Addressing Modes of 8086 - I

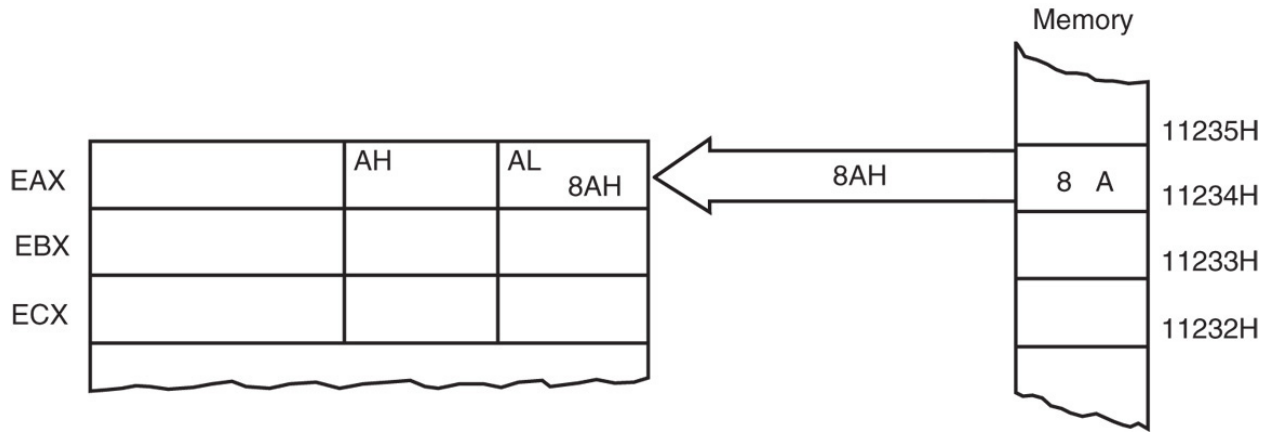
Direct Data Addressing

• Applied to many instructions in a typical program.
• Two basic forms of direct data addressing: – direct addressing, which applies to a MOV between a memory location and AL, AX, or EAX – displacement addressing, which applies to almost any instruction in the instruction set
• Address is formed by adding the displacement to the default data segment address or an alternate segment address.

Direct Addressing

• Direct addressing with a MOV instruction transfers data between a memory location, located within the data segment, and the AL (8-bit), AX (16-bit), or EAX (32-bit) register.
– usually a 3-byte long instruction
• MOV AL,DATA loads AL from the data segment memory location DATA (1234H).

– DATA is a symbolic memory location, while 1234H is the actual hexadecimal location



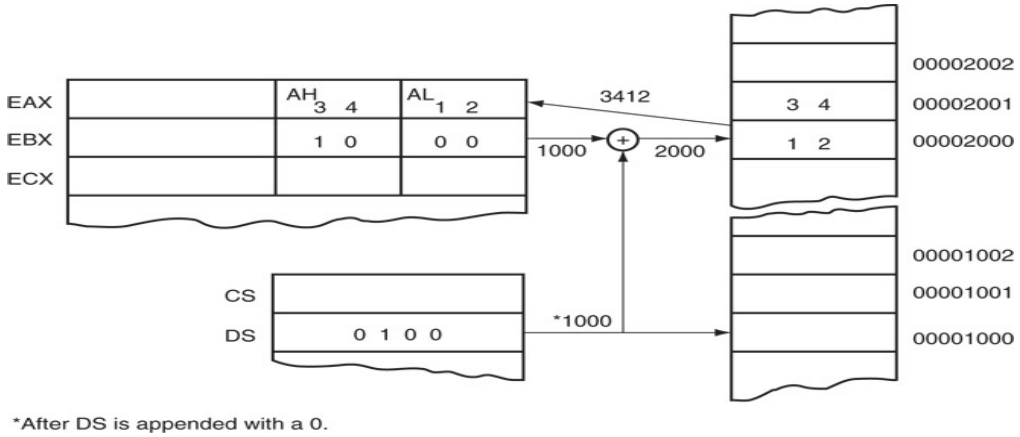
This instruction transfers a copy contents of memory location 11234H into AL. – the effective address is formed by adding 1234H (the offset address) and 10000H (the data segment address of 1000H times 10H) in a system operating in the real mode

Displacement Addressing

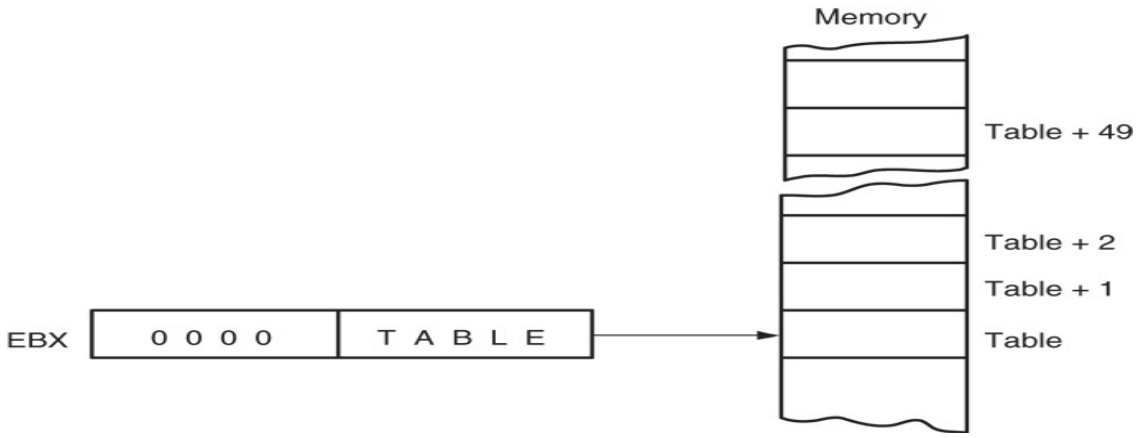
• Almost identical to direct addressing, except the instruction is 4 bytes wide instead of 3.
• In 80386 through Pentium 4, this instruction can be up to 7 bytes wide if a 32-bit register and a 32-bit displacement are specified.
• This type of direct data addressing is much more flexible because most instructions use it.

Register Indirect Addressing

• Allows data to be addressed at any memory location through an offset address held in any of the following registers: BP, BX, DI, and SI.
• In addition, 80386 and above allow register indirect addressing with any extended register except ESP.
• In the 64-bit mode, the segment registers serve no purpose in addressing a location in the flat model.
The **data segment** is used by default with register indirect addressing or any other mode that uses BX, DI, or SI to address memory.
• If the BP register addresses memory, the **stack segment** is used by default. – these settings are considered the default for these four index and base registers
• For the 80386 and above, EBP addresses memory in the stack segment by default.
• EAX, EBX, ECX, EDX, EDI, and ESI address memory in the data segment by default.
When using a 32-bit register to address memory in the real mode, contents of the register must never exceed 0000FFFFH.



• In the protected mode, any value can be used in a 32-bit register that is used to indirectly address memory. – as long as it does not access a location outside the segment, dictated by the access rights byte
• In the 64-bit mode, segment registers are not used in address calculation; the register contains the actual linear memory address. In some cases, indirect addressing requires specifying the size of the data by the **special assembler directive** BYTE PTR, WORD PTR, DWORD PTR, or QWORD PTR. – these directives indicate the size of the memory data addressed by the memory **pointer** (PTR)
• The directives are with instructions that address a memory location through a pointer or index register with immediate data.
• With SIMD instructions, the octal OWORD PTR, represents a 128-bit-wide number Indirect addressing often allows a program to refer to tabular data located in memory.
• Figure 3–7 shows the table and the BX register used to sequentially address each location in the table.
• To accomplish this task, load the starting location of the table into the BX register with a MOV immediate instruction.
• After initializing the starting address of the table, use register indirect addressing to store the 50 samples sequentially



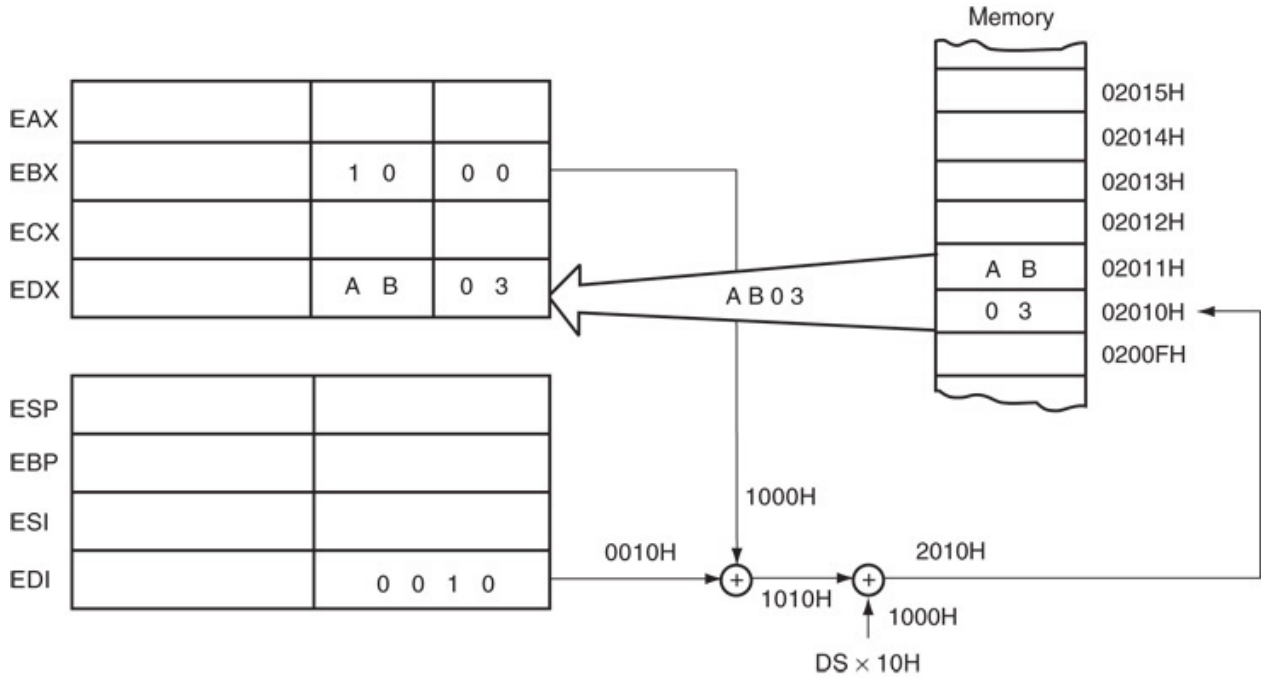
Base-Plus-Index Addressing

• Similar to indirect addressing because it indirectly addresses memory data.

• The base register often holds the beginning location of a memory array. – the index register holds the relative position of an element in the array – whenever BP addresses memory data, both the stack segment register and BP generate the effective address

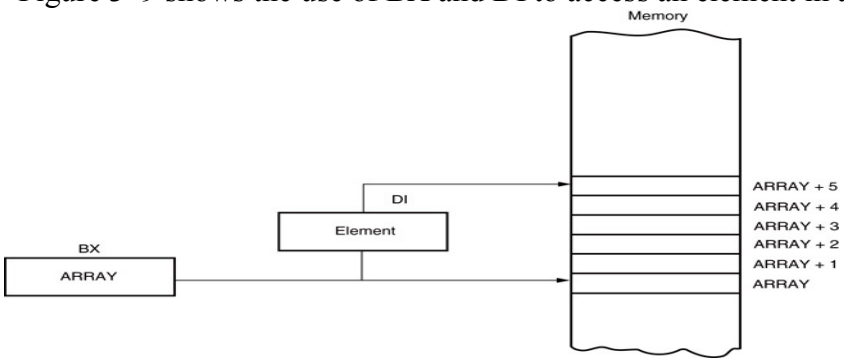
Locating Data with Base-Plus-Index Addressing

• Figure 3–8 shows how data are addressed by the MOV DX,[BX + DI] instruction when the microprocessor operates in the real mode.
• The Intel assembler requires this addressing mode appear as [BX][DI] instead of [BX + DI].
• The MOV DX,[BX + DI] instruction is MOV DX,[BX][DI] for a program written for the Intel ASM assembler.



Plus-Index Addressing

• A major use is to address elements in a memory array.
• To accomplish this, load the BX register (base) with the beginning address of the array and the DI register (index) with the element number to be accessed.
• Figure 3–9 shows the use of BX and DI to access an element in an array of data.



Source : http://nprcet.org/e%20content/Misc/e-Learning/IT/IV%20Sem/CS%202252-Microprocessors%20and%20Microcontrollers.pdf