

PROGRAMACIÓN S-7

TIPOS DE DATOS BÁSICOS

TIPO	Tamaño (bits)	SIMBOLO	COMENTARIO
BOOL	1		{0,1}
INT	16		con signo
DINT	32	L	con signo
BYTE	8	B	sin signo
WORD	16	W	sin signo
DOUBLE WORD	32	DW	sin signo
REAL	32	notación número	con signo
DOUBLE WORD	32	DW	sin signo
S5TIME	16	S5T	aHbbMccSdddMS Tmax: 2H46M30S (9990s)
TIME	32	T	Hora
DATE_AND_TIME	64	DT	Fecha y hora
CHAR	8	'B'	alfanumérico
STRING		'ABCD'	

CONSTANTES / FORMATO

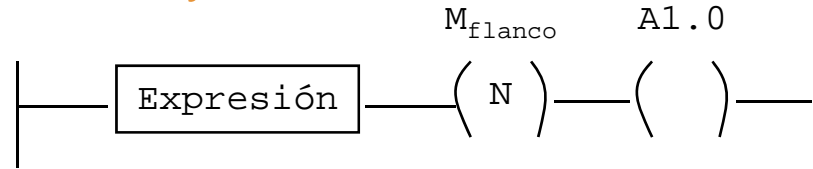
Número	Tamaño (bits)	Formato	Rango	Ejemplo
12	16		$\pm 2^{15}$	L 12
100000	32	L#	$\pm 2^{31}$	L L#10000
0,0053	32		$[\pm 1.17 \cdot 10^{-38}, \pm 3.4 \cdot 10^{-38}]$	L 5.3E-3
AF	8	B#16#	2^8	L B#16#AF
FF99	16	W#16#	2^{16}	L W#16#FF99
FF99AF99	32	DW#16#	2^{32}	L DW#16#FF99AF99
Binario	16	2#	2^{16}	L 2#0010101...
Binario	32	2#	2^{32}	L 2#100101000101...
5m3seg	16	S5T#	9990 seg.	L S5T# 5M3S
BCD	16	C#	0-999	L C#890

AWL-OPERADORES DE VARIABLES BINARIAS

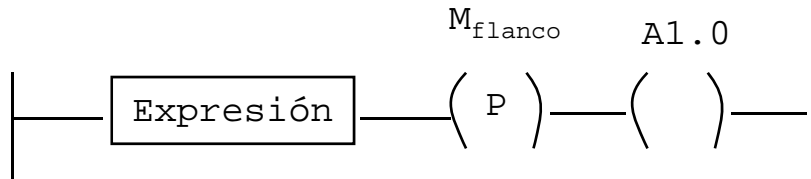
OPERADOR	MODIFICADOR	OPERANDO	COMENTARO
U, O, X	N	Variable lógica	1ª consulta
=		Booleano	Asignación
U, O, X	N, (Booleano	And, Or, Xor
)			Eval. op. Aplazada
S		Booleano	Pone a '1' si RLO=1
R		Booleano	Pone a '0' si RLO=1
NOT		RLO	
CLR		RLO	
SET		RLO	
SAVE		RB	Guarda RLO en RB
FP		Marca flanco	Flanco de subida
FN		Marca flanco	Flanco de bajada

BLOQUES OPERACIONES BINARIAS

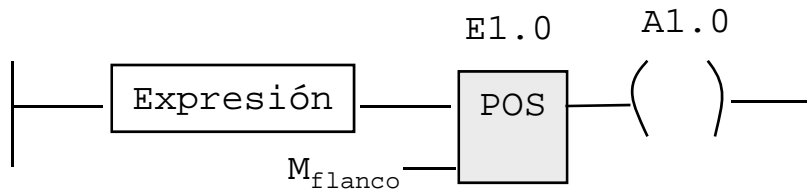
FLANCOS y CONECTORES



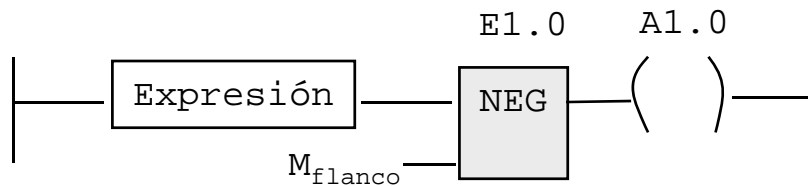
$$A1.0 = \text{Expresión} \downarrow$$



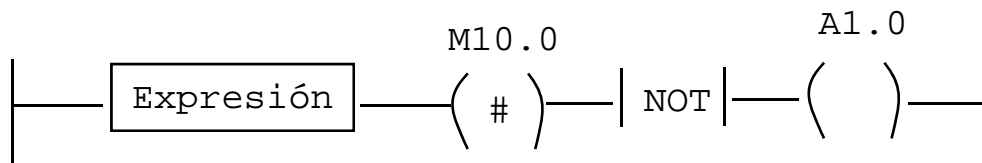
$$A1.0 = \text{Expresión} \uparrow$$



$$A1.0 = \text{Expresión} \cdot (E1.0 \uparrow)$$



$$A1.0 = \text{Expresión} \cdot (E1.0 \downarrow)$$



$$A1.0 = \overline{\text{Expresión}}$$

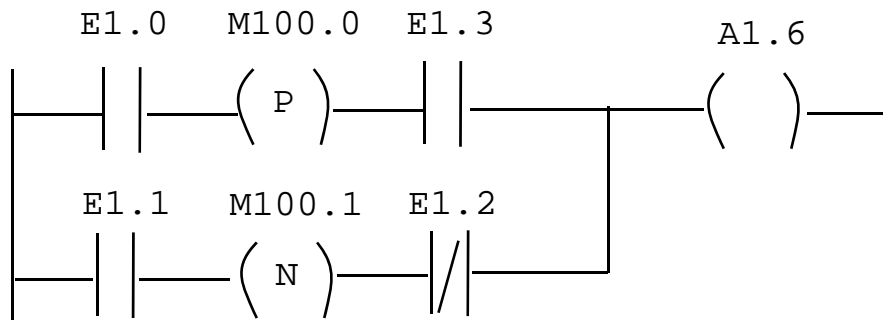
$$M10.0 = \text{Expresión}$$

Bloque conector

EJEMPLO

$$A1.6 = (E1.0 \uparrow \cdot E1.3) + (E1.1 \downarrow \cdot \overline{E1.2})$$

CONTACTOS (KOP)



AWL S7

```

U E1.0
FP M100.0
= M100.6
U M100.6
U E1.3
O(
U E1.1
FN M100.1
= M100.7
U M100.7
UN E1.2
)
= A1.6
    
```

AWL S7

```

U E1.0
FP M100.0
= M100.6
U E1.1
FN M100.1
= M100.7

U M100.6
U E1.3
O
U M100.7
UN E1.2
= A1.6
    
```

PALABRA DE ESTADO

RB	A1	A0	OV	OS	OR	STA	RLO	/ER
----	----	----	----	----	----	-----	-----	-----

BIT	Nombre	Significado
0	/ER	Primera consulta a nivel bajo
1	RLO	Resultado lógico
2	STA	Valor del bit direccionado
3	OR	Combinación $Y = 1$
4	OS	Overflow memorizado (<i>stored</i>)
5	OV	Overflow última operación
6	A0	Res. Comparación / Artimética / Op. lógicas / Op. Shift+Rot / Errores
7	A1	Res. Comparación / Artimética / Op. lógicas / Op. Shift+Rot / Errores
8	RB	R esultado B inario
9-15	Sin uso	

CONSULTA DE PALABRA DE ESTADO

OB1

U ==0
S M 1.0
U BIE
S M1.1

Consulta	Devuelve TRUE si se cumple
==0	A1=0 AND A0=0
<>0	A1 distinto de A0
>0	A1=1 AND A0=0
<0	A1=0 AND A0=1
>=0	==0 OR >0
<=0	==0 OR <0
UO	A1=1 AND A0=1 (Error en tiempo de ejecución: e.g. división por cero)
BIE	RB=1
OS	OS=1
OV	OV=1

ACTUALIZACIÓN PALABRA DE ESTADO

OB1

L 5
 L 0
 /I
 U UO
 = A1.0



RB	A1	A0	OV	OS	OR	STA	RLO	/ER
----	----	----	----	----	----	-----	-----	-----

0	0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---

0	0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---

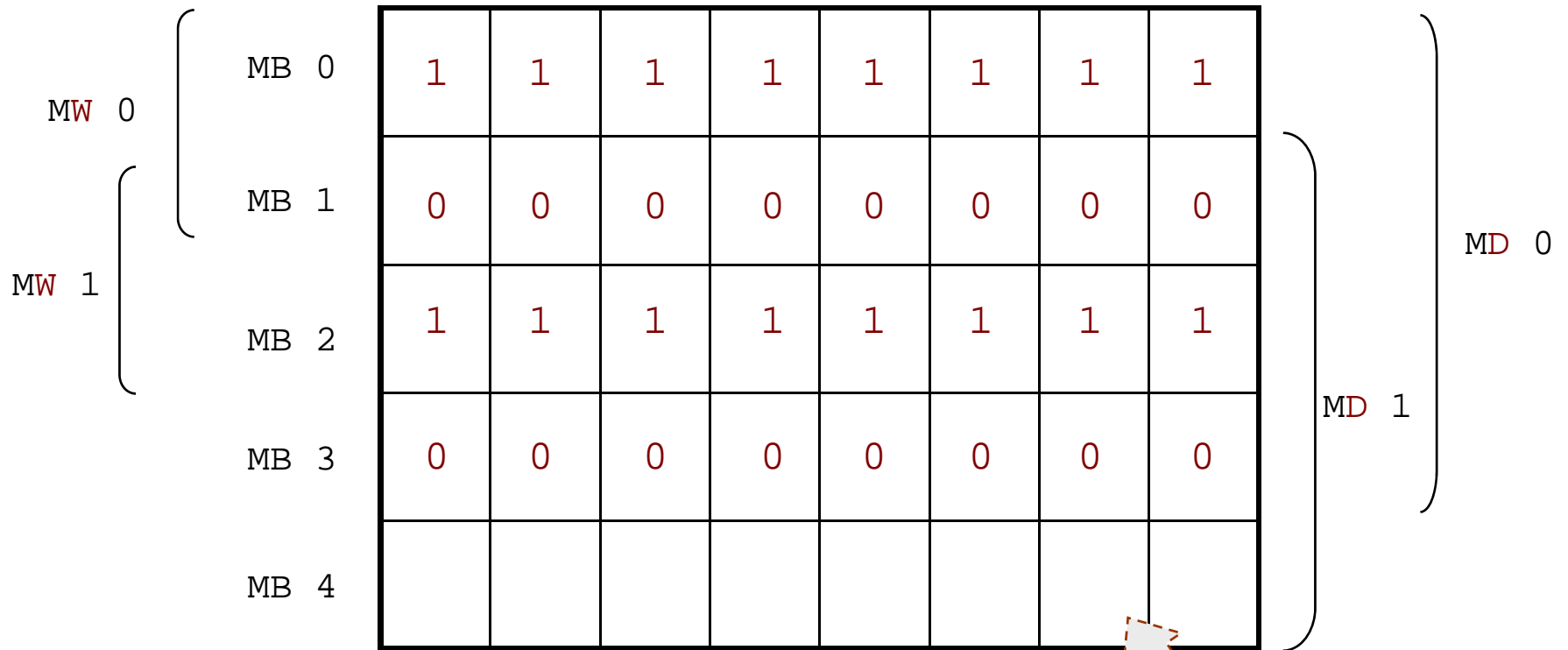
0	1	1	1	1	0	1	0	0
---	---	---	---	---	---	---	---	---

0	1	1	1	1	0	1	1	1
---	---	---	---	---	---	---	---	---

0	1	1	1	1	0	1	1	0
---	---	---	---	---	---	---	---	---

DIRECCIONAMIENTO DE MEMORIA: **BIG-ENDIAN**

MEMORIA DE USUARIO (E, M, A)



```

L DW#16#FF00FF00
T MD 0
    
```

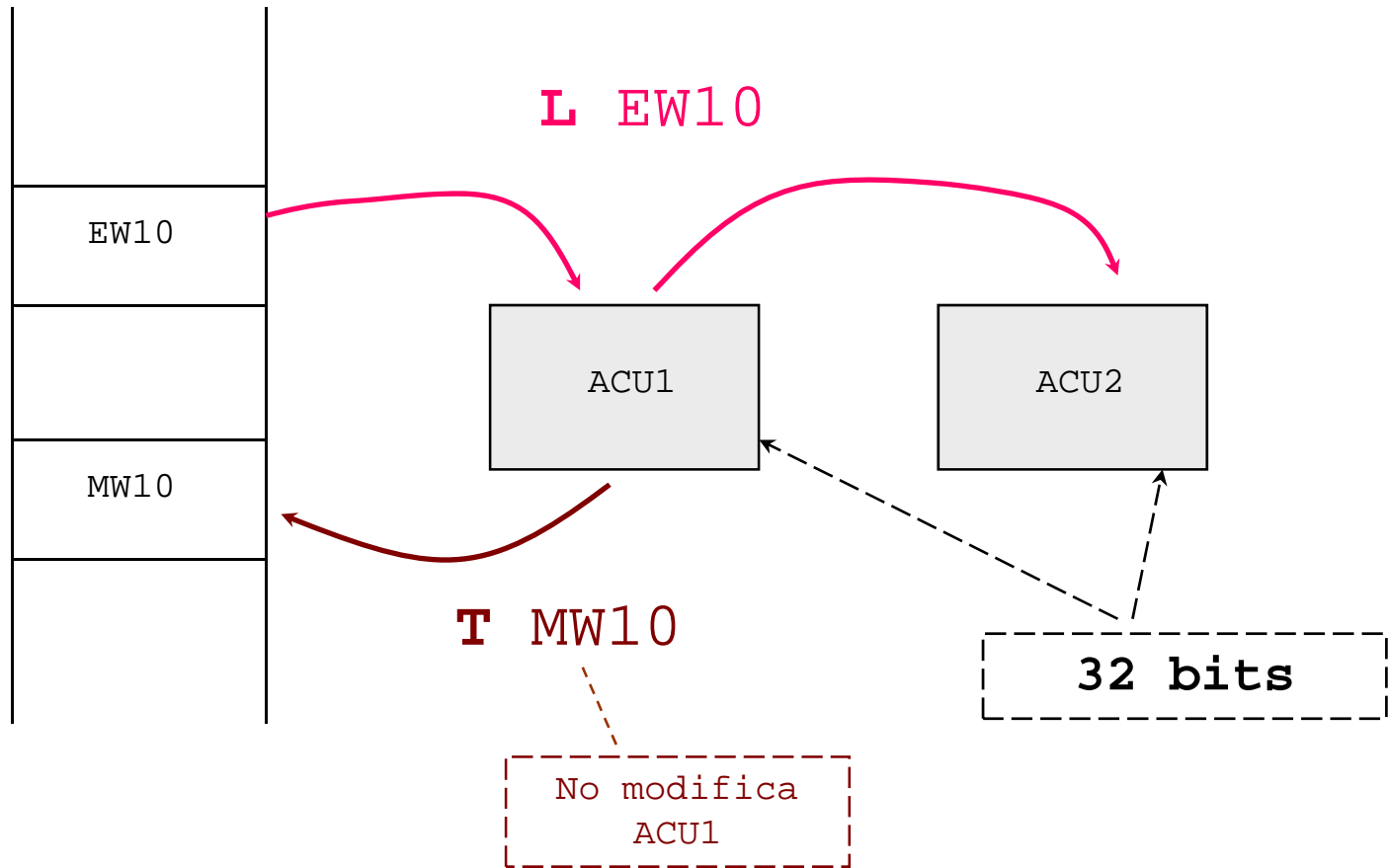
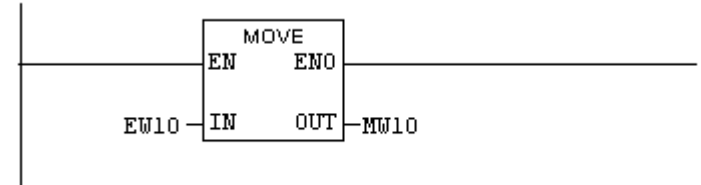
OPERADORES SOBRE VALORES ANALÓGICOS

- Operadores de carga (L) y transferencia (T)
- Operaciones Aritméticas (+,-,/,x)
- Operaciones de Comparación (>,<,<>,>=,...)
- Operaciones con conjuntos de variables lógicas (UW, OW, XOW...)
 - Máscaras de bits
- Otras
 - Operaciones de Desplazamiento y Rotación
 - Operaciones de conversión
 - Operaciones de redondeo etc.

CARGA Y TRANSFERENCIA

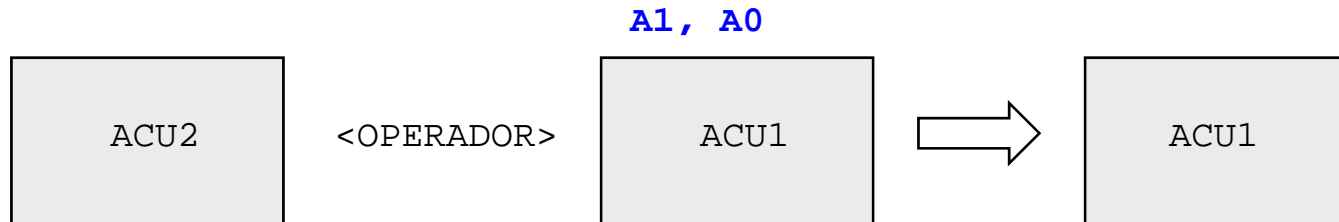
OPERADORES

- OPERADOR CARGA: **L**
- OPERADOR TRANSFERENCIA: **T**



OPERACIONES ARITMÉTICAS

DOS OPERANDOS IMPLÍCITOS



Operador	Tamaño(bits)	Comentario
(+, -, /, *) I	16	/I: Cociente ACU1-L, Resto ACU1-H
(+, -, /, *) D	32	/D: Cociente ACU1
MOD	32	Resto de división entera
(+, -, /, *) R	32	
+ <entero>	16 / 32	Único operando implícito ACU1

ARGUMENTOS IMPLÍCITOS EN LOS ACUMULADORES

¿OVERFLOW?

NO MODIFICAN EL BIT /ER

OPERACIONES ARITMÉTICAS

REGLAS GENERALES BITS DE ESTADO A1 y A0

- $A1=0$ Y $A0=0$: $ACU1 == 0$
- $A1=1$ Y $A0=0$ sin overflow : $ACU1 > 0$
- $A1=0$ Y $A0=1$ sin overflow : $ACU1 < 0$

VALOR **ACU1** EN
RELACIÓN CON 0

- $A1=1$ Y $OV=1$: Superado el valor máximo
- $A0=1$ Y $OV=1$: Minorado el valor mínimo
- $A1=1$ Y $A0=1$: División por cero,...

OVERFLOW

EJEMPLOS

OPERACIONES ARITMÉTICAS

SUMA CTE ENTERA (ACU1)

```
L 5
L 8
+13
```

ACU1=21, ACU2=5

A1=1, A0=0

MULTIPLICACIÓN 16 bits

```
L 1000
L 100
*I
```

ACU1=100000 A1=1,
A0=0, OV=1, OS=1

SUMA 16 bits

```
L -8
L 6
+I
```

ACU1=-2

A1=0, A0=1

MULTIPLICACIÓN 16 bits

```
L 1000
L 100
*I
L 200
L 5
*I
```

ACU1=100000

A1=1, A0=0, OV=1, OS=1

ACU1=1000

A1=1, A0=0, OV=0, OS=1

DIVISIÓN 16 bits

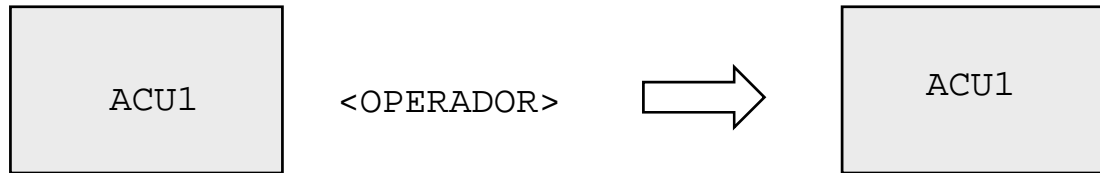
```
L 2
L 6
/I
```

ACU1=0x20000

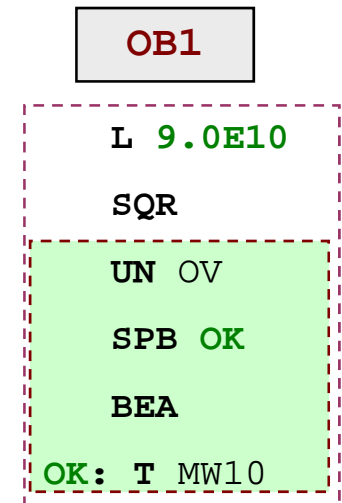
A1=1, A0=0

OPERACIONES ARITMÉTICAS

UN OPERANDO IMPLÍCITO

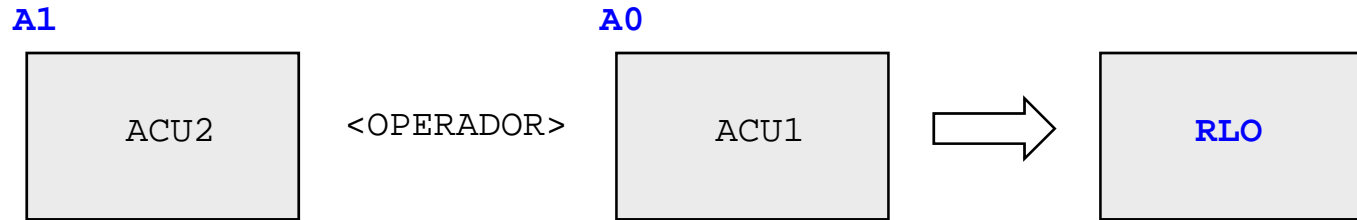


ABS SQR SQRT EXP
LN SIN COS TAN
ASIN ACOS ATAN...



Números reales (coma flotante, 32 bits)

OPERACIONES DE COMPARACIÓN



Operador	Tamaño(bits)	Comentario
(==,<>,>,<,>=,<=) I	16	ACU-1 y ACU-2 no varían
(==,<>,>,<,>=,<=) D	32	
(==,<>,>,<,>=,<=) R	32	

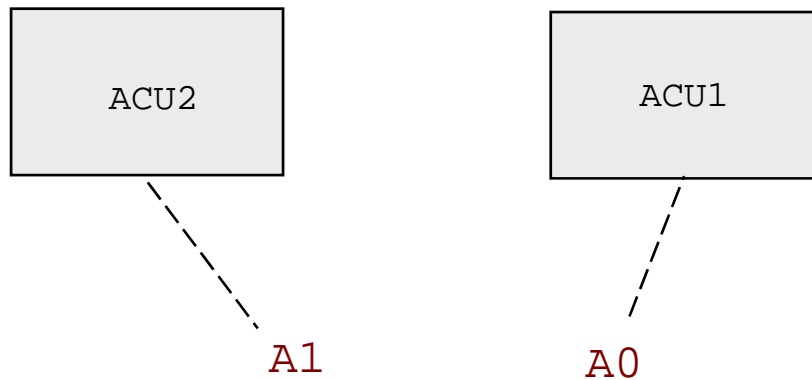
Los bits A1 y A0 mantienen su significado intuitivo respecto a los valores de ACU2 y ACU1

NO MODIFICAN EL BIT 'ER'

EJEMPLOS

OPERACIONES COMPARACIÓN

Bits A1 y A0 de la palabra de estado



Los valores de A1 y A0 no dependen de la operación de comparación en concreto, sino del tamaño relativo del valor del ACU2 y del ACU1 entre sí.

```
L 5
L 8
>=I
```

ACU1=8, ACU2=5

A1=0, A0=1, RLO=0

```
L 5
L 8
<=I
```

ACU1=8, ACU2=5

A1=0, A0=1, RLO=1

```
L 9.7E+6
L 2
<I
```

ACU1=2, ACU2=9.7E+6

A1=1, A0=0, OV=0, OS=0

EJEMPLOS (II)

OPERACIONES COMPARACIÓN

```

L 5
L 8
<I
= A 1.0

```

A1.0 = 1
(5 < 8)

```

L 5
L 8
U >=0
= A 1.1

```

A1.1 = 1
(A1=0, A0=0)

```

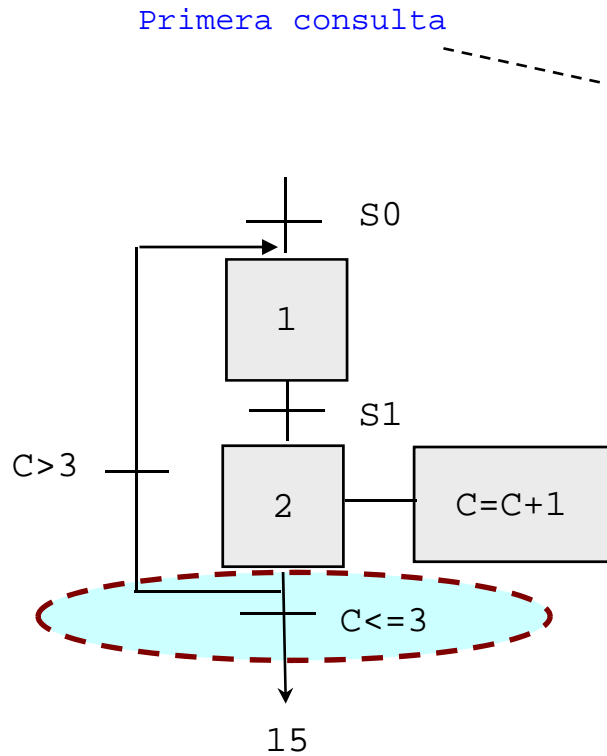
L EW0
L EW2
<I
= A 1.0
U ==0
= A1.1

```

A1.0 = 1 si EW0 < EW2 (con signo)

A1.1 = 1 si EW0 == EW2

EJEMPLOS (III)



OB1

```
//...
U E2
ZV Z1
// Transición
L Z1
L 3
>I
SPB _MA
U E2
S E15
R E2
SPA FIN
_MA: NOP 0
U E2
S E1
FIN: NOP 0
//...
```

OB1

```
//...
U E2
ZV Z1
// Transición
L Z1
L 3
>I
S E1
R E2
U <=0
S E15
R E2
//...
```

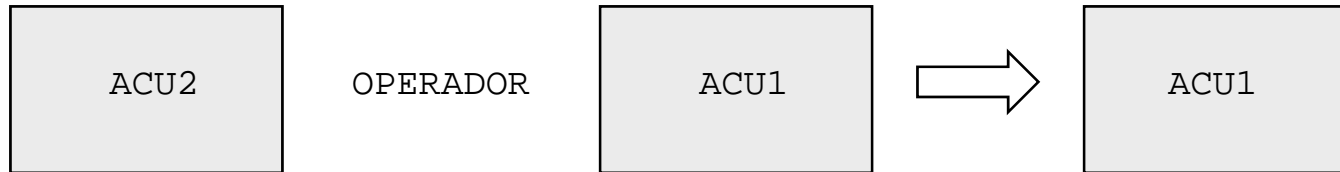
CONSULTA DE PALABRA DE ESTADO

Consulta	TRUE (RLO = 1)	Tras Op. Aritmética	Tras Op. Comparación
==0	A1=0 AND A0=0	ACU1=0	ACU2 == ACU1
<>0	A1 distinto de A0	ACU1<>0	ACU2 <> ACU1
>0	A1=1 AND A0=0	ACU1>0	ACU2 > ACU1
<0	A1=0 AND A0=1	ACU1<0	ACU2 < ACU1
>=0	==0 OR >0	ACU1>=0	ACU2 >= ACU1
<=0	==0 OR <0	ACU1<=0	ACU2 <= ACU1
UO	A1=1 AND A0=1 (i.e. División por cero)		
BIE	RB=1		
OS	OS=1		
OV	OV=1		

A1, A0 -> ACU1

A0 -> ACU1, A1 -> ACU2

OPERACIONES CON CONJUNTOS DE V. LÓGICAS



Operador	Tamaño(bits)	Comentario
UW	16	Y lógica entre acumuladores (L)
OW	16	O lógica entre acumuladores (L)
XOW	16	XOR lógica entre acumuladores (L)
UD	32	Y lógica entre acumuladores
OD	32	O lógica entre acumuladores
XOD	32	XOR lógica entre acumuladores

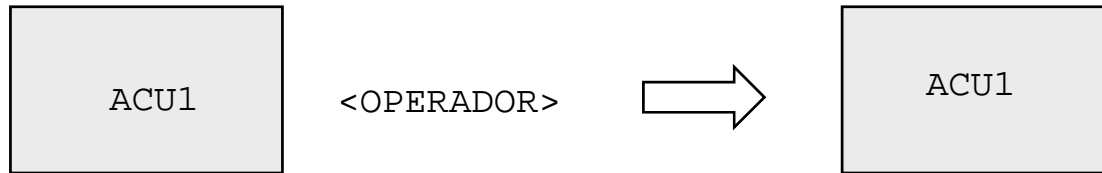
OB1

- L MW10
- L MW12
- UW
- T MW14

SIN ARGUMENTOS EXPLÍCITOS

CONVERSIÓN DE TIPOS

UN OPERANDO IMPLÍCITO



OB1

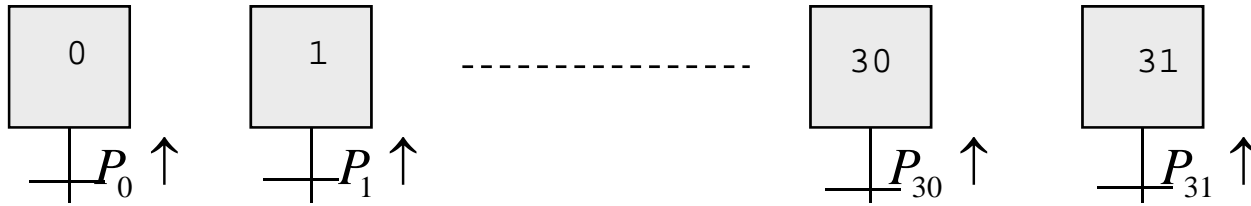
BTI ITD DTB BTD DTR ITD

RND RND+ RND- TRUNC

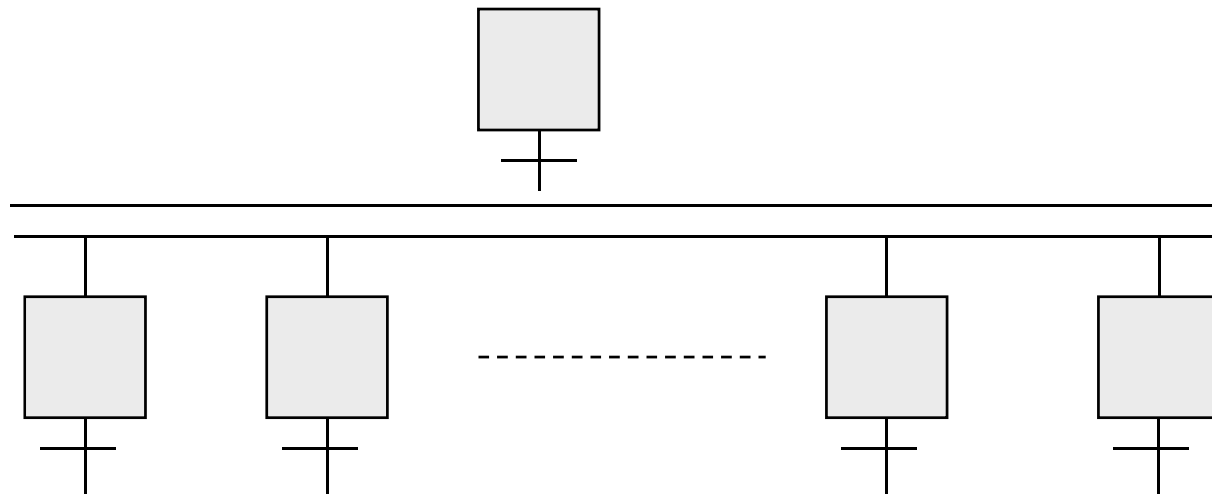
L 9
 T MW10
 L MW10
 ITD
 T MD 20
 DTR
 T MD 30
 SQRT
 T MD40
 TRUNC
 T MW 50

EJEMPLOS-ESCALABILIDAD

1-DETECCIÓN SIMULTÁNEA DE FLANCOS



2-ACTIV./DESACTIV. SIMULTÁNEA DE ETAPAS



3-ARRANQUE EN FRIO

EJEMPLOS-ESCALABILIDAD

1-FLANCO DE 32 VARIABLES

ED10	Entrada (P _i)
MD100	Flancos
MD10	Pulsos

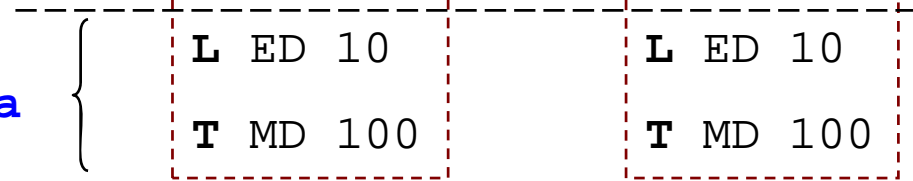
Flanco positivo

Flanco negativo

L MD 100
 L ED 10
 XOD
 L ED 10
 UD
 T MD 10
 L ED 10
 T MD 100

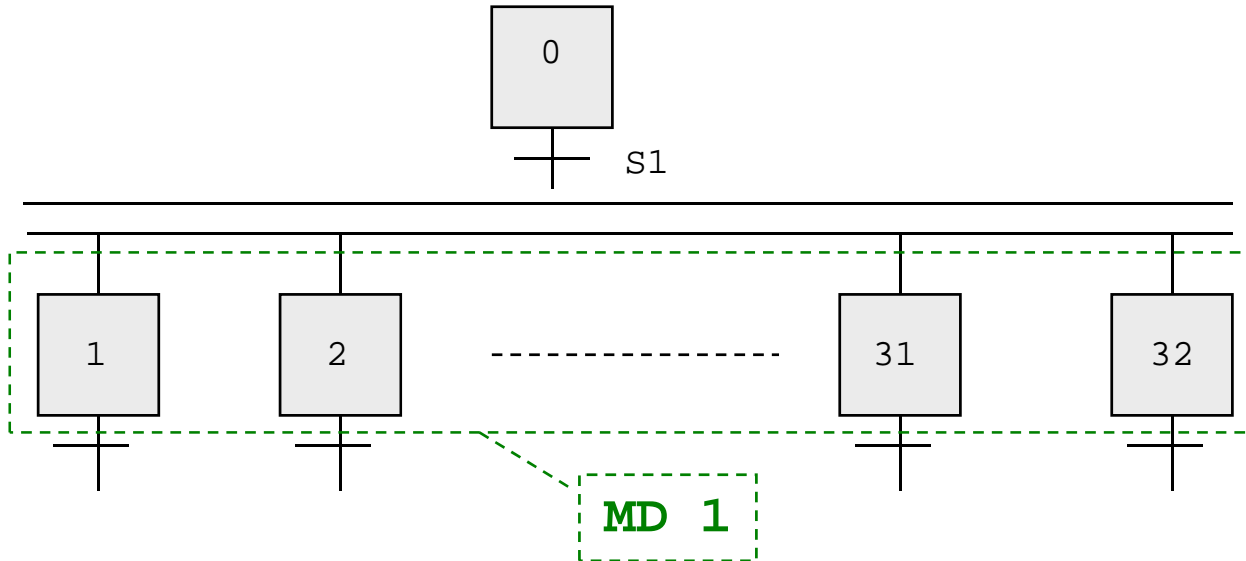
L MD 100
 L ED 10
 XOD
 L MD 100
 UD
 T MD 10
 L ED 10
 T MD 100

Memoria



EJEMPLOS-ESCALABILIDAD

2-ACTIVACIÓN SIMULTÁNEA DE ETAPAS



Ecuación de activación

```


U E0
U S1 E0
L DW#16#FFFFFFFF
T MD1 NOAC
R E0


```

Activación

```

L DW#16#FFFFFFFF
T MD 1
R E0
NOAC: NOP 0

```

DESACTIVACIÓN SIMULTÁNEA DE ETAPAS

```

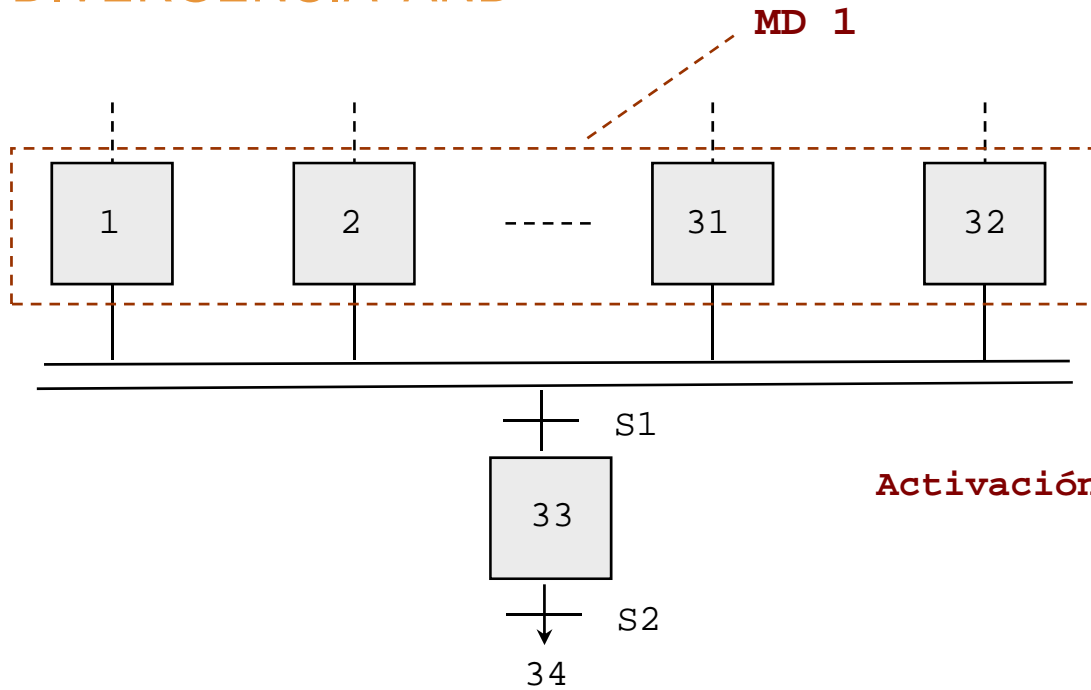
//Desactiva 32 etapas
L DW#16#0
T ETAPAS_CODIFICADAS_EN BITS_CONSECUTIVOS

```

EJEMPLOS-ESCALABILIDAD

Desactivación simultánea

DIVERGENCIA 'AND'



Transición a E33

```

L MD 1
L DW#16#FFFFFFFF
==D
SPBN NOAC
U S1
= E33
L DW#16#0
T MD1
NOAC: NOP 0

//Retención
U E33
UN E34
= E33
//...
    
```

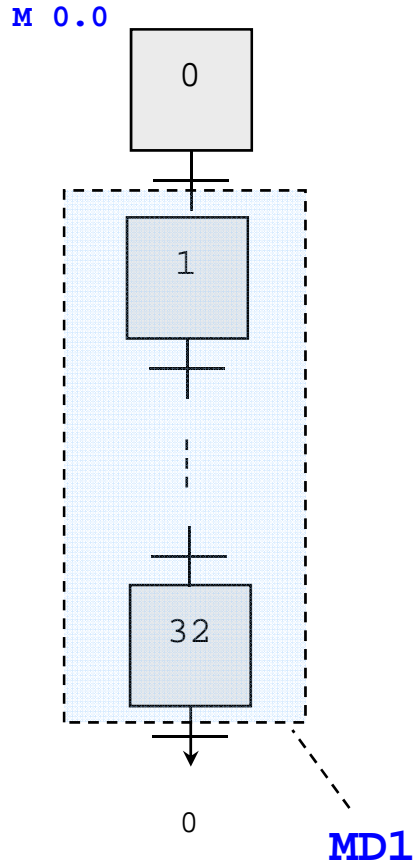
Activación

Retención

EMPLEAR ECUACIÓN DE RETENCIÓN
EN LA CODIFICACIÓN

EJEMPLOS-ESCALABILIDAD

3-ARRANQUE / REARME



OB100

```

UN M1.0
UN M1.1
UN M1.2
//...
S M0.0
R Act.1
R Act.2
//...
R Act.N
    
```

EN FRIO

OB100

```

SET
S M0.0
R Act.1
R Act.2
//...
R Act.N
    
```

OB100

```

SET
S M0.0
L DW#16#0
T MD1
    
```

EN CALIENTE

OB102

```

L MD 1
L DW#16#0
==D
S M0.0
R Act.1
R Act.2
//...
R Act.N
    
```

OB102

```

L MD 1
L DW#16#0
==D
SPBN _001

S M0.0
L DW#16#0
L AD1

_001:NOP 0
//...
    
```

INSTRUCCIONES DE CONTROL

INDICE

- **SALTO DENTRO DE UN BLOQUE**
 - INCONDICIONAL
 - CONDICIONAL
- **CONTROL DE BLOQUES**
 - CALL, UC, CC, BEA, BEB
- **CONTROL DE EJECUCIÓN DE GRUPO DE INSTRUCCIONES (MCR)**
 - SEGURIDAD

SALTO DENTRO DEL BLOQUE

CONSIDERACIONES GENERALES

- Interrupciones sin salirse del bloque
- Se puede saltar hacia adelante o hacia atrás
- Pueden modificar el bit que condiciona el salto
 - RLO, RB, OS, OV
- Pueden modificar el bit de primera consulta
 - ER=0
- Todas las etiquetas tienen un máximo de 4 caracteres alfanuméricos y un primer carácter obligatoriamente no numérico
 - **SPA E001**
 - ~~**SPA 0E01**~~
 - **SPA _001**

SALTO DENTRO DEL BLOQUE (II)

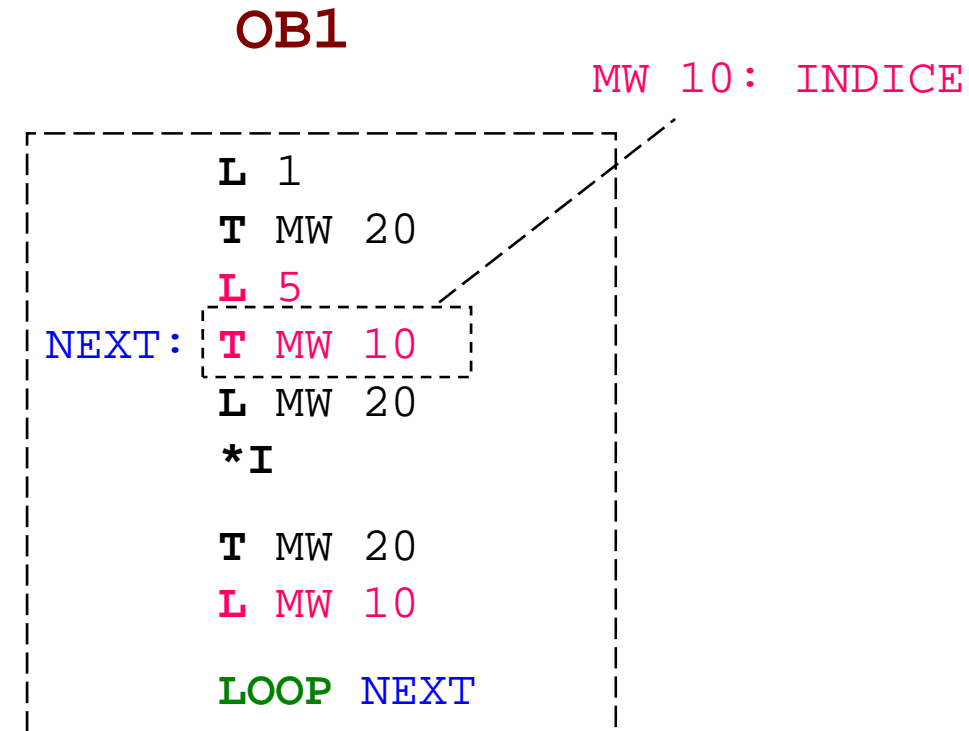
TIPOS

- **INCONDICIONAL**
 - **SPA** : Salto incondicional (absoluto)
 - **LOOP** : Bucle FOR
 - **SPL** : Switch-case
- **CONDICIONAL**
 - Operaciones lógicas / comparación (RLO)
 - Operaciones aritméticas (A1, A0)
 - Errores (OV / OS / A1=1y A0=1)

SALTO INCONDICIONAL - LOOP

LOOP

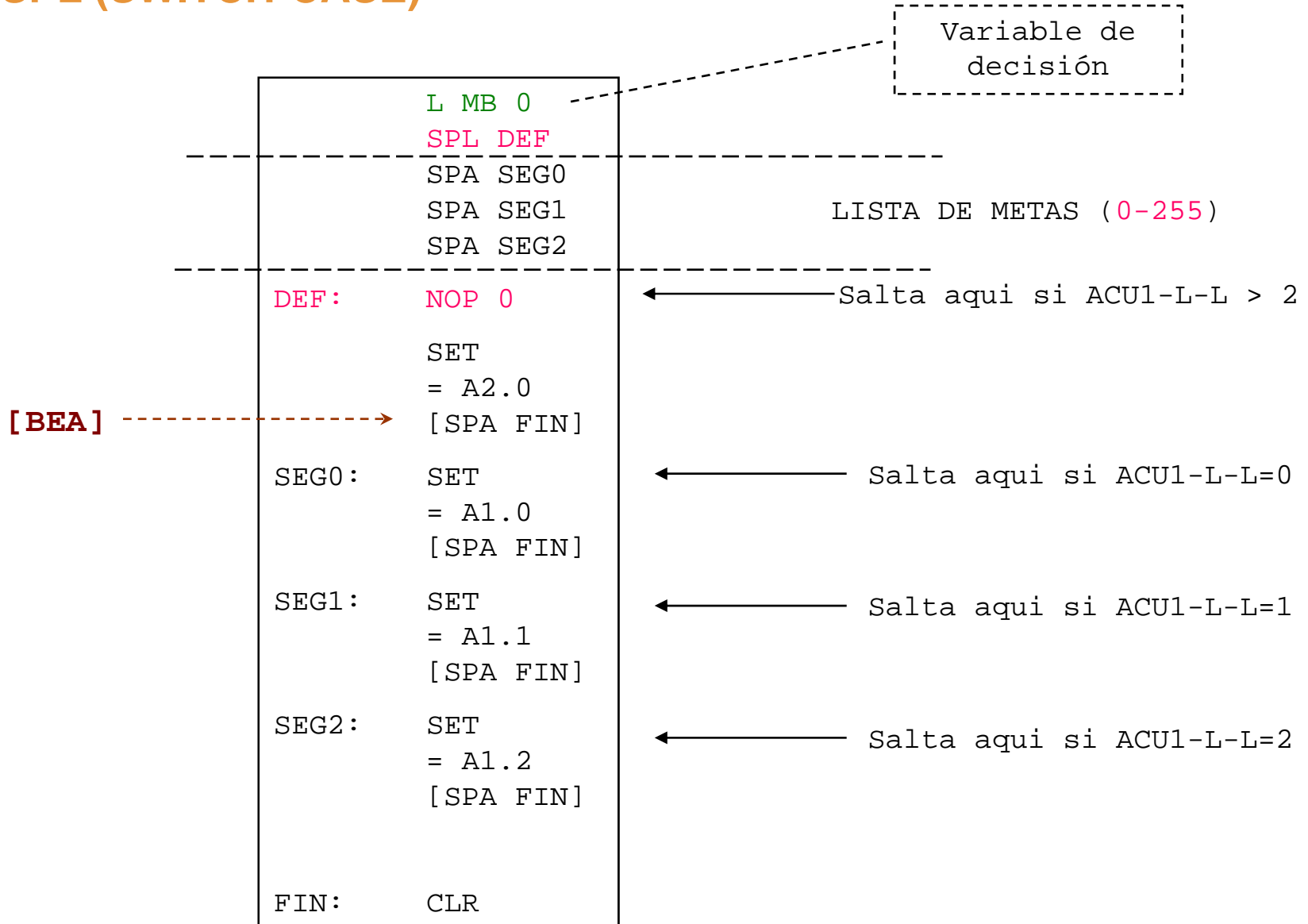
Decrementa el contenido del ACU1 y salta a la meta NEXT, si el $ACU1-L > 0$



MW 20 = ? 120

SALTO INCONDICIONAL - SPL

SPL (SWITCH-CASE)



SALTO CONDICIONAL

SABER

Operador	Condición	Nueva palabra estado
SPB	RLO=1	RLO =1 / ER=0
SPBN	RLO=0	RLO =1 / ER=0
SPBB	RLO=1	RLO =1 / RB=X / ER=0
SPBNB	RLO=0	RLO =1 / RB=X / ER=0
SPBI	RB=1	ER=0
SPBIN	RB=0	ER=0

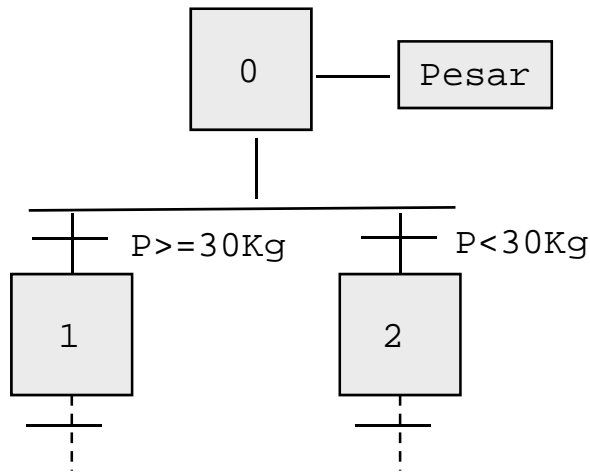
SIEMPRE PRIMERA CONSULTA

NO CONCATENAR

Copia del RLO en el momento de la ejecución de la instrucción

EJEMPLOS

SPB: Switch-Case



EW10	Peso (Kg)
M1.0	Etapa 0
M1.1	Etapa 1
M1.2	Etapa 2

IMPLEMENTACION-AWL

Divergencia
OR

```

L EW10
L 30
>=I
= Maux
//Act Etapa1
S M1.1
R M1.0
//Act Etapa2
U M1.0
UN Maux
S M1.2
R M1.0
    
```

Divergencia
OR

```

L EW10
L 30
>=I
SPB _MAY
//Act. Etapa2
S M1.2
R M1.0
SPA CONT
//Act. Etapa1
_MAY: S M1.1
R M1.0
_CONT: NOP 0
//...
    
```

EJEMPLO (II)

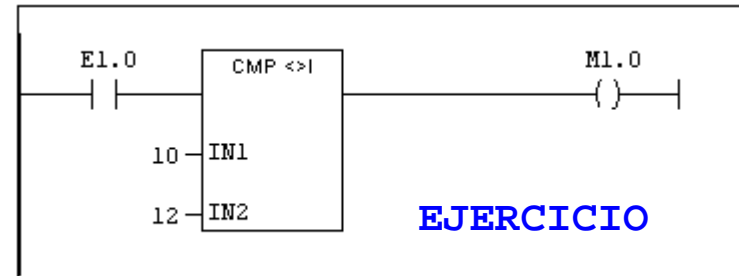
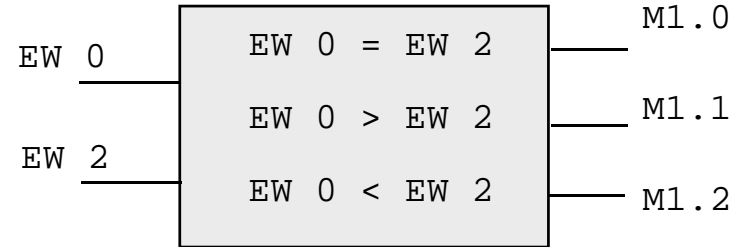
SPB: Switch-Case

```

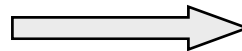
L EW 0
L EW 2
==I
SPB _IGU
U >0
SPB _MAY
U <0
SPB _MEN

_IГУ: = M1.0
SPA FIN
_MAY: = M1.1
SPA FIN
_MEN: = M1.2
FIN: NOP 0
    
```

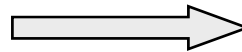
BLOQUE COMPARADOR



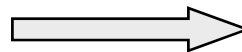
EJERCICIO



EW0 = EW2



EW0 > EW2



EW0 < EW2

```

U E1.0
U(
L 10
L 12
<>I
)
= M1.0
    
```

SALTO CONDICIONAL

DEPENDENCIA BITS: A1,A0 (Op. Aritméticas / Comparación)

Operador (Id. asociado)	Resultado Op. Aritmética	Resultado Op. Comparación
SPZ (=0)	ACU1=0	ACU2 = ACU1
SPN (<>0)	ACU1<>0	ACU2 <> ACU1
SPP (>0)	ACU1>0	ACU2 > ACU1
SPPZ (>=0)	ACU1>=0	ACU2 >= ACU1
SPM (<0)	ACU1<0	ACU2 < ACU1
SPMZ (<=0)	ACU1<=0	ACU2 <= ACU1

NO MODIFICAN LOS BITS DE LA PALABRA DE ESTADO

EJEMPLO (III)

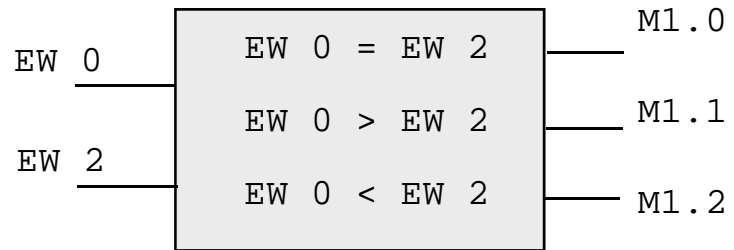
SPB: Switch-Case

```

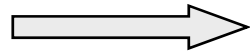
L EW 0
L EW 2
==I
SPB _IGU
SPP _MAY
SPM _MEN

_IGU: = M1.0
SPA FIN
_MAY: = M1.1
SPA FIN
_MEN: = M1.2
FIN: NOP 0
    
```

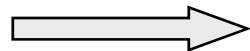
BLOQUE COMPARADOR



EW0 = EW2



EW0 > EW2



EW0 < EW2

SALTO CONDICIONAL

GESTIÓN ERRORES

Operador	Resultado Op. Artimética	Palabra de estado
SPO	Salta is OV=1	x
SPS	Salta si OS=1	OS = 0
SPU	Salta si el resultado no es válido (i.e. 3/0)	x

**NO MODIFICAN LOS BITS DE LA PALABRA DE ESTADO
(Excepto OS(SPS))**

¿Como borrar Bit OS?

```

SPS SIG
SIG:  NOP 0
//...
```


EJEMPLOS

GESTIÓN DE ERRORES

FC-1

```

SGTE:      SPS SGTE          //Borra bit OS inicio bloque
             L MW 12
             L MW 14
             *I
             L MW 16
             /I

             SPS OVER        //Overflow (Borra Bit OS)
             SPP _>0         //Resultado >0
             SPA _COM

OVER:     NOP 0
             ...             //Tratamiento Overflow
             SPA _COM

_>0:      NOP 0
             ...             //Código relativo (>0)

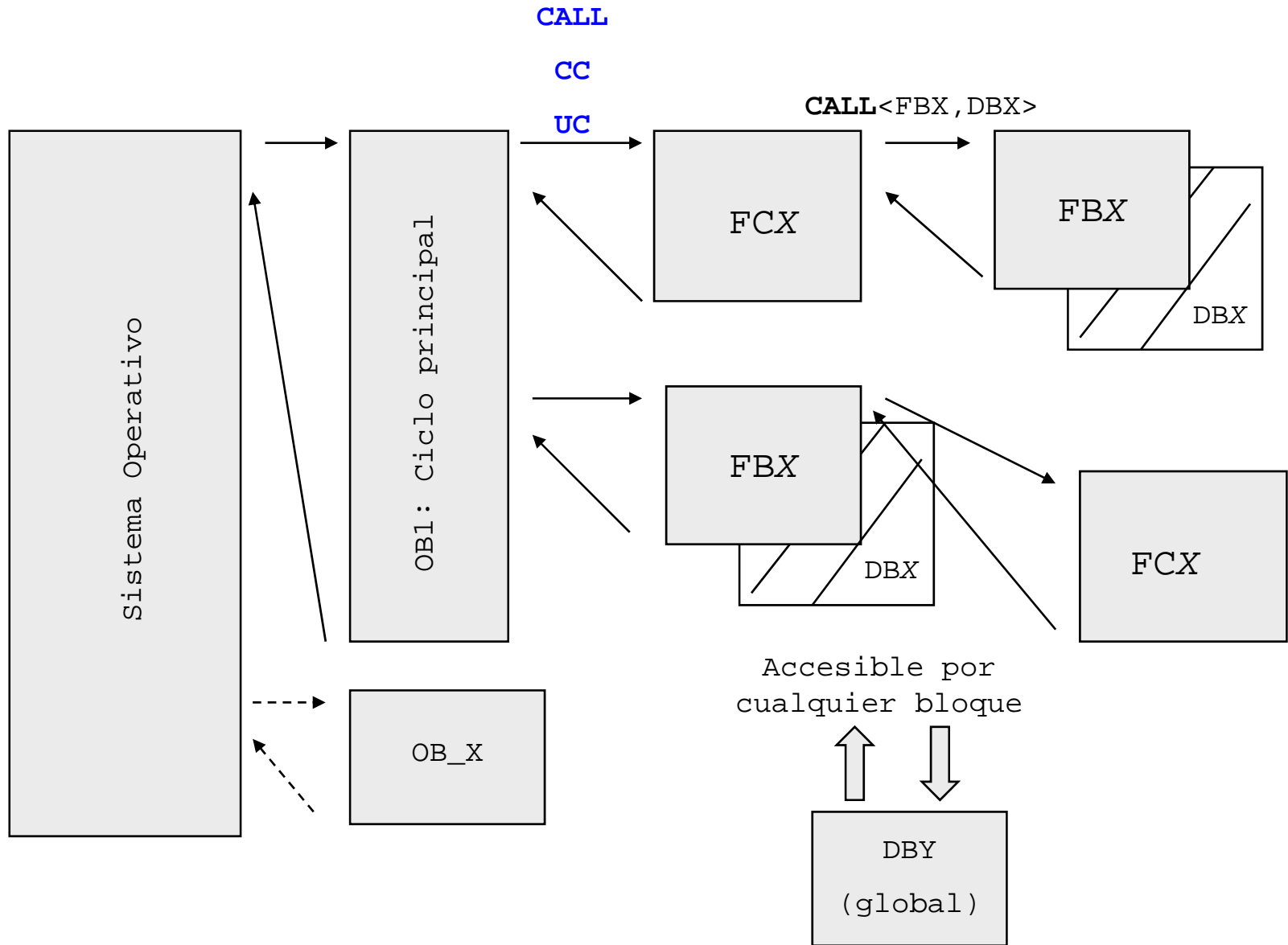
_COM:     NOP 0
             ...             //Programa común

```

RESUMEN SALTOS CONDICIONALES EN EL BLOQUE

Operador	Palabra Estado	Operador (Id. asociado)	Palabra Estado
SPB	RLO =1 / ER=0	SPZ (=0)	x
SPBN	RLO =1 / ER=0	SPN (<>0)	x
SPBB	RLO =1 / RB=X / ER=0	SPP (>0)	x
SPBNB	RLO =1 / RB=X / ER=0	SPPZ (>=0)	x
SPBI	x	SPM (<0)	x
SPBIN	x	SPMZ (<=0)	x
SPO	x	SPU	x
SPS	OS=0		

ESQUEMA DE LLAMADAS ENTRE BLOQUES



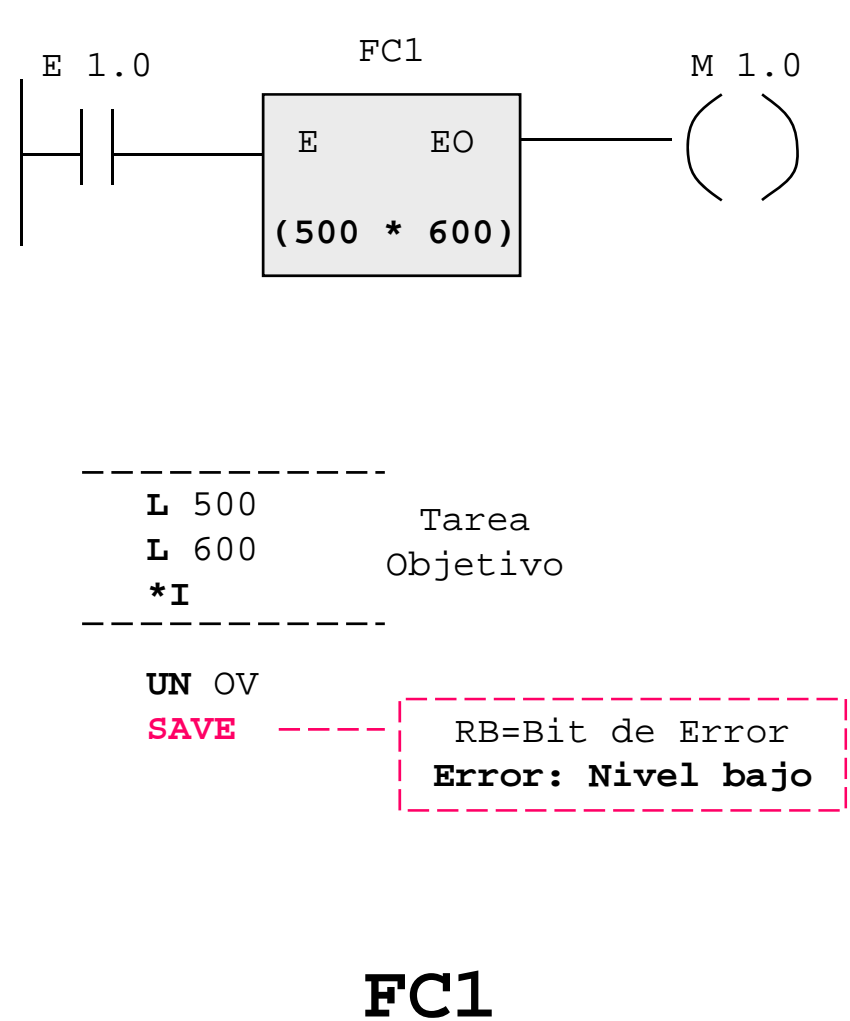
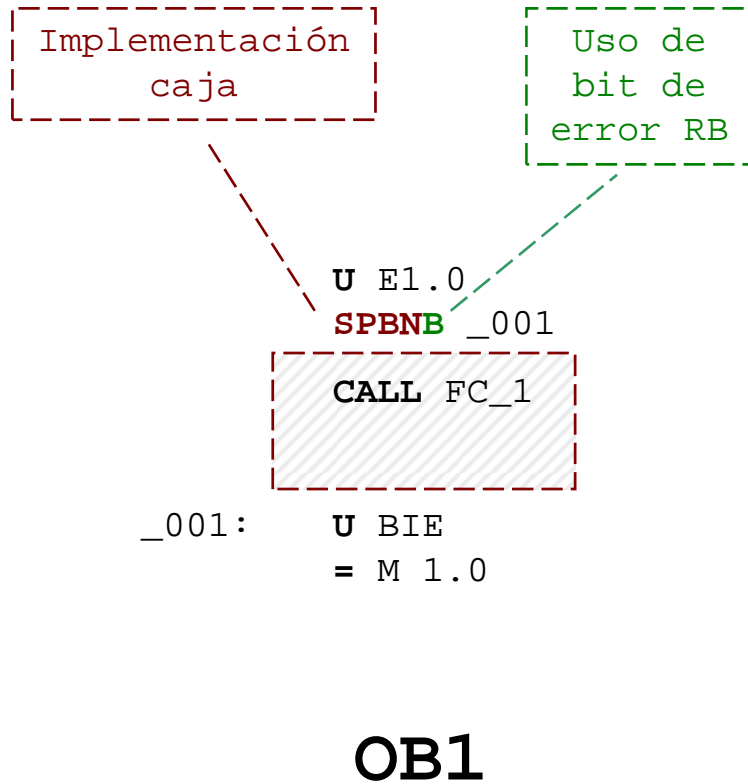
INSTRUCCIONES DE CONTROL ENTRE BLOQUES

**Primera consulta siempre
(ER=0)**

Operador	Descripción	Observaciones
CALL <ID_BLOQUE> [, <ID_BLOQUE_DATOS>]	LLamada incondicional de un bloque	
UC <ID_FUNCION>	Llamada incondicional de un bloque	No permite paso de parámetros ni llamadas a bloques FB
CC <ID_FUNCION> [RLO=1]	Llamada condicionada de un bloque	No permite paso de parámetros ni llamadas a bloques FB
BE	Fin incondicional de bloque (autómata)	
BEA	Fin incondicional de bloque (usuario)	
BEB [RLO=1]	Fin condicionado a RLO=1	

INVOCACIÓN CON GESTIÓN DE ERROR (Bit RB)

ENABLE-ENABLE OUT



EJEMPLO

GESTIÓN DE ERROR MEDIANTE EL BIT RB

Bloque invocante

```

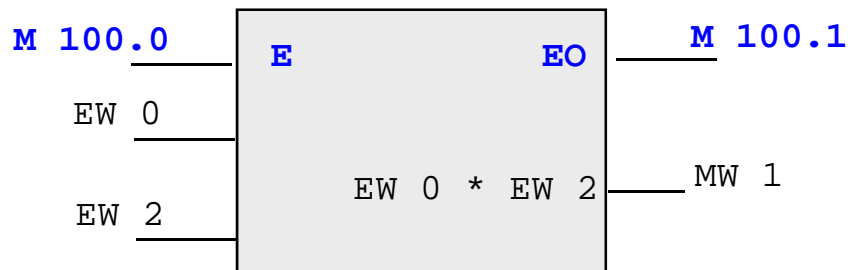
U M100.0
SPBNB _001
CALL FC1
_001: U BIE
      = M100.1
    
```

FC1

```

L EW 0
L EW 2
*I
SPO _ERR
T MW 1
BEA
_ERR: CLR
      SAVE
    
```

BLOQUE MULTIPLICADOR



EJEMPLO (II)

GESTIÓN DE ERROR MEDANTE EL BIT RB

```

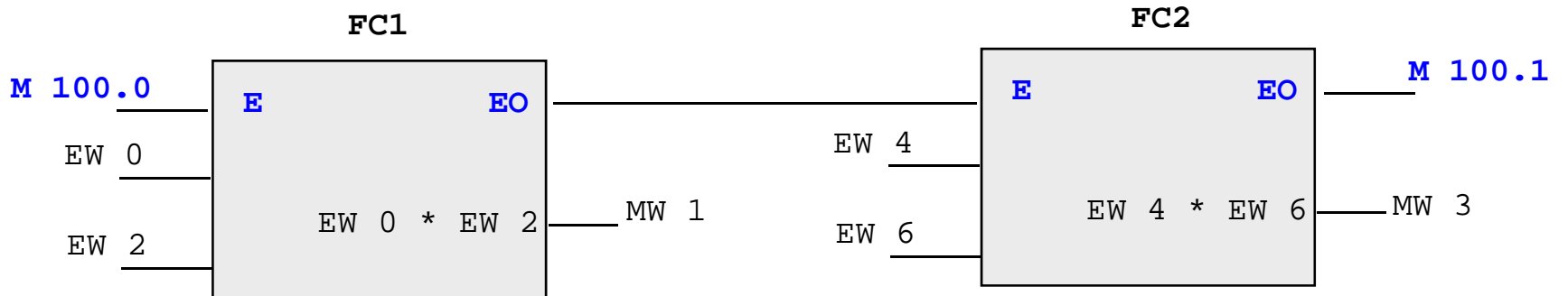
    U M100.0
    SPBNB _001
    CALL FC1
    _001:  U BIE

    SPBNB _002
    CALL FC2
    _002:  U BIE
           = M100.1
    
```

¿ERROR?

M100.0=1 y M100.1=0

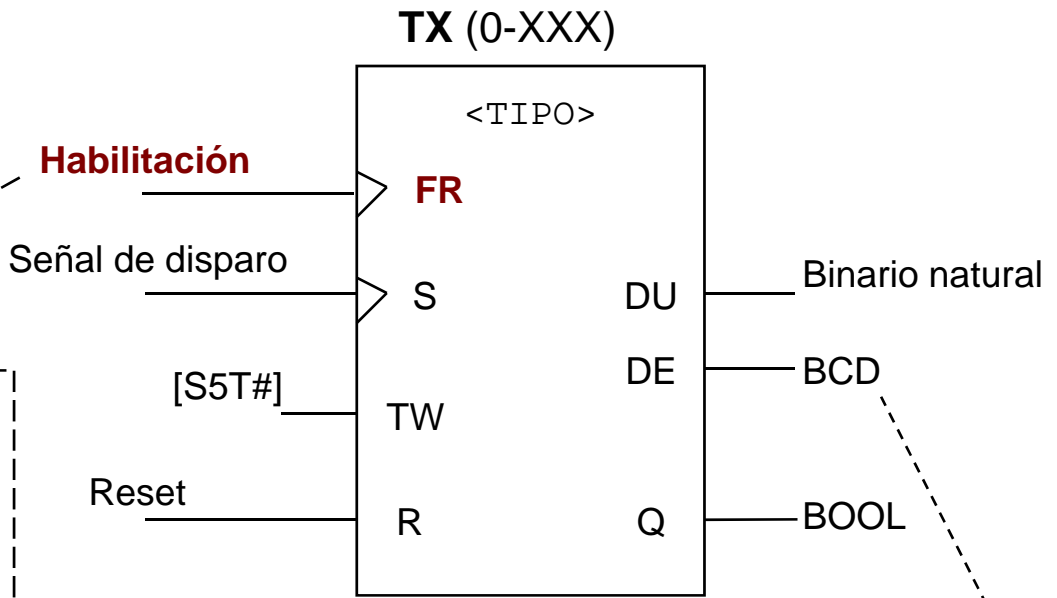
CADENA DE BLOQUES MULTIPLICADORES



TEMPORIZADORES

BLOQUES KOP

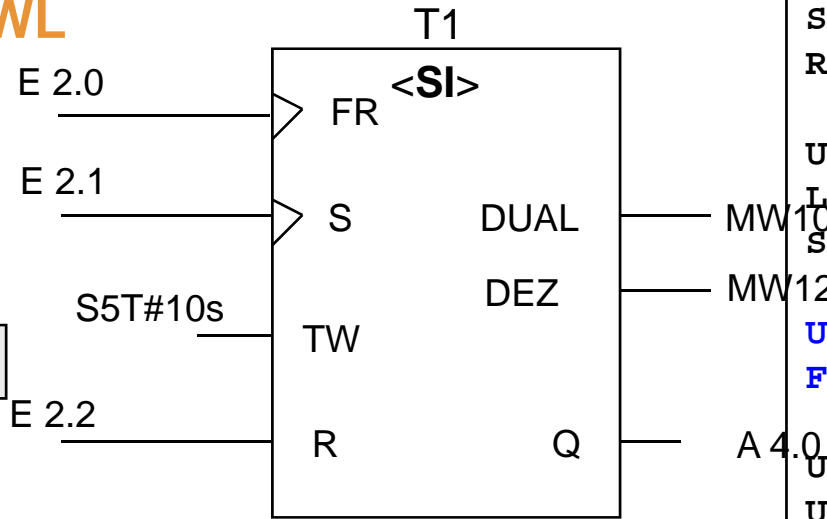
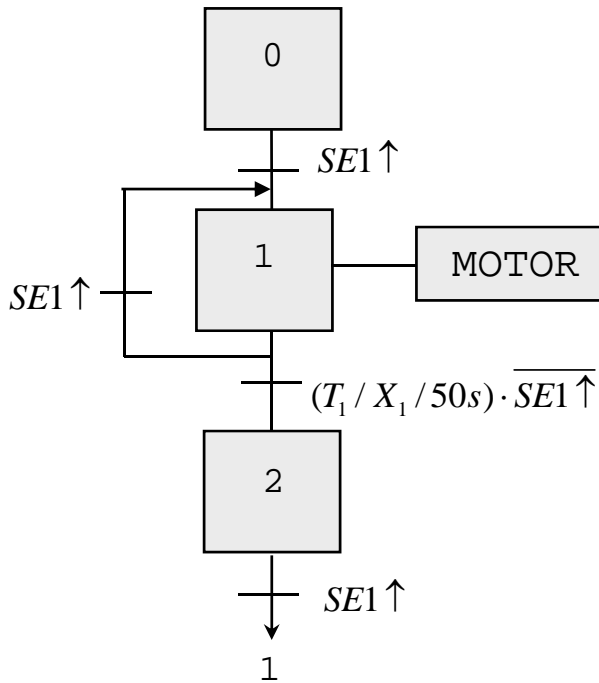
Habilitación (FR)
 Borra el biestable que detecta el flanco de la señal de disparo.
 En consecuencia redispara el temporizador si se encuentra la señal de disparo activada por nivel.



REGISTROS TX, TW, DU, DE \longrightarrow 16 BITS

TEMPORIZADORES (II)

IMPLEMENTACIÓN AWL



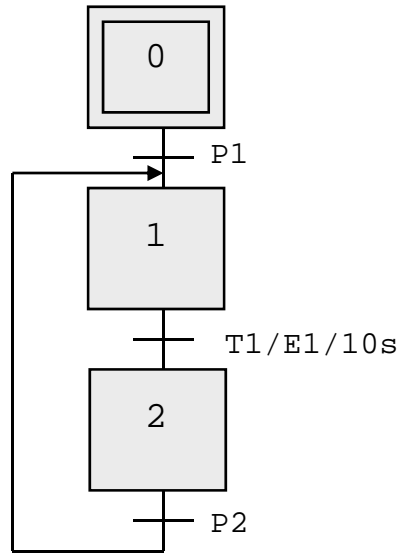
U	E0
U	FlancoSE1
S	E1
R	E0
U	E1
L	S5T#50s
SE	T1
U	SE1
FR	T1
U	E1
U	T1
S	E2
R	E1
//...	

¿PRIORIDAD T1 o Flanco de SE1?

```

U      E 2.0
FR    T1      //Habilitar T1
U      E 2.1
L      S5T#10s // ACU1_L = 10 segundos
SI    T1
U      E 2.2
R      T1      //Pone T1 a 0
U      T1      //Consulta el estado T1.
=      A 4.0
L      T1      //ACU1_L= valor de T1 (binario)
T      MW10
LC    T1      // ACU1_L= valor de T1 (BCD)
=      MW12
    
```

MOSTRAR EL 'CERO' AL TEMPORIZADOR



Ecuación Activación Etapa 2

```

U E1
SPBN _002
L S5T#10s
SE T1
    
```

```

U E1
U T1
S E2
R E1
    
```

```

_002:  NOP 0
// ...
    
```

SOLUCION

OPCIÓN A
OPCIÓN B
CORRECTO?

```

U E1
U E2
SPBN _002
SPBN _003
L S5T#10s
SE T1
SE T1
U E1 P2
U E1 E1
U T1 E1
U T1 E2
S E2
S E2
R E1
R NOP 0
R T1
    
```

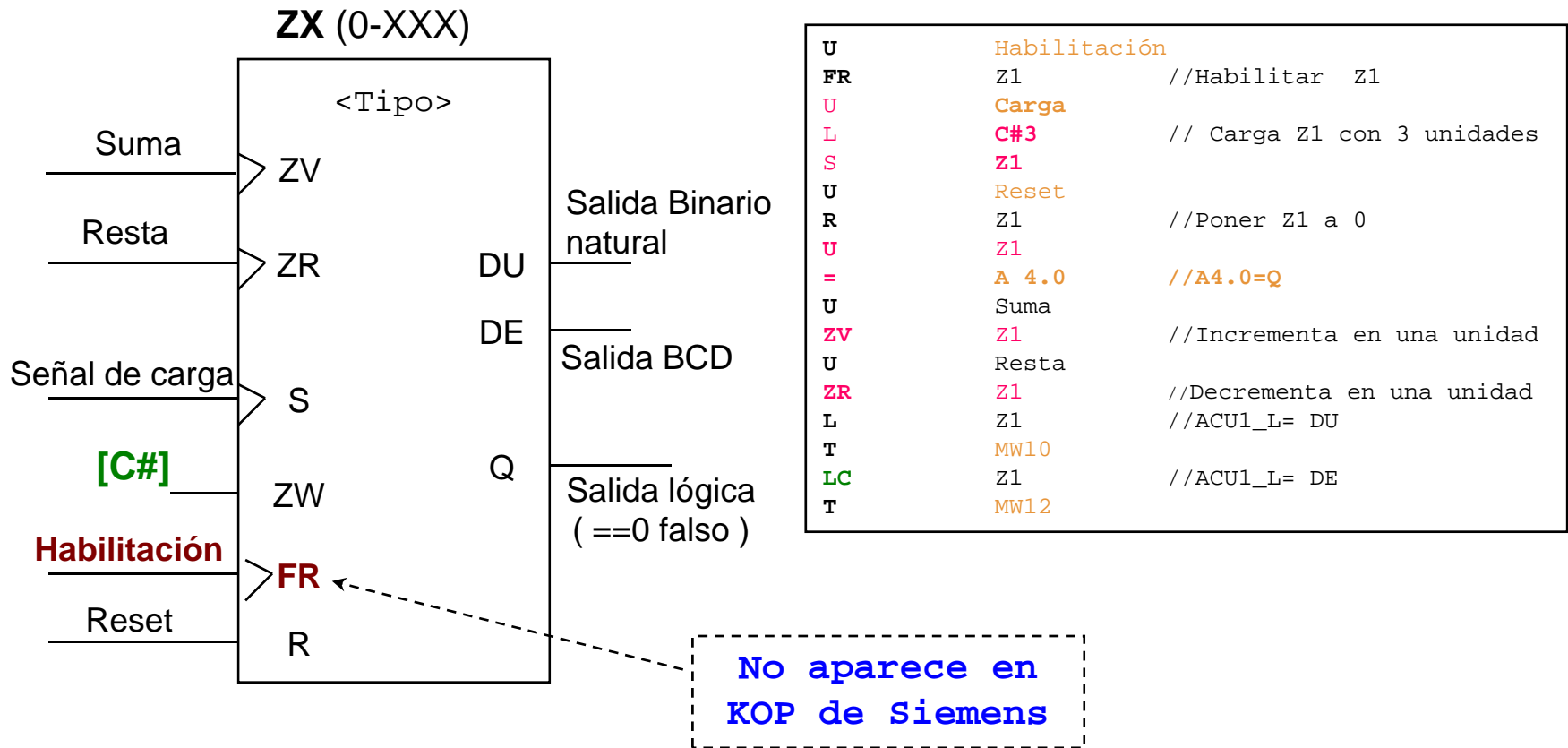
```

_003:  NOP 0
_002:  NOP 0
FR T1
_002:  NOP 0
    
```

El temporizador T1 no funciona a partir de la segunda activación de la etapa 1. Hay que 'mostrarle el cero'

CONTADORES

BLOQUES KOP



REGISTROS ZX, ZW, DU, DE  16 BITS

PASO DE PARÁMETROS

BLOQUE SUMADOR: FUNCIÓN

The screenshot shows the SIMATIC Manager interface for a function block (FC1) named 'Sumador'. The main window displays the parameter configuration for the function, which is a summing function for two 16-bit unsigned integers. The parameters are defined in a table:

Nombre	Tipo de datos	Comentario
Sum1	Word	Sumando 1
Sum2	Word	Sumando2

Below the table, the function's description is shown: 'Función que suma dos enteros de 16 bits sin signo'. The parameter list is as follows:

```

Segm. 1: Titulo:
Comentario:

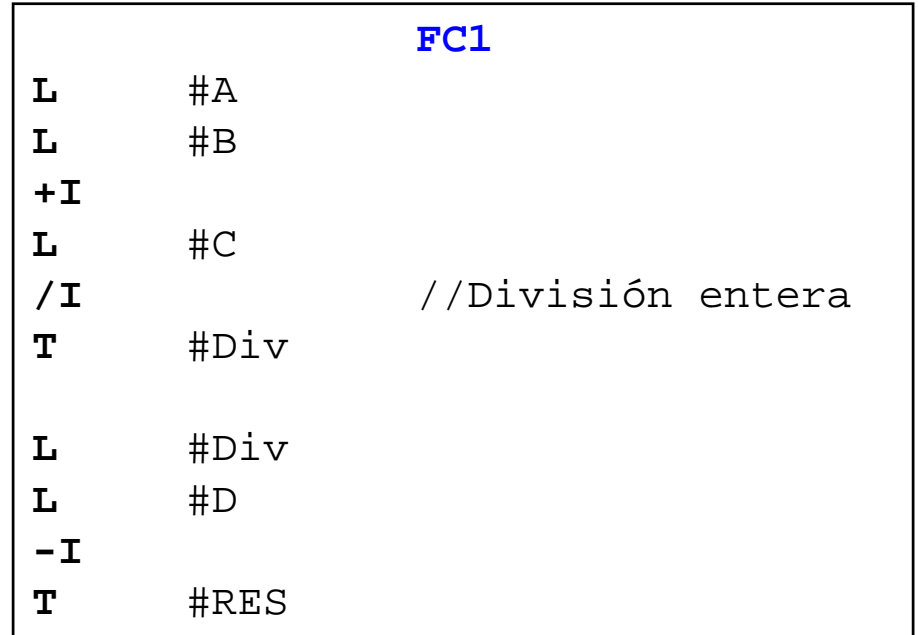
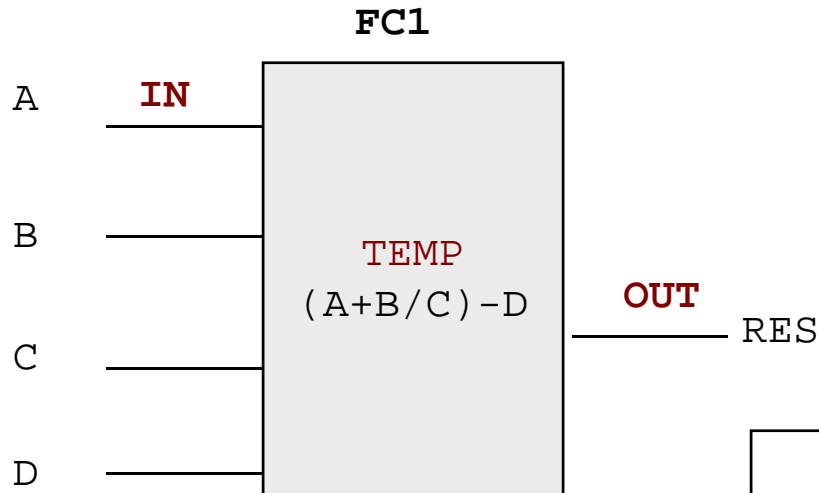
L   #Sum1
L   #Sum2
+I
SP0 _ERR
T   #Salida
BEA
_ERR: CLR
SAVE
L   0
T   #Salida
  
```

A callout box points to the parameter symbols (#Sum1 and #Sum2) with the text: 'Se usa # para indicar parámetro'.

The status bar at the bottom indicates the current position: 'Abs < 5.2 Seg 1 Lin 1 INS'.

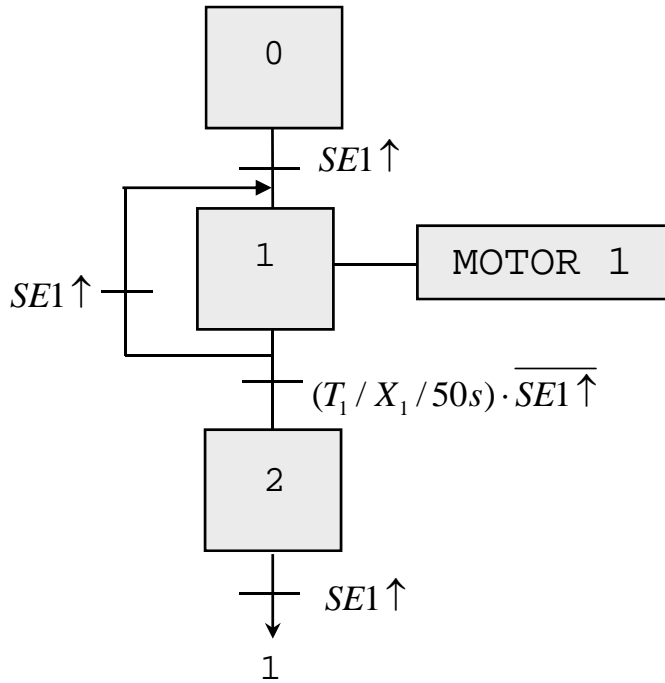
PASO DE PARÁMETROS

BLOQUE OPERACION NUMÉRICA: $[(A+B)/C]-D$



PASO DE PARÁMETROS

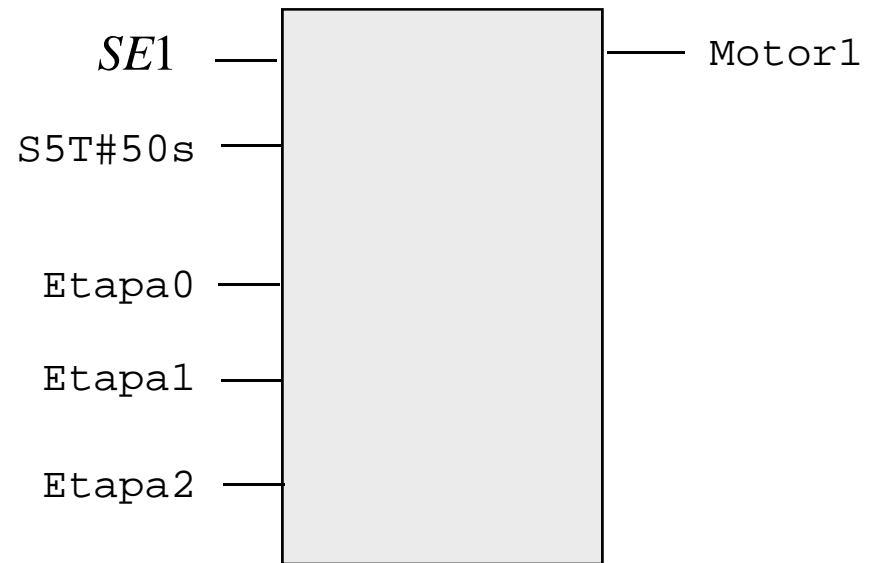
IMPLEMENTACIÓN AWL



Tipo IN

FC1

Tipo OUT



PASO DE PARÁMETROS

BLOQUE SUMADOR: BLOQUE INVOCANTE

The screenshot displays the SIMATIC Manager interface for a project named 'KOP/AWL/FUP - [OB1 -- S7_Pro21\Equipo SIMATIC 300\CPU312C(1)]'. The left-hand pane shows a project tree with 'Programa S7' containing 'OB1', 'FC1', 'OB35', and 'FC2'. The main workspace shows the 'Interface' environment for 'Entorno\Interface', with a table listing the parameter 'TEMP'. Below this, the ladder logic for 'OB1 : "Main Program Sweep (Cycle)"' is visible, including a 'Segm. 1' section with a call to function block 'FC 1'. The call parameters are: 'Sum1 :=MW10', 'Sum2 :=MW12', and 'Salida:=MW14'. A red dashed box highlights the 'FC 1' call, and a blue dashed box points to it with the text 'Completar la gestión del Enable Out'. The status bar at the bottom indicates 'STOP' and 'Seg 1 Lín 6'.

Contenido de: 'Entorno\Interface'

Nombre
TEMP

OB1 : "Main Program Sweep (Cycle)"

Comentario:

Segm. 1: Título:

Comentario:

```
CALL FC 1
Sum1 :=MW10
Sum2 :=MW12
Salida:=MW14
```

Completar la gestión del Enable Out

1: Error 2: Info 3: Referencias cruzadas 4: Información operando 5: Forzado 6: Diagnóstico 7: (

Pulse F1 para obtener ayuda. STOP Abs < 5.2 Seg 1 Lín 6 INS