

XCHG

Intercambia el contenido de cualquier registro con el contenido de cualquier otro registro o localidad de memoria, no incluyendo los registros de segmento o intercambios de memoria a memoria.

Ejemplo:

AX = 1F60 BX = 472D CX = 310B DX = 8472

XCHG AX,BX

AX = 472D BX = 1F60 CX = 310B DX = 8472

XCHG CL,DH

AX = 472D BX = 1F60 CX = 3184 DX = 0B72



IN y OUT

IN:

Lee un dato de un dispositivo de E/S y lo almacena en el registro AL o AX.

OUT:

Transfiere un dato del registro AL o AX a un puerto de E/S.



IN y OUT

Existen dos formas para el direccionamiento de puertos con las instrucciones IN y OUT, las cuales son: **puerto fijo** y **puerto variable**.

- **Puerto fijo:** Permite la transferencia de datos entre AL o AX y un puerto de E/S con dirección de 8 bits.

Ejemplos: IN AX, 85 IN AL, 3F OUT 42, AL OUT 2A, AX

- **Puerto variable:** Permite la transferencia de datos entre AL o AX y un puerto de E/S con dirección de 16 bits, y el numero de puerto se almacena en DX.

Ejemplos: IN AX, DX IN AL, DX OUT DX, AL OUT DX, AX



IN y OUT

Tabla 2. Instrucciones IN y OUT.

Código de operación	Función
IN AL, pp	Un dato de 8 bits se transfiere del puerto pp a AL
IN AX,pp	Un dato de 16 bits se transfiere del puerto pp a AX
IN AL,DX	Un dato de 8 bits se transfiere del puerto DX a AL
IN AX,DX	Un dato de 16 bits se transfiere del puerto DX a AX
OUT pp,AL	Un dato de 8 bits se transfiere de AL al puerto DX
OUT pp,AX	Un dato de 16 bits se transfiere de AX al puerto pp
OUT DX,AL	Un dato de 16 bits se transfiere de AL al puerto DX
OUT DX,AX	Un dato de 16 bits se transfiere de AX al puerto DX

Nota: pp= un puerto de E/S de con dirección de 8 bits y DX = contiene la dirección de un puerto de E/S con dirección de 16 bits.



PUSH y POP

Se utilizan para empujar o sacar datos del **Segmento de Pila**.

El registro **SP** (Apuntador de Pila) almacena la dirección del segmento de Pila de donde se van a ingresar o sacar datos.

En el 8088 se dice que la Pila crece a direcciones mas pequeñas, ya que cuando se **ingresan** datos se **decrementa** el apuntador de pila, y cuando se **extraen** datos se **incrementa** el apuntador. Es decir, conforme se van ingresando datos SP apunta a direcciones cada vez mas pequeñas.



PUSH y POP

PUSH:

Ingresa 2 bytes de información a la pila.

Cuando un dato es metido a la pila, el byte **mas significativo** es almacenado en la localidad direccionada por **SP-1**, y el byte **menos significativo** es almacenado en la localidad **SP-2**.

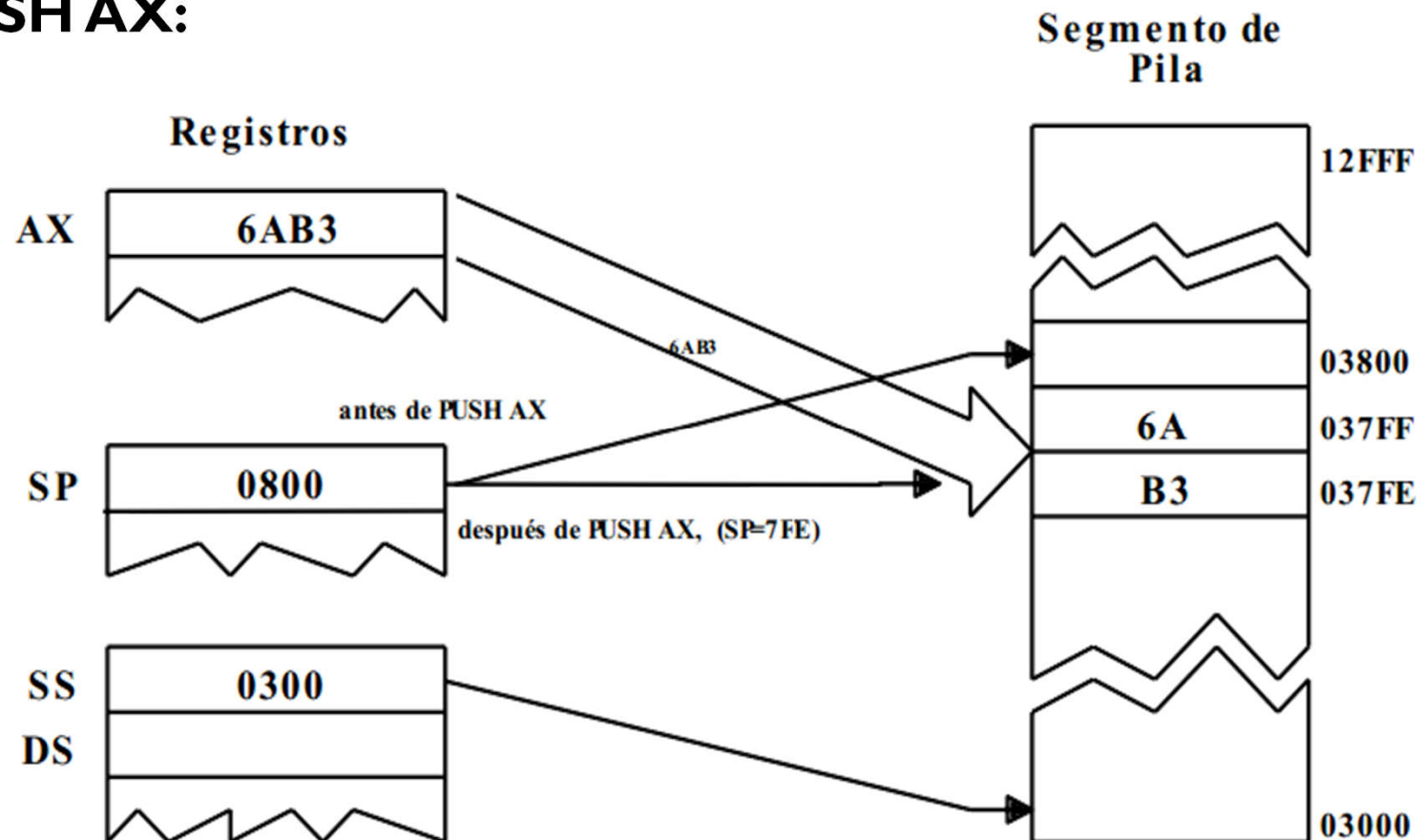
Después de que el dato se ha empujado a la pila, el registro **SP** es **decrementado en 2**.



PUSH y POP

Ejemplo:

PUSH AX:



PUSH y POP

POP:

Realiza lo inverso a la instrucción PUSH. POP **remueve 2 bytes** de la pila.

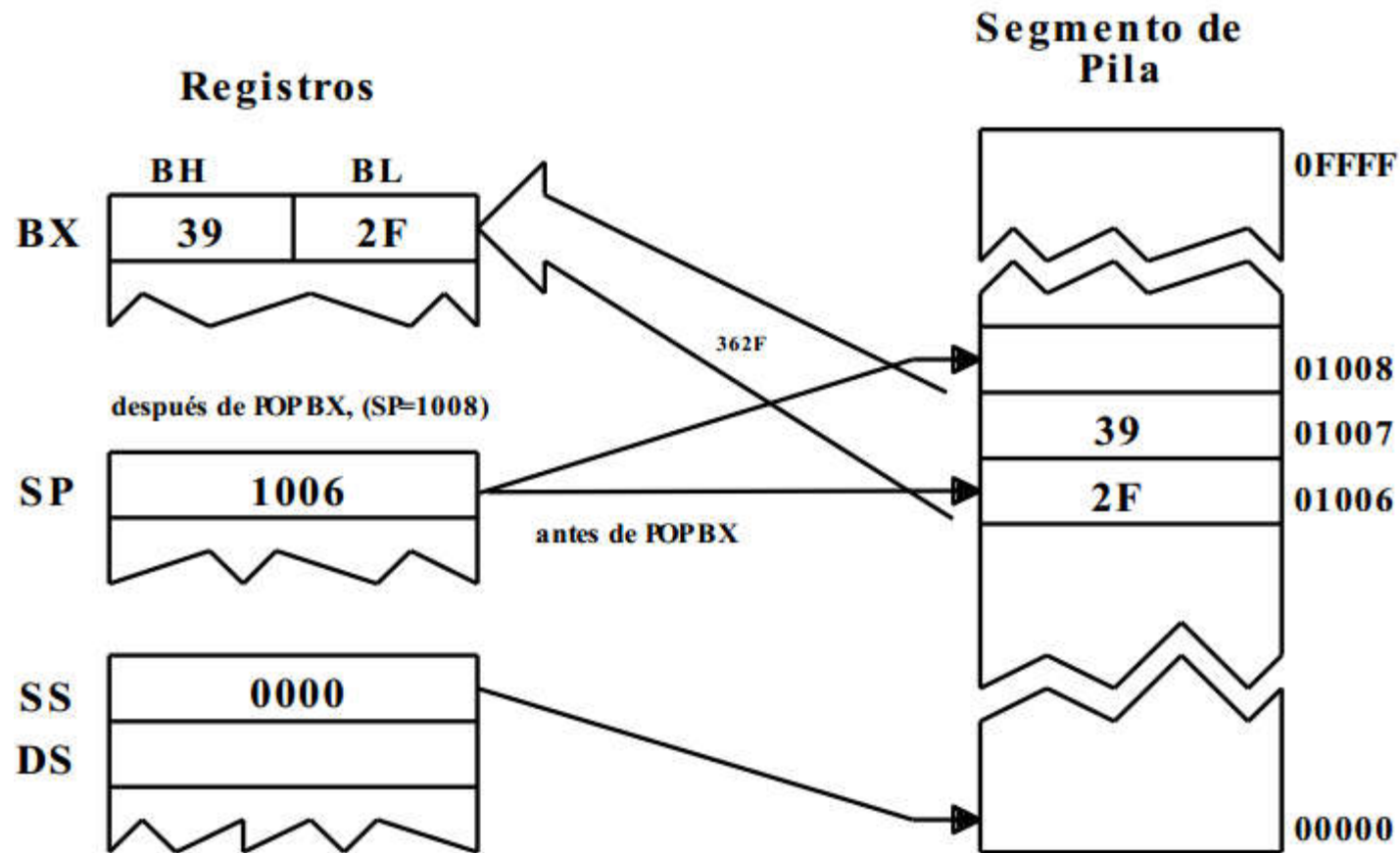
Suponga que la instrucción POP BX es ejecutada. El byte apuntado por **SP** es removido de la pila y almacenado en el registro BL. Después se remueve el byte apuntado por **SP+1** y se coloca en el registro BH.

Después de que ambos datos fueron extraídos de la pila, **SP** es **incrementado en 2**.



PUSH y POP

POP BX:



PUSH y POP

Las instrucciones PUSH y POP pueden operar sobre registros, memoria y registros de segmento.

Importante:

Sí se permite empujar a la pila el registro de segmento de Código (CS), pero no se permite extraer un valor de la pila y almacenarlo en CS, ya que no se permite la modificación de este registro.

PUSH CS ✓ (permitido)

POP CS ó **MOV CS,valor** , etc ✗



PUSHF y POPF

PUSHF:

Copia el contenido del registro de banderas a la pila.

POPF:

Realiza la operación inversa de PUSHF, remueve de la pila un dato de 16 bits que es cargado como el contenido del registro de banderas.

