

INSTRUCCIONES DEL 8086

(T-2)

INSTRUCCIONES, TRANSFERENCIA DE DATOS.–

• **Mov.**

Transferencia de datos (1 BYTE O 1 PALABRA) entre fuente y destino. Donde destino es el lugar a donde se moverán los datos y fuente es el lugar donde se encuentran dichos datos.

MOV Destino, fuente

- Destino puede ser un registro, elemento de memoria (1 byte o palabra).
- Fuente: 1 registro, un elemento de memoria o un valor inmediato.

El contenido de fuente después del movimiento queda inalterado. No afecta a las banderas.

Importante:

- No se pueden mover datos entre dos elementos de memoria, hay que utilizar un registro intermedio.

Ej:

MOV AX, Pepe

MOV Pepe,AX

- No se puede mover un valor inmediato a u registro de segmento, pero si utilizando un registro intermedio.

Ej:

MOV AX,112

MOV ES,AX

- No se puede utilizar el registro CS como destino.

Tipos permitidos para esta instrucción son:

DESTINO	FUENTE	EJEMPLO
Memoria	Acumulador	MOV mem_byte,AL
Acumulador	Memoria	MOV AL,mem_byte
Segmento	Memoria	MOV DE,mem_byte
Segmento	Registro	
Memoria	Segmento	
Registro	Registro	

Registro	Memoria	MOV AX,2B35h
Memoria	Registro	
Registro	Valor inmediato	
Memoria	Valor inmediato	

• LEA.

Transfiere el desplazamiento del operando fuente al operando destino. El operando fuente debe ser un operando de memoria (byte o palabra). El operando destino es un registro de 16 bits, pero no un registro de segmento.

LEA destino,fuente

Ej:

MOV SI,OFFSET VAR1 Equivale a LEA SI,VAR1

DESTINO	FUENTE	EJEMPLO
Registro 16 Bits	Memoria	LEA AX,VAR2

• PUSH.

Apila el operando siempre a nivel de palabra.

Carga el operando en la dirección apuntada por SS:SP, que es la cima de la pila y decrementa en 2 unidades a SP. No afecta a los flags y no se puede especificar como operando al registro CS.

PUSH fuente

Tipos permitidos para esta instrucción son:

FUENTE	EJEMPLO
Memoria	PUSH [BX+DI+1000h]
Registro 16 bits	PUSH BX

•POP.

Instrucción complementaria a PUSH, realiza el trabajo inverso, es decir, transfiere la palabra situada en la cima de la pila, apuntada por SS:SP, al operando (de 2 bytes).

No se puede especificar como operando al registro CS.

POP destino

Tipos permitidos para esta instrucción son:

DESTINO	EJEMPLO
Memoria	POP [BX+DI+1000h]
Registro 16 bits	POP BX

INSTRUCCIONES ARITMETICAS.–

• ADC.

Adición con acarreo. Suma los operandos más 1 si está activada la bandera de acarreo (CF), El resultado lo guarda en el operando destino. Los operandos deben ser del mismo tipo (byte o palabra).

Afecta a los flags: OF, SF, ZF, AF, PF, CF

ADC destino, fuente

Lleva a cabo la suma de dos operandos y suma una al resultado en caso de que la bandera CF esté activada, esto es, en caso de que exista acarreo.

El resultado se guarda en el operando destino.

DESTINO	FUENTE	EJEMPLO
Registro	Registro	ADC BL, CL
Registro	Memoria	ADC BL, mem_byte
Memoria	Registro	
Acumulador	Valor	ADC AL, 12h
Registro	Valor	ADC BX, 1234h
Memoria	Valor	

• ADD.

Adición de los operandos. El resultado lo guarda en el operando destino. Los operandos deben ser del mismo tipo (byte, palabra).

Afecta a los flags: OF, SF, ZF, AF, PF, CF

ADD destino, fuente

DESTINO	FUENTE	EJEMPLO
Registro	Registro	ADD BX, CX
Registro	Memoria	ADD BL, mem_byte
Memoria	Registro	ADD mem_byte, BL
Acumulador	Valor	ADD AL, 12H
		ADD AX, 1234H
Registro	Valor	ADD BX, 11H
Memoria	Valor	ADD mem_byte, 60h

• SUB.

Resta el operando fuente del operando destino y el resultado lo guarda en el operando destino. Los operandos deben ser del mismo tipo (byte, palabra).

Afecta a los flags: OF, SF, ZF, AF, PF, CF

SUB destino,fuente

DESTINO	FUENTE	EJEMPLO
Registro	Registro	ADD BX,CX
Registro	Memoria	ADD BL,mem_byte
Memoria	Registro	ADD mem_byte,BL
Acumulador	Valor	ADD AL,12H
		ADD AX,1234H
Registro	Valor	ADD BX,1234
Memoria	Valor	ADD mem_byte,1234h

•SUB.

Resta el operando fuente del operando destino y el resultado lo guarda en el operando destino. Los operandos deben ser del mismo tipo (byte, palabra).

Afecta a los flags: OF,SF, ZF, AF, PF, CF

SUB destino,fuente

DESTINO	FUENTE	EJEMPLO
Registro	Registro	SUB BX,CX
Registro	Memoria	SUB BL,mem_byte
Memoria	Registro	SUB mem_byte,BL
Acumulador	Valor	SUB AL,12H
		SUB AX,1234H
Registro	Valor	SUB BX,1234H
Memoria	Valor	SUB mem_byte,1234H

• SBB.

Resta el operando fuente del operando destino teniendo en cuenta el acarreo, y el resultado lo guarda en el operando destino. Destino=destino-(fuente+CF).

Los operandos deben ser del mismo tipo (byte, palabra).

Afecta a los flags: OF, SF, ZF, AF, PF, CF

SBB destino,fuente

DESTINO	FUENTE	EJEMPLO
Registro	Registro	SBB BX,CX
Registro	Memoria	SBB BL,mem_byte
Memoria	Registro	SBB mem_byte,BL
Acumulador	Valor	SBB AL,12H

		SBB AX,1234H
Registro	Valor	SBB BX,1234H
Memoria	Valor	ABB mem_byte,1234H

• INC.

Incrementa el operando. La instrucción suma uno al operando destino y guarda el resultado en el mismo operando destino.

Afecta a los flags: OF, SF, ZF, AF, PF

INC destino

DESTINO	EJEMPLO
Registro 8 Bits	INC AL
Registro 16 bits	INC AX
Memoria	INC mem_byte

• DEC.

Decrementa el operando. La instrucción resta uno al operando destino y guarda el resultado en el mismo operando destino.

Afecta a los flags: OF, SF, ZF, AF, PF

DEC destino

DESTINO	EJEMPLO
Registro 8 bits	DEC AL
Registro 16 bits	DEC AX
Memoria	DEC mem_byte

• NEG.

Genera el complemento a 2. Calcula el valor negativo del operando. Resta el operando de 0 y devuelve el resultado en el mismo operando. Para hacer esto el operando se resta del número compuesto por todos 1 y se le añade 1

NEG destino=NOT destino

INC destino

DESTINO	EJEMPLO
Registro 8 bits	NEG AL
Registro 16 bits	NEG AX
Memoria	NEG mem_byte

• CMP.

Compara dos operandos. Resta fuente de destino, pero no almacena el resultado. Los operandos quedan intactos pero las banderas se actualizan pudiéndose consultar a continuación mediante una instrucción de bifurcación condicional.

Los operandos pueden ser del tipo byte o palabra pero siempre del mismo tipo los dos.

Afecta a los flags: OF, SF, ZF, AF, PF, CF

CMP destino, fuente

DESTINO	FUENTE
Registro	Registro
Registro	Memoria
Memoria	Registro
Acumulador	Valor
Registro	Valor
Memoria	Valor

Si ZF=1 los operandos son iguales.

Si CF=0 con operandos sin signo operando1 >= operando 2, no ha habido acarreo.

Si CF=1 con operandos sin signo operando 1 < operando 2, ha habido acarreo.

Si OF=SF con operandos con signo operando 1 >= operando 2, desbordamiento del bit de signo.

Si OF<>SF con operandos con signo operando 1 < operando 2.

INSTRUCCIONES LÓGICAS.–

• **AND.**

Operación lógica and a nivel de bit entre los dos operandos. El resultado se almacena en el destino.

Afecta a los flags: CF y OF se ponen siempre a 0.

SF, ZF, AF, PF según la operación toman un valor u otro.

AND destino,fuente

DESTINO	FUENTE	EJEMPLO
Registro	Registro	CMP BX,CX
		CMP BL,CL
Registro	Memoria	CMP BL,mem_byte
Memoria	Registro	CMP mem_byte,CL
Acumulador	Valor	CMP AL,01H
Registro	Valor	CMP BH,1BH
		CMP BX,1234H

Memoria	Valor	CMP mem_byte,123BH
---------	-------	--------------------

• NOT.

Lleva a cabo la negación bit por bit del operando destino. El resultado se guarda en el mismo operando destino. Cambia los bits del operando por su complementario. No afecta a los flags.

NOT destino

DESTINO	EJEMPLO
Registro 8 bits	NOT AL
Registro 16 bits	NOT AL
Memoria	NOT mem_byte

• OR.

OR inclusivo lógico, la instrucción OR lleva a cabo, bit por bit, la disyunción inclusiva lógica de los dos operandos. El resultado lo almacena en el destino.

Afecta a los flags CF y OF que se ponen a 0.

SF, ZF, AF, PF.

OR destino,fuente

DESTINO	FUENTE	EJEMPLO
Registro	Registro	OR AX,CX
Registro	Memoria	OR BL,mem_byte
Memoria	Registro	OR mem_byte,BL
Acumulador	Valor	OR AX,0120H
Registro	Valor	OR BL,30H
Memoria	Valor	OR mem_byte,12H

• XOR.

OR exclusivo lógico. La instrucción OR lleva a cabo, bit por bit, la disyunción exclusiva lógica de los operandos. El resultado lo almacena en el destino.

Afecta a los flags CF y OF, que se ponen siempre a 0.

SF, ZF, AF, PF

XOR destino,fuente

DESTINO	FUENTE	EJEMPLO
Registro	Registro	XOR AX,CX
Registro	Memoria	XOR BL,mem_byte
Memoria	Registro	XOR mem_byte,BL
Acumulador	Valor	XOR AX,0120H

Registro	Valor	XOR BL,30H
Memoria	Valor	XOR mem_byte,12H

INSTRUCCIONES DESPLAZAMIENTO Y ROTACION.–

• SAR.

Desplazamiento aritmético a la derecha. Desplaza a la derecha los bits del operando destino, el n° de bits especificado en el segundo operando o contador. Los bits de la izquierda se rellenan directamente. Si es mayor que 1 su valor debe cargarse en CL y especificar CL como 2° operando.

Afecta a los flags: OF, SF, ZF, PF, CF

SAR destino, contador

DESTINO	FUENTE	EJEMPLO
Registro	Registro	SAR BL,CL
		SAR BX,CX
Registro	Valor	SAR BL,1
Memoria	Registro	SAR VAR1,CL
Memoria	Valor	SAR VAR2,1

• SHR.

Desplazamiento aritmético a la derecha. Desplaza a la derecha los bits del operando destino, el n° de bits especificado en el segundo operando o contador. Los bits de la izquierda se rellenan con 1. Si el n° de bits a desplazar es 1 se puede especificar directamente. Si es mayor que 1 su valor debe cargarse en CL y especificar CL como 2° operando.

Afecta a los flags OF, SF, ZF, PF, CF

SHR destino,contador

DESTINO	FUENTE	EJEMPLO
Registro	Registro	SHR BL,CL
		SHR BX,CX
Registro	Valor	SHR BL,1
Memoria	Registro	SHR VAR1,CL
Memoria	Valor	SHR VAR2,1

• SAL–SHL.

Realizan la misma instrucción y son la misma operación. Desplazamiento aritmético a la izquierda. Desplaza a la izquierda los bits del operando destino, el n° de bits especificado en el segundo operando o contador. Los bits de la derecha se rellenan con 0. Si el n° de bits a desplazar es 1 se puede especificar directamente. Si es mayor que 1 su valor debe cargarse en CL y especificar CL como 2° operando.

SAL destino,contador

SHL destino,contador

DESTINO	FUENTE	EJEMPLO
Registro	Registro	SAL-SHL BL,CL
		SAL-SHL BX,CX
Registro	Valor	SAL-SHL BL,1
Memoria	Registro	SAL-SHL VAR1,CL
Memoria	Valor	SAL-SHL VAR2,1

CONTROL DEL PROCESADOR.–

• **NOP.**

No hace nada.

• **CLC.**

Pone a 0 el bit de acarreo (CF=0).

• **STC.**

Pone a 1 el bit de acarreo (CF=1).

INSTRUCCIONES DE CONTROL DE FLUJO.–

• **Salto incondicional.**

JMP. Salto incondicional. Transfiere el control del programa a la dirección del operando.

JMP etiqueta

• **Llamada subrutina.**

Transfiere el control a la subrutina especificada en operando. La rutina invocada con CALL debe terminar con una sentencia RET.

CALL subrutina

Fases que efectúa:

- Apila la dirección de retorno.
- Ejecuta la rutina.
- Desapila la dirección de retorno.
- Devuelve al control de programa.

• **Llamada a una interrupción.**

Se genera una interrupción por software. Esto nos permitirá usar unas determinadas funciones o rutinas del BIOS y del DOS.

Su forma de trabajar es muy semejante a la llamada a una subrutina.

INT tipo_int

Fases que efectúa:

- Apila los flags.
- IF=0 y TF=0, inhibe las interrupciones.
- Apila la dirección de retorno.
- Accede a la tabla de interrupciones según el tipo de interrupción
- Desapila la dirección de retorno.
- Devuelve al control al programa.

Veremos 4 servicios de la interrupcion 21h.

Servicio 01. Lee un carácter introducido por teclado introduciendo su código ASCII en el registro AL.

MOV AH,01

INT 21H

Servicio 07. Hace lo mismo que el servicio 01 pero este carácter no aparece por pantalla.

MOV AH,07

INT 21H

Servicio 02. Muestra un carácter por pantalla. Para ello debemos introducir en un registro de datos como puede ser DL, el código ASCII de dicho carácter.

MOV AH,02

MOV DL,41H

INT 21H

Servicio 09. Escribe una cadena de caracteres por pantalla. Esta cadena de caracteres ha de definirse en el segmento de datos en una variable.

MOV AH,09

MOV DX,OFFSET VAR1

INT 21H

INSTRUCCIONES CONDICIONALES.–

• **Jxx.**

Salta si se cumple la condición. La distancia de la etiqueta está limitada por –128 y 127 bytes.

Jxx desplazamiento

El desplazamiento (valor con signo de 8 bits) debe estar comprendido entre –128 y 127 bytes desde la instrucción, que es el valor de desplazamiento máximo con signo que se puede representar con 8 bits.

Transfiere el control a la instrucción IP+Desplazamiento si se cumple la condición especificada por la instrucción.

No afecta a las banderas.

• Tratamiento de datos sin signo.

En esas instrucciones no se considera el signo, por lo que coge el menor independientemente del signo que tenga. No afecta a las banderas.

INSTRUCCION	DESCRIPCION	CONDICION
JB	Salta si inferior	CF=1
JNAE	Salta si no superior o igual	CF=1
JC	Salta si hay carry	CF=1
JBE	Salta si inferior o igual	CF=1 ó ZF=1
JNA	Saltar si no superior	CF=1 ó ZF=1
JE	Salta si es igual	ZF=1
JZ	Salta si es 0	ZF=1
JNC	Salta si no hay carry	CF=0
JA	Salta si superior o igual	CF=0
JNB	Salta si no inferior	CF=0
JA	Salta si superior	CF=0 y ZF=0
JNE	Salta si no igual	ZF=0
JNZ	Salta si no cero	ZF=0
JCXZ	Salta si CX es 0	CX=0

• Tratamiento de datos con signo.

INSTRUCCION	DESCRIPCION	CONDICION
JL	Salta si es menor	SF<>OF
JNGE	Salta si no es mayor ni igual	SF<>OF
JLE	Salta si menor o igual	ZF=1 ó SF<>OF
JNG	Salta si no mayor	ZF=1 ó SF<>OF
JG	Salta si mayor	ZF=0 ó SF<>OF
JNLE	Salta si no menor o igual	ZF=0 ó SF<>OF
JGE	Salta si mayor o igual	SF=OF
JNL	Salta si no menor	SF=OF

• Otros.

INSTRUCCIÓN	DESCRIPCIÓN	CONDICIÓN
JP	Salta si paridad	PF=1
JPE	Salta si paridad par	PF=1

JNP	Salta si no paridad	PF=0
JPO	Salta si paridad impar	PF=0
JS	Salta si signo	SF=1
JNS	Salta si no signo	SF=0

TABLA ASCII

dec		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
------------	--	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------	-----------	-----------	-----------	-----------