8086 Memory Segmentation

The Intel Microprocessors 8th Edition
by Brey
>2–2 REAL MODE MEMORY ADDRESSING
Segments and offset

Assembly language programming Ytha Yu
> Chapter 3 section 3.2

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**.

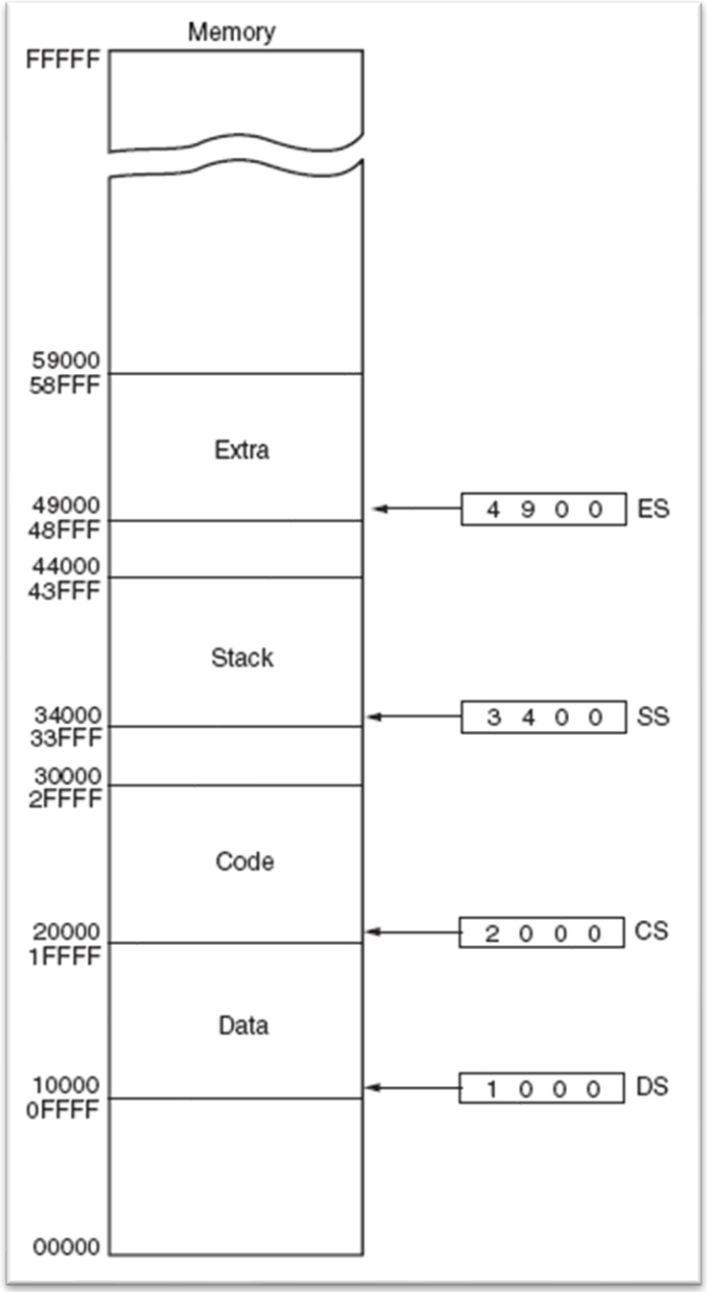


Figure : The 8086 memory segments.

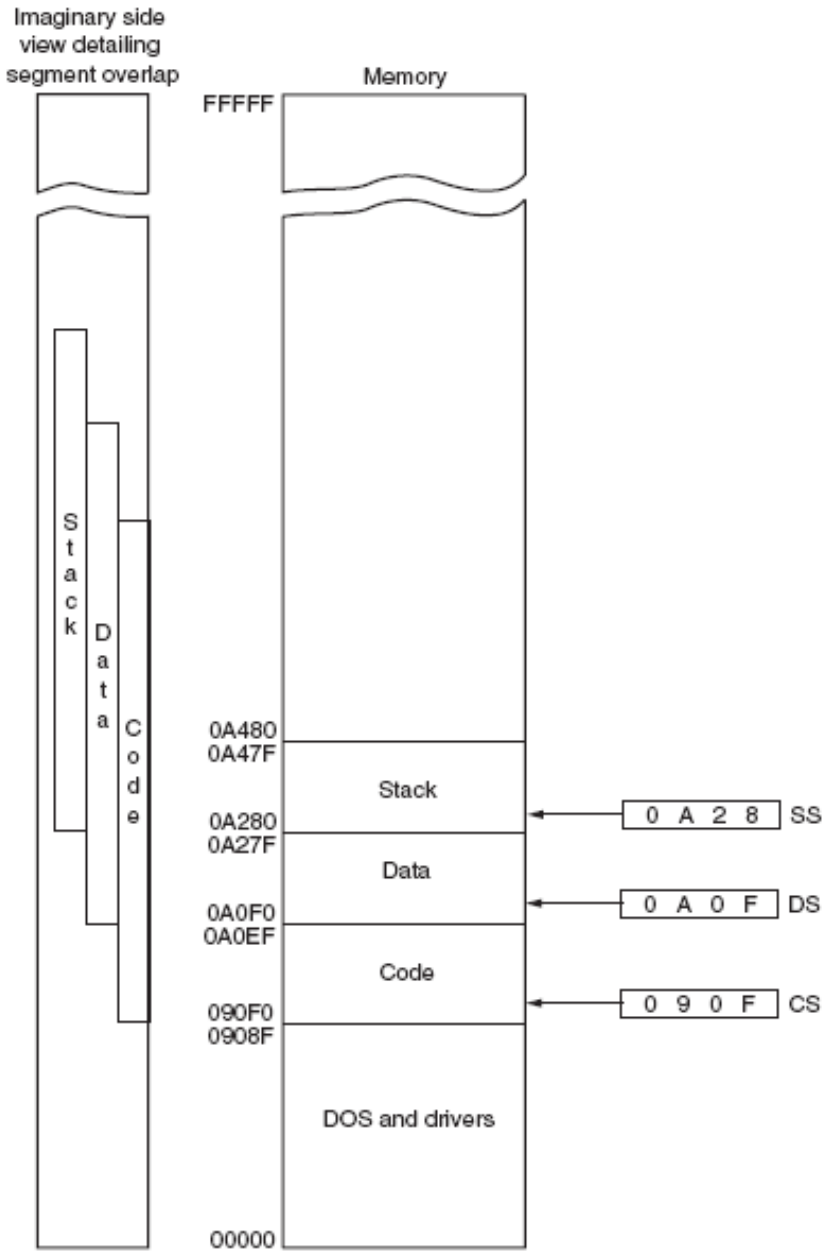


Figure : Overlapping of the 8086 memory segments.

SEGMENTATION AND OFFSET

- Book chapter: Ytha yu 3.2.3 *Segment Registers*
- Memory Segment

A memory segment is a block of 64 KB memory blocks. Each segment is identified by a segment number, starting with 0000H. A segment number is 16 bits, so the highest segment number is FFFFH.

Within a segment, a memory location is specified by an offset. This is the number of bytes from the beginning of the segment. The first byte in a segment has offset 0. The last offset in a segment is FFFFH.

- Segment: Offset Address

A memory location may be specified by providing a segment number and an offset, written in the form segment: offset; this is known as a logical address. For example, A4FB:4872h means offset 4872H within segment A4FBH.

SEGMENTATION AND OFFSET

- Segmentation is used to increase the execution speed of computer system so that processor can able to fetch and execute the data from memory easily and quickly.

- The size of address bus of 8086 is 20 and is able to address 1 Mbytes of physical memory.

- The compete 1 Mbytes memory can be divided into 16 segments, each of 64 Kbytes size.

Once the beginning address is known, the **ending address** is found by adding **FFFFH** (**64K**).

For example, if a segment register contains **3000H**, the first address of the segment is **30000H**, and the last address is or **3FFFFH**.

The following table shows several examples of segment register contents and the starting and ending addresses of the memory segments selected by each segment address.

Segment Register	*Starting Address*	*Ending Address*
2000H	20000H	2FFFFH
2001H	20010H	3000FH
2100H	21000H	30FFFH
AB00H	AB000H	BAFFFH
1234H	12340H	2233FH

- A segment starts at a particular address and its maximum size can go up to 64 Kbytes. But if another segment starts along this 64Kbytes location of the first segment, the two segments are said to be overlapping segment.

MEMORY SEGMENTATION

- In 8086, memory has four different types of segments.
- These are:

-Code Segment

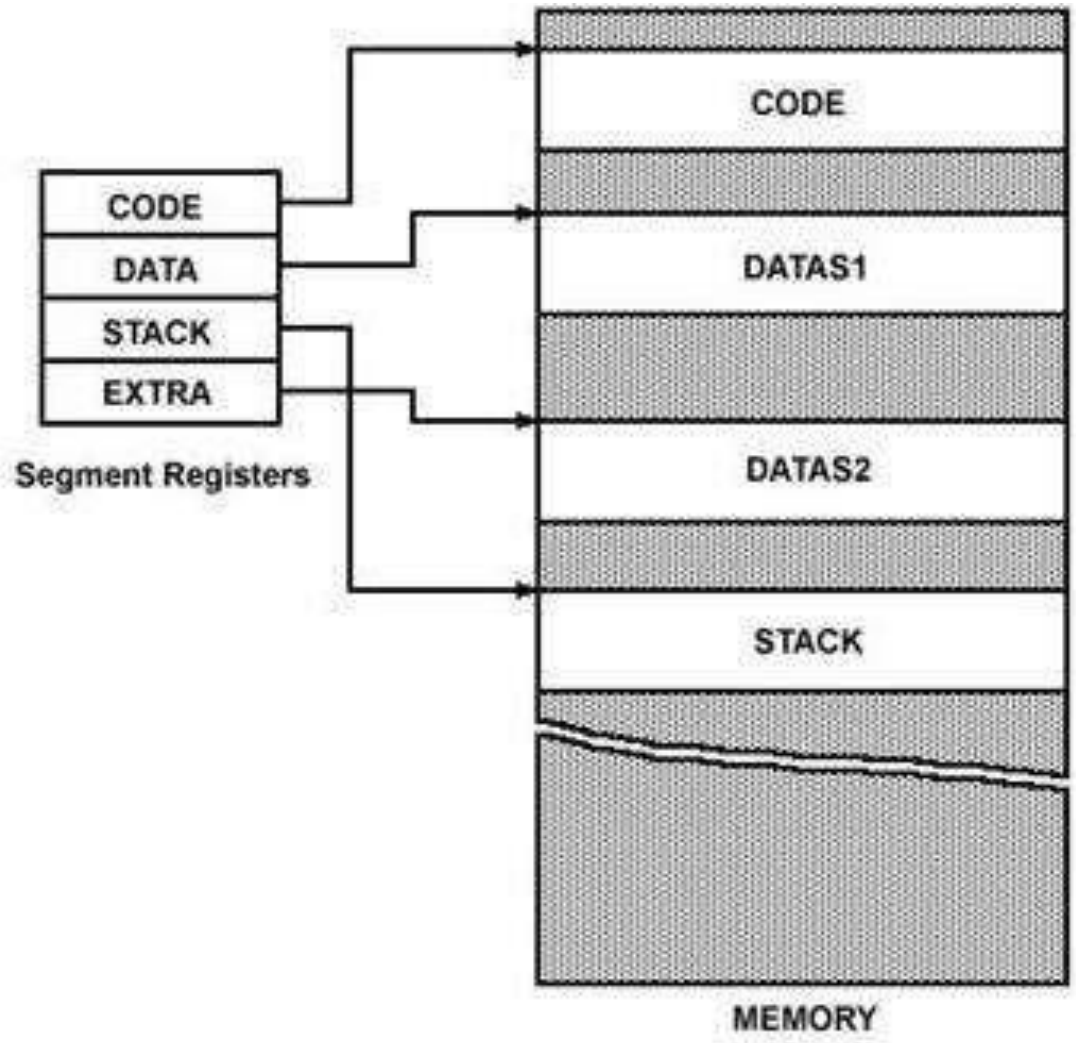
-Data Segment

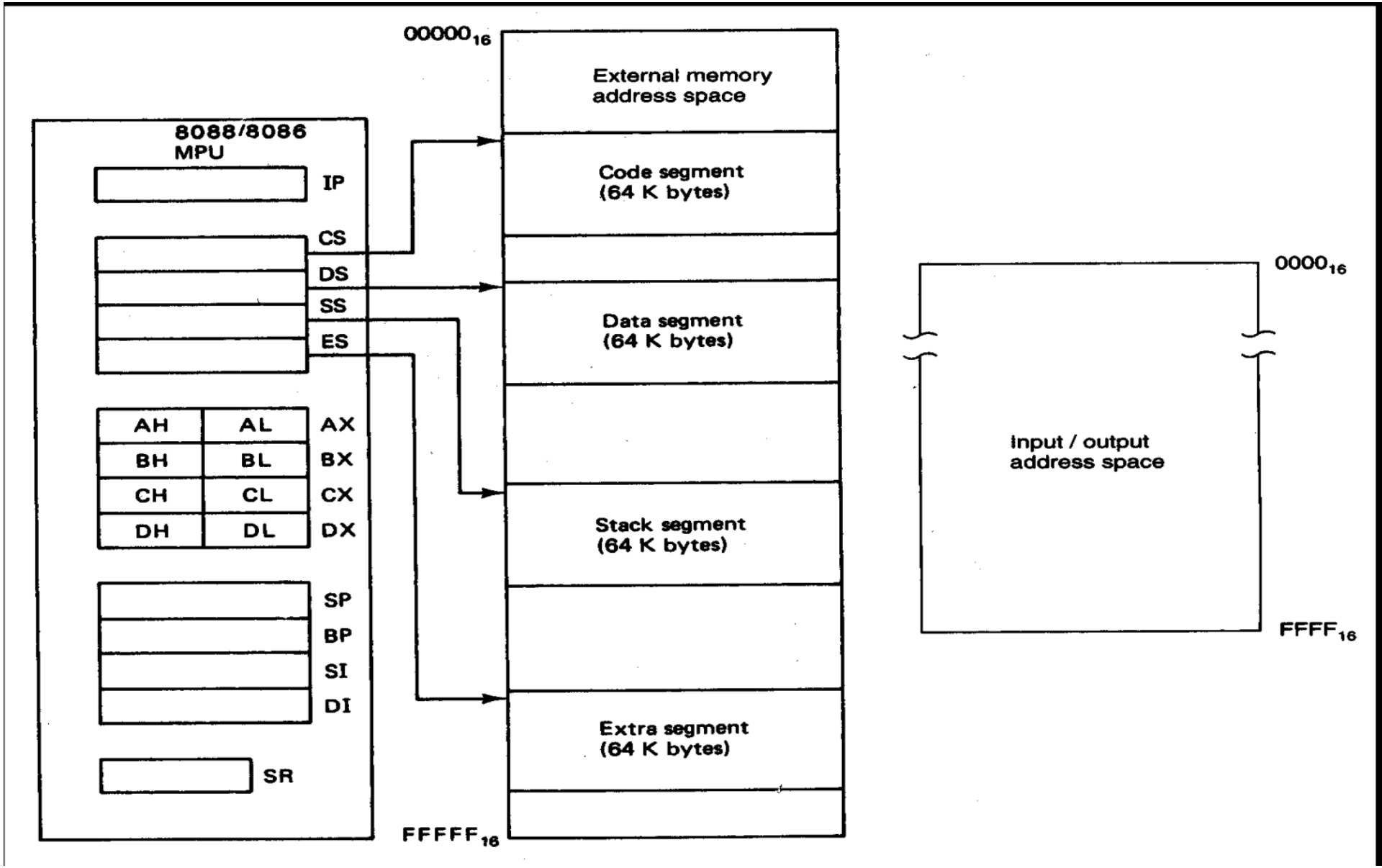
-Stack Segment

-Extra Segment

SEGMENT REGISTERS

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


CODE
DATA
STACK
EXTRA
Segment Registers
CODE
DATAS1
DATAS2
STACK
MEMORY


00000₁₆
External memory address space
8088/8086 MPU
IP
CS
DS
SS
ES
Code segment (64 K bytes)
Data segment (64 K bytes)
Stack segment (64 K bytes)
Extra segment (64 K bytes)
AH
AL
AX
BH
BL
BX
CH
CL
CX
DH
DL
DX
SP
BP
SI
DI
SR
FFFFF₁₆
0000₁₆
Input / output address space
FFFF₁₆

SEGMENT REGISTERS

- Q: How is a 20-bit address obtained if there are only 16-bit registers?

- The 20-bit address is called **Physical Address**.

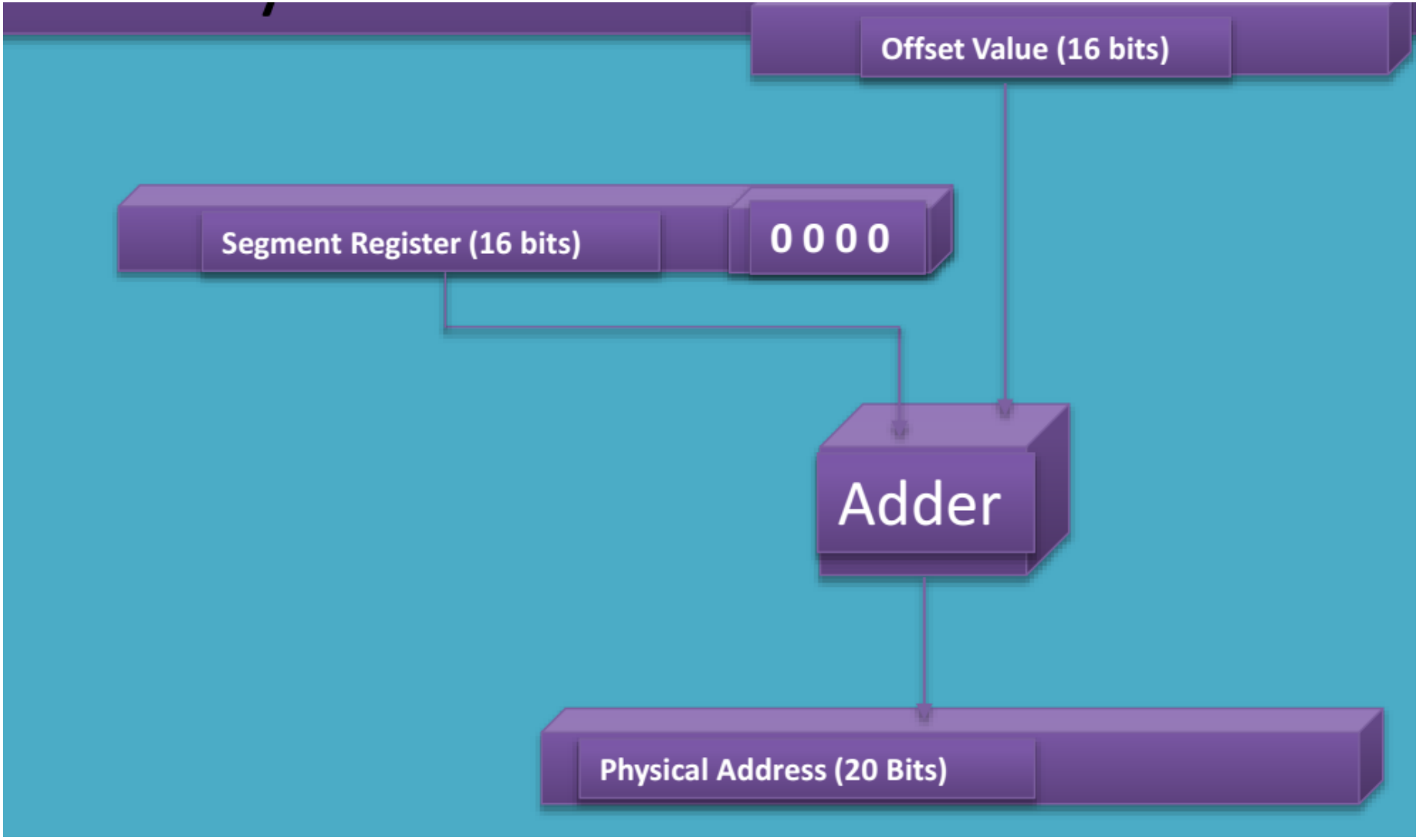
  The 16-bit address is called **Logical Address**.

- 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.

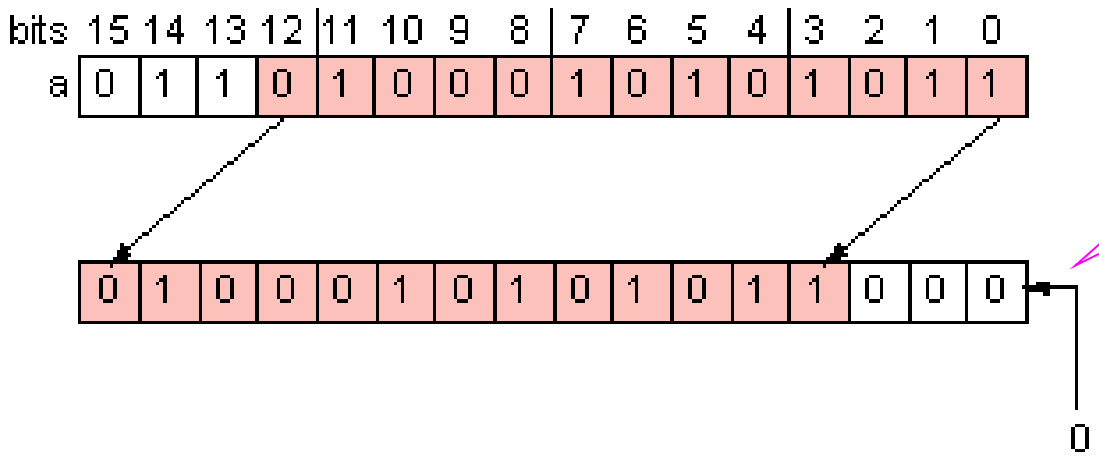
  Thus a memory location may be specified by providing the 16-bit segment base address, and a 16-bit offset, written in the form **segment : offset**; this is known as a logical address for the memory location.


Offset Value (16 bits)
Segment Register (16 bits)
0 0 0 0
Adder
Physical Address (20 Bits)

SEGMENT REGISTERS

For example, the logical address **A4FBH : 4872H** means offset 4872H within segment A4FBH, that is, the segment starting at physical address A4FB0H.
To obtain the corresponding 20-bit physical (i.e., absolute) address, the 8086 microprocessor first shifts the segment base address 4 bits to the left (this is equivalent to multiplying by 10H), and then adds the offset. Thus the physical address for A4FB:4872h is:

```
   A4FB0h
+   4872h
   A9822h      (20-bit physical address)
```



Left shift 3 bit

Bits positions vacated by shift are filled with zeros

EXAMPLE

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

2222 H : 0016 H

Then, 2222 H is the value of Base address. The number 0016 H is the offset . Calculate the physical address?

EXAMPLE

- The value of Code Segment Register (CS) is 2222 H
- Convert this 16-bit address into 20-bit,
- The starting address of the Code Segment becomes 22220H

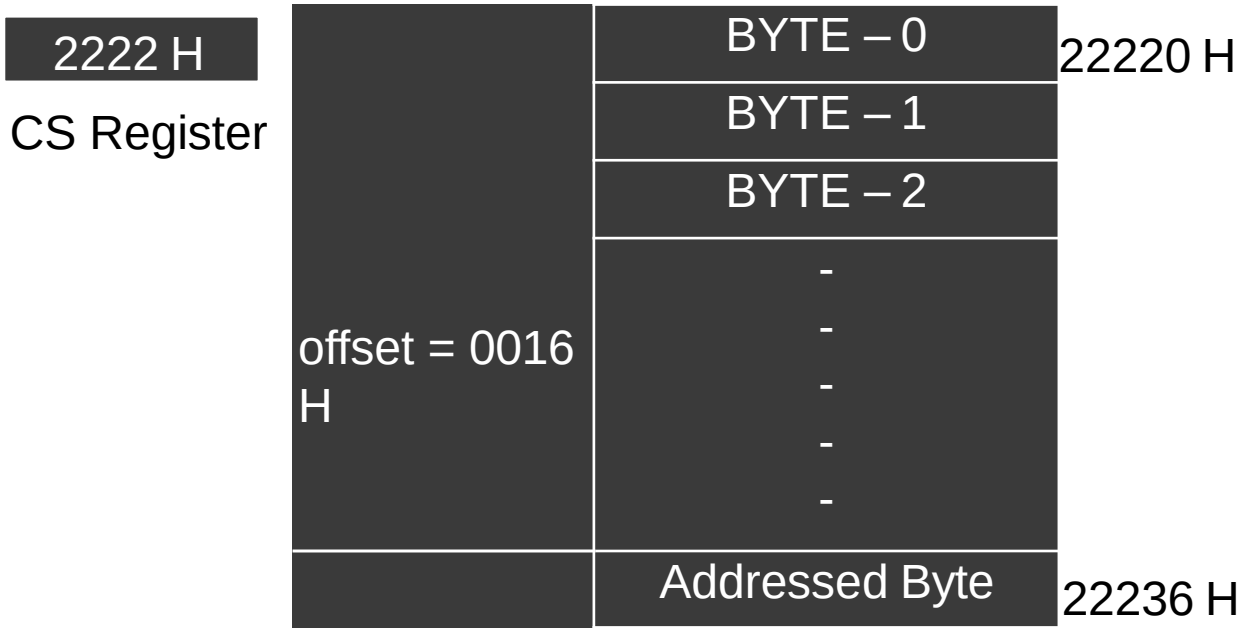
EXAMPLE (CONTD.)

- Therefore:

2222 * 10 H

+ 0016 H

- Physical address = 22236 H



EXAMPLE (CONTD.)

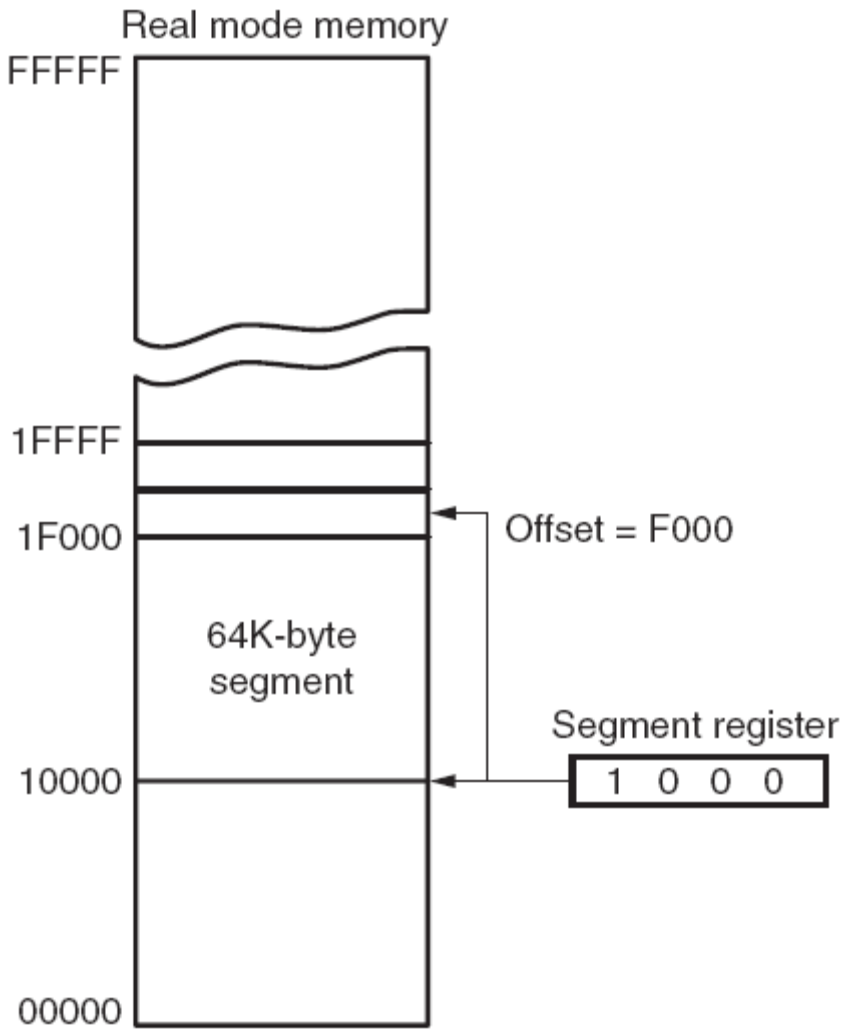


Figure : The 8086 memory-addressing, using a segment address plus an offset.

Segment: Offset Address (Logical Address)

- To obtain a 20 bit physical address,
  - - Shift the segment address 4 bit to the left (i.e. multiply by 10h)
  - - Add the offset

  Example:

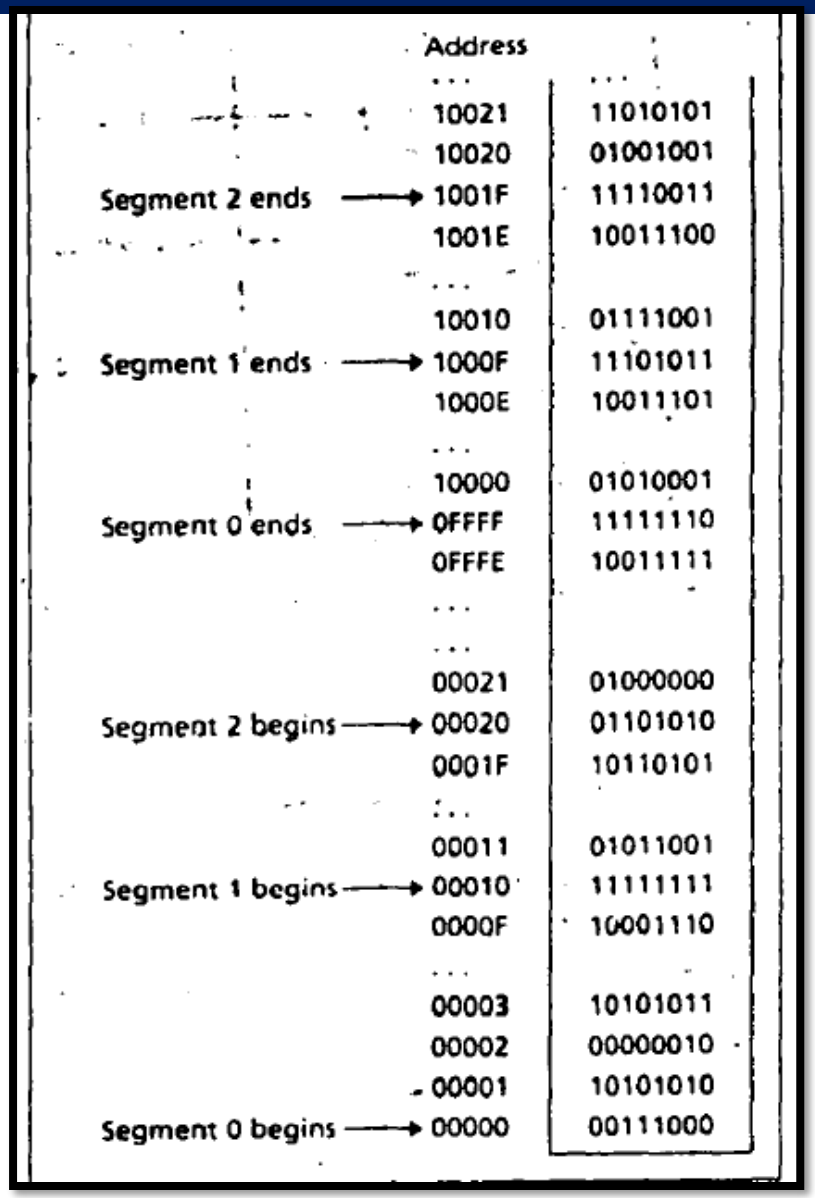
  Segment : offset = A4FB:4872h

  Physical address

  = segment number * 10h + offset

  = A4FBh * 10h + 4872h = A9822h

Segments may overlap, so the segment : offset form of an address is not unique.



Default Segment and Offset Registers

- The 8086 has a set of rules that apply to segments whenever memory is addressed. These rules, define the segment register and offset register combination. For example, the **code segment register** (**CS**) is always used with the **instruction pointer** (**IP**) to address the next instruction in a program.
- The **code segment register** defines the **start** of the **code segment** and the **instruction pointer** locates the **next instruction** within the code segment.
- This combination (**CS:IP**) locates the next instruction executed by the CPU.
- For example, if **CS**=**1400H** and **IP**=**1200H** , the microprocessor fetches its next instruction from memory location or **15200H**.

Default Segment and Offset Registers

- Another default combinations is the stack.

Stack data are referenced through the **stack segment register** (**SS**) at the memory location addressed by either the **stack pointer** (**SP**) or the pointer (**BP**).

- These combinations are referred to as **SS:SP** or **SS:BP**. For example, **SS**=**2000H** and **BP**=**3000H** , the CPU addresses memory location **23000H** for the stack segment memory location. Other defaults of segment and offset combinations are shown in the following table.

WHERE TO LOOK FOR THE OFFSET

Segment	Offset Registers	Function
CS	IP	Address of the next instruction
DS	BX, DI, SI	Address of data
SS	SP, BP	Address in the stack
ES	BX, DI, SI	Address of destination data (for string operations)

QUESTION

- The contents of the following registers are:
  - CS = 1111 H
  - DS = 3333 H
  - SS = 2526 H
  - IP = 1232 H
  - SP = 1100 H
  - DI = 0020 H
- Calculate the corresponding physical addresses for the address bytes in CS, DS and SS.

SOLUTION

1. CS = 1111 H

- The base address of the code segment is 11110 H.
- Physical address of memory is given by 11110H + 1232H = 12342H.

2. DS = 3333 H

- The base address of the data segment is 33330 H.
- Physical address of memory is given by 33330H + 0020H = 33350H.

3. SS = 2526 H

- The base address of the stack segment is 25260 H.
- Physical address of memory is given by 25260H + 1100H = 26360H

- Because segments may overlap, the segment : offset form of an address is not unique for a particular memory location as is the case for the physical address of that memory location.
- **Example 3.1** For the memory location whose physical address is specified by 1256Ah, give the address in **segment : offset** for segments 1256h and 1240h

- **Example 3.1** For the memory location whose physical address is specified by 1256Ah, give the address in segment : offset for segments 1256h and 1240h
- **Solution**:

  Physical address = segment number * 10h + offset

  => 1256AH = 1256h * 10H + offset

  => Offset = 1256AH – 12560H = AH

  So, **1256AH = 1256 H : 000AH**

  Physical address = segment number * 10h + offset

  => 1256AH = 1240H * 10H + offset

  => Offset = 1256AH – 12400H = 16AH

  So, **1256AH = 1240 H : 016AH**

Segment: Offset Address (Logical Address)

- **Example 3.2** A memory location has physical address 805D2h. In what segment does it have offset BFD2h?

Segment: Offset Address (Logical Address)

- **Example 3.2** A memory location has physical address 805D2h. In what segment does it have offset BFD2h?
- **Solution**:

  Physical address = segment number * 10h + offset

  => 805D2h = segment number * 10h + BFD2h

  ⇒ Segment number = (805D2h – BFD2h)/10h = 74600h/10h = 7460h

  So, **805D2h = 7460 h : BFD2h**