# Other processor units

- **Bus Interface Unit:**
  - Manages the communication between the CPU and the system's buses.
- **Segmentation and Paging Units:**
  - Segmentation involves dividing memory into segments
  - Paging involves dividing it into fixed-size pages.
- **Decoding Unit:**
  - It decodes the binary instructions into actionable commands for the CPU.
- **Anticipation Unit (Queue):**
  - It predict and anticipate the next set of instructions, enabling efficient pre-processing and preparation for upcoming tasks.

# Processor registers

## General registers

- Fast memories inside the processor (Handle data at high speed).
- The address of a register is associated with its name.
- Save intermediate results (avoids memory access).
- Are available to the programmer ( Load, Store, Transfer, and Increment)

# Processor registers

- **Address registers**
  - **Program counter (PC):** Keeps track of the memory address of the next instruction to be executed.
  - **Stack pointer register (Stack Pointer):** Manages the memory stack, keeping track of the top of the stack.
  - **Index registers (SI/DI):** Used for indexing operations in memory.
  - **Address register:** Holds a specific memory address

# Processor registers

- **Instruction register:** Stores the current instruction being executed.

- **Memory word register or data register:** Holds data fetched from or to be written to memory.

- **Accumulator register:** Holds the results of arithmetic and logic operations.

- **Status register (Program Status Word, PSW):** Contains flags and status information about the processor state.

# Steps for executing an instruction

- Finding the instruction to process

- Decoding the instruction and finding the operand

- Instruction execution

# Processor performance

- One can characterize the power of a processor by the number of instructions it can process per second.

- The CPI (Cycles Per Instruction) represents the average number of clock cycles needed for the execution of one instruction.

- The MIPS (Millions of Instructions Per Second) represents the processing power of the microprocessor.

# Processor performance

- **Example:**

Consider a machine with a clock frequency of 700MHz. It consists of 5 instruction classes with the following CPIs:

| | |
|---|---|
| Class A | CPI =1 |
| Class B | CPI =2 |
| Class C | CPI =3 |
| Class D | CPI =5 |
| Class E | CPI =7 |

| Compilers | Classes (%) | | | | |
|---|---|---|---|---|---|
| | A | B | C | D | E |
| Compiler 1 | 10 | 20 | 10 | 30 | 30 |
| Compiler 2 | 20 | 15 | 15 | 30 | 20 |
| Compiler 3 | 5 | 10 | 15 | 40 | 30 |

- What is the CPI of the code generated by each of the compilers?
- Which compiler produces code with the highest MIPS?
- Which compiler produces the most efficient code?

# Processor performance

- To enhance the performance of a processor:
  - Increase the clock frequency (hardware limitation)
  - Decrease the CPI (selection of a suitable instruction set).
- The collective improvements in processors aim to reduce the program's execution time.

# Processor performance

**Parallelism:**

- ➢Parallelism involves simultaneously executing instructions related to the same program on different processors.

- ➢This translates to breaking down a program into multiple processes treated in parallel to gain execution time efficiency.

# Processor performance

**Pipeline architecture:**

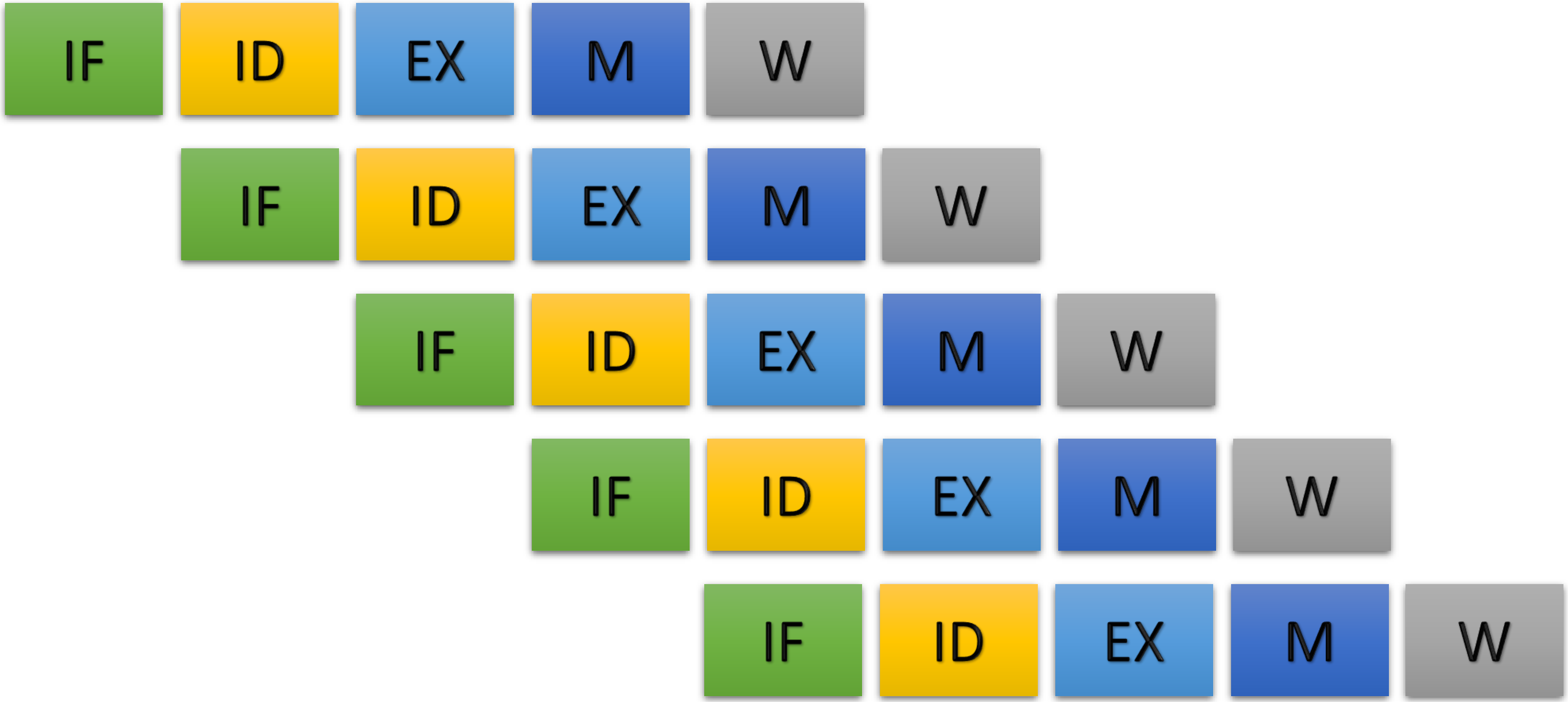
- ➢it is a design approach that divides the execution of instructions into a series of stages.
- ➢The pipeline allows different stages of multiple instructions to be processed simultaneously.

**Example**

- IF: Instruction Fetch from the cache;
- ID: Instruction Decode and Operand Fetch;
- EX: Instruction Execution;
- M: Load or Write to Memory;
- W: Write the calculated value to the registers.

# Processor performance

## **Pipeline architecture**



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| IF | ID | EX | M | W | | | | |
| | IF | ID | EX | M | W | | | |
| | | IF | ID | EX | M | W | | |
| | | | IF | ID | EX | M | W | |
| | | | | IF | ID | EX | M | W |

# Processor performance

## **Superscalar architecture:**

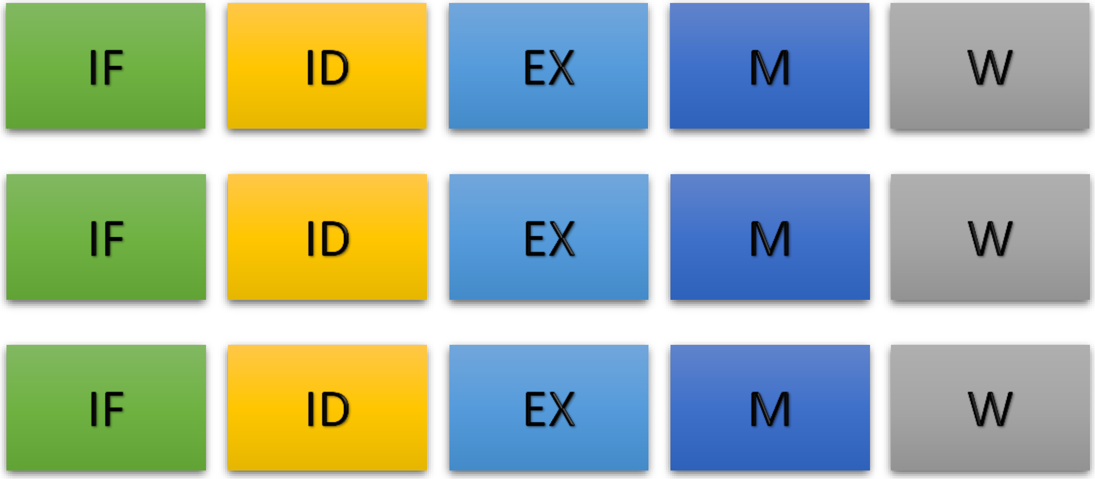
- Another way to enhance performance is to execute multiple instructions simultaneously.

- The superscalar approach involves equipping the processor with multiple processing units working in parallel.

# Processor performance

## **Pipeline and superscalar architecture:**

- The principle is to execute instructions in a pipelined manner within each of the processing units working in parallel.



IF ID EX M W

IF ID EX M W

IF ID EX M W

# Processor performance

## **HyperThreading:**

- HyperThreading technology involves defining two logical processors within a physical processor.

- The system recognizes two physical processors and behaves as a multitasking system by simultaneously processing two threads.

# Processor performance

## **Multi-core:**

A multi-core processor is simply a processor composed not of 1, but of 2 (or 4 or 8) computing units.