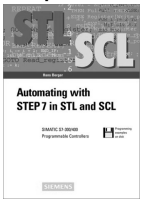
Tema 4

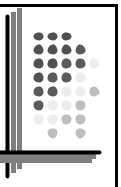
Instrucciones Básicas:
Operaciones lógicas con bits

Bibliografía

- Título: “Step7 Avanzado”
  - Autor: José Martínez Torres
  - Descargar de la página web
- Manual Siemens “Step7-AWL para S7-300 y S7-400”
- Manual Siemens “Step7-KOP para S7-300 y S7-400”
- Manual Siemens “Step7-FUP para S7-300 y S7-400”
- Título: “Automating with Step7 in STL and SCL”
  - Autor: Hans Berger
  - ISBN: 3-89578-140-1

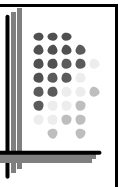


Índice



- Operaciones lógicas con bits
  - Operaciones básicas
  - Instrucciones de terminación de cadenas lógicas
  - Combinación de operaciones básicas
  - Función memoria
  - Instrucciones que afectan al RLO
  - Operaciones que detectan cambios en el resultado lógico

Índice



- **Operaciones lógicas con bits**
  - Operaciones básicas
  - Instrucciones de terminación de cadenas lógicas
  - Combinación de operaciones básicas
  - Función memoria
  - Instrucciones que afectan al RLO
  - Operaciones que detectan cambios en el resultado lógico

Operaciones lógicas con bits

- Las operaciones lógicas con bits operan con dos dígitos, 1 y 0.
  - Estos dos dígitos constituyen la base de un sistema numérico denominado sistema binario. Los dos dígitos 1 y 0 se denominan dígitos binarios o bits.
  - En el ámbito de los contactos y bobinas, un 1 significa activado (”conductor”) y un 0 significa desactivado (”no conductor”).
- Las operaciones lógicas con bits interpretan los estados de señal 1 y 0, y los combinan de acuerdo con la lógica del Álgebra de Boole.
  - Estas combinaciones producen un 1 ó un 0 como resultado y se denominan ”resultado lógico” (RLO). Las operaciones lógicas con bits permiten ejecutar las más diversas funciones.

Operaciones lógicas con bits

1.- Las operaciones básicas para las operaciones lógicas con bits son:

U	Y
UN	Y-No
O	O
ON	O-No
X	O-exclusiva
XN	O-exclusiva-No

2.- Para terminar una cadena lógica se puede utilizar una de las tres operaciones:

=	Asignar
R	Desactivar
S	Activar

3.- Las siguientes operaciones permiten ejecutar una cadena lógica encerrada entre paréntesis:

U(	Y con abrir paréntesis
UN(	Y-No con abrir paréntesis
O(	O con abrir paréntesis
ON(	O-No con abrir paréntesis
X(	O-exclusiva con abrir paréntesis
XN(	O-exclusiva-NO con abrir paréntesis
)	Cerrar paréntesis

4.- Las operaciones siguientes permiten modificar el resultado lógico (RLO):

NOT	Negar el RLO
SET	Activar el RLO (=1)
CLR	Desactivar RLO (=0)
SAVE	Memorizar el RLO en el registro RB

5.- Otras operaciones detectan cambios en el resultado lógico y reaccionan correspondientemente:

FN	Flanco negativo
FP	Flanco positivo

Índice

- Operaciones lógicas con bits
  - Operaciones básicas
  - Instrucciones de terminación de cadenas lógicas
  - Combinación de operaciones básicas
  - Función memoria
  - Instrucciones que afectan al RLO
  - Operaciones que detectan cambios en el resultado lógico

Operaciones básicas

Formato

U <bit>

Operando	Tipo de datos	Area de memoria
<bit>	BOOL	E, A, M, L, D, T, Z

Descripción de la operación

U consulta el bit direccionado para saber si tiene el estado de señal "1", y combina el resultado de la consulta con el RLO realizando una Y lógica.

Formato

UN <bit>

Operando	Tipo de datos	Area de memoria
<bit>	BOOL	E, A, M, L, D, T, Z

Descripción de la operación

UN consulta el bit direccionado para saber si tiene el estado de señal "0" y combina el resultado de la consulta con el RLO realizando una Y lógica.

Operaciones básicas

Formato

O <bit>

Operando	Tipo de datos	Area de memoria
<Bit>	BOOL	E, A, M, L, D, T, Z

Descripción de la operación

O consulta el bit direccionado para saber si tiene el estado de señal "1", y combina el resultado de la consulta con el RLO realizando una O lógica.

Formato

ON <bit>

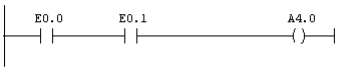
Operando	Tipo de datos	Area de memoria
<bit>	BOOL	E, A, M, L, D, T, Z

Descripción de la operación

ON consulta el bit direccionado para saber si tiene el estado de señal "0", y combina el resultado de la consulta con el RLO realizando una O lógica.

Operaciones básicas

- Serie



SOLUCIÓN EN AWL

```
U   E   0.0
U   E   0.1
=   A   4.0
```

- Paralelo

SOLUCIÓN EN AWL

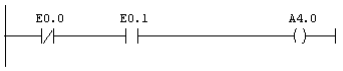
```
U   E   0.0     (también  O   E   0.0)
O   E   0.1
=   A   4.0
```

Operaciones básicas

- Contactos negados

SOLUCIÓN EN AWL

```
UN    E    0.0
U     E    0.1
=     A    4.0
```



Índice

- Operaciones lógicas con bits
  - Operaciones básicas
  - **Instrucciones de terminación de cadenas lógicas**
  - Combinación de operaciones básicas
  - Función memoria
  - Instrucciones que afectan al RLO
  - Operaciones que detectan cambios en el resultado lógico

Instrucciones de terminación de cadenas lógicas

Formato

= <bit>

Operando	Tipo de datos	Area de memoria
<bit>	BOOL	E, A, M, L, D, T, Z

Descripción de la operación

= <bit> escribe el RLO en el bit direccionado si el Master Control Relay está conectado (MCR = 1). Si el MCR es 0, en el bit direccionado se escribe el valor "0" en vez del RLO.

Instrucciones de terminación de cadenas lógicas

Formato

R <bit>

Operando	Tipo de datos	Area de memoria
<bit>	BOOL	E, A, M, L, D

Descripción de la operación

R (Desactivar bit) escribe el valor "0" en el bit direccionado si el RLO es 1 y si el Master Control Relay (MCR = 1) está conectado. Si el MCR es 0, el bit direccionado no varía.

Formato

S <bit>

Operando	Tipo de datos	Area de memoria
<bit>	BOOL	E, A, M, L, D

Descripción de la operación

S (Activar bit) escribe el valor "1" en el bit direccionado si el RLO es 1 y si el Master Control Relay (MCR = 1) está conectado. Si el MCR es 0, el bit direccionado no varía.

Instrucciones de terminación de cadenas lógicas

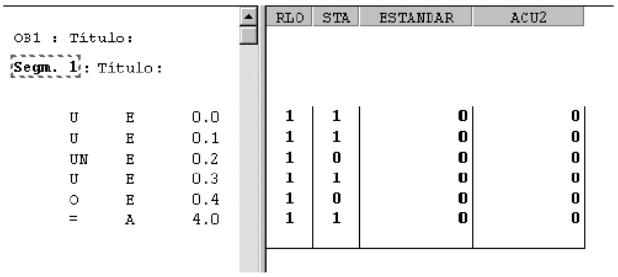
- Las instrucciones SET y RESET son instrucciones de memoria.
- Si programamos un SET de una salida o de una marca con unas condiciones, se activará cuando se cumplan dichas condiciones. Aunque las condiciones dejen de cumplirse, no se desactivará hasta que se haga un RESET de la salida o marca.
- Estas instrucciones tienen prioridad. Dependen del orden en que las programemos. Siempre va a tener prioridad la última que programemos.
- En nuestro caso, si hacemos un SET y un RESET dentro del mismo ciclo de scan, al final de cada ciclo hará efecto lo último que hayamos programado.

Instrucciones de terminación de cadenas lógicas

- **RLO**
  - Las instrucciones vistas hasta ahora son consultas y asignaciones. Esto significa: el procesador examina el estado de las señales de entrada, salida y marcas y le asigna a un estado de señal a las salidas y a las marcas.
  - Dos o más primeras consultas generan una operación lógica. El resultado de estas consultas es el resultado de la operación lógica (RLO). El resultado de la operación lógica proveniente de una operación lógica AND o una OR puede ser asignado a una salida o a una marca.
- **Primera Consulta**
  - La instrucción que hace la primera consulta después de una asignación se denomina de primera consulta. Esto significa que se genera un resultado de la operación lógica completamente nuevo, independiente del resultado previo de la operación lógica. Carece de importancia si la instrucción de primera consulta es una AND o una OR.

	RLO	estado de señal	
U E 1.0	....	....	
UN E 1.1	....	....	
U M 4.0	....	....	
= A 8.0	....	....	
U E 2.0			Primera consulta

Instrucciones de terminación de cadenas lógicas



Índice

- Operaciones lógicas con bits
  - Operaciones básicas
  - Instrucciones de terminación de cadenas lógicas
  - **Combinación de operaciones básicas**
  - Función memoria
  - Instrucciones que afectan al RLO
  - Operaciones que detectan cambios en el resultado lógico

Combinación de operaciones básicas

Formato

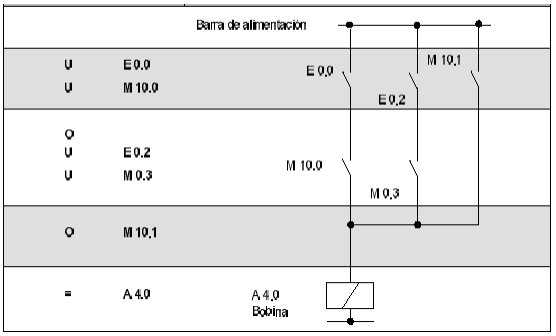
O

Descripción de la operación

La operación **O** realiza una O lógica de combinaciones Y siguiendo la regla Y antes de O.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe:	-	-	-	-	-	x	1	-	x



19

Combinación de operaciones básicas

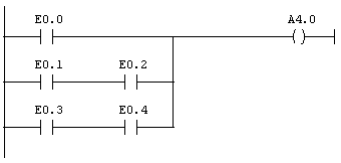
Programa AWL		Esquema de conexiones de relé
		barra de alimentación
U(O O)	E 0.0 M 10.0	E 0.0 M 10.0
U(O O)	E 0.2 M 10.3	E 0.2 M 10.3
U	M 10.1	M 10.1
=	A 4.0	A 4.0 Bobina

Combinación de operaciones básicas

- Utilización de parentesis

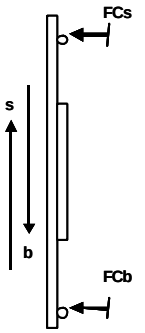
SOLUCIÓN EN AWL

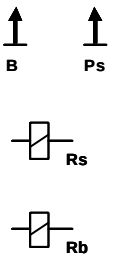
```
U  E    0.0
O(
U  E    0.1
U  E    0.2
)
O
U  E    0.3
U  E    0.4
=  A    4.0
```



Ejercicio 1

Control automático taladradora vertical





Proyectar un circuito para el control automático de una taladradora vertical. Dicha máquina deberá realizar la siguiente función:

1º.- Mediante el pulsador B iniciamos el descenso de la herramienta, la cual, al llegar a un minirruptor fin de carrera FCb, debe interrumpir el descenso e iniciar la subida.

2º.- Al llegar, en la subida, a un minirruptor fin de carrera FBs, la herramienta debe detenerse.

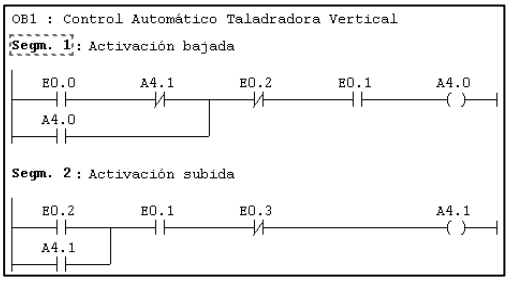
3º.- El circuito deberá llevar un pulsador de emergencia Ps, mediante al cual pueda interrumpirse el descenso de la herramienta, para que automáticamente se inicie la subida.

4º.- Cuando la herramienta esté subiendo de ninguna manera deberá poder iniciarse la bajada, aunque se pulse B.

Definido el diagrama de circuito, traducir a lenguaje STEP 7 la lógica cableada.

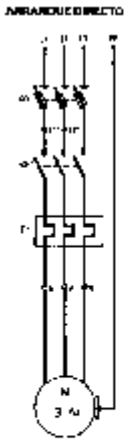
Ejercicio 1: Resolución

Asignación de variables	
Pulsador B	E0.0
Pulsador de emergencia Ps	E0.1
Final de carrera bajada FCb	E0.2
Final de carrera subida FCs	E0.3
Actuador bajada Rb	A4.0
Actuador subida Rs	A4.1




```
OB1 :
Segm. 1 : Activación bajada
      U(
      U     E     0.0
      UN    A     4.1
      O     A     4.0
      )
      UN    E     0.2
      U     E     0.1
      =     A     4.0

Segm. 2 : Activación subida
      U(
      O     E     0.2
      O     A     4.1
      )
      U     E     0.1
      UN    E     0.3
      =     A     4.1
```


Ejercicio 2



Arranque directo de un motor trifásico con rotor en cortocircuito

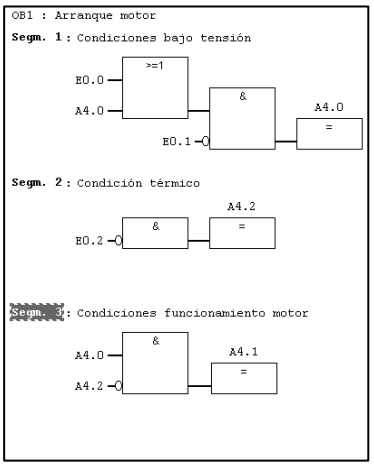
1- Elementos del esquema:

Q1	Seccionador con fusibles incorporados
KM1	Contactor de potencia
F1	Relé térmico de protección
M	Motor trifásico
LM	Lámpara que señala motor en marcha
LF1	Lámpara que señala disparo de F1
LBT	Lámpara que señala tensión en el circuito

2- Funcionamiento

Para poner en marcha, pulsar en S1.
Para parar, pulsar en S2
El motor también se desconectará por disparo de F1
Lámparas de señalización de circuido bajo tensión (LBT), motor en servicio (LM) y disparo de relé térmico (LF1)

Ejercicio 2: Resolución



Asignación de variables

Pulsador S1	E0.0
Pulsador S2	E0.1
Relé térmico F1(n.c)	E0.2
Lámpara LBT	A4.0
Lámpara LM	A4.1
Lámpara LF1	A4.2

```
OB1 : Arranque motor
Segm. 1 : Condiciones bajo tensión
      U(
      O     E     0.0
      O     A     4.0
      )
      UN    E     0.1
      =     A     4.0

Segm. 2 : Condición térmico
      UN    E     0.2
      =     A     4.2

Segm. 3 : Condiciones funcionamiento motor
      U     A     4.0
      UN    A     4.2
      =     A     4.1
```


Índice

- Operaciones lógicas con bits
  - Operaciones básicas
  - Instrucciones de terminación de cadenas lógicas
  - Combinación de operaciones básicas
  - **Función memoria**
  - Instrucciones que afectan al RLO
  - Operaciones que detectan cambios en el resultado lógico

Función memoria

- MARCAS
  - Las marcas son bits internos de la CPU. Disponemos de una cantidad limitada de marcas. Esta cantidad depende de la CPU con la que estemos trabajando.
  - Estos bits podremos activarlos o desactivarlos como si fueran salidas. En cualquier punto del programa los podremos consultar.
  - A las marcas les llamaremos M. A continuación tenemos que decir a que bit en concreto nos estamos refiriendo. Por ejemplo tenemos las marcas, M 0.0, M 10.7, M 4.5, etc.

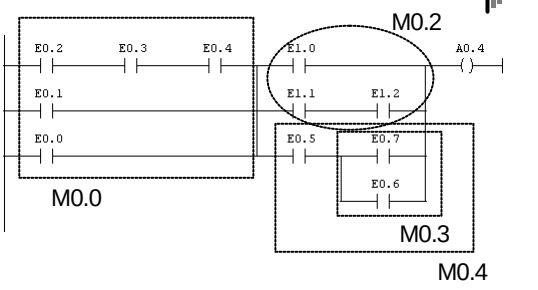
Función memoria

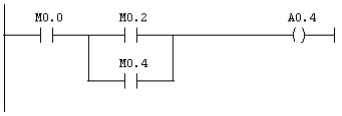
```
U(
  U   E   0.2
  U   E   0.3
  U   E   0.4
  O   E   0.1
  O   E   0.0
  )
  U(
  O   E   1.0
  O
  U   E   1.1
  U   E   1.2
  O
  U   E   0.5
  U(
  O   E   0.7
  O   E   0.6
  )
  )
  =   A   0.4
```

- Ejemplo MARCAS

SOLUCIÓN EN AWL

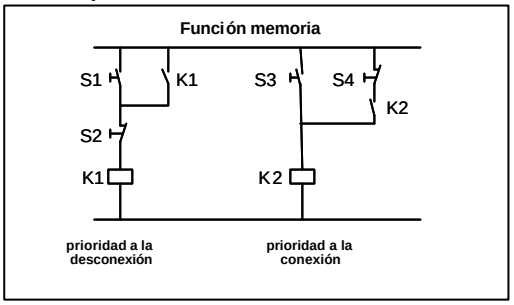
```
U   E   0.0
O   E   0.1
O(
U   E   0.2
U   E   0.3
U   E   0.4
)
=   M   0.0
U   E   0.6
O   E   0.7
=   M   0.1
U   E   1.1
U   E   1.2
O   E   1.0
=   M   0.2
U   E   0.5
U   M   0.1
=   M   0.3
U   M   0.0
U(
U   M   0.3
O   M   0.2
)
=   A   4.0
```



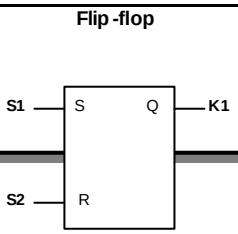


Función Memoria

- Si el sensor es un pulsador (momentáneamente conectado), entonces la señal está activada tanto tiempo como mantengamos accionado el pulsador. Para poder almacenar el estado de esta operación, deberemos emplear la **Función memoria**.



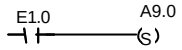
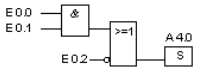
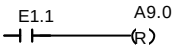
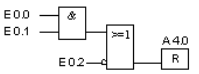
Función Memoria



- **Función memoria en un PLC**
  - En un PLC, la función memoria se elabora mediante el flip-flop S-R.El flip-flop dispone de dos entradas: una para la instrucción de activación **S** y otra para la instrucción de desactivación **R**.
- **Set**
  - Un "**1**" en la entrada de **SET**, activa la función memoria. La salida Q del flip-flop alcanza el valor de señal "**1**".
- **Reset**
  - Un "**1**" en la entrada de **RESET**, desactiva la función memoria. La salida Q del flip-flop alcanza el valor de señal "**0**".
- Debemos analizar el caso de que en ambas entradas se alcance al valor "1". Según el flip-flop que utilicemos (Set/Reset o Reset/Set) la prioridad será a la desconexión o a la conexión respectivamente.
- El valor de señal **"0"** en cualquiera de las dos entradas no modifica el valor del resultado del flip-flop. (prevalece el valor antiguo)

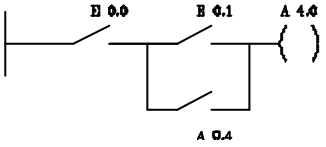
Función Memoria

Instrucciones de Set y Reset

KOP	FUP	AWL
SET	Set	Set
		U E1.0 S A9.0
RESET	Reset	Reset
		U E1.1 R A9.0

Función Memoria

- Ejemplo



- Solución con Set y Reset

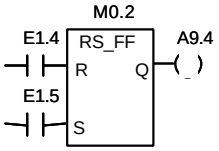
SOLUCIÓN EN AWL

U	E	0.0
S	A	4.0
U	E	0.1
R	A	4.0

Función Memoria

Flip-flop "Prioridad al Set"

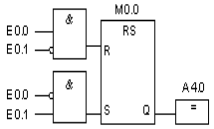
Set/reset el flip flop



Reset/set flip flop

```
U   E1.4
R   M0.2
U   E1.5
S   M0.2
U   M0.2
=   A9.4
```

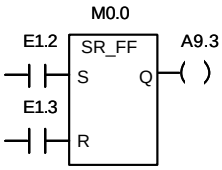
Reset/set flip flop



Función Memoria

Flip-flop "Prioridad al Reset"

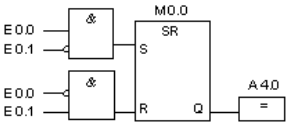
Set/reset el flip flop



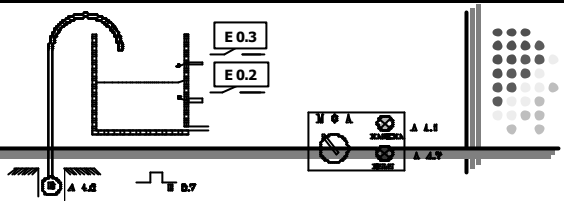
Set/reset flip flop

```
U   E1.2
S   M0.0
U   E1.3
R   M0.0
U   M0.0
=   A9.3
```

Set/reset flip flop



Ejemplo



- Depósito de agua
  - Tenemos un depósito de agua. Para manejarlo tenemos un selector de mando. Podemos seleccionar modo manual(E 0.0) o modo automático (E 0.1). Si seleccionamos modo manual, lo que queremos es que mientras esté conectada, la bomba esté funcionando(A 4.0), y cuando desconectemos que se pare la bomba. No queremos que se haga caso a las boyas de nivel.
  - Si lo tenemos en modo automático queremos que el nivel se mantenga entre las dos boyas. Cuando el agua llegue al nivel de abajo(E 0.2) queremos que se ponga en marcha la bomba, y cuando el agua llegue al nivel de arriba(E 0.3) queremos que se pare la bomba.
  - Además tenemos un relé térmico(E 0.7) que actúa tanto cuando tenemos la bomba en funcionamiento manual como cuando la tenemos en funcionamiento automático. Cuando salta el relé, queremos que se pare la bomba y que nos avise con un indicador luminoso en el cuadro de mando(A 4.7).
  - Además tenemos una luz de marcha que nos indica cuando está en marcha la bomba(A 4.1).

Ejemplo:solución

```
SOLUCIÓN EN AWL
Segmento 1: MANUAL
U   E    0.0      //Si activamos en modo manual
=   A    4.0      //Pon en marcha la bomba
=   A    4.1      //Enciende la luz de marcha
Segmento 2: AUTOMÁTICO
U   E    0.1      //Si está en automático
U   E    0.7      //Y está bien el relé
U   E    0.2      //Y está activo el nivel de abajo
UN  E    0.3      //Y no está activo el nivel de arriba
S   A    4.0      //Pon en marcha la bomba
S   A    4.1      //Y enciende la luz de marcha
U   E    0.1      //Si está en automático
U   E    0.7      //Y está bien el relé
UN  E    0.2      //Y no está activo el nivel de abajo
U   E    0.3      //Y se ha activado el nivel de arriba
ON  E    0.7      //O ha saltado el relé
R   A    4.0      //Para la bomba
R   A    4.1      //Apaga la luz de marcha
UN  E    0.7      //Si ha saltado el relé
=   A    4.7      //Avísame con la luz de
```

Ejemplo:solución

- Si hacemos la prueba de este circuito veremos que no funciona correctamente. Vemos que en modo manual sí que funciona pero en modo automático no para la bomba cuando debería.
- Para resolver este circuito correctamente, nos hace falta utilizar marcas auxiliares. En un mismo bloque no podemos activar la misma salida dos veces con condiciones diferentes porque se interfieren entre ellas.
- Memoria imagen de salidas y entradas

Ejemplo:solución

```
OB1 :   "Main Program Sweep (Cycle)"
Comentario:

Segm. 1: Titulo:
Comentario:

      U     E     0.0                  //Si activamos en modo manual
      =     A     4.0                  //Pon en marcha la bomba
      =     A     4.1                  //Enciende la luz de marcha


Segm. 2 : Titulo:
Comentario:

      U     E     0.1                  //Si está en automático
      U     E     0.7                  //Y está bien el relé
      U     E     0.2                  //Y está activo el nivel de abajo
      UN    E     0.3                  //Y no está activo el nivel de arriba
      S     A     4.0                  //Pon en marcha la bomba
      S     A     4.1                  //Y enciende la luz de marcha
      U     E     0.1                  //Si está en automático
      U     E     0.7                  //Y está bien el relé
      UN    E     0.2                  //Y no está activo el nivel de abajo
      U     E     0.3                  //Y se ha activado el nivel de arriba
      ON    E     0.7                  //O ha saltado el relé
      R     A     4.0                  //Para la bomba
      R     A     4.1                  //Apaga la luz de marcha
      UN    E     0.7                  //Si ha saltado el relé
      =     A     4.7                  //Avísame con la luz de
```

RLO	STA	ESTANDAR
0	0	0
0	0	0
0	0	0
0	0	0
0	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	1	0
0	0	0
0	0	0
0	1	0
0	0	0
0	1	0
0	0	0

Ejemplo:Manual

```
OB1 :   "Main Program Sweep (Cycle)"
Comentario:

Segm. 1: Título:
Comentario:

      U     E      0.0                  //Si activamos en modo manual
      =     A      4.0                  //Pon en marcha la bomba
      =     A      4.1                  //Enciende la luz de marcha


Segm. 2: Título:
Comentario:

      U     E      0.1                  //Si está en automático
      U     E      0.7                  //Y está bien el relé
      U     E      0.2                  //Y está activo el nivel de abajo
      UN    E      0.3                  //Y no está activo el nivel de arriba
      S     A      4.0                  //Pon en marcha la bomba
      S     A      4.1                  //Y enciende la luz de marcha
      U     E      0.1                  //Si está en automático
      U     E      0.7                  //Y está bien el relé
      UN    E      0.2                  //Y no está activo el nivel de abajo
      U     E      0.3                  //Y se ha activado el nivel de arriba
      ON    E      0.7                  //O ha saltado el relé
      R     A      4.0                  //Para la bomba
      R     A      4.1                  //Apaga la luz de marcha
      UN    E      0.7                  //Si ha saltado el relé
      =     A      4.7                  //Avísame con la luz de
```

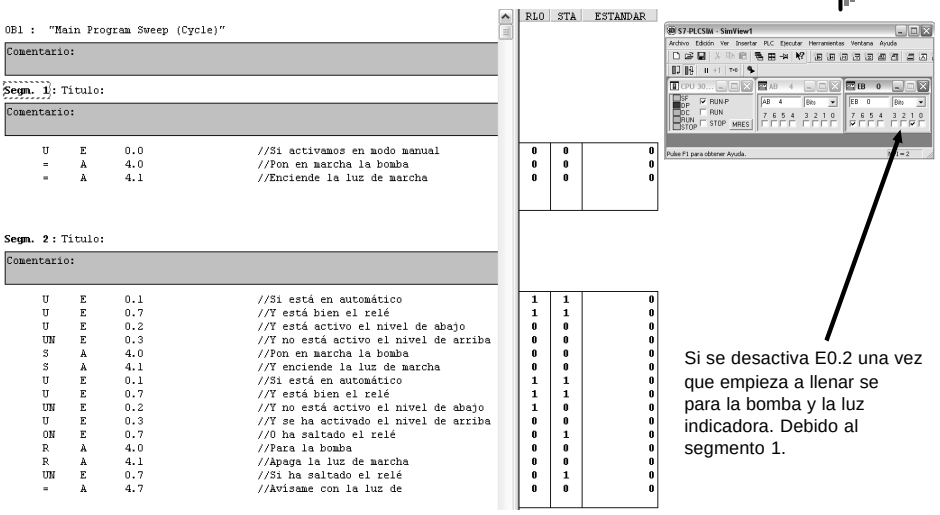
RLO	STA	ESTANDAR
1	1	0
1	1	0
1	1	0
0	0	0
0	1	0
0	0	0
0	0	0
0	1	0
0	0	0
0	1	0
0	0	0
0	0	0
0	1	0
0	1	0
0	1	0
0	1	0
0	0	0

Ejemplo:Automático

```
OB1 :   "Main Program Sweep (Cycle)"
Comentario:

Segm. 1: Título:
Comentario:

      U     E      0.0                  //Si activamos en modo manual
      =     A      4.0                  //Pon en marcha la bomba
      =     A      4.1                  //Enciende la luz de marcha


Segm. 2: Título:
Comentario:

      U     E      0.1                  //Si está en automático
      U     E      0.7                  //Y está bien el relé
      U     E      0.2                  //Y está activo el nivel de abajo
      UN    E      0.3                  //Y no está activo el nivel de arriba
      S     A      4.0                  //Pon en marcha la bomba
      S     A      4.1                  //Y enciende la luz de marcha
      U     E      0.1                  //Si está en automático
      U     E      0.7                  //Y está bien el relé
      UN    E      0.2                  //Y no está activo el nivel de abajo
      U     E      0.3                  //Y se ha activado el nivel de arriba
      ON    E      0.7                  //O ha saltado el relé
      R     A      4.0                  //Para la bomba
      R     A      4.1                  //Apaga la luz de marcha
      UN    E      0.7                  //Si ha saltado el relé
      =     A      4.7                  //Avísame con la luz de
```

RLO	STA	ESTANDAR
0	0	0
0	0	0
0	0	0
1	1	0
1	1	0
1	0	0
1	1	0
1	1	0
1	1	0
1	1	0
0	1	0
0	0	0
0	1	0
0	1	0
0	1	0
0	1	0
0	0	0

Ejemplo:Automático Funcionamiento Incorrecto



Si se desactiva E0.2 una vez que empieza a llenar se para la bomba y la luz indicadora. Debido al segmento 1.

Ejemplo:solución

Correctamente

Segmento 1 : MANUAL

```
U   E   0.0   //Si está en manual
=   M   0.0   //Activa la marca 0.0
=   M   0.1   //Y activa la marca 0.1
```

Segmento 2: AUTOMÁTICO

```
U   E   0.1   //Si está en automático
U   E   0.7   //Y está el relé bien
U   E   0.2   //Y está activo el nivel inferior
UN  E   0.3   //Y no está activo el nivel superior
S   M   0.2   //Activa la marca 0.2
S   M   0.3   //Y activa la marca 0.3
U   E   0.1   //Si está en automático
U   E   0.7   //Y está el relé bien
UN  E   0.2   //Y no está activo el nivel inferior
U   E   0.3   //Y se ha activado el nivel superior
ON  E   0.7   //O ha saltado el relé
R   M   0.2   //Desactiva la marca 0.2
R   M   0.3   //Y desactiva la marca 0.3
UN  E   0.7   //Si no está el relé
=   A   4.7   //Activa la luz de relé.
```

```
U   M   0.0   //Si está activa la marca 0.0
O   M   0.2   //O está activa la marca 0.2
=   A   4.0   //Pon en marcha la bomba
U   M   0.1   //Si está activa la marca 0.1
O   M   0.3   //O la marca 0.3
=   A   4.1   //Enciende la luz de marcha
```

Ejemplo: Automático

```
OB1 :  "Main Program Sweep (Cycle)"
Comentario:

Segm. 1: Título:
Comentario:

      U     E     0.0                  //Si está en manual
      =     M     0.0                  //Activa la marca 0.0
      =     M     0.1                  //Y activa la marca 0.1

Segm. 2: Título:
Comentario:

      U     E     0.1                  //Si está en automático
      U     E     0.7                  //Y está el relé bien
      U     E     0.2                  //Y está activo el nivel inferior
      UN    E     0.3                  //Y no está activo el nivel superior
      S     M     0.2                  //Activa la marca 0.2
      S     M     0.3                  //Y activa la marca 0.3
      U     E     0.1                  //Si está en automático
      U     E     0.7                  //Y está el relé bien
      UN    E     0.2                  //Y no está activo el nivel inferior
      U     E     0.3                  //Y se ha activado el nivel superior
      ON    E     0.7                  //O ha saltado el relé
      R     M     0.2                  //Desactiva la marca 0.2
      R     M     0.3                  //Y desactiva la marca 0.3
      UN    E     0.7                  //Si no está el relé
      =     A     4.7                  //Activa la luz de relé.
      UN    E     0.7                  //Si ha saltado el relé
      R     M     0.0                  //Desactiva la marca 0.0
      R     M     0.1
      U     M     0.0                  //Si está activa la marca 0.0
      O     M     0.2                  //O está activa la marca 0.2
      =     A     4.0                  //Pon en marcha la bomba
      U     M     0.1                  //Si está activa la marca 0.1
      O     M     0.3                  //O la marca 0.3
      =     A     4.1                  //Enciende la luz de marcha
```

RLO	STA	ESTANDAR
0	0	0
0	0	0
0	0	0
1	1	0
1	1	0
1	1	0
1	1	0
1	0	0
1	1	0
1	1	0
1	1	0
0	1	0
0	0	0
0	1	0
0	1	0
0	1	0
0	0	0
0	1	0
0	0	0
0	0	0
1	1	0
1	1	0
0	0	0
1	1	0
0	0	0
1	1	0

Ejemplo: Automático sin que se pare la bomba/luz

```
OB1 :  "Main Program Sweep (Cycle)"
Comentario:

Segm. 1: Título:
Comentario:

      U     E     0.0                  //Si está en manual
      =     M     0.0                  //Activa la marca 0.0
      =     M     0.1                  //Y activa la marca 0.1

Segm. 2: Título:
Comentario:

      U     E     0.1                  //Si está en automático
      U     E     0.7                  //Y está el relé bien
      U     E     0.2                  //Y está activo el nivel inferior
      UN    E     0.3                  //Y no está activo el nivel superior
      S     M     0.2                  //Activa la marca 0.2
      S     M     0.3                  //Y activa la marca 0.3
      U     E     0.1                  //Si está en automático
      U     E     0.7                  //Y está el relé bien
      UN    E     0.2                  //Y no está activo el nivel inferior
      U     E     0.3                  //Y se ha activado el nivel superior
      ON    E     0.7                  //O ha saltado el relé
      R     M     0.2                  //Desactiva la marca 0.2
      R     M     0.3                  //Y desactiva la marca 0.3
      UN    E     0.7                  //Si no está el relé
      =     A     4.7                  //Activa la luz de relé.
      UN    E     0.7                  //Si ha saltado el relé
      R     M     0.0                  //Desactiva la marca 0.0
      R     M     0.1
      U     M     0.0                  //Si está activa la marca 0.0
      O     M     0.2                  //O está activa la marca 0.2
      =     A     4.0                  //Pon en marcha la bomba
      U     M     0.1                  //Si está activa la marca 0.1
      O     M     0.3                  //O la marca 0.3
      =     A     4.1                  //Enciende la luz de marcha
```

RLO	STA	ESTANDAR
0	0	0
0	0	0
0	0	0
1	1	0
1	1	0
0	0	0
0	1	0
0	1	0
1	1	0
1	1	0
1	0	0
0	0	0
0	1	0
0	1	0
0	1	0
0	0	0
0	1	0
0	0	0
0	0	0
1	1	0
1	1	0
0	0	0
1	1	0
0	0	0
1	1	0

Ejemplo: solución

- Ahora ya no funciona el térmico en el modo manual. Al utilizar marcas diferentes para cada tipo de funcionamiento, el térmico sólo actúa sobre las marcas de modo automático. Sólo estamos haciendo un reset de una de las marcas que activan la bomba. Nos falta resetear la otra marca. Tendremos que añadir las siguientes líneas.

```
UN  E   0.7     //Si ha saltado el relé
R   M   0.0     //Desactiva la marca 0.0
R   M   0.1     //Y desactiva la marca 0.
```


Ejemplo: solución

- Ahora podemos hacer todas las objeciones que queramos y corregir sobre lo que ya tenemos hecho.
- Por ejemplo, en este caso no he tenido en cuenta la situación de que después de haber estado en manual o en automático, volvamos a la posición de reposo. En automático he hecho sets a ciertas marcas. Cuando volvamos a la posición de reposo esas marcas tendrán que volver a cero. De lo contrario podría darse el caso de que estando en la posición de reposo, tengamos la bomba en marcha. Para remediar esto podría añadir las siguientes instrucciones:

```
UN  E   0.0
UN  E   0.1
R   A   4.0
R   A   4.1
```

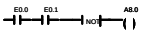
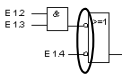
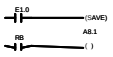
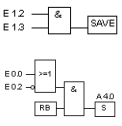
Índice

- Operaciones lógicas con bits
  - Operaciones básicas
  - Instrucciones de terminación de cadenas lógicas
  - Combinación de operaciones básicas
  - Función memoria
  - **Instrucciones que afectan al RLO**
  - Operaciones que detectan cambios en el resultado lógico

Instrucciones que afectan al RLO

- **NOT**
  - NOT es la instrucción para invertir el RLO.
- **CLR/SET**
  - El RLO pasa a 0 con la instrucción borrar CLEAR, y el RLO pasa a1 con la instrucción SET.
- **SAVE/U RB**
  - Con la instrucción SAVE (grabar memoria), el contenido del RLO se almacena en un registro (palabra de estado). El RLO almacenado puede ser consultado de nuevo con la instrucción U RB.

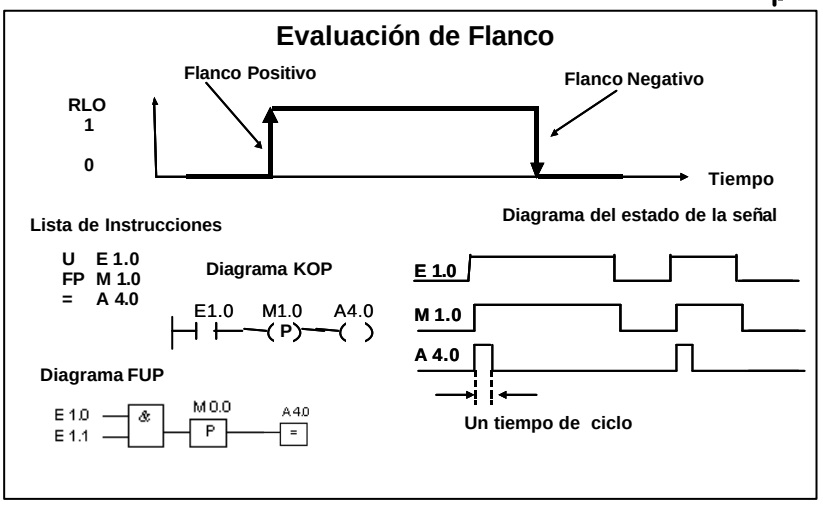
Instrucciones que Afectan al RLO

	KOP	FUP	AWL
NOT			U E0.0 U E0.1 NOT = A8.0
CLR SET	*No se visualiza en KOP*	*No se visualiza en FUP*	CLR SET
SAVE			U E1.6 SAVE U RB = A8.1

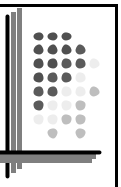
Índice

- Operaciones lógicas con bits
  - Operaciones básicas
  - Instrucciones de terminación de cadenas lógicas
  - Combinación de operaciones básicas
  - Función memoria
  - Instrucciones que afectan al RLO
  - **Operaciones que detectan cambios en el resultado lógico**

Flancos



Flancos



- En ocasiones necesitamos que una determinada acción sólo se realice una vez mientras se cumplan las condiciones para la activación de la misma.
- Una gran cantidad de sets de variables mejorarían si se les aplicase una señal de flanco positivo a sus condiciones de activación.
- La señal de flanco, tanto positivo como negativo en el Step 7 requiere de una marcha que no puede ser utilizada en otra parte del programa, por lo que es importante simbolizarla como exclusiva de ese flanco en cuestión

```
U     E     0.0                "Marcha"                --
FP    M     0.0                "Flanco_P_Marcha"       --
S     M     0.1                "Marca_Motor_Brazo"     --
U     E     0.1                "Paro"                  --
R     M     0.1                "Marca_Motor_Brazo"     --
U     M     0.1                "Marca_Motor_Brazo"     --
=     A     4.0                "Motor_Brazo"           --
```