Chapter Two : Addressing Data Memory

- When programming in assembly language, you have to distinguish between the address of a memory location and its contents.
- **There are two types of addressing schemes:**
  1. **An Absolute Address,** such as **$04A26_H$,** is a 20 bit value that directly references a specific location.
  2. **A Segment Offset Address,** combines the starting address of a segment with an offset value.

Segments and Addressing

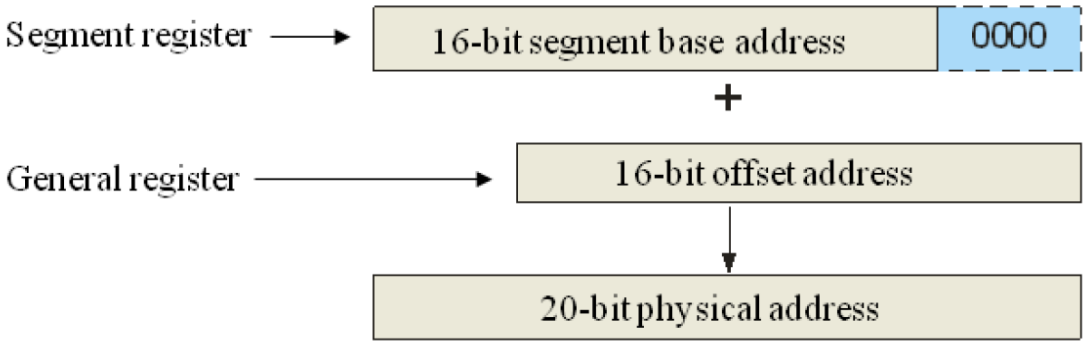
- Segments are special area defined in a program for containing the code, the data, and the stack. It includes up to **64K bytes.**

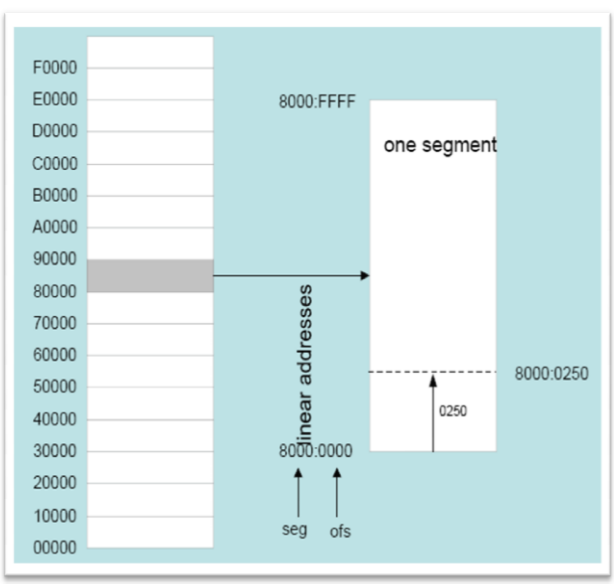
- **Segment Offset** within **a program**, all memory locations within a segment are relative to the segment starting address. The distance in bytes from the segment address to another location within the segment is expressed as an **offset** (or displacement).

- To reference any memory location in a segment, **the processor combine the segment address in a segment register with the offset value of that location, that is, its distance in byte from the start of the segment.**
- The offset address is always added to the segment starting address to locate the data. All real mode memory addresses must consist of a segment address plus an offset address.

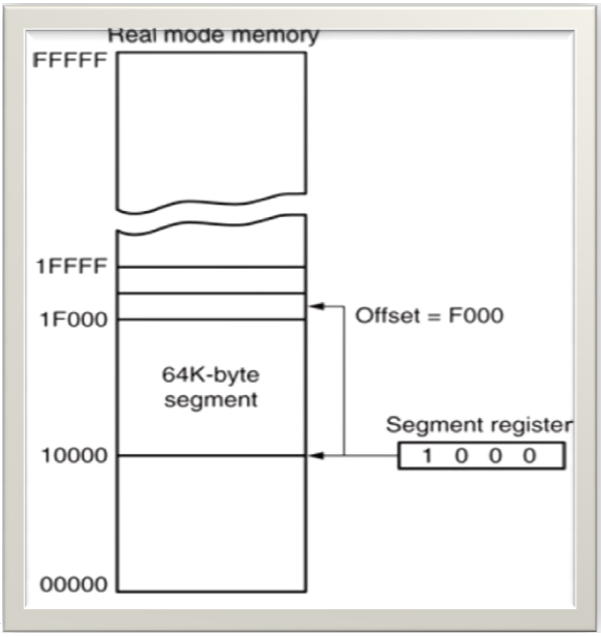
- **Segment address defines the beginning address of any 64K-byte memory segment**

- **Offset address selects any location within the 64K byte memory segment.**





Linear and Segmented Memory



- The memory of 8086 microprocessor is divided into sixteen parts or segments.
- In 8086, memory has four different types of segments. These are:
  - **Code segment,**
  - **Data segment,**
  - **Stack segment and**
  - **Extra segment.**

Each of these segments are addressed by an address stored in corresponding segment register.

16-bit

- Each **register stores the base address** (starting address) of the corresponding segment.

Because the segment registers cannot store 20 bits, they only store the upper 16 bits

- How is a 20-bit address obtained if there are only 16- bit Registers ?
- The 20-bit address of a byte is called its **physical Address**.
- But, it is specified as a **Logical Address.**

Specifying Addresses

- To reference any memory location in a segment, the address in a **segment register** with the **offset** byte from the start of the segment.
- **Offset** is the displacement of the memory location from the starting location of the segment.
- To represent a segment address and its relative offset we use the

Segment: offset

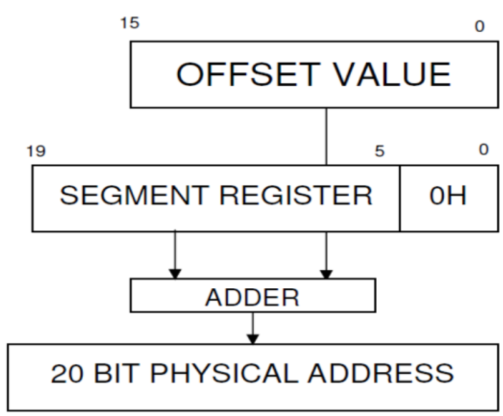
For example: 020A:1BCD denotes offset $1BCD_H$ from segment $020A_H$.

- The actual address it refers to is obtained in the following way:
  1. Add zero to the right hand side of the segment address.
  2. Add to this the offset.

  Hence the actual address referred to by **020A:1BCD** ⟶ **$03C6D_H$**

```
  1  ──────────→ carry
  020A0
+  1BCD
─────────
  03C6DH
```

To get total physical address, put the lower nibble 0H to segment address and add offset address. The figure shows the formation of 20-bit physical address.

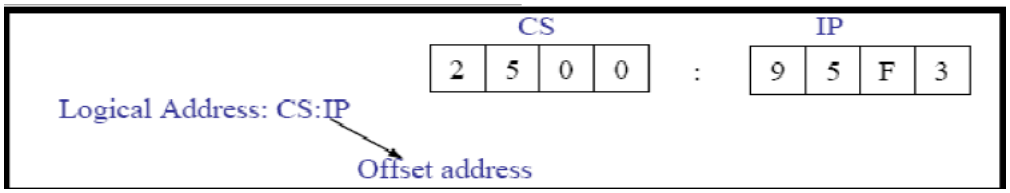


Logical and Physical Address

- Physical Address is the 20-bit address that actually put on the address bus. In 8086 has a range of $00000_H - FFFFF_H$.
- Offset Address is a location within 64K byte segment range. Has a range **$0000_H - FFFF_H$**.
- Logical Address consists of segment address and offset address in the form: **Base Address : Offset**

Addressing in Code Segment

- The logical address of an instruction consists of CS (Code Segment) and IP (instruction pointer).



Ex: If **CS:IP** given as $\rightarrow$ **2500_H:$95F3_H$, then calculate the physical address**

Ans.

1. Start with CS 2500
2. Shift left CS 25000
3. Add IP to CS 25000
   95F3 +

   $2E5F3_H$

Ex: If $CS=24F6_H$ and $IP=634A_H$, determine:

a) The logical address

b) The offset address
c) The physical address
d) The lower range of the code segment
e) The upper range of the code segment

Ans.

a) The logical address is ⟶ 24F6:634A
b) The offset address is ⟶ 634A

c) The Physical address is;

$$\begin{array}{r} 24F60 \\ +\quad 634A \\ \hline 2B2AA_H \end{array}$$

d) The lower range of the code segment: 24F6:0000 → 24F60+0000
= 24F60

e) The upper range of the code segment: 24F6:FFFF →
24F60+FFFF
= 34F5F

Addressing in Data Segment

- The area of memory allocated strictly for data is called data segment.
- The data segment uses DS and BX, SI and DI are used to hold the offset address.

Ex: If DS=$7FA2_H$ and the offset is $438E_H$, determine:

a) The physical address
b) The lower range of the data segment
c) The upper range of the data segment
d) Show the logical address

Ans:

a) The Physical address is;

$$\begin{array}{r} 7FA20 \\ +\quad 438E \\ \hline \mathbf{83DAE_H} \end{array}$$

b) The lower range: (7FA20+0000) ⟶ 7FA20
c) The upper range: (7FA20+FFFF) ⟶ 8FA1F

d) The logical address is ⟶ 7FA2:438E

Addressing in Data Segment

- Calculating the physical address for the stack, the same principle is applied as was used for the code and data segments.
- Physical address depends on the value of stack segment (SS) register and the stack pointer (SP).

Ex: **If SS=3500_H and SP:$FFFE_H$, then**

a) Calculate the physical address:

```
  35000
+ FFFE
-------
  44FFE
```

b) Calculate the lower range of the stack:

```
  35000
+ 0000
---------
  35000H
```

c) Calculate the upper range of the stack segment:

```
  35000
+ FFFF
---------
  44FFFH
```

d) Show the logical address of the stack: 3500:FFFE