Concept of Memory Segmentation and Physical address calculation

Introduction

Memory segmentation is nothing which is the methods where whole memory is divided into the smaller parts.

In 8086 microprocessor memory are divided into four parts which is known as the segments.

These segments are data segment, code segment, stack segment and extra segment.

Memory Segmentation

The total memory size is divided into segments of various sizes.

A segment is just an area in memory.

The process of dividing memory this way is called Segmentation.

Memory Segmentation

In memory, data is stored as bytes.

Each byte has a specific address.

Intel 8086 has 20 lines address bus.

With 20 address lines, the memory that can be addressed is 2 power20 bytes.

2power20= 1,048,576 bytes (1 MB).

8086 can access memory with address

It ranges from 0000H to FFFFFH

Segments.

In 8086, memory has four different types of segments.

These are:

Code Segment

Data Segment

Stack Segment

Extra Segment

Segment Register

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

These registers are of 16-bit in size.

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.

CREATION OF A 20 BIT PHYSICAL ADDRESS

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.

Logical address is in the form of:

Base Address : Offset

Offset is the displacement of the memory location from the starting location of the segment.

CREATION OF A 20 BIT PHYSICAL ADDRESS

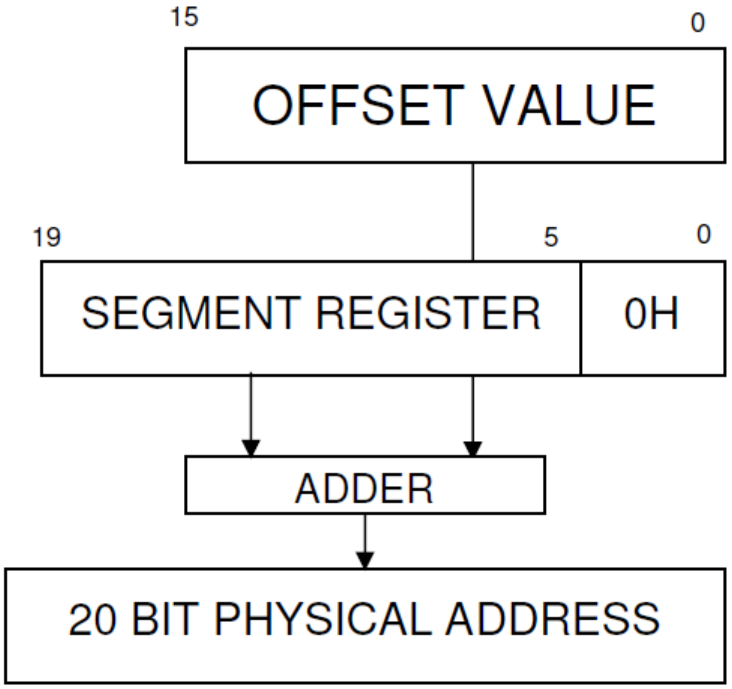
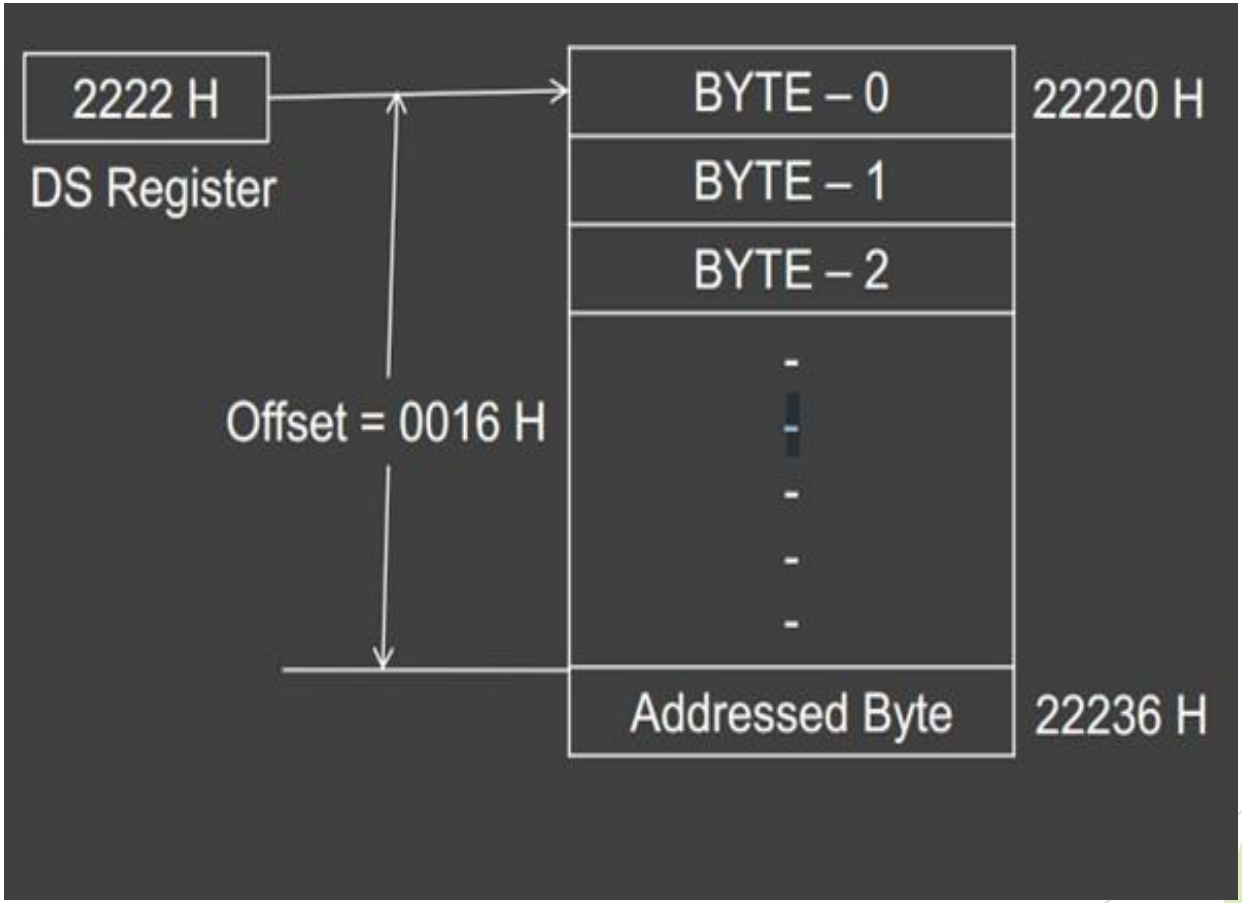


Fig. 1.3. Physical address formation


2222 H
DS Register
Offset = 0016 H
BYTE – 0
BYTE – 1
BYTE – 2
Addressed Byte
22220 H
22236 H

CREATION OF A 20 BIT PHYSICAL ADDRESS

The 8086 addresses a segmented memory. The complete physical address which is 20-bits long is generated using segment and offset registers each of the size 16-bit.

The content of a segment register also called as segment address, and content of an offset register also called as offset address.

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.

Example 1

Suppose the Data Segment holds the Base Address as 1000h and the data you need is present in the 0020h memory location (Offset) of the Data Segment. The calculation of the actual address is done as follows.

1. Left shift the 16-bit address present in the segment register by 4-bits

0001 0000 0000 0000 (0000)

2. Add the 16-bit offset address to this shifted base address

0001 0000 0000 0000 0000

+ 0000 0000 0010 0000

0001 0000 0000 0010 0000

So, the actual address turns out to be 10020h.

At any point of time we can change the base address of the segment registers and use the memory locations in those segments using the offset.

Example 2

The segment input will be from one of the segment registers: CS, DS, SS, ES. The other input will be from a base, index, or control register, or will be an immediate value. To generate the 20-bit address, the CPU shifts the segment 4 bits (1 hex digit) to the left and adds the offset to it.

A common combination for segment and offset is the CS and IP registers: together they indicate the address of the next instruction.

As an example, suppose the CS register contains ABCD **H** and the IP register contains 0046h.

- Shift the CS register to the left by 1 hex digit: ABCD0h
- Add the IP register to the new CS value: ABCD0h + 0046h
- The absolute address is ABD16h.

Example 3

The value of Data Segment Register (DS) is 2222 H.

- To convert this 16-bit address into 20-bit, the BIU appends 0H to the LSBs of the address.
- After appending, the starting address of the Data Segment becomes 22220H.

Contd.

If the data at any location has a logical address specified as:

2222 H : 0016 H

- Then, the number 0016 H is the offset.
- 2222 H is the value of DS.

Contd.

To calculate the effective address of the memory, BIU uses the following formula:

- Effective Address = Starting Address of Segment + Offset
- To find the starting address of the segment, BIU appends the contents of Segment Register with 0H.
- Then, it adds offset to it.

Contd.

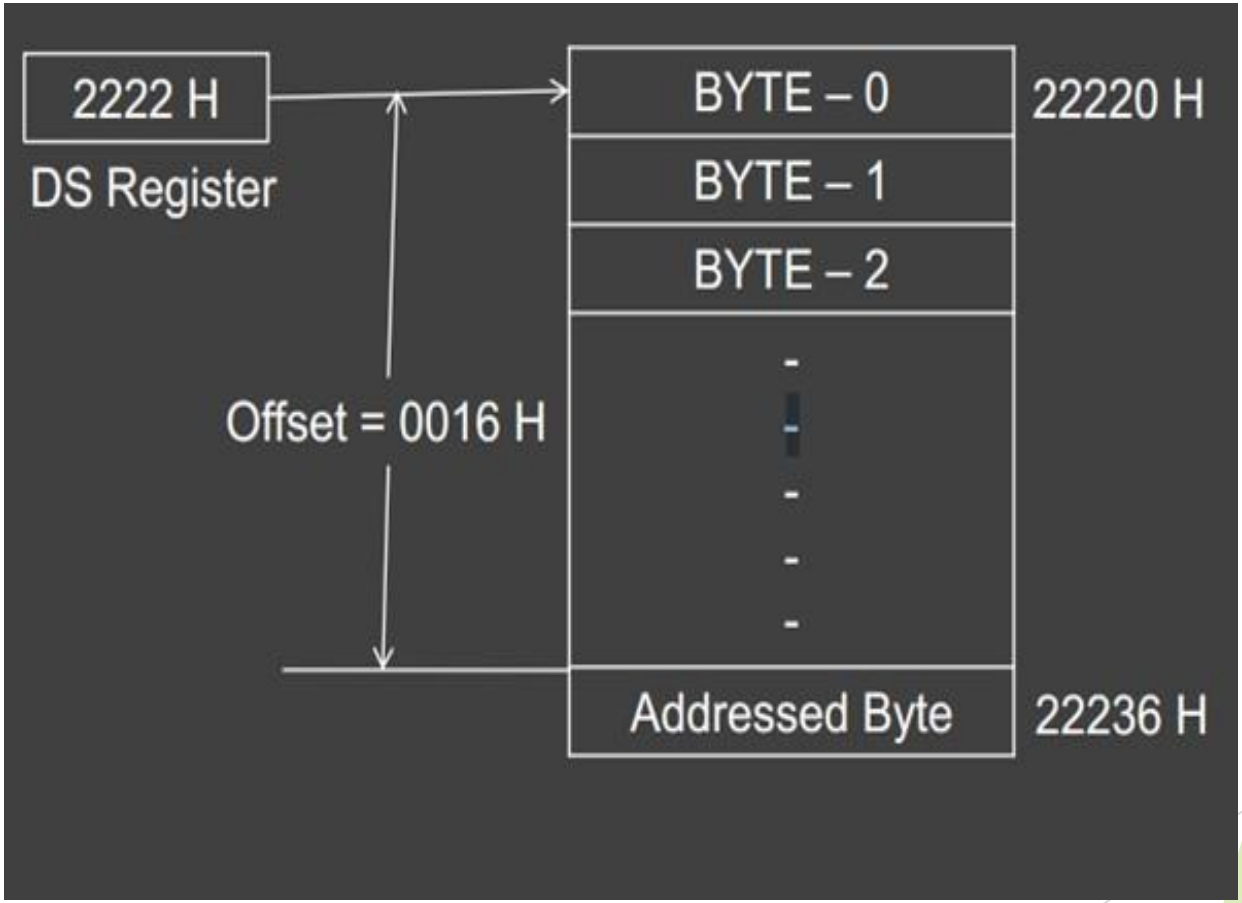
Therefore:

EA = 22220 H

+ 0016 H

22236 H

Contd.


2222 H
DS Register
Offset = 0016 H
BYTE – 0
BYTE – 1
BYTE – 2
Addressed Byte
22220 H
22236 H