Chapter One

Introduction to the Computer and Microprocessor

1.8 Physical Addressing:

In the 8086 there are three types of addresses:

(1) Physical address: *is the 20-bit address that is actually put on the address pins of 8086microprocessor. This address can have a range of 00000H to FFFFFH (1M byte).*

(2) Offset address: *is a location within 64K-byte segment range. Therefore, an offset address can range from 0000H to FFFFH.*

(3) Logical address: *is consist of a segment value and an offset address*

Physical Address (PA) = (Segment Register *10) + Effective Address

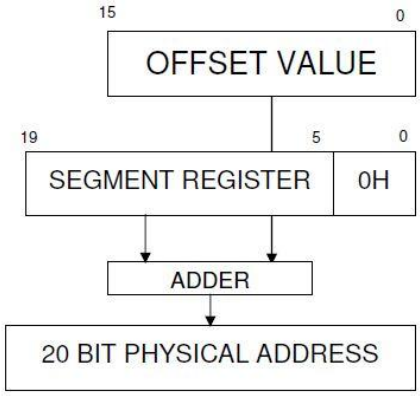


Figure (8) Physical address formation

Table (2) 8086 through Pentium Pro segment and offset address combinations.

Segment register	Registers (8086)	Registers (80286,80386, Pentium and Pentium Pro)
CS	IP	EIP
ES	DI	EDI
SS	SP, BP	ESP, EBP
DS	BX, SI, DI, (8 and 16-bits) and direct addressing	EAX, EBX, ECX, EDX, ESI, EDI and (8bit-32bit)number

Example (2): *If CS=2500H, IP=24D6H find:*

a) Logical address b) offset address c) physical address d) The lower and upper range

Solution:

a) CS: IP = 2500H: 24D6H.

b) IP = 24D6H

*c) PA = (Segment register * 10) + EA*

*= CS*10 + IP*

= 25000H+ 24D6H =274D6H

d) The lower range (IP=EA=0000H)

*PA = (Segment register * 10) + EA*

*= CS*10 + 0000H*

= 25000H+ 0000H =25000H

The upper range (IP=EA=FFFFH)

*PA = (Segment register * 10) + EA*

*= CS*10 + FFFFH*

= 25000H+ FFFFH =34FFFH

Example (3): *If CS=2500H, SS=4850H, SP= 98EAH find:*

a) Logical address b) offset address c) physical address d) The lower and upper range

Solution:

a) SS: SP = 4850H: 98EAH.

b) SP = 98EAH

*c) PA = (Segment register * 10) + EA*

*= SS*10 + SP*

= 48500H+ 98EAH =51DEAH

e) d) The lower range (SP=EA=0000H)

*PA = (Segment register * 10) + EA*

*= SS*10 + 0000H*

= 45800H+ 0000H =45800H

The upper range (SP=EA=FFFFH)

*PA = (Segment register * 10) + EA*

*= SS*10 + FFFFH*

= 45800H+ FFFFH =557FFH

1.9 Stack Concept

The stack is a section of read/write memory and it is used for temporary storage of information such as data and addresses. The stack is 64Kbytes long. The CPU needs this storage area since there are only a limited number of registers. The main disadvantage of the stack is its access time.

The first address in the Stack segment (SS: 0000) is called End of Stack. The last address in the Stack segment (SS: FFFE) is called Bottom of Stack. The address (SS: SP) is called Top of Stack. Figure (9) shows the stack memory.

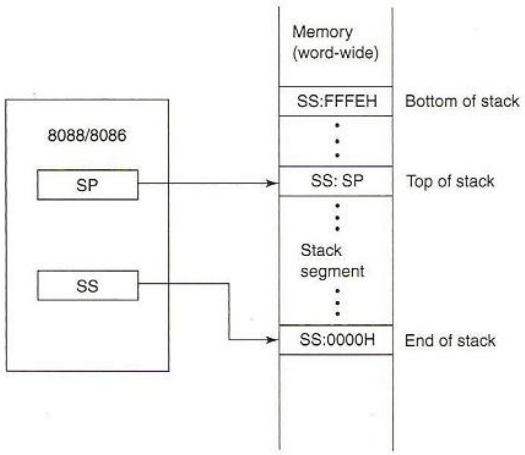


Figure (9) shows the stack memory.

The stack instructions are important instructions that store and retrieve data from the FIFO (First In First Out) stack memory. The general forms of these instructions are as shown below.

Mnem.	***Meaning***	***Format***	***Operation***	***Flags affected***
PUSH	*Push word onto stack*	*PUSH S*	*((SP))←(S)* *SP =SP-2*	*None*
POP	*Pop word off stack*	*POP D*	*(D)←((SP))* *SP= SP+2*	*None*

Example (4): *Assuming that SP=1238H, AX=24B6H, and DI=5F93H, show the contents of the stack as each of the following instruction is executed?*

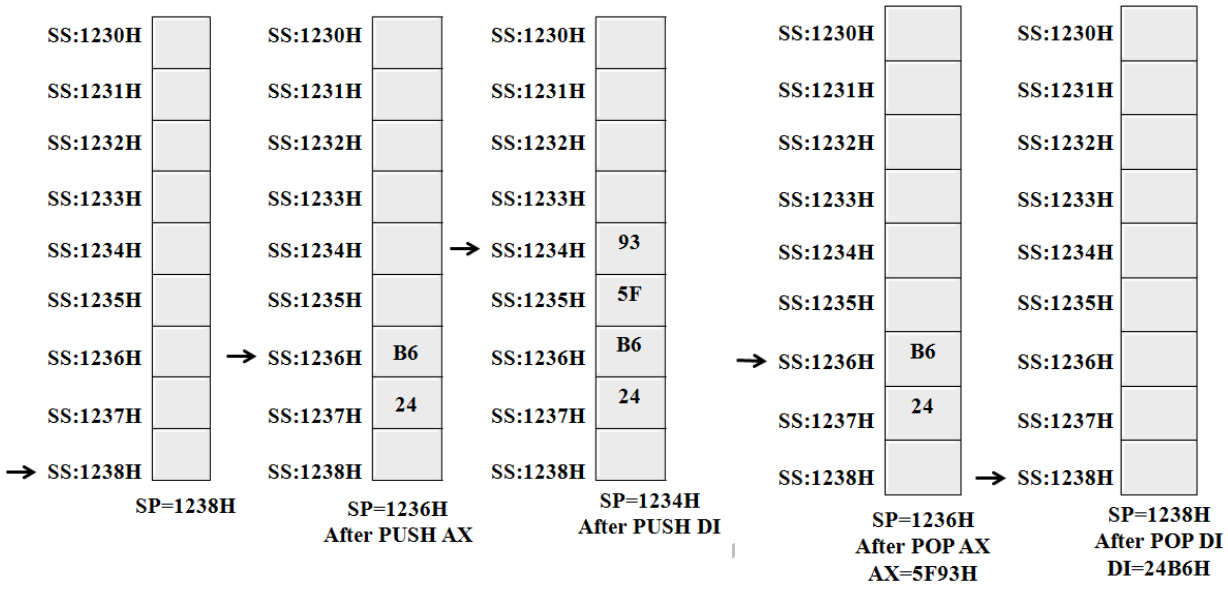
PUSH AX

PUSH DI

POP AX

POP DI

Solution:



1.10 Data-Addressing Modes:

The programming model of the 8086 though the Pentium Pro is considered ***program visible*** *because its registers are used during programming and are specified by the instructions. Other registers are considered program invisible because they are not addressed directly during applications programming, but may be used indirectly during system programming. Only 80286 and above contain the program invisible registers used to control and operate the protected memory system.*

The earlier 8086, 8088 and 80286 microprocessors contain 16-bit internal architectures, a subset of the registers shown in Figure (10). The 80386, 80483, Pentium and Pentium Pro microprocessor contain full 32-bit internal architectures

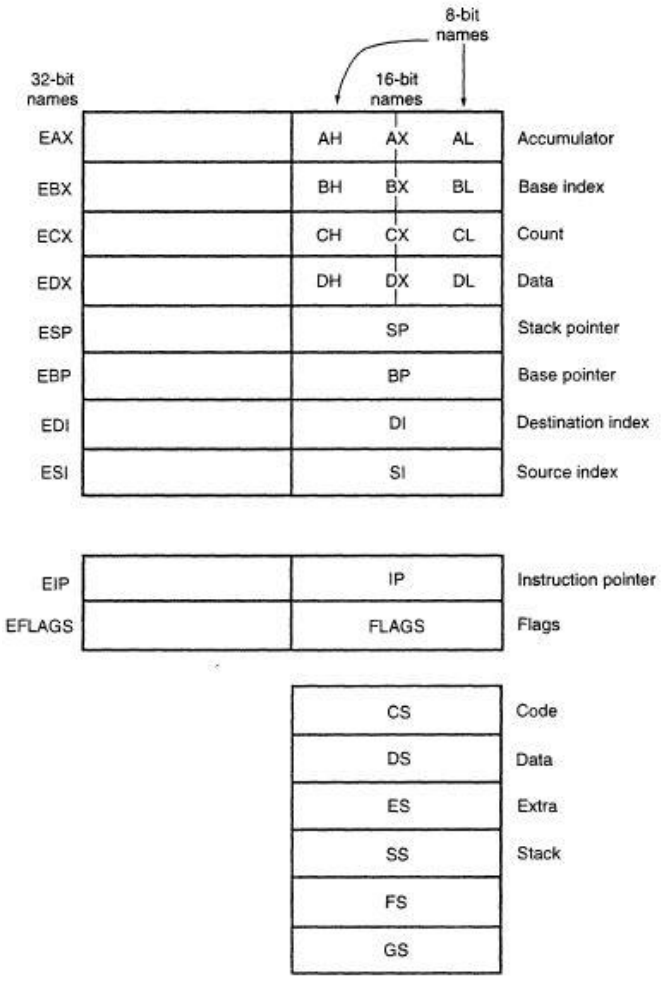


Figure (10) the programming model of the Intel 8086 through Pentium Pro

*In this section we use the **MOV** instruction is a very common and flexible instruction, it provides a basis for the explanation of the data-addressing modes. Figure (11) illustrates the MOV instruction and defines the direction of data flow. The **source** is to the right and the **destination** is to the left, next to the opcode MOV. (An **opcode**, or operation code, tells the microprocessor which operation to perform.)*

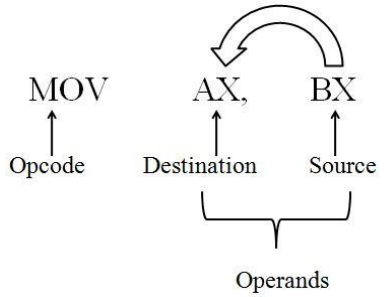


Figure (11) The mov instruction.

- ***Register Addressing:*** *Register addressing transfers a copy of a byte or word from the source register or contents of a memory location to the destination register*

TABLE (3) Examples of register-addressed instructions

Assembly Language	***Size***	***Operation***
MOV AL, BL	*8 bits*	*Copies BL into AL*
MOV CH, CL	*8 bits*	*Copies CL into CH*
MOV AX, CX	*16 bits*	*Copies CX into AX*
MOV SP, BP	*16 bits*	*Copies BP into SP*
MOV DS, AX	*16 bits*	*Copies AX into DS*
MOV SI, DI	*16 bits*	*Copies DI into SI*
MOV BX, ES	*16 bits*	*Copies ES into BX*
MOV DS, CX	*16 bits*	*Copies CX into DS*
MOV ES, DS	—	*Not allowed (segment-to-segment)*
MOV BL, DX	—	*Not allowed (mixed sizes)*
MOV CS, AX	—	*Not allowed (the code segment register may not be the destination register)*

- ***Immediate Addressing:*** *Immediate addressing transfers the source, an immediate byte, word into the destination register or memory location.*

TABLE (4) Examples of immediate addressing using the MOV instruction

Assembly Language	***size***	***Operation***
MOV BL, 44	*8 bits*	*Copies 44 decimal (2CH) into BL*
MOV CH,100	*8 bits*	*Copies 100 decimal (64H) into CH*
MOV AL,'A'	*8 bits*	*Copies ASCII A into AL*
MOV CL,11001110B	*8 bits*	*Copies 11001110 binary into CL*
MOV SI,0	*16 bits*	*Copies 0000H into SI*
MOV AX,44H	*16 bits*	*Copies 0044H into AX*
MOV DS, 50H	_	*Not allowed (copies direct to segment register)*

1.11 Memory Addressing Modes:

The 8086 MPU is provided with a group of addressing modes known as the memory addressing modes

- ***Direct Addressing Modes:*** *Direct addressing moves a byte or word between a memory location and a register.*

TABLE (5) Examples of Direct addressing using the MOV instruction

Assembly Language	***size***	***Operation***
MOV BL, [1000H]	*8 bits*	*Copies the content of [1000H] into BL*
MOV [200H], SI	*16 bits*	*Copies SI into [200H]*
MOV DS, [5000H]	*16 bits*	*Copies the content of [5000H] into DS*

MOV [5060H], 50H	*8 bits*	*Copies 50H into [5060H]*
MOV [200H], [500H]	*16 bits*	*Not allowed (memory to memory)*

Example (5): *For the following instruction calculate the Physical Addressing (PA) where DS=3F40H*

1) *MOV AX, [9000H]*
2) *MOV SS, [8750H]*
3) *MOV DI, [6A21H] + 05H*

Solution:

1) *PA = (Segment register * 10) + EA*
   *= DS*10 + 9000H*
   = 3F400H+ 9000H =48400H

2) *PA = (Segment register * 10) + EA*
   *= DS*10 + 8750H*
   = 3F400H+ 8750H =47B50H

3) *PA = (Segment register * 10) + EA*
   *= DS*10 + (6A21 +05) H*
   = 3F400H+ 2A26H =45E26H

- ***Register Indirect Addressing Modes:*** *Register indirect addressing transfers a byte or word between a register and a memory location addressed by an index or base register. The index and base registers are BP, BX, DI, and SI.*

TABLE (6) Examples of Register indirect addressing using the MOV instruction

Assembly Language	***size***	***Operation***
MOV [BP], DL	*8 bits*	*Copies DL into the stack segment memory location addressed by BP*

MOV [DI], BH	*8 bits*	*Copies BH into the data segment memory location addressed by DI*
MOV CX, [BX]	*16 bits*	*Copies the word contents of the data segment memory location addressed by BX into CX*

Example (6): *For the following instruction calculate the Physical Addressing (PA) where [DS=5500H, SS =4500H, BP =1000H and DI =30H]*

1) *MOV AX, [BP]*
2) *MOV CX, [DI]*

Solution:

1) *PA = (Segment register * 10) + EA*
   *= SS*10 + 1000H*
   = 45000H+ 1000H =46000H
2) *PA = (Segment register * 10) + EA*
   *= DS*10 + 30H*
   = 55000H + 30H =55030H

- ***Based Relative Addressing Modes:*** *In the based relative addressing mode, based register BX and BP, as well as a displacement value, are used to calculate the effective address.*

$$\begin{bmatrix} BX \\ BP \end{bmatrix} + \begin{bmatrix} 8-bit\ displacment \\ 16-bit\ displacment \end{bmatrix}$$

Example (7): *Calculate the Physical Addressing (PA) if SS=5500H and BP=3000H where:*

MOV AX, [BP] + 1234H

Solution:

*PA = (Segment register * 10) + EA*

*= SS*10 + (3000H+ 1234H)*

= 55000H+ 4234H =59234H

- ***Indexed Relative Addressing Modes:*** *The indexed relative addressing mode works the same as the based relative addressing modes, except that registers DI and SI hold the offset address.*

$$\begin{bmatrix} DI \\ SI \end{bmatrix} + \begin{bmatrix} 8-bit\ displacment \\ 16-bit\ displacment \end{bmatrix}$$

Example (8): *Calculate the Physical Addressing (PA) if DS=E463H and SI=500H where:*

MOV [SI] + 6H, ES

Solution:

*PA = (Segment register * 10) + EA*

*= DS*10 + (500H+ 6H)*

= E4630H+ 506H =E4B36H

- ***Based-Indexed Relative Addressing Modes:*** *By combining based and indexed addressing modes, a new addressing mode is derived called the based indexed addressing mode. In this mode, one base register and one index register are used.*

$$\begin{bmatrix} BX \\ BP \end{bmatrix} + \begin{bmatrix} DI \\ SI \end{bmatrix} + \begin{bmatrix} 8-bit\ displacment \\ 16-bit\ displacment \end{bmatrix}$$

Example (9): *Calculate the Physical Addressing (PA) if SS=6000H, DS=8000H, BP=3000H and SI=500H where:*

MOV [BP + SI] + 200H, ES

Solution:

*PA = (Segment register * 10) + EA*

*= SS*10 + (3000H+ 500H+ 200H)*

= 60000H+ 3700H =63700H

Example (10): *Calculate the Physical Addressing (PA) if DS=6000H, ESI=200H, ECX=100H, EBX = 50H and EDI=100H for the following instructions.*

A) *MOV AX, [2000+ESI*4]* ***B)*** *MOV AX, [5000+ECX*2]*

C) *MOV ECX, [2400+EBX*4]* ***D)*** *MOV DX, [100+EDI*8]*

Solution:

A. *PA = (Segment register * 10) + EA*

*= DS*10 + (2000H+200H*4)*

= 60000H+ (2000H+800H) =60000H+2800H= 62800H

B. *PA = (Segment register * 10) + EA*

*= DS*10 + (5000H+100H*2)*

= 60000H+ (5000H+200H) =60000H+5200H= 65200H

C. *PA = (Segment register * 10) + EA*

*= DS*10 + (2400H+50H*4)*

= 60000H+ (2400H+140H) =60000H+2540H= 62540H

D. *PA = (Segment register * 10) + EA*

*= DS*10 + (100H+100H*8)*

= 60000H+ (100H+800H) =60000H+900H= 60900H

Example (11): *What is the result of executing the following instruction sequence?*

Where DS = 5000H

1. ***MOV EAX, 7698E39FH***
2. ***MOV [4524H], AX***
3. ***MOV [8000H], EAX***

Solution:

1. *EAX = 7698E39FH*
2. *PA = (Segment register * 10) + EA*
   *= DS*10 + 4524H*
   = 50000H+ 4524H =54524H
3. *PA = (Segment register * 10) + EA*
   *= DS*10 + 8000H*
   = 50000H+ 8000H =58000H