Comparación CMP

La instrucción de comparación **CMP** es una resta que no cambia el valor de los operandos con los que se trabaja, solamente cambia las banderas.

Se emplea una comparación para verificar el contenido de un registro o de una localidad de memoria con otro valor.

Normalmente después de un CMP hay una instrucción de salto condicional, la cual prueba la condición en las banderas.

No se permite usar esta instrucción en los registros de segmento.

Comparación CMP

Tabla 12. Instrucciones de Comparación.

Instrucciones	Comentarios
CMP CL,BL	Realiza:CL-BL. CL y BL no cambian
CMP AX,SP	Compara: AX-SP
CMP AX,0CCCCH	Compara:AX-DS:0CCCC
CMP [DI],CH	Compara: DS:[DI]-CH
CMP CL,[BP]	Compara: CL-SS:[BP]
CMP AH,TEMP	Compara: AH-DS:TEMP
CMP DI,TEMP[BX]	Compara: DI-DS:TEMP[BX]

Control del Programa

Las instrucciones de control de programa permiten que el flujo de ejecución del programa cambie.

Esos cambios en el flujo ocurren después de decisiones hechas con las instrucciones **CMP** o **TEST** seguidas por una instrucción de **salto condicional**, o por medio de una instrucción de **salto incondicional**.

Salto incondicional **JMP**

El salto (JMP), es el tipo principal de instrucciones de control de programa, permite al programador saltar a secciones de un programa y bifurcar a cualquier parte de la memoria para la siguiente instrucción.

```
        MOV AX,1A7Fh
        MOV CL, 2
@@next: ROL AX,CL
        INC AX
        JMP @@next
        ADD BX,8
```

Nunca se llega a ejecutar esta instrucción

Salto incondicional **JMP**

Si se visualizan instrucciones JMP en lenguaje máquina, nos podemos encontrar con tres tipos diferentes de saltos:

- **Salto Corto**
- **Salto Cercano**
- **Salto Lejano**

El ensamblador escoge la mejor forma de la instrucción de salto.

Nunca se usa una dirección hexadecimal con cualquier instrucción de salto.

Salto incondicional **JMP**

Salto Corto:

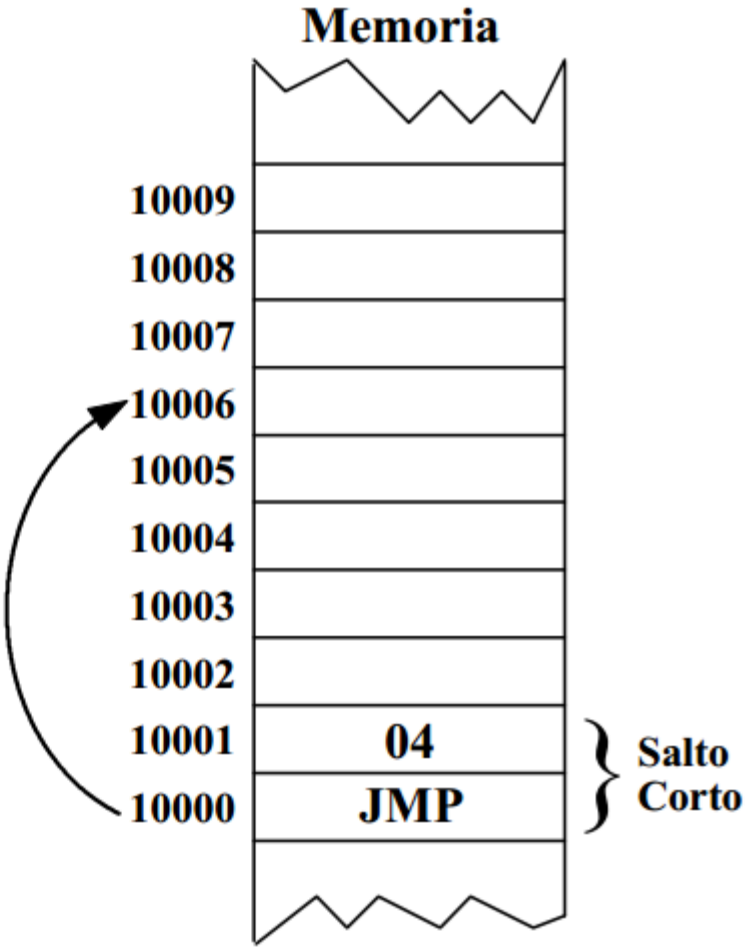
Después del código de operación de la instrucción de salto (de tamaño 1 byte), le sigue 1 byte que representa una **distancia** que toma valores entre **+127 a -128**.

Cuando el 8088 ejecuta una instrucción de salto corto, **suma el valor de distancia a IP** (el apuntador de instrucción), lo que ocasiona que el procesador bifurque hacia la instrucción almacenada en la nueva dirección apuntada por IP.

Salto incondicional **JMP**

Salto Corto:

CS=1000H
IP=0002H
Nuevo IP=IP+04
Nuevo IP=0006H



Salto incondicional **JMP**

Salto Cercano:

El salto cercano es similar al salto corto excepto que la distancia de salto puede ser mayor.

En este tipo de salto después del código de operación le siguen 2 bytes que indican la distancia de desplazamiento, la cual puede tomar valores de **±32Kb**.

Salto incondicional **JMP**

Salto Cercano:

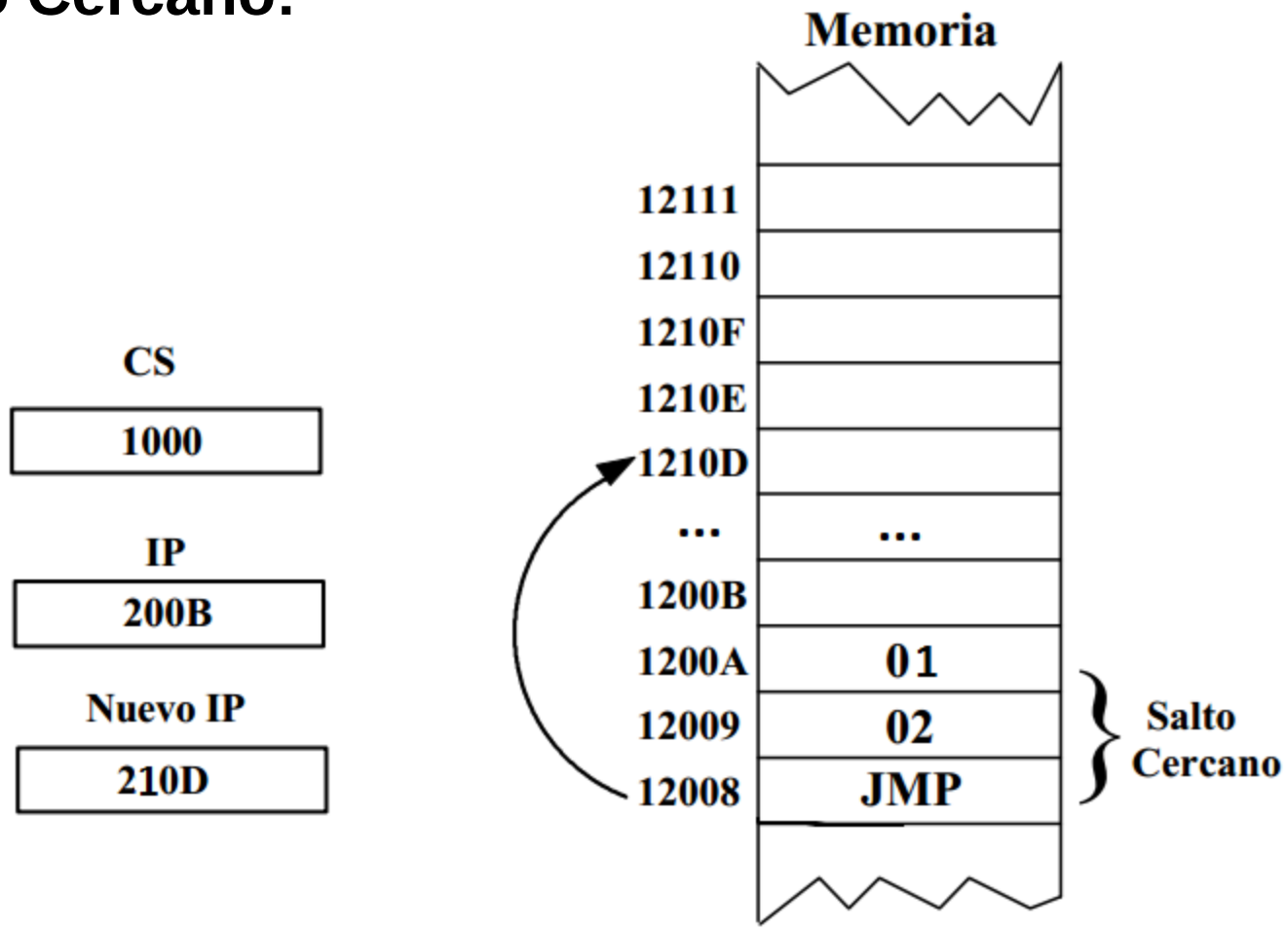


FIGURA 3 Un JMP cercano, el cual suma el desplazamiento a el contenido del registro IP.

Salto incondicional **JMP**

Salto Lejano:

Los saltos lejanos obtienen un nuevo segmento y dirección de desplazamiento para efectuar el salto.

Después del código de operación de la instrucción le siguen 4 bytes. De estos cuatro bytes, los 2 primeros establecen el **nuevo valor de IP**, y los siguientes 2 el **nuevo valor de CS** (registro de segmento de código).

A diferencia de los tipos de salto anteriores, los nuevos valores no se suman a IP y CS, sino que se establecen directamente en los registros.

Salto incondicional **JMP**

Salto Lejano:

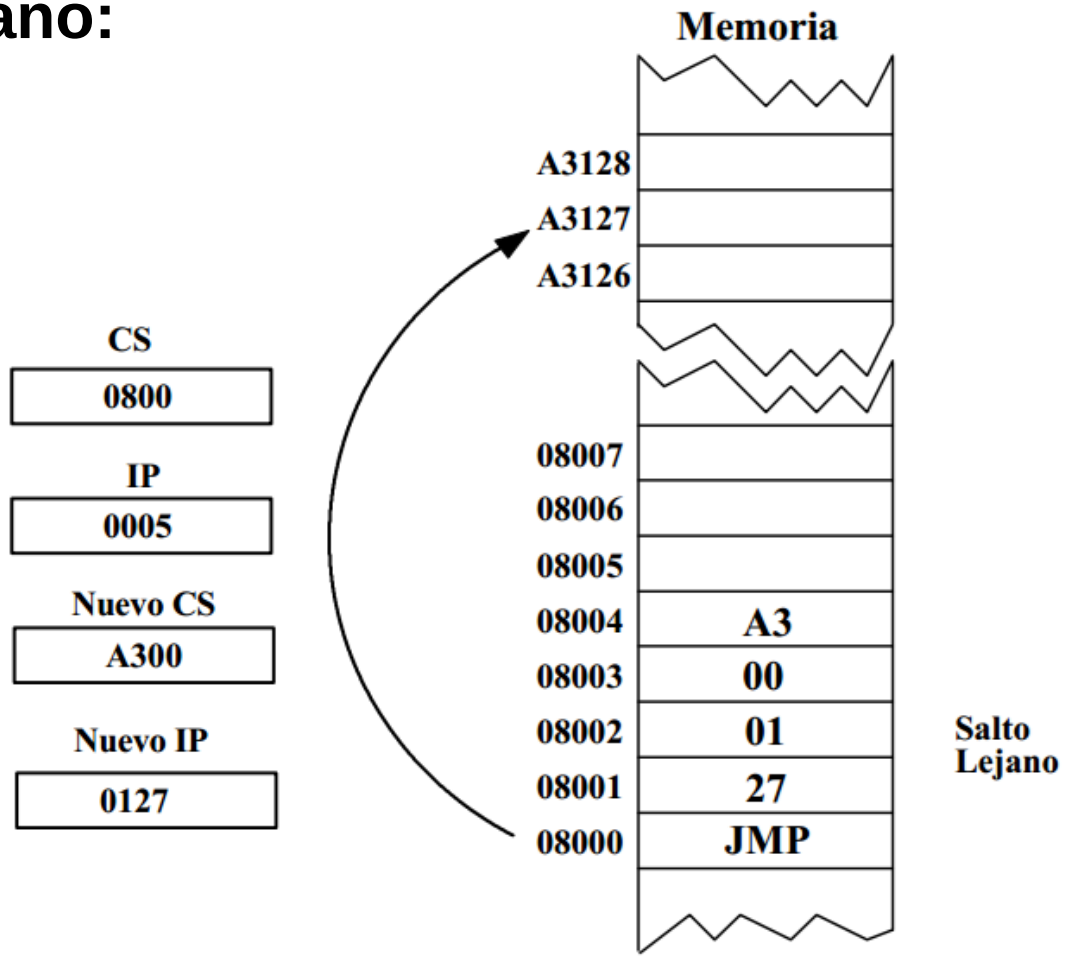


FIGURA 4 Un JMP lejano que remplaza el contenido de los registros CS e IP con los 4 bytes que siguen al código de instrucción.

Salto incondicional **JMP**

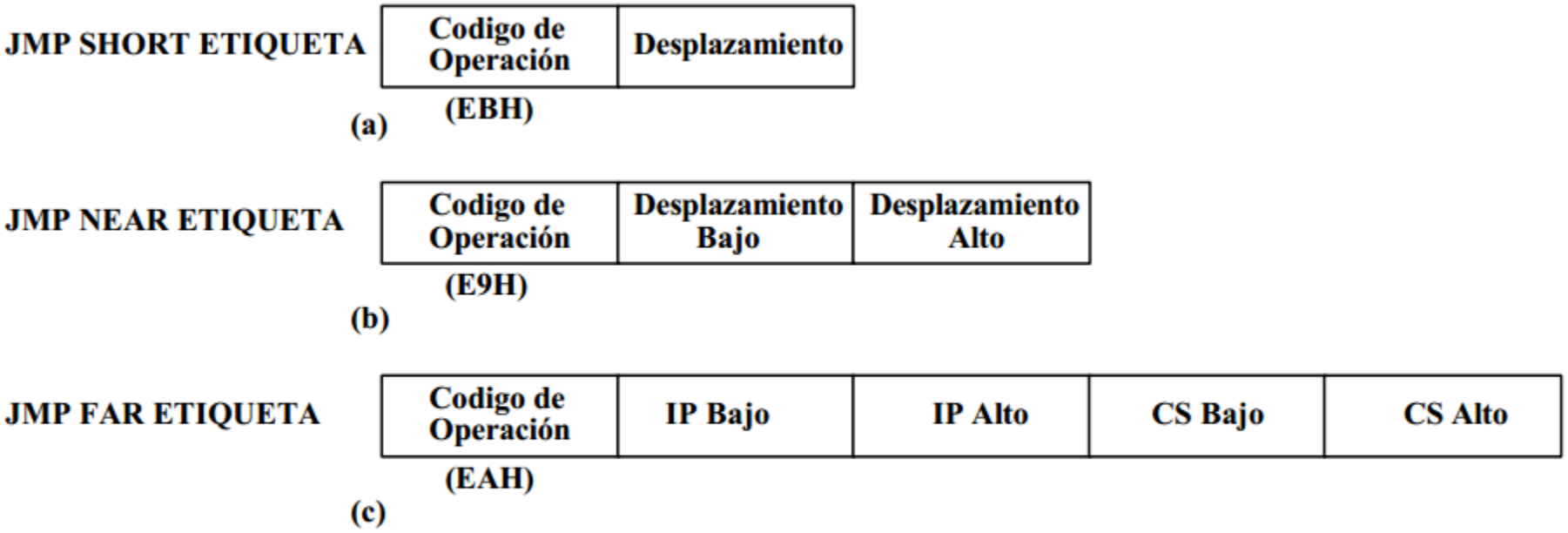


FIGURA 1. Los tipos de instrucciones de salto. (a) JMP corto (2 bytes), (b) JMP cercano (3 bytes) y (c) JMP lejano (5 bytes).

Salto incondicional **JMP**

Saltos con Registros como Operando:

La dirección del salto está en el **registro** especificado por la instrucción de salto.

Distinto al salto cercano, el contenido del registro se transfiere directamente en el apuntador de instrucción (no se suma al apuntador de instrucción como en los saltos cortos y cercanos).

Ejemplo:

JMP AX

Salto incondicional **JMP**

Saltos Indirectos Usando un Índice:

La dirección de salto también se puede obtener por medio de direccionamiento indirecto a memoria.

Ejemplo:

JMP [SI]

La instrucción de salto puede direccionar a un nuevo valor de desplazamiento de 16 bits (IP), o un nuevo valor de segmento y un nuevo valor de desplazamiento (IP y CS, 32 bits en total).

Se asume que se está direccionando a un nuevo valor desplazamiento de 16 bits a menos que se utilice la directiva FAR PTR. **Ejemplo**: **JMP FAR PTR [SI]**

Saltos Condicionados

Saltos condicionados

Los saltos condicionados son siempre saltos cortos.

Los saltos condicionados examinan los siguientes bits del registro de banderas:

signo (**S**), cero (**Z**), acarreo (**C**), paridad (**P**), y desbordamiento (**O**)

Si la condición bajo prueba es verdadera, ocurre una bifurcación a la etiqueta asociada con la instrucción de salto.

Si la condición es falsa, no se realiza el salto y por lo tanto se ejecuta la instrucción que le sigue.

Saltos condicionados

En JA es "C = 0 y Z = 0"

TABLA 1. Instrucciones de salto condicionales

Instrucción	Condicion probada	Comentario
JA	C=0 y Z≠0	Salta si esta arriba
JAE	C=0	Salta si esa arriba o igual
JB	C=1	Salta si esta abajo
JBE	C=1 o Z=1	Salta si esta abajo o igual
JC	C=1	Salta si hay acarreo
JE o JZ	Z=1	Salta si es igual o Sata si es 0
JG	Z=0 y S=0	Salta si es mayor
JGE	S=0	Salta si es mayor o igual a
JL	S≠0	Salta si es menor
JLE	Z=1 o S=1	Salta si es menor o igual
JNC	C=0	Salta si no hay acarreo
JNE o JNZ	Z=0	Salta si no es igual o Salta si no es 0
JNO	O=0	Salta si no hay sobre flujo
JNS	S=0	Salta si no hay signo
JNP	P=0	Salta si no hay paridad
JO	O=1	Salta si hay sobre flujo
JP	P=1	Salta si hay paridad
JS	S=1	Salta si hay signo
JCXZ	CX=0	Salta si CX=0

Saltos condicionados

Cuando se comparan **números sin signo**, se usan las instrucciones de salto:

JA, JB, JAE, JBE, JE, y JNE.

Cuando se comparan **números con signo**, se usan las instrucciones de salto:

JG, JL, JGE, JLE, JE, y JNE

Saltos condicionados

Cuando se trabaja con números sin signo, FFH esta arriba (es mayor) del 00H.

Sin embargo, al manejar los números con signo, un FFH (-1) es menor que 00H.

El siguiente diagrama muestra el orden de los números con signo y sin signo de 8 bits.

Saltos condicionados

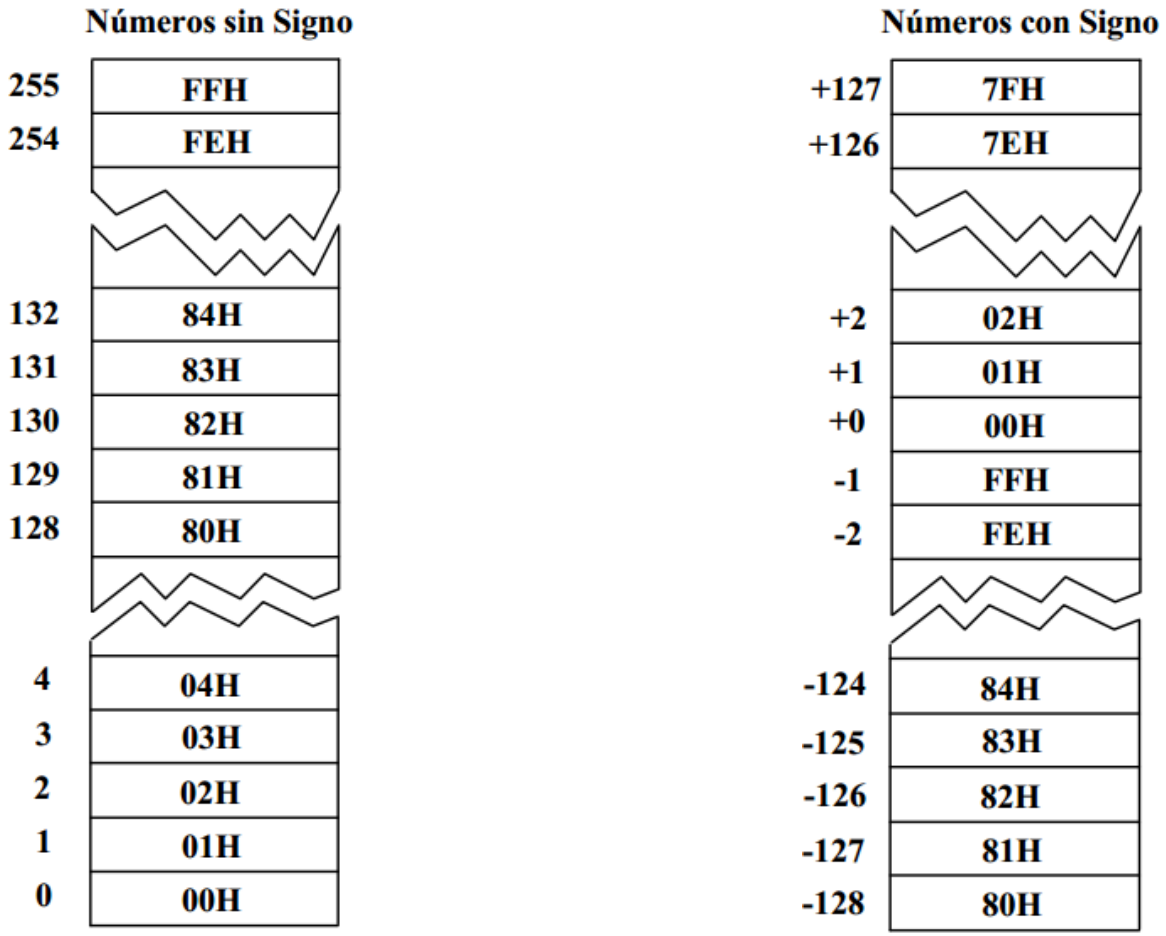


FIGURA 5. Numero con signo y sin signo.

Saltos condicionados

JCXZ:

Esta es la única instrucción de salto condicional que no examina los bits de bandera.

JCXZ examina el contenido del registro CX sin afectar los bits de bandera. Si **CX=0**, **ocurre el salto**, y si CX es diferente de cero, el salto no ocurre.

Saltos

Ejemplo 1 de saltos:

Programa que lee un dato de 8 bits del puerto 0x2456, si el dato es mayor o igual a 0x1F, envía un 0xAA en el puerto 0x2899, caso contrario envía un 0xBB.

```
            MOV DX,2456h
            IN AL,DX
            CMP AL,1Fh
            JAE @@mayor
            MOV AL,0BBh
            JMP @@out
@@mayor:    MOV AL,0AAh
@@out:      MOV DX,2899h
            OUT DX,AL
```

Saltos

Ejemplo 2:

Programa que lee un caracter (un dato de 8 bits) del puerto 0x80, si el dato es una letra *a*, envía un carácter *T* (indicando TRUE) en el puerto 0x92, caso contrario envía una F (indicando FALSE).

```
            IN AL,80h
            CMP AL,'a'
            JE @@es_a
            MOV AL,'F'
            JMP @@out
@@es_a:     MOV AL,'T'
@@out:      OUT 92h,AL
```

Saltos

Ejemplo 3:

Programa que lee un dato de 8 bits del puerto 0x2345, si el bit 3 del dato esta activo (si es un 1), envía un carácter *T* (indicando TRUE) en el puerto 0x92, caso contrario envía una F (indicando FALSE).

```
                MOV DX,2345h
                IN AL,DX
                TEST AL,08h
                JNZ @@es_1
                MOV AL,'F'
                JMP @@out
@@es_1:         MOV AL,'T'
@@out:          OUT 92h,AL
```

Ciclos

LOOP

La instrucción LOOP es una combinación de un decremento en CX y un salto condicionado.

Esta instrucción decrementa a CX y si CX es diferente de cero, se salta a la dirección indicada por la etiqueta.

Si CX llega a cero, no ocurre el salto y por lo tanto se ejecuta la instrucción que le sigue.

LOOP

Ejemplo:

Escriba un programa que envié los caracteres del abecedario por el puerto 0x92, uno a uno.

```
        MOV AL,'a'
        MOV CX,26  ;26 decimal, por lo que no se incluye la h
@@out:  OUT 92h,AL
        INC AL
        LOOP @@out
```

LOOPs Condicionados

La instrucción LOOP también tiene las formas condicionales: **LOOPE** y **LOOPNE**.

La instrucción **LOOPE** (**cicla mientras sea igual**) salta si CX es diferente de cero mientras una condición de igual existe. Se **saldrá del ciclo si la condición no es igual o** si el registro **CX se decrementa a cero**.

La instrucción **LOOPNE** (**cicla mientras no sea igual**) salta si CX es diferente de cero mientras una condición de no igual exista. Se **saldrá del ciclo si la condición es igual o** si el registro **CX se decrementa a cero.**