3.2.2 Arithmetic Instructions

The 8086 MP has a group of arithmetic instructions that perform different types of arithmetic operations. The arithmetic operations such as Addition (**ADD**, **ADC**, and **INC**) instructions, Subtraction (**SUB**, **SBB**, **DEC**, and **NEG**) instructions, Multiplication (**MUL** and **IMUL**) instructions, and Division (**DIV** and **IDIV**) instructions. The status flags (CF, AC, SF, PF, ZF, OF) are affected by the arithmetic operations and changed depending on the results of the operations.

3.2.2.1 Addition instructions

i. ADD, ADC instructions

The **ADD** instruction is used to perform the addition between two operands (i.e. between the Destination and Source) operands. It simply adds the content of the source to the content of the destination and stores the result into the destination operand. The status flags that will be affected by the ADD instruction are CF, AC, SF, PF, ZF, OF.

Examples:

ADD AL, BL ; $AL_{new}=AL_{old}+BL$.

ADD SI, [3000h] ; $SI_{new}=SI_{old}$+the content of the M.L. with offset of 3000h.

ADD CX, 5000h ; $CX_{new}=CX_{old}+5000h$

ADD BYTEPTR[BP+5020h], 3Eh ; the content (8-bit) of M.L. with offset of (BP+5020h)+3Eh.

The **ADC** (ADD with carry) instruction has the same operation of the ADD instruction but will add the content of the **old CF** to the result. The status flags that will be affected by the ADD instruction are CF, AC, SF, PF, ZF, OF.

Examples:

ADC SI, DI ; $SI_{new}=SI_{old}$+DI+old CF

ADC DX, [BX] ; $DX_{new}=DX_{old}$+ the content of the M.L. with offset of BX+old CF

ADC WORDPTR[2000h], 6600h ; the content of M.L. (16-bit) with offset of (2000h)+6600h+ old CF.

Important note: in case of addition with *immediate addressing mode*, it is acceptable to add the (8-bit) number to the (16-bit) contents

Example:

ADD CX, 50h

ADC SI, FFh

Mohammed Ali Saffah

ii. **INC** (increment) instruction is considered as one of the addition arithmetic instructions. It simply adds ***one*** to the content of the register or the content of the memory locations. It takes ***One operand only that represents the Source and Destination*** in the same time. It also affects the status flags AC, SF, PF, ZF, OF except the CF.

Examples:

INC BYTEPTR[BX+SI+5000h] ; the content of the M.L. (8-bit) with offset of (BX+SI+5000h)+1

INC CX ; $CX_{new}=CX_{old}+1$

Important note: the **INC** instruction does not take the *immediate addressing mode*.

Table below shows the operations of the addition instructions.

Mnemonics	Meaning	Format	Operation	Flags affected
ADD	Addition	ADD D,S	D+S ⟶ D	CF, AC, SF, PF, ZF, OF
ADC	ADD with Carry	ADC D,S	D+S+old CF ⟶ D	CF, AC, SF, PF, ZF, OF
INC	ADD 1 to the content	INC D	D+1 ⟶ D	AC, SF, PF, ZF, OF

Note: in the above table D means Destination and S means Source.

Example 2: Write an Assembly Language Program (A.L.P.) to add with carry two consecutive bytes (8-bit) of data stored in data segment of start address of 7000h and an offset specified by [BX+SI]. Store the result in register BL.

Solution:

```
MOV AX, 7000h
MOV DS, AX
MOV BL, 00h
ADC BL, [BX+SI]
ADC BL, [BX+SI+01h]
HLT
```

Mohammed Ali Saffah

H.W. 1: Write an Assembly Language Program (A.L.P.) to add two consecutive words (16-bit) of data stored in data segment of start address of 2000h and an offset specified by the PA [27000h]. Store the result in the next 2 M.L.s followed the 2 consecutive words.

3.2.2.2 Subtraction instructions

i. SUB, SBB instructions

The **SUB** instruction is used to perform the subtraction between two operands (i.e. between the Destination and Source) operands. It simply subtracts the content of the source from the content of the destination and stores the result into the destination operand. The status flags that will be affected by the SUB instruction are CF, AC, SF, PF, ZF, OF.

Examples:

SUB AL, BL ; $AL_{new}=AL_{old}$ – BL ($AL_{new}=AL_{old}$ + 2's comp. of BL)

SUB SI, [3000h] ; $SI_{new}=SI_{old}$ – the content of the M.L. with offset of 3000h (SI_{old} +2's comp. of the content of the M.L. with offset of 3000h)

SUB CX, 5000h ; $CX_{new}=CX_{old}$ – 5000h (CX_{old} + the 2's comp. of the number 5000h).

SUB BYTEPTR[BP+5020h], 3Eh ; the content (8-bit) of M.L. with offset of (BP+5020h) – 3Eh(the content (8-bit) of M.L. with offset of (BP+5020h) + 2's comp. of the number 3Eh)

The **SBB** (SUB with Borrow) instruction has the same operation of the SUB instruction but will subtract the content of the **old CF(borrow)** from the result. The status flags that will be affected by the SBB instruction are CF, AC, SF, PF, ZF, OF.

Examples:

SBB SI, DI ; $SI_{new}=SI_{old}$–DI– old CF (SI_{old} + 2's comp. of (DI+old CF))

SBB DX, [BX] ; $DX_{new}=DX_{old}$ – the content of the M.L. with offset of BX– old CF (DX_{old} + the 2's comp. of (the content of the M.L. with offset of BX+ old CF))

SBB WORDPTR[2000h], 6600h ; the content of M.L. (16-bit) with offset of (2000h) – 6600h+ old CF(the content of M.L. (16-bit) with offset of (2000h) + the 2's comp. of (6600h+ old CF))

Important note: in case of subtraction with *immediate addressing mode*, it is acceptable to subtract the (8-bit) number to the (16-bit) contents

Mohammed Ali Saffah

Examples:

SUB CX, 50h

SBB SI, FFh

ii. **DEC** (decrement) instruction is considered as one of the subtraction arithmetic instructions. It simply subtracts ***one*** from the content of the register or the content of the memory locations. It takes ***One operand only that represents the Source and Destination*** in the same time. It also affects the status flags AC, SF, PF, ZF, OF except the CF.

Examples:

DEC BYTEPTR[BX+SI+5000h] ; the content of the M.L. (8-bit) with offset of (BX+SI+5000h) – 1

DEC CX ; $CX_{new}=CX_{old}-1$

Important note: the **DEC** instruction does not take the *immediate addressing mode*.

iii. **NEG** (negate or negative) instruction ***inverts the sign of the content*** of either the M.L. or of a register. It actually takes the ***2's complement of the content***.

Examples:

NEG AX ; $AX_{new}= - AX_{old}$ (2's comp. of AX).

NEG WORDPTR[BP+DI] ; – the content (16-bit) of the M.L. with offset of (BP+DI) (2's comp. of the content (16-bit) of the M.L. with offset of (BP+DI)).

Important note: the **NEG** instruction does not take the *immediate addressing mode*.

Table below shows the operations of the subtraction instructions.

Mnemonics	Meaning	Format	Operation	Flags affected
SUB	Subtraction	SUB D,S	D – S ⟶ D	CF, AC, SF, PF, ZF, OF
SBB	SUB with Carry	SBB D,S	D – S – old CF ⟶ D	CF, AC, SF, PF, ZF, OF
DEC	SUB 1 from the content	DEC D	D – 1 ⟶ D	AC, SF, PF, ZF, OF
NEG	Negate	NEG D	2's comp. (D)	CF, AC, SF, PF, ZF, OF

Mohammed Ali Saffah

Example 3: Trace the following program step by step showing the values of the (CF, ACF, PF, ZF, SF) assuming all initial values of the flags are ZEROS:

MOV AX, 1623h

MOV BX, F014h

MOV CX, 2002h

ADD AX, BX

DEC BX

SBB BX,CX

NEG CX

HLT

Solution:

STEP	AX	BX	CX	CF	ACF	PF	SF	ZF
1	1623	0000	0000	0	0	0	0	0
2	1623	F014	0000	0	0	0	0	0
3	1623	F014	2002	0	0	0	0	0
4	0637	F014	2002					
5	0637	F013	2002					
6	0637	D010	2002					
7	0637	D010	DFFE					

H.W. 2: Trace the following program step by step showing the values of the (CF, ACF, PF, ZF, SF) assuming all initial values of the flags are ZEROS:

MOV BX, 1234h

MOV DX, 2500h

INC CX

NEG CX

ADC DX, CX

ADD CX,32h

HLT

H.W. 3: Write an A.L.P to perform the following operations:

M1= M2+5

M2=M1–12

Mohammed Ali Saffah

M3=M2+25

M4= –M3

Where M1, M2, M3, and M4 are memory locations that have an offsets of 0300h, 0400h, 0500h, and 0600h respectively.

Mohammed Ali Saffah