

# Set de instrucciones del **MC68HC11**

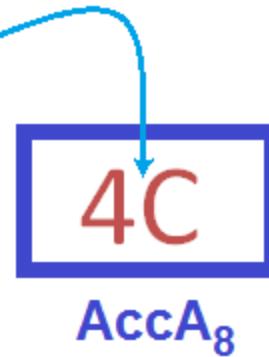
Instrucciones de transferencia

# LDAA #\$4C

M → A

- Modo de direccionamiento **inmediato (#)**
- **Opcode:** \$86      **Ciclo:**2      **Byte:**2
- Inicializa el contenido del acumulador “A” con el operando de 8 bits de forma inmediata.
- Actualiza banderas N, Z y se fuerza V←0

LDAA #\$**4C**

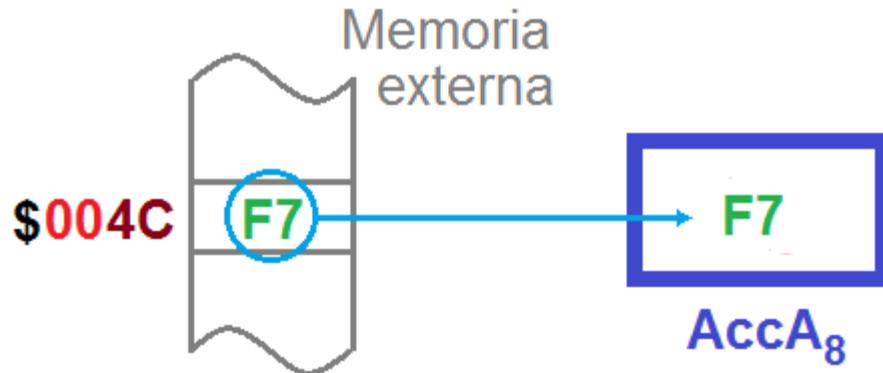


4CH = 01001100b  
↑  
MSB ∴ N = 0  
Z = 0    V = 0

# LDAA \$4C

M → A

- Modo de direccionamiento **directo**
- **Opcode:** \$96    **Ciclo:**3    **Byte:**2
- Inicializa el contenido del acumulador “A” con el dato de memoria de 8 bits de forma directa. El operando es la dirección de 8 bits, donde se encuentra el dato efectivo.
- Actualiza banderas N, Z y se fuerza V←0

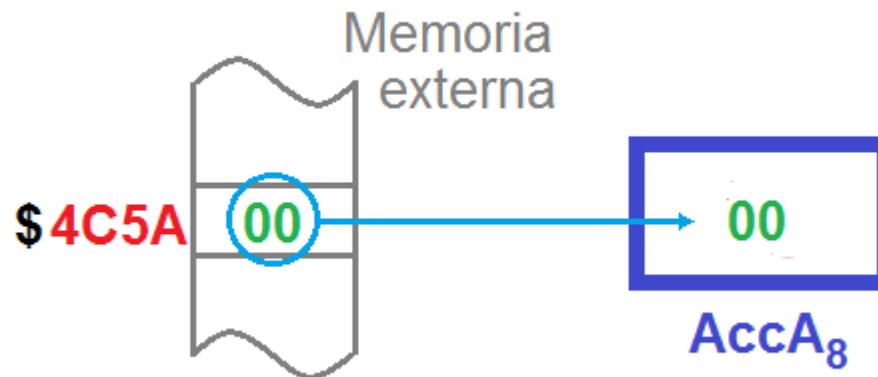


F7H = 11110111 b  
↑  
MSB ∴ N = 1  
Z = 0    V = 0

# LDAA \$4C5A

M → A

- Modo de direccionamiento **extendido**
- **Opcode:** \$B6      **Ciclo:**4      **Byte:**3
- Inicializa el contenido del acumulador “A” con el dato de memoria de 8 bits de forma directa. El operando es la dirección de 16 bits, donde se encuentra el dato efectivo.
- Actualiza banderas N, Z y se fuerza V←0

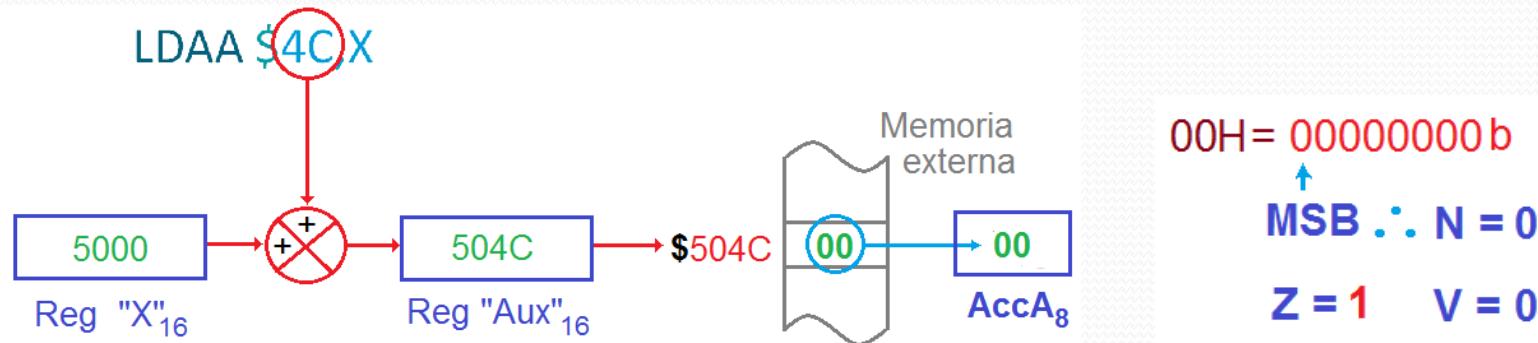


00H = 00000000 b  
↑  
MSB ∴ N = 0  
  
Z = 1    V = 0

# LDAA \$4C,X

M → A

- Modo de direccionamiento **indexado respecto de “X”**
- **Opcode: \$A6 Ciclo:4 Byte:2**
- Inicializa el contenido del acumulador “A” con el dato de memoria de 8 bits de forma indexada. Se suma el operando con el contenido del registro “X” para determinar la dirección del dato efectivo.
- Actualiza banderas N, Z y se fuerza V←0



# LDAB #\$A1

M → B

- Modo de direccionamiento **inmediato (#)**
- **Opcode: \$C6      Ciclo:2      Byte:2**
- Inicializa el contenido del acumulador “B” con el operando de 8 bits de forma inmediata.
- Actualiza banderas N, Z y se fuerza V←0

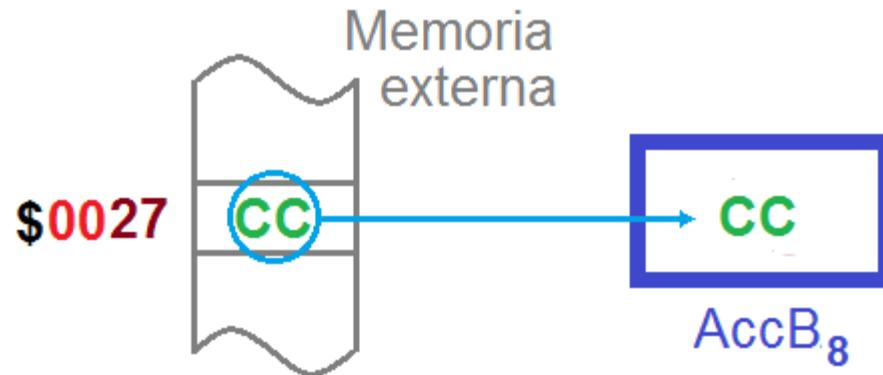


A1H = 10100001b  
↑  
MSB ∴ N = 1  
Z = 0    V = 0

# LDAB \$27

M → B

- Modo de direccionamiento **directo**
- **Opcode: \$D6 Ciclo:3 Byte:2**
- Inicializa el contenido del acumulador “B” con el dato de memoria de 8 bits de forma directa. El operando es la dirección de 8 bits, donde se encuentra el dato efectivo.
- Actualiza banderas N, Z y se fuerza V←0

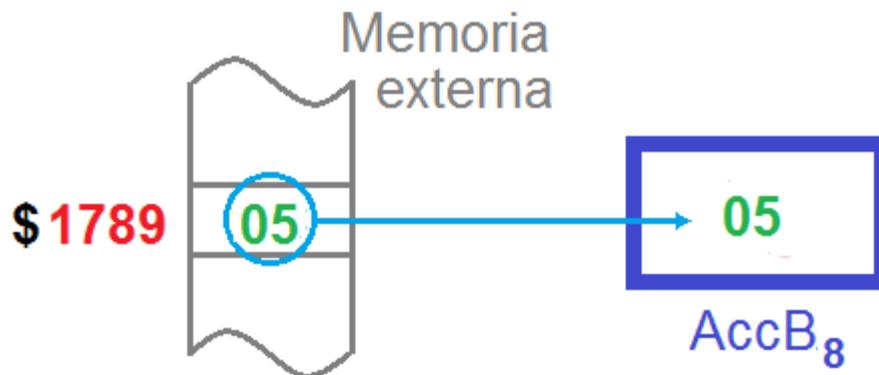


CCH = 11001100b  
↑  
MSB ∴ N = 1  
Z = 0 V = 0

# LDAB \$1789

M → B

- Modo de direccionamiento **extendido**
- **Opcode:** \$F6      **Ciclo:**4      **Byte:**3
- Inicializa el contenido del acumulador “B” con el dato de memoria de 8 bits de forma directa. El operando es la dirección de 16 bits, donde se encuentra el dato efectivo.
- Actualiza banderas N, Z y se fuerza V←0

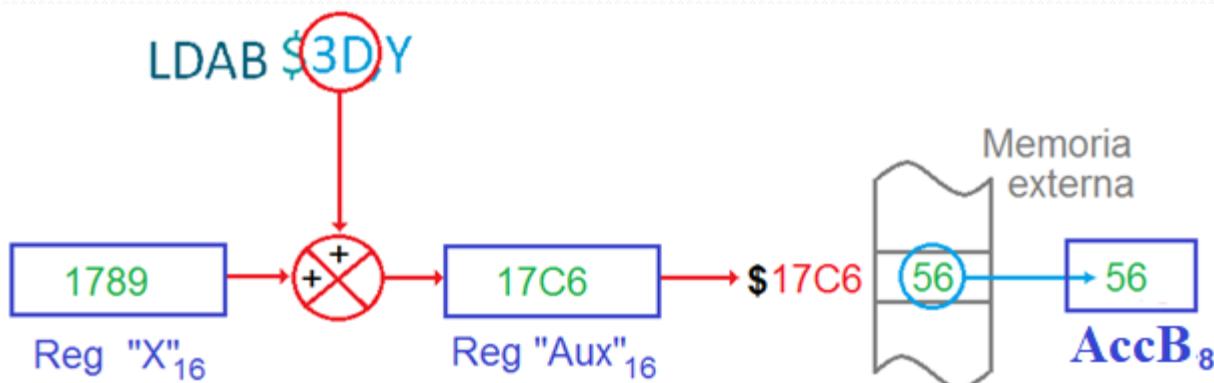


05H = 00000101b  
↑  
MSB ∴ N = 0  
  
Z = 0    V = 0

# LDAB \$3D,Y

M → B

- Modo de direccionamiento **indexado respecto de “Y”**
- **Opcode:** \$18E6      **Ciclo:**5      **Byte:**3
- Inicializa el contenido del acumulador “B” con el dato de memoria de 8 bits de forma indexada. Se suma el operando con el contenido del registro “Y” para determinar la dirección del dato efectivo.
- Actualiza banderas N, Z y se fuerza V←0

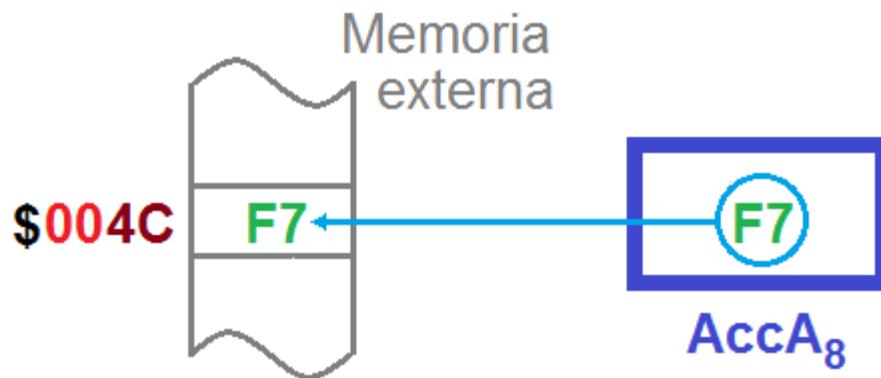


56H = 01010110b  
↑  
**MSB** ∴ N = 0  
Z = 0      V = 0

# STAA \$4C

A → M

- Modo de direccionamiento **directo**
- **Opcode: \$97      Ciclo:3      Byte:2**
- Guarda el contenido del acumulador “A” de forma directa. El operando es la dirección de 8 bits, donde se guarda el contenido del registro acumulador “A”.
- Actualiza banderas N, Z y se fuerza V←0

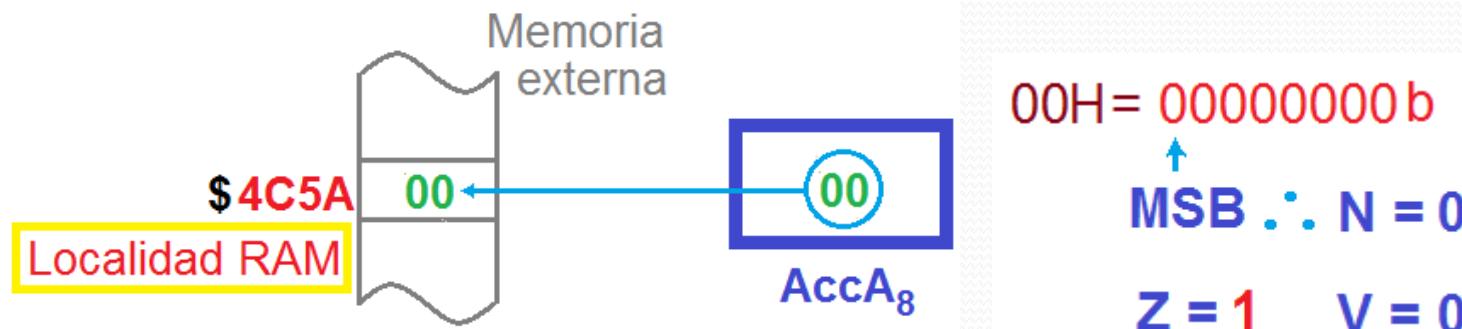


F7H = 11110111 b  
↑  
MSB ∴ N = 1  
  
Z = 0    V = 0

# STAA \$4C5A

A → M

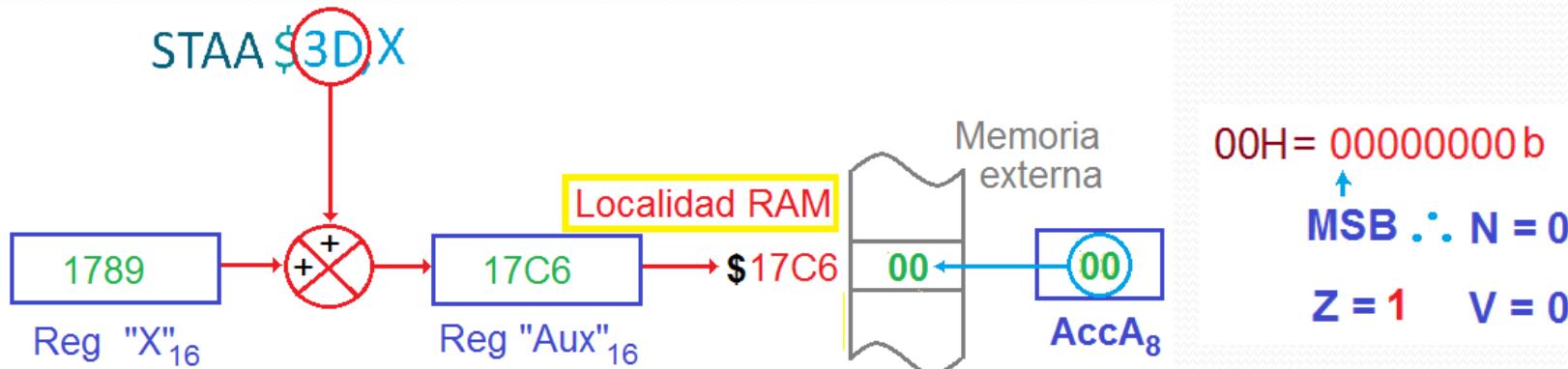
- Modo de direccionamiento **extendido**
- **Opcode:** \$B7      **Ciclo:**4      **Byte:**3
- Guarda el contenido del acumulador “A” de forma extendida. El operando es la dirección de 16 bits, donde se guarda el contenido del registro acumulador “A”.
- El programador es responsable de elegir una dirección **RAM**
- Actualiza banderas N, Z y se fuerza V←0



# STAA \$3D,X

A → M

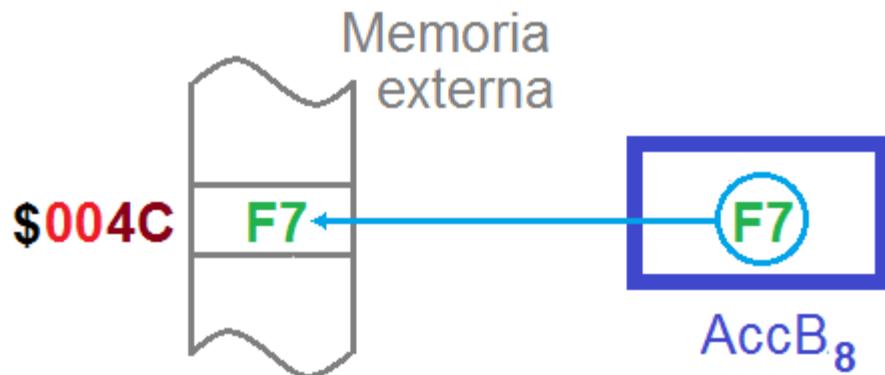
- Modo de direccionamiento **indexado respecto de “X”**
- **Opcode: \$A7 Ciclo:4 Byte:2**
- Guarda el contenido del acumulador “A” de forma indexada respecto al registro “X”. El operando se suma con el contenido del registro “X” para determinar la dirección del dato efectivo.
- El programador es responsable de elegir una dirección **RAM**
- Actualiza banderas N, Z y se fuerza V←0



# STAB \$4C

B → M

- Modo de direccionamiento **directo**
- **Opcode: \$D7 Ciclo:3 Byte:2**
- Guarda el contenido del acumulador “B” de forma directa. El operando es la dirección de 8 bits, donde se guarda el contenido del registro acumulador “B”.
- Actualiza banderas N, Z y se fuerza V←0

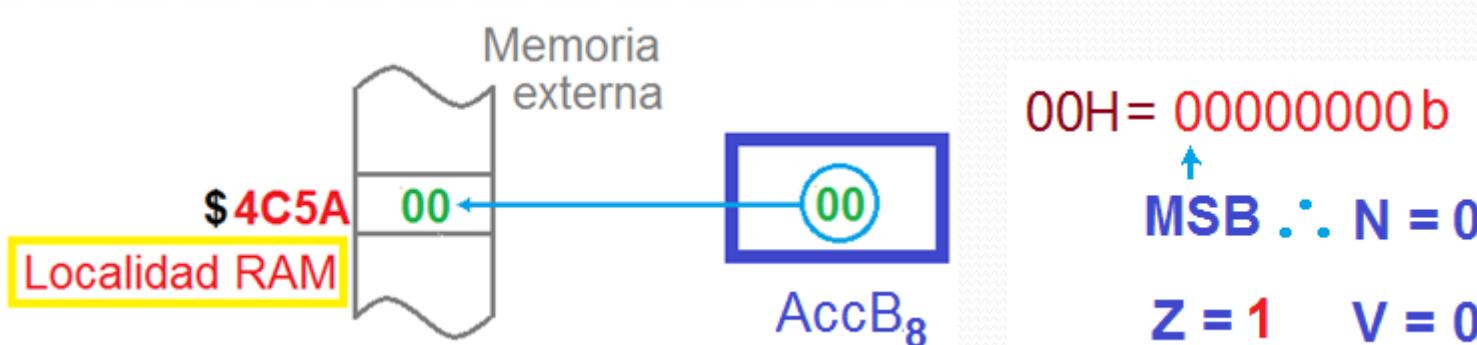


F7H = 11110111 b  
↑  
MSB ∴ N = 1  
Z = 0    V = 0

# STAB \$4C5A

B → M

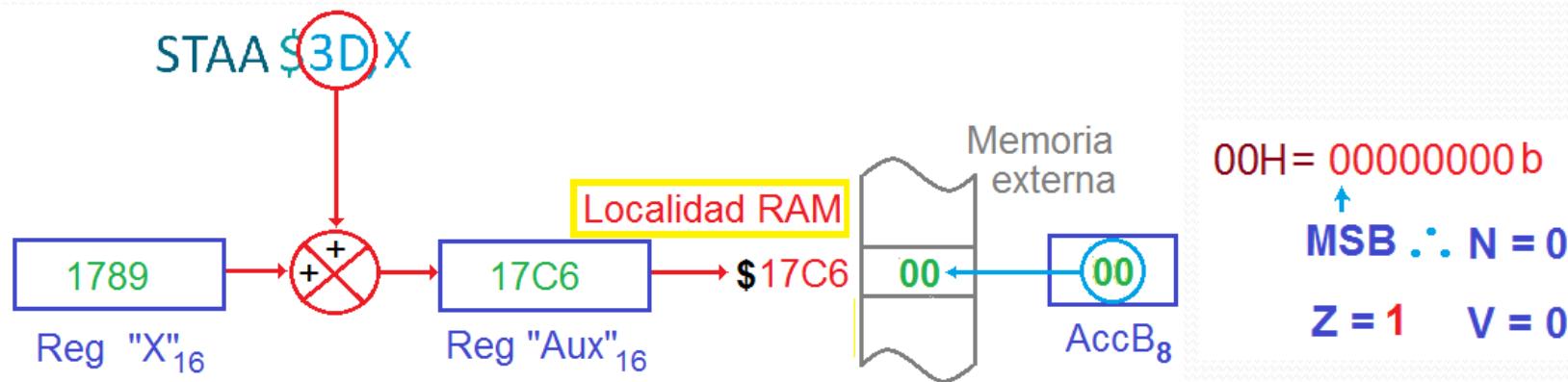
- Modo de direccionamiento **extendido**
- **Opcode:** \$F7      **Ciclo:**4      **Byte:**3
- Guarda el contenido del acumulador “B” de forma extendida. El operando es la dirección de 16 bits, donde se guarda el contenido del registro acumulador “B”.
- El programador es responsable de elegir una dirección **RAM**
- Actualiza banderas N, Z y se fuerza V←0



# STAB \$3D,X

B → M

- Modo de direccionamiento **indexado respecto de “X”**
- **Opcode: \$E7 Ciclo:4 Byte:2**
- Guarda el contenido del acumulador “B” de forma indexada respecto al registro “X”. El operando se suma con el contenido del registro “X” para determinar la dirección del dato efectivo.
- El programador es responsable de elegir una dirección **RAM**
- Actualiza banderas N, Z y se fuerza V←0

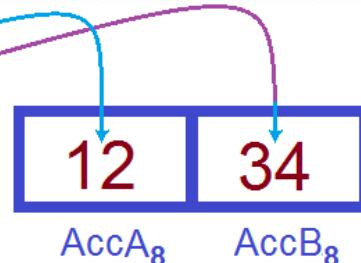


LDD #\$1234

M → A; M+1 → B

- Modo de direccionamiento **inmediato** (#)
- **Opcode:** \$CC      **Ciclo:**3      **Byte:**3
- Inicializa el contenido de los acumuladores “A” y “B” con el operando de 16 bits de forma inmediata.
- Actualiza banderas N, Z y se fuerza V←0

LDD #\$1234



1234H = 0001001000110100b

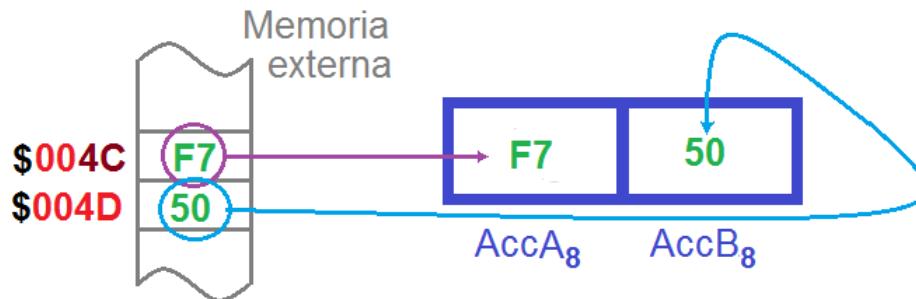
MSB ∴ N = 0

Z = 0    V = 0

# LDD \$4C

# M $\rightarrow$ A; M+1 $\rightarrow$ B

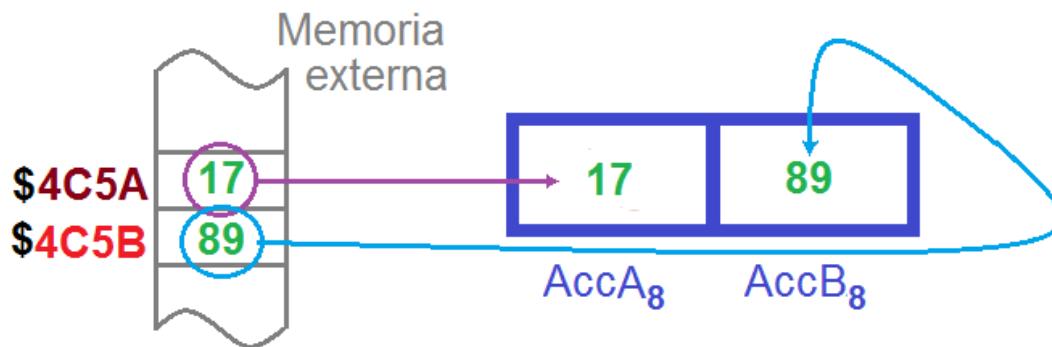
- Modo de direccionamiento **directo**
- **Opcode:** \$DC      **Ciclo:**4      **Byte:**2
- Inicializa el contenido de los acumuladores “A” y “B” con el dato de memoria de 16 bits de forma directa. El operando es la dirección de 8 bits, donde se encuentra la parte alta del dato efectivo (A) y en la localidad subsecuente, la parte baja (B).
- Actualiza banderas N, Z y se fuerza V $\leftarrow$ 0



F750H = 1111011101010000b  
↑  
MSB ∴ N = 1  
Z = 0    V = 0

# LDD \$4C5A M→A; M+1→ B

- Modo de direccionamiento **extendido**
- **Opcode:** \$FC      **Ciclo:**5      **Byte:**3
- Inicializa el contenido de los acumuladores “A” y “B” con el dato de memoria de 16 bits de forma extendida. El operando es la dirección de 16 bits, donde se encuentra el dato efectivo alto (A) y en la localidad subsecuente, el dato efectivo bajo (B).
- Actualiza banderas N, Z y se fuerza V←0

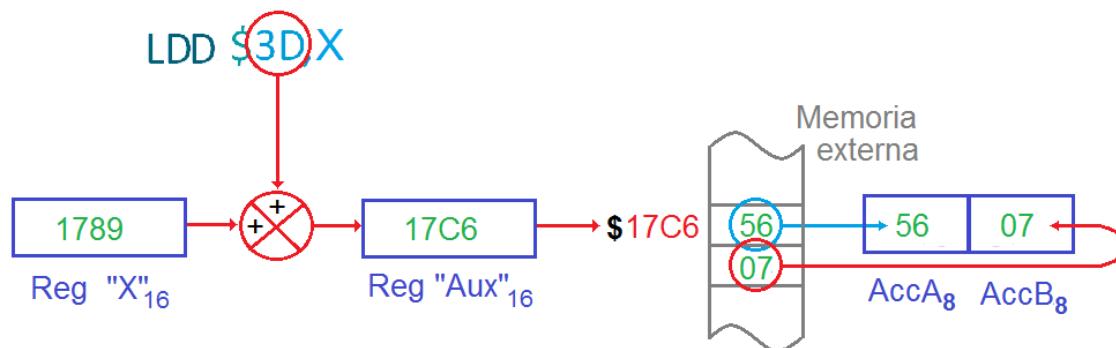


$1789H = 0001011110001001b$   
↑  
**MSB** ∴ **N = 0**  
**Z = 0**    **V = 0**

# LDD \$3D,X

# M → A; M+1 → B

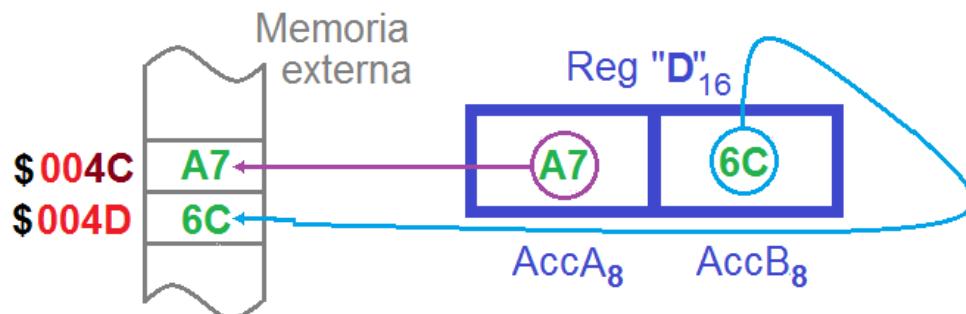
- Modo de direccionamiento **indexado respecto de “X”**
- **Opcode: \$EC Ciclo:5 Byte:2**
- Inicializa el contenido de los acumuladores “A” y “B” con el dato de memoria de 16 bits de forma indexada. Se suma el operando con el contenido del registro “X” para determinar la dirección del dato efectivo alto (A). En la localidad subsecuente se encuentra el dato efectivo bajo (B).
- Actualiza banderas N, Z y se fuerza V←0



# STD \$4C

A → M; B → M+1

- Modo de direccionamiento **directo**
- **Opcode:** \$DD      **Ciclo:**4      **Byte:**2
- Guarda el contenido de los acumuladores “A” y “B” de forma directa. El operando es la dirección de 8 bits, donde se encuentra la guarda la parte alta del dato efectivo (A) y en la localidad subsecuente, se guarda la parte baja (B).
- Actualiza banderas N, Z y se fuerza V←0

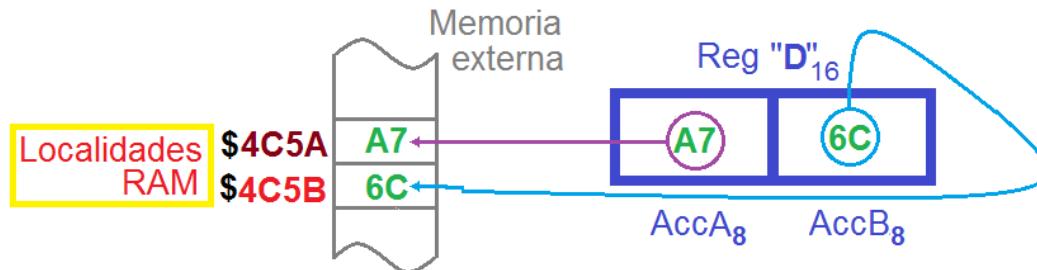


A76CH = 1010011101101100b  
↑  
MSB .∴ N = 1  
  
Z = 0    V = 0

# STD \$4C5A

# A → M; B → M+1

- Modo de direccionamiento **extendido**
- **Opcode:** \$FD    **Ciclo:**5    **Byte:**3
- Guarda el contenido de los acumuladores “A” y “B” de forma extendida. El operando es la dirección de 16 bits, donde se encuentra la guarda la parte alta del dato efectivo (A) y en la localidad subsecuente, se guarda la parte baja (B).
- El programador es responsable de elegir una dirección **RAM**
- Actualiza banderas N, Z y se fuerza V←0



$$A76CH = 1010011101101100b$$

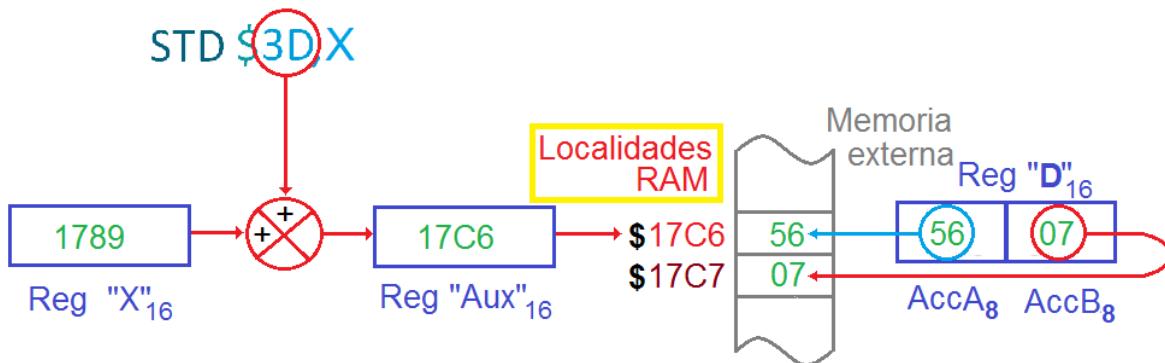
↑  
**MSB** ∴ **N = 1**

**Z = 0**    **V = 0**

# STD \$3D,X

# A → M; B → M+1

- Modo de direccionamiento **indexado respecto de “X”**
- **Opcode: \$ED Ciclo:5 Byte:2**
- Guarda el contenido de los acumuladores “A” y “B” de forma indexada respecto a “X”. El operando se suma con el contenido del registro “X” para determinar la dirección donde se guarda el contenido del acumulador “A”. En la localidad subsecuente se guarda el contenido del acumulador “B”.
- El programador es responsable de elegir una dirección **RAM**
- Actualiza banderas N, Z y se fuerza V←0



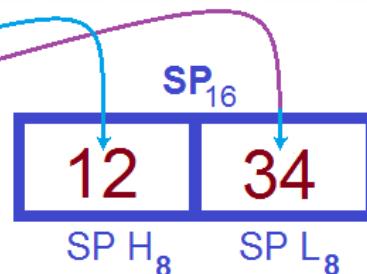
$5607H = 0101011000000111b$   
↑  
**MSB** ∴ **N = 0**  
**Z = 0**    **V = 0**

# LDS #\$1234

# M:M+1 → SP

- Modo de direccionamiento **inmediato** (#)
- **Opcode:** \$8E      **Ciclo:**3      **Byte:**3
- Inicializa el contenido del registro “Apuntador de Pila” o “Stack Pointer” con el operando de 16 bits de forma inmediata.
- Actualiza banderas N, Z y se fuerza V←0

LDS #\$1234



$$1234H = 0001001000110100b$$

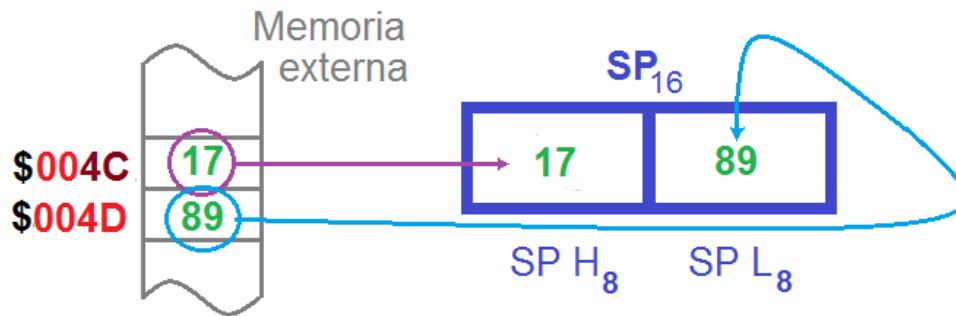
MSB ∴ N = 0

Z = 0    V = 0

# LDS \$4C

M:M+1 → SP

- Modo de direccionamiento **directo**
- **Opcode:** \$9E    **Ciclo:**4    **Byte:**2
- Inicializa el contenido del registro “Apuntador de Pila” o “Stack Pointer” con el operando de 16 bits de forma directa. El operando es la dirección de 8 bits, donde se encuentra la parte alta del dato efectivo (SP H) y en la localidad subsecuente, la parte baja (SP L).
- Actualiza banderas N, Z y se fuerza V←0

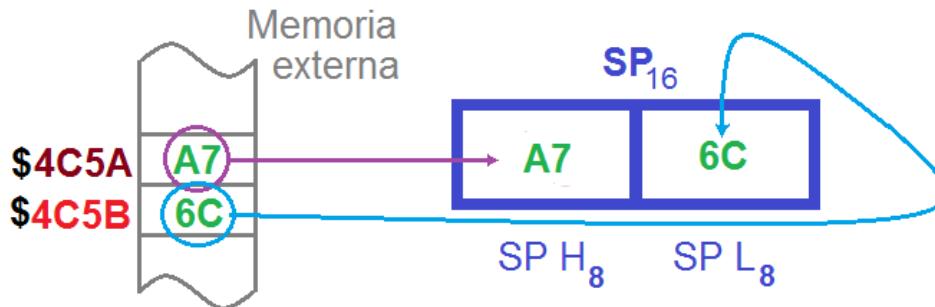


1789H = 0001011110001001b  
↑  
**MSB** ∴ **N = 0**  
**Z = 0**    **V = 0**

# LDS \$4C5A

M:M+1 → SP

- Modo de direccionamiento **extendido**
- **Opcode:** \$BE    **Ciclo:**5    **Byte:**3
- Inicializa el contenido del registro “Apuntador de Pila” o “Stack Pointer” con el operando de 16 bits de forma extendida.  
El operando es la dirección de 8 bits, donde se encuentra la parte alta del dato efectivo (SP H) y en la localidad subsecuente, la parte baja (SP L).
- Actualiza banderas N, Z y se fuerza V←0

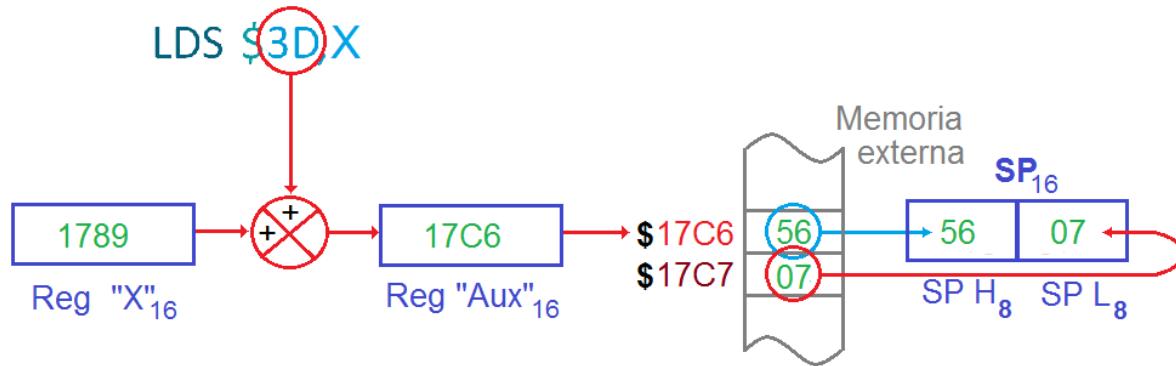


$$\begin{aligned} A76CH &= 1010011101101100b \\ \text{MSB} \therefore N &= 1 \\ Z = 0 & \quad V = 0 \end{aligned}$$

# LDS \$3D,X

M:M+1 → SP

- Modo de direccionamiento **indexado respecto de “X”**
- **Opcode: \$AE    Ciclo:5    Byte:2**
- Inicializa el contenido del registro “Apuntador de Pila” o “Stack Pointer” con el operando de 16 bits de forma indexada respecto a “X”.  
Se suma el operando con el contenido del registro “X” para determinar la dirección del dato efectivo alto (SP H). En la localidad subsecuente se encuentra el dato efectivo bajo (SP L).
- Actualiza banderas N, Z y se fuerza V←0



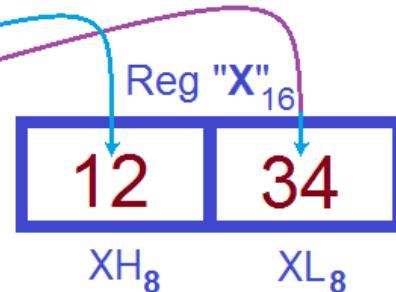
5607H = 0101011000000111b  
↑  
**MSB ∴ N = 0**  
**Z = 0    V = 0**

# LDX #\$1234

# M:M+1 → X

- Modo de direccionamiento **inmediato (#)**
- **Opcode:** \$CE      **Ciclo:**3      **Byte:**3
- Inicializa el contenido del registro “X” con el operando de 16 bits de forma inmediata.
- Actualiza banderas N, Z y se fuerza V←0

LDX #\$1234



1234H = 0001001000110100b

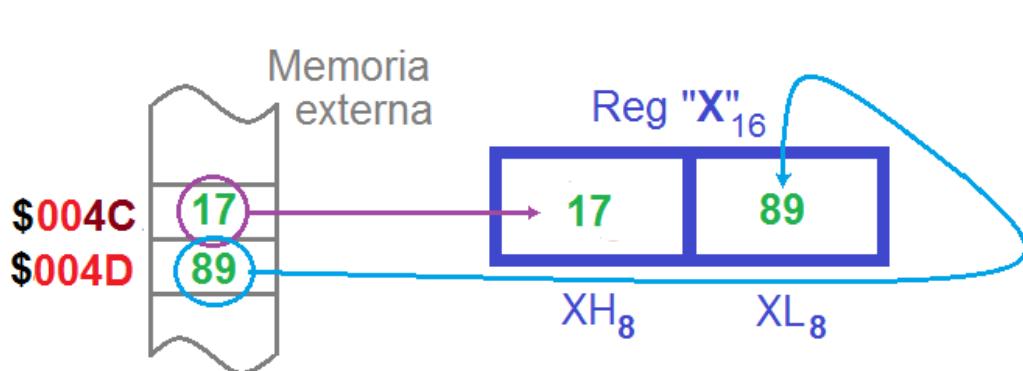
MSB ∴ N = 0

Z = 0    V = 0

# LDX \$4C

# M:M+1 → X

- Modo de direccionamiento **directo**
- **Opcode:** \$DE    **Ciclo:**4    **Byte:**2
- Inicializa el contenido del registro “X” con el operando de 16 bits de forma directa. El operando es la dirección de 8 bits, donde se encuentra la parte alta del dato efectivo (XH) y en la localidad subsecuente, la parte baja (XL).
- Actualiza banderas N, Z y se fuerza V←0



1789H = 0001011110001001b  
↑  
**MSB** ∴ N = 0  
  
**Z** = 0    **V** = 0

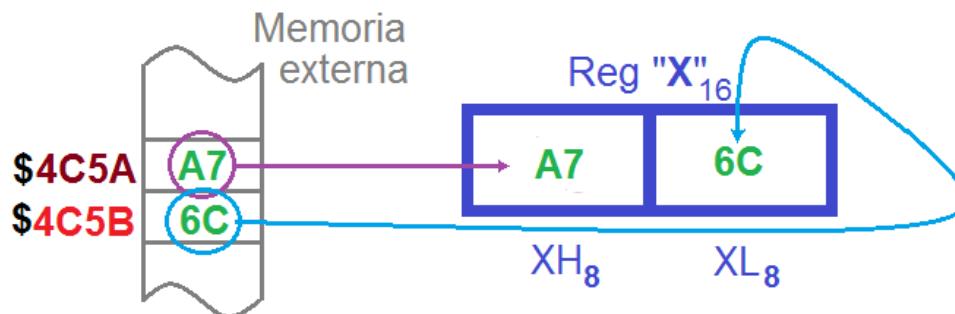
# LDX \$4C5A

M:M+1 → X

- Modo de direccionamiento **extendido**
- **Opcode:** \$FE    **Ciclo:**5    **Byte:**3
- Inicializa el contenido del registro “X” con el operando de 16 bits de forma extendida.

El operando es la dirección de 8 bits, donde se encuentra la parte alta del dato efectivo (XH) y en la localidad subsecuente, la parte baja (XL).

- Actualiza banderas N, Z y se fuerza V←0



A76CH = 1010011101101100b  
↑  
**MSB** ∴ N = 1  
  
Z = 0    V = 0

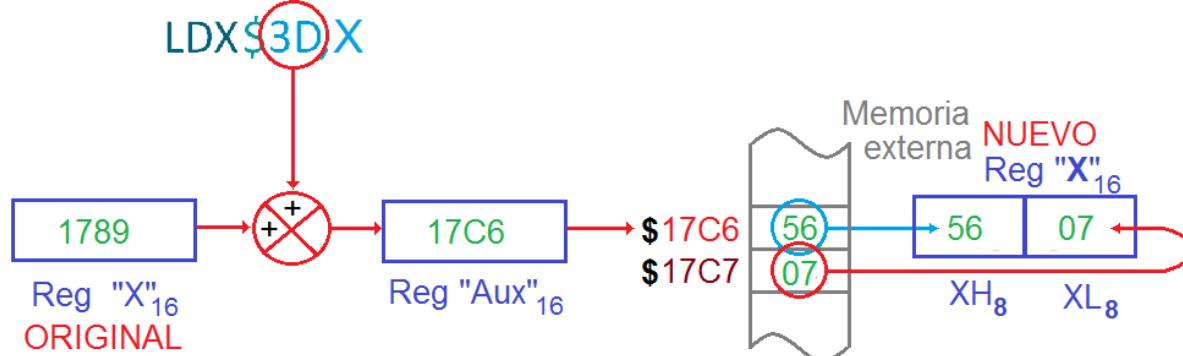
# LDX \$3D,X

M:M+1 → X

- Modo de direccionamiento **indexado respecto de “X”**
- **Opcode: \$EE Ciclo:5 Byte:2**
- Inicializa el contenido del registro “X” con el operando de 16 bits de forma indexada respecto a “X”.

Se suma el operando con el contenido del registro “X” para determinar la dirección del dato efectivo alto (XH). En la localidad subsecuente se encuentra el dato efectivo bajo (XL).

- Actualiza banderas N, Z y se fuerza V←0



$$5607H = 0101011000000111b$$

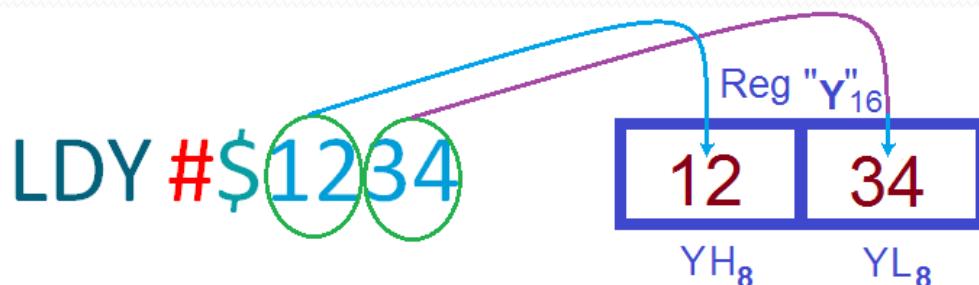
↑  
MSB ∴ N = 0

Z = 0 V = 0

# LDY #\$1234

# M:M+1 → Y

- Modo de direccionamiento **inmediato (#)**
- **Opcode:** \$18CE      **Ciclo:**4      **Byte:**4
- Inicializa el contenido del registro “Y” con el operando de 16 bits de forma inmediata.
- Actualiza banderas N, Z y se fuerza V←0

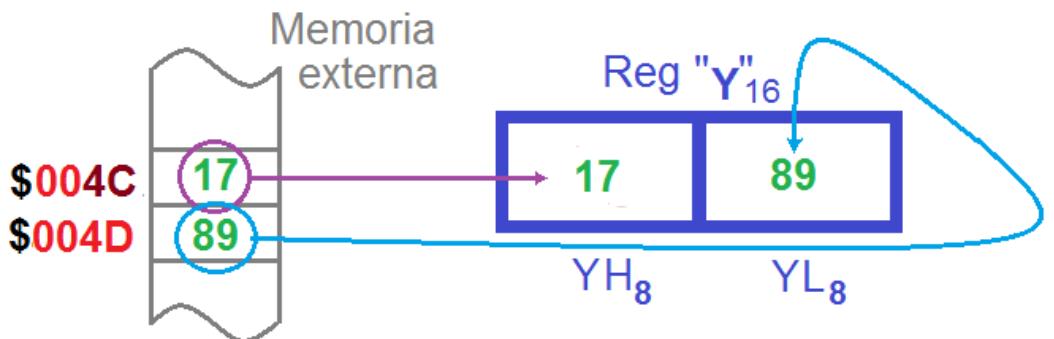


1234H = 0001001000110100b  
MSB ∴ N = 0  
Z = 0 V = 0

# LDY \$4C

# M:M+1 → Y

- Modo de direccionamiento **directo**
- **Opcode:** \$18DE    **Ciclo:4**    **Byte:2**
- Inicializa el contenido del registro “Y” con el operando de 16 bits de forma directa. El operando es la dirección de 8 bits, donde se encuentra la parte alta del dato efectivo (YH) y en la localidad subsecuente, la parte baja (YL).
- Actualiza banderas N, Z y se fuerza V←0



1789H = 0001011110001001b  
↑  
**MSB** ∴ N = 0  
  
**Z** = 0    **V** = 0

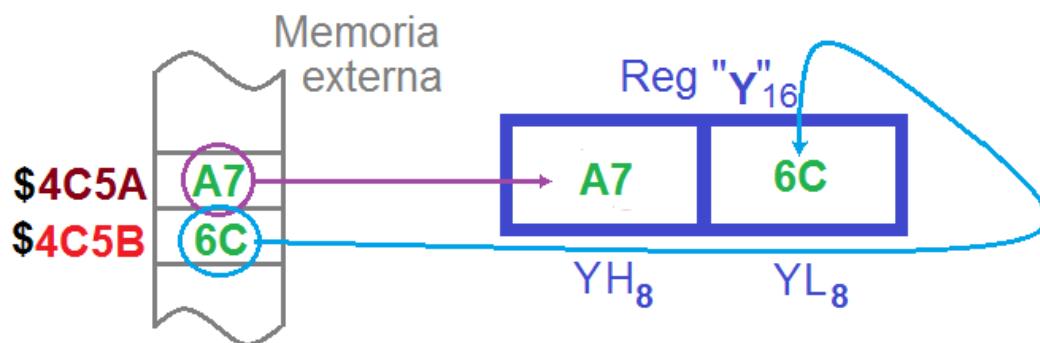
# LDY \$4C5A

M:M+1 → Y

- Modo de direccionamiento **extendido**
- **Opcode:** \$18FE    **Ciclo:**6    **Byte:**4
- Inicializa el contenido del registro “Y” con el operando de 16 bits de forma extendida.

El operando es la dirección de 8 bits, donde se encuentra la parte alta del dato efectivo (YH) y en la localidad subsecuente, la parte baja (YL).

- Actualiza banderas N, Z y se fuerza V←0



A76CH = 1010011101101100b  
↑  
**MSB** ∴ N = 1  
  
Z = 0    V = 0

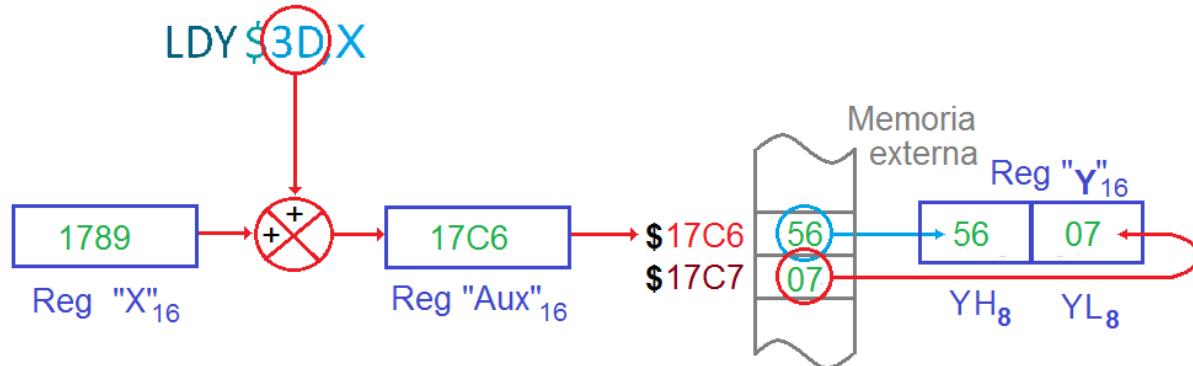
# LDY \$3D,X

M:M+1 → Y

- Modo de direccionamiento **indexado respecto de “X”**
- **Opcode:** \$1AEE    **Ciclo:5**    **Byte:2**
- Inicializa el contenido del registro “Y” con el operando de 16 bits de forma indexada respecto a “X”.

Se suma el operando con el contenido del registro “X” para determinar la dirección del dato efectivo alto (YH). En la localidad subsecuente se encuentra el dato efectivo bajo (YL).

- Actualiza banderas N, Z y se fuerza V←0



5607H = 0101011000000111b  
↑  
MSB ∴ N = 0  
Z = 0   V = 0

# TAB

A → B

- Modo de direccionamiento **inherente**
- **Opcode:** \$16    **Ciclo:**2    **Byte:**1
- Transfiere (copia) el contenido del registro acumulador “A” hacia el registro acumulador “B”.
- Actualiza banderas N, Z y se fuerza V←0



# TBA

B → A

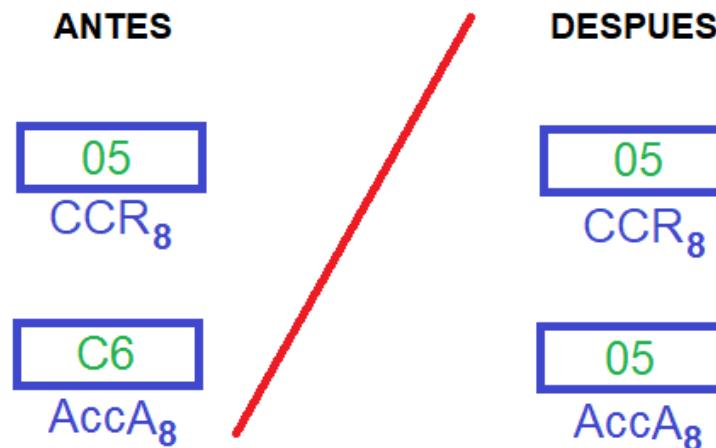
- Modo de direccionamiento **inherente**
- **Opcode:** \$17    **Ciclo:**2    **Byte:**1
- Transfiere (copia) el contenido del registro acumulador “B” hacia el registro acumulador “A”.
- Actualiza banderas N, Z y se fuerza V←0



# TPA

CCR → A

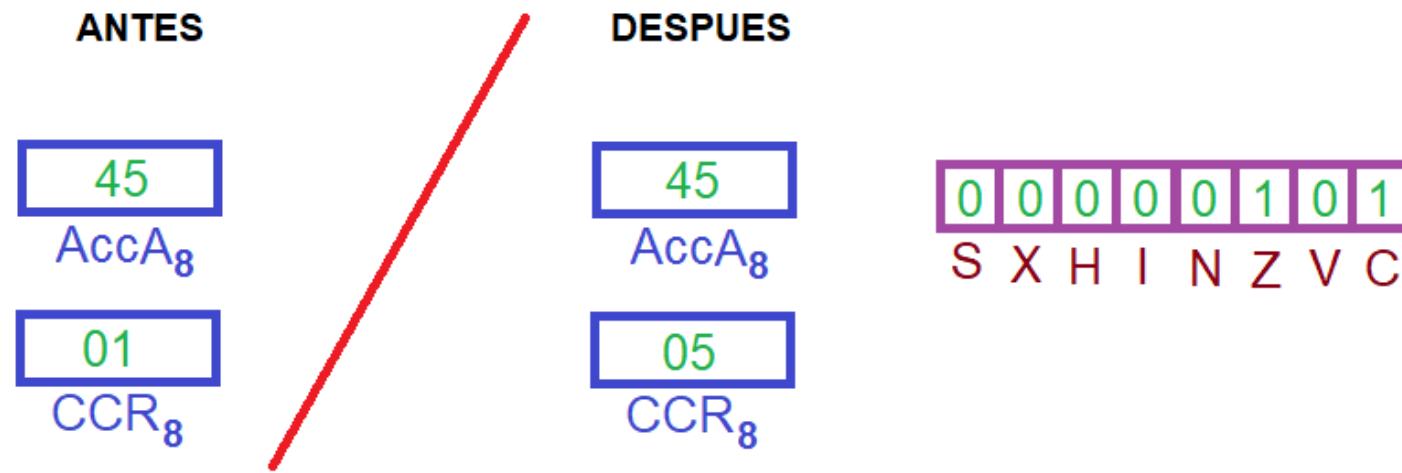
- Modo de direccionamiento **inherente**
- **Opcode:** \$06      **Ciclo:**2      **Byte:**1
- Transfiere (copia) el contenido del registro de estados “CCR” hacia el registro acumulador “A”.
- Ninguna bandera del registro de estados se actualiza



# TAP

A → CCR

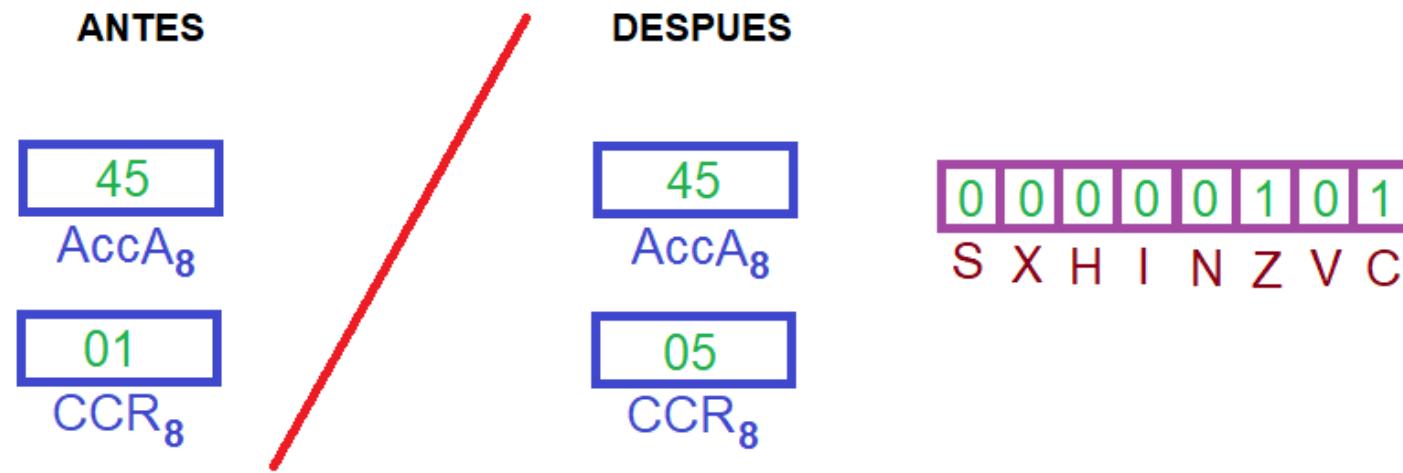
- Modo de direccionamiento **inherente**
- **Opcode:** \$07    **Ciclo:**2    **Byte:**1
- Transfiere (copia) el contenido del registro acumulador “A” hacia el registro de estados “CCR”.
- Todas las banderas del registro de estado se actualizan, pero si el bit 6 del CCR tiene un valor lógico de cero, no podrá ponerse en alto.



# TAP

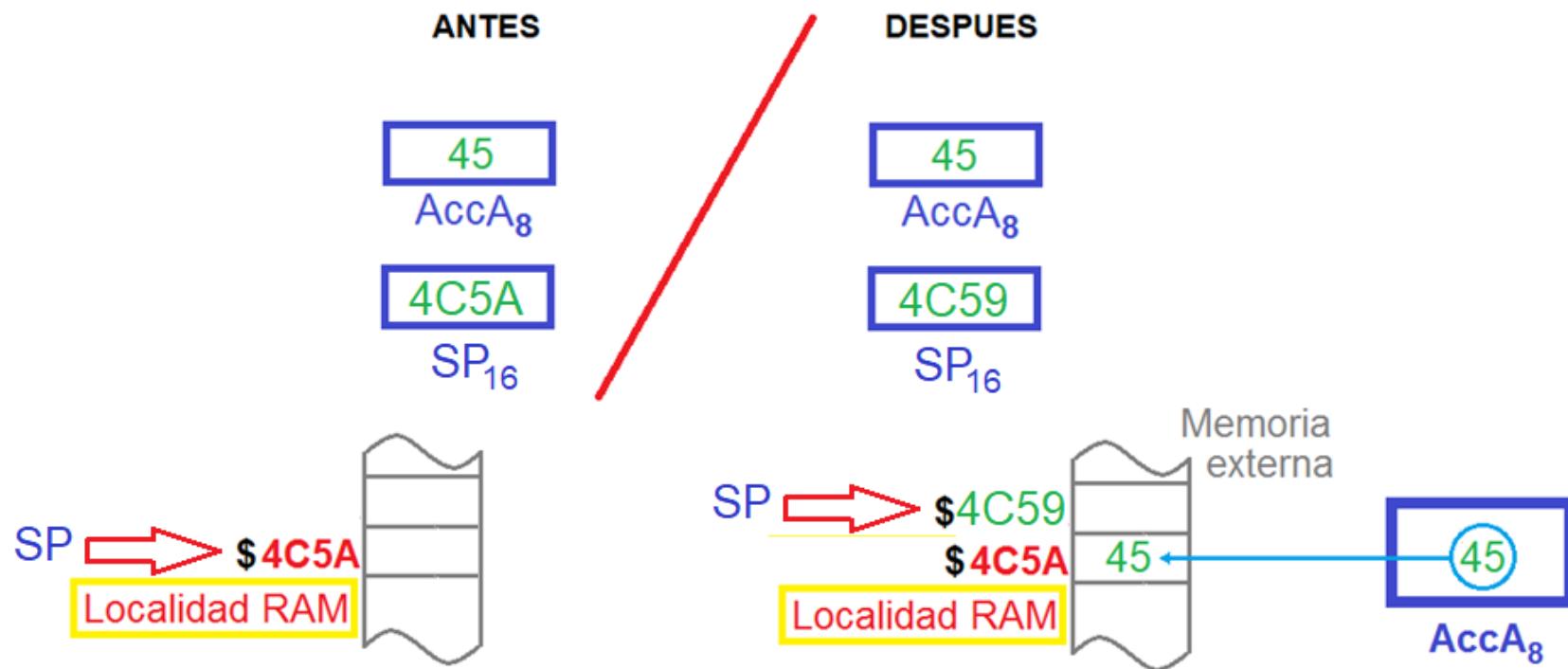
A → CCR

- Modo de direccionamiento **inherente**
- **Opcode:** \$07    **Ciclo:**2    **Byte:**1
- Transfiere (copia) el contenido del registro acumulador “A” hacia el registro de estados “CCR”.
- Todas las banderas del registro de estado se actualizan, pero si el bit 6 del CCR tiene un valor lógico de cero, no podrá ponerse en alto.



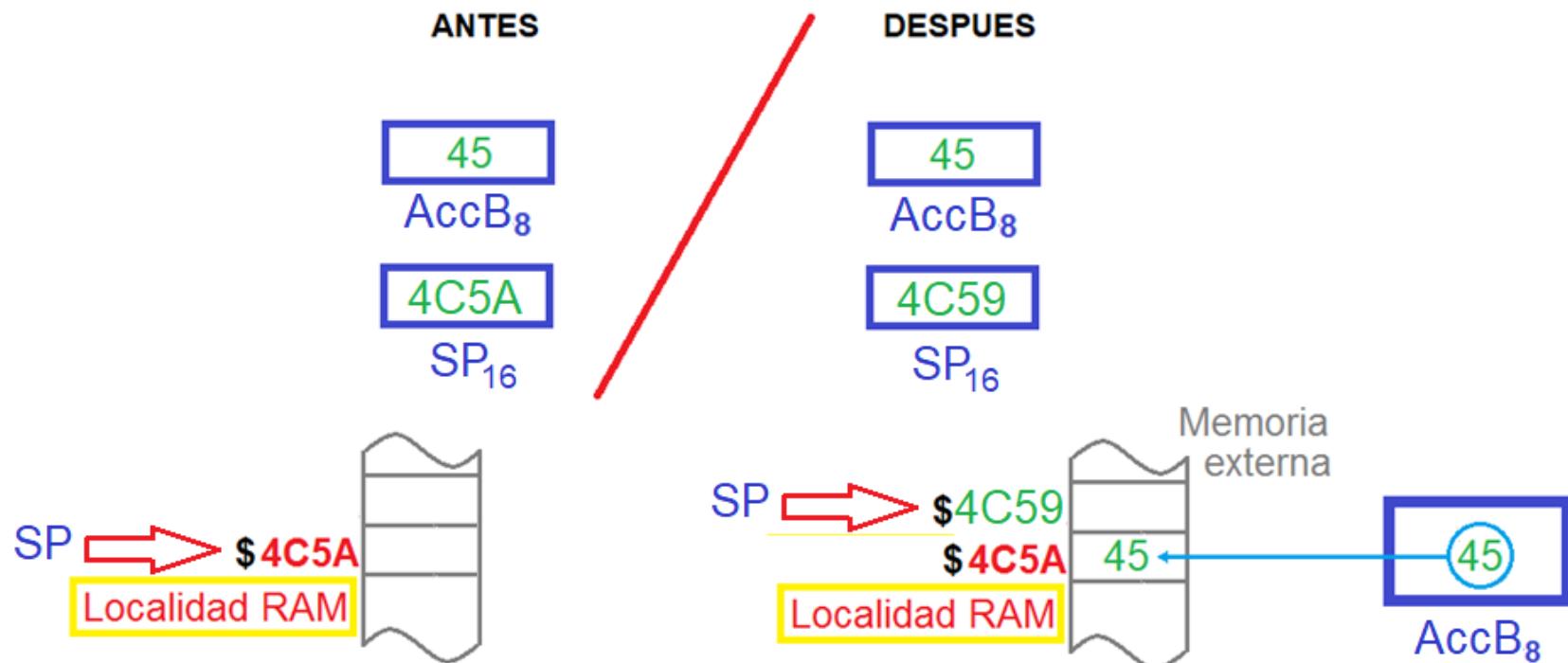
# PSHA A $\rightarrow$ (SP); (SP) -1 $\rightarrow$ SP

- Modo de direccionamiento **inherente**
- **Opcode:** \$36      **Ciclo:**3      **Byte:**1
- Transfiere (copia) el contenido del registro acumulador “A” en la dirección apuntada por el SP y después hace un decremento del SP en una unidad.
- Ninguna bandera del CCR se actualiza.



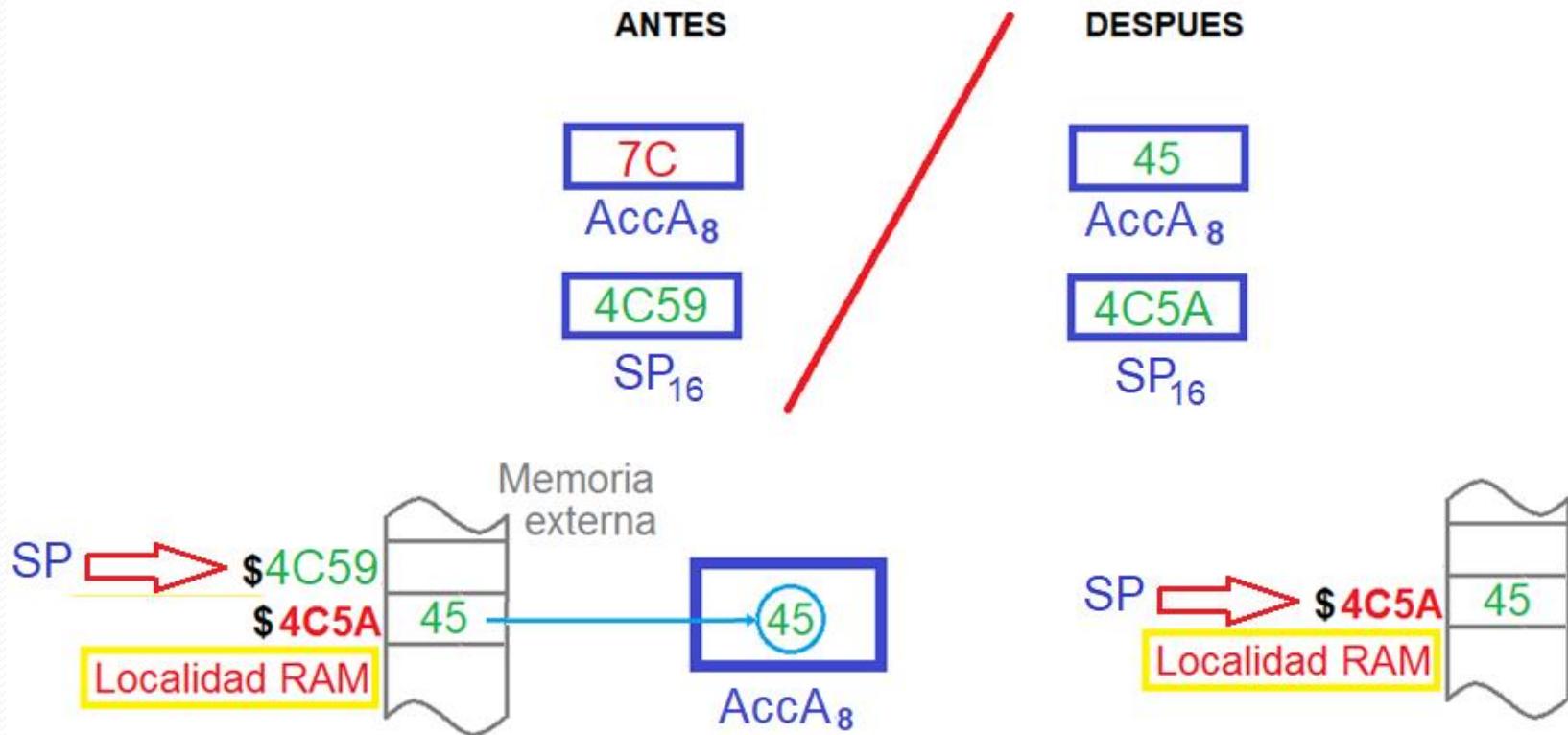
# PSHB    $B \rightarrow (SP); (SP) - 1 \rightarrow SP$

- Modo de direccionamiento **inherente**
- **Opcode:** \$37    **Ciclo:**3    **Byte:**1
- Transfiere (copia) el contenido del registro acumulador “B” en la dirección apuntada por el SP y después hace un decremento del SP en una unidad.
- Ninguna bandera del CCR se actualiza.



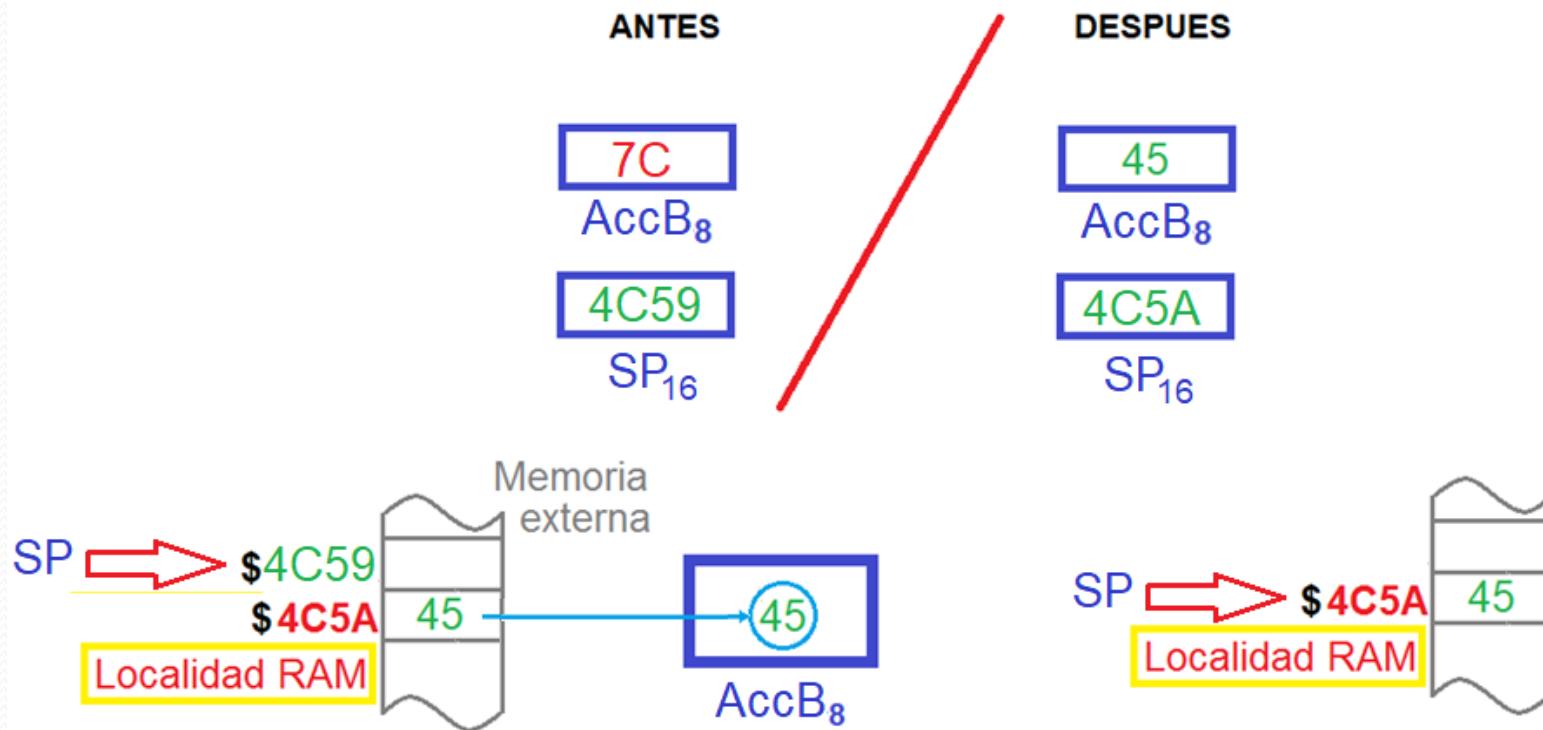
# PULA $((SP)+1) \rightarrow A; (SP) + 1 \rightarrow SP$

- Modo de direccionamiento **inherente**
- **Opcode:** \$32    **Ciclo:**4    **Byte:**1
- Transfiere (copia) el contenido de la dirección apuntada por el SP hacia el registro acumulador “A” y después hace un incremento del SP en una unidad.
- Ninguna bandera del CCR se actualiza.



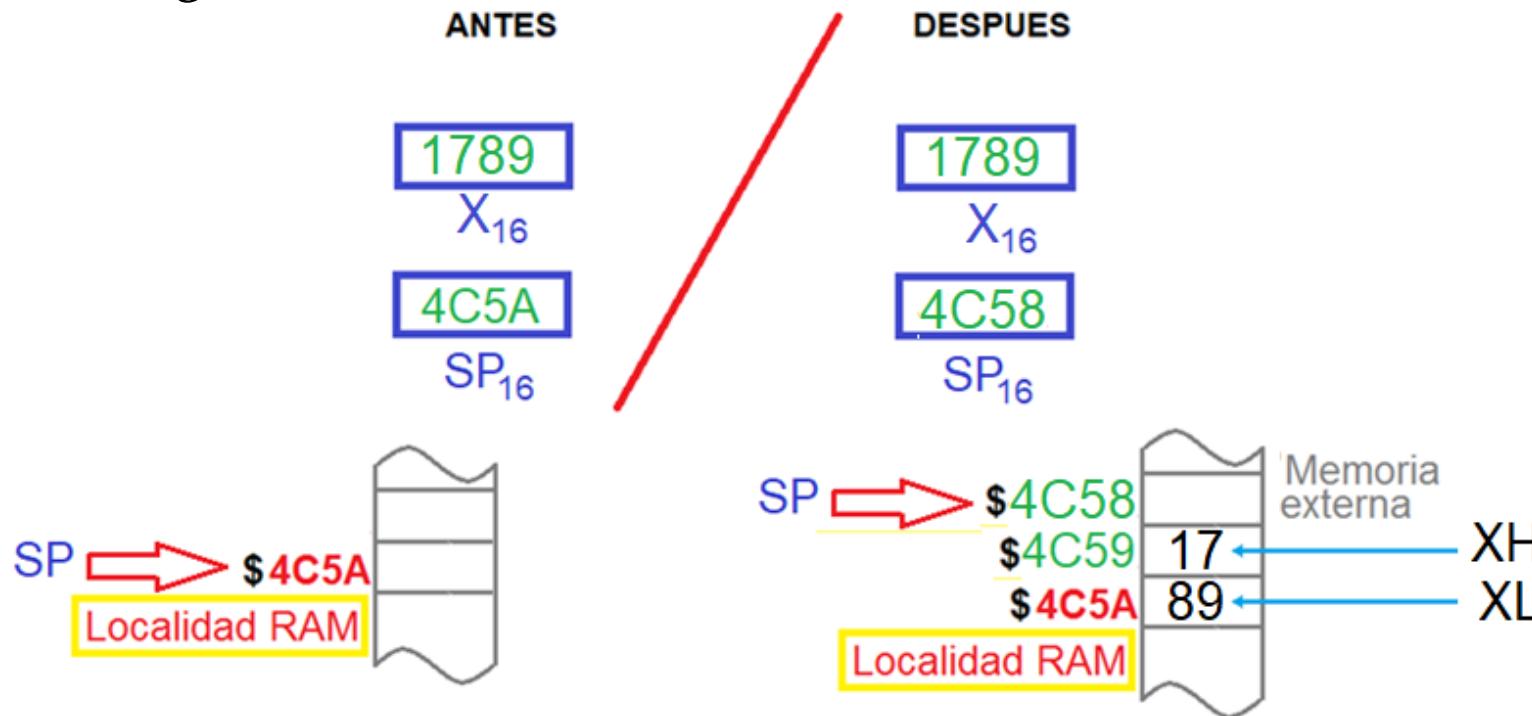
# PULB $((SP)+1) \rightarrow B; (SP) + 1 \rightarrow SP$

- Modo de direccionamiento **inherente**
- **Opcode:** \$33    **Ciclo:**4    **Byte:**1
- Transfiere (copia) el contenido de la dirección apuntada por el SP hacia el registro acumulador “A” y después hace un incremento del SP en una unidad.
- Ninguna bandera del CCR se actualiza.



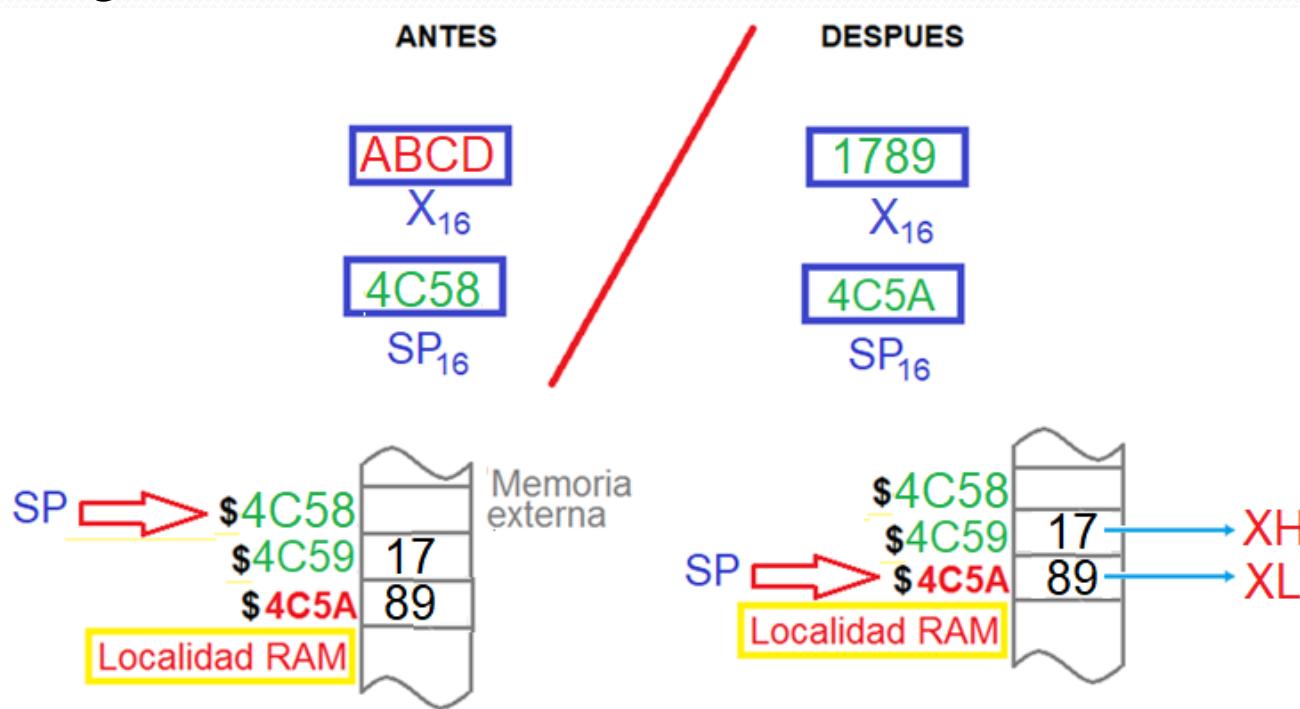
# PSHX    X→(SP); (SP) -2 → SP

- Modo de direccionamiento **inherente**
- **Opcode:** \$3C    **Ciclo:**4    **Byte:**1
- Transfiere (copia) el contenido del registro registro “X” en la dirección apuntada por el SP y después hace un decremento del SP en dos unidades.
- Ninguna bandera del CCR se actualiza.



# PULX    $(SP) \rightarrow X; (SP) + 2 \rightarrow SP$

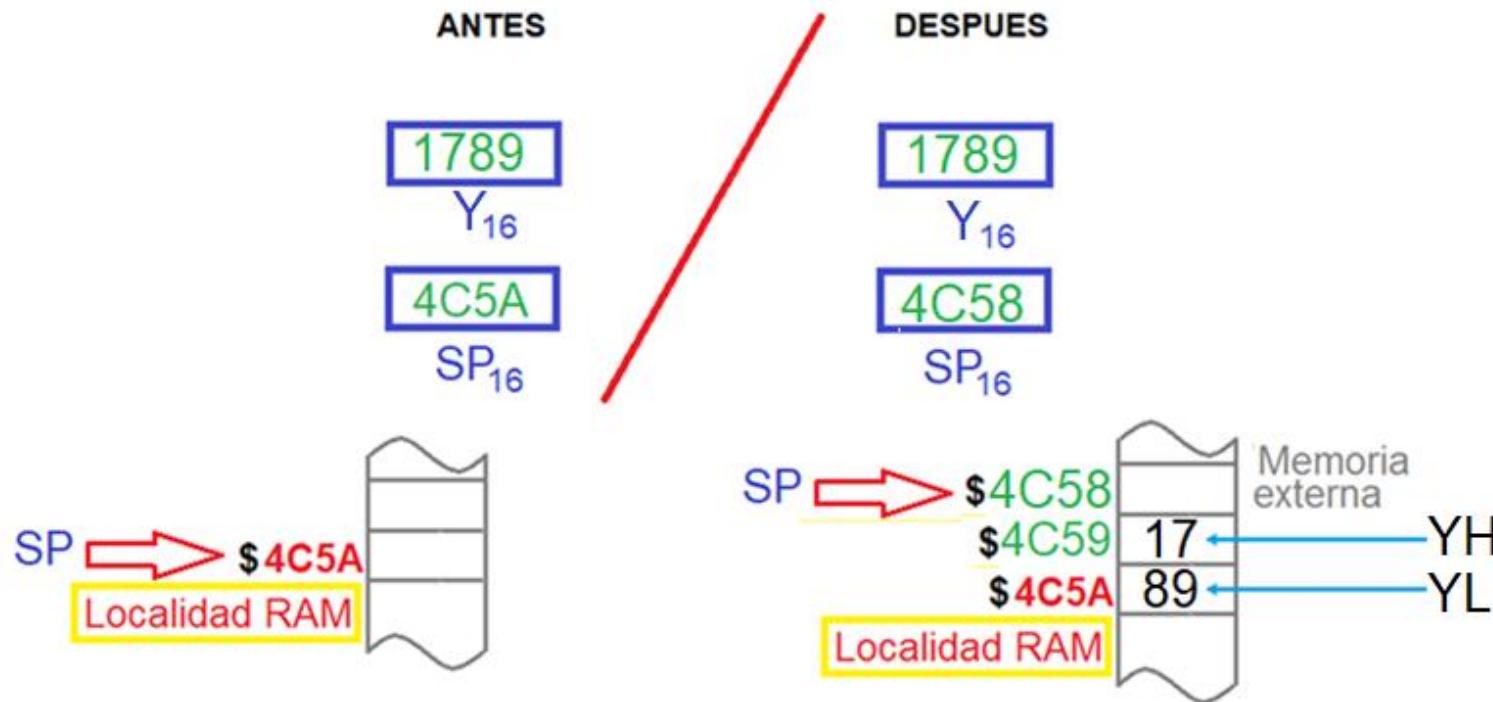
- Modo de direccionamiento **inherente**
- **Opcode:** \$38      **Ciclo:**4      **Byte:**1
- Transfiere (copia) el contenido de las dos localidades de la cúspide del SP hacia el registro “X” haciendo un incremento del SP en dos unidades.
- Ninguna bandera del CCR se actualiza.



# PSHY

# $Y \rightarrow (SP); (SP) - 2 \rightarrow SP$

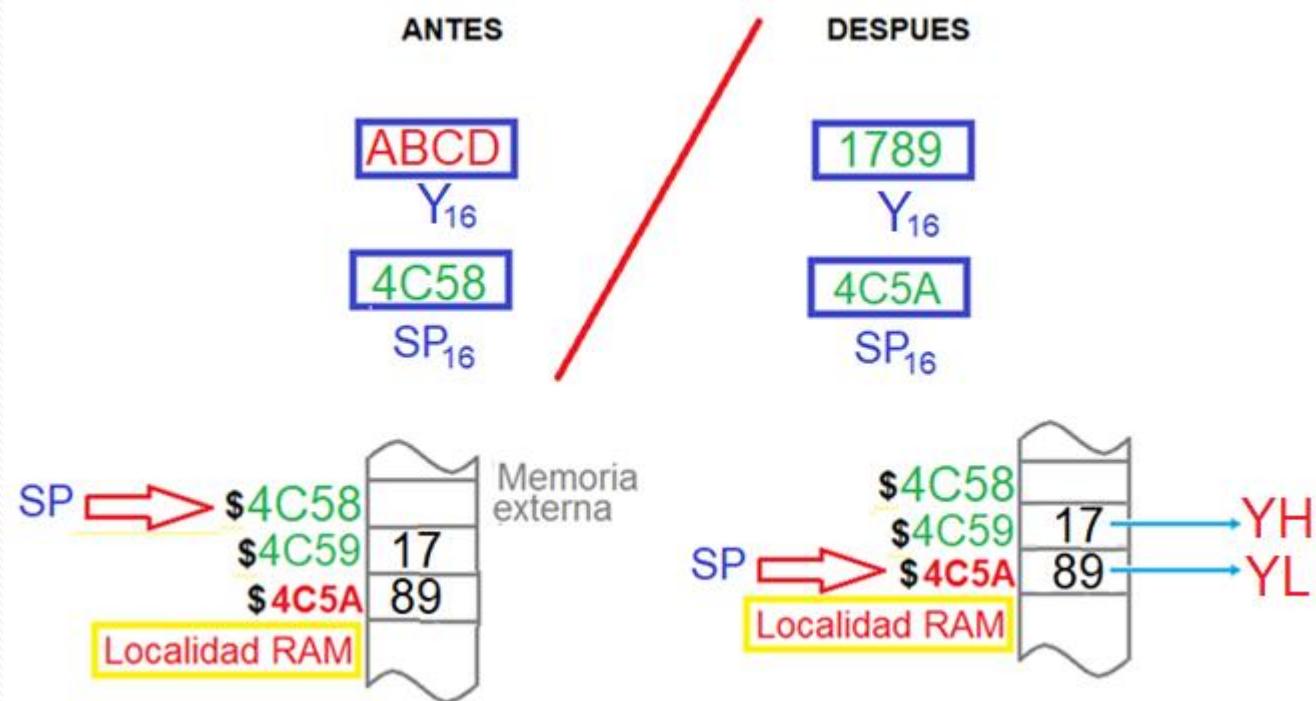
- Modo de direccionamiento **inherente**
- **Opcode:** \$183C    **Ciclo:** 5    **Byte:** 2
- Transfiere (copia) el contenido del registro **registro “Y”** en la dirección apuntada por el SP y después hace un decremento del SP en dos unidades.
- Ninguna bandera del CCR se actualiza.



# PULY

# $(SP) \rightarrow Y; (SP) + 2 \rightarrow SP$

- Modo de direccionamiento **inherente**
- **Opcode:** \$1838    **Ciclo:**5    **Byte:**2
- Transfiere (copia) el contenido de las dos localidades de la cúspide del SP hacia el registro “Y” haciendo un incremento del SP en dos unidades.
- Ninguna bandera del CCR se actualiza.



# El orden es importante...

PSHA

PSHB

PSHX

PSHY

⋮

PULY

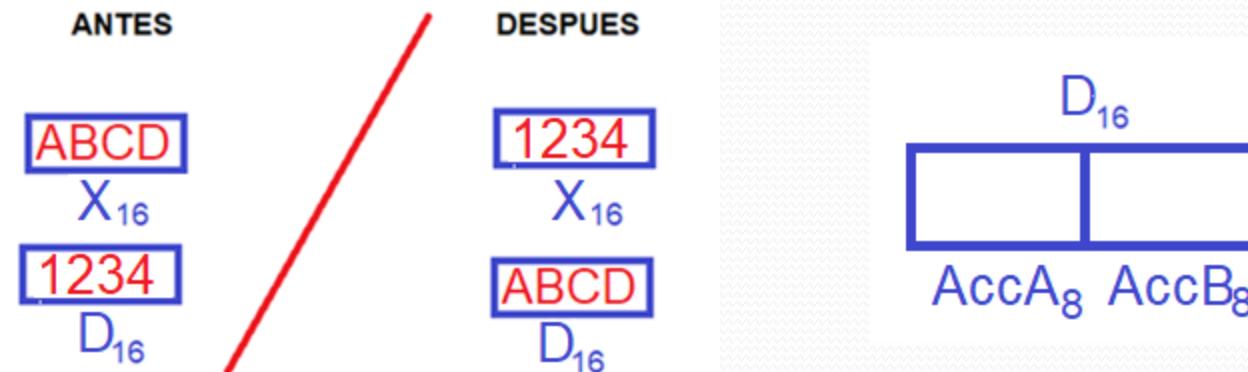
PULX

PULB

PULA

# XGDX

- Modo de direccionamiento **inherente**
- **Opcode:** \$8F      **Ciclo:**3      **Byte:**1
- Intercambia el contenido del registro “X” con el contenido del registro “D”
- Ninguna bandera del CCR se actualiza.



# XGDX

- Modo de direccionamiento **inherente**
- **Opcode:** \$188F      **Ciclo:4**      **Byte:2**
- Intercambia el contenido del registro “Y” con el contenido del registro “D”
- Ninguna bandera del CCR se actualiza.

