Intro to assembler programming

- **Machine Instructions**

  - **Machine instruction:**

    - **Machine instruction** = a **instruction** that **directs** the **computer** to perform a ***single* computer operation**

      - **Each machine instructions** is **represented** by a **specific *binary* pattern**

        See: **instruction encoding** click here

  - ***Single* machine operation**:

    - A ***single* machine operation** will perform a **very *simple* operation**

    - ***Typical* machine operations** are:

      - **Copy** the **value** stored in a **memory location** to ***another* memory location**
      - **Add** two **values** and **store** the **sum** in the **destination**
      - And so on....

- **Assembler instructions**

  - **Assembler instruction:**

    - **Assembler instruction** = a **mnemonic code** to **represent** a **machine instruction**:

      - **Each machine instruction** corresponds to ***one* assembler instruction**

  - **Examples** of **assembler instructions**:

```
move   x, y      ; copy data stored in memory location x to mem loc y

add   #1, y      ; y = y + 1
```

- **Assembler programming:**

    - The **programmer** writes his/her **program** using **assembler instructions** using a **text editor (`gedit`)**

        - **Assembler program** = **program** written in **assembler instructions** (or **assembler *code***)

    - A **computer program** (called an **assembler**) is used to **translate** the the **assembler program** into (executable) **machine instructions**

- **Assembler programming and CPU structure**

    - **Fact:**

        - **Assembler instructions must** use ***resources*** inside the **CPU** (Central Processing Unit)

    Therefore:

    - We will **briefly** review the **structure** of a **CPU**

    - Then we will **study** the **structure** of the **M68000 CPU** in more **details** in order to:

        - **program** the **M68000 (CPU)** using the **M68000 assembler instructions**