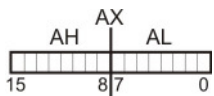


GUÍA DE REFERENCIA RÁPIDA DEL 8086

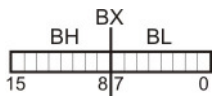
INSTRUCCIONES DE TRANSFERENCIA					
Nemon.	Operandos		Formato	Descripción	Flags
MOV	REG REG/MEM REG/MEM	REG/MEM REG NUM	MOV Dest, Fuente	COPIA EL OPERANDO FUENTE EN EL OPERANDO DESTINO. Los operandos deben ser del mismo tipo (BYTE O PALABRA). RESTRICCIONES: No se pueden mover datos entre dos elementos de memoria. No se puede usar para transferir un valor inmediato a un registro de segmento. Ante cualquiera de estos dos casos, si es necesario, se usará un registro auxiliar. El registro CS no puede ser usado como destino.	No altera
XCHG	REG/MEM	REG	XCHG Op1,Op2	INTERCAMBIA EL CONTENIDO DE LOS DOS OPERANDOS. Pueden ser byte o palabra, pero los DOS del mismo tamaño. RESTRICCIONES: No se puede intercambiar dos posiciones de memoria. Para ello, se usa un registro auxiliar. No pueden ser operandos registros de segmento	No altera
STC			STC	ACTIVA FLAG DE ACARREO. Pone un '1' en el flag de acarreo.	CF=1
CLC			CLC	BORRA FLAG DE ACARREO. Pone un '0' en el flag de acarreo.	CF=0
CMC			CMC	COMPLEMENTA FLAG DE ACARREO. Si el flag está a '1' lo pone a '0', y si está a '0' lo pone a '1'.	CF = \overline{CF}
STD			STD	ACTIVA FLAG DE DIRECCIÓN. Pone un '1' en el flag de dirección.	DF=1
CLD			CLD	BORRA FLAG DE DIRECCIÓN. Pone un '0' en el flag de dirección.	DF=0
STI			STI	ACTIVA FLAG DE INTERRUPCIÓN. Pone un '1' en el flag de interrupción.	IF=1
CLI			CLI	BORRA FLAG DE INTERRUPCIÓN. Pone un '0' en el flag de interrupción.	IF=0
PUSH	REG 16 / MEM 16		PUSH Fuente	METE EL OPERANDO EN LA PILA. Decrementa dos unidades el Puntero de Pila (SP) y después almacena el operando en la posición indicada por el puntero de pila (SS:SP). RESTRICCIONES: Solo se pueden guardar en la pila operandos tipo palabra, y antes hay que inicializar los registros SS y SP.	No altera
POP	REG 16 / MEM 16		POP Fuente	SACA UNA PALABRA DE LA PILA. Copia la palabra de la pila, direccionada por SS:SP al operando destino (tipo palabra), y después incrementa en dos unidades el registro SP. El contenido de la pila no se borra, sino que el puntero es modificado. RESTRICCIONES: El registro CS no puede ser destino.	No altera
IN	AL / AX	PORT / DX	IN Dest, Port IN Dest, DX	COPIA EL CONTENIDO DE UN PUERTO DE ENTRADA EN EL ACUMULADOR. El dato de tamaño byte o palabra, se almacena en AL o AX respectivamente. Si la dirección del puerto está entre 0 y 255, se especifica directamente en la instrucción. En general, para direcciones entre 0h y FFFFh, se carga en DX la dirección (variable).	No altera
OUT	PORT / DX	AL / AX	OUT Port, Fuente OUT DX, Fuente	TRANSFIERE UN BYTE O PALABRA (ubicado en AL o AX respect.) AL PUERTO DE SALIDA ESPECIFICADO. La dirección del puerto puede ser un valor fijo de un byte (de 0 a FFh), o un valor variable que puede ser modificado por el programa, almacenado en DX (de 0 a FFFFh).	No altera
LEA	REG 16	MEM	LEA Dest,Fuente	CARGA LA DIRECCIÓN EFECTIVA. Carga el desplazamiento (offset) de la dirección de memoria fuente en el registro de 16 bits indicado como destino. RESTRICCIONES: No puede usarse como destino ningún registro de segmento. NOTA: (LEA=dirección, MOV=contenido de la dirección)	No altera
INSTRUCCIONES ARITMÉTICAS					
Nemon.	Operandos		Formato	Descripción	Flags
ADD	REG REG/MEM REG/MEM	REG/MEM REG NUM	ADD Dest,Fuente	SUMA. Suma el operando fuente al operando destino, almacenando el resultado en el operando destino. Los operandos pueden ser de tipo byte o palabra, pero AMBOS DEL MISMO TIPO. RESTRICCIONES: No se permite la suma de dos posiciones de memoria.	OF, SF, ZF, AF, PF, CF
ADC	REG REG/MEM REG/MEM	REG/MEM REG NUM	ADC Dest,Fuente	SUMA CON ACARREO. Suma el operando fuente al operando destino, almacenando el resultado en el operando destino. Los operandos pueden ser de tipo byte o palabra, pero AMBOS DEL MISMO TIPO. Si el flag de CARRY (CF) está activado, suma uno al resultado. RESTRICCIONES: No se permite la suma de dos posiciones de memoria.	OF, SF, ZF, AF, PF, CF
SUB	REG REG/MEM REG/MEM	REG/MEM REG NUM	SUB Dest,Fuente	RESTA. Resta el operando fuente del operando destino, almacenando el resultado en el operando destino. Los operandos pueden ser de tipo byte o palabra, pero AMBOS DEL MISMO TIPO. RESTRICCIONES: No se permite la resta de dos posiciones de memoria.	OF, SF, ZF, AF, PF, CF
SBB	REG REG/MEM REG/MEM	REG/MEM REG NUM	SBB Dest,Fuente	RESTA LLEVANDO (CON BORROW). Resta el operando fuente del operando destino, almacenando el resultado en el operando destino. Los operandos pueden ser de tipo byte o palabra, pero AMBOS DEL MISMO TIPO. Si el flag de CARRY (CF) está activado, se resta 1 al resultado. RESTRICCIONES: No se permite la resta de dos posiciones de memoria.	
MUL	REG/MEM		MUL Operando	MULTIPlicACIÓN SIN SIGNO. Multiplica un n° sin signo tamaño byte por un n° sin signo contenido en AL guardando el resultado en AX (AX= AL * operando byte) o multiplica un n° sin signo tamaño word por un n° sin signo contenido en AX guardando el resultado en DX (palabra más significativa) y en AX (palabra menos significativa)(DX AX = AX * operando word).	OF y CF a '1' si la mitad superior del resultado (DX ó AH) no es 0
IMUL	REG/MEM		IMUL Operando	MULTIPlicACIÓN CON SIGNO. Multiplica un n° con signo tamaño byte por un n° con signo contenido en AL guardando el resultado en AX (AX= AL * operando byte) o multiplica un n° con signo tamaño word por un n° con signo contenido en AX guardando el resultado en DX (palabra más significativa) y en AX (palabra menos significativa)(DX AX = AX * operando word).	OF y CF a '1' si la mitad superior del resultado (DX ó AH) no es 0
DIV	REG/MEM		DIV Operando	DIVISIÓN SIN SIGNO. Divide el acumulador y su extensión (AH AL si el operando es de tipo byte, o DX AX si el operando es de tipo word) entre el operando fuente. Hay dos opciones: Dividir 16 bits entre 8 bits (Dividendo=AX, Divisor=operando 8bits, Cociente=AL, Resto=AH) o dividir 32 bits entre 16 bits (Dividendo=DX AX, Divisor=operando 16bits, Cociente=AX, Resto=DX). Genera una interrupción tipo 0 si el cociente supera FFh o FFFFh respectivamente.	Todos los flags quedan INDEFINIDOS.
IDIV	REG/MEM		IDIV Operando	DIVISIÓN CON SIGNO. Divide el acumulador y su extensión (AH AL si el operando es de tipo byte, o DX AX si el operando es de tipo word) entre el operando fuente. Hay dos posibilidades: Dividir 16 bits entre 8 bits (Dividendo=AX, Divisor=operando 8bits, Cociente=AL, Resto=AH) o dividir 32 bits entre 16 bits (Dividendo=DX AX, Divisor=operando 16bits, Cociente=AX, Resto=DX). Genera una interrupción tipo 0 si 7Fh<cociente<81h ó 7FFFh<cociente<8001h respectivamente.	Todos los flags INDEFINIDOS. Signo resto=signo dividendo
INC	REG/MEM		INC Operando	INCREMENTA EL DESTINO EN UNA UNIDAD. Suma uno al operando especificado que puede ser byte o palabra. Si el operando contiene FFFFh pasará a valer 0000h después de la ejecución de INC, y no se pondrá a '1' el flag CF	OF,SF,ZF,AF,PF (no afecta a CF)
DEC	REG/MEM		DEC Operando	DECREMENTA EL DESTINO EN UNA UNIDAD. Resta uno al operando especificado que puede ser byte o palabra.	OF,SF,ZF,AF,PF (no afecta a CF)
CMP	REG REG/MEM REG/MEM	REG/MEM REG NUM	CMP Dest,Fuente	COMPARA DOS OPERANDOS. Lo hace mediante la resta del operando fuente del destino. El resultado NO es almacenado, y sólo se actualiza el contenido de los flags. Los operandos pueden ser de tipo byte o palabra, pero AMBOS DEL MISMO TIPO. RESTRICCIONES: No se permite la comparación entre dos posiciones de memoria.	OF,DF,SF,ZF,AF,PF,CF
INSTRUCCIONES LÓGICAS					
Nemon.	Operandos		Formato	Descripción	Flags
NEG	REG/MEM		NEG Operando	FORMA EL COMPLEMENTO A 2. Invierte el signo del operando. Realiza la operación Operando = 0 – Operando.	OF,SF,ZF,AF,PF,CF
NOT	REG/MEM		NOT Operando	NO LÓGICO. Forma el complemento a uno del operando, esto es, cambia los ceros por unos y los unos por ceros.	No altera
AND	REG REG/MEM REG/MEM	REG/MEM REG NUM	AND Dest,Fuente	Y LÓGICO. Realiza la operación lógica AND entre los operandos fuente y destino, realizada bit a bit y almacenada en el destino. RESTRICCIONES: No se puede realizar la operación entre dos posiciones de memoria.	OF=0,SF,ZF,PF, CF=0 (AF indefinido)
OR	REG REG/MEM REG/MEM	REG/MEM REG NUM	OR Dest,Fuente	O LÓGICO. Realiza la operación OR entre los operandos fuente y destino, realizada bit a bit y almacenada en el destino. RESTRICCIONES: No se puede realizar la operación entre dos posiciones de memoria.	OF=0,SF,ZF,PF, CF=0 (AF indefinido)
XOR	REG REG/MEM REG/MEM	REG/MEM REG NUM	XOR Dest,Fuente	O LÓGICO EXCLUSIVO. Realiza la operación lógica OR exclusiva entre los operandos fuente y destino, realizada bit a bit y almacenada en el destino. RESTRICCIONES: No se puede realizar la operación entre dos posiciones de memoria.	OF=0,SF,ZF,PF, CF=0 (AF indefinido)
TEST	REG REG/MEM REG/MEM	REG/MEM REG NUM	TEST Dest,Fuente	AND LÓGICA. Realiza la operación lógica AND entre los operandos fuente y destino, realizada bit a bit pero SIN ALMACENAR el resultado en destino. Se actualizan los flags. RESTRICCIONES: No se puede realizar la operación entre dos posiciones de memoria.	OF=0,SF,ZF,PF, CF=0 (AF indefinido)
INSTR. DE DESPLAZAMIENTO Y ROTACIÓN					
Nemon.	Operandos		Formato	Descripción	Flags
SAL	REG/MEM REG/MEM	CL 1	SAL Dest,CL SAL Dest,1	DESPLAZAMIENTO ARITMÉTICO A LA IZQUIERDA. Desplaza hacia la izquierda los bits del operando destino el n° de posiciones indicado por el operando fuente, colocando ceros en el bit menos significativo. Es idéntica a SHL.	OF,SF,ZF,PF,CF
SAR	REG/MEM REG/MEM	CL 1	SAR Dest,CL SAR Dest,1	DESPLAZAMIENTO ARITMÉTICO A LA DERECHA. Desplaza hacia la derecha los bits del operando destino el n° de posiciones indicado por el operando fuente, colocando una copia del bit más significativo en el bit más significativo en cada desplazamiento para mantener el bit de signo inicial.	OF,SF,ZF,PF,CF
SHL	REG/MEM REG/MEM	CL 1	SHL Dest,CL SHL Dest,1	DESPLAZAMIENTO LÓGICO A LA IZQUIERDA. Desplaza hacia la izquierda los bits del operando destino el n° de posiciones indicado por el operando fuente, colocando ceros en el bit menos significativo. Es idéntica a SAL.	OF,SF,ZF,PF,CF
SHR	REG/MEM REG/MEM	CL 1	SHR Dest,CL SHR Dest,1	DESPLAZAMIENTO LÓGICO A LA DERECHA. Desplaza hacia la derecha los bits del operando destino el n° de posiciones indicado por el operando fuente, colocando ceros en el bit más significativo.	OF,SF,ZF,PF,CF
ROL	REG/MEM REG/MEM	CL 1	ROL Dest,CL ROL Dest,1	ROTAR A LA IZQUIERDA los bits del operando destino el n° de posiciones indicado por el operando fuente. CF copia en cada rotación el bit más significativo.	OF, CF
ROR	REG/MEM REG/MEM	CL 1	ROR Dest,CL ROR Dest,1	ROTAR A LA DERECHA los bits del operando destino el n° de posiciones indicado por el operando fuente. CF copia en cada rotación el bit más significativo.	OF, CF
RCL	REG/MEM REG/MEM	CL 1	RCL Dest,CL RCL Dest,1	ROTAR A LA IZQUIERDA USANDO EL ACARREO. Rota hacia la izquierda los bits del operando destino y el flag de acarreo, el n° de posiciones indicado por el operando fuente. En cada desplazamiento el bit más significativo se almacena en CF y el contenido de CF pasa a ser el bit menos significativo.	OF, CF
RCR	REG/MEM REG/MEM	CL 1	RCR Dest,CL RCR Dest,1	ROTAR A LA DERECHA USANDO EL ACARREO. Rota hacia la derecha los bits del operando destino y el flag de acarreo, el n° de posiciones indicado por el operando fuente. En cada desplazamiento el bit menos significativo se almacena en CF y el contenido de CF pasa a ser el bit más significativo.	OF, CF

INSTRUCCIONES DE CONTROL DE PROGRAMA			Descripción	Flags
Nemon.	Operandos	Formato		
CALL	Dirección	CALL proc CALL proc NEAR CALL proc FAR	LLAMADA A SUBROUTINA O PROCEDIMIENTO. Transfiere la ejecución del programa principal a una subrutina. Salva en la pila la dirección de la instrucción siguiente para continuar cuando termine la subrutina. Hay dos tipos de llamadas: NEAR y FAR. NEAR es para llamar a subrutinas en el mismo segmento de código (CALL procedimiento NEAR: decrementa SP en dos unidades y salva en la pila el IP correspondiente a la siguiente instrucción). FAR es para llamar a subrutinas en otro segmento (CALL procedimiento FAR: decrementa SP en dos unidades, salva en la pila el contenido del registro CS, decrementa SP otras dos unidades, y salva en la pila el IP de la siguiente instrucción (CS:IP que son 20 bits), y por último coloca en CS:IP la dirección de comienzo de la subrutina. RET termina la subrutina).	No altera
RET	Número (opcional)	RET RET n	RETORNO DE SUBROUTINA. Devuelve el control al programa principal. Carga la dirección completa de la instrucción siguiente a la CALL que origino la llamada. Si la subrutina es NEAR, RET sustituye el contenido del registro IP por la palabra situada en la parte más baja de la pila (apuntada por SP). Tras esto, SP incrementa dos unidades. Si la subrutina es FAR, se sacan dos palabras de la pila. La primera se carga en el registro IP y la segunda en el CS. En total SP se incrementa en 4 unidades. Si se coloca un valor numérico 'n' tras RET, SP se incrementa en la misma cantidad (esto es útil para borrar parámetros pasados a través de la pila: n = 2 por cada parámetro si la llamada es NEAR y n=4 por cada parámetro si la llamada es FAR)	No altera
INT	Tipo de interrupción	INT n (n = 0...255)	INTERRUPCIÓN. Realiza una interrupción por software. Abandona el curso normal del programa para ejecutar la rutina de atención de la interrupción, salvando antes el contenido de IF e IP en la pila. El vector de interrupción está ubicado en la posición de memoria (0000:4*n) a la (0000:4*n+3)	IF=0, TF=0
JMP	Dirección	JMP Destino	SALTO INCONDICIONAL. Puede ser directo (lo que sigue a JMP es la dirección que se carga en IP) o indirecto (la dirección de salto está contenida en el registro o dirección que sigue a JMP). Dependiendo del segmento, es un salto NEAR o FAR.	No altera
JA JNBE	Desplazamiento	JA Desplaz. JNBE Desplaz.	SALTO SI SUPERIOR. Salta si se cumple CF=0 y ZF=0. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. NOTA: Se usan para números sin signo. Son la misma instrucción JA = JNBE.	No altera
JAE JNB	Desplazamiento	JAE Desplaz. JNB Desplaz.	SALTO SI SUPERIOR O IGUAL. Salta si se cumple CF=0. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. NOTA: Se usan para números sin signo. Son la misma instrucción JAE = JNB = JNC.	No altera
JB JNAE	Desplazamiento	JB Desplaz. JNAE Desplaz.	SALTO SI INFERIOR. Salta si se cumple CF=1. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. NOTA: Se usan para números sin signo. Son la misma instrucción JB = JNAE = JC.	No altera
JBE JNA	Desplazamiento	JNA Desplaz. JBE Desplaz.	SALTO SI INFERIOR O IGUAL. Salta si se cumple CF=0. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. NOTA: Se usan para números sin signo. Son la misma instrucción JAE = JNB = JBE.	No altera
JC	Desplazamiento	JC Desplaz.	SALTO SI HAY ACARREO. Salta si el flag de acarreo (CF) está a '1'. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. Son la misma instrucción JC = JNAE = JB.	No altera
JNC	Desplazamiento	JNC Desplaz.	SALTO SI NO HAY ACARREO. Salta si el flag de acarreo (CF) está a '0'. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. Son la misma instrucción JNC = JNB = JAE.	No altera
JCXZ	Desplazamiento	JCXZ Desplaz.	SALTO SI CX ES CERO. Salta si el registro CX contiene '0'. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante.	No altera
JE	Desplazamiento	JE Desplaz.	SALTO SI IGUAL. Salta si el flag de cero (ZF) está a '1'. El salto debe estar entre 128 bytes hacia atrás y 127 bytes hacia delante. Son la misma instrucción JE = JZ.	No altera
JNE	Desplazamiento	JNE Desplaz.	SALTO SI NO IGUAL. Salta si el flag de cero (ZF) está a '0'. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. Son la misma instrucción JNE = JNZ.	No altera
JG JNLE	Desplazamiento	JG Desplaz. JNLE Desplaz.	SALTO SI MAYOR. Salta si se cumple ZF=0 y SF=OF. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. NOTA: Se usan para números con signo. Son la misma instrucción JG = JNLE.	No altera
JGE JNL	Desplazamiento	JGE Desplaz. JNL Desplaz.	SALTO SI MAYOR O IGUAL. Salta si se cumple SF=OF. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. NOTA: Se usan para números con signo. Son la misma instrucción JGE = JNL.	No altera
JL JNGE	Desplazamiento	JL Desplaz. JNGE Desplaz.	SALTO SI MENOR. Salta si se cumple SF distinto de OF. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. NOTA: Se usan para números con signo. Son la misma instrucción JL = JNGE.	No altera
JLE JNG	Desplazamiento	JLE Desplaz. JNG Desplaz.	SALTO SI MENOR O IGUAL. Salta si se cumple ZF=1 o SF distinto de OF. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. NOTA: Se usan para números con signo. Son la misma instrucción JLE = JNG.	No altera
JO	Desplazamiento	JO Desplaz.	SALTO SI HAY DESBORDAMIENTO. Salta si el flag de desbordamiento (OF) está a '1'. El salto debe estar entre 128 bytes hacia atrás y 127 bytes hacia delante.	No altera
JNO	Desplazamiento	JNO Desplaz.	SALTO SI NO HAY DESBORDAMIENTO. Salta si el flag de desbordamiento (OF) está a '0'. El salto debe estar entre 128 bytes hacia atrás y 127 bytes hacia delante.	No altera
JP JPE	Desplazamiento	JP Desplaz. JPE Desplaz.	SALTO SI HAY PARIDAD (Paridad PAR). Salta si el flag de paridad (PF) está a '1'. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. Son la misma instrucción JP = JPE.	No altera
JPO JNP	Desplazamiento	JPO Desplaz. JNP Desplaz.	SALTO SI NO HAY PARIDAD (Paridad IMPAR). Salta si el flag de paridad (PF) está a '0'. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. Son la misma instrucción JPO = JNP.	No altera
JS	Desplazamiento	JS Desplaz.	SALTO SI SIGNO NEGATIVO. Salta si el flag de signo (SF) está a '1'. El salto debe estar entre 128 bytes hacia atrás y 127 bytes hacia delante.	No altera
JNS	Desplazamiento	JNS Desplaz.	SALTO SI SIGNO POSITIVO. Salta si el flag de signo (SF) está a '0'. El salto debe estar entre 128 bytes hacia atrás y 127 bytes hacia delante.	No altera
JZ	Desplazamiento	JZ Desplaz.	SALTO SI CERO. Salta si el flag de cero (ZF) está a '1'. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. Son la misma instrucción JZ = JE.	No altera
JNZ	Desplazamiento	JNZ Desplaz.	SALTO SI NO CERO. Salta si el flag de cero (ZF) está a '0'. El salto debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante. Son la misma instrucción JNZ = JNE.	No altera
LOOP	Desplazamiento	LOOP Desplaz.	BUCLE. Provoca la repetición de un grupo de instrucciones un número de veces, especificado por el registro CX. Si CX es distinto de '0', salta al desplazamiento. El desplazamiento debe estar comprendido entre 128 bytes hacia atrás y 127 bytes hacia delante.	No altera

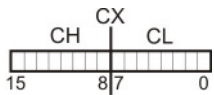
Registros Generales:



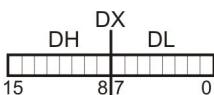
Acumulador



Base



Contador



Datos

Programa ejemplo:

DOSSEG	;Programa de ejemplo	
.MODEL SMALL		
.STACK 100h		
.DATA		
Var_1 DB ?	;Reserva un byte sin inicializar	
Var_2 DW 101001010101b	;Reserva una palabra, inicializada en binario	
Var_3 DW 2567	;Reserva una palabra, inicializada en decimal	
Var_4 DD 0A2FFFFFFh	;Reserva una palabra doble, inicializada en hexadecimal	
Texto DB 'Mensaje',10,13	;Reserva espacio para la cadena Mensaje + LF + CR	
Diez EQU 10	;Define una constante	
.CODE		
Inicio: mov ax,@DATA	;Carga en AX segmento de datos	
mov ds,ax	;Inicializa DS con el segmento de datos	
mov Var_1,32	;Inicializa Var_1 con el valor 32	
mov Var_4,-2345	;Inicializa Var_4 con el valor -2345	
mov bx,OFFSET Texto	;Carga el desplazamiento (dirección) de Texto en BX	
mov [bx],A'	;Carga en la dirección apuntada por BX el ASCII 'A'	
lea di,Var_1	;Carga el desplazamiento (dirección) de Var_1 en DI	
mov ch,[di]	;Carga en CH el byte de la dirección apuntada por DI	
mov ax,Var_3	;Pone en AX el contenido de Var_3	
add ax,Var_2	;Suma AX con Var_2	
mov cl,Diez	;Carga la constante de valor 10 en CL	
mov ah,4Ch	;Servicio de la int 21h para regresar al sistema	
int 21h	;Llamada a la interrupción	
END Inicio		

Palabra de estado del microprocesador (Flags):

- - - - O D I T S Z - A - P - C

- O:** Overflow. Desbordamiento en una operación con signo.
- D:** Dirección. Signo del incremento para los registros índice en operaciones de cadena (D = 1 = decrementa; D = 0 = incrementa).
- I:** Interrupciones. Habilita interrupciones enmascarables.

- T:** Trampa. Activa la ejecución de instrucciones paso a paso.
- S:** Signo. El resultado es negativo (S = 1).
- Z:** Cero. El resultado es cero. (Z = 1).
- A:** Acarreo Auxiliar. En operaciones BCD, desbordamiento en una operación sin signo o resultado negativo.
- P:** Paridad. El resultado tiene un número par de unos (P = 1 = paridad par; P = 0 = paridad impar).
- C:** Acarreo. Desbordamiento en una operación sin signo o resultado negativo.