

## Instruct Transference

Tuesday, September 17, 2019 2:01 PM

→ LDAA

► LDAA # \$4C // M. Inmediato

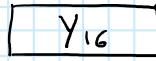
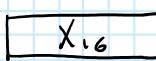
- Lo que estamos asignando es "4C" en una localidad de RAM

► LDAA \$4C // M. Directo

Check it



SP<sub>16</sub> "Stack Pointer"



//Registros

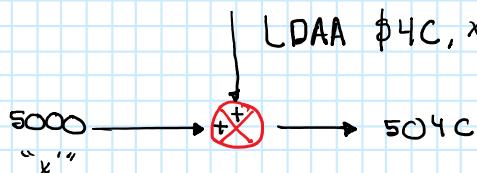
- Lo que estamos diciendo es que carguemos lo que hay en la localidad 4C
- \* Memoria desde 0000 hasta FFFF (RAM)

► LDAA \$4C5A // M. Extendido

- Lo mismo que el anterior pero con 3 bytes
- 3 bytes { 16 bits → Acomulador  
          { 8 bits → Operando

► LDAA \$4C,X // M. Indexado respecto en X

- Se suma el operando con el contenido del registro "X" para determinar la dirección del dato efectivo



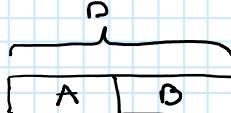
→ STAA // No tiene inmediato

► STAA \$4C // M. Directo

- Guarda el contenido en el acumulador "A" en la dirección de RAM (Responsabilidad programador)
- El operando es la dirección de 8 bits

→ LDD

## → LDU

-  D es el conjunto de ambas acumuladoras

► LDU # \$1234 // M. Inmediato

- Inicializa el contenido en los acumuladores "A" y "B"
- NOTA | si  $\# 16 = \# 0016$

► LDU \$4C // M. Directo

- Inicializa el contenido en los acumuladores no con el valor sino con lo que hay en esas direcciones

**NOTE** | MSB  $\triangleq$  Most Significant Bit

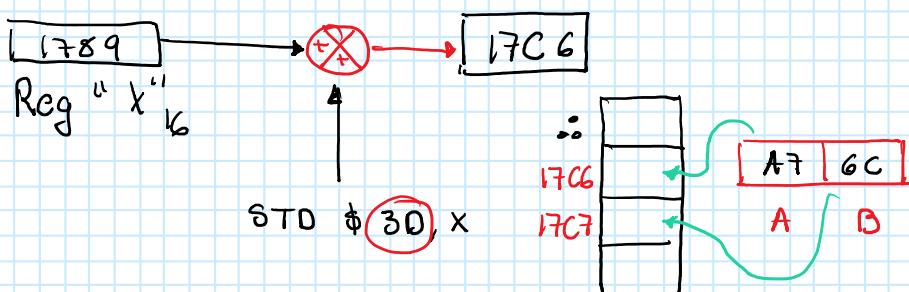
► LDU \$4C5A // M. Extendido

→ STD

► STD \$4C5A

- Guardamos lo que hay en ambos acumuladores en la dirección de memoria (Cuidado)

► STD \$30, X



→ LDS / → LD X / → LD Y

- Inicializa el contenido del registro  
"Stack Pointer"

► LOS # \$1234 // M. Inmediato

► LOS \$4C

**NOTA** • LD X \$30, X si se puede e incluso modifica el valor en el mismo registro

# Inst. Sin operandos | Stack

Tuesday, September 17, 2019 3:04 PM

► TAB

// M. Inherente

- Transfiere (copia) el contenido del registro acumulador "A" hacia el registro "B"

Before	After				
<table border="1"><tr><td>17</td></tr><tr><td>Acc A<sub>8</sub></td></tr></table>	17	Acc A <sub>8</sub>	<table border="1"><tr><td>17</td></tr><tr><td>Acc A<sub>8</sub></td></tr></table>	17	Acc A <sub>8</sub>
17					
Acc A <sub>8</sub>					
17					
Acc A <sub>8</sub>					
<table border="1"><tr><td>CB</td></tr><tr><td>Acc B<sub>8</sub></td></tr></table>	CB	Acc B <sub>8</sub>	<table border="1"><tr><td>17</td></tr><tr><td>Acc B<sub>8</sub></td></tr></table>	17	Acc B <sub>8</sub>
CB					
Acc B <sub>8</sub>					
17					
Acc B <sub>8</sub>					

► TBA

// Es el inverso del anterior

► TAP

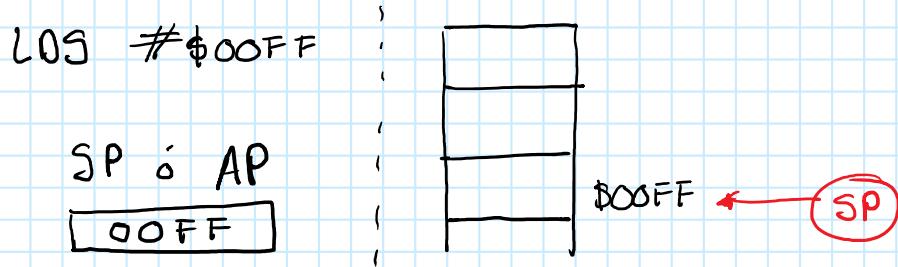
- Transfiere una copia del registro de estados hacia el acumulador "A"

Before	After																					
<table border="1"><tr><td>17</td></tr><tr><td>Acc R<sub>8</sub></td></tr></table>	17	Acc R <sub>8</sub>	<table border="1"><tr><td>17</td></tr><tr><td>Acc R<sub>8</sub></td></tr></table>	17	Acc R <sub>8</sub>	<table border="1"><tr><td>S</td><td>X</td><td>H</td><td>I</td><td>N</td><td>Z</td><td>V</td><td>C</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	S	X	H	I	N	Z	V	C	0	0	0	0	0	0	0	0
17																						
Acc R <sub>8</sub>																						
17																						
Acc R <sub>8</sub>																						
S	X	H	I	N	Z	V	C															
0	0	0	0	0	0	0	0															
<table border="1"><tr><td>CB</td></tr><tr><td>Acc A<sub>8</sub></td></tr></table>	CB	Acc A <sub>8</sub>	<table border="1"><tr><td>17</td></tr><tr><td>Acc A<sub>8</sub></td></tr></table>	17	Acc A <sub>8</sub>	<p>// Registro de estados</p>																
CB																						
Acc A <sub>8</sub>																						
17																						
Acc A <sub>8</sub>																						

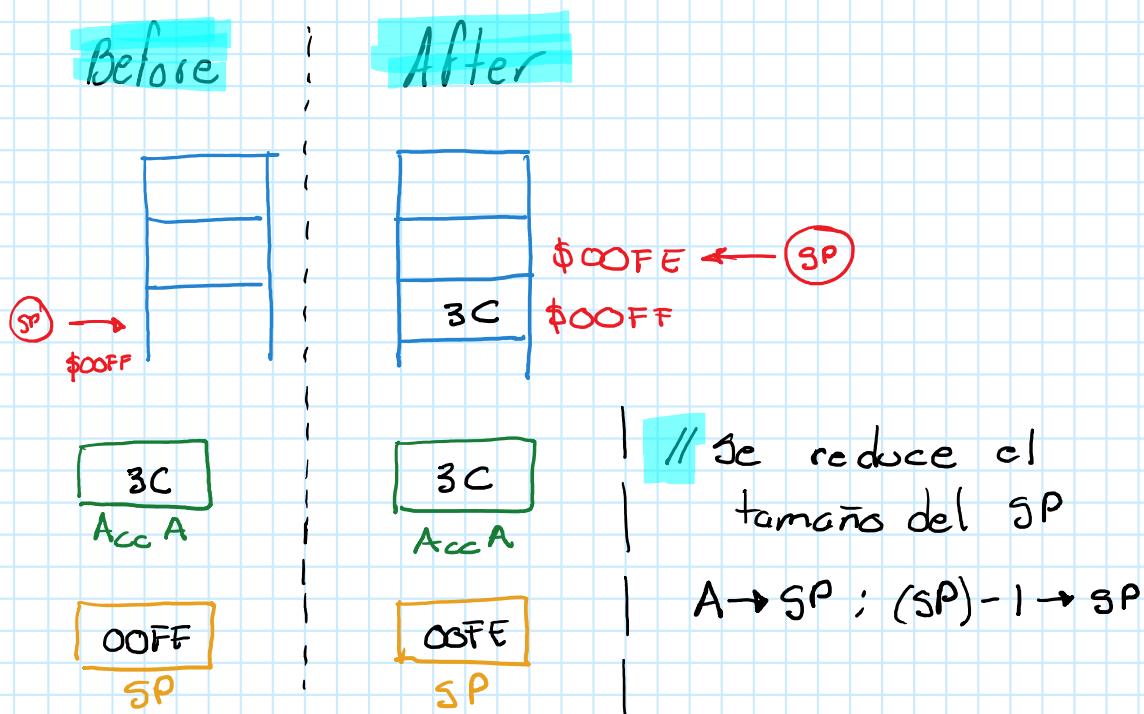
► TAP ← // Inverso

- Tener cuidado con los valores que le mandare mas

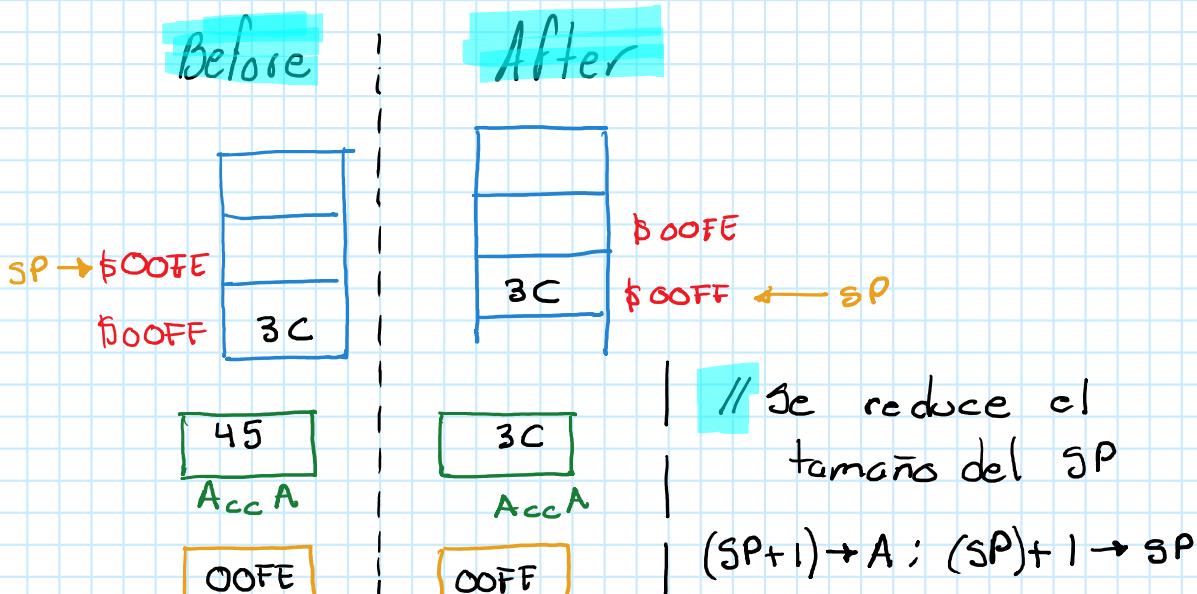
## ► Instrucciones de Stack



## ► PSHA



## ► PULA



Acc<sup>n</sup>



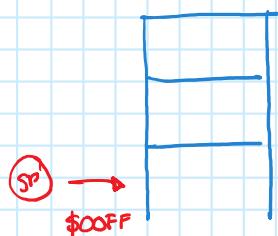
Acc<sup>n</sup>



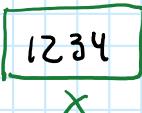
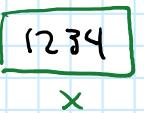
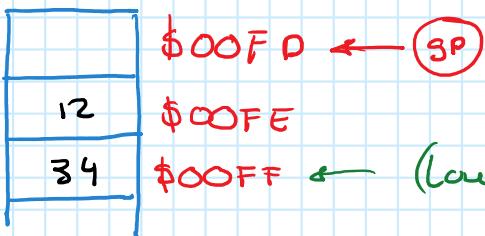
$(SP+1) \rightarrow A; (SP)+1 \rightarrow SP$

### ► PSHX

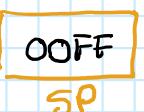
Before



After



// Se reduce el tamaño del SP



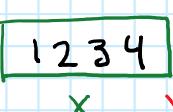
$(iX) \rightarrow SP-1; (SP-2);$   
 $\therefore SP-2 \rightarrow SP$

### NOTA

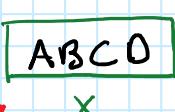
- Para el PULX primero borramos la primera parte de "X"
- → Cabe destacar que solo "movemos" el stack pointer (Sigue el contenido en las localidades).

### ► X6DX

Before



After



$\begin{array}{ c c }\hline D & \\ \hline A & B \\ \hline\end{array}$		$\begin{array}{ c c }\hline D & \\ \hline 12 & 34 \\ \hline A & B \\ \hline\end{array}$
---	--	---

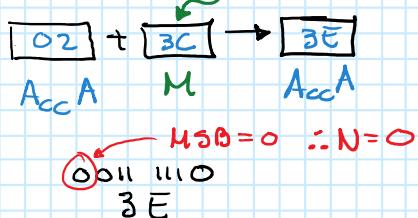
# Arithmetic Operations

Thursday, September 19, 2019 2:55 PM

**IMM**  
(Inmediato)

$$A + M \rightarrow A$$

ADDA # \$3C

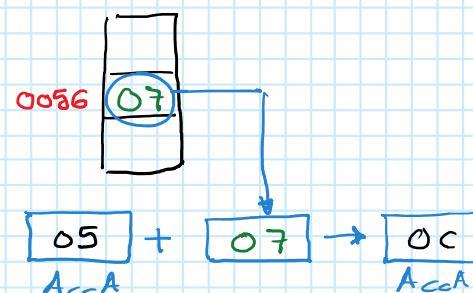


\* Ambos operandos deben estar en BCD y el resultado > 9  
  
N  $\triangleq$  Negative C  $\triangleq$  Carry  
Z  $\triangleq$  Zero

**DIR**

$$A + M \rightarrow A$$

ADDA \$ 56



$$\begin{array}{r} 0000\ 0101 \\ + 0000\ 0111 \\ \hline 0000\ 1011 \end{array}$$

C C  
→ R>9

$$\begin{array}{r} 0000\ 0101 \\ + 0000\ 0111 \\ \hline 0001\ 0010 \end{array}$$

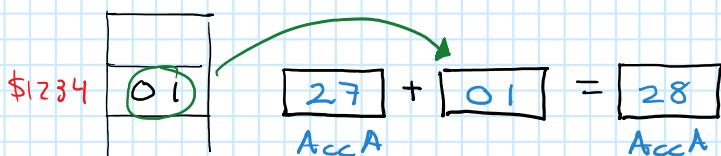
BCD  
→ 12 en BCD

// Flags  
N=0 Y=0  
Z=0 H=1  
C=0 comple

**EXT**

// Es lo mismo que el directo salvo por la dirección de memoria que almacena suman do

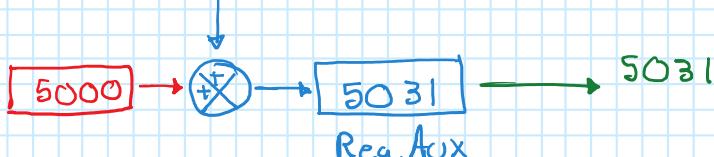
ADDA \$1234

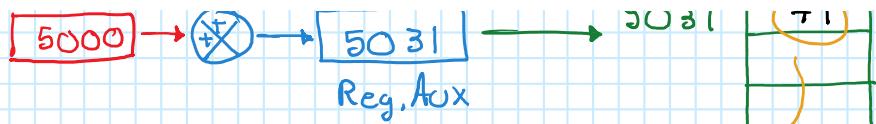


N=0 H=0  
Z=0  
C=0  
Y=0

**IND, X**

ADDA \$31,X





$$\therefore A + M \rightarrow A$$

$$N=0 \quad Y=0$$

$$Z=0 \quad H=0$$

$$C=0$$

$$03 \quad M \quad 24$$

AccA AccA

IMM  
(Inmediato)

SUBA # \$3C

BCD

$$02 - 3C \rightarrow C6$$

AccA

$$N=1 \quad Y=0$$

$$Z=0 \quad H=0$$

$$C=0$$

// Para restar

$$3C = 0011\ 1100$$

$$+ 1100\ 0011$$

$$\hline$$

Complemento a 2

$$\begin{array}{r} 1100 \\ \hline 1100 \end{array}$$

C 4

∴ Ahora "restamos"

MSB

$$+ 1100\ 0100$$

$$+ 0000\ 0010$$

$$\hline$$

$$\therefore N=1 \quad \begin{array}{r} 1100 \\ \hline 0110 \end{array}$$

C G

// Otro ejemplo

$$27 - 01 =$$

$$\begin{array}{r} 0000\ 0001 \\ + 1111\ 1100 \\ \hline 1111\ 1111 \end{array}$$

$$+ 0010\ 0111 \leftarrow 27$$

$$\hline$$

$$\textcircled{1} 0010\ 0110 \leftarrow 26$$

Carry

# Arithmetic Operations

Tuesday, September 24, 2019 2:05 PM

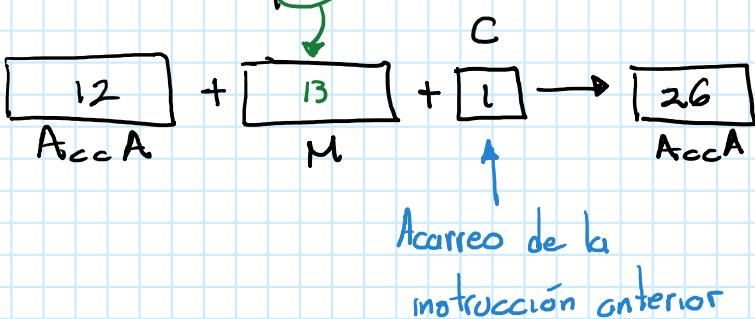
► IMM

$$A + M + C \rightarrow A$$

► ADCA

\* Suma también el carry

ADCA # \$13



$$\begin{array}{r} 12 \\ + 0001\ 0010 \\ 0001\ 0011 \\ 0000\ 0001 \\ \hline 0010\ 0100 \end{array}$$

$$\begin{array}{l} N=0 \quad V=0 \\ Z=0 \quad H=0 \\ C=0 \end{array}$$

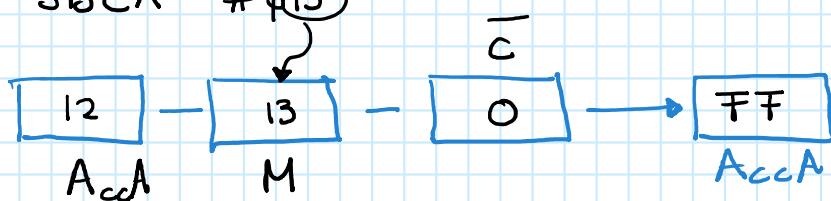
► SBCA

} Resta donde incluye el carry

IMM

$$A - M - \bar{C} \rightarrow A$$

SBCA # \$13



\* If  $C = 1$  then  $\bar{C} = 0$

// Operación

// Complemento a 2 (Negativo)

$$13 = 0001\ 0011$$

$$\begin{array}{r} 1110\ 1100 \\ + 1110\ 1100 \\ \hline 1110\ 1101 \end{array}$$

LD  
(C.2) - 13

$$\begin{array}{r} 1110\ 1101 \leftarrow -13 \\ + 0001\ 0010 \leftarrow 12 \\ \hline 1111\ 1111 \\ \overbrace{\quad\quad\quad\quad\quad}^F \overbrace{\quad\quad\quad\quad\quad}^F \end{array}$$

$$N=1 \quad Z=0$$

$$V=0$$

$$C=0 \quad // \text{Nuevo Carry}$$

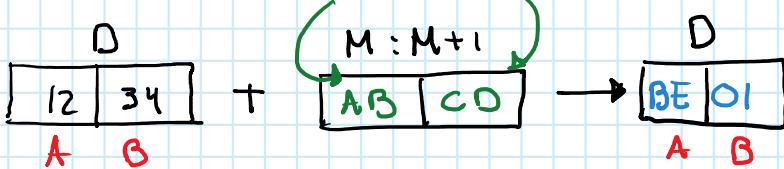
// No aplica H en resta

► ADDD

**IMM**

$$D + [M:M+1] \rightarrow D$$

$$ADD D \# \$ABCD$$



$$\begin{array}{r}
 0001 \quad 0010 \quad 0011 \quad 0100 \leftarrow 1234 \\
 1010 \quad 1011 \quad 1100 \quad 1101 \leftarrow ABCD \\
 \hline
 1011 \quad 1110 \quad 0000 \quad 0001 \\
 \text{D} \qquad \text{E} \qquad \text{O} \qquad \text{T}
 \end{array}$$

$$\begin{cases} N = 1 \\ V = 0 \\ Z = 0 \\ C = 0 \end{cases}$$

Similar

$$ADD D \# \$CD$$

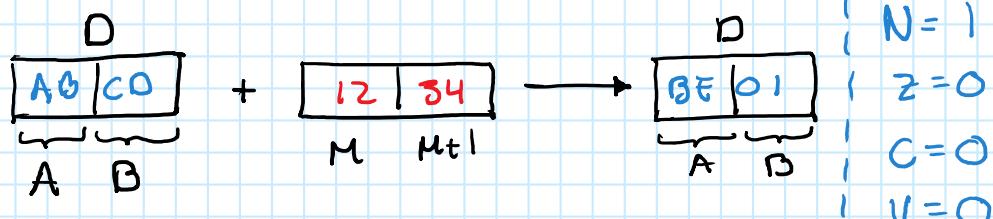
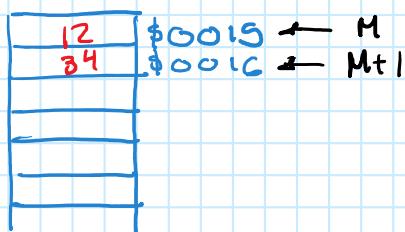
~~$$ADD A \# \$AB$$~~

Para considerar el carry de la primera suma

**DIR**

$$D + [M:M+1] \rightarrow D$$

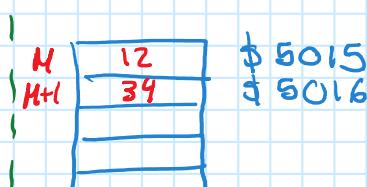
$$ADD D \$15$$

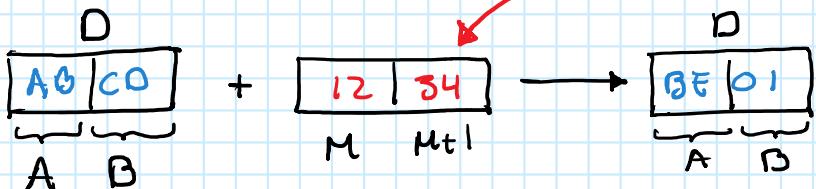
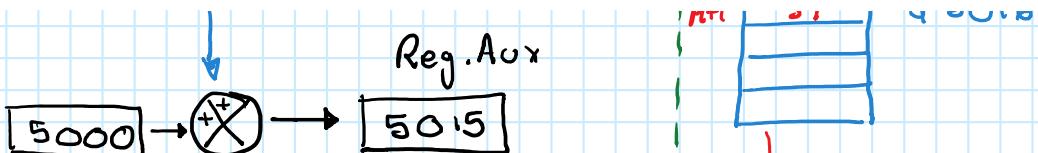
**IND**

$$ADD \$15, X$$

$$\text{if } x = 5000$$

Reg. Aux

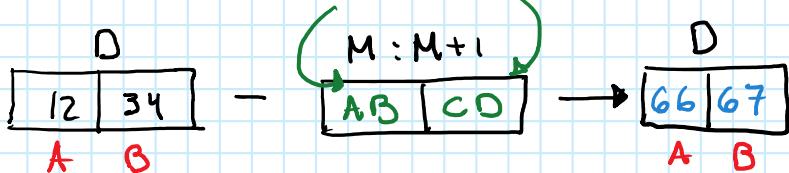




► SUBD

IMM  $D - [M:M+1] \rightarrow D$

$ADD D \# \$ABC0$



$$ABC0 \rightarrow 1010 \ 1011 \ 1100 \ 1101$$

$$\begin{array}{r} 0101 \ 0100 \ 0011 \ 0010 \\ + 0101 \ 0100 \ 0011 \ 0010 \\ \hline 0101 \ 0100 \ 0011 \ 0011 \end{array}$$

$$-ABC0 \rightarrow \underbrace{0101}_5 \ \underbrace{0100}_4 \ \underbrace{0011}_3 \ \underbrace{0011}_3$$

$$\begin{array}{r} 1234 \rightarrow 0001 \ 0010 \ 0011 \ 0100 \\ -ABC0 \rightarrow 0101 \ 0100 \ 0011 \ 0011 \\ \hline 0110 \ 0110 \ 0110 \ 0111 \\ 0 \rightarrow \underbrace{0110}_6 \ \underbrace{0110}_6 \ \underbrace{0110}_6 \ \underbrace{0111}_7 \end{array}$$

$N=0$   
 $Z=0$   
 $V=0$   
 $C=0$

► MUL

INH  $MUL (AccA)_8 \cdot (AccA)_8 = (AccP)_{16}$

$$\begin{array}{r} 7 \rightarrow 0000 \ 0111 \\ 5 \rightarrow \times 0000 \ 0101 \\ \hline 00000111 \end{array} \quad \begin{array}{r} 32 \ 16 \ 8 \ 4 \ 2 \ 1 \\ | \quad | \quad | \quad | \quad | \quad | \\ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \\ \hline 25 .. \end{array}$$

$$\begin{array}{r}
 \textcircled{5} \rightarrow \times \begin{array}{r} 00000 \\ 0101 \end{array} \\
 \hline
 \begin{array}{r} 000000111 \\ 000000000 \\ 00000111 \end{array} \\
 \vdots \\
 \begin{array}{r} 000000000000100011 \\ \underbrace{\hspace{1cm}}_0 \quad \underbrace{\hspace{1cm}}_0 \quad \underbrace{\hspace{1cm}}_2 \quad \underbrace{\hspace{1cm}}_3 \\ A \quad B \end{array} \\
 \hline
 0
 \end{array}$$

$$\begin{array}{c}
 \overbrace{\phantom{000000000000100011}}^{\text{35}} // \\
 \boxed{7} \cdot \boxed{5} = \boxed{00} \boxed{23} \\
 A \quad B \quad A \quad B
 \end{array}$$

► IDIV

► F DIV

- Ambas hacen  $\frac{D}{X} \Rightarrow X$   $\underbrace{r}_{\text{residuo}} \Rightarrow D$

$$\begin{array}{r}
 D \boxed{0007} \\
 \times \boxed{0002} \\
 \hline
 D \boxed{0003} \\
 \times \boxed{0005} \\
 \text{Después}
 \end{array}$$

$$\begin{array}{r}
 D \boxed{0007} \\
 \times \boxed{0002} \\
 \hline
 D \boxed{0003} \\
 \times \boxed{8000} \\
 \text{Después}
 \end{array}$$

$$\begin{array}{r}
 0003, 8000 \\
 \downarrow \downarrow \downarrow \downarrow \\
 \boxed{0000} \boxed{0000} \boxed{0000} \boxed{00} \boxed{11} \\
 \underbrace{\hspace{1cm}}_3 \quad \underbrace{\hspace{1cm}}_3 \quad \underbrace{\hspace{1cm}}_3 \quad \underbrace{\hspace{1cm}}_3 \quad \underbrace{\hspace{1cm}}_3 \\
 \boxed{.5} \boxed{.25} \boxed{.125} \\
 \downarrow \downarrow \downarrow \\
 \boxed{1000} \boxed{0000} \boxed{0000} \boxed{0000} \\
 \underbrace{\hspace{1cm}}_8 \quad \underbrace{\hspace{1cm}}_0 \quad \underbrace{\hspace{1cm}}_0 \quad \underbrace{\hspace{1cm}}_0
 \end{array}
 \quad \left. \begin{array}{l} 8421 \\ \{ D \\ \{ X \end{array} \right)$$

► Comparaciones

→ No se modifica el valor localidad

ABA

$A+B \rightarrow A$

ADDA

$A+M \rightarrow A$

SBA

$A-B \rightarrow A$

SUBA

$A-M \rightarrow A$

SBA	$A - B \rightarrow A$	SUBA	$A - M \rightarrow A$
CBA	$A - B$	CMPA	$A - M$

ADD	$D + [M : M+1] \rightarrow D$	CPX CPY	$x - [M : M+1]$
SUBD	$D - [M : M+1] \rightarrow D$		$y - [M : M+1]$
CPD	$D - [M : M+1]$		

### ► Extra (Teating)

TSTA       $A - O$       | Similar a LDDA

// Modifica las banderas de 'A'

# Arithmetic Operations

Tuesday, October 1, 2019 2:12 PM

→ CLRA

antes	después
Acc A	Acc A
4C	00

// Equivale a

LDAA # \$00

→ CLR  $\# 457C \leftarrow RAM$

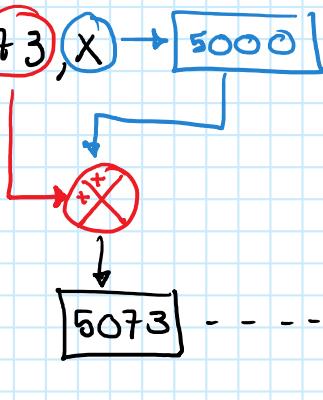
before	After
FA	\$457C

N=0 V=0

Z=1 C=0

→ CLR

~~\$73~~, X → 5000



After

00	\$5073
----	--------

→ COM A

Before

4C
Acc A

After

B3
Acc A

4  
C

0100 1100 → 1011 0011  
complemento

V=0 C=1 N=1 Z=0

COM  $\$457C \leftarrow RAM$

Before

457C
F

After

05
5

1111 1010 → 0000 0101

→ NEG A

→ NEG  $\$457C \leftarrow RAM$

→ NEGA

Before		After
<b>4C</b>		<b>B4</b>
Acc A		Acc A
4 C		
0100 1100	$\rightarrow$	1011 0011
	complement to	1
V=0		1011
N=1		0100
C=1		1
Z=0		0

→ NEG \$457C ← RAM

Before	After
$\$457C$  $\underbrace{1111}_{F} \quad \underbrace{1010}_{A}$	$\$457C$  $0000 \quad 0101$
	$+$ <hr/> $0000 \quad 0110$
	$O$ <hr/> $6$

→ DECA

INH | Ext

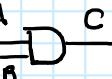
→ DEC 457C

BEFORE	AFTER	BEFORE	AFTER
4C AccA	4B AccA	\$457C FA	\$457C F9

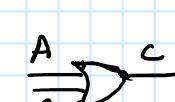
// También hay decremento en B y sus reactivos en X, Y y B con incremento y decremento.

# Logic Operations

Tuesday, October 1, 2019 2:38 PM

→ AND  $\wedge$  

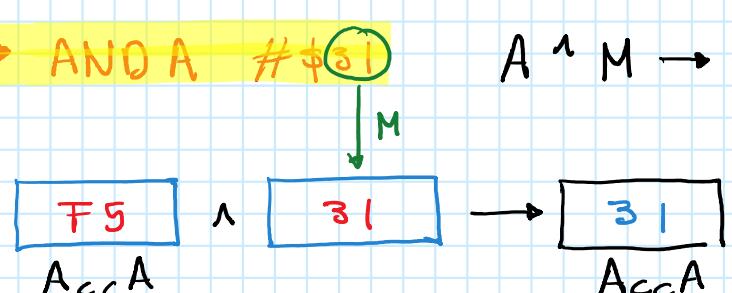
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

→ OR  $\vee$  

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

→ XOR  $\oplus$  

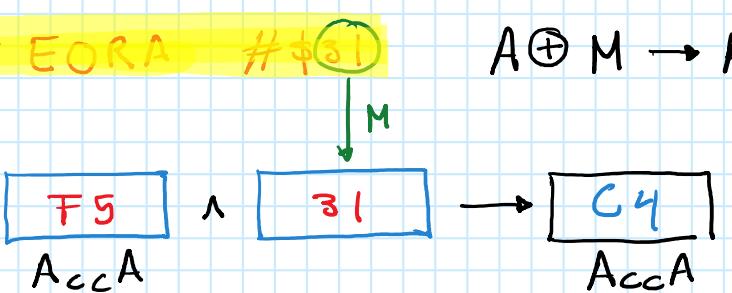
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

► ANDA # $\#31$  

$$A \wedge M \rightarrow A$$

$$\begin{array}{r} 1111 \quad 0101 \\ \wedge \quad 0011 \quad 0001 \\ \hline 0011 \quad 0001 \end{array}$$

3      1

► EORA # $\#31$  

$$A \oplus M \rightarrow A$$

$$\begin{array}{r} 1111 \quad 0101 \\ \oplus \quad 0011 \quad 0001 \\ \hline 1100 \quad 0101 \end{array}$$

C      4

NOTA

} ANDA  $A \wedge M \rightarrow A$  // Alteramos  
 R  $\leftarrow A$   $A \wedge M$  // En el analizador

**NOTA**

$\left\{ \begin{array}{ll} \text{ANUDAR} & A^M \rightarrow A \\ \text{BITA} & A^M \end{array} \right.$

"mitigaciones"

// Jó lo evolvamos

↳ Alteran banderas

# Salto

Tuesday, October 1, 2019 3:37 PM

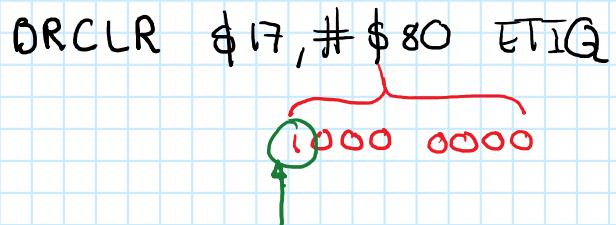
→ BRCLR \$17, #\$80 ETIQ

// Directo

- ① Salta a "ETIQ" si el contenido de la localidad "\$17" tiene el bit más significativo (# \$80) en bajo



0100 0101



// Si estos son ceros  
respecto a los del  
otro ENTONCES  
saltamos

// Con los unos  
indicamos  
que evaluaremos  
(Alto)

→ BRCLR \$17, #\$80 ETIQ

// Códigos similares

→ LDAA \$17  
ANDA #\$80  
BEQ ETIQ

→ BRSET \$25, #\$55 ETIQUETA

// Directo



\$0025

0111 1001  
1000 0110  
^ 0101 0101  
0000 0100

Negado

?  $M \cdot mm = \emptyset$  // Lo que hace

// Forma alternativa a

LDAA \$25

COMA

ANDA #\$55

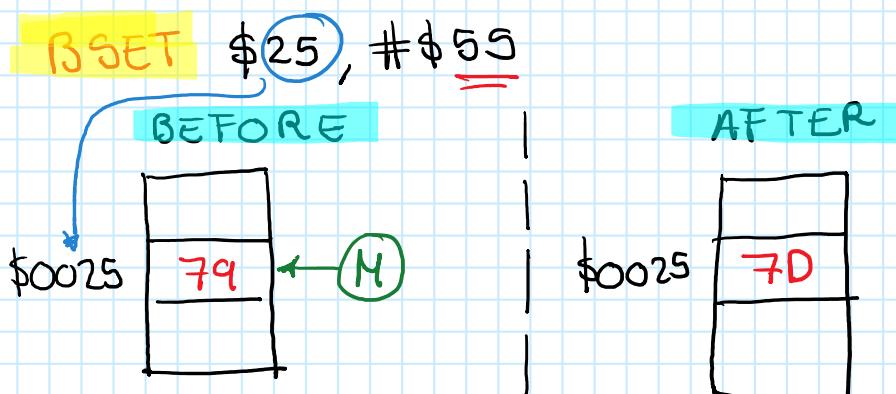
BEQ Etiqueta

// Recuerda que  
"NEGA" es comple  
mento a 2

→ BSET \$25, x, #\$55 Etiqueta // Indexado

// Lo mismo sólo que se suma "x" a "25"  
para llegar a una localidad más alta

### ► Instrucciones de Control



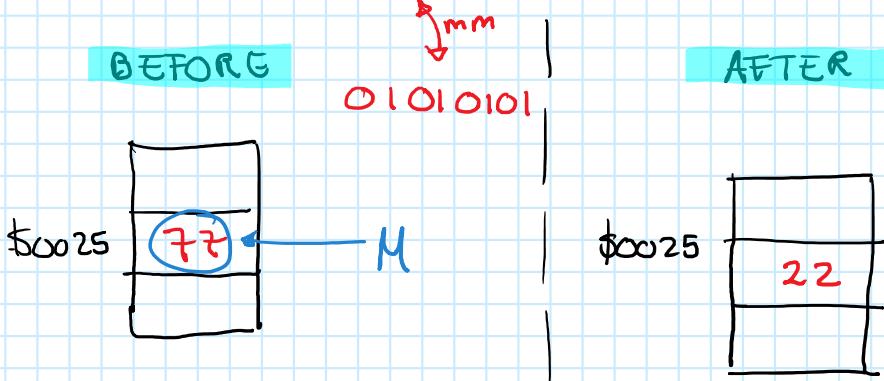
$$\begin{array}{r} (OR) \vee \\ \text{M} \quad 0111\ 1001 \leftarrow M (\text{Máscara}) \\ mm \quad 0101\ 0101 \leftarrow mm \\ \hline 0111\ 1101 \end{array}$$

0 1 1 1 1 0 1  
7      D

- Subimos los bits
- Utilizamos valor en la localidad y "mm"
- Utilizamos "OR"

BCLR \$25, #~~\$55~~

**BCLR** \$25, #~~\$55~~



(AND)

$$\begin{array}{r}
 01110111 \leftarrow M \\
 10101010 \leftarrow mm \\
 \hline
 00100010
 \end{array}$$

2      2

## ► Modificadores de indicadores

CLC	Clear Carry Bit	$0 \rightarrow C$
CLI	Clear Interrupt Mask	$0 \rightarrow I$
CLV	Clear Overflow Flag	$0 \rightarrow V$
SEC	Set Carry	$1 \rightarrow C$
SEI	Set Interrupt Mask	$1 \rightarrow I$
SEV	Set Overflow	$1 \rightarrow V$

- Carry, Overflow e Interrupt Mask
- Clear  $\rightarrow 0$
- Set  $\rightarrow 1$