2. Arithmetic Instructions:

ADD	ADC	SUB	SBB	MUL	IMUL	DIV	IDIV	CMP	NEGATE
INC	DEC	DAA	DAS	AAA	AAS	AAM	AAD	CBW	CWD

These instructions usually perform the arithmetic operations, like addition, subtraction, multiplication and division along with the respective ASCII and decimal adjust instructions. The increment and decrement operations also belong to this type of instructions. The arithmetic instructions affect all the conditional code flags. The operands are either the **registers or memory** locations immediate data depending upon the addressing mode.

ADD: Addition: This instruction adds an immediate data or contents of a memory location specified in the instruction or a register (source) to the contents of another register (destination) or memory location. The result is in the destination operand. However, both the source and destination operands **cannot** be memory operands. That means memory to memory addition is not possible. Also, the contents of the segment registers **cannot** be added using this instruction. All the condition code **flags** are affected depending upon the result.

Syntax:

i. ADD mem/reg1, mem/reg2 اجمع القيمة الثانية مع الأولى واخزن الناتج في الاولى
[mem/reg1] ⟵ [mem/reg2] + [mem/reg2]

Ex: ADD BL, [ST]
ADD AX, BX

ii. ADD mem, data تم التعامل مع الذاكرة

[mem] ⟵ [mem]+data

Ex: ADD Start, 02H start هو عنوان في الذاكرة

ADD [SI], 0712H

iii. ADD reg, data تم إضافة رقم قيمة ثابتة الى المسجل

[reg] ⟵ [reg]+data

Ex: ADD CL, 05H

ADD DX, 0132H

iv. ADD A, data جمع رقم مع المسجل

[A] ⟵ [A]+data

Ex: ADD AL, 02H

ADD AX, 1211H

Examples with addressing modes:

1. ADD AX, 0100H Immediate
2. ADD AX, BX Register
3. ADD AX, [SI] Register Indirect
4. ADD AX, [5000H] Direct
5. ADD [5000H], 0100H Immediate
6. ADD 0100H Destination AX (implicit)

ADC:

Add with carry:

This instruction performs the same operation as ADD instruction, but adds the carry flag bit (which may be set as a result of the previous calculations) to the result. All the condition code flags are affected by this instruction.

Syntax:

ADC mem/reg1, mem/reg2

[mem/reg1] ⟵ [mem/reg1]+[mem/reg2]+CY

Ex: ADC BL, [SI] ADC AX, BX

ADC mem, data [mem] ⟵[mem]+data+CY

Ex: ADC start, 02H

ADC [SI],0712H

iii. ADC reg, data

[reg] ⟵[reg]+data+CY

Ex: ADC AL, 02H

ADC AX, 1211H

Examples with addressing modes:

1. ADC 0100H Immediate (AX implicit)
2. ADC AX, BX Register
3. ADC AX, [SI] Register indirect
4. ADC AX, [5000H] Direct
5. ADC [5000H],0100H Immediate

SHR: -

SHR AL, 4

☆ Meaning

SHR AL, 4 means:

Shift the bits in register AL to the right by 4 positions.

What actually happens?

- AL is an 8-bit register.
- SHR = Shift Right (logical shift).
- When shifting right:
- Every bit moves one step to the right.
- The bits that fall off on the right go into the Carry Flag (CF).
- The left side is filled with zeros.

EX:-

AL=1111 0000 (يعني F0H)

عند التنفيذ

SHR AL,4 (تزاح 4 بتات نحو اليمين

0000 1111 قبل الازاحة

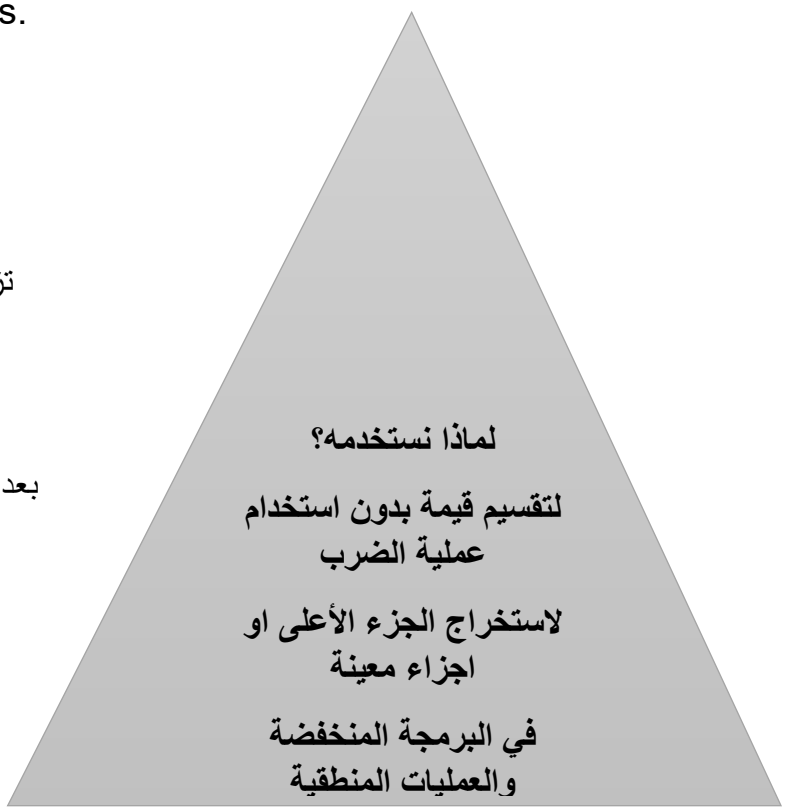
1000 0111 بعد 1 خطوة

1100 0011 بعد 2خطوة

1110 0001 بعد3 خطوة

1111 0000 بعد4 خطوة

AL=0000 1111 =0FH



AND:

AL=1111 0000

AND =0000 0000

SHR	AND
إزاحة البتات نحو اليمين	عملية منطقية لا تحرك البتات
كل البتات تتحرك بمقدار 4	تحذف (تصفر)البتات العليا فقط
البتات التي تسقط تذهب الى CARRY FLAG	تحتفظ بالبتات السفلى كما هي
تشبه القسمة على 16	لا تشبه القسمة والضرب

Sub:

Subtract: The subtract instruction subtracts the source operand from the destination operand and the result is left in the destination operand. Source operand may be a register or a memory location, but source and destination operands both must not be memory operands. Destination operand cannot be an immediate data.

Ex: SUB BL, [SI]

SUB AX, BX

BL=BL- القيمة الموجودة في الذاكرة عند العنوان المشار اليه

Ex: SUB AL, 02H

SUB AX, 1211H

AL=AL-02H

AX=AX-1211H

INC:

Increment: This instruction increments the contents of the specified register or memory location by 1. All the condition flags are affected except the carry flag CF. This instruction adds a to the content of the operand. Immediate data cannot be operand of this instruction.

EX:

INC X

X=X+1

DEC: Decrement: The decrement instruction subtracts 1 from the contents of the specified register or memory location. All the condition code flags except carry flag are affected depending upon the result. Immediate data cannot be operand of the instruction.

Ex: DEC BX

BX=BX-1

MUL:

Unsigned multiplication Byte or Word: This instruction multiplies unsigned byte or word by the content of AL. The unsigned byte or word may be in any one of the general-purpose register or memory locations. The most significant word of result is stored in DX, while the least significant word of the result is stored in AX. All the flags are modified depending upon the result. Immediate operand is not allowed in this instruction. If the most significant byte or word of the result is '0' IF and OF both will be set.

8 bit*8 bit=16 bit توضع النتيجة في AX

EX:

MUL BL

AX=AL*BL

16 bit*16 bit=32 bit

Ex:

MUL BX توضع الناتج في DX:AX

DX:AX=AX*BX

Asmaa.dheyaa@uomus.edu.iq

DIV:

Unsigned division: This instruction performs unsigned division. It divides an unsigned word or double word by a 16-bit or 8-bit operand. The dividend must be in AX for 16-bit operation and divisor may be specified using any one of the addressing modes except immediate. The result will be in AL (quotient) while AH will contain the remainder. If the result is too big to fit in AL, type 0(divide by zero) interrupt is generated. In case of a double word dividend (32-bit), the higher word should be in DX and lower word should be in AX. The divisor may be specified as already explained. The quotient and the remainder, in this case, will be in AX and DX respectively. This instruction does not affect any flag.

EX:

DIV BL

بمعنى

AL=AX ÷ BL

AH=AX MOD BL

في عملية القسمة نحتاج الى المقسوم-المقسوم عليه-الناتج-الباقي

EX:

AX=0022H

BL=05H

AL=6, AH=4

34/5=6

الباقي=4