

Instrucciones Lógicas

OR

Instrucción	Descripción	Flags Afectadas
OR REG1, REG2	REG1 = REG1 ó REG2	O, S, Z, A, P, C
OR REG1, CTE	REG1 = REG1 ó CTE	O, S, Z, A, P, C
OR MEM, REG1	[MEM] = [MEM] ó REG1	O, S, Z, A, P, C
OR REG1, MEM	REG1 = [MEM] ó REG1	O, S, Z, A, P, C
*OR S MEM, CTE	[MEM] = [MEM] ó CTE	O, S, Z, A, P, C

*Se debe indicar en el sufijo S el tamaño de la constante (byte, word, dword).

El operador lógico se aplica bit a bit.

AND (mismas opciones de operandos que OR)

Instrucción	Descripción	Flags Afectadas
AND OP1, OP2	OP1 = OP1 y OP2	O, S, Z, A, P, C

XOR (mismas opciones de operandos que OR)

Instrucción	Descripción	Flags Afectadas
XOR OP1, OP2	OP1 = OP1 xor OP2	O, S, Z, A, P, C

NOT

Instrucción	Descripción	Flags Afectadas
NOT REG	REG = ¬ REG	O, S, Z, A, P, C
*NOT S MEM	[MEM] = ¬ [MEM]	O, S, Z, A, P, C

*Se debe indicar en el sufijo S el tamaño de la variable almacenada en MEM (byte, word, dword)

SHL

Instrucción	Descripción	Flags Afectadas
SHL REG, CTE	Desplazamiento hacia la Izquierda	O, S, Z, A, P, C
*SHL S MEM, CTE		O, S, Z, A, P, C
SHL REG, CL		O, S, Z, A, P, C
*SHL S MEM, CL		O, S, Z, A, P, C

*Se debe indicar en S el tamaño de la variable almacenada en MEM o la CTE (byte, word, dword)

Desplaza el contenido del Operando (REG o MEM) hacia la izquierda tantas veces como indica CTE o CL.

El bit más significativo en cada desplazamiento se setea en CF, y el menos significativo se setea en cero.

La constante puede ser de a lo sumo 8 bits, solo puede utilizarse el registro CL.

SHR (mismas opciones de operandos que SHL)

Instrucción	Descripción	Flags Afectadas
SAR OP1, OP2	Desplazamiento hacia la Derecha	O, S, Z, A, P, C

Desplaza el contenido del Operando (REG o MEM) hacia la derecha, tantas veces como indica CTE o CL. En cada desplazamiento el bit menos significativo se carga en CF, y el más significativo se setea en cero (No extiende el signo).

La constante puede ser de a lo sumo 8 bits, solo puede utilizarse el registro CL.

Instrucciones de Comparación y Comprobación

CMP

Instrucción	Descripción	Flags Afectadas
CMP REG1, REG2	FLAGS = REG1 - REG2	O, S, Z, A, P, C
CMP REG1, CTE	FLAGS = REG1 - CTE	O, S, Z, A, P, C
CMP MEM, REG1	FLAGS = [MEM] - REG1	O, S, Z, A, P, C
CMP REG1, MEM	FLAGS = [MEM] - REG1	O, S, Z, A, P, C
*CMP S MEM, CTE	FLAGS = [MEM] - CTE	O, S, Z, A, P, C

*Se debe indicar en el sufijo S el tamaño de la constante (byte, word, dword).

Realiza una resta entre el primer y segundo operando, y no modifica ninguno de estos, el resultado se ve reflejado en los Flags (ver guía de Saltos Condicionales).

TEST (mismas opciones de operandos que CMP)

Instrucción	Descripción	Flags Afectadas
TEST OP1, OP2	FLAGS = OP1 and OP2	O, S, Z, A, P, C

Realiza una conjunción bit a bit entre el primer y segundo operando, y no modifica ninguno de estos, el resultado se ve reflejado en los Flags (ver guía de Saltos Condicionales).

BT

Instrucción	Descripción	Flags Afectadas
BT REG1, REG2	CF = REG1 _[REG2]	O, S, Z, A, P, C
BT MEM, REG2	CF = [MEM] _[REG2]	O, S, Z, A, P, C
BT REG1, CTE	CF = REG1 _[CTE]	O, S, Z, A, P, C
BT MEM, CTE	CF = [MEM] _[CTE]	O, S, Z, A, P, C

Testea un bit del primer Operando el cual es indexado por el segundo Operando y devuelve el valor en CF. El segundo Operando debe ser un número entre 0 y 31, o entre 0 y 15 dependiendo el tamaño del primer Operando el cual solo puede ser de 32 o 16 bits.

BTC (mismas opciones de operandos que BT)

Instrucción	Descripción	Flags Afectadas
BT REG1, REG2	CF = REG1 _[REG2]	O, S, Z, A, C

Realiza lo mismo que BT, pero además complementa el bit del primer Operando.

BTR (mismas opciones de operandos que BT)

Instrucción	Descripción	Flags Afectadas
BT REG1, REG2	CF = REG1 _[REG2]	O, S, Z, A, C

Realiza lo mismo que BT, pero además setea en cero el bit del primer Operando.

BTS (mismas opciones de operandos que BT)

Instrucción	Descripción	Flags Afectadas
BT REG1, REG2	CF = REG1 _[REG2]	O, S, Z, A, C

Realiza lo mismo que BT, pero además setea en uno el bit del primer Operando.

CLC / CMC

Instrucción	Descripción	Flags Afectadas
CLC	CF = 0	C
CMC	CF = -CF	C