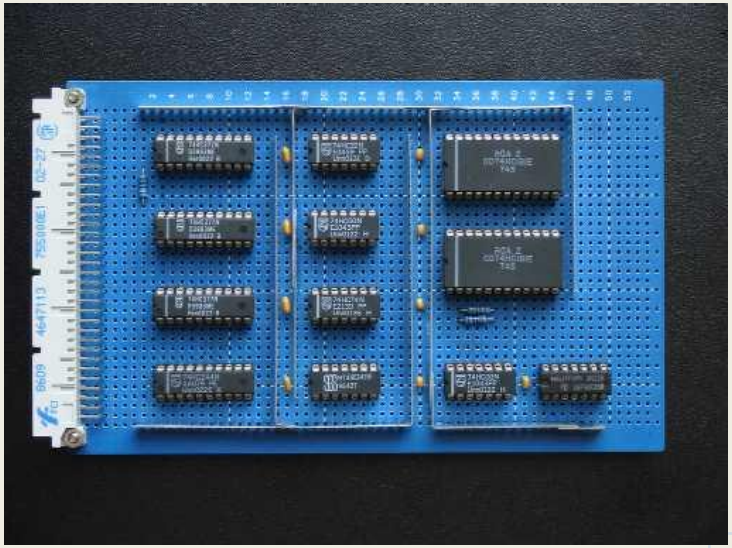
- This value is then joint with the final value of inputs A and B, then generates a final result in full.

by Gisela Lima

ALU AND GATES

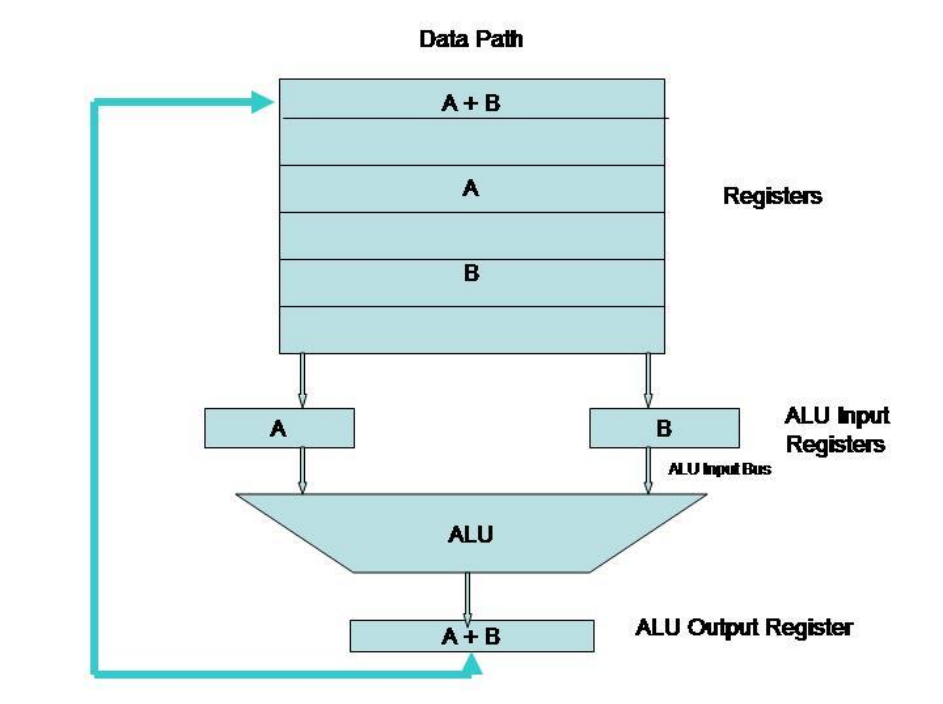
We all know that computers need certain components to be able to work, such as CPU,ALU and registers.

.ALU is considered to be the heart of a CPU, which means that if the computers did not have ALU then is the same, for us humans not having a heart and not being being able to live.



by Gisela Lima

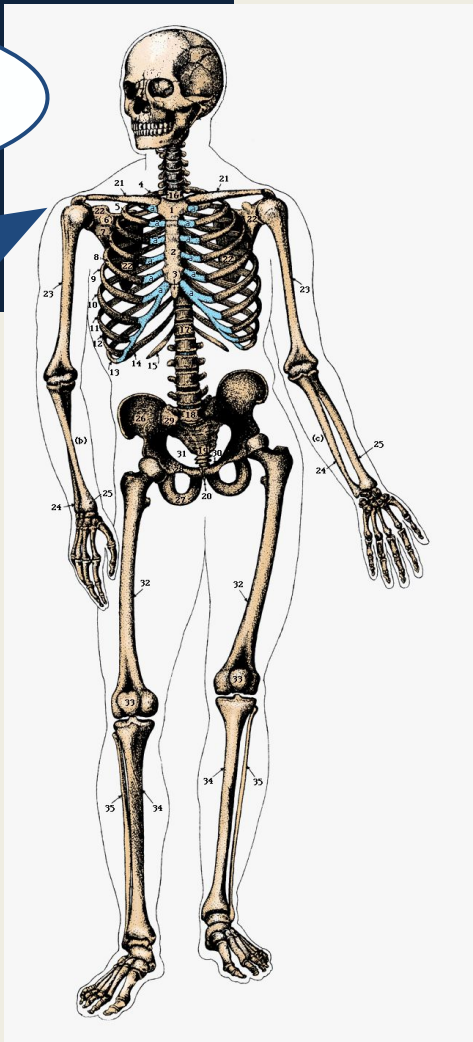
ALU and gates operations



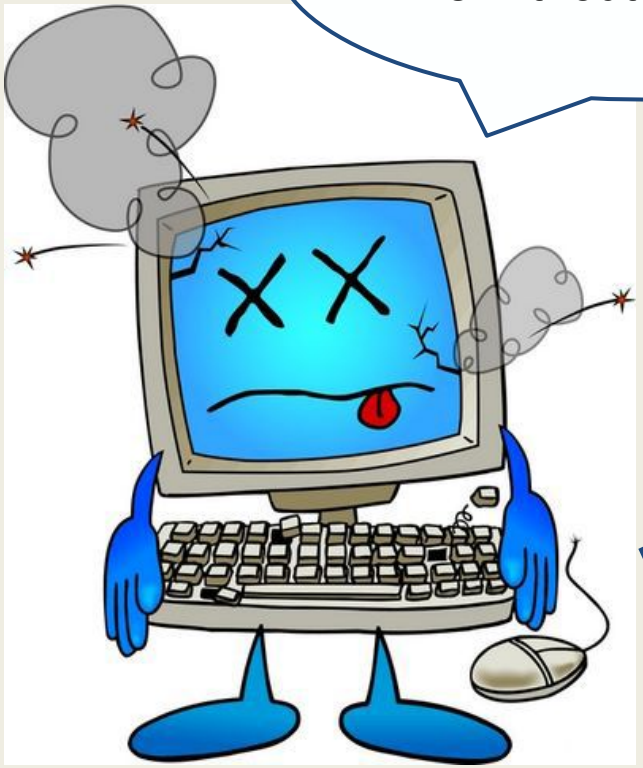
ALU is a part of the microprocessor that handles all Boolean and mathematical operations, for example it will deal with logical values of true or false and also add gates together to make the CPU perform correctly.

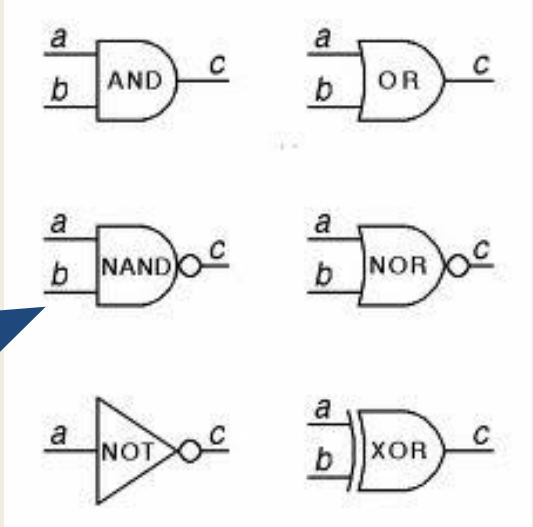
by Gisela Lima

Hey I can't cope without a heart!!!!



Thats nothing I can't live without these.....

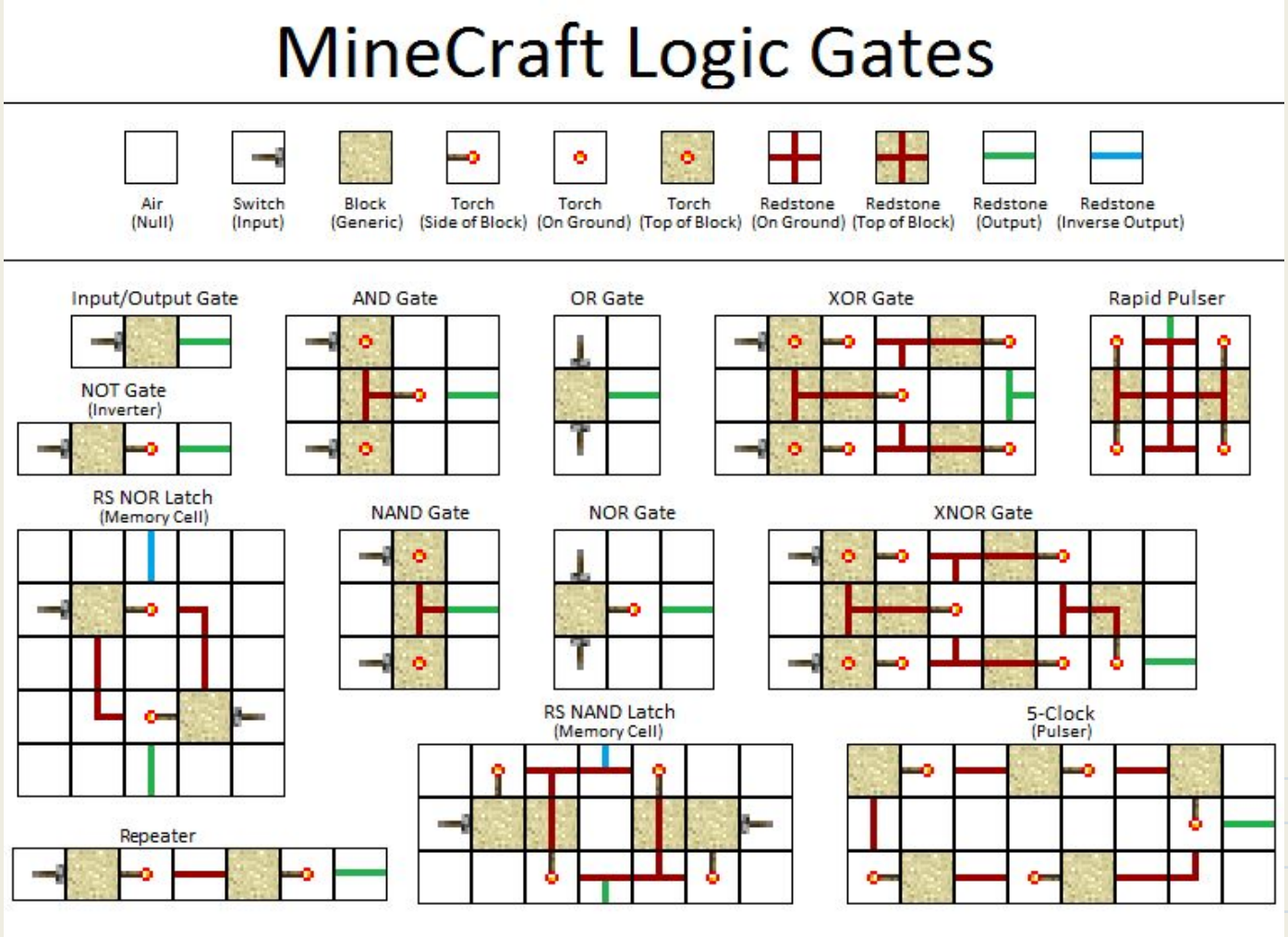




by Gisela Lima

More complex systems

Real World Examples



Conclusion

- CPU relies on ALU, which operates the logical calculations that then perform actions in the CPU.
- ALU is very essential in order for the CPU to work in a logical way, as it deals with basic operations by adding or subtracting gates which then gives us an overall result.
- When designing a CPU you must know what actions that you want your system to perform, according to the ALU,as this is the component essential to make actions work in an effective way.

by Gisela Lima