TYPES OF PROCESSORS AND INSTRUCTION EXECUTION (OPERATE C 1.3)

The **Instruction Set Architecture (ISA)** is the part of the computer processor that is visible to the user. In fact, it is necessary to understand this architecture in order to write assembly language. It is a set of basic instructions that the processor can understand and can be considered as a boundary between software and hardware. These basic instructions are simple arithmetic or logic operations that the processor can understand.

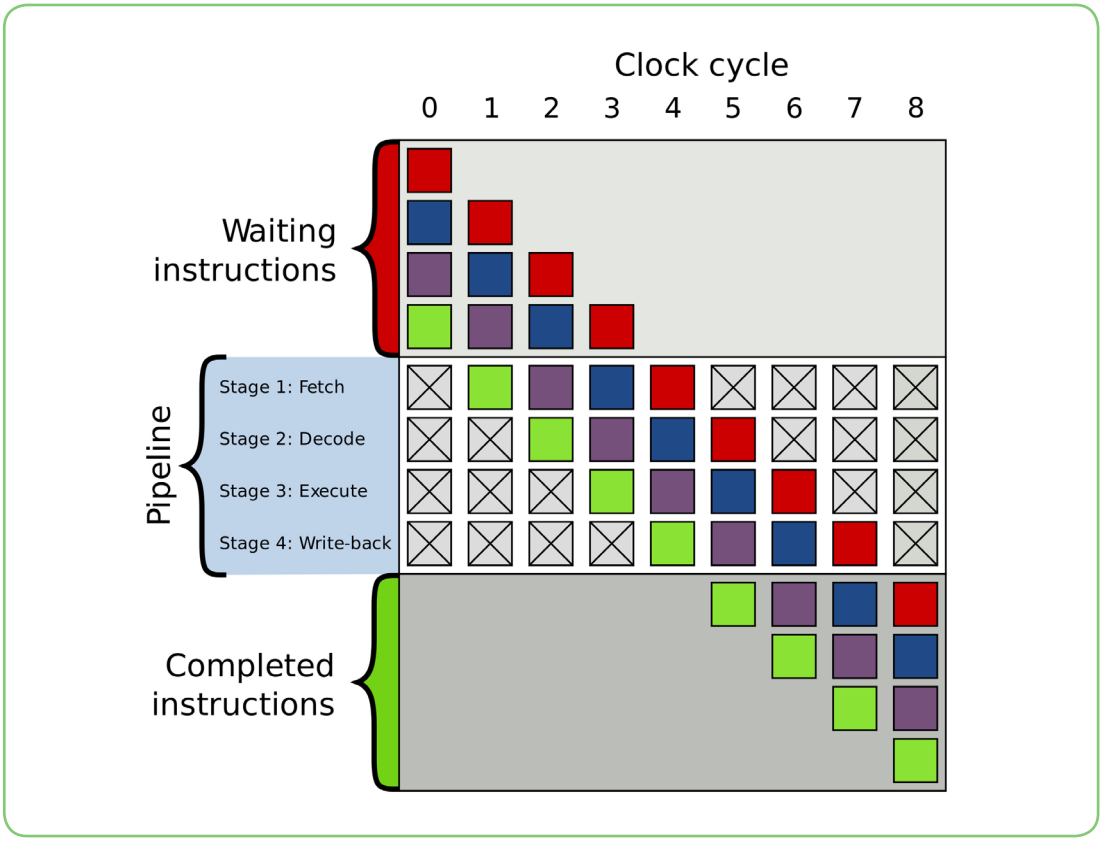
Here are some examples:

- Arithmetic: add, subtract
- Logic: AND, OR, NOT
- Data transfer instructions: move, input, output, load, store
- Control transfer instructions: go to, if ... go to, call, return.

There are two main types of ISA:

1. **Reduced Instruction Set Computer (RISC)**
2. **Complex Instruction Set Computer (CISC).**

RISC processors are used for a smaller number of computer instructions at a high speed, i.e. millions of instructions per second (MIPS).
CISC processors have a full set of computer instructions. Each instruction can execute several lower level operations in a single instruction.



Instruction Pipelining is a technique to increase the instruction throughput, i.e. the number of instructions per unit of time. To operate at full performance, the pipeline will need to run four subsequent independent instructions while the first is completing. Therefore, the main advantage of pipelining is a reduced cycle time as it permits a new instruction every clock cycle because all the components work in parallel. Unfortunately, programs can sometimes run incorrectly because of pipeline, although non-pipeline architectures are inefficient because some CPU components are idle.

Non-pipeline processors, on the other hand, are used to execute one single instruction at a time for simple projects because the instruction latency, i.e. the time needed for feedback, is lower.

Instruction level parallelism (ILP) measures how many operations in a program can be performed simultaneously. ILP also allows the order in which the operations are executed to be changed and is used in graphics and scientific computing.

However, as in pipeline techniques the instructions are given in order, if one is stalled, i.e. blocked, no later instruction can proceed. For this reason, **dynamic scheduling** has been devised. In fact, with dynamic scheduling, the hardware rearranges execution of instructions to reduce stalls. It simplifies the compiling process, i.e. code translation, and allows the code compiled for a pipeline to run efficiently with another.

1 **After studying the page on the left, cover it and try this test. You have to choose the right answer for each question.**

1. ISA stands for...
- **a.** Instruction System Architecture.
- **b.** Instruction Set Architecture.
- **c.** Instruction Set Archive.
- **d.** Instruction System Archive.

2. Move, input, output, load, are examples of...
- **a.** arithmetic instructions.
- **b.** logic instructions.
- **c.** data transfer instructions.
- **d.** control transfer instruction.

3. A RISC processor...
- **a.** manages complex instructions.
- **b.** works at high speed.
- **c.** is slow.
- **d.** executes multiple instructions at a time.

4. A CISC processor...
- **a.** is similar to a RISC processor.
- **b.** executes one instruction at a time.
- **c.** executes complex instructions.
- **d.** works at high speed.

5. Instruction pipelining is...
- **a.** a technique to increase the total number of instructions.
- **b.** a technique to increase the number of instructions per unit of time.
- **c.** a system to measure the quantity of instructions performed simultaneously.
- **d.** a technique to execute one instruction at a time.

6. Non-pipeline architectures....
- **a.** are more efficient.
- **b.** are less efficient.
- **c.** execute instructions in parallel.
- **d.** use dynamic scheduling.

7. ILP...
- **a.** means Instruction Level Parallelism.
- **b.** measures the total amount of instructions executed by a processor in its life span.
- **c.** executes the instructions in order.
- **d.** is used in business systems.

8. Dynamic scheduling...
- **a.** is a type of instruction pipelining.
- **b.** does not guarantee code portability.
- **c.** complicates the compiling process.
- **d.** solves a typical problem of instruction pipelining.