ALU (Arithmetic Logic Unit)

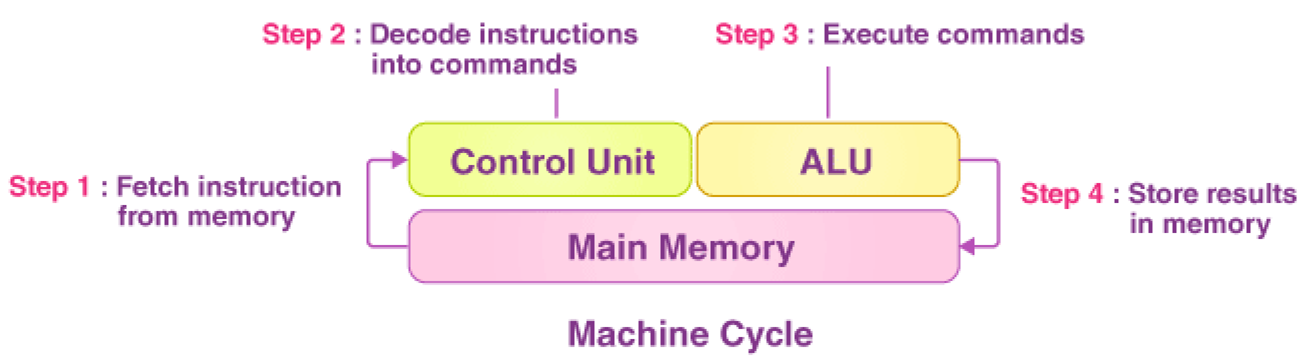
An arithmetic logic unit (ALU) is a key component of a computer's central processor unit. The ALU performs all arithmetic and logic operations that must be performed on instruction words. The ALU is split into two parts in some microprocessor architectures: the AU and the LU.

Table of Contents

What is ALU (Arithmetic Logic Unit)?

ALU conducts arithmetic and logic operations. It is a major component of the CPU in a computer system. An integer unit (IU) is just an integrated circuit within a GPU or GPU that performs the last calculations in the processor.

It can execute all arithmetic and logic operations, including Boolean comparisons, such as subtraction, addition, and shifting (XOR, OR, AND, and NOT operations). Binary numbers can also perform bitwise and mathematical operations. AU (arithmetic unit) and LU (logic unit) are two types of arithmetic logic units. The ALU's operands and code instruct it on which operations to perform based on the incoming data. When the ALU has finished processing the data, it sends the result to the computer memory.



Uses of ALU

ALUs, in addition to doing addition and subtraction calculations, also handle the process of multiplication of two integers because they are designed to perform integer calculations; thus, the result is likewise an integer. Division operations, on the other hand, are frequently not done by ALU since division operations can result in a floating-point value. Instead, division operations are normally handled by the floating-point unit (FPU), which may also execute other non-integer calculations.

Engineers can also design the ALU to do any operation they choose. However, as the operations become more sophisticated, ALU becomes more expensive since it generates more heat as well as takes up more space on the CPU. Therefore, engineers create powerful ALUs, ensuring that the CPU is both quick and powerful.

The ALU performs the computations required by the CPU; most of the operations are logical in nature. If the CPU is built more powerful, it will be designed on the basis of the ALU. Then it generates more heat and consumes more energy or power. As a result, there must be a balance between how intricate and strong ALU is and how much it costs. The primary reason why faster CPUs are more expensive is that they consume more power and generate more heat due to their ALUs. The ALU's major functions are arithmetic and logic operations, as well as bit-shifting operations.

Operations Performed by ALU

Although the ALU is a critical component of the CPU, the design and function of the ALU may vary amongst processors. Some ALUs, for example, are designed solely to conduct integer calculations, whereas others are built to perform floating-point computations. Some processors have a single arithmetic logic unit that performs operations, whereas others have many ALUs that conduct calculations. ALU's operations are as follows:

1. Arithmetic Operators: It refers to bit subtraction and addition, despite the fact that it does multiplication and division. Multiplication and division processes, on the other hand, are more expensive to do. Addition can be used in place of multiplication, while subtraction can be used in place of division.

2. Bit-Shifting Operators: It is responsible for a multiplication operation, which involves shifting the locations of a bit to the right or left by a particular number of places.

3. Logical Operations: These consist of NOR, AND, NOT, NAND, XOR, OR, and more.

ALU Signals

The ALU contains a variety of electrical input and output connections, which result in the digital signals being cast between the ALU and the external electronics. External circuits send signals to the ALU input, and the ALU sends signals to the external electronics.

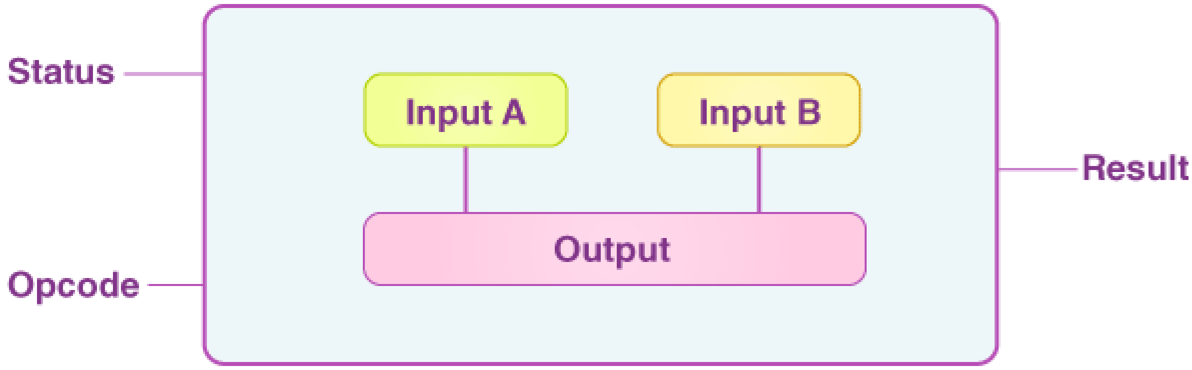
Opcode: The operation selection code specifies whether the ALU will conduct arithmetic or a logic operation when it performs the operation.

Data: The ALU contains three parallel buses, each with two input and output operands. These three buses are in charge of the same amount of signals.

Status

Input: Once ALU has completed the operation, then the status inputs allow the ALU to obtain more information needed to complete the process successfully. A single "carry-in" is used, which is a stored carry-out from the prior ALU operation.

Output: The status outputs, which are numerous signals, offer the results of an ALU operation in the form of extra data. Overflow, carry out, zero, negative, and other status signals are usually handled by general ALUs. The status output signals were stored in the external registers after the ALU completed each operation. These signals are saved in external registers, which allows them to be used in future ALU operations.



ALU Configurations

The following is a description of how the ALU interacts with the processor. These configurations are included in every arithmetic logic unit:

- Accumulator
- Instruction Set Architecture
- Stack
- Register Stack
- Register to Register
- Register Memory

Pros of ALU

The following are some of the benefits of ALU:

- It supports high-performance parallel architecture and applications.
- It can provide the desired output at the same time and combine integers and floating-point variables.
- It has the ability to carry out instructions on a large number of items and has a high level of precision.
- The ALU can combine two arithmetic operations in the same code, such as multiplication and addition or subtraction and addition, or any two operands. A+B*C is an example.
- They remain consistent throughout the presentation, and they're spaced in such a way that they do not interrupt any of the segments.
- It is, in general, highly rapid, and as a result, it produces results swiftly.
- With ALU, there are no sensitivity difficulties or memory wastage.
- They are less costly and reduce the number of logic gates required.

Cons of ALU

The following are some of ALU's drawbacks:

- Floating variables have higher delays with the ALU, and the intended controller is difficult to grasp.
- If memory space were fixed, bugs would appear in the results.

- Amateurs are tough to understand since their circuits are complex, and the principle of pipelining is also difficult to grasp.
- The inconsistencies in latencies are a known drawback of ALU.
- Another flaw is rounding off, which reduces precision.

he architecture of the Central Processing Unit (CPU) operates the capacity to function from “Instruction Set Architecture” to where it was designed. The architectural design of the CPU is Reduced instruction set computing (RISC) and Complex instruction set computing (CISC). CISC has the capacity to perform multi-step operations or addressing modes within one instruction set. It is the CPU design where one instruction works several low-level acts. For instance, memory storage, loading from memory, and an arithmetic operation.

Reduced instruction set computing is a Central Processing Unit design strategy based on the vision that a basic instruction set gives great performance when combined with a microprocessor architecture. This architecture has the capacity to perform the instructions by using some microprocessor cycles per instruction. This article discusses the RISC and CISC architecture with appropriate diagrams. The hardware part of the Intel is named as Complex Instruction Set Computer (CISC), and Apple hardware is Reduced Instruction Set Computer (RISC).

What is RISC and CISC Architectures?

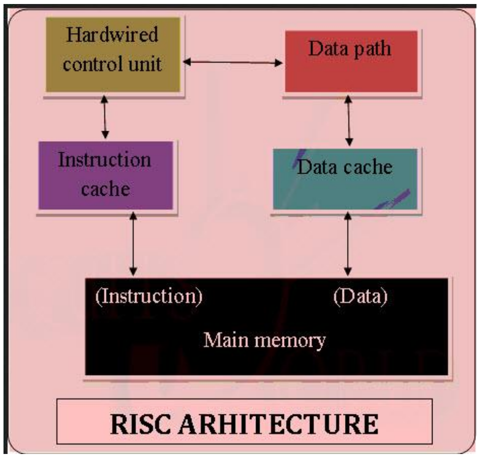
A complex instruction set computer is a computer where single instructions can perform numerous low-level operations like a load from memory, an arithmetic operation, and a memory store or are accomplished by multi-step processes or addressing modes in single instructions, as its name proposes “Complex Instruction Set ”.

A reduced instruction set computer is a computer that only uses simple commands that can be divided into several instructions which achieve low-level operation within a single CLK cycle, as its name proposes “Reduced Instruction Set ”

RISC Architecture

The term RISC stands for ‘’Reduced Instruction Set Computer’’. It is a CPU design plan based on simple orders and acts fast.

This is a small or reduced set of instructions. Here, every instruction is expected to attain very small jobs. In this machine, the instruction sets are modest and simple, which help in comprising more complex commands. Each instruction is about a similar length; these are wound together to get compound tasks done in a single operation. Most commands are completed in one machine cycle. This pipelining is a crucial technique used to speed up RISC machines.



RISC Architecture

Reduced Instruction Set Computer is a microprocessor that is designed to carry out few instructions at a similar time. Based on small commands, these chips need fewer transistors, which makes the transistors inexpensive to design and produce. The features of RISC include the following.

Please refer to this link to know more about Pipelining in Computer Architecture MCQs

- The demand for decoding is less
- Few data types in hardware
- General-purpose register Identical
- Uniform instruction set
- Simple addressing nodes

Also, while writing a program, RISC makes it easier by letting the computer programmer eliminate needless codes and stops wasting cycles.

Characteristics

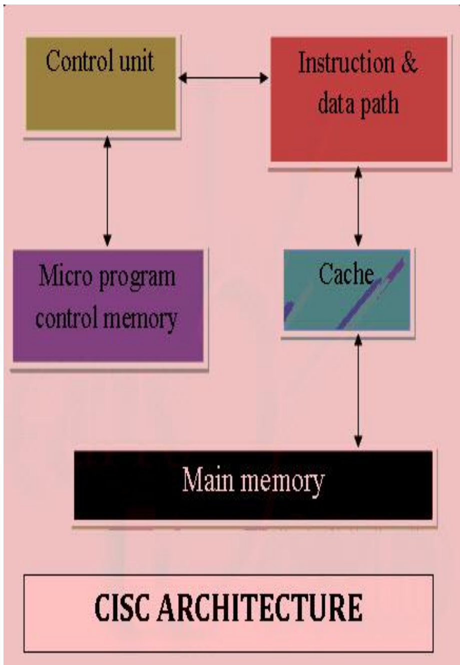
The characteristics of RISC architecture include the following.

- Simple Instructions are used in RISC architecture.
- RISC helps and supports few simple data types and synthesizes complex data types.
- RISC utilizes simple addressing modes and fixed-length instructions for pipelining.
- RISC permits any register to use in any context.
- One Cycle Execution Time
- The amount of work that a computer can perform is reduced by separating "LOAD" and "STORE" instructions.
- RISC contains a Large Number of Registers to prevent various interactions with memory.
- In RISC, Pipelining is easy as the execution of all instructions will be done in a uniform interval of time i.e. one click.
- In RISC, more RAM is required to store assembly-level instructions.
- Reduced instructions need a less number of transistors in RISC.
- RISC uses the Harvard memory model means it is Harvard Architecture.
- A compiler is used to perform the conversion operation means converting a high-level language statement into the code of its form.

CISC Architecture

The term CISC stands for ''Complex Instruction Set Computer''. It is a CPU design plan based on single commands, which are skilled in executing multi-step operations.

CISC computers have small programs. It has a huge number of compound instructions, which take a long time to perform. Here, a single set of instructions is protected in several steps; each instruction set has an additional than 300 separate instructions. Maximum instructions are finished in two to ten machine cycles. In CISC, instruction pipelining is not easily implemented.



CISC Architecture

The CISC machines have good acts, based on the overview of program compilers; as the range of innovative instructions are simply obtainable in one instruction set. They design compound instructions in a single, simple set of instructions.

They achieve low-level processes, which makes it easier to have huge addressing nodes and additional data types in the hardware of a machine. But, CISC is considered less efficient than RISC, because of its incompetence to eliminate codes which leads to wasting of cycles. Also, microprocessor chips are difficult to understand and program for, because of the complexity of the hardware.

- Instruction Set Architecture is a medium to permit communication between the programmer and the hardware. Data execution part, copying of data, deleting, or editing is the user commands used in the microprocessor, and with this microprocessor, the Instruction set architecture is operated.
- The main keywords used in the above Instruction Set Architecture are as below

Instruction Set: Group of instructions given to execute the program and they direct the computer by manipulating the data. Instructions are in the form – Opcode (operational code) and Operand. Where, the opcode is the instruction applied to load and store data, etc. The operand is a memory register where instruction is applied.