

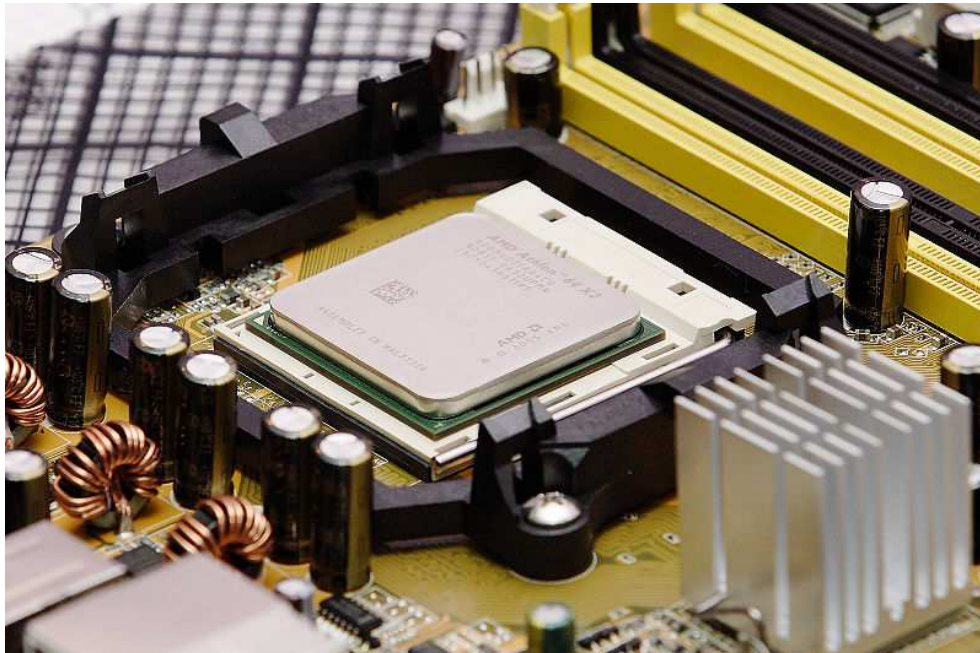
The human computer Role-sheet: ALU

You're the ALU. ALU stands for “arithmetic and logic unit”. You're the only part in the main computer that can do any maths. The other parts are the control unit, an automaton that calls for your services every now and then, and the memory, which stores the data on which you do the maths.

The ALU is capable of performing two classes of operations: arithmetic and logic. The set of arithmetic operations that a particular ALU supports may be limited to adding and subtracting or might include multiplying or dividing, trigonometry functions (sine, cosine, etc) and square roots. Some can only operate on whole numbers (integers) whilst others use so-called “floating point format” to represent numbers with decimal places- albeit with limited precision.

An ALU may also compare numbers and return boolean truth values (true or false) depending on whether one is equal to, greater than or less than the other (“is 64 greater than 65?”).

Logic operations involve Boolean logic: AND, OR, XOR and NOT. These can be useful both for creating complicated conditional statements and processing boolean logic. (we won't deal with this in this course, ask me if you're interested)



Preparation

The ALU in our human computer is composed of 3 inputs and 2 outputs (which we collectively refer to as “ports”). Each input/output port is role-played by one person (hence a total of 5 persons).

1. Each of the 5 persons chooses one port in the following list:

| |
|--|
| Input A (receives the first argument in the operation to be performed, e.g. A+B) |
| Input B (receives the second argument to be performed, e.g. A+B) |
| Input O (receives the identifier of the operation to be performed, e.g. addition) |
| Output S (outputs the status of the ALU’s operation. If S=true, then the following output ® contains the result of the operation requested using inputs A,B,O. If S=false, then the output ® is not ready, and its value is undefined) |
| Output R (outputs the result of the ALU’s operation requested using inputs A,B,O; IF the output S = true. |

Each person should remember its port carefully. When a request comes to the ALU (from other parts of the computer), it will refer to a specific input or output port by its name. For instance, “put 1 in A”, “read the value of S”, “read the value of R”. The ALU (i.e. the 5 persons composing it) should be able to react very quickly to requests like: “what is the value in output R?”

2. Modifications will be made to the value of the input ports by other parts of the computer. For instance, the value in input A may be changed to “0A” (hexadecimal values, naturally). Similarly, the values of the output ports S and R can be read by other parts of the computer. Each port/person is responsible to keep track of her own value. Write your current value on a piece of paper, and be ready to erase it and replace by a new value at any time.
3. You should be prepared to deal with hexadecimal values: every piece of data in a computer is handled as binary values, which we represent here as hexadecimal values. Hence, you should be prepared collectively to very quickly compute e.g. Addition ($01 + 0A = 0B$) or comparisons ($03 < 0B$), and to write the result to the result output port as a *hexadecimal value*.

You will perform only two types of operations in this computer role-play: addition (this is “arithmetic”) and comparison (this is “logic”).

4. Addition (A+B)

As soon as another part of the computer writes the value “addition” in the input “O”, the ALU should write the value “false” to output S, take the value in input A,

take the value in input B, compute $A+B$, write the result to output R, and write the value “true” to output S.

Practice this, and especially the order:

- § Imagine that another part of the computer writes $A=01$ (the person playing port input A writes this value down, and starts wondering what this hexadecimal value means in decimal numbers),
- § then $B=0B$ (the person playing port input B writes this value down and starts wondering what this hexadecimal value means in decimal number)
- § finally, another part of the computer writes $O = \text{“add”}$ (the person playing port input O notes this and informs the other person in the ALU that it should start computing)
- § The ALU writes $S = \text{false}$ (the person playing port output S writes this new value down)
- § Collectively, the 5 person in the ALU work to establish that $01+0B = 0C$.
- § The ALU writes $R = 0C$. (the person playing port output R writes this new value down)
- § Then the ALU writes $S = \text{true}$ (the person playing port output S writes this new value down)

5. Comparison ($A \leq B$)

As soon as another part of the computer writes the value “comparison” in the input “O”, the ALU should write the value “false” to output S, take the value in input A, take the value in input B, determine whether $A \leq B$, write the result to output R (true: A is less than B or equal to B, false: A is greater than B), and write the value “true” to output S.

Practice this, and especially the order:

- § Imagine that another part of the computer writes $A=01$ (the person playing port input A writes this value down, and starts wondering what this hexadecimal value means in decimal numbers),
- § then $B=0B$ (the person playing port input B writes this value down and starts wondering what this hexadecimal value means in decimal number)
- § finally, another part of the computer writes $O = \text{“compare”}$ (the person playing port input O notes this and informs the other person in the ALU that it should start computing)
- § The ALU writes $S = \text{false}$ (the person playing port output S writes this new value down)

- § Collectively, the 5 person in the ALU work to establish that $01 < 0B$.
- § The ALU writes $R = \text{true}$. (the person playing port output R writes this new value down)
- § Then the ALU writes $S = \text{true}$ (the person playing port output S writes this new value down)

6. Organize collectively so that these 2 operations (addition, comparison) can be made at maximum speed and maximum reliability.