Conjunto de instrucciones del 8086
con códigos de operación y campos de direccionamiento.

Formato de Instrucciones

CodOperación[d,v,s,w] + mod [reg] r/m + [byte bajo dato] + [byte alto dato, si w=1]
(1 byte) (1 byte) (1 byte) (1 byte)

Resumen del segundo byte de la instrucción: **mod [reg] r/m**

Aplicaciones de bits del campo "mod"

mod	Desplazamiento
00	DISP=0 (Desplazaminto ausente)
01	DISP= Desplazamiento bajo de 1 6 bits con signo. Desplazamiento alto ausente.
10	DISP=Desplazamiento Alto:Desplazamiento Bajo
11	r/m se trata como un campo "reg"

Aplicaciones de bits del campo "reg"

16 bits (w = 1)		8 bits (w=0)		Segmento	
000	AX	000	AL	00	ES
001	CX	001	CL	01	CS
010	DX	010	DL	10	SS
011	BX	011	BL	11	DS
100	SP	100	AH		
101	BP	101	CH		
110	SI	110	DH		
111	DI	111	BH		

Aplicaciones de bits del campo "r/m"

r/m	Dirección de los operandos
000	BX + SI + Desplazamiento
001	BX + DI + Desplazamiento
010	BP + SI + Desplazamiento
011	BP + DI + Desplazamiento
100	SI + Desplazamiento
101	DI + Desplazamiento
110	BP + Desplazamiento*
111	BX + Desplazamiento

*Excepto si mod=00 y r/m=110; entonces EA=Desplazamiento Alto:Desplazamiento Bajo.

Si d=1, "a", si d=0 "de";
Si w=1, instrucción de una palabra, si w=0, instrucción de un byte.
Si s:w=01, operando formado por dato inmediato de 16 bits.
Si s:w=11, operando de 16 bits formado por un dato inmediato de 1 byte con extensión de signo.
Si v=0, "cuenta"=1, si v=1, "cuenta" en CL.

Transferencia de datos

mov (Mover)

-Registro/Memoria a/de registro: 1000 10dw + mod reg r/m

-Inmediato a registro/memoria: 1100 011w + mod 000 r/m + [byte bajo dato] + [byte alto dato, si w=1]

-Inmediato a registro: 1011 w reg + [byte bajo dato] + [byte alto dato, si w=1]

-Memoria a Acumulador: 1010 000w + [byte bajo dirección] + [byte alto dirección]

-Acumulador a Memoria: 1010 001w + [byte bajo dirección] + [byte alto dirección]

-Registro/Memoria a Registro de Segmentación: 1000 1110 + mod 0 reg r/m

-Registro de Segmentación a registro/memoria: 1000 1100 + mod 0 reg r/m

push (Introducir a la pila)

-Registro/memoria: 1111 1111 + mod 110 r/m

-Registro: 0101 0 reg

pop (Extraer de la pila)

-Registro/memoria: 1000 1111 + mod 000 r/m

-Registro: 0101 1 reg

-Registro de segmentación: 000 reg 110

xchg (Intercambiar)

-Registro/memoria con registro: 1000 011w + mod reg r/m

-Registro con acumulador: 1001 0 reg

in (Entrada de puerto E/S aAL/AX)

-Puerto fijo: 1110 010w + puerto

-Puerto variable (en DX): 1110 110w

out (Salida de AL/AX a puerto de E/S)

-Puerto fijo: 1110 011w + puerto

-Puerto variable (en DX): 1110 111w

lea (Cargar EA (effective address) a registro: 10001101 + mod reg r/m

Aritméticas

add (suma)

-Registro/memoria con registro a cualquiera de ellos: 0000 00dw + mod reg r/m

-Inmediato a registro/memoria: 1000 00sw + mod 000 r/m + [byte bajo dato] + [byte alto dato, si s:w=01]

-Inmediato a acumulador: 0010 110w + [byte bajo dato] + [byte alto dato, si w=1]

adc (Suma con bit de acarreo)

-Registro/memoria con registro a cualquiera de ellos: 0001 00dw + mod reg r/m

-Inmediato a registro/memoria: 1000 00dw + mod 010 r/m + [byte bajo dato] + [byte alto dato, si s:w=01]

-Inmediato a acumulador: 0001 010w + [byte bajo dato] + [byte alto dato, si w=1]

sub (Resta)

-Registro/memoria con registro a cualquiera de ellos: 0010 10dw + mod reg r/m

-Inmediato con registro/memoria: 100000sw + mod 111 r/m + [byte bajo dato] + [byte alto dato, si s:w=01]

-Inmediato con acumulador: 0011 110w + [byte bajo dato] + [byte alto dato, si w=1]

sbb (Resta con bit de borrow)

-Registro/memoria con registro a cualquiera de ellos: 0001 10dw + mod reg r/m

-Inmediato con registro/memoria: 100000sw + mod 011 r/m + [byte bajo dato] + [byte alto dato, si s:w=01]

-Inmediato con acumulador: 0001 110w + [byte bajo dato] + [byte alto dato, si w=1]

inc (Incrementar)

-Registro/memoria: 1111 111w + mod 000 r/m

-Registro: 0100 0 reg

dec (Decrementar)
-Registro/memoria: 1111 111w + mod 001 r/m
-Registro: 0100 1 reg
mul (Multiplicación entera sin signo): 1111 011w + mod 100 r/m
imul (Multiplicación entera con signo): 1111 011w + mod 101 r/m
div (División entera sin signo): 1111 011w + mod 110 r/m
idiv (División entera con signo): 1111 011w + mod 111 r/m
cmp (Comparar, resta aritmética para actualizar banderas, sin resultado)
-Registro/memoria y registro: 0011 10dw + mod reg r/m
-Inmediato a acumulador: 0011 110w + [byte bajo dato] + [byte alto dato, si w=1]
-Inmediato a registro: 1000 00sw 1111 1reg + [byte bajo]
cbw (Conversión byte en AL a palabra en AX): 1001 1000
cwd (Conversión palabra en AX a doble palabra en DX:AX): 1001 1001

Lógicas
and (Producto lógico, conjunción)
-Registro/memoria con registro a cualquiera de ellos: 0010 00dw + mod reg r/m
-Inmediato a registro/memoria: 1000 000w + mod 100 r/m + [byte bajo dato] + [byte alto dato, si w=1]
-Inmediato a acumulador: 0010 010w + [byte bajo dato] + [byte alto dato, si w=1]
or (Suma lógica, disyunción)
-Registro/memoria con registro a cualquiera de ellos: 0000 10dw + mod reg r/m
-Inmediato a registro/memoria: 1000 000w + mod 001 r/m + [byte bajo dato] + [byte alto dato, si w=1]
-Inmediato a acumulador: 0000 110w + [byte bajo dato] + [byte alto dato, si w=1]
not (Negación, complemento): 1111 011w + mod 010 r/m
xor (O exclusivo)
-Registro/memoria con registro a cualquiera de ellos: 001100dw+mod reg r/m
-Inmediato a registro/memoria: 1000 000w + mod 110 r/m + [byte bajo dato] + [byte alto dato, si w=1]
-Inmediato a acumulador: 0011010w + [byte bajo dato] + [byte alto dato, si w=1]
test (Prueba lógica, operación AND para actualizar banderas, sin resultado)
-Registro/memoria con registro: 1000 010w + mod reg r/m
-Inmediato y registro/memoria: 1111 011w + mod 000 r/m + [byte bajo dato] + [byte alto dato, si w=1]
-Inmediato y acumulador: 1010 100w + [byte bajo dato] + [byte alto dato, si w=1]
shl/sal (Corrimiento lógico/aritmético a la izquierda): 1101 00vw + mod 100 r/m
shr (Corrimiento lógico a la derecha): 1101 00vw + mod 101 r/m
sar (Corrimiento aritmético a la derecha): 1101 00vw + mod 111 r/m

Control de programa
jmp (Salto incondicional)
-Directo en segmento: 11101001 + [byte alto desplazamiento] + [byte bajo desplazamiento]
-Directo en segmento corto: 11101011 + [byte desplazamiento]
-Indirecto en segmento: 1111 1111 + mod 100 r/m
-Directo intersegmento: 1110 1010 + [byte bajo desplazamiento] + [byte alto desplazamiento]
+[byte bajo segmento] + [byte alto segmento]
-Indirecto intersegmento: 1111 1111 + mod 101 r/m
je/jz (Salto condicional si igual/zero): 01110100 + desplazamiento
jne/jnz (Salto condicional si no igual/no cero): 01110101 + desplazamiento
jl/jnge (Salto condicional si menor/no mayor o igual): 01111100 + desplazamiento
jle/jng (Salto condicional si menor o igual/no mayor): 01111110 + desplazamiento

jg/jnle (Salto condicional si mayor/no menor o igual): 01111111 + desplazamiento
jge/jnl (Salto condicional si no mayor o igual/no menor): 01111101 + desplazamiento
loop (Ejecutar bucle CX veces): 11100010 + desplazamiento

call (Llamada a procedimiento)
-Directo en segmento: 1110 1000 + [byte bajo desplazamiento] + [byte alto desplazamiento]
-Indirecto en segmento: 1111 1111 + mod 010 r/m
-Directo intersegmento: 1001 1010 + [byte bajo desplazamiento] + [byte alto desplazamiento]
+[byte bajo segmento] + [byte alto segmento]
-Indirecto intersegmento: 11111 1111 + mod 100 r/m
ret (Retorno de procedimiento)
-En el segmento: 1100 0011
-En el segmento, añadiendo un inmediato a SP: 1100 0010 + [byte bajo dato] + [byte alto dato]
-Intersegmento: 1100 1011
-Intersegmento, añadiendo un inmediato a SP: 1100 1010 + [byte bajo dato] + [byte alto dato]
int (Interrupción de software): 1100 1101 + byte tipo
iret (retorno de interrupción): 1100 1111

Control del procesador
clc (Borrar bandera de acarreo, clear carry flag): 1111 1000
stc (Activar bandera de acarreo, set carry flag): 1111 1001
cld (Borrar bandera de dirección, clear direction flag): 1111 1100
std (Activar bandera de dirección, set direction flag): 1111 1101