8086 INSTRUCTION SET

DATA TRANSFER INSTRUCTIONS

MOV - MOV Destination, Source

The MOV instruction copies a word or byte of data from a specified source can be a register, a memory location or an immediate number. The source and destination cannot both be memory locations. They must both be of the same type (bytes or words) MOV instruction does not affect any flag.

> MOV CX, 037AH — Put immediate number 037A to CX

> MOV BL, [437AH] — copy byte in Ds at offset 437AH to BL

> MOV AX, BX — Copy ~~byte~~ content of regist BX to AX

> MOV DL, [BX] — Copy byte from memory at [BX] to DL

> MOV DS. BX Copy word from BX to DS register

> MOV ~~RE~~ RESULT[BP], AX Copy AX to two memory locations; AL to the first location, AH to the second; EA of the first memory location insun of the ~~de~~ displacement ~~+esp~~ represnted by RESULTS and content of BP.
Physical address = EA+ss

> MOV ES: RESULTS [BP], AX Same as the ~~abov~~ above instruction, but physical ~~adA address=EA+~~ address = EA+ES. because of the segme override perfix ES

XCHG – XCHG Destination, Source

The XCHG instruction exchanges the content of a register with the content of another register or with the content of memory location(s). It cannot directly exchange the content of two memory ~~location~~ locations. The source and destination must both be of the same type (bytes or words). The segment registers cannot be used in this instruction. ~~The~~ This instruction does not affect any flag.

> LEA BX, PRICES Load BX with offset of PRICE in DS

> LEA BP, SS: STACK_TOP Load BP with offset of STACK_TOP in SS

> LEA CX, [BX] [DI] Load CX with EA = [BX] + [DI]

LDS - LDS Register, Memory address of the first word

This instruction loads new values into the specified register and into the DS register from four successive memory locations. The words from two memory locations is copied into the specified register and the word from the next two memory locations is copied into the DS registers. LDS does not affect any flag.

> LDS BX, [4326] : Copy content of memory at displacement 4326H in DS to BL content of 4327H to BH. Copy content at displacement of 4328H and 4329H in DS to DS register.

> LDS SI, SPTR Copy content of memory at displacement SPTR and SPTR +1 in DS to SI register. Copy content of memory at displacements SPTR+2 and SPTR +3 in DS to Ds register. DS: SI now points at ~~start~~ start of the desired string.

LES-LES Register, Memory address of the first word

This instruction loads new values into the specified register and into the ES register from four successive memory locations. The word from the first two memory locations in copied into the specified register, and the word from the ~~next~~ next two memory

locations in copied into the ES register.
LES does not affect any flag.

> LES BX, [789AH] Copy content of memory at displacement 789AH in DS to BL. content of 789BH to BH content memory at displacement 789CH and 789DH in DS is copied to ES register

> LES DI, [BX] Copy content of memory at offset [BX] and offset [BX] +1 in DS to DI register. Copy content of memory at offset [BX] +2 and [BX] +3 to ES register.

ARITHMETIC INSTRUCTIONS

ADD – ADD Destination, Source

ADC – ADC Destination, Source

These instructions add a number from some source to a number in some destination and the result in the specified destination. The ADC also adds the status of the carry flag to the result. The source may be an immediate number, a register, or a memory location. The destination may be a register or ~~a m~~ a memory location. The source and the ~~destina~~ destination must be of the same type (bytes or words). If you want to add a byte to a word, you must copy the byte to a word location and fill the upper byte of the word

with 0's before adding. Flags affected: AF, CF, OF, SF, ZF.

> ADD AL, 74H Add immediate number 74H to content of AL. Result in AL

> ADC CL, BL Add content of BL plus carry status to content of CL

> ADD DX, BX Add content of BX to content of DX.

> ADD DX, [SI] Add word from memory at offset [SI] in DS to content of DX.

> ADC AL, PRICES [BX] Add byte from effective address. PRICES [BX] plus carry status to content

~~of A~~

of AL

> ADD AL, PRICES [BX] Add content of memory at effective address PRICES [BX] to AL