REPERTORIO DE INSTRUCCIONES DEL MICROPROCESADOR 8086

- ## INSTRUCCIONES DE TRANSFERENCIA:

NINGUNA INSTRUCCIÓN DE TRANSFERENCIA AFECTA AL REGISTRO DE ESTADO (SALVO LAS EXPLÍCITAS CON EL REGISTRO DE ESTADO)

A) GENÉRICAS	**MOV**
B) DE CADENAS	**LODSB, LODSW, STOSB, STOSW, MOVSB, MOVSW**
C) DE PILA	**PUSH, POP, PUSHF, POPF**
D) DE INTERCAMBIO	**XCHG**
E) DE ACCESO A TABLA	**XLAT**
F) DE ENTRADA O SALIDA	**IN, OUT**
G) DE DIRECCIONES LÓGICAS	**LEA, LDS, LES**
H) CON REGISTRO DE ESTADO	**LAHF, SAHF**

INSTRUCCIONES DE TRANSFERENCIA (I)

A) GENÉRICAS:

Mnemotécnico: MOV (MOVe)

Sintaxis: MOV destino, origen

Operación: destino ← origen

POSIBLES COMBINACIONES:

DESTINO	←	ORIGEN	EJEMPLOS
REGISTRO	←	DATO INMEDIATO	MOV AX, 23759 MOV CL, 7Fh
MEMORIA	←	DATO INMEDIATO	MOV byte ptr [BX], 0Ch MOV VARIABLE_1, 345
REGISTRO	←	MEMORIA	MOV BL, TABLA.[SI] MOV CX, [BX+SI–4]
MEMORIA	←	REGISTRO	MOV 2000[BX], DX MOV [BX+2000], DX
REGISTRO (*)	←	REGISTRO	MOV DS, AX MOV BP, SP

(*) PARA EL CASO REGISTRO ← REGISTRO, NO ESTÁ PERMITIDO QUE AMBOS SEAN REGISTROS DE SEGMENTO

UTILIDAD:

TOMAR DATOS PARA UBICARLOS EN EL LUGAR ADECUADO ANTES DE PROCESARLOS, Y PARA GUARDAR EL RESULTADO EN EL LUGAR OPORTUNO.

INSTRUCCIONES DE TRANSFERENCIA (II)

B) DE CADENAS (I):

UTILIDAD:

ACCEDER DE MANERA SECUENCIAL A LOS DATOS O ELEMENTOS DE UNA CADENA, LISTA O VECTOR SIN TENER QUE ACTUALIZAR DE MANERA EXPLÍCITA, POR PROGRAMA, EL HIPOTÉTICO PUNTERO UTILIZADO PARA EL ACCESO

INSTRUCCIONES PARA TOMAR DATOS DE UN VECTOR

Mnemotécnico: LODSB (LoaD String Byte)

Sintaxis: LODSB

Operación: AL ← DS:[SI]
SI ← SI+K (K= 1 si bandera D=0
K=-1 si bandera D=1)

Mnemotécnico: LODSW (LoaD String Word)

Sintaxis: LODSW

Operación: AX ← DS:[SI]
SI ← SI+K (K= 2 si bandera D=0
K=-2 si bandera D=1)

INSTRUCCIONES DE TRANSFERENCIA (III)

B) DE CADENAS (II):

INSTRUCCIONES PARA GUARDAR DATOS EN UN VECTOR

Mnemotécnico: STOSB **(STOre String Byte)**

Sintaxis: **STOSB**

Operación: **DS:[DI] ← AL**
DI ← DI+K **(K= 1 si bandera D=0**
K=-1 si bandera D=1)

Mnemotécnico: STOSW **(STOre String Word)**

Sintaxis: **STOSW**

Operación: **DS:[DI] ← AX**
DI ← DI+K **(K= 2 si bandera D=0)**
(K=-2 si bandera D=1)

INSTRUCCIONES PARA MOVIMIENTO DE BLOQUES

Mnemotécnico: MOVSB **(MOVe String Byte)**

Sintaxis: **MOVSB**

Operación: **bye ptr ES:[DI] ← byte ptr DS:[SI]**
SI ← SI+K **(K= 1 si bandera D=0**
DI ← DI+K **K=-1 si bandera D=1)**

Mnemotécnico: MOVSW **(MOVe String Byte)**

Sintaxis: **MOVSW**

Operación: **word ptr ES:[DI] ← word ptr DS:[SI]**
SI ← SI+K **(K= 1 si bandera D=0**
DI ← DI+K **K=-1 si bandera D=1)**

INSTRUCCIONES DE TRANSFERENCIA (IV)

C) DE PILA (I):

Mnemotécnico: PUSH

Sintaxis: **PUSH origen**

Operación: **SP ← SP-2**
SS:[SP] ← palabra origen

El operando origen puede ser una palabra inmediata, un registro interno tipo palabra (incluso de segmento) o una palabra en memoria

Utilidad:
- **Guardar en la pila datos para salvarlos temporalmente y así liberar recursos que son precisos para otras tareas.**
- **Pasar parámetros a un procedimiento.**

Mnemotécnico: POP

Sintaxis: **POP destino**

Operación: **destino palabra ← SS:[SP]**
SP ← SP+2

El operando destino puede ser un registro interno tipo palabra (incluso de segmento) o una palabra en memoria

Utilidad:
- **Recuperar datos previamente salvados en la pila**

INSTRUCCIONES DE TRANSFERENCIA (V)

C) DE PILA (II):

Mnemotécnico: PUSHF (PUSH Flags)

Sintaxis: PUSHF

Operación: SP ← SP-2
SS:[SP] ← registro de estado

Utilidad:
- **Salvaguardar el estado del procesador para poder restaurarlo posteriormente**

Mnemotécnico: POPF (POP Flags)

Sintaxis: POPF

Operación: Registro de estado ← SS:[SP]
SP ← SP+2

Utilidad:
- **Restaurar el estado del procesador o forzar un cierto estado (es la única manera de actuar sobre el registro de estado en su conjunto)**

INSTRUCCIONES DE TRANSFERENCIA (VI)

D) DE INTERCAMBIO:

Mnemotécnico: XCHG (eXCHanGe)

Sintaxis: XCHG operando_1, operando_2

Operación: operando_1 ↔ operando_2

E) DE ACCESO A TABLA:

Mnemotécnico: XLAT (transLATe)

Sintaxis: XLAT

Operación: AL ← DS:[BX+AL]

Utilidad:
Permite resolver problemas mediante la técnica de acceso a tabla (*look-up tables)*. BX ha de apuntar al inicio de la tabla con las soluciones, y AL es el dato original utilizado como índice de acceso

INSTRUCCIONES DE TRANSFERENCIA (VII)

F) DE ENTRADA O SALIDA

Mnemotécnico: IN

Sintaxis: **IN acumulador, puerto**

Operación: **acumulador ← puerto**

“Acumulador” es o AL (entrada de octeto) o AX (entrada de palabra).

“Puerto” es o un valor directo de tamaño octeto (dirección de puerto desde el 0 hasta el 255) o DX. En este último caso DX hace las veces de un puntero al puerto utilizado para la entrada (puertos desde el 0 hasta el 65535).

Utilidad:
Permite realizar operaciones de entrada de datos desde la unidad de entradas/salidas (lectura de los registros de los dispositivos periféricos: de entrada de datos o de estado del periférico)

Mnemotécnico: OUT

Sintaxis: **OUT puerto, acumulador**

Operación: **puerto ← acumulador**

“Acumulador” y “Puerto” es lo mismo que en IN.

Utilidad:
Permite realizar operaciones de salida de datos hacia la unidad de entradas/salidas (escritura de los registros de los dispositivos periféricos: de salida de datos o de control del periférico)

INSTRUCCIONES DE TRANSFERENCIA (VIII)

G) DE DIRECCIONES LÓGICAS (SÓLO PARTE DE DESPLAZAMIENTO)

Mnemotécnico: LEA (Load Efective Address)

Sintaxis: LEA destino_reg16, origen

Operación: reg16 ← palabra formada a partir de origen

El operando origen sigue el patrón de cualquier direccionamiento de la memoria, pero no se usa para direccionar la memoria sino para especificar una palabra que es la dirección efectiva objeto de transferencia.

Utilidad:
Se usa para iniciar registros punteros (aunque también de cualquier otro tipo) de una manera flexible.

Ejemplo:
LEA SI, [BX+8]

Si BX=1000 entonces el resultado es SI← 1008

INSTRUCCIONES DE TRANSFERENCIA (IX)

G) DE DIRECCIONES LÓGICAS COMPLETAS

Mnemotécnico: LDS (Load Data Segment)

Sintaxis: LDS destino_reg16, fuente_mem

Operación: reg16 ← palabra en dir. efec. fuente_mem
DS ← palabra en dir. efec. fuente_mem+2

Fuente_mem es un direccionamiento de la memoria. A partir de la dirección efectiva de la memoria se encuentra una doble palabra que es una dirección lógica completa (desplazamiento y segmento, en este orden). La parte de desplazamiento se mete en el registro destino tipo palabra, y la parte de segmento se mete en el registro DS.

Utilidad: rápida iniciación de un puntero y el registro segmento de datos, para así poder trabajar con datos

Ejemplo:
LDS SI, [BX]

SI ← palabra en memoria apuntada por BX
DS ← palabra en memoria apuntada por BX+2
Esa doble palabra en memoria es una dirección lógica completa.

Mnemotécnico: LES (Load Extra Segment)

Sintaxis: LES destino_reg16, fuente_mem

Operación: reg16 ← palabra en dir. efec. fuente_mem
ES ← palabra en dir. efec. fuente_mem+2

Utilidad:
Similar a LDS, pero para trabajar en el espacio de datos extra.

INSTRUCCIONES DE TRANSFERENCIA (X)

H) CON EL REGISTRO DE ESTADO (OCTETO BAJO)

Mnemotécnico: LAHF (Load AH with Flags)

Sintaxis: LAHF

Operación: AH ← octeto bajo del registro de estado

La parte baja del registro de estado es la que tiene las banderas de acarreo (bit 0), paridad (bit 2), acarreo auxiliar (bit 4), resultado cero (bit 6) y signo (bit 7).

Mnemotécnico: SAHF (Save AH in Flags)

Sintaxis: SAHF

Operación: octeto bajo del registro de estado ← AH

Utilidad:
Aparte de la que pueda tener para la manipulación de varias banderas, el origen de estas instrucciones está en el deseo de facilitar la migración del software 8085 a 8086.

- **INSTRUCCIONES LÓGICAS:**

A) SUMA LÓGICA: OR

LA UTILIDAD ES LA PUESTA A "1" LÓGICO DE BITS SELECCIONADOS DE UN OPERANDO

B) PRODUCTO LÓGICO: AND

LA UTILIDAD ES LA PUESTA A "0" LÓGICO DE BITS SELECCIONADOS DE UN OPERANDO, ASÍ COMO UN PASO PREVIO PARA LA EVALUACIÓN DE BITS INDIVIDUALES DE UN OPERANDO

C) SUMA LÓGICA EXCLUSIVA: XOR

LA UTILIDAD ES LA INVERSIÓN LÓGICA DE BITS SELECCIONADOS DE UN OPERANDO

D) NEGACIÓN: NOT

LA UTILIDAD ES LA NEGACIÓN DE UN OPERANDO

E) COMPROBACIÓN DE BITS TEST

LA UTILIDAD ES, COMO CON "AND", PERMITIR LA EVALUACIÓN DE BITS INDIVIDUALES DE UN OPERANDO, PERO SIN QUE ÉSTE SE VEA AFECTADO

Sintaxis: NOT operando (no afecta a las banderas)
Sintaxis: OR/AND/XOR/TEST destino, origen

Operación de OR, AND, XOR y TEST:

La operación lógica bit a bit entre los del operando destino y los del origen (usado como máscara), quedando el resultado en el operando destino, salvo en el caso de la instrucción TEST, en que el resultado se pierde.

El registro de estado, siempre, refleja las cualidades del resultado (bits S, Z y P, mientras que O←0, C←0, A←?).

- **INSTRUCCIONES ARITMÉTICAS:**

A) SUMA SIN Y CON ACARREO: ADD / ADC

B) RESTA SIN Y CON PRÉSTAMO: SUB / SBB

C) INCREMENTO Y DECREMENTO: INC / DEC

D) NEGATIVACIÓN: NEG

E) PRODUCTO SIN Y CON SIGNO: MUL / IMUL

F) DIVISIÓN SIN Y CON SIGNO: DIV / IDIV

G) AJUSTE BCD:

PARA LA SUMA (BCD empaqueta.): DAA
(Decimal Adjust for Addition)
PARA LA RESTA (ídem): DAS
(Decimal Adjust for Subtraction)
PARA LA SUMA (BCD desempaq.): AAA
(Ascii Adjust for Addition)
PARA LA RESTA (ídem): AAS
(Ascii Adjust for Subtraction)
PARA LA MULTIPLICACIÓN (ídem): AAM
(Ascii Adjust for Multiplication)
PARA LA DIVISIÓN (ídem): AAD
(Ascii Adjust for Division)

H) EXTENSIÓN DE SIGNO:

DE OCTETO A PALABRA: CBW
(Convert Byte in Word)

DE PALABRA A DOBLE PALABRA: CWD
(Convert Word in Double word)

INSTRUCCIONES ARITMÉTICAS (II)

ADD	**destino, origen**	**(destino←destino+origen)**
ADC	**destino, origen**	**(destino←destino+origen+C)**
SUB	**destino, origen**	**(destino←destino – origen)**
SBB	**destino, origen**	**(destino←destino – origen – C)**
INC	**operando**	**(operando←operando+1)**
DEC	**operando**	**(operando←operando – 1)**
MUL	**operando_8**	**(AX←AL*operando_8, sin signo)**
MUL	**operando_16**	**(DXAX←AX*operando_16, ídem)**
IMUL	**operando_8**	**(AX←AL*operando_8, con signo)**
IMUL	**operando_16**	**(DXAX←AX*operando_16, ídem)**
DIV	**operando_8**	**(AL←AX÷operando_8, sin signo)** **(AH← resto de la división)**
DIV	**operando_16**	**(AX←AX÷operando_16, ídem)** **(DX← resto de la división)**
IDIV	**operando_8**	**(AX←AL÷operando_8, con signo)** **(AH← resto de la división)**
IDIV	**operando_16**	**(DXAX←AX÷operando_16, ídem)** **(DX← resto de la división)**
CBW		**(AH← extensión del signo de AL)**
CWD		**(DX← extensión del signo de AX)**

DAA, DAS, AAA, AAS y AAM trabajan con el dato en AL
AAD trabaja con los datos BCD desempaquetados en AH y AL

• INSTRUCCIONES DE COMPARACIÓN:

LAS INSTRUCCIONES DE COMPARACIÓN SON INSTRUCCIONES DE RESTA ARITMÉTICA, AUNQUE EL RESULTADO DE LA RESTA SE PIERDE QUEDANDO INAFECTADO EL OPERANDO DESTINO.
LAS BANDERAS DEL REGISTRO DE ESTADO INFORMAN SOBRE LAS CUALIDADES DEL RESULTADO, PUDIÉNDOSE UTILIZARLAS COMO CONDICIÓN EN LAS INSTRUCCIONES DE BIFURCACIÓN CONDICIONADA.

A) COMPARACIÓN NORMAL CMP

Sintaxis: CMP operando_1, operando_2

Operación: operando_1 – operando_2

B) COMPARACIÓN DE CADENAS SCAS y CMPS

Sintaxis: SCASB

Operación: AL – byte ptr ES:[DI] y DI←DI ± 1

Sintaxis: SCASW

Operación: AX – word ptr ES:[DI] y DI←DI ± 2

Sintaxis: CMPSB

Operación: byte ptr DS:[SI] – byte ptr ES:[DI] y DI←DI ± 1

Sintaxis: CMPSW

Operación: word ptr DS:[SI] –word ptr ES:[DI] y DI←DI ± 2

Utilidad de SCAS: encontrar el inicio de una cadena en otra

Utilidad de CMPS: comparar dos cadenas

- **INSTRUCCIONES DE ROTACIÓN Y DE DESPLAZAMIENTO:**

A) ROTACIÓN

EXISTEN VARIOS ASPECTOS A CONSIDERAR:

- **SENTIDO DE LA ROTACIÓN:**
  - **IZQUIERDA: ROL y RCL**
  - **DERECHA: ROR y RCR**
- **PAPEL DE LA BANDERA DE ACARRERO, C:**
  - **TESTIGO DEL ÚLTIMO BIT SALIENTE: ROL y ROR**
  - **PARTE DEL OPERANDO A ROTAR: RCL y RCR**

B) DESPLAZAMIENTO

EXISTEN VARIOS ASPECTOS A CONSIDERAR:

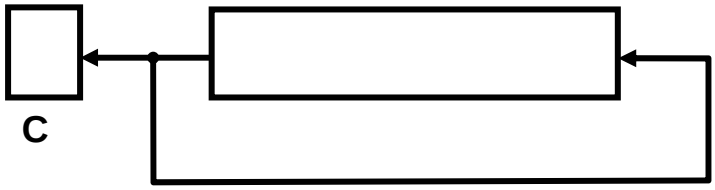
- **SENTIDO DEL DESPLAZAMIENTO:**
  - **IZQUIERDA: SHL**
  - **DERECHA: SHR y SAR**
- **CONSERVACIÓN O NO DEL BIT DE SIGNO:**
  - **DESPLAZAMIENTO ARITMÉTICO: SAR**
  - **DESPLAZAMIENTO LÓGICO: SHL y SHR**

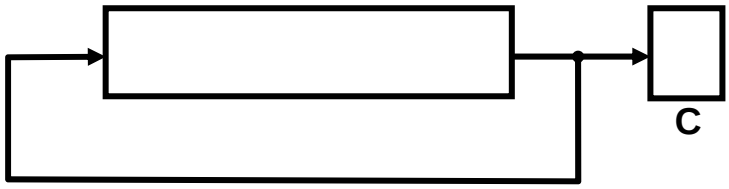
INSTRUCCIONES DE ROTACIÓN (II)

Sintaxis: **MNEMOTÉCNICO operando, 1**
MNEMOTÉCNICO operando, CL

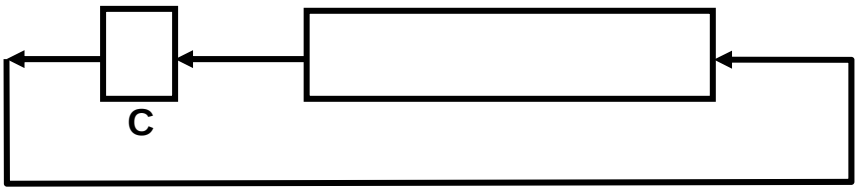
Si se utiliza CL como indicador del factor de rotación, CL no se ve afectado una vez hecha la operación.
La bandera de acarreo, C, siempre actúa como testigo del último bit rotado

A) ROTACIÓN SIMPLE: ROL y ROR (ROtate Left, ROtate Right)





B) ROTACIÓN A TRAVÉS DEL ACARREO: RCL y RCR (Rotate through Carry Left, Rotate through Carry Right)

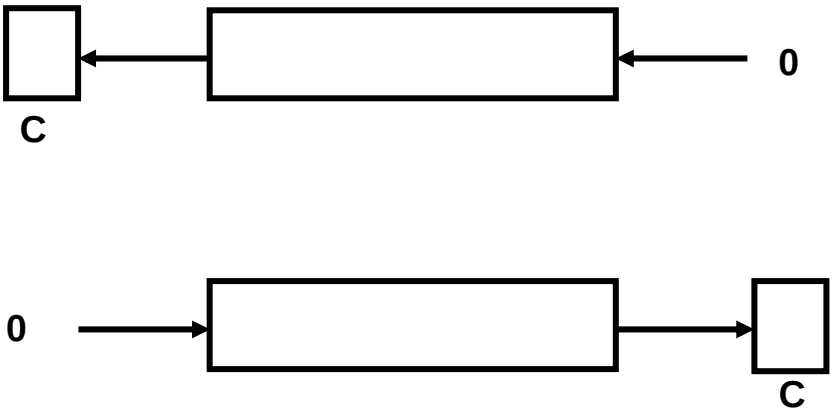


INSTRUCCIONES DE DESPLAZAMIENTO (III)

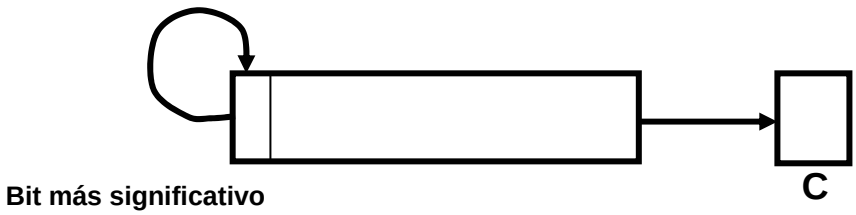
Sintaxis: **MNEMOTÉCNICO operando, 1**
MNEMOTÉCNICO operando, CL

Si se utiliza CL como indicador del factor de desplazamiento, CL no se ve afectado una vez hecha la operación.
La bandera de acarreo, C, siempre actúa como testigo del último bit desplazado

C) DESPLAZAMIENTO LÓGICO: SHL y SHR (SHift Left, SHift Right)



D) DESPLAZAMIENTO ARITMÉTICO: SAR (Shift Arithmetic Right)



- **INSTRUCCIONES DE CONTROL DEL FLUJO DE PROGRAMA:**

A) SALTO INCONDICIONAL

JMP (JuMP)

B) BIFURCACIÓN CONDICIONADA

Jcondicion

C) LLAMADA Y RETORNO DE PROCEDIMIENTO

CALL
RET (RETurn)
RETF (RETurn Far)

D) PETICIÓN Y RETORNO DE INTERRUPCIÓN

INT (INTerrupt)
IRET (Interrupt RETurn)

E) CONTROL DE UN BUCLE

GENÉRICO: LOOP, LOOPE y LOOPNE
PREFIJOS DE REPETICIÓN:
REP, REPE y REPNE

A) SALTO INCONDICIONAL:

Sintaxis: JMP dirección

La "dirección" puede expresarse con cualquiera de los modos de direccionamiento de datos, pero teniendo en cuenta que lo que se expresa es la dirección efectiva:

- **Dirección directa**
- **Dirección por registro**
- **Dirección indirecta**
- **etcétera**

INSTRUCCIONES DE CONTROL DE FLUJO DE PROGRAMA (II)

B) BIFURCACIÓN CONDICIONADA

- **ESTAS INSTRUCCIONES SÓLO ADMITEN SALTOS RELATIVOS AL CONTADOR DEPROGRAMA**
- **EL RANGO DEL SALTO ESTÁ RESTRINGIDO A +127 Y –128 POSICIONES DE MEMORIA CON RESPECTO A LA DEL SALTO (A PARTIR DEL 80386 SON POSIBLES SALTOS RELATIVOS DE MAGNITUD 16 BITS CON SIGNO)**

Sintaxis: MNEMOTÉCNICO dirección_del_salto

SALTOS PRECISOS TRAS EVALUAR DOS MAGNITUDES:

CONDICIÓN	DATOS SIN SIGNO	DATOS CON SIGNO
MAYOR	JA JNBE	JG JNLE
MAYOR O IGUAL	JAE JNB	JGE JNL
IGUAL	JE	JE
DISTINTO	JNE	JNE
MENOR O IGUAL	JBE JNA	JLE JNG
MENOR	JB JNAE	JL JNGE

(*) COMO PUEDE OBSERVARSE, HAY INSTRUCCIONES QUE PUEDEN RECIBIR DISTINTO NOMBRE MNEMOTÉCNICO SEGÚN EL PUNTO DE VISTA QUE SE UTILICE. POR EJEMPLO, “JA” Y “JNBE”

INSTRUCCIÓN	Operación: saltar si...	Condición
JO	si hay rebosamiento	O=1
JNO	si no hay rebosamiento	O=0
JB JC JNAE	si es menor (sin signo) si hay acarreo si no es mayor ni igual (sin signo)	C=1
JAE JNB	si es mayor o igual (sin signo) si no es menor (sin signo)	C=0
JE JZ	si es igual si el resultado es cero	Z=1
JNE JNZ	si no es igual si el resultado no es cero	Z=0
JBE JNA	si es menor o igual (sin signo) si no es mayor (sin signo)	ó C=1 ó Z=1
JA JNBE	si es mayor (sin signo) si no es menor ni igual (sin signo)	C=1 y Z=0
JS	si es negativo	S=1
JNS	si es positivo	S=0
JP JPE	si P está activa si la paridad es par	P=1
JNP JPO	si P no está activa si la paridad es impar	P=0
JL JNGE	si es menor (con signo) si no es mayor ni igual (con signo)	S=1 y O=1
JGE JNL	si es mayor o igual (con signo) si no es menor (con signo)	S=O
JLE JNG	si es menor o igual (con signo) si no es mayor (con signo)	ó Z=1 ó (S=O=1)
JG JNLE	si es mayor (con signo) si no es menor ni igual (con signo)	ó Z=0 ó S=0
JCXZ	Si CX vale cero	CX=0

INSTRUCCIÓN	Significado del mnemotécnico	Condición
JO	Jump if Overflow	O=1
JNO	Jump if Not Overtflow	O=0
JB JC JNAE	Jump if Below Jump if Carry Jump if Not Above or Equal	C=1
JAE JNB	Jump if Above or Equal Jump if Not Below	C=0
JE JZ	Jump if Equal Jump if Zero	Z=1
JNE JNZ	Jump if Not Equal Jump if Not Zero	Z=0
JBE JNA	Jump if Below ot Equala Jump if Not Above	ó C=1 ó Z=1
JA JNBE	Jump if Above Jump if Not Below or Equal	C=1 y Z=0
JS	Jump if Sign	S=1
JNS	Jump if Not Sign	S=0
JP JPE	Jump if Parity Jump if Parity Even	P=1
JNP JPO	Jump if Not Parity Jump if Parity Odd	P=0
JL JNGE	Jump if Less Jump if Not Greater or Equal	S=1 y O=1
JGE JNL	Jump if Greater or Equal Jump if Not Less	S=O
JLE JNG	Jump if Less or Equal Jump if Not Greater	ó Z=1 ó (S=O=1)
JG JNLE	Jump if Greater Jump if Not Less or Equal	ó Z=0 ó S=0
JCXZ	Jump if CX is Zero	CX=0

INSTRUCCIONES DE CONTROL DE FLUJO DE PROGRAMA (II)

E) CONTROL DE BUCLES

- **BUCLES GENÉRICOS**

LOOP dirección

Operación: CX←CX-1
Si CX≠0 entonces bifurcar a *dirección*

LOOPE / LOOPZ dirección (LOOP if Equal / Zero)

Operación: CX←CX-1
Si CX≠0 y Z=1 entonces bifurcar a *dirección*

LOOPNE / LOOPNZ dirección (LOOP if Not Equal / Zero)

Operación: CX←CX-1
Si CX≠0 y Z=0 entonces bifurcar a *dirección*

- **BUCLES MONOINSTRUCCIÓN (DE CADENAS) (PREFIJOS DE REPETICIÓN)**

REP (REPeat)

Operación: CX←CX-1
Si CX≠0 entonces repetir instrucción MOVS o STOS

REPE / REPZ (REPeat if Equal / Zero)

Operación: CX←CX-1
Si CX≠0 y Z=1 repetir instrucción CMPS o SCAS

REPNE / REPNZ (REPeat if Not Equal/Zero)

Operación: CX←CX-1
Si CX≠0 y Z=0 repetir instrucción CMPS o SCAS

- **INSTRUCCIONES DE CONTROL DEL SISTEMA**

ESTAS INSTRUCCIONES CARECEN DE OPERANDO EXPLÍCITO

INSTRUCCIÓN	OPERACIÓN	EFECTO
CLC	Desactivar el acarreo	C←0
STC	Activar el acarreo	C←1
CMC	Complementar el acarreo	C←!C
CLI	Inhibir las interrupciones	I←0
STI	Habilitar las interrupciones	I←1
CLD	Borrar la dirección (incremento)	D←0
STD	Activar la dirección (decremento)	D←1
NOP	No hacer nada	-------
ESC	Prefijo para instrucciones de un coprocesador (matemático, etc). Esa instrucción será ignorada por la UCP	
HLT	Detención del procesador. Sólo se puede salir de este estado reiniciando el procesador	
LOCK	Activación de la señal LOCK. Útil en sistemas multiprocesadores	
WAIT	Espera hasta la activación de la señal TEST	

RESUMEN DEL REPERTORIO DE INSTRUCCIONES 8086

GRUPO	INSTRUCCIÓN	DESCRIPCIÓN
TRANSFERENCIA	MOV	Genérica
	MOVSB, MOVSW	Movimiento de cadenas
	LODSB, LODSW	Carga en acum. (cadenas)
	STOSB, STOSW	Guarda acum. (cadenas)
	PUSH, POP	De pila (meter y sacar)
	PUSHF, POPF	Id. (registro de estado)
	XCHG	Intercambio
	XLAT	De acceso a tabla
	SAHF, LAHF	De registro de estado
	LEA, LDS, LES	De direcciones
	IN, OUT	De entrada y salida
LÓGICAS	AND	Producto lógico
	OR	Suma lógica
	XOR	Suma lógica exclusiva
	NOT	Negación (inversión)
	TEST	Comprobación
ARITMÉTICAS	ADD, ADC	Suma (sin y con acarreo)
	SUB, SBB	Resta (sin y con préstamo)
	NEG	Negativación
	INC, DEC	Incremento y decremento
	MUL, IMUL	Multiplicación (s/c signo)
	DIV, IDIV	División (sin/con signo)
	DAA, DAS	Ajuste BCD (empaqueta.)
	AAA, AAS, AAM, AAD	Ídem (desempaquetado)
	CBW, CWD	Extensión de signo
COMPARACIÓN	CMP	Comparación
	SCASB, SCASW	Acumulador con cadena
	CMPSB, CMPSW	Dos cadenas
ROTACIÓN Y DESPLAZAMIENTO	ROR, ROL, RCR, RCL	Rotaciones varias
	SHR, SHL, SAR	Desplazamientos varios
CONTROL DE FLUJO	JMP	Salto incondicional
	Jcondición	Saltos condicionados
	CALL	Llamada a subrutina
	RET, RETF	Retorno de subrutina
	INT, IRET	Interrupción, retorno
	LOOP, LOOPE, LOOPNE	Gestión de bucles
	REP, REPE, REPNE	Ídem (cadenas)
CONTROL DEL SISTEMA	CLC, STC, CMC	Bandera de acarreo
	CLI, STI	Bandera de interrupción
	CLD, STD	Bandera de dirección
	NOP	No operación
	ESC	Escape
	HLT, LOCK, WAIT	Control hardware vario