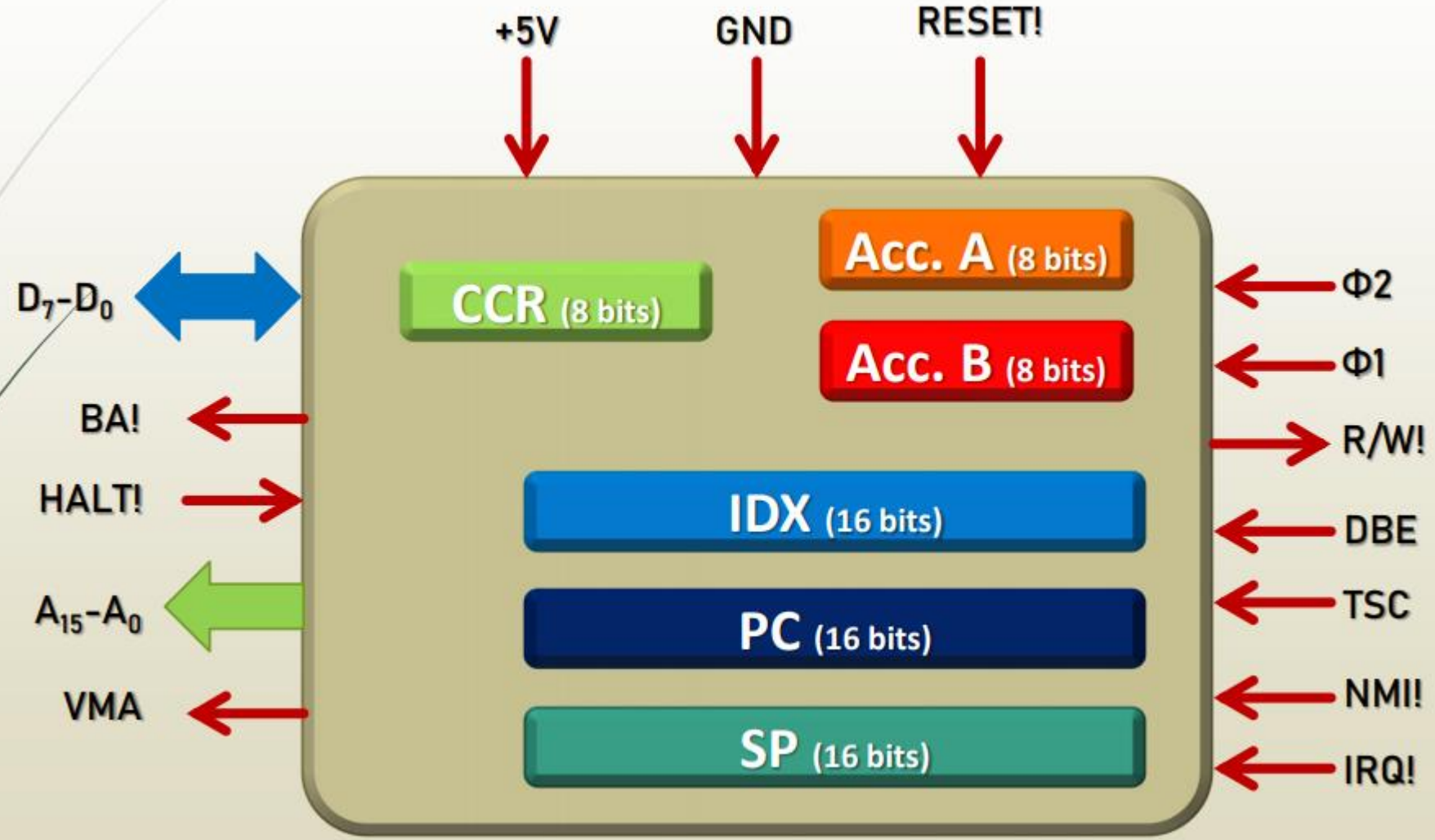
Facultad de Ingeniería
Universidad Nacional de Jujuy

TÉCNICAS Y ESTRUCTURAS DIGITALES

Microprocesador Motorola 6800

Programación M6800 → Arquitectura M6800



Programación M6800 → Modos de Direccionamiento 6800

- Formato de las instrucciones

- Los modos de direccionamiento:
  - Implicado

  - Inmediato

#$Hex #%Bin

  - Directo

$00 $FF

  - Extendido

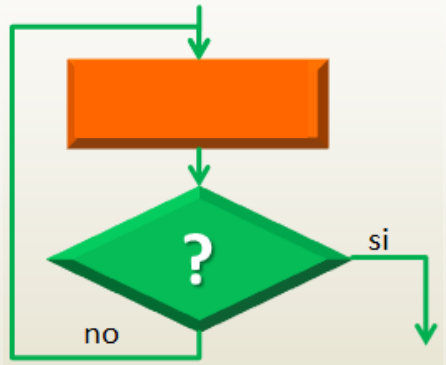
  - Indexado

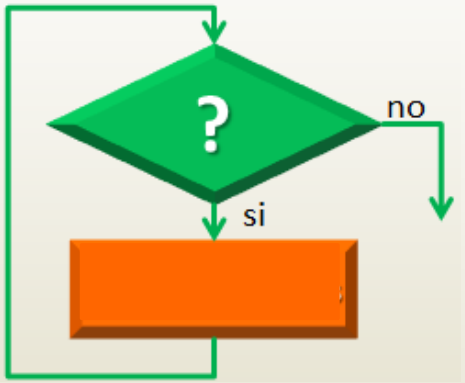
Desplaz,X

  - Relativo

Estructura Condicional e Iterativa

- Uso de instrucciones de comparación combinadas con instrucciones de salto, o instrucciones de cálculo que alteran los bits del CCR combinadas con saltos.





COMPARACIÓN

CMPA	(A - M)
CMPB	(B - M)
CPX	(X_H / X_L) – M / M + 1)
CBA	(A - B)
TST	(M - 00)
TSTA	(A - 00)
TSTB	(B - 00)

BIFURCACION

INCONDICIONAL

BRA

CONDICIONALES

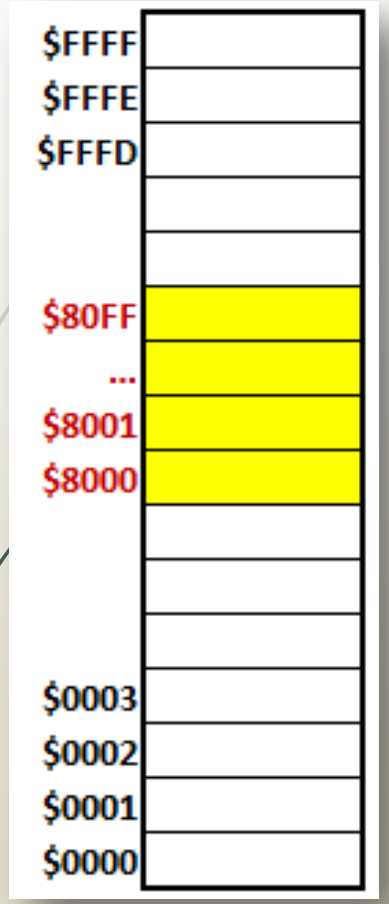
BCC Branch if C = 0

BCS Branch if C = 1

BEQ Branch if 0

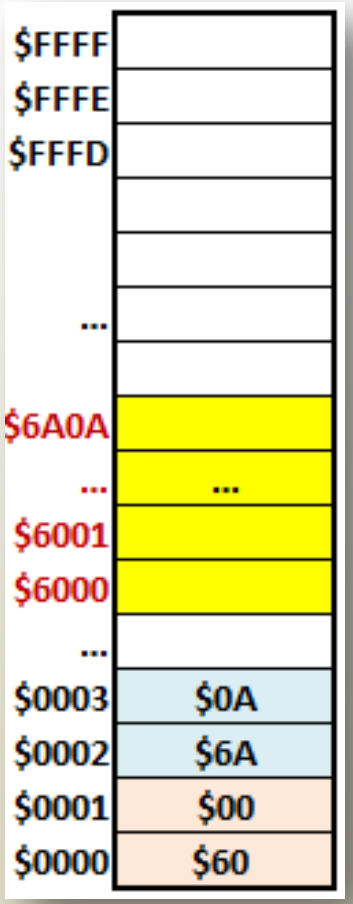
BGE Branch if grather or equal 0

Recorridos de porciones de memoria



LDX #$8000

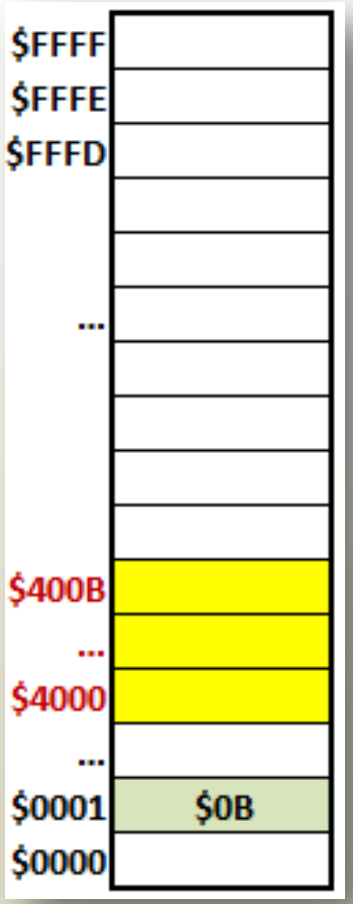
IDX=$8000

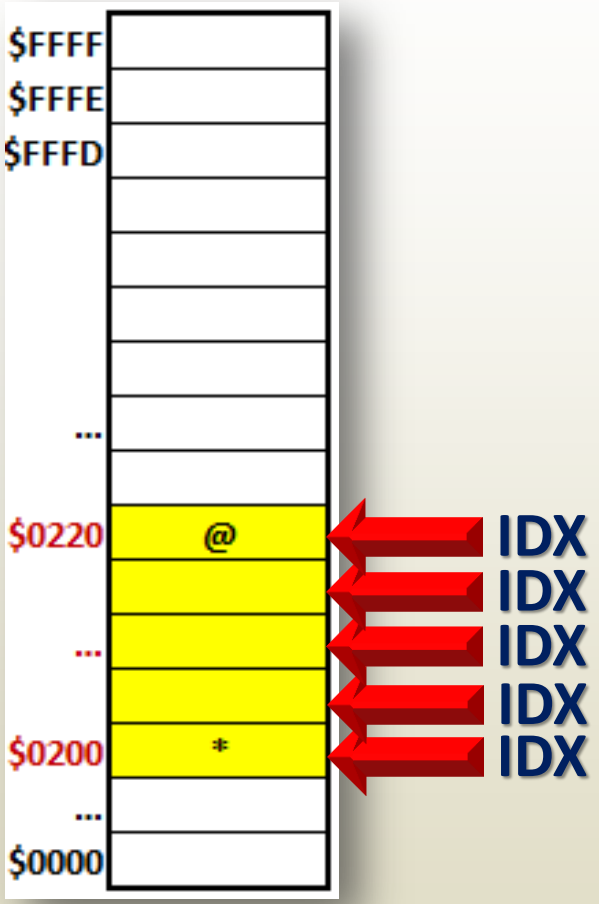


LDX $0000

LDX $00

IDX=$6000





Operaciones de Reg. Índice:
LDX - CPX - INX - DEX

Operaciones de Registro Índice

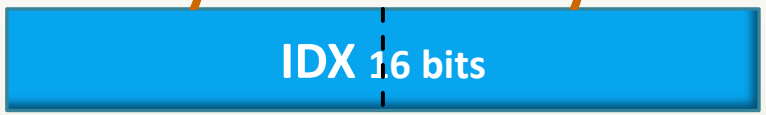
CPX $(X_H / X_L) - (M / M + 1)$

LDX $M \rightarrow X_H, (M + 1) \rightarrow X_L$

STX $X_H \rightarrow M, X_L \rightarrow (M + 1)$

H high (Alto) L low (bajo)

1 byte 1 byte



$0007	$31
$0006	$20
$0005	$40
$0004	$00
$0003	$30
$0002	$45
$0001	$20
$0000	$40

LDX #$0040 → IDX = $0040

LDX $02 → IDX = $4530

LDX $0005 → IDX = $4020

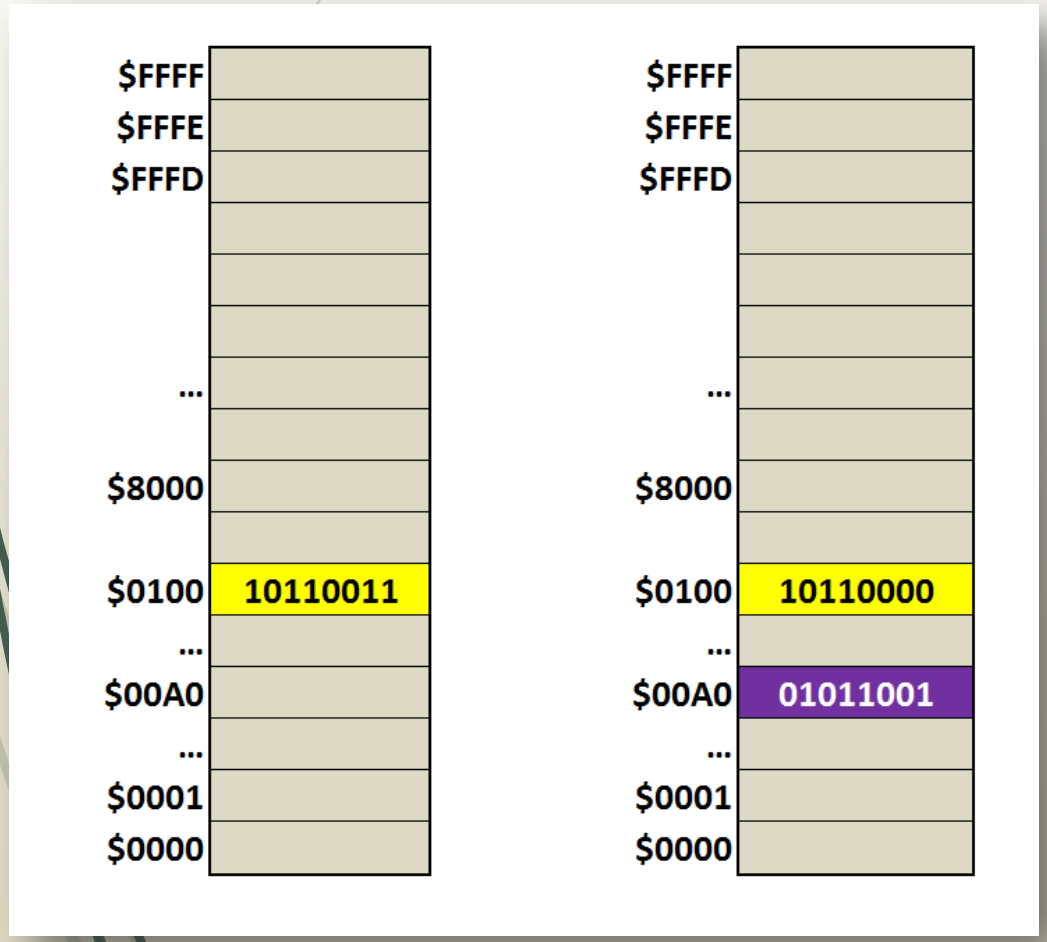
CPX #$4020 → VERDADERO

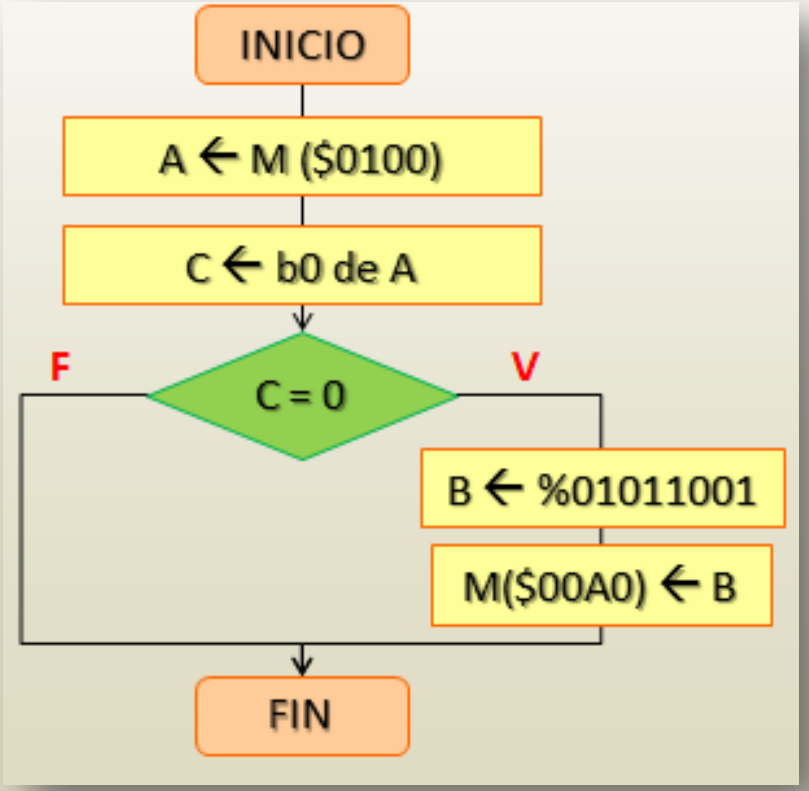
CPX $0003 → FALSO

STX $0000 → IDX = $4020

Ejemplos (1)

- Escriba un programa que analice si el contenido de la dirección de memoria $0100 es un número par. El resultado del análisis se guardará en la dirección $00A0: %01011001 (ASCII de “Y”, YES) solamente si se cumple la condición. Utilice etiquetas para los saltos.



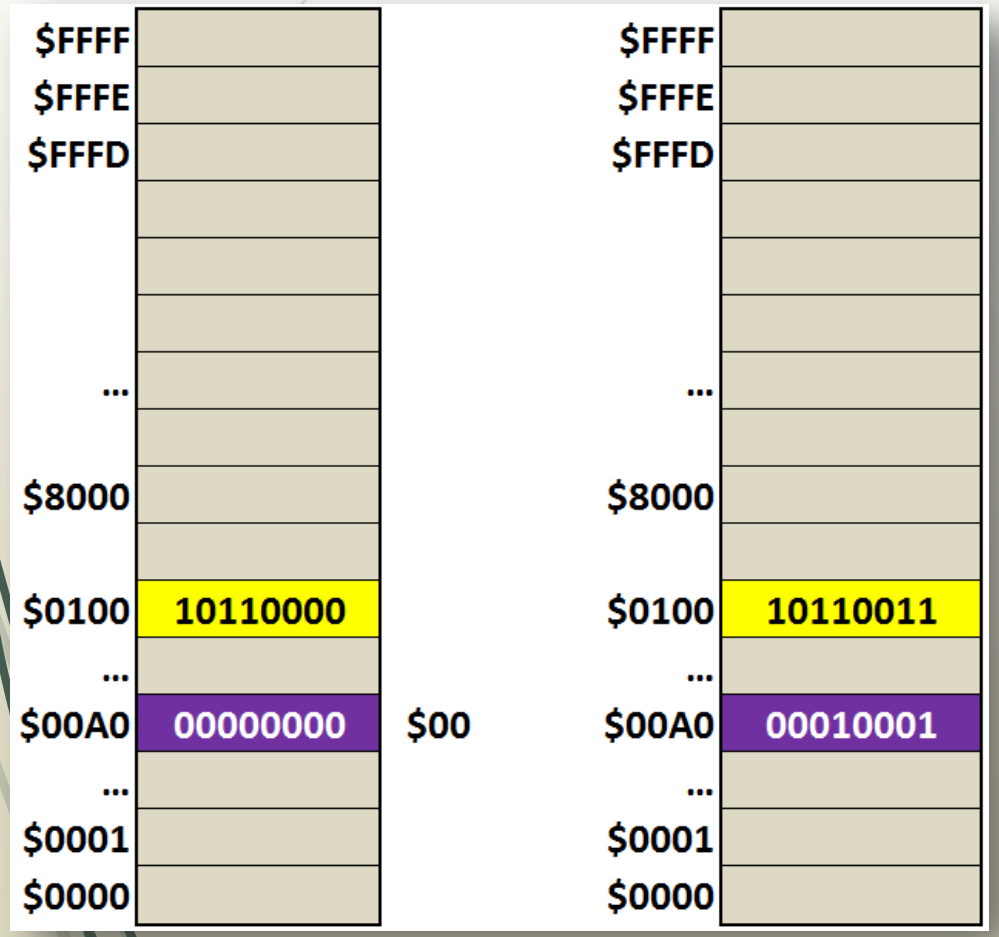


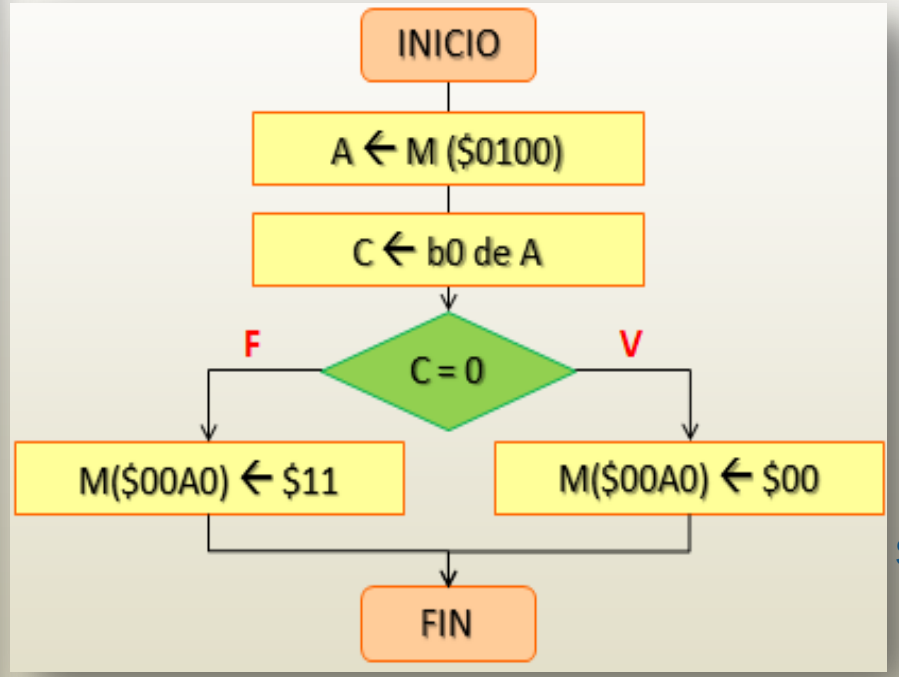

```
      LDAA $0100
      RORA
      BCS   fin
      LDAB #$59
      STAB $A0
fin   SWI
```

BCS salto si C=1

Ejemplos (2)

- Escriba un programa que determine si el contenido en la dirección de memoria $0100, es par o impar. El resultado del análisis se guardará en $00A0: $00 para nº par, $11 para nº impar. Utilice etiquetas para los saltos.



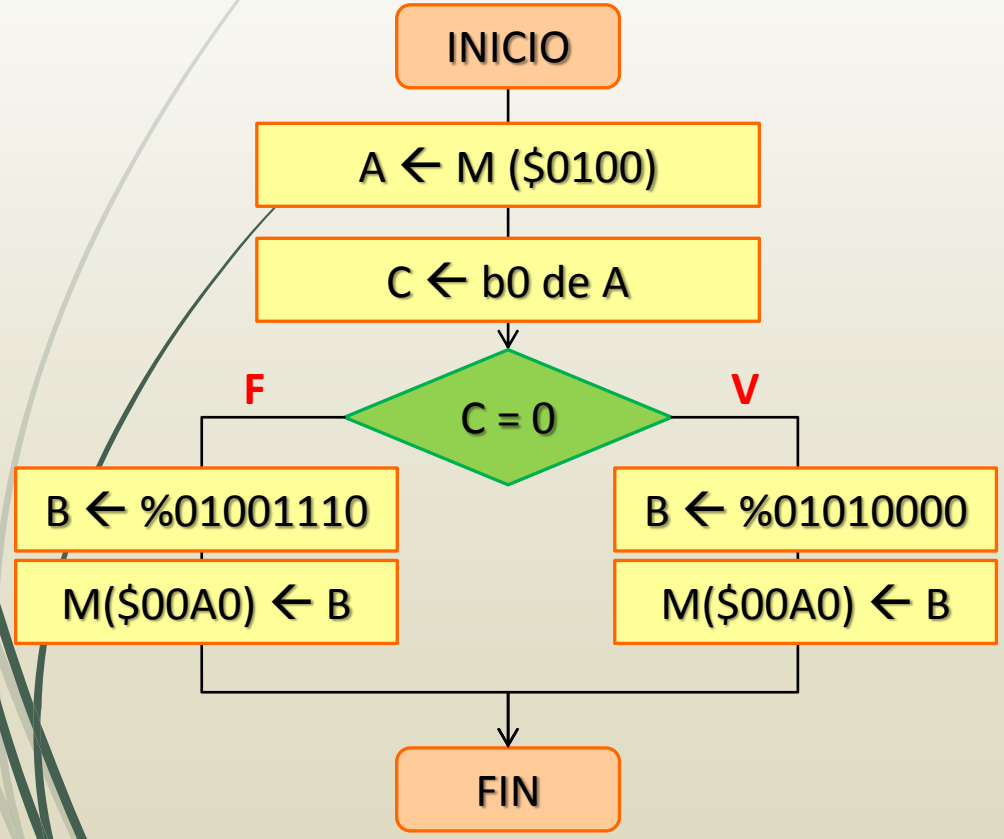



```
     LDAA $0100
     RORA
     BCC    saltoP
     LDAA #$11
     STAA $00A0
     BRA    fin
saltoP LDAA #$00
     STAA $00A0
fin  SWI
```

BCC salto si C=0

Ejemplos (3)

- Escriba un programa que determine si el contenido de la dirección de memoria $0100, es par o impar. El resultado del análisis se guardará en $00A0: ASCII de “P” (par), ASCII de “I” (impar). Utilice etiquetas para los saltos.

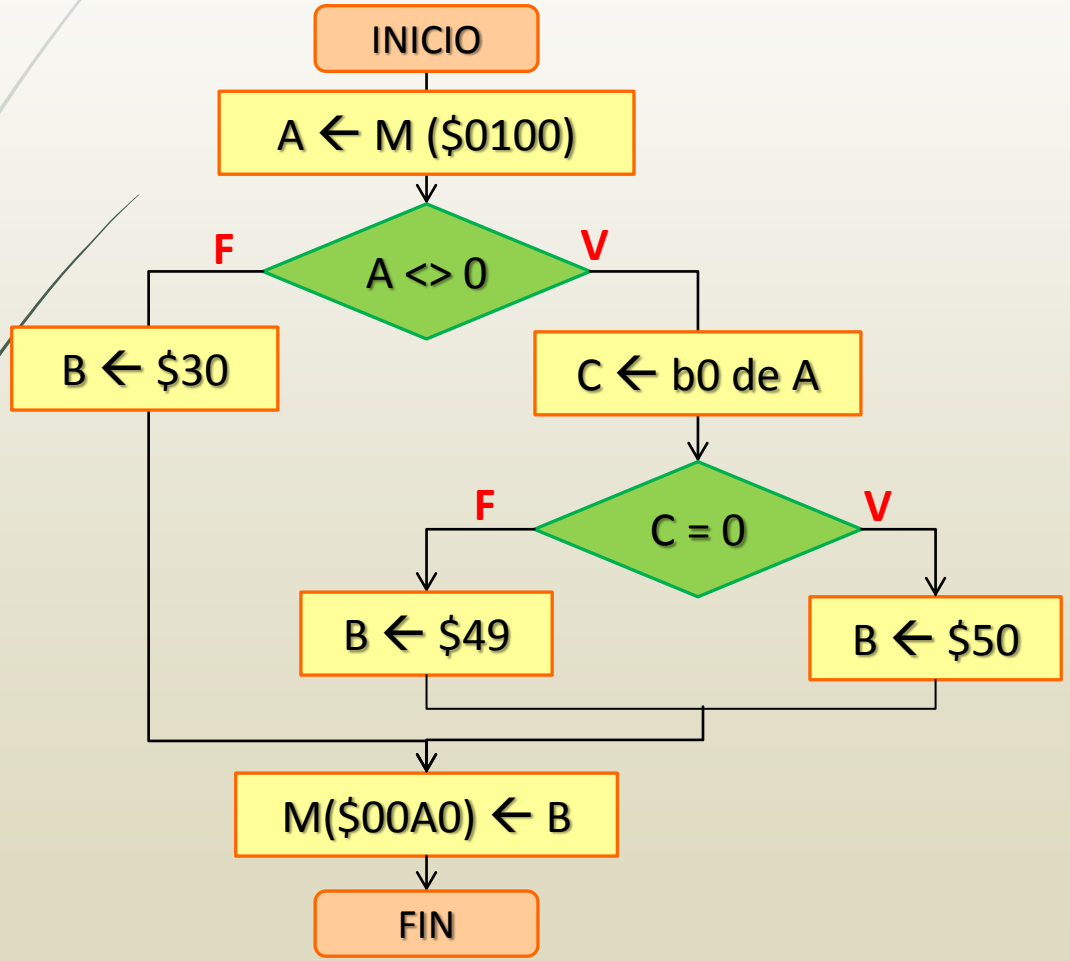



```
        LDAA $0100
        RORA
        BCC     saltoP
        LDAB #$4E
        STAB $00A0
        BRA     fin
saltoP  LDAB #$50
        STAB $00A0
fin     SWI
```

```
        LDAA $0100
        RORA
        BCC     saltoP
        LDAB #$4E
        BRA     fin
saltoP  LDAB #$50
fin     STAB $00A0
        SWI
```

Ejemplos (4)

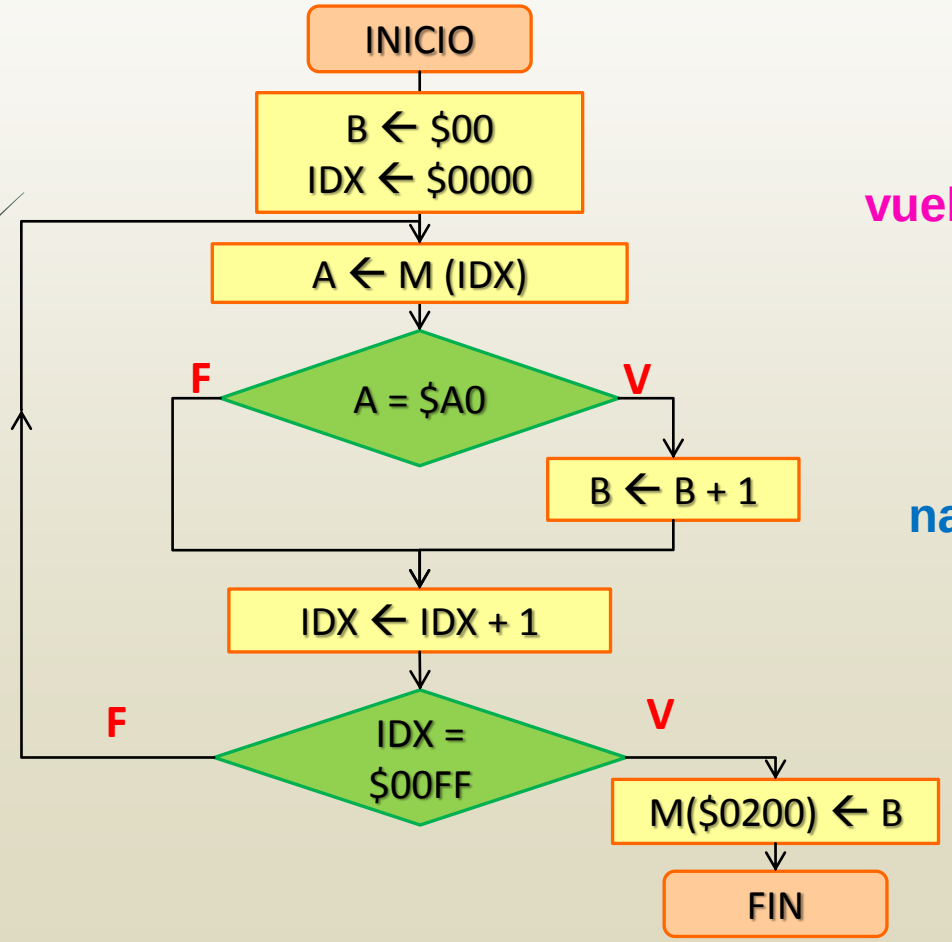
- Escriba un programa que determine si el contenido en la dirección de memoria $0100, es cero, par o impar. El resultado del análisis se guardará en $00A0: ASCII de “P” (par), ASCII de “I” (impar) y el ASCII de “0” para 0 (cero)




```
        LDAA $0100
        CMPA #$00
        BEQ s0
        RORA
        BCC sP
        LDAB #$49
        BRA guarda
sP      LDAB #$50
        BRA guarda
s0      LDAB #$30
guarda  STAB $A0
        SWI
```

Ejercicio (5)

- Escriba un programa que cuente las palabras vacías (“ ” - espacio) guardadas en la primera página de memoria. El valor de cuenta debe almacenarse en la posición $0200. Utilice etiquetas para los saltos.




```
        CLRB
        LDX #$0000
vuelve  LDAA $00,X
        CMPA #$A0
        BNE nada
        INCB
nada    INX
        CPX #$0100
        BNE vuelve
        STAB $0200
        SWI
```

```
        CLRB
        LDX #$0000
vuelve  LDAA $00,X
        CMPA #$A0
        BNE nada
        INCB
nada    INX
        CPX #$00FF
        BLE vuelve
        STAB $0200
        SWI
```

2 opciones, ver CPX y el salto

Ejercicio (6)

- Escriba un programa que reemplace los valores negativos impares de una porción de memoria por el valor registrado en $08FF. El segmento de memoria inicia en la dirección $B000 y su longitud se encuentra $0900. Además, indique la cantidad de reemplazos realizados en la posición $0000.

Planteo

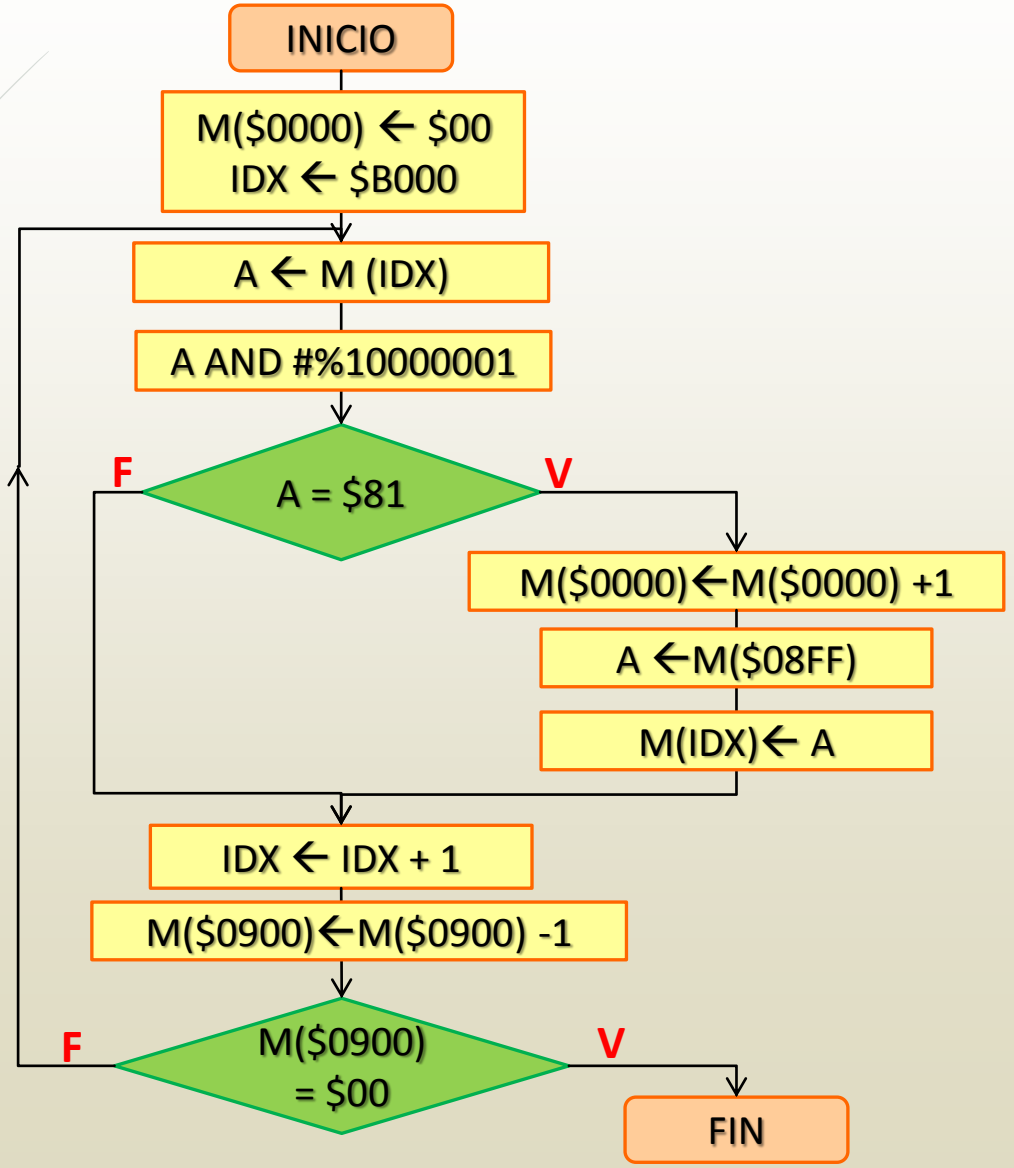
ANTES

MP	Direcciones	
	$B00C	
...	$B00B	
11111111	$B00A	
00011111	$B009	
01111110	$B008	
00000000	$B007	
11100101	$B006	
10101010	$B005	
01010101	$B004	
11110001	$B003	
00001111	$B002	
10011101	$B001	
10000000	$B000	<< IDX
...		
$0A	$0900	Cant.elementos a recorrer
VALOR	$08FF	Valor de reemplazo
...		
	$0000	

DESPUÉS

MP	Direcciones		Valores de $0900
	$B00C		
...	$B00B		
11111111	$B00A	<< IDX	$00
00011111	$B009		$01
01111110	$B008		$02
00000000	$B007		$03
VALOR	$B006		$04
10101010	$B005		$05
01010101	$B004		$06
VALOR	$B003		$07
00001111	$B002		$08
VALOR	$B001		$09
10000000	$B000		$0A
...			
$00	$0900		
VALOR	$08FF		
...			
$03	$0000		

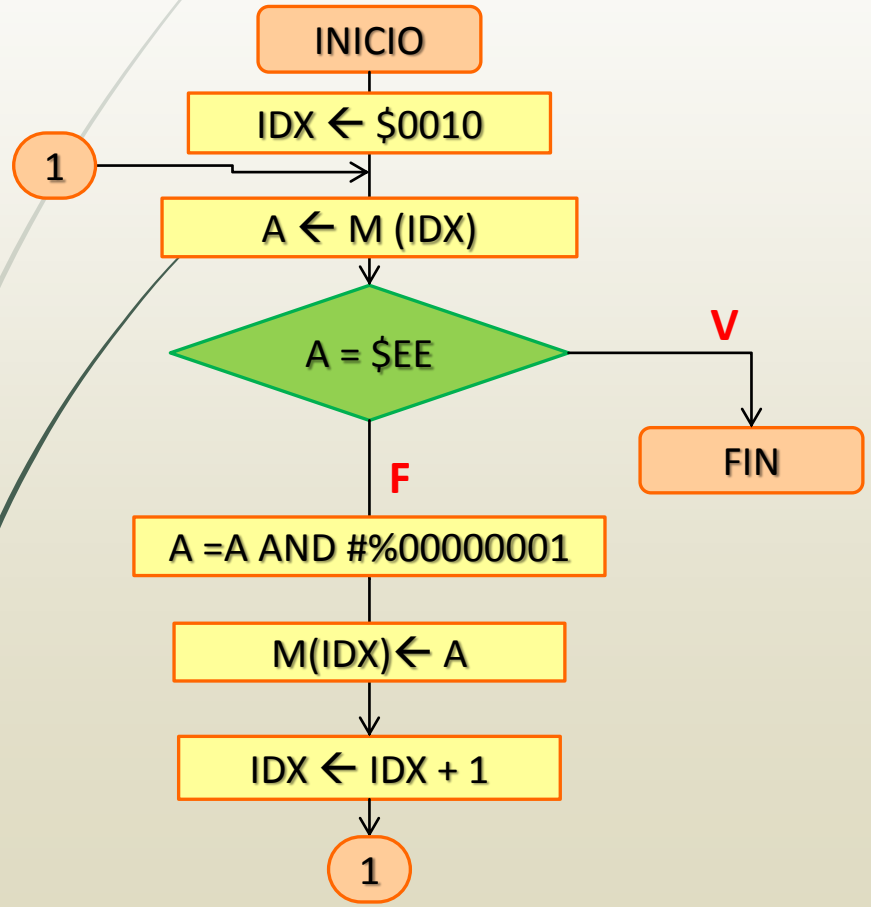
Ejercicio (6)




```
        CLR $00
        LDX #$B000
vuelve  LDAA $00,X
        ANDA #%10000001
        CMPA #$81
        BNE salto
        INC $0000
        LDAA $08FF
        STAA $00,X
salto   INX
        DEC $0900
        BNE vuelve
        SWI
```

Ejercicio (7)

Plantee y codifique un programa que blanquee la mitad más significativa de las palabras de memoria ubicadas desde la dirección $0010 hasta que se encuentre el valor $EE.
Pruebe en el SDK6800




```
        LDX #$0010
vuelve  LDAA $00,X
        CMPA #$EE
        BEQ fin
        ANDA #%00001111
        STAA $00,X
        INX
        BRA vuelve
fin     SWI
```

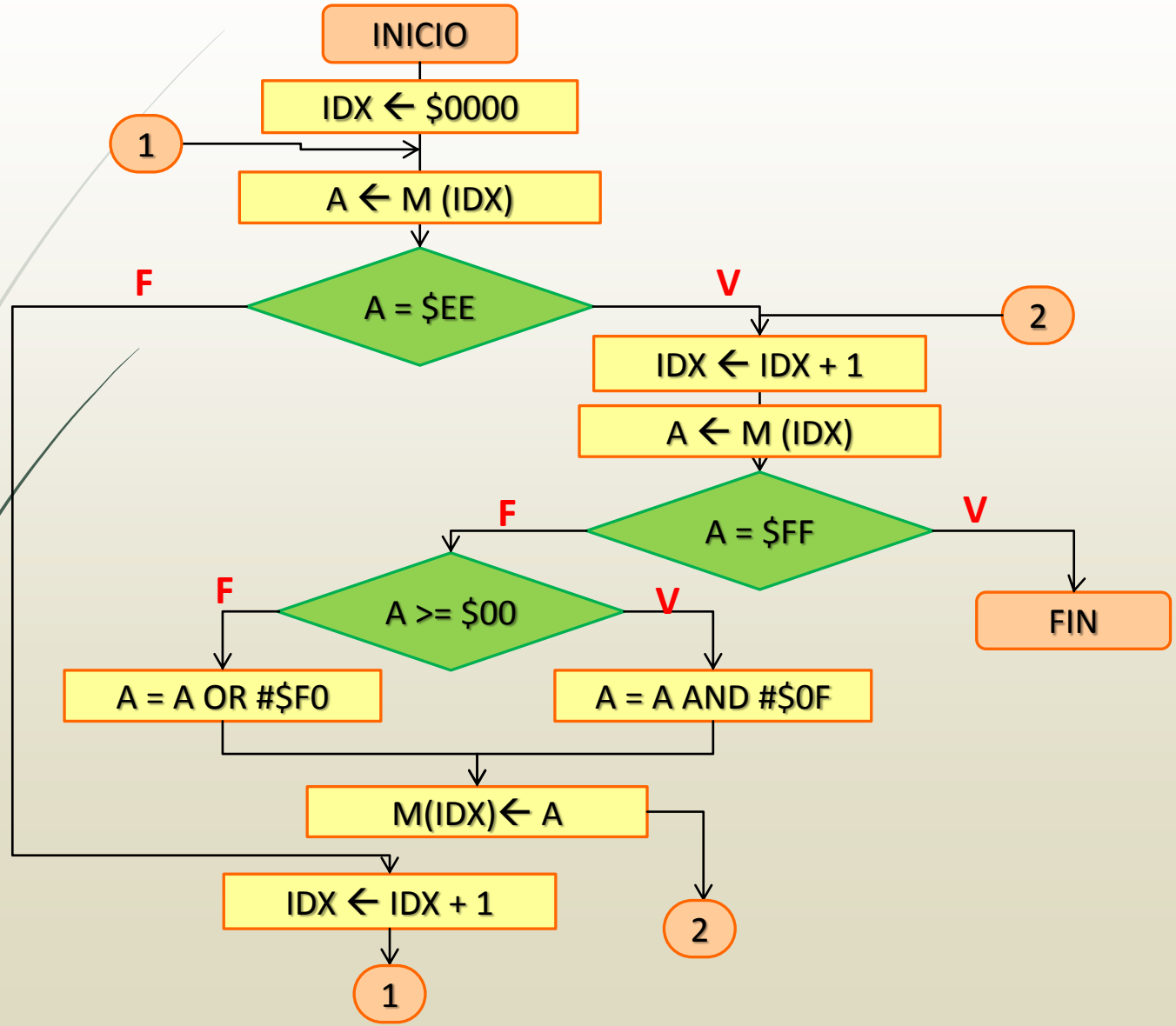
… en el emulador SDK6800

```
;programa ejercicio7
inicio .org $0000
 ldx #datos
vuelve ldaa $00,x
 cmpa #$EE
 beq fin
 anda #$0F
 staa $00,x
 inx
 bra vuelve
fin nop
datos .byte
$04,$E5,$AA,$10,$05
,$02,$EE
 .end
```

Ejercicio 8

- Plantee, programe y pruebe un programa que recorra una porción de memoria delimitada por el valor $EE (marca de inicio) y $FF (marca de fin) y en su recorrido, de acuerdo al número hallado, realice lo siguiente:
  - Si el valor es positivo o 0, debe reemplazar la mitad más significativa del byte por $0 y conservar la mitad menos significativa.
  - Si el valor es negativo, debe reemplazar la mitad más significativa del byte por $F y conservar la mitad menos significativa.

Ejercicio 8




```
;Reemplaza valores
 .org $0000
;carga valores para prueba
 jsr $007f

        ldx #$0080
repite  ldaa $00,x
        cmpa #$ee
        beq INICIO
        inx
        bra repite

INICIO  inx
        ldaa $00,x
        cmpa #$ff
        beq FIN
        cmpa #$00
        bge mayor
        oraa #$f0
        bra reemplazo
mayor   anda #$0f
reemplazo staa $00,x
        bra INICIO
        jmp FIN
 .org $007f
 rts
 .byte $01,$ee,$04,$a6,$78,$c4,$ff
FIN .end
```