Computer Memory

- The memory of a computer is where the program and data are stored before the calculations begin.
- **The memory** is equivalent to thousands of registers, each storing a binary word.
- **Registers:** can be made from basic logic gates to remember, or store, a binary value.
- **Registers** are fast stand-alone storage locations that hold data temporarily. **The main types of registers are :**

  ✓ Data Registers (R1, …R3) hold intermediate results to speed up operations.

  ✓ Instruction register (I) holds one fetched instruction from memory to be executed after interpretation.

  ✓ Program Counter (PC) register points (تؤشر) to the address of the next instruction in memory to be executed. See figure (2.1).

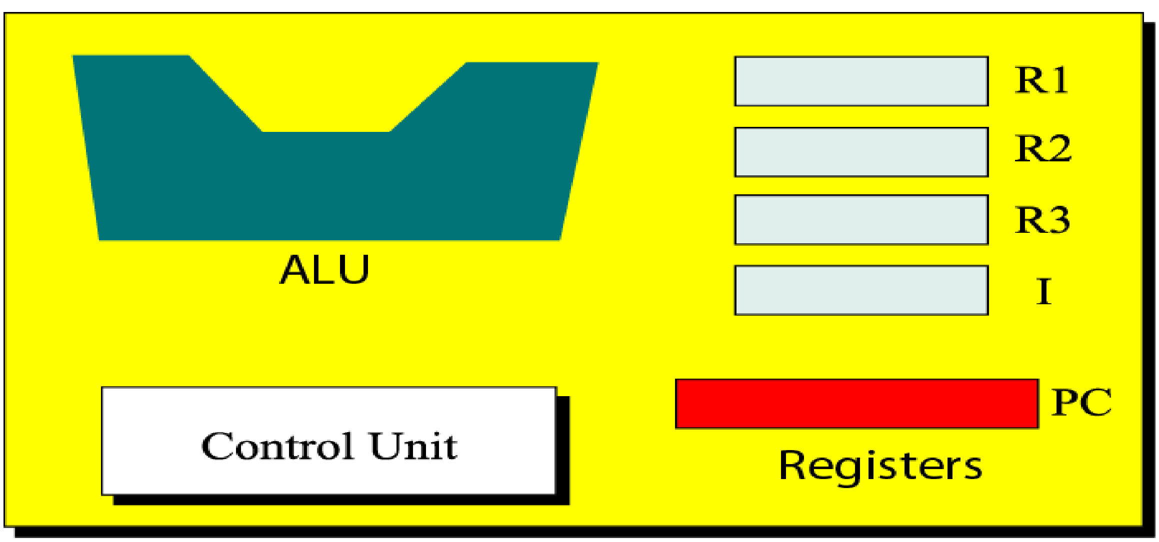


Figure 2.1: CPU Structure

- **Main memory** is a collection of storage location, each with a unique identifier called **address**.
- **A program** (data & Instructions) should be stored in memory in order to get executed.
- The total number of uniquely identifiable locations in memory is called **address space**.

Memory units

Unit	*Number of bytes*
Kilobyte	**2^{10} bytes**
Megabyte	**2^{20} bytes**
Gigabyte	**2^{30} bytes**
Terabyte	**2^{40} bytes**
Petabyte	**2^{50} bytes**

Memory locations and addresses::: determine how the computer's memory is organized so that the user can efficiently store or retrieve information from the computer.

- The computer's memory is made of a silicon chip which has millions of storage cell, where each storage cell is capable to store a bit of information which value is either 0 or 1.

- But the fact is, computer memory holds instructions and data, and a single bit is very small to hold this information so bits are rarely (نادرا) used individually. As a solution to this, the bits are grouped in fixed sizes of **n bits**.

- Each byte is given a unique numeric address, which represents its location just as a street address represents the location of a house,
- The group of **n bit** is termed as word where n is termed as the word length. General-purpose computers nowadays have 32 to 64 bits. See figure (2.2).

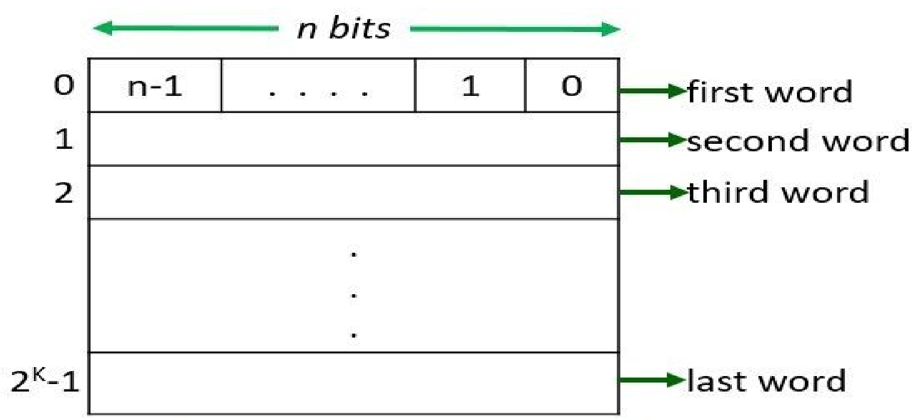


Figure 2.2: Memory Words

- Data are written into memory and readout of memory, based on their address.
- Now, whenever you want to store any instruction or data may it be of a byte or a word you have to access a memory location.

NOT

The memory locations are addressed from 0 to 2^K-1, i.e. a memory has 2^K addressable locations. Thus the address space of the computer has 2^K addresses.

(Where K represents the number of address lines available on the chip.)

- The 8086 microprocessor has 20 address lines (pins) coming out of it.
- Each of these address lines **can carry a single binary digit (a 0 or a 1**
- Figure (2.3) diagrams a section of memory. Note that the first word of data has a decimal value of (01111001= 121 decimal) and an address of 0000.

- Because computers operate by storing numbers as bit pattern, the address is also represented as bit pattern
- Address starts from 0 to last addressable word in address space.
- To address 64 kB (2^{16}) of memory, you need to use 16 bit for addressing.

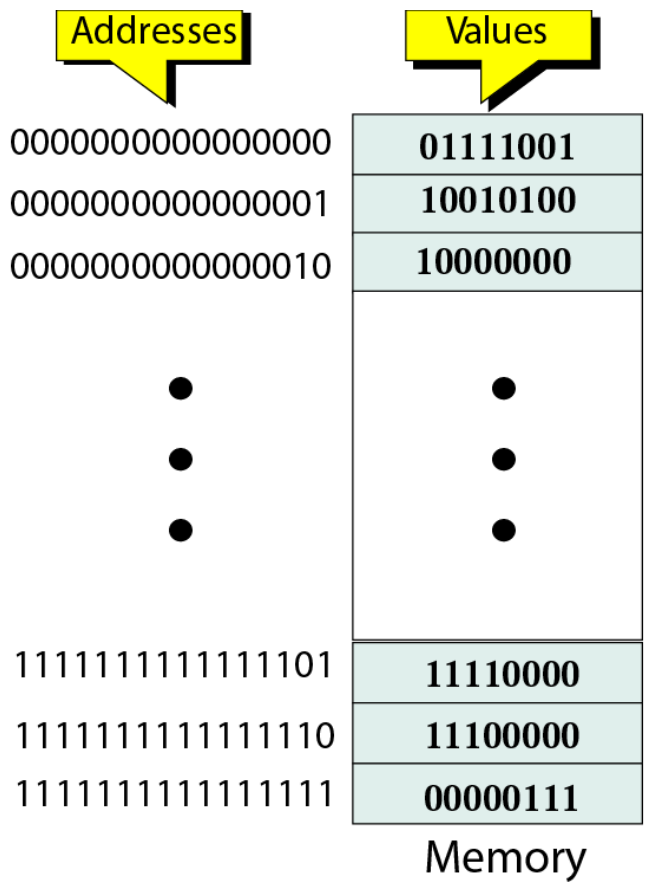


Figure 2.3:A section of memory

Memory addressing Size (حجم عنونة الذاكرة)

The size of the address bus indicates the maximum addressable number of bytes. Table (2.1) shows the size of addressable memory for a given address bus size. For example:

- ✓ **A 1-bit address bus can address up to two locations (that is 0 and 1).**
- ✓ **A 2-bit address bus can address 2^2 or 4 locations (that is 00, 01, 10 and 11).**
- ✓ **A 20-bit address bus can address up to 2^{20} addresses =(1 MB).**
- ✓ **A 24-bit address bus can address up to 16 MB.**
- ✓ **A 32-bit address bus can address up to 4 GB.**

Table 2. 1. Addressable memory (in bytes) related to address bus size

Address bus size	*Addressable memory (bytes)*
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1K
14	16K
15	32K
16	64K
17	128K
18	256K
19	512K
20	1M
21	2M
22	4M
23	8M
24	16M
25	32M
26	64M
32	4G
64	16G

Example: How many address inputs are required to access 256 bytes memory?

Solution:

Here , the memory size is 256 bytes

$2^K = 256 \rightarrow 2^K = 2^8$ ---> K=8 bits

Example: How many address lines are required to access 16 KB of memory size?

Solution:

Here , the memory size is 16 KB

$2^K = 16$ KB $\rightarrow 2^K = 16 \times 1$ KB ---> $2^K = 2^4 \times 1$ KB

$2^K = 2^4 \times 2^{10}$

So K= 14 bits

H.W: the number of bits needed to address 4KB are -------

Solution:

Address Space

What is address space with example?

A computer's address space is the total amount of memory that can be addressed by the computer. The term may refer to the physical memory (RAM chips).

- It means the 8086 can generate addresses from **00000H to FFFFFH** (H : hexadecimal), **which is a total of 1,048,576 different addresses, allowing it to access up to 1 MB of memory**.
- A 32-bit computer can address 4GB of physical memory.

Memory Space (Explain)

An address space is a range of valid addresses in memory that are available for a program or process. That is, it is the memory that a program or process can access. The memory can be used for executing instructions and storing.

Addressable Memory (الذاكرة القابلة للعنونة)

- Computers usually have two kinds of addressable memory:
  The first **is random access memory (RAM),** which allows the computer to read and write data at any of its addresses (it is also called **read/write memory or RWM).**

- Alt data in this type of memory are lost when the power is turned off and is called volatile memory (ذاكرة غير مستقرة أو متطايرة).

- The second type of memory is **read-only memory (ROM),** which is similar to RAM except that new data cannot be written in; all data in ROM are loaded at the factory and cannot be changed by the computer.

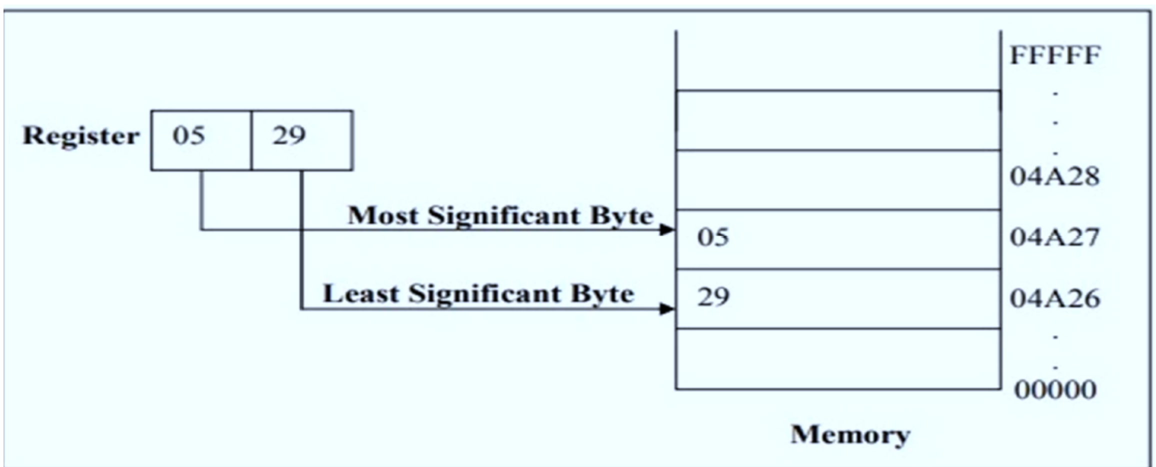
- This memory does not lose its data when power is turned off and is called non- volatile memory,

- Most microprocessor systems have both RAM and ROM. RAM is used for temporary program storage

- ROM is used to store programs and data that need to be always available.

- Depending on the model, the processor can access one or more bytes of memory at a time.
- Consider the Hexa value **(0529H)** which requires two bytes or one word of memory. It consist of high order **(most significant)** byte **05** and a low order **(least significant)** byte **29**.

- The processor store the data in memory in reverse byte sequence i.e. the low order byte in the low memory address and the high order byte in the high memory address. For example, the processor transfer the value **0529H** from a register into memory address **04A26 H** and **04A27H** as shown below:



- The processor expects numeric data in memory to be in reverse byte sequence and processes the data accordingly, again reverses the bytes, restoring them to correctly in the register as **hexa 0529H**. When programming in assembly language, you have to distinguish between the address of a memory location and its contents. In the above example the content of address **04A26H** is **29**, and the content of address **04A27H** is = **05**.