

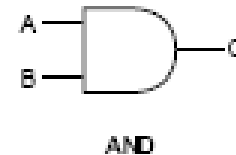
Set de instrucciones del MC68HC11

Instrucciones lógicas

ANDA #\$4C

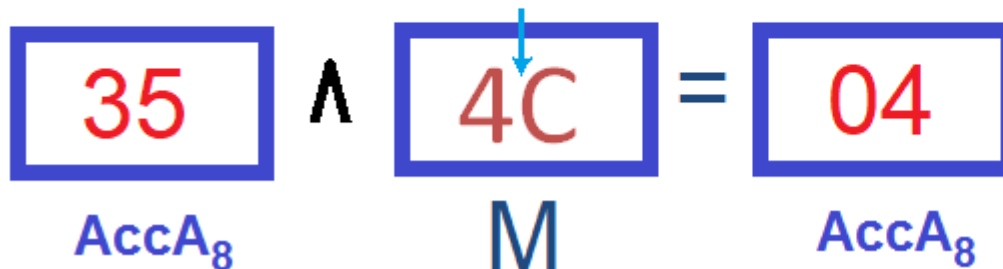
$$A \wedge M \rightarrow A$$

- Modo de direccionamiento **inmediato** (#)
- **Opcode:** \$84 **Ciclo:**2 **Byte:**2
- Ejecuta la función lógica de AND entre el contenido del acumulador "A" con el operando de 8 bits de forma inmediata.
- Actualiza banderas N, Z, V =0



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

ANDA #\$4C



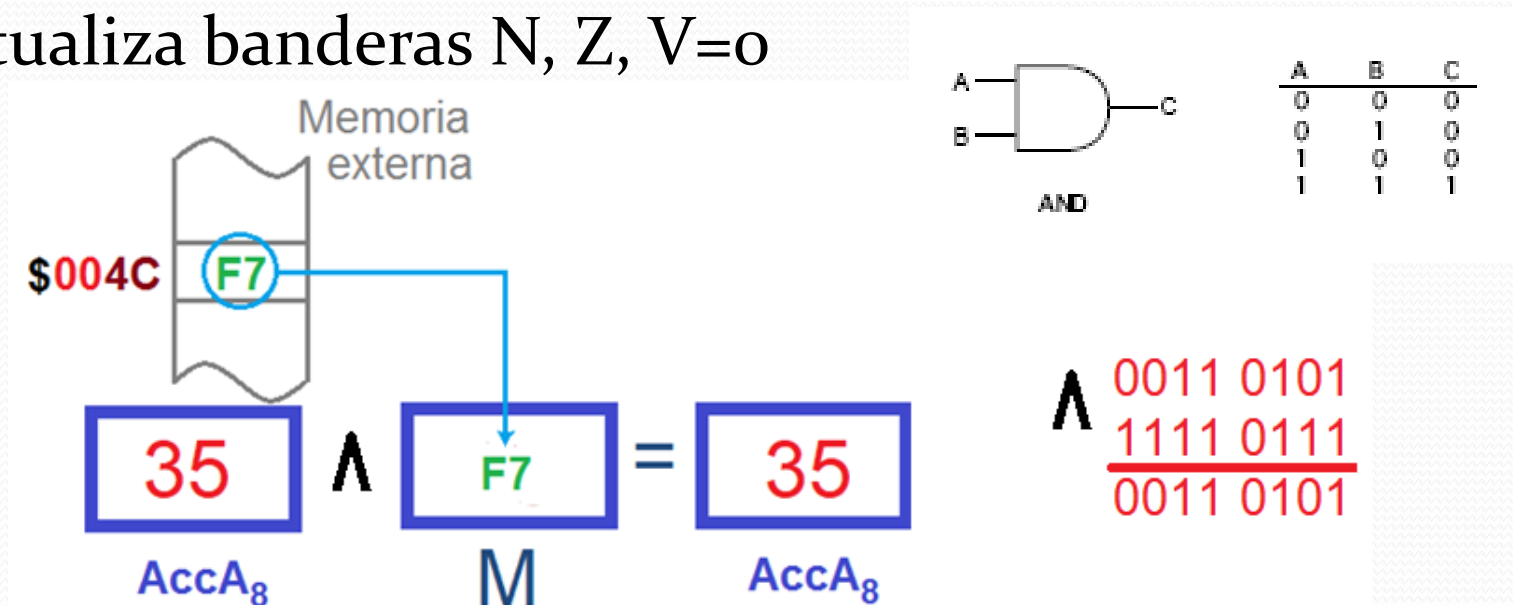
\wedge

0011	0101
0100	1100
<hr/>	
0000	0100

ANDA \$4C

$$A \wedge M \rightarrow A$$

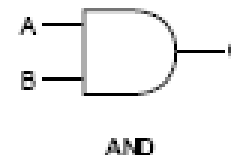
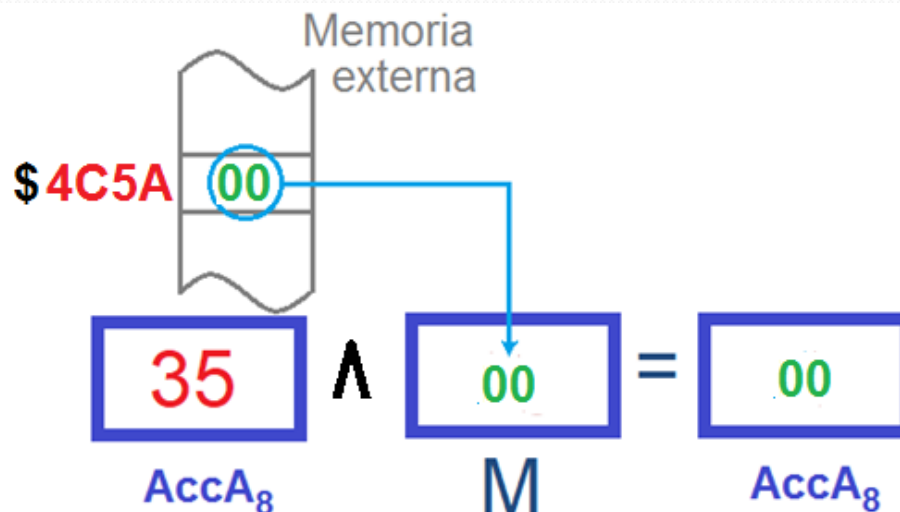
- Modo de direccionamiento **directo**
- **Opcode:** \$94 **Ciclo:**3 **Byte:**2
- Ejecuta la función lógica de AND entre el contenido del acumulador "A" con el contenido de la dirección dada por el operando de 8 bits.
- Actualiza banderas N, Z, V=0



ANDA \$4C5A

$$A \wedge M \rightarrow A$$

- Modo de direccionamiento **extendido**
- **Opcode:** \$B4 **Ciclo:**4 **Byte:**3
- Ejecuta la función lógica de AND entre el contenido del acumulador "A" con el contenido de la memoria dada por el operando de 16 bits.
- Actualiza banderas N, Z, V=0



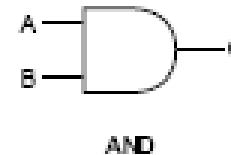
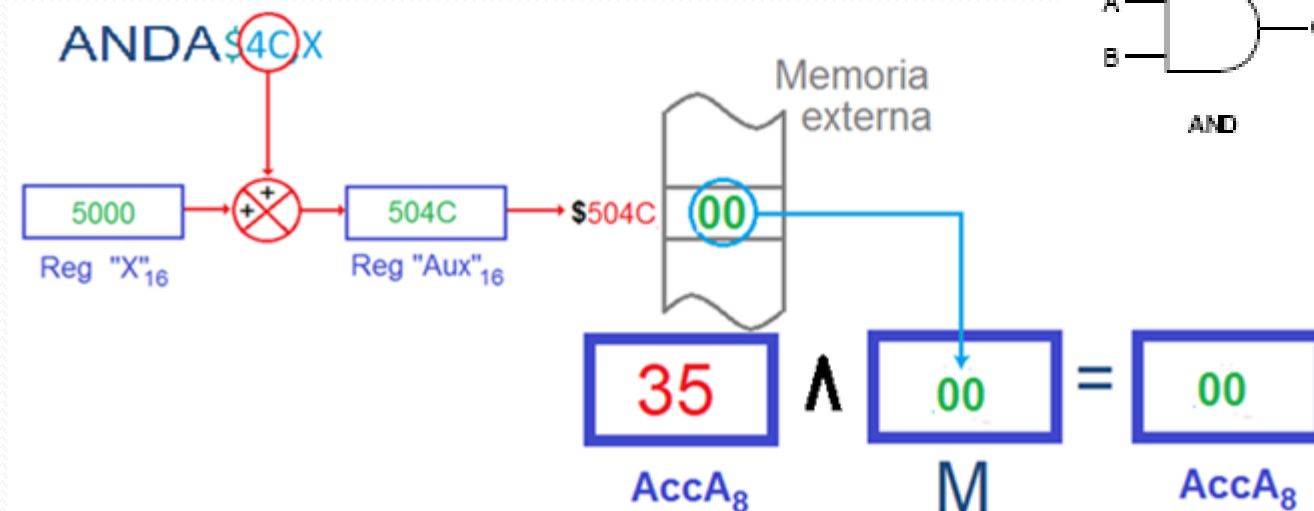
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

$$\begin{array}{r} \wedge \quad 0011 \ 0101 \\ \quad 0000 \ 0000 \\ \hline \quad 0000 \ 0000 \end{array}$$

ANDA \$4C,X

$$A \wedge M \rightarrow A$$

- Modo de direccionamiento **Indexado**
- **Opcode:** \$AB **Ciclo:**4 **Byte:**2
- Ejecuta la función lógica de AND entre el contenido del acumulador "A" y el contenido de la localidad de memoria dada por la suma del operando con el registro X.
- Actualiza banderas N, Z, V=0



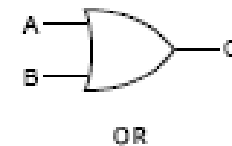
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

$$\begin{array}{r} \wedge \quad 0011 \ 0101 \\ \quad 0000 \ 0000 \\ \hline \quad 0000 \ 0000 \end{array}$$

ORAA #\$4C

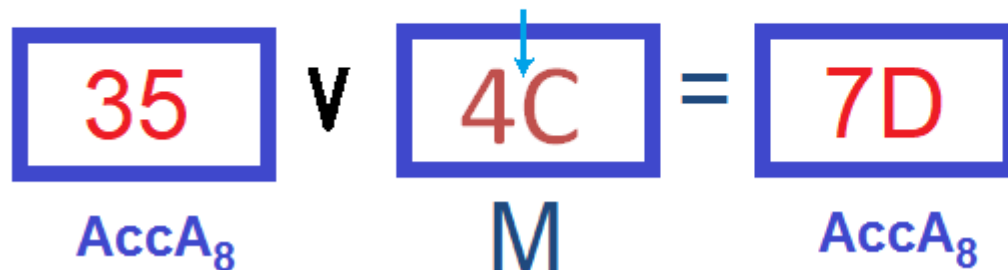
$A \vee M \rightarrow A$

- Modo de direccionamiento **inmediato** (#)
- **Opcode:** \$8A **Ciclo:**2 **Byte:**2
- Ejecuta la función lógica de OR entre el contenido del acumulador "A" con el operando de 8 bits de forma inmediata.
- Actualiza banderas N, Z, V =0



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

ORAA#\$4C

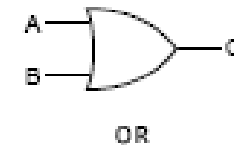
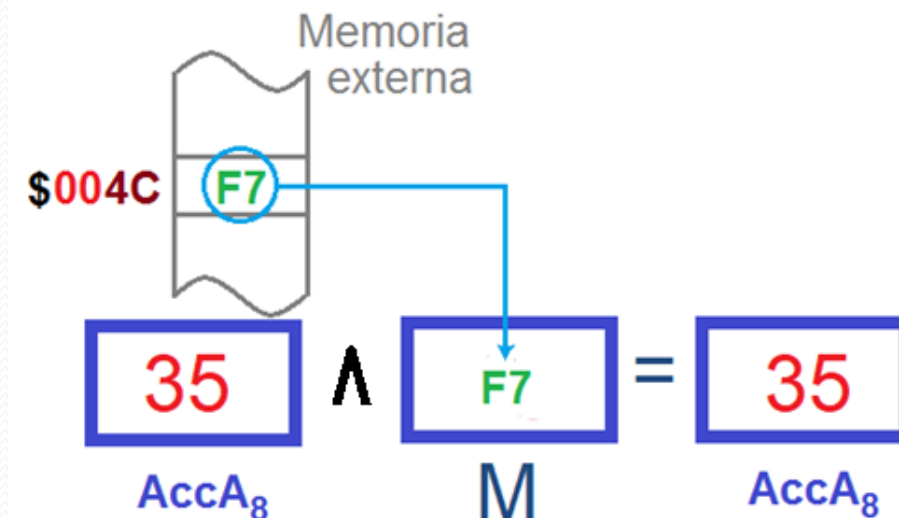


$$\begin{array}{r} \vee \quad 0011 \ 0101 \\ \quad 0100 \ 1100 \\ \hline \quad 0111 \ 1101 \end{array}$$

ORAA \$4C

$$A \vee M \rightarrow A$$

- Modo de direccionamiento **directo**
- **Opcode:** \$9A **Ciclo:**3 **Byte:**2
- Ejecuta la función lógica de OR entre el contenido del acumulador "A" con el contenido de la dirección dada por el operando de 8 bits.
- Actualiza banderas N, Z, V=0



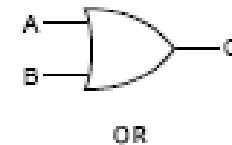
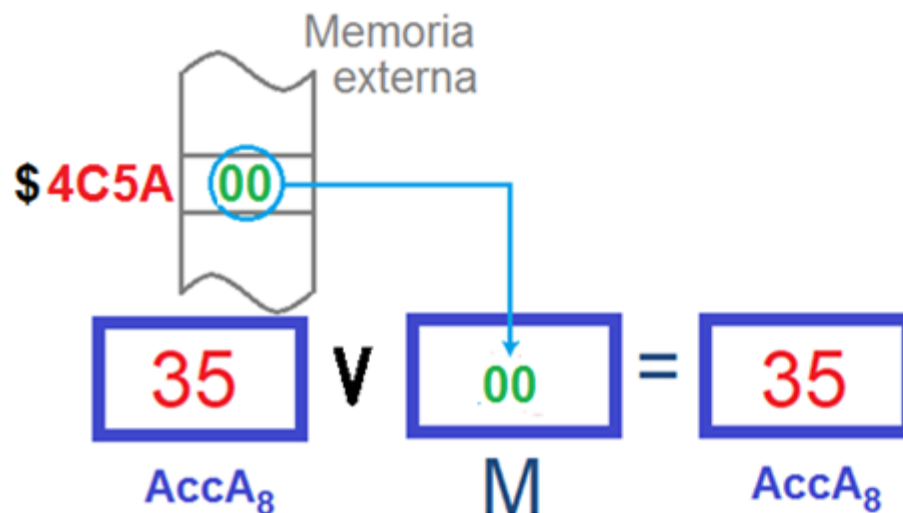
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

$$\begin{array}{r} \Lambda \quad 0011 \ 0101 \\ \quad 1111 \ 0111 \\ \hline \quad 0011 \ 0101 \end{array}$$

ORAA \$4C5A

$A \vee M \rightarrow A$

- Modo de direccionamiento **extendido**
- **Opcode:** \$BA **Ciclo:**4 **Byte:**3
- Ejecuta la función lógica de OR entre el contenido del acumulador "A" con el contenido de la memoria dada por el operando de 16 bits.
- Actualiza banderas N, Z, V=0



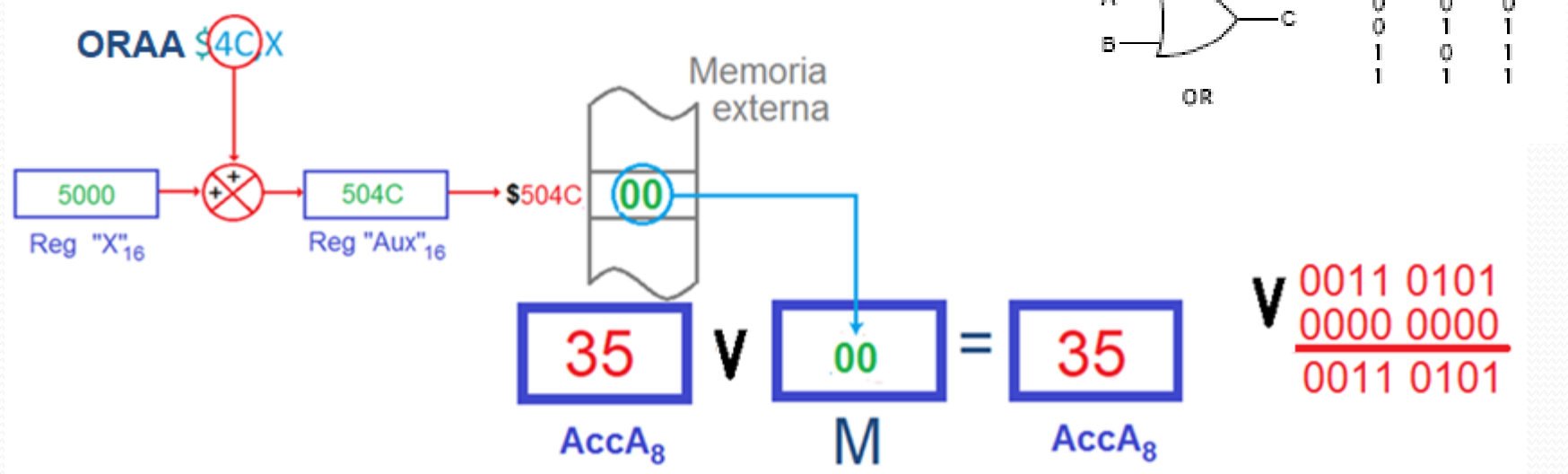
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

$$\begin{array}{r}
 \vee \quad 0011 \ 0101 \\
 \quad 0000 \ 0000 \\
 \hline
 \quad 0011 \ 0101
 \end{array}$$

ORAA \$4C,X

A M → A

- Modo de direccionamiento **Indexado**
- **Opcode:** \$AA **Ciclo:**4 **Byte:**2
- Ejecuta la función lógica de OR entre el contenido del acumulador "A" y el contenido de la localidad de memoria dada por la suma del operando con el registro X.
- Actualiza banderas N, Z, V=0



EORA #\$4C

$$A \oplus M \rightarrow A$$

- Modo de direccionamiento **inmediato** (#)
- **Opcode:** \$88 **Ciclo:**2 **Byte:**2
- Ejecuta la función lógica de XOR entre el contenido del acumulador "A" con el operando de 8 bits de forma inmediata.
- Actualiza banderas N, Z, V =0



Exclusive-OR

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

EORA #\$4C

$$\boxed{35} \oplus \boxed{4C} = \boxed{79}$$

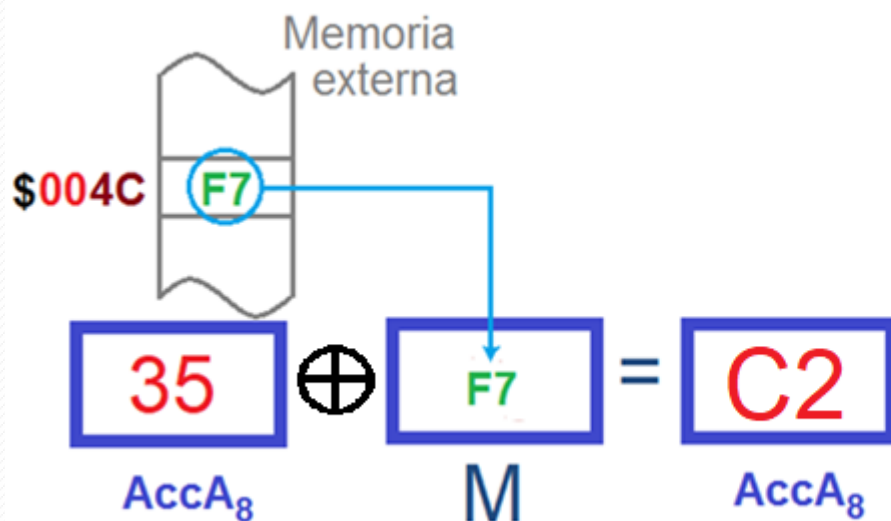
$\text{AccA}_8 \qquad \qquad \qquad M \qquad \qquad \qquad \text{AccA}_8$

$$\oplus \begin{array}{r} 0011 \ 0101 \\ 0100 \ 1100 \\ \hline 0111 \ 1001 \end{array}$$

EORA \$4C

$$A \oplus M \rightarrow A$$

- Modo de direccionamiento **directo**
- **Opcode:** \$98 **Ciclo:**3 **Byte:**2
- Ejecuta la función lógica de XOR entre el contenido del acumulador "A" con el contenido de la dirección dada por el operando de 8 bits.
- Actualiza banderas N, Z, V=0



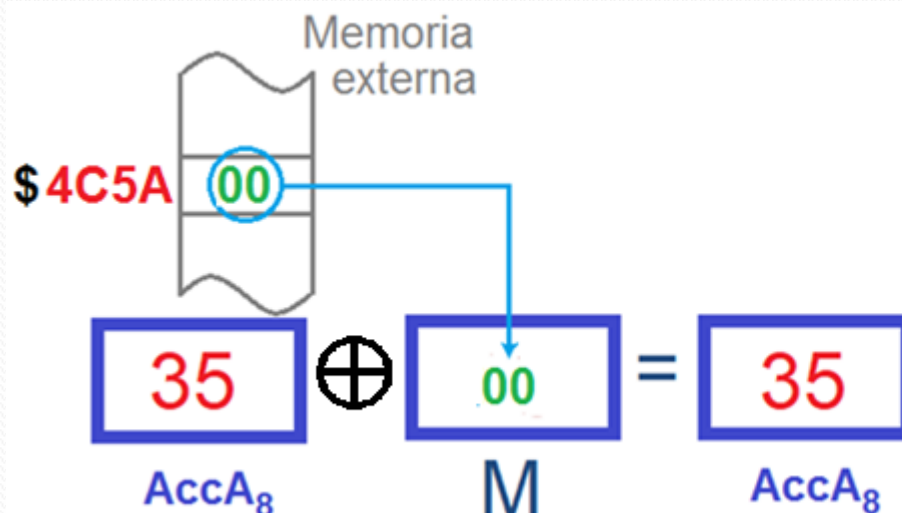
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

$$\begin{array}{r} \oplus \quad 0011 \ 0101 \\ \quad 1111 \ 0111 \\ \hline \quad 1100 \ 0010 \end{array}$$

EORA \$4C5A

$$A \oplus M \rightarrow A$$

- Modo de direccionamiento **extendido**
- **Opcode:** \$B8 **Ciclo:**4 **Byte:**3
- Ejecuta la función lógica de XOR entre el contenido del acumulador "A" con el contenido de la memoria dada por el operando de 16 bits.
- Actualiza banderas N, Z, V=0



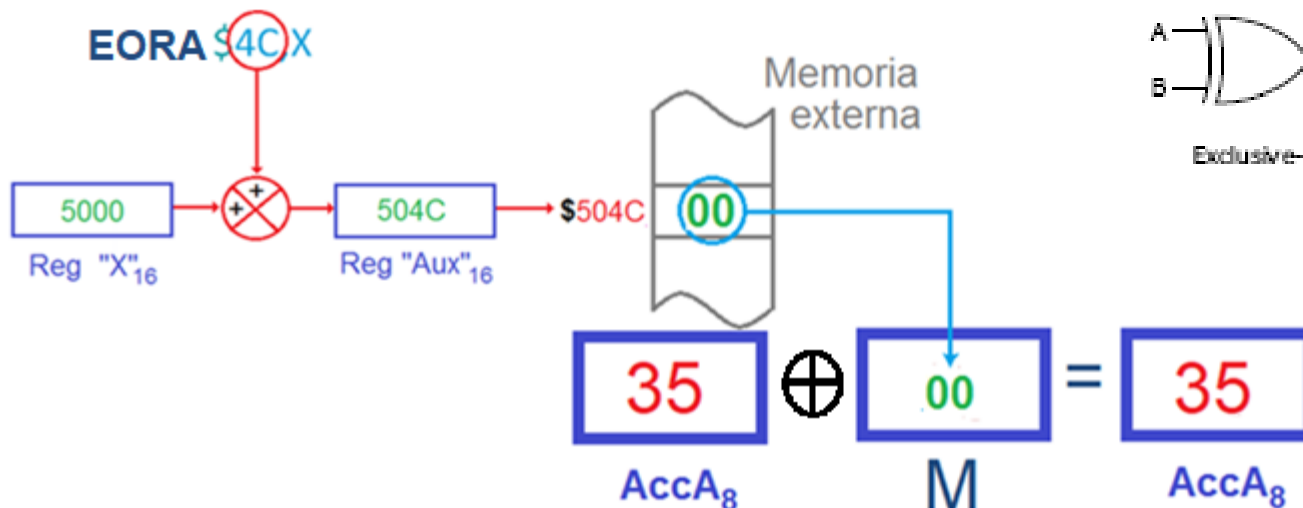
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

$$\begin{array}{r} \oplus \quad 0011 \ 0101 \\ \quad 0000 \ 0000 \\ \hline \quad 0011 \ 0101 \end{array}$$

EORA \$4C,X

$$A \oplus M \rightarrow A$$

- Modo de direccionamiento **Indexado**
- **Opcode:** \$A8 **Ciclo:**4 **Byte:**2
- Ejecuta la función lógica de XOR entre el contenido del acumulador "A" y el contenido de la localidad de memoria dada por la suma del operando con el registro X.
- Actualiza banderas N, Z, V=0



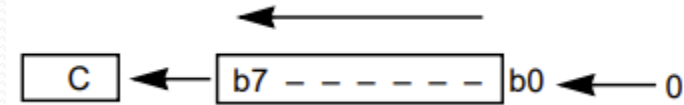
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

$$\oplus \begin{array}{r} 0011\ 0101 \\ 0000\ 0000 \\ \hline 0011\ 0101 \end{array}$$

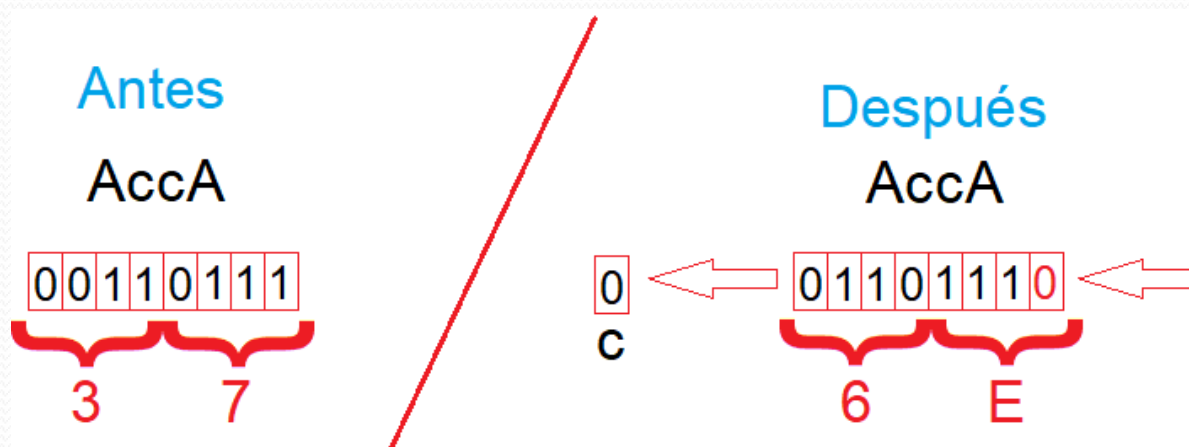
Set de instrucciones del MC68HC11

Instrucciones de rotación y desplazamiento

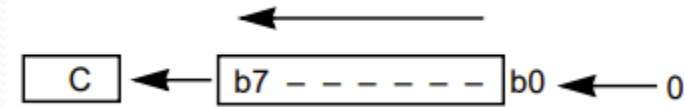
ASLA LSL



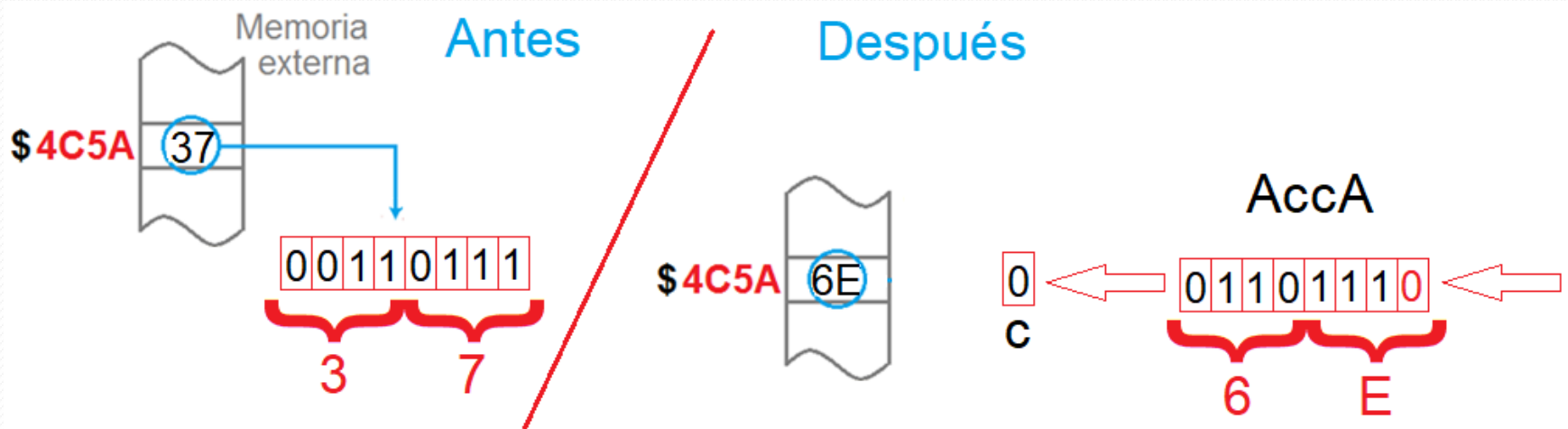
- Modo de direccionamiento **Inherente**
- **Opcode:** \$48 **Ciclo:**2 **Byte:**1
- Coloca un cero en el bit menos significativo del registro acumulador “A” y recorre todos sus bits originales hacia el lado izquierdo una posición.
- Actualiza banderas N, Z, V,C



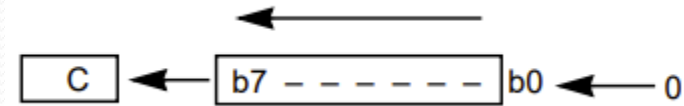
ASL \$4C5A LSL \$4C5A



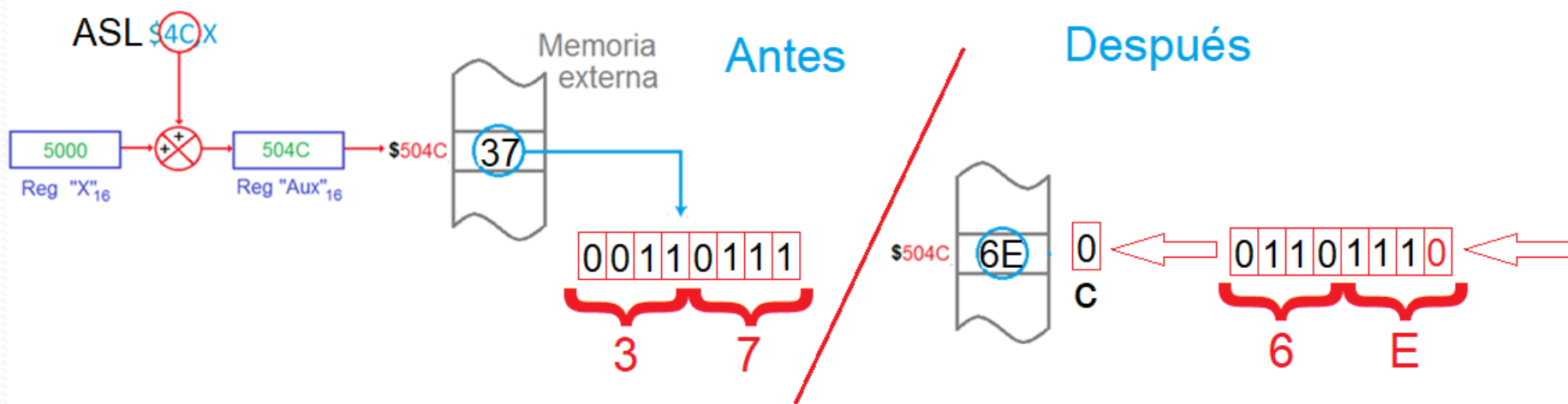
- Modo de direccionamiento **Extendido**
- **Opcode:** \$78 **Ciclo:**6 **Byte:**3
- Toma el contenido de la dirección dada por el operando de 8 bits y coloca un cero en el bit menos significativo y recorre todos sus bits originales hacia el lado izquierdo una posición.
- Actualiza banderas N, Z, V, C



ASL \$4C,X LSL \$4C,X

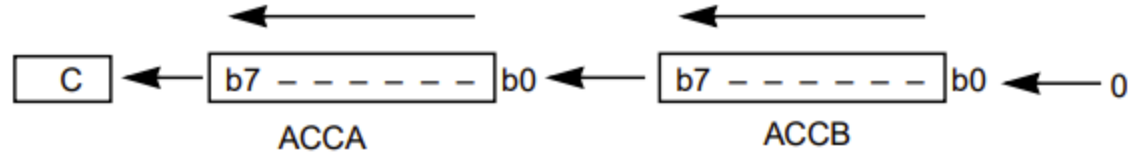


- Modo de direccionamiento **Indexado**
- **Opcode:** \$68 **Ciclo:**6 **Byte:**3
- Toma el contenido de la dirección dada por el operando de 8 bitsy el registro X, y coloca un cero en el bit menos significativo y recorre todos sus bits originales hacia el lado izquierdo una posición.
- Actualiza banderas N, Z, V, C

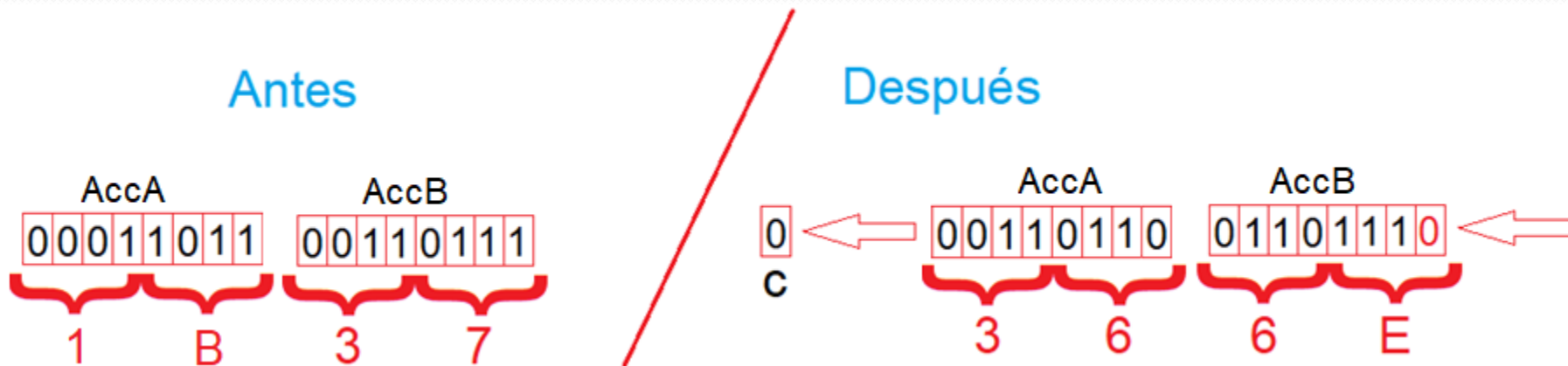


ASLD LSLD

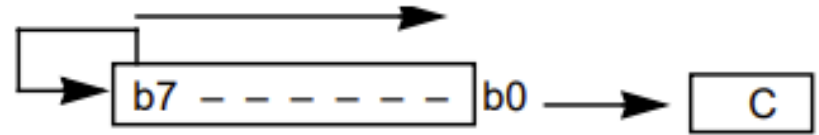
Operation:



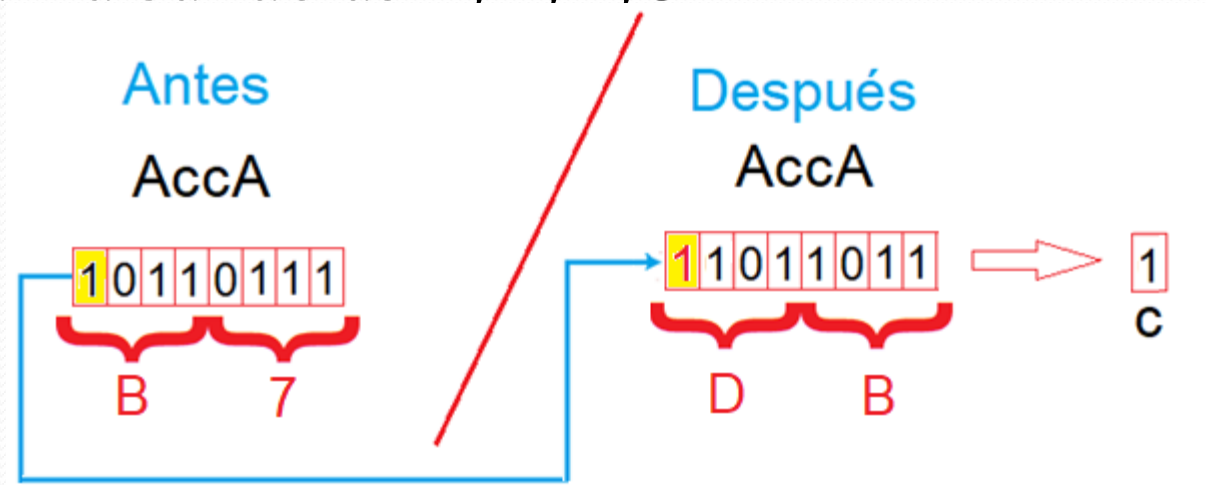
- Modo de direccionamiento **Inherente**
- **Opcode:** \$05 **Ciclo:**3 **Byte:**1
- Coloca un cero en el bit menos significativo del registro acumulador “D” y recorre todos sus bits originales hacia el lado izquierdo una posición.
- Actualiza banderas N, Z, V,C



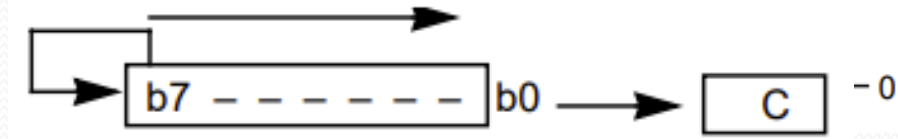
ASRA



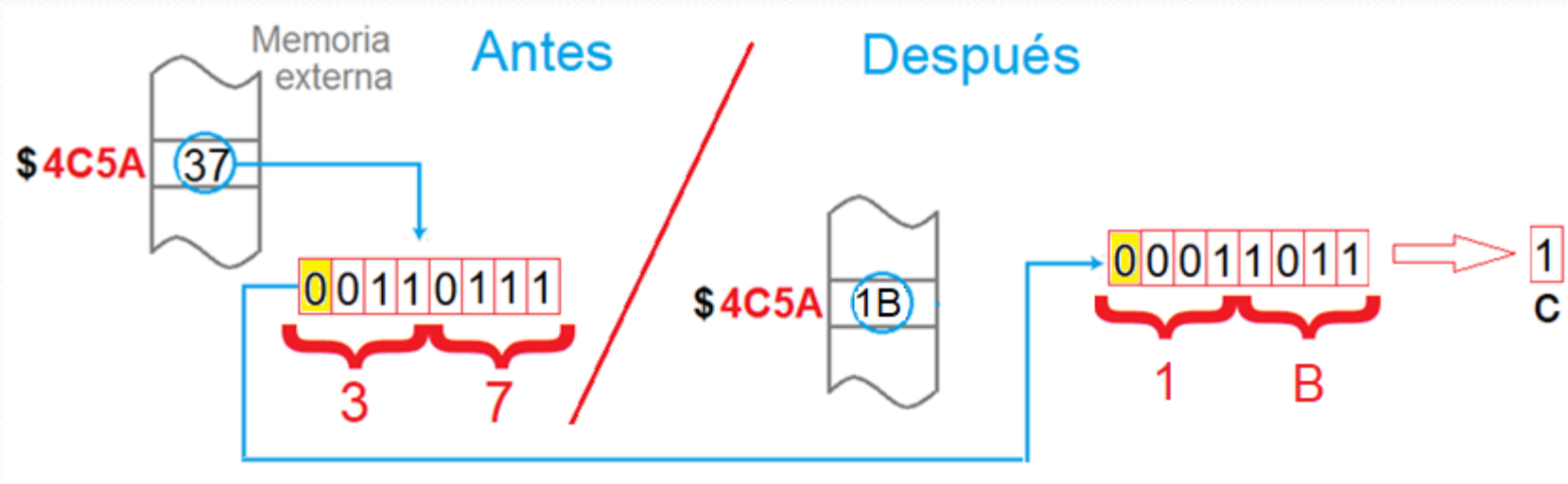
- Modo de direccionamiento **Inherente**
- **Opcode:** \$47 **Ciclo:**2 **Byte:**1
- Recicla el bit mas significativo del registro acumulador “A” y recorre todos sus bits originales hacia el lado derecho una posición.
- Actualiza banderas N, Z, V,C



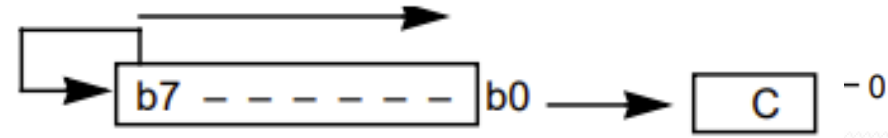
ASR\$4C5A



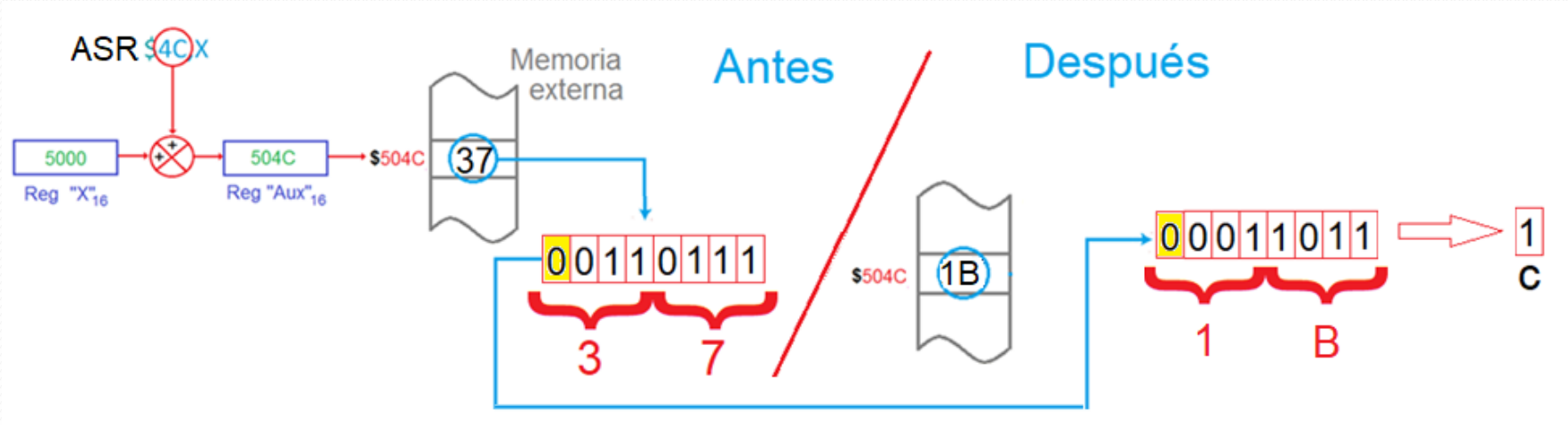
- Modo de direccionamiento **Extendido**
- **Opcode:** \$77 **Ciclo:**6 **Byte:**3
- Toma el contenido de la dirección dada por el operando de 8 bits y recicla el bit más significativo, recorriendo todos sus bits originales hacia el lado derecho una posición.
- Actualiza banderas N, Z, V, C



ASR\$4C,X



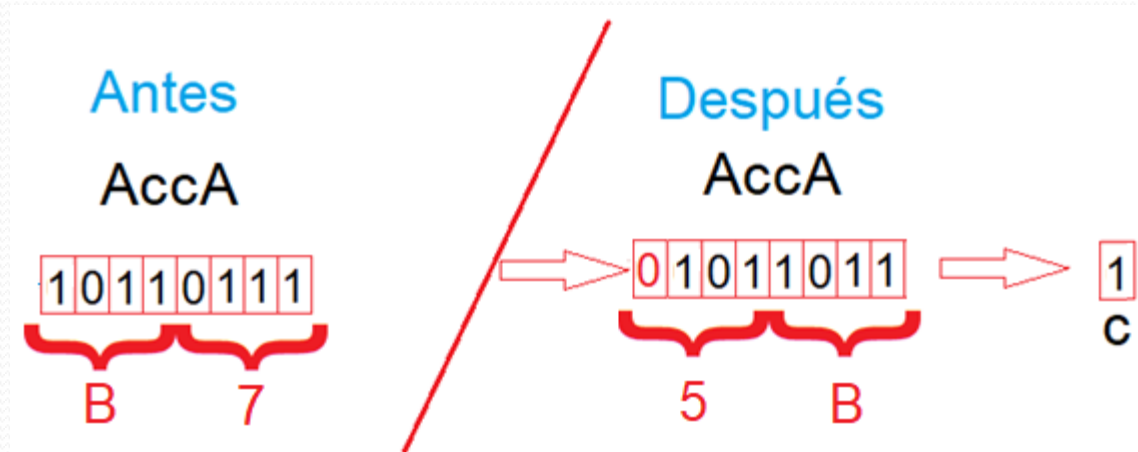
- Modo de direccionamiento **Indexado**
- **Opcode:** \$67 **Ciclo:**6 **Byte:**2
- Toma el contenido de la dirección dada por el operando de 8 bits y recicla el bit más significativo, recorriendo todos sus bits originales hacia el lado derecho una posición.
- Actualiza banderas N, Z, V, C



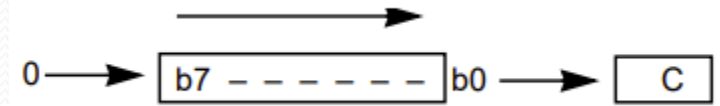
LSRA



- Modo de direccionamiento **Inherente**
- **Opcode:** \$44 **Ciclo:**2 **Byte:**1
- Inserta un cero en el bit mas significativo del registro acumulador “A” y recorre todos sus bits originales hacia el lado derecho una posición.
- Actualiza banderas N=0, Z, V,C



LSR\$4C5A



- Modo de direccionamiento **Extendido**
- **Opcode:** \$77 **Ciclo:**6 **Byte:**3
- Toma el contenido de la dirección dada por el operando de 8 bits e inserta un cero en el bit mas significativo , recorriendo todos sus bits originales hacia el lado derecho una posición.
- Actualiza banderas N=0, Z, V, C



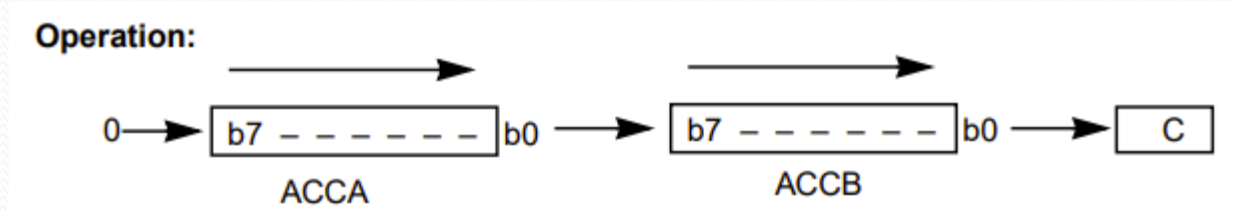
LSR\$4C,X



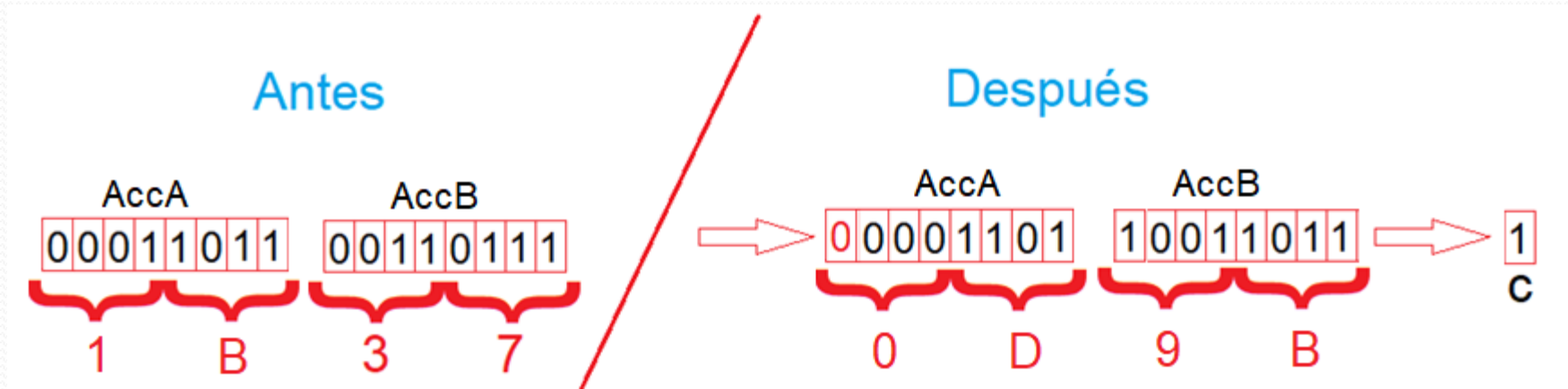
- Modo de direccionamiento **Indexado**
- **Opcode:** \$64 **Ciclo:**6 **Byte:**2
- Toma el contenido de la dirección dada por el operando de 8 bits e inserta un cero en el bit mas significativo , recorriendo todos sus bits originales hacia el lado derecho una posición.
- Actualiza banderas N, Z, V, C



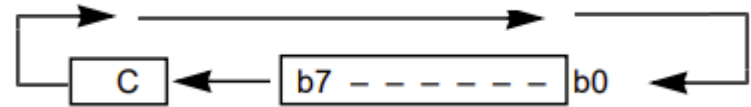
LSRD



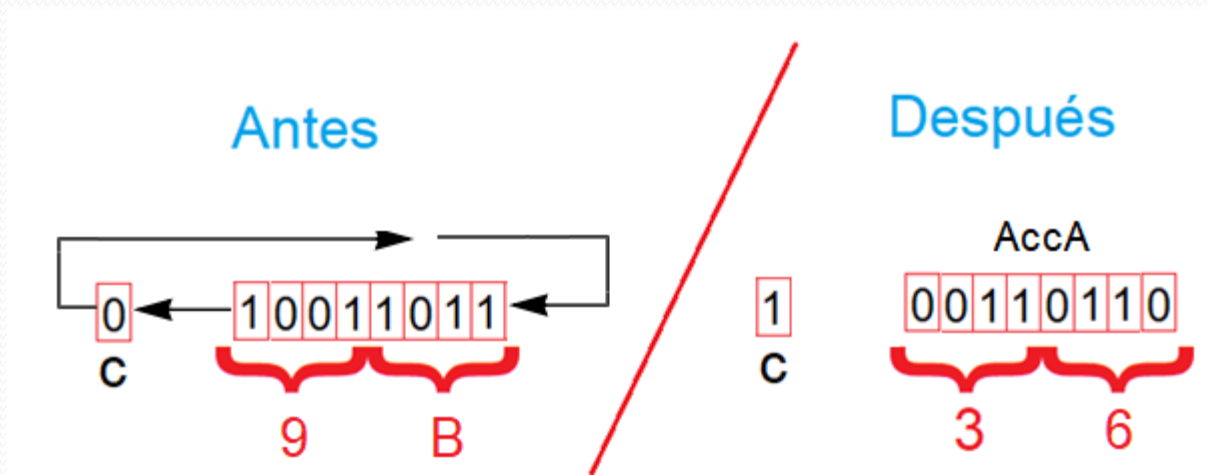
- Modo de direccionamiento **Inherente**
- **Opcode:** \$04 **Ciclo:**3 **Byte:**1
- Coloca un cero en el bit mas significativo del registro acumulador “D” y recorre todos sus bits originales hacia el lado derecho una posición.
- Actualiza banderas N=0, Z, V,C



ROLA



- Modo de direccionamiento **Inherente**
- **Opcode:** \$49 **Ciclo:**2 **Byte:**1
- Rota hacia la izquierda el contenido del registro acumulador “A” en serie con el bit de acarreo.
- Actualiza banderas N, Z, V,C



RORA



- Modo de direccionamiento **Inherente**
- **Opcode:** \$46 **Ciclo:**2 **Byte:**1
- Rota hacia la derecha el contenido del registro acumulador “A” en serie con el bit de acarreo.
- Actualiza banderas N, Z, V,C

