

# Introducción al Software del 8085

## ■ Directivas:

### ■ ORG: (Origen Absoluto)

- | Sintaxis: *ORG <dir>*
- | Las sentencias posteriores son ubicadas a partir de la dirección especificada.
- | Ejemplo: **ORG 1000H**

### ■ END: (Fin del módulo)

- | Indica al ensamblador que el programa fuente ha terminado.

### ■ EQU: (Igualar símbolo con valor)

- | Sintaxis: *ETIQUETA EQU VALOR*
- | Asigna un valor al símbolo del campo etiqueta. Este valor no puede modificarse durante el programa.
- | Ejemplo: **PI EQU 32**

# Introducción al Software del 8085

## ■ Pseudoinstrucciones:

### ■ DB: Definir byte

- | Sintaxis: *ETIQUETA DB VALOR*
- | Inicializa una posición de memoria con un valor especificado en el campo VALOR.
- | Ejemplo: **DATO DB 10H**

### ■ DS: Reservar memoria

- | Sintaxis: *ETIQUETA DS VALOR*
- | Reserva posiciones de memoria sin inicializar. El número de posiciones reservadas es igual a VALOR.
- | Ejemplo: **ZONA DS 10H**

### ■ DC: Define una constante

- | Sintaxis: *ETIQUETA DC DATO*
- | Inicializa una posición de memoria con el valor ASCII de DATO.
- | Ejemplo: **ASCII DC 'ABC'**

# Introducción al Software del 8085

## ■ Modos de direccionamiento I:

### ■ Inmediato:

- | El valor del operando está especificado en la propia instrucción.
- | Ejemplo:           MVI B,02H

### ■ Directo:

- | El operando viene especificado por la dirección de memoria donde se halla almacenado. La instrucción contiene la dirección de memoria.
- | Ejemplo:           LDA 1800H

### ■ Por registro:

- | La instrucción contiene el registro o pareja de registros donde se halla el operando.
- | Ejemplo:           ADD B

# Introducción al Software del 8085

## ■ Modos de direccionamiento II:

### ■ Por par de registros indirecto:

- | 1.- La dirección del operando viene determinada por una pareja de registros codificados en la instrucción (B-C, D-E).
  - Ejemplo:   LDAX B
- | 2.- Existe un caso particular cuando la pareja de registros es H-L. En este caso se utiliza el nemónico "M" para referenciar dicho direccionamiento.
  - Ejemplo:   MOV A,M

# Introducción al Software del 8085

## ■ Clasificación de las instrucciones:

- Instrucciones de manipulación y transferencia.
- Instrucciones aritméticas y lógicas.
- Instrucciones de ruptura de secuencia.
  - | Saltos condicionales e incondicionales.
  - | Instrucciones de llamada y retorno de subrutinas.
- Instrucciones de control:
  - | Manejo de la pila.
  - | De entrada y salida.
  - | Interrupciones.

## Instrucciones de transferencia I

### MOV r1,r2

(r2) → r1

Formato:

0 1 D D D S S S

Direccionam.: Por registro

Señalizadores: Ninguno

Ciclos M.: 1      Estados: 4

### MOV r,M

[(H)(L)] → r

Formato:

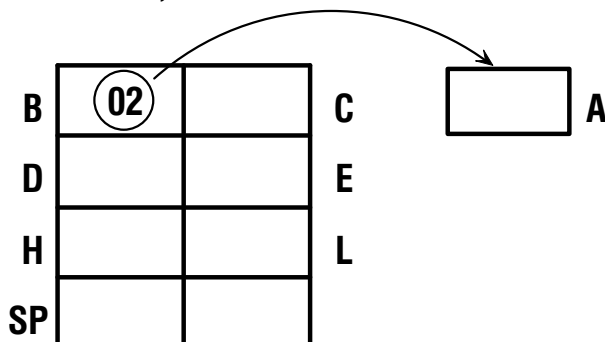
0 1 D D D 1 1 0

Direccionam.: Par de reg. Indirecto

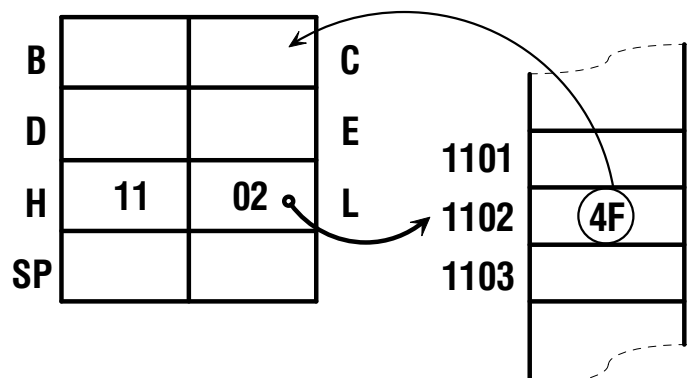
Señalizadores: Ninguno

Ciclos M.: 2      Estados: 7

### EJEMPLO: MOV A,B



### EJEMPLO: MOV C,M



# Instrucciones de transferencia II

## MOV M,r

(r) → (H)(L)

Formato:

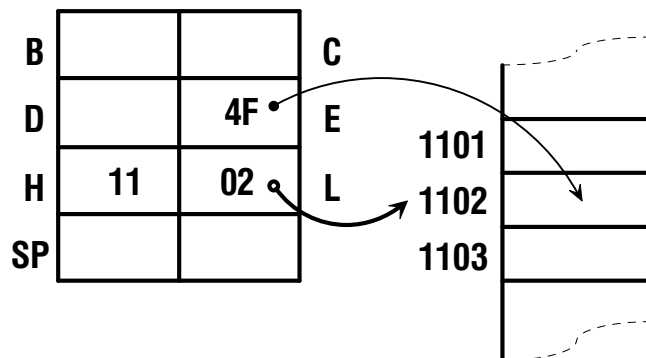
0 1 1 1 0 S S S

Direccionam.: Par de reg. Indirecto

Señalizadores: Ninguno

Ciclos M.: 2 Estados: 7

### EJEMPLO: MOV M,E



## MVI r,byte

2º Byte → r

Formato:

0 0 D D D 1 1 0

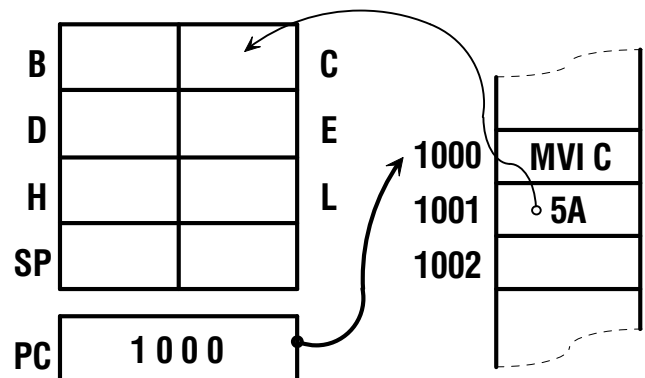
Operando

Direccionam.: Inmediato

Señalizadores: Ninguno

Ciclos M.: 2 Estados: 7

### EJEMPLO: MVI C,5AH



# Instrucciones de transferencia III

## MVI M,byte

2º Byte → (H)(L)

Formato:

0 0 1 1 0 1 1 0

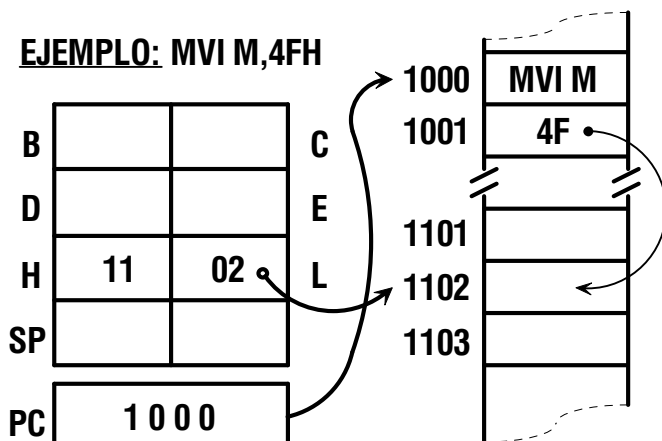
Operando

Direccionam.: Inmediato/P. Reg. Indirec.

Señalizadores: Ninguno

Ciclos M.: 3 Estados: 10

### EJEMPLO: MVI M,4FH



## LXI rp,dble

2º byte → (r<sub>L</sub>)

3º byte → (r<sub>H</sub>)

Formato:

0 1 1 1 0 S S S

2º Byte

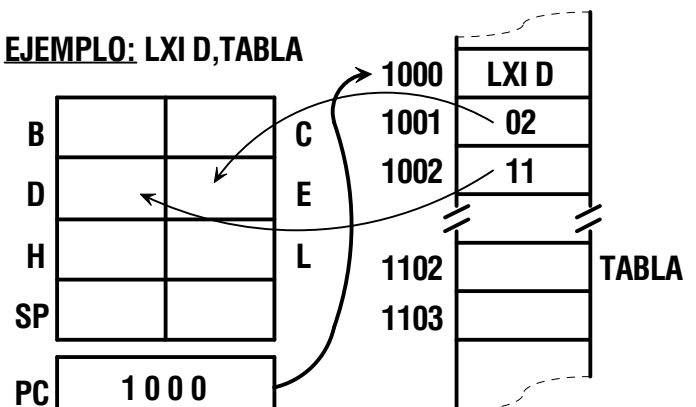
3º Byte

Direccionam.: Inmediato

Señalizadores: Ninguno

Ciclos M.: 3 Estados: 10

### EJEMPLO: LXI D,TABLA



# Instrucciones de transferencia IV

## LDA addr

$[(3^{\text{er}} \text{ Byte})(2^{\text{o}} \text{ Byte})] \rightarrow A$

Formato:

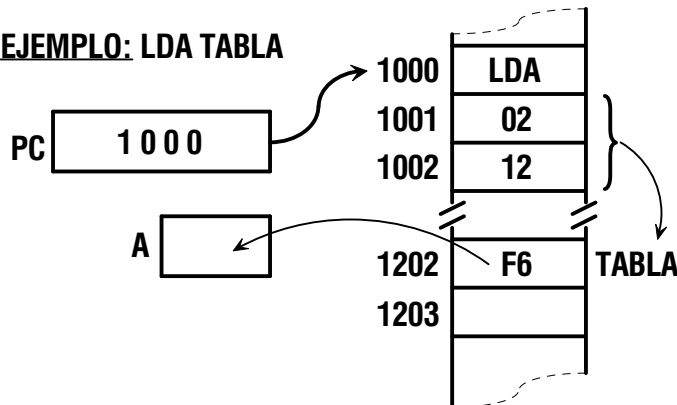
0 0 1 1 1 0 1 0
2 <sup>o</sup> Byte
3 <sup>er</sup> Byte

Direccionam.: Directo

Señalizadores: Ninguno

Ciclos M.: 4 Estados: 13

EJEMPLO: LDA TABLA



## STA addr

$(A) \rightarrow (3^{\text{er}} \text{ Byte})(2^{\text{o}} \text{ Byte})$

Formato:

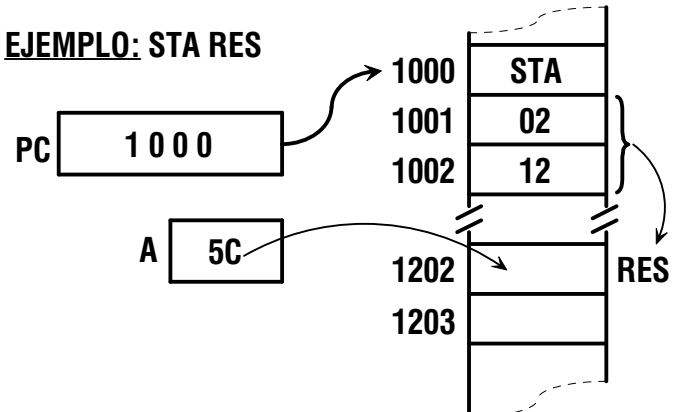
0 0 1 1 0 0 1 0
2 <sup>o</sup> Byte
3 <sup>er</sup> Byte

Direccionam.: Directo

Señalizadores: Ninguno

Ciclos M.: 4 Estados: 13

EJEMPLO: STA RES



# Instrucciones de transferencia V

## LHLD addr

$[(3^{\text{er}} \text{ Byte})(2^{\text{o}} \text{ Byte})] \rightarrow L$   
 $[(3^{\text{er}} \text{ Byte})(2^{\text{o}} \text{ Byte}) + 1] \rightarrow H$

Formato:

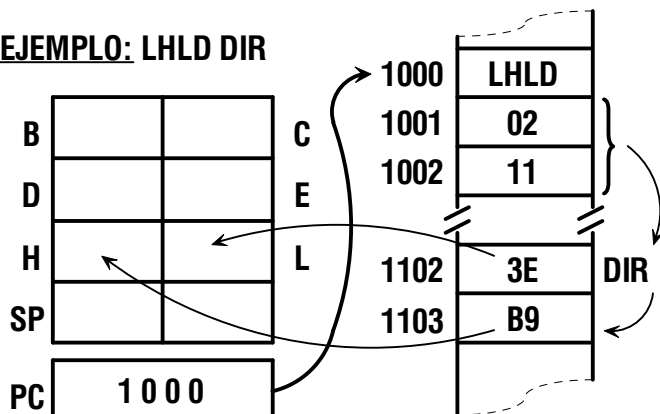
0 0 1 0 1 0 1 0
2 <sup>o</sup> Byte
3 <sup>er</sup> Byte

Direccionam.: Directo

Señalizadores: Ninguno

Ciclos M.: 5 Estados: 16

EJEMPLO: LHLD DIR



## SHLD addr

$(L) \rightarrow (3^{\text{er}} \text{ Byte})(2^{\text{o}} \text{ Byte})$   
 $(H) \rightarrow (3^{\text{er}} \text{ Byte})(2^{\text{o}} \text{ Byte}) + 1$

Formato:

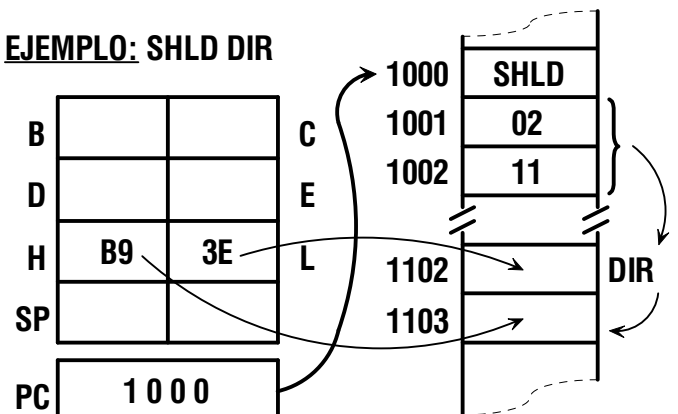
0 0 1 0 0 0 1 0
2 <sup>o</sup> Byte
3 <sup>er</sup> Byte

Direccionam.: Directo

Señalizadores: Ninguno

Ciclos M.: 5 Estados: 16

EJEMPLO: SHLD DIR



# Instrucciones de transferencia VI

## LDAX rp

$[(rp)] \rightarrow A$

Formato:

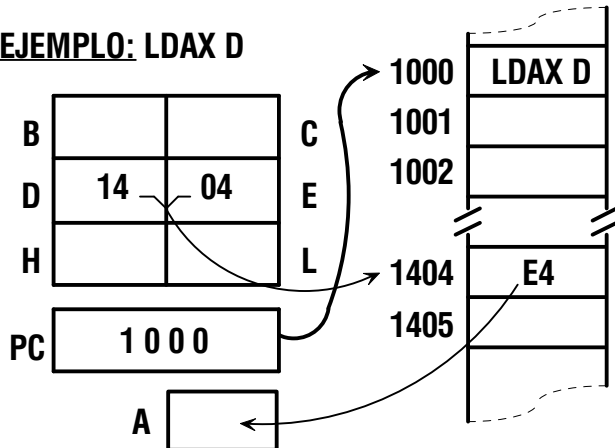
0 0 R P 1 0 1 0

Direccionam.: Par de reg. Indirecto

Señalizadores: Ninguno

Ciclos M.: 2 Estados: 7

### EJEMPLO: LDAX D



## STAX rp

$(A) \rightarrow (rp)$

Formato:

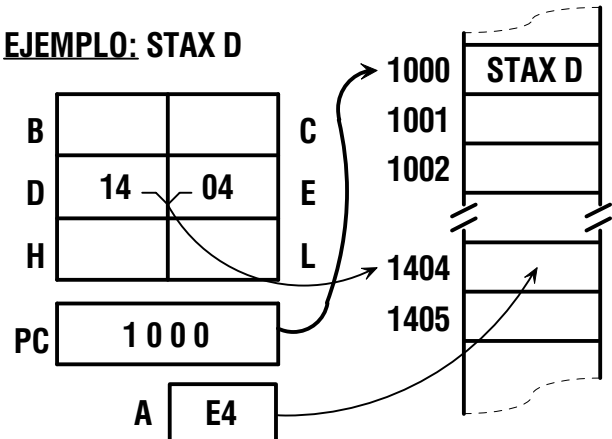
0 0 R P 0 0 1 0

Direccionam.: Par de reg. Indirecto

Señalizadores: Ninguno

Ciclos M.: 2 Estados: 7

### EJEMPLO: STAX D



# Instrucciones de transferencia VI

## XCHG

$(H) \leftrightarrow (D)$

$(L) \leftrightarrow (E)$

Formato:

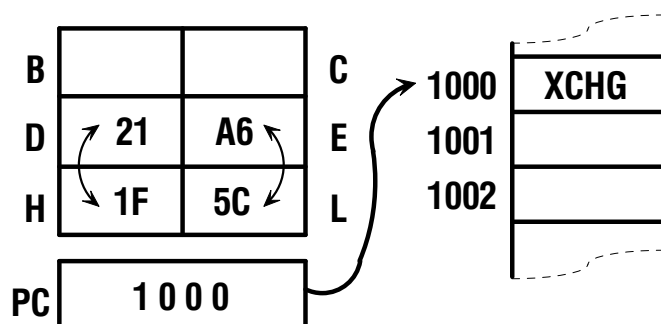
1 1 1 0 1 0 1 1

Direccionam.: Por registro

Señalizadores: Ninguno

Ciclos M.: 1 Estados: 4

### EJEMPLO: XCHG



# Instrucciones aritméticas I

## ADD / SUB r

$(A) + (R) \rightarrow A$   
 $(A) - (R) \rightarrow A$

## ADC / SBB r

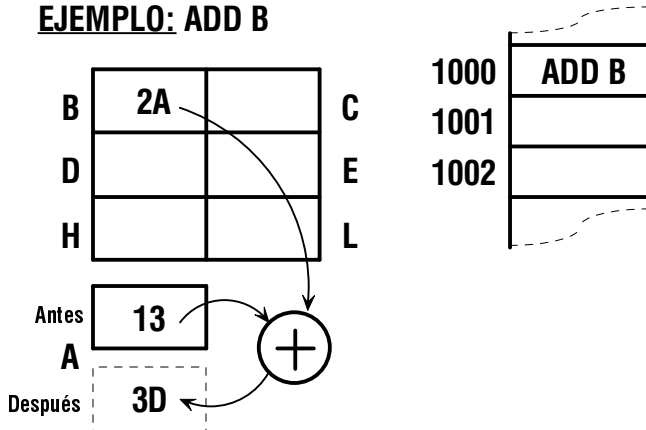
$(A) + (R) + C \rightarrow A$   
 $(A) - (R) - C \rightarrow A$

Direccinam.: Por registro

Señalizadores: Z S AC P CY

Ciclos M.: 1 Estados: 4

**EJEMPLO: ADD B**



## ADI / SUI byte

$(A) + 2^{\text{o}} \text{ byte} \rightarrow A$   
 $(A) - 2^{\text{o}} \text{ byte} \rightarrow A$

## ACI / SBI byte

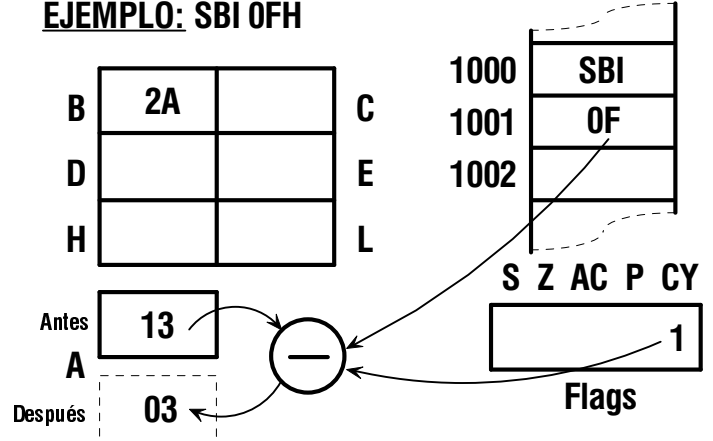
$(A) + 2^{\text{o}} \text{ byte} + C \rightarrow A$   
 $(A) - 2^{\text{o}} \text{ byte} - C \rightarrow A$

Direccinam.: Inmediato

Señalizadores: Z S AC P CY

Ciclos M.: 2 Estados: 7

**EJEMPLO: SBI 0FH**



# Instrucciones aritméticas II

## ADD / SUB M

$(A) + [(H)(L)] \rightarrow A$   
 $(A) - [(H)(L)] \rightarrow A$

## ADC / SBB M

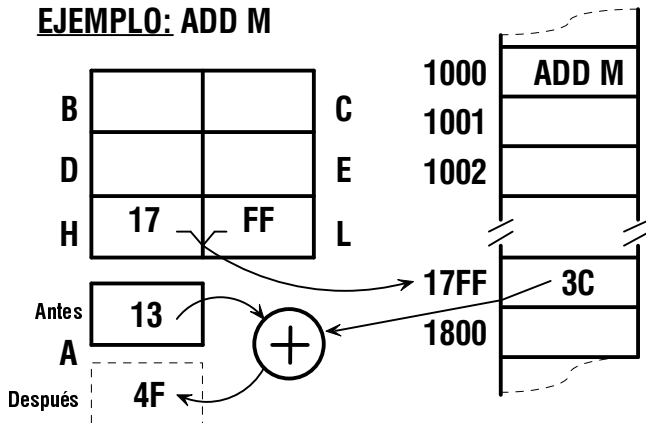
$(A) + [(H)(L)] + C \rightarrow A$   
 $(A) - [(H)(L)] - C \rightarrow A$

Direccinam.: Par de reg. Indirecto

Señalizadores: Z S AC P CY

Ciclos M.: 2 Estados: 7

**EJEMPLO: ADD M**



## DAA

Ajuste Decimal

Formato:

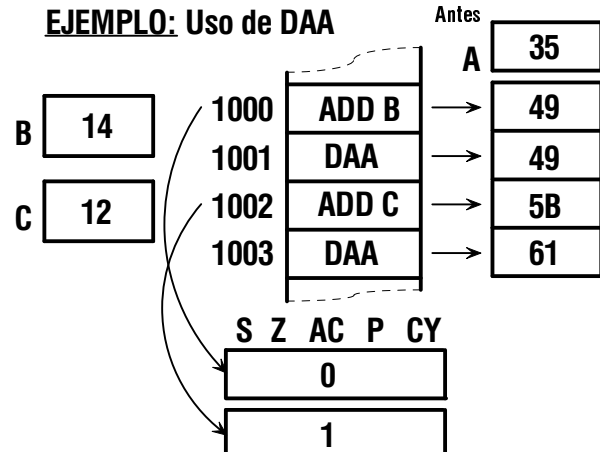
0 0 1 0 0 1 1 1

Direccinam.: Implícito

Señalizadores: Z S AC P CY

Ciclos M.: 1 Estados: 4

**EJEMPLO: Uso de DAA**



## Instrucciones aritméticas III

**INR r**

$(R) + 1 \rightarrow R$

**DCR r**

$(R) -- 1 \rightarrow R$

Direccionam.: Por registro

Señalizadores: Z S AC P

Ciclos M.: 1 Estados: 4

**INR M**

$[(H)(L)] + 1 \rightarrow (H)(L)$

**DCR M**

$[(H)(L)] - 1 \rightarrow (H)(L)$

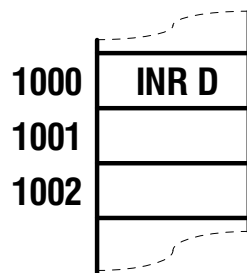
Direccionam.: Par reg. Indirecto

Señalizadores: Z S AC P

Ciclos M.: 3 Estados: 10

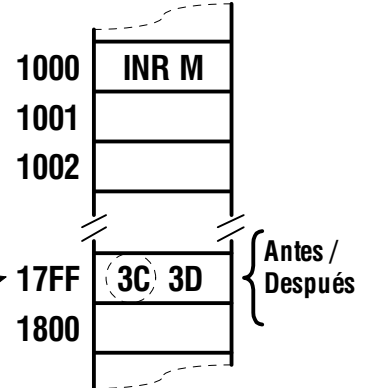
### EJEMPLO: INR D

D	2A	Antes
D	2B	Después



### EJEMPLO: INR M

B		C
D		E
H	17	L



## Instrucciones aritméticas IV

**INX rp**

$(rp) + 1 \rightarrow rp$

**DCX rp**

$(rp) - 1 \rightarrow rp$

Direccionam.: Por registro

Señalizadores: Ninguno

Ciclos M.: 1 Estados: 6

**DAD rp**

$(HL) + (rp) \rightarrow HL$

Formato:

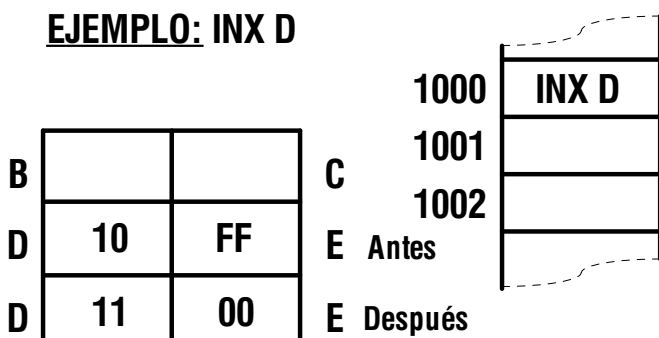
0 0 R P 1 0 0 1

Direccionam.: Por registro

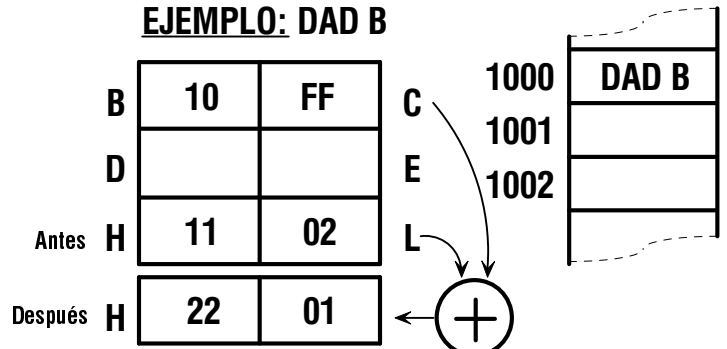
Señalizadores: C

Ciclos M.: 3 Estados: 10

### EJEMPLO: INX D



### EJEMPLO: DAD B





# Instrucciones lógicas I

## ANA / XRA r

$(A) \cdot (R) \rightarrow A$   
 $(A) \oplus (R) \rightarrow A$

## ORA r

$(A) + (R) \rightarrow A$

Direccionam.: Por registro  
 Señalizadores: Z S P; AC=1 CY=0  
 Ciclos M.: 1 Estados: 4

## ANA / XRA M

$(A) \cdot [(H)(L)] \rightarrow A$   
 $(A) \oplus [(H)(L)] \rightarrow A$

## ORA M

$(A) + [(H)(L)] \rightarrow A$

Direccionam.: Par reg. Indirecto  
 Señalizadores: Z S P; AC=1 CY=0  
 Ciclos M.: 2 Estados: 7

## ANI / XRI byte

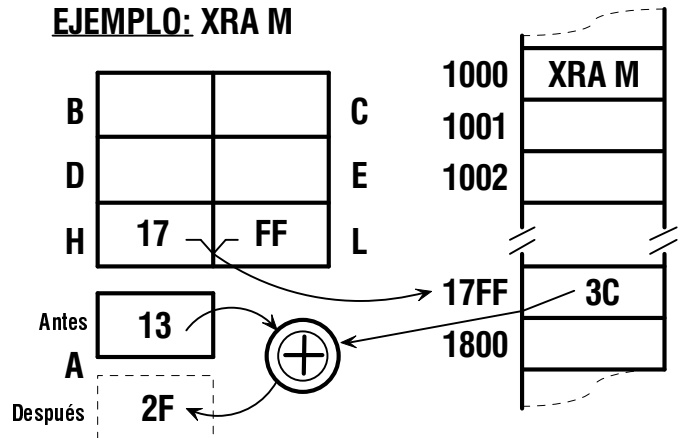
$(A) \cdot \text{byte} \rightarrow A$   
 $(A) \oplus \text{byte} \rightarrow A$

## ORI byte

$(A) + \text{byte} \rightarrow A$

Direccionam.: Inmediato  
 Señalizadores: Z S P; AC=1 CY=0  
 Ciclos M.: 2 Estados: 7

### EJEMPLO: XRA M



# Instrucciones lógicas II

## CMP r

$(A) - (R)$

Direccionam.: Por registro  
 Ciclos M.: 1 Estados: 4

## CPI byte

$(A) - 2^{\text{o}} \text{ byte}$

Direccionam.: Inmediato  
 Ciclos M.: 2 Estados: 7

## CMP M

$(A) - [(H)(L)]$

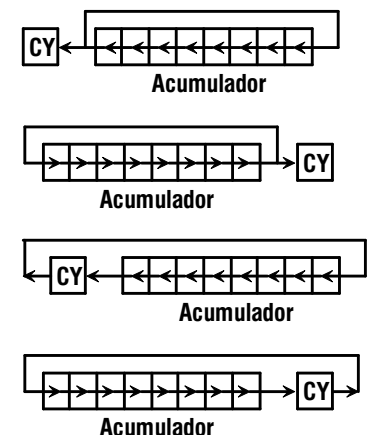
Ciclos M.: 2 Estados: 7  
 Direccionam.: Par reg. Indirecto  
 Señalizadores: Z S P AC CY

## RLC

## RRC

## RAL

## RAR



Longitud: 1 Byte (Opcode)  
 Direccionam.: Implícito  
 Señalizadores: CY  
 Ciclos M.: 1 Estados: 4

# Instrucciones lógicas III

## CMA

$(\bar{A}) \rightarrow A$

Formato:

0 0 0 1 1 1 1 1

Direccionam.: Implícito

Señalizadores: Ninguno

Ciclos M.: 1 Estados: 4

## CMC

$\bar{C} \rightarrow C$

## STC

$1 \rightarrow C$

Señalizadores: CY

Direccionam.: Implícito

Ciclos M.: 1 Estados: 4

### EJEMPLO: CMA

A	2A	Antes
A	D5	Después

1000	CMA
1001	
1002	

### EJEMPLO: Uso de CMC y STC

	S	Z	AC	P	CY	
A					1	Antes
1000					0	
1001					1	
1002						
1003						

# Instrucciones de ruptura de secuencia

- Hay tres clases de instrucciones para la ruptura de la secuencia de un programa:

- De salto* : Este tipo de instrucciones saltan a cualquier instrucción del programa.
- De llamada a subrutina* : Son instrucciones de salto que guardan la dirección de retorno (dirección de la siguiente instrucción) en pila.
- De retorno* : Originan un salto a la instrucción cuya dirección está almacenada en la pila.

- Todas estas instrucciones pueden ser:

- Incondicionales* : Siempre se produce el salto o la bifurcación.
- Condicionales* : El salto se produce si se cumple la condición especificada en la instrucción. Las posibles condiciones están resumidas en la tabla adjunta.

CCC	Símbolo	Condición
0 0 0	NZ	NO ZERO ( Z = 0 )
0 0 1	Z	ZERO ( Z = 1 )
0 1 0	NC	NO ACARREO ( CY = 0 )
0 1 1	C	ACARREO ( CY = 1 )
1 0 0	PO	PARIDAD IMPAR ( P = 0 )
1 0 1	PE	PARIDAD PAR ( P = 1 )
1 1 0	P	POSITIVO ( S = 0 )
1 1 1	M	NEGATIVO ( S = 1 )

# Instrucciones de ruptura de secuencia I

## JMP Addr

(3<sup>er</sup> byte) (2<sup>o</sup> byte) → PC

Formato:

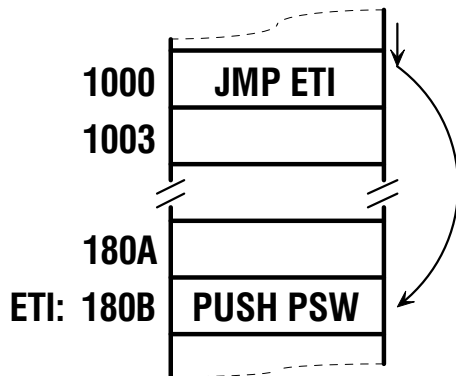
11000011 Addr(2)

Direccionam.: Inmediato

Señalizadores: Ninguno

Ciclos M.: 3 Estados: 10

### EJEMPLO: JMP ETI



## Jccc Addr

Si CCC

(3<sup>er</sup> byte)(2<sup>o</sup> byte) → PC

Si no (PC) + 2 → PC

Formato:

11CCCC010 Addr(2)

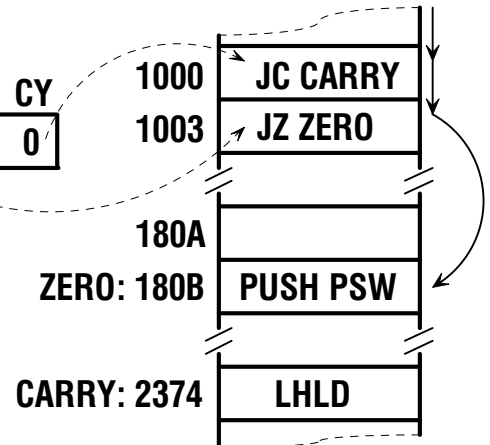
Direccionam.: Inmediato

Señalizadores: Ninguno

Ciclos M.: 2/3 Estados: 7/10

### EJEMPLOS:

S	Z	AC	P	CY
1	0			0



# Instrucciones de ruptura de secuencia II

## CALL Addr

(PC<sub>L</sub>) → (SP) - 1

(PC<sub>L</sub>) → (SP) - 2

(SP) - 2 → SP

(3<sup>er</sup> byte)(2<sup>o</sup> byte) → PC

Formato:

11001101 Addr (2)

Direccionam.: Inmediato

Señalizadores: Ninguno

Ciclos M.: 5 Estados: 18

## Cccc Addr

Si CCC

Idem CALL

Si no (PC) + 2 → PC

Formato:

11CCCC100 Addr(2)

Direccionam.: Inmediato

Señalizadores: Ninguno

Ciclos M.: 2/5 Estados: 9/18

### EJEMPLOS:

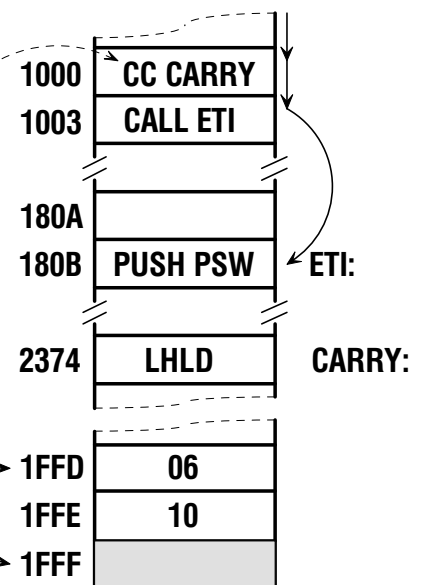
S	Z	AC	P	CY
				0

Stack Pointer

1 F F F H

Después de  
CALL ETI

Antes de  
CALL ETI



# Instrucciones de ruptura de secuencia III

## RET

$[(SP)] \rightarrow PC_L$   
 $[(SP)+1] \rightarrow PC_H$   
 $(SP) - 2 \rightarrow SP$

Formato:

1 1 0 0 1 0 0 1

Direccionam.: Implícito

Señalizadores: Ninguno

Ciclos M.: 3 Estados: 10

## Rccc

Si CCC  
 Idem RET  
 Si no  $(PC) + 2 \rightarrow PC$

Formato:

1 1 C C C 0 0 0

Direccionam.: Implícito

Señalizadores: Ninguno

Ciclos M.: 1/3 Estados: 6/12

### EJEMPLOS:

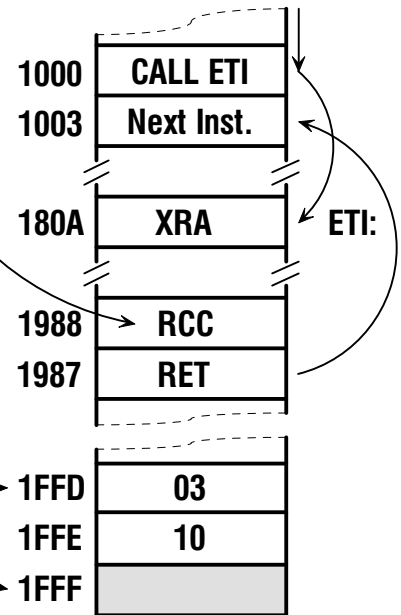
S Z AC P CY  
 0

Stack Pointer

1 F F D H

Antes de RET

Después de RET



# Instrucciones de ruptura de secuencia IV

## PCHL

$(H) \rightarrow PC_H; (L) \rightarrow PC_L$

Formato:

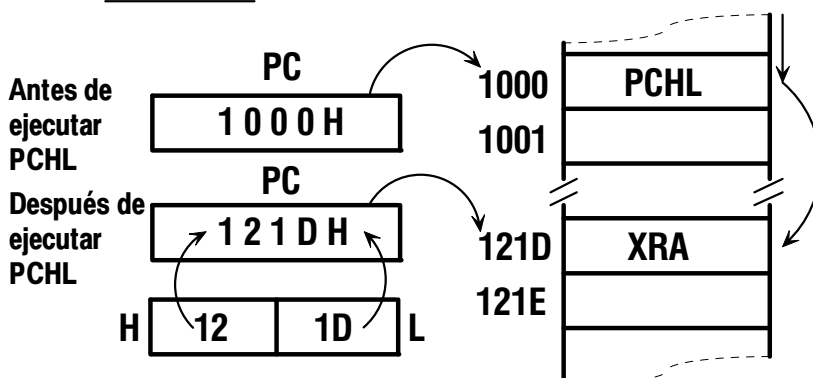
1 1 1 0 1 0 0 1

Direccionam.: Implícito / Por registro

Señalizadores: Ninguno

Ciclos M.: 1 Estados: 6

### EJEMPLO: PCHL



# Instrucciones de manejo de la Pila e I/O I

## PUSH rp

$(R_H) \rightarrow (SP) - 1$   
 $(R_L) \rightarrow (SP) - 2$   
 $(SP) - 2 \rightarrow SP$

Formato:

1 1 R P 0 1 0 1

Direccionam.: Implícito

Señalizadores: Ninguno

Ciclos M.: 3 Estados: 12

## PUSH PSW

$(A) \rightarrow (SP) - 1$   
 $(R_{cc}) \rightarrow (SP) - 2$   
 $(SP) - 2 \rightarrow SP$

Formato:

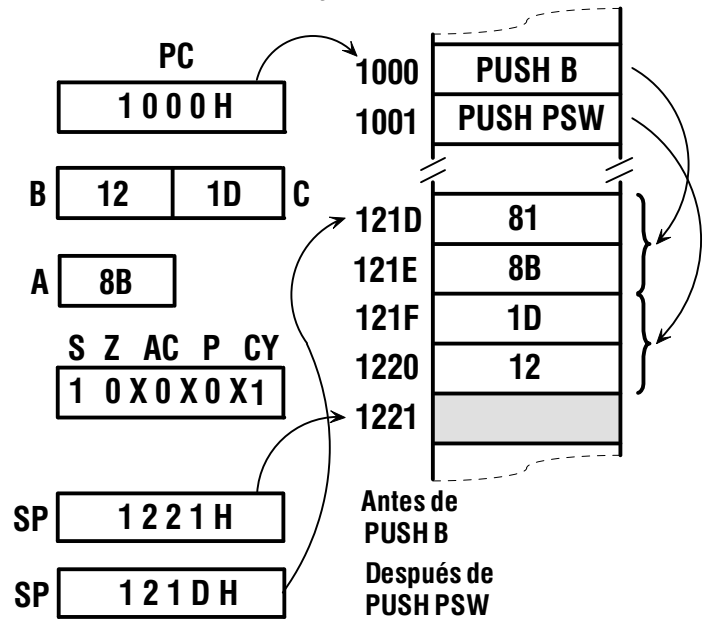
1 1 1 1 0 1 0 1

Direccionam.: Implícito

Señalizadores: Ninguno

Ciclos M.: 3 Estados: 12

### EJEMPLO: PUSH B y PUSH PSW



# Instrucciones de manejo de la Pila e I/O II

## POP rp

$[(SP)] \rightarrow R_L$   
 $[(SP) + 1] \rightarrow R_H$   
 $(SP) + 2 \rightarrow SP$

Formato:

1 1 R P 0 0 0 1

Direccionam.: Implícito

Señalizadores: Ninguno

Ciclos M.: 3 Estados: 10

## POP PSW

$[(SP)] \rightarrow R_{cc}$   
 $[(SP) + 1] \rightarrow A$   
 $(SP) + 2 \rightarrow SP$

Formato:

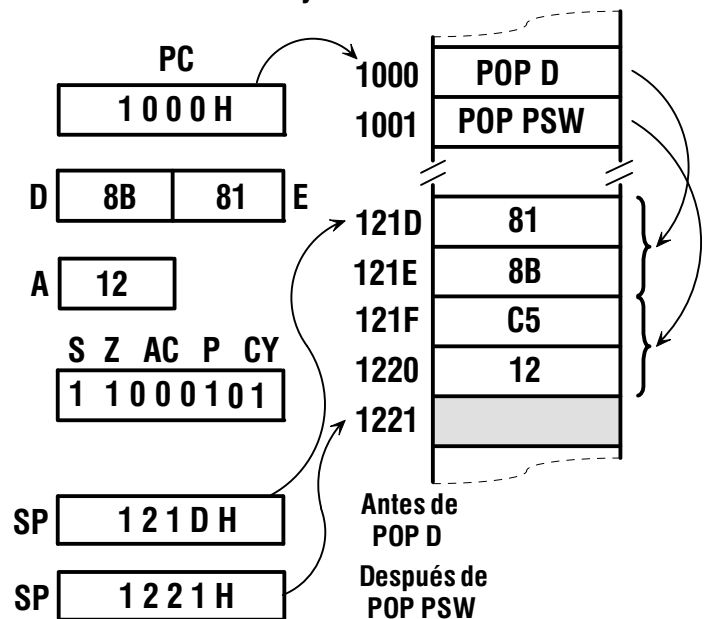
1 1 1 1 0 0 0 1

Direccionam.: Implícito

Señalizadores: Todos

Ciclos M.: 3 Estados: 10

### EJEMPLO: POP D y POP PSW



# Instrucciones de manejo de la Pila e I/O III

## XTHL

$[(SP)] \leftarrow \rightarrow L$   
 $[(SP) + 1] \leftarrow \rightarrow H$

Formato:

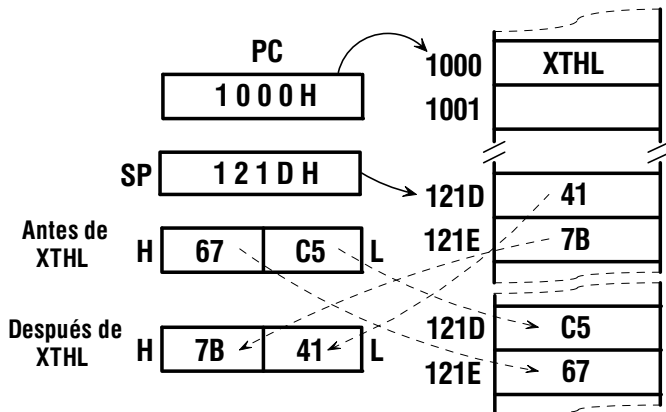
1 1 1 0 0 0 1 1

Direccionam.: Implícito

Señalizadores: Ninguno

Ciclos M.: 5 Estados: 16

### EJEMPLO: XTHL



## SPHL

$(H) \rightarrow SP_H$   
 $(L) \rightarrow SP_L$

Formato:

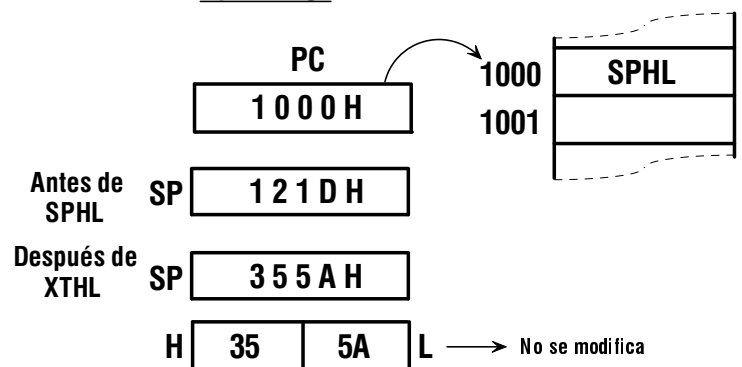
1 1 1 1 1 0 0 1

Direccionam.: Implícito

Señalizadores: Ninguno

Ciclos M.: 1 Estados: 6

### EJEMPLO: SPHL



# Instrucciones de manejo de la Pila e I/O IV

## IN Port

$(\text{Puerto}) \rightarrow A$

Formato:

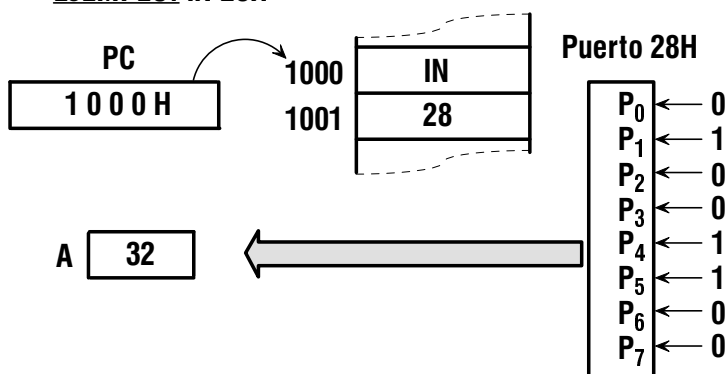
11011011 Port(1)

Direccionam.: Directo

Señalizadores: Ninguno

Ciclos M.: 3 Estados: 10

### EJEMPLO: IN 28H



## OUT Port

$(A) \rightarrow \text{Puerto}$

Formato:

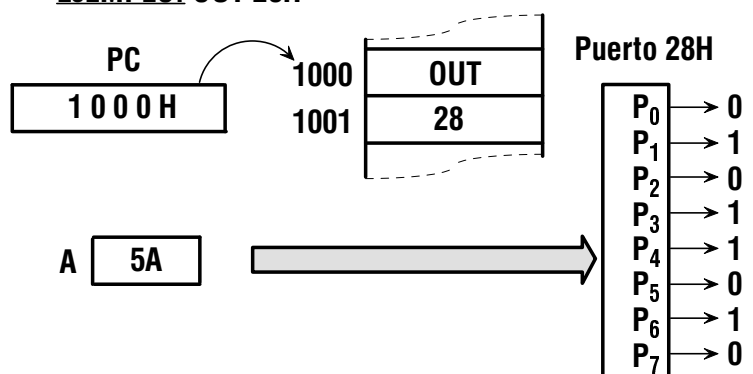
11010011 Port(1)

Direccionam.: Directo

Señalizadores: Ninguno

Ciclos M.: 3 Estados: 10

### EJEMPLO: OUT 28H



# Instrucciones de control I

**EI**

Habilita las interrupciones

Formato:

1 1 1 1 1 0 1 1

Direccionam.: Ninguno

Señalizadores: Ninguno

Ciclos M.: 1 Estados: 4

**DI**

Deshabilita las Interrupciones

Formato:

1 1 1 1 1 0 1 1

Direccionam.: Ninguno

Señalizadores: Ninguno

Ciclos M.: 1 Estados: 4

**NOP**

No Operación

Formato:

0 0 0 0 0 0 0 0

Direccionam.: Ninguno

Señalizadores: Ninguno

Ciclos M.: 1 Estados: 4

**HLT**

Para el microprocesador

Formato:

0 1 1 1 0 1 1 0

Direccionam.: Ninguno

Señalizadores: Ninguno

Ciclos M.: 1 Estados: 5

# Instrucciones de control I

**SIM**

Posiciona las máscaras de interrupción.

Formato:

0 0 1 1 0 0 0 0

Direccionam.: Ninguno

Señalizadores: Ninguno

Ciclos M.: 1 Estados: 4

**RIM**

Lectura de las máscaras de interrupción.

Formato:

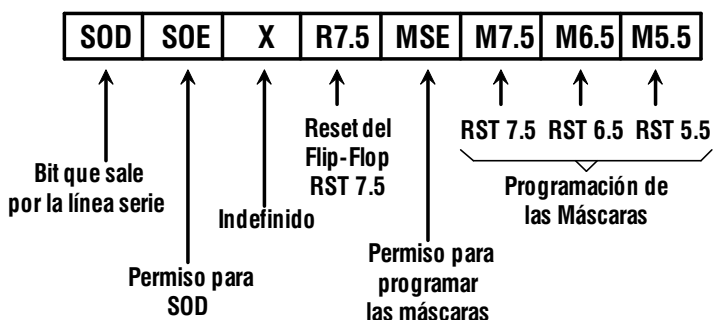
0 0 1 0 0 0 0 0

Direccionam.: Ninguno

Señalizadores: Ninguno

Ciclos M.: 1 Estados: 4

Significado del contenido del acumulador antes de SIM



Contenido del acumulador tras ejecutar la instrucción RIM

