processor Architecture types

By Aya Ahmed Shareef

---

1. Introduction

What is architectures?

When dealing with any computer system, we note that it is based on a set of functional modules such as processor, memory, input and output gateways, and switching networks. We can think of Mother's PC motherboard as an excellent example of computer systems. It includes microprocessor, RAM, ROMs, and input / output gateways. The process of putting these components together, and connecting them to each other will not get the required functions of the motherboard, which is to drive the computer as a

whole, to show what happens on the screen and to receive various commands from the user, via input terminals such as keyboard and mouse . There must be a certain methodology that governs the process of the computer. The processor should know what the mouse click process means, or even move the mouse pointer. The computer should also explain and understand the buttons used by the user. He should also interpret these messages. , And show them on the screen. At the same time, the computer must regulate the mechanism by which information is exchanged between the memory and the processor. All of these tasks are primarily related to the concept of "architecture." **So What is architecture?**

Architecture is the mechanism by which the various functional units are arranged within the computer system. The architecture determines how the data and information are transferred between the computer units, such as the central processing unit, the memory and the input and output ports.

There is an important point regarding the concept of architecture:

architecture relates not only to the organization and design of computer hardware, but also to software, applications and operating systems. When we say that a processor is built on an architecture, we mean the mechanism according to which the design and organization of the internal units of this processor. When we say that an operating system is designed with 64-bit architecture, we mean designing and organizing instruction sets and how they work within the operating system. In other words, we must distinguish between the "hardware" and "software" architecture

3. Processor Arch. Types

Modern Architectures

We can say that the previous simple presentation is a good display and is suitable for understanding the architectural principle, but can not be said that it is comprehensive, especially with the modern architectures that companies are putting up. If we want to research and dive into the architectures that Intel is launching and developing, we need thousands of pages, illustrations and illustrations to say that we have presented the idea fully, not to mention other companies that are also developing their processors according to their architectures, such as AMD.

In addition, we have presented the concept of architecture from the point of view of computer systems completely: this means that the preceding speech applies to any computer system, whether a personal computer, laptop, tablet, smartphone, or systems-based computer systems (Such as digital signal processors, DSP, and microcontrollers). We must take into account recent advances in microprocessors, such as multi-core processors (no longer a modern computer or intelligent device but a multi-core processing unit), and other features such as Hyper-Threading The processor operates twice as much as the processing nuclei (ie, two-core processors will work as if they have four processing nuclei), and other technologies and features that companies offer to improve the performance of their processors and the computer systems they will work on. All these details must be taken into

account in the study of any processor and computer system, and it is not possible to rely solely on primary information and simple concepts relating to the subject of architecture and the structure of the computer system.

In this sense, because of the increasing complexity and complexity of computer systems, we must distinguish between the following **types of computer architectures:**

- (Eg micro-computers), such as Intel and AMD architectures, and characterize these architectures in personal computers and laptop
- Architecture of embedded systems: such as ARM architecture or Atmel's AVR architectures. These architectures are highly differentiated in smartphones and smartphones
- Architecture of computer systems operating in servers Supercomputers

There are two main types of architecture designs built according to the **CISC** and **RISC** instruction set. In this article, we will review each other and make a simple comparison to this article.

CISC Architecture

The CISC architecture is a short description of the "Complex Instruction Set Computer". It is found in microprocessors that contain a wide range of complex instructions. The primary purpose of this architecture is to terminate the task with fewer instructions and lines in assembly language Assembly Language. This is done by designing devices capable of understanding and applying a series of operations by reading a single instruction. Let's say, for example, a symbol with MULT. Suppose that this instruction works on multiplying two values.

MULT 0x100, 0x101

When this instruction is applied, two values will be loaded on different registrars, where the values are stored in addresses 0x100 and 0x101 in the data memory, and then their operators will multiply their values. As a last step, the result is kept in an appropriate record. We have achieved a multiplication process with only one "complex" instruction.

One of the main advantages of using this system is to reduce the work on compiler Compiler because of the lack of number of lines in the assembly language of any code because of the comprehensiveness and complexity of a single instruction. Also, because of the small program, you need a small

memory to save the instructions. But what this architecture needs is complex instructions and hardware devices that it can understand.

The CISC methodology was the only computation that existed until 1975 when IBM experimented with a new idea quite different from the CISC, which would later be called the RISC architecture. In fact, the IBM project was not the only one at the time when there were almost two similar projects at both Stanford University and Berkeley University in the United States but credited as the first RISC machine for IBM 801.

Architecture of RISC

RISC is the abbreviation for "Reduced Instruction Set Computer". Unlike computers based on CISC architecture, computers based on RISC architecture have a limited number of instructions, meaning that the program will need a number of simple instructions to implement its mission. To take our earlier example, the task of the computer is to strike two values but the previous complex ramifications are not available in this type of architecture. We will then need several simple instructions to reach the desired result. First, load to load the recorders with the required values, a mul instruction to apply the multiplication of the parameters and a mov instruction to transfer the result to the desired recorder. The previous program will be as follows:

```
load RegA, 0x100
load RegB, 0x101
 mul RegA, RegB
mov RegA, RegC
```

Where the mul instruction multiplies the two values and stores the output in the RegA register, which corresponds to the following instruction in C: RegA = RegA * RegB. The mov instruction then returns the result stored in RegA to the desired registrar in this case, the RegC register.

As a first glance, you may think that this approach is not efficient enough to handle tasks. Because of the simplicity of the instructions, we will need a large number of them to implement complex tasks. This is why there is a need for more software memory than CISC to save multiple instructions. The

translator's role will also be greater for converting programs into assembly language because the number of lines per code increases. But because of the simplicity of the instructions, the program will be implemented faster, as each instruction needs only one cycle of the "1 clock cycle" processor clock for the application. Also, because of that simplicity, hardware design will be much simpler. The processor will need fewer transistors, leaving more room for developing other key things, such as increasing the number of recorders and the size of the memory.

So far there is debate about the identity of the best architect. There is a clear preference for the RISC approach by most manufacturers for two main reasons, its high speed and low cost (because of the low number of transistors as we mentioned).

3. Conclusion:

If we want to summarize our previous presentation in the form of a simple table comparing the two architectures, it would be as follows:

CISC	RISC
1. Stresses the work of hardware	1. Stresses work on the program
2. Multi-frequency "Multi Clock"	2. Stresses work on the program
3. Only complicated instructions	3. Simple instructions only
4. A large number of cycles per second The size of the program is small	4. A small number of cycles per second The size of the program is great
5. Employ more transistors to store complex instructions	5. Employ more transistors on memory recorders

4. References

Sources: "High Performance Computer Architecture" by Harold S. Stone

Harold S. Stone. "The Book" Computer Organization & Architecture "by William Stallings