

Mulador de transferencia

MOV	REG	REG/MEM	copia el operando fuente en el destino, los operadores deben ser del mismo tipo, no se puede mover datos entre dos porciones de memoria ni un valor inmediato a reg de segmento. CS no puede ser destino
	REG/MEM	Reg	
	REG/MEM	NUM	

XCHG REG/MEM Reg
Intercambia contenido de operandos puede ser byte o palabra pero mismo tamaño en ambos, no se pueden cambiar 2 porciones de memoria ni un reg de segmento

STC Activa Flag de acarreo CF=1

CLC Bona Flag de acarreo CF=0

CMC Complementa Flag de acarreo si CF=1 → CF=0 si CF=0 → CF=1

STD Activa Flag de dirección

CLD Bona Flag de dirección

STI Activa Flag de interrupción

CLI Bona Flag de interrupción

PUSH REG 16/MEM 16
Mete operando en la pila, decrementa dos unidades SP y luego lo almacena. Solo se almacenan datos tipo palabra SS y SP deben estar inicializados

POP REG 16/MEM 16
Saca una palabra de la pila, copia valor SS:SP e incrementa en 2 valor de SP, CS no puede ser destino

IN AL/AX PORT/DX
Copia el contenido del puerto de entrada en el acumulador

OUT PORT/DX AL/AX
Transfiere un Byte o palabra

LEA Reg 16 Mem
Carga la dirección efectiva, guarda el desplazamiento (offset)

LEA = dirección MOV = contenido

LDS y LES
Carga el contenido de una dirección de memoria de 32 bits, la parte base en el registro especificado y la alta en DS y ES
(carga segmento en DS / ES y offset en registro)

Instrucciones aritméticas

ADD	Reg Reg/Mem Reg/Mem	Reg/Mem Reg Num	Suma dos operandos fuentes al destino ambos del mismo tipo y no se pueden sumar 2 posiciones de mem actualiza banderas
ADC	Reg Reg/Mem Reg/Mem	Reg/Mem Reg Num	Suma llevando en cuenta la bandera de acarreo (CF), actualiza banderas
SUB	Reg Reg/Mem Reg/Mem	Reg/Mem Reg Num	Resta dos operandos y cambia el resultado en destino, deben ser del mismo tipo y no se puede con 2 posiciones de memoria
SBB	Reg Reg/Mem Reg/Mem	Reg/Mem Reg Num	Resta llevando en cuenta la bandera de acarreo, actualiza banderas
MUL	Reg/Mem		Multiplicación sin signo si el operando es un byte $AX = AL * \text{operando}$, si no, en DX se guarda la palabra más significativa y en AX la menos significativa
IMUL	Reg/Mem		Similar a mul pero tiene en cuenta el signo
DIV	Reg/Mem		División sin signo, si el divisor es de 1 byte, se guarda el cociente en AL, resto en AH, si es de 16 bits, dividiendo DX:AX, divisor operando de 16 bits, se guarda cociente en AX y resto en DX
INC	Reg/Mem		incrementa en 1 unidad
DEC	Reg/Mem		decrementa en 1 unidad
CMP	Reg Reg/Mem Reg/Mem	Reg/Mem Reg Num	Compara 2 resultados, el resultado no se almacena, pero si lo hacen las banderas ambos operandos deben ser del mismo tipo

$$\begin{array}{r} \text{dividendo} \\ 50 \overline{) 50} \\ \underline{50} \\ 0 \\ \underline{0} \\ 0 \\ \underline{0} \\ 0 \end{array}$$
 5 ← divisor
10 ← cociente
0 ← resto

Instrucciones lógicas

NEG	Reg/Mem		Forma el complemento a 2
NOT	Reg/Mem		No lógico, cambia los ceros por 1 y 1 por cero no altera las banderas
AND	Reg Reg/Mem Reg/Mem	Reg/Mem Reg Num	y lógico (altera las banderas)
OR	Reg/Mem Reg/Mem Reg/Mem	Reg/Mem Reg Num	o lógico
XOR	O lógico exclusivo
test	And lógico sin almacenar resultado, altera las banderas

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array}$$