

1

Name - Tazanaab Kabir, section - 7

ID - 1620404042

MOV-MOV Destination, Source

The MOV instruction copies a word or byte of data from a specified source to a specified destination.

The destination can be a register or a memory location. The 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). A MOV instruction

2.

does not affect any flag.

→ MOV CX, 037AH , Put immediate number 037AH to CX

→ MOV BL,[437AH] , Copy byte in DS at offset 437H to BL

→ MOV AX,BX , Copy content at register BX to AX

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

→ MOV DS,BX , Copy word from BX to DS register

3.

→ MOV RESULT[BP], AX, Copy AX to
two memory locations; AL to the
first memory location is sum of
the displacement represented by
RESULTS and content of BP.
Physical address = EA + SS.

→ MOV ES:RESULTS[BP], AX, same
as the above instruction, but physical
address = EA + ES, because of the
segment override prefixes

4.

LEA - LEA Register, Source:

This instruction determines the offset of the variable or memory location named as the source and puts this offset in the indicated 16-bit register. LEA 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]

5.

ADD - ADD - Pdestination, Source;

ADC - ADC Destination, Source.

This instructions add a number from some Source to a number in some ~~desti~~ destination may and put the result in the specified destination. The Ape also adds the status of the carry flag to the result. The Source may be an immediate number or register, or a memory location. If you want to add a byte to a word, you must copy the byte to a word

6.

location and fill the upper byte of the word with 0's before adding. Flags affected: AF, CF, OF, SF, ZF.

→ ADD AL, 74H

→ ADC CL, BL

→ ADD DX, BX

→ ADD DX, [BX]

→ ADC AL, PRICES [BX]

→ ADD AL, PRICES [BX]

SUB - SUB Destination, Source;

SBB - SBB Destination, Source;

These instructions subtract

7.

the number in some source from the number in some destination and put the result in the destination. If you want to subtract a byte from a word, you must first move the byte to a word location such as a 16-bit register and tell the upper byte of the word with 0s.

Flags affected: AF, CF, OF, PF, SF, ZF

→ SUB CX, BX

→ SBB CH, AL

→ SUB AX, 3427H

→ SBB BX, [3427H]

8.

→ SUB PRIPOSE[BX], 04H

→ \$ SBB CX, TABLE[BX]

→ SBB TABLE[BX], CX

MUL - MUL Source:

This instruction multiplies an unsigned byte in AL register or an unsigned word in some source with an unsigned word in AX register. It the most significant byte of a 16-bit result or the most significant word of a 32-bit result is 0, CF and OF will both be 0's. AF, PF, SF and ZF are undefined after a MUL.

instruction.

→ MUL BH

→ MUL CX

→ MUL BYTE PTR [BX]

→ MUL FACTOR [BX]

DIV-DIV Source;

This instruction is used to divide an unsigned double word (32 bits) by a word. If an attempt is made to divide by 0 or if the quotient is too large to fit in the destination (greater than FFH (FFFF)) the 8086 will generate a type 0 interrupt. All flags are

undeborred after a DIV instruction.

→ DIV BL

→ DIV BX

→ DIV SCALE[BX]

INC-INC Destination:

The INC instruction adds 1 to a specified register or to a memory location. AF, OF, PF, SF and ZF are updated, but CF is not affected.

This means that if an 8-bit destination containing FFH or a 16-bit destination containing FFFFH is incremented, the result

will be all 0's with no carry.

→ INC BL

→ INC CL

→ INC BYTE PTR [BX]

→ INC WORD PTR [BX]

→ INC TEMP

→ INC PRICEST[BX]

DEE-DEC Destination:

This instruction subtracts 1 from the destination word or byte.

The destination can be a register or a memory location. AF, OF, SF, PF and ZF are updated, but CF is not affected.

→ DEC CL

→ DEC BP

→ DEC BYTE PTR [BX]

→ DEC WORD PTR [BP]

→ DEC COUNT

DAA (DECIMAL ADJUST AFTER BED ADDITION):

This instruction is used to make sure the result of adding two packed Bed numbers is adjusted to be a legal Bed number.

The result of the addition must be in AL for DAA to work correctly.

→ Let AL = 59, BL and BL = 35 BCD

ADD AL, BL

DAA

→ Let AL = 88 BCD and BL = 49 BCD

ADD AL, BL

DAA

~~DAS (DECIMAL ADJUST AFTER
BCD SUBTRACTION)~~

AAA (ASCII ADJUSTER FOR ADDITION)

Numerical data coming into a computer from a terminal is usually in ASCII code. In this code, the numbers 0 to 9 are represented by the ASCII codes 30H to 39H.

→ Let $AL = 00110101$ (ASCII 5)
~~ADD~~ and $BL = 00111001$ (ASCII 9)
 $AL = 01101110$ (6EH, which is incorrect ~~BD~~)

$AL = 00000100$ (unpacked ~~BD~~ 4)
 $CF = 1$ indicates answer is 14 decimal.

ADD AL, BL

AAA

AND-AND Destination, Source

This instruction ANDs each bit in a source byte or word with the same numbered bit in a destination byte or word.

The result is put into the specified destination. PF and OF are both 0 after AND, SF and ZF are updated by the AND instruction. AF is undefined.

→ AND CX, [SI]

→ AND BH, CL

→ AND BX, 0FFFH

OR - OR Destination, Source

This instruction ORs each bit in a source byte or word with the same numbered bit in a destination byte or word. CF and OF are both 0 after OR. PF, SF and ZF are updated by the OR instruction. AF is undefined.

- OR AH, CL
- OR BP, SI
- OR SI, BP
- OR BL, 80H
- OR EX, TABLE[BI]

XOR - XOR Destination, Source

This instruction Exclusive-ORs each bit in a source byte or word with the same numbered bit in a destination byte or word. CF and OF are both 0 after XOR. PF, SF and ZF are updated. PF has meaning only from an 8-bit operand. AF is undefined.

→ XOR CL,BH

→ XOR BP, DI

→ XOR WORD PTR[BX], 00FFH

EMP-CMP Destination, Source:

This instruction compares a byte/word in the specified source with a byte/word in the specified destination. AF, OF, SF, ZF, PF and CF are updated by the cmp instruction.

	CF	ZF	SF
CX = BX	0	1	0
CX > BX	0	0	0
CX < BX	1	0	0

→ CMP AL, 01H

→ CMP BH, 0CL

→ CMP CX, TEMP

→ CMP PRICES[BX], 49H

TEST - TEST Destination, Source

This is an instruction ANDs the byte word in the specified source with the byte/word in the specified destination. PF, SF, and ZF will be updated to show the results at the destination. AF is undefined.

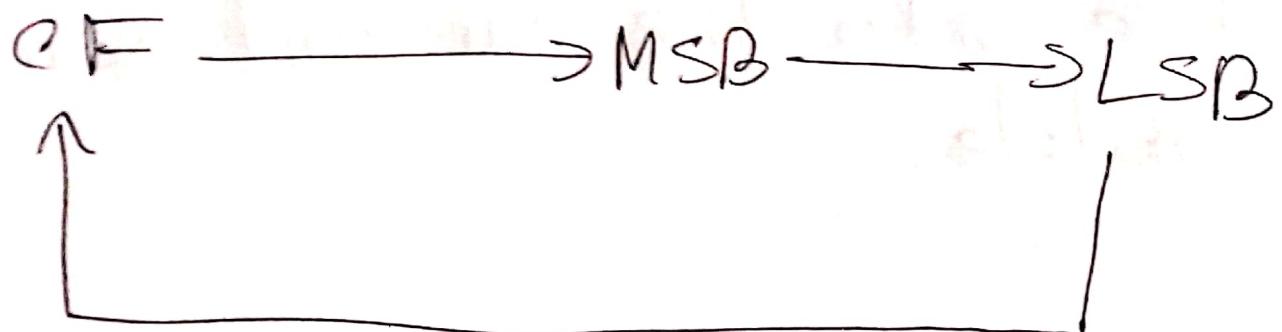
→ TEST AL, BH

→ TEST CX, 0001H

→ TEST BP, [BX][DI]

RCR - RCR Destination, Count;

this instruction rotates all the bits in a specified word or byte some number of bit position to the right.



→ RCR BX, 1

→ MOV CL, 4

RCR BYTE PTR [BX], 4

INCLUDE (INCLUDE SOURCE
CODE FROM FILE);

This directive is used to tell the assembler to insert a block of source code from the named file into the current source module.