Computer Science

Supplement.

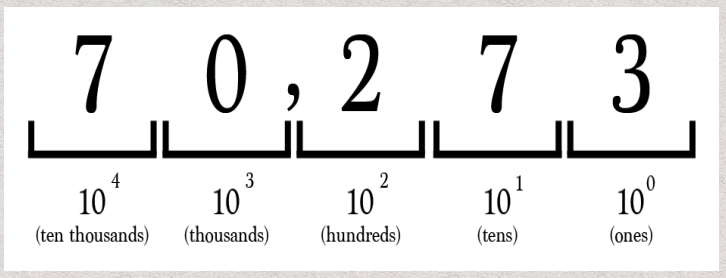
I. How the Software and Hardware work together ?

- Software is nothing but data which is a list of instructions that are given to the hardware.
Executing software is ultimately just reading those basic instructions and then doing what they say. where "reading" means transmitting the data containing the instructions from the storage device to the CPU and "doing" ultimately means the CPU calculates and process the data (a set o binary numbers).

II. What is the Binary System ?

- The binary system is based on two symbols: 0 and 1. Thus, you count only 0 and 1 and there isn't any symbol for two and it is represented by ten. Just like there's units, tends, hundreds and thousands in the decimal number system, the binary number system contains units, fours, eights, sixteens and so on.
One binary number represent the BIT and with 8 BITS, we can represent a BYTE.

Binary System .



Decimal System .

III. Why Computers use Binary ?

- computers use electrical signals to transform and handle data, in each component on it has a lot of transistors, each transistor act as a bit (1 or 0), it can be ON or OFF.

- With the help of switches(transistors), you can encode numbers into binary format. This system could be used by several digital devices.

- Switches controlling electrical signals are what most computers' circuitry act as and 'on' and 'off' are the only two states required by these switches. This means that just two numbers: 0 and 1, are required by the switches to represent each state.

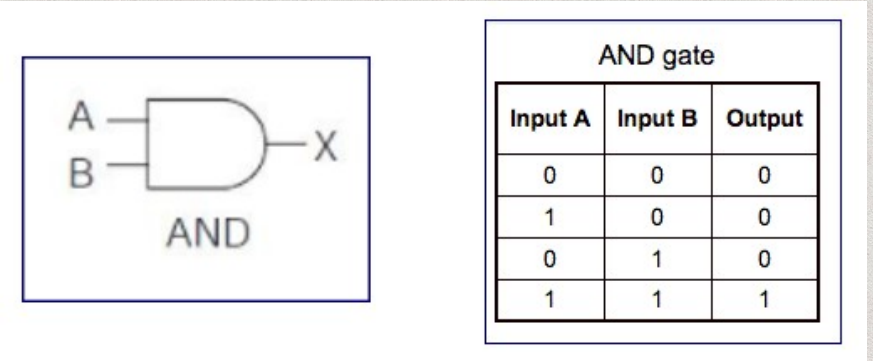
III. Boolean Logic :

- Boolean logic is a form of algebra in which all values are reduced to either TRUE or FALSE. Boolean logic is especially important for computer science because it fits nicely with the binary numbering system, in which each bit has a value of either 1 or 0. Another way of looking at it is that each bit has a value of either TRUE or FALSE.

- Boolean is centered around three simple words known as Boolean Operators :

- AND :

  - This logic says that if both the comparative values have a True value (1) then the outcome would be a value of TRUE (1).

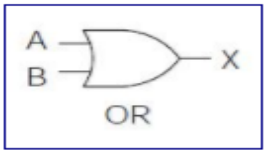


AND gate

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

- OR:

  - This logic says if that if either of the comparative values have a True value (1) then the outcome would be a value of TRUE (1).

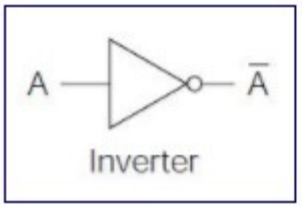


OR gate

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

- NOT:

  - This logic simply reverses a given value. For example, if the given value is a True value then this value will invert it to False and if it is False value then it will be inverted to a True value.



Input	Output
0	1
1	0