SIEMENS

SIMATIC

Diagrama de funciones (FUP) para S7-300 y S7-400

Manual de referencia

Este manual forma parte del paquete de documentación con la referencia:

6ES7810-4CA10-8DW1

05/2010
A5E02790133-01

Notas jurídicas

Filosofía en la señalización de advertencias y peligros

Este manual contiene las informaciones necesarias para la seguridad personal así como para la prevención de daños materiales. Las informaciones para su seguridad personal están resaltadas con un triángulo de advertencia; las informaciones para evitar únicamente daños materiales no llevan dicho triángulo. De acuerdo al grado de peligro las consignas se representan, de mayor a menor peligro, como sigue.

⚠PELIGRO
Significa que, si no se adoptan las medidas preventivas adecuadas **se producirá** la muerte, o bien lesiones corporales graves.

⚠ADVERTENCIA
Significa que, si no se adoptan las medidas preventivas adecuadas **puede producirse** la muerte o bien lesiones corporales graves.

⚠PRECAUCIÓN
con triángulo de advertencia significa que si no se adoptan las medidas preventivas adecuadas, pueden producirse lesiones corporales.

PRECAUCIÓN
sin triángulo de advertencia significa que si no se adoptan las medidas preventivas adecuadas, pueden producirse daños materiales.

ATENCIÓN
significa que puede producirse un resultado o estado no deseado si no se respeta la consigna de seguridad correspondiente.

Si se dan varios niveles de peligro se usa siempre la consigna de seguridad más estricta en cada caso. Si en una consigna de seguridad con triángulo de advertencia se alarma de posibles daños personales, la misma consigna puede contener también una advertencia sobre posibles daños materiales.

Personal cualificado

El producto/sistema tratado en esta documentación sólo deberá ser manejado o manipulado por **personal cualificado** para la tarea encomendada y observando lo indicado en la documentación correspondiente a la misma, particularmente las consignas de seguridad y advertencias en ella incluidas. Debido a su formación y experiencia, el personal cualificado está en condiciones de reconocer riesgos resultantes del manejo o manipulación de dichos productos/sistemas y de evitar posibles peligros.

Uso previsto o de los productos de Siemens

Considere lo siguiente:

⚠ADVERTENCIA
Los productos de Siemens sólo deberán usarse para los casos de aplicación previstos en el catálogo y la documentación técnica asociada. De usarse productos y componentes de terceros, éstos deberán haber sido recomendados u homologados por Siemens. El funcionamiento correcto y seguro de los productos exige que su transporte, almacenamiento, instalación, montaje, manejo y mantenimiento hayan sido realizados de forma correcta. Es preciso respetar las condiciones ambientales permitidas. También deberán seguirse las indicaciones y advertencias que figuran en la documentación asociada.

Marcas registradas

Todos los nombres marcados con ® son marcas registradas de Siemens AG. Los restantes nombres y designaciones contenidos en el presente documento pueden ser marcas registradas cuya utilización por terceros para sus propios fines puede violar los derechos de sus titulares.

Exención de responsabilidad

Hemos comprobado la concordancia del contenido de esta publicación con el hardware y el software descritos. Sin embargo, como es imposible excluir desviaciones, no podemos hacernos responsable de la plena concordancia. El contenido de esta publicación se revisa periódicamente; si es necesario, las posibles las correcciones se incluyen en la siguiente edición.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
ALEMANIA

A5E02790133-01
℗ 02/2010

Prológo

Objetivo del manual

Este manual le servirá de ayuda al crear programas de usuario con el lenguaje de pogramación FUP.

Describe los elementos del lenguaje de programación FUP, así como su sintaxis y sus funciones.

Nociones básicas

Este manual está dirigido a programadores de programas S7, operadores y personal de mantenimiento que dispongan de conocimientos básicos sobre los autómatas programables.

Además es necesario estar familiarizado con el uso de ordenadores o equipos similares a un PC (p. ej. unidades de programación) bajo los sistemas operativos MS Windows XP, MS Windows Server 2003 o MS Windows 7.

Objeto del manual

El software en el que se basan las indicaciones del manual es STEP 7 V5.5.

Cumplimiento de la normativa IEC 1131-3

FUP sigue los principios del lenguaje ”Diagrama de funciones” fijados en la norma DIN EN-61131-3 (int. IEC 1131-3). En la tabla sobre cumplimiento de normas contenida en el archivo NORM_TAB.RTF de STEP 7 encontrará información más detallada sobre el cumplimiento de las normas.

Requisitos

Para entender correctamente el presente manual de FUP se requieren conocimientos teóricos acerca de los programas S7, que se pueden consultar en la Ayuda en pantalla de STEP 7. Como que los paquetes acerca de los lenguajes de programación se basan en el software estándar de STEP 7, debería conocerse ya mínimamente el uso del software y su documentación.

Este manual forma parte del paquete de documentación "STEP 7 Información de referencia“.

La tabla siguiente da una visión de conjunto de la documentación de STEP 7:

Manuales	**Tema**	**Referencia**
Información básica de STEP 7 compuesta por: • STEP 7: Introducción y ejercicios prácticos • Programar con STEP 7 • Configurar el hardware y la comunicación con STEP 7 • De S5 a S7, Guía para facilitar la transición	Nociones básicas para el personal técnico. Describe cómo realizar soluciones de control con el software STEP 7 para los sistemas S7-300/400.	6ES7810-4CA10-8DW0
Información de referencia para STEP 7, compuesta por • Manuales KOP/FUP/AWL para S7-300/400 • Funciones estándar y funciones de sistema para S7-300/400 Tomo 1 y Tomo 2	Esta obra de consulta describe los lenguajes de programación KOP, FUP y AWL así como las funciones estándar y las funciones de sistema como complemento a la 'Información básica de STEP 7.	6ES7810-4CA10-8DW1

Ayudas en pantalla	**Tema**	**Referencia**
Ayuda de STEP 7	Nociones básicas para diseñar programas y configurar el hardware con STEP 7. Disponible en forma de Ayuda en pantalla.	Componente del paquete de software STEP 7
Ayudas de referencia para AWL/KOP/FUP Ayudas de referencia para SFBs/SFCs Ayudas de referencia para los bloques de organización	Información de referencia sensible al contexto	Componente del paquete de software STEP 7

Ayuda en pantalla

Como complemento al manual puede recurrir a la Ayuda en pantalla integrada en el software.

A la Ayuda que está integrada en el software se accede de distinta manera:

- La Ayuda sensible al contexto ofrece información sobre el contexto actual, p. ej. sobre el cuadro de diálogo que esté abierto o sobre la ventana activa. Para acceder a esta ayuda pulse el botón de comando ”Ayuda” o bien la tecla F1.
- El menú **Ayuda** ofrece varios comandos de menú: **Temas de Ayuda** abre el índice de la Ayuda de STEP 7.
- A través de **"Glosario"** se accede al glosario para todas las aplicaciones de STEP 7.

Este manual es un extracto de la Ayuda de FUP. Debido a que la estructura del manual se corresponde a grandes rasgos con la de la Ayuda en pantalla puede alternar la lectura del manual con la de la Ayuda en pantalla.

Asistencia adicional

Si tiene preguntas relacionadas con el uso de los productos descritos en el manual a las que no encuentre respuesta, diríjase a la sucursal o al representante más próximo de Siemens, en donde le pondrán en contacto con el especialista.

Encontrará a su persona de contacto en la página de Internet:

http://www.siemens.com/automation/partner

Encontrará una guía sobre el conjunto de la información técnica correspondiente a los distintos productos y sistemas SIMATIC en la página de Internet:

http://www.siemens.com/simatic-tech-doku-portal

Encontrará el catálogo y el sistema de pedidos on-line en:

http://mall.automation.siemens.com/

Centro de formación SIMATIC

Para ofrecer a nuestros clientes un fácil aprendizaje de los sistemas de automatización SIMATIC S7, les ofrecemos distintos cursillos de formación. Diríjase a su centro de formación regional o a la central en D 90026 Nuernberg.

Internet: http://www.sitrain.com

Technical Support

Podrá acceder al Technical Support de todos los productos de Industry Automation and Drive Technology

- a través del formulario de Internet para el Support Request
  http://www.siemens.com/automation/support-request

Encontrará más información sobre nuestro Technical Support en la página de Internet
http://www.siemens.com/automation/service

Service & Support en Internet

Además de nuestra documentación, en Internet le ponemos a su disposición todo nuestro know-how.

http://www.siemens.com/automation/service&support

En esta página encontrará:

- "Newsletter" que le mantendrán siempre al día ofreciéndole informaciones de última hora,
- La rúbrica "Support al producto" con un buscador que le permitirá acceder a la información que necesita,
- El "Foro" en el que podrá intercambiar sus experiencias con cientos de expertos en todo el mundo,
- El especialista o experto de Industry Automation and Drive Technology de su región,
- Informaciones sobre reparaciones, piezas de repuesto y consulting.

Índice

1 Operaciones lógicas con bits

1.1 Lista de operaciones lógicas con bits

Descripción

Las operaciones lógicas con bits operan con dos dígitos, 1 y 0. Estos dos dígitos consituyen 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 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.

Se dispone de las operaciones lógicas con bits siguientes:

- & Y, >=1 O y XOR O-exclusiva: Éstas consultan el estado de señal y emiten resultados que se copian en el bit RLO o bien que se enlazan con el mismo.
- Operación O lógica de operaciones Y, y operación Y lógica de operaciones O
- = Asignación y # Conector: Éstas asignan el RLO o lo guardan de forma provisional.

Las siguientes operaciones reaccionan ante un RLO de 1:

- R : Desactivar salida
- S : Activar salida
- RS : Flipflop de desactivación/activación
- SR : Flipflop de activación/desactivación

Otras operaciones reaccionan ante un cambio de flanco positivo o negativo para ejecutar las siguientes funciones:

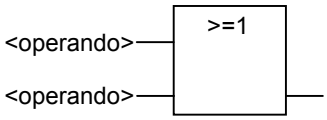
- N : Detectar flanco negativo (1 -> 0)
- P : Detectar flanco positivo (0 -> 1)
- NEG : Detectar flanco de señal 1 -> 0
- POS : Detectar flanco de señal 0 -> 1

Las restantes operaciones afectan directamente al RLO:

- Insertar una entrada binaria
- Invertir una entrada binaria
- SAVE : Cargar resultado lógico (RLO) en el registro RB

1.2 >=1 : Operación lógica O

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando>	BOOL	E, A, M, T, Z, D, L	El operando indica el bit cuyo estado de señal está siendo consultado.

Descripción

La operación **O** lógica sirve para consultar los estados de señal de dos o más operandos especificados en las entradas de un cuadro O.

Si el estado de señal de uno de estos operandos es ”1", la operación da como resultado "1". Si el estado de señal de todos los operandos es ”0" no se cumple la condición exigida por la operación lógica, por lo que el resultado de la misma será "0".

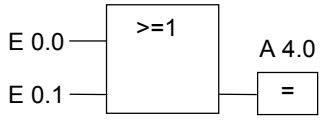
Si la operación O lógica es la primera operación de una cadena lógica, almacena el resultado de la consulta del estado de señal en el bit de resultado lógico (RLO).

Toda operación O lógica que no sea la primera operación de una cadena lógica combina el resultado de la consulta del estado de señal con el valor almacenado en el bit RLO. Esta operación lógica se realiza siguiendo la tabla de verdad O.

Palabra de estado

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

Ejemplo

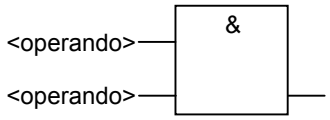


La salida A 4.0 está activada, cuando

- el estado de señal de la entrada E 0.0 O de la entrada E 0.1 es ”1".

1.3 & : Operación lógica Y

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando>	BOOL	E, A, M, T, Z, D, L	El operando indica el bit cuyo estado de señal está siendo consultado.

Descripción

La operación **Y** lógica sirve para consultar los estados de señal de dos o más operandos especificados en las entradas de un cuadro Y.

Si el estado de señal de todos los operandos es ”1", y sólo en este caso, la operación da como resultado "1". Cuando el estado de señal de un operando es ”0" no se cumple la condición exigida por la operación lógica, por lo que el resultado de la operación será "0".

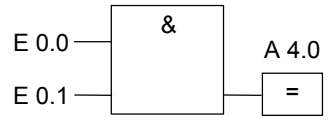
Si la operación lógica Y es la primera operación de una cadena lógica, almacena el resultado de la consulta del estado de señal en el bit de resultado lógico (RLO).

Toda operación Y que no sea la primera operación de una cadena lógica combina el resultado de la consulta del estado de señal con el valor almacenado en el bit RLO. Esta operación lógica se realiza siguiendo la tabla de verdad Y.

Palabra de estado

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

Ejemplo



La salida A 4.0 está activada, cuando

el estado de señal de las entradas E 0.0 Y E 0.1 es ”1".

1.4 Operación O lógica de operaciones Y, y operación Y lógica de operaciones O

Descripción

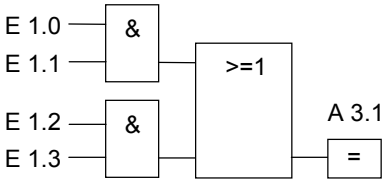
La operación **O lógica de operaciones Y** sirve para consultar el estado de señal de acuerdo con la tabla de verdad O.

En una operación **O lógica de operaciones Y**, el estado de señal es ”1" cuando al menos una operación Y lógica es verdadera.

Palabra de estado

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

Ejemplo



El estado de señal de la salida A 3.1 es ”1" cuando al menos una operación Y es verdadera.

El estado de señal de la salida A 3.1 es ”0" cuando ninguna operación Y es verdadera

Descripción

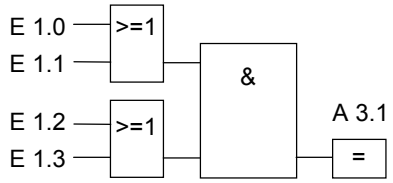
La operación **Y lógica de operaciones O** sirve para consultar el estado de señal de acuerdo con la tabla de verdad Y.

En una operación **O lógica de operaciones O**, el estado de señal es ”1" cuando todas las operaciones O lógicas son verdaderas.

Palabra de estado

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

Ejemplo

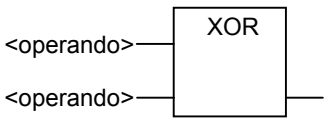


El estado de señal de la salida A 3.1 es ”1"cuando ambas operaciones O son verdaderas.

El estado de señal de la salida A 3.1 es ”0" cuando alguna de las operaciones O no es verdadera.

1.5 XOR : Operación lógica O-exclusiva

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando>	BOOL	E, A, M, T, Z, D, L	El operando indica el bit cuyo estado de señal está siendo consultado.

Descripción

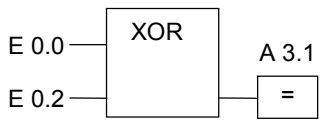
La operación lógica **O-exclusiva** sirve para consultar el estado de señal de acuerdo con la tabla de verdad O-exclusiva.

En una operación lógica **O-exclusiva**, el estado de señal es ”1" cuando el estado de señal de uno de los dos operandos es ”1". En los elementos XOR para consultar más de dos operandos el resultado lógico común es "1" si un número impar de los operandos consultados da el resultado lógico "1".

Palabra de estado

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

Ejemplo



El estado de señal de la salida A 3.1 es ”1" cuando, de forma EXCLUSIVA, el estado de señal es ”1" en la entrada E 0.0 O en la entrada E 0.2.

1.6 Insertar una entrada binaria

Símbolo

<operando>

Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando>	BOOL	E, A, M, T, Z, D, L	El operando indica el bit cuyo estado de señal está siendo consultado.

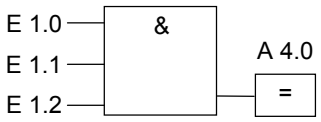
Descripción

La operación **Insertar una entrada binaria** inserta otra entrada binaria en un cuadro de los tipos Y, O u O-exclusiva, detrás de la señal.

Palabra de estado

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

Ejemplo



La salida A 4.0 es ”1" cuando el estado de señal de E 1.0 Y de E 1.1 Y de E 1.2 es ”1".

1.7 Negar entrada binaria

Símbolo

Descripción

La operación **Invertir una entrada binaria** niega el RLO.

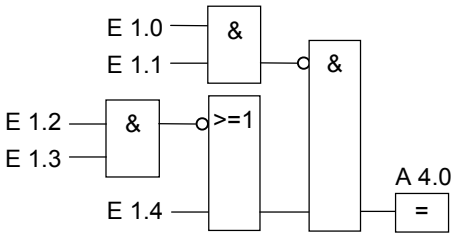
Al invertir (negar) el resultado lógico deberá respetar las siguientes reglas:

- Si se invierte el resultado lógico en la primera entrada de un cuadro O o de un cuadro Y, no se abre ningún paréntesis.
- Si el resultado lógico no es invertido en la primera entrada de un cuadro O, toda la combinación lógica binaria se integra antes de la entrada.
- Si el resultado lógico no es invertido en la primera entrada de un cuadro Y, toda la combinación lógica binaria se integra antes de la entrada.

Palabra de estado

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

Ejemplo

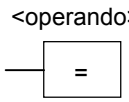


La salida A 4.0 será "1", cuando:

- el estado de señal de E 1.0 Y de E 1.1 NO es ”1"
- Y el estado de señal de E 1.2 Y de E 1.3 NO es ”1"
- O el estado de señal de E 1.4 NO es ”1".

1.8 = : Asignación

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando>	BOOL	E, A, M, D, L	El operando indica el bit al que se asigna el estado de señal de la cadena lógica.

Descripción

La operación **Asignación** suministra el resultado lógico. El cuadro del final de la operación lógica da la señal 1 ó 0 de acuerdo a los criterios siguientes:

- La salida da la señal 1 cuando se cumplen las condiciones de la operación lógica antes del cuadro de salida.
- La salida da la señal 0 cuando no se cumplen las condiciones de la operación lógica antes del cuadro de salida.

La operación lógica FUP asigna el estado de señal a la salida direccionada por la operación (ésto es lo mismo que asignar el estado de señal del bit RLO al operando). Si se cumplen las condiciones de las operaciones lógicas FUP, el estado de señal del cuadro de salida es ”1"; en caso contrario, el estado de señal es ”0".

La operación **Asignación** es afectada por el MCR (Master Control Relay).

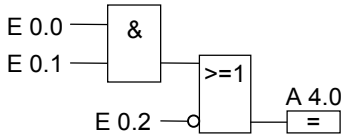
El cuadro Asignación solamente puede posicionarse en el extremo derecho de una cadena lógica. No obstante, es posible utilizar varios cuadros Asignación.

Si quiere crear una asignación inversa realice la operación Invertir una entrada.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	0	X	-	0

Ejemplo

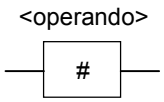


El estado de señal de la salida A 4.0 es ”1" cuando:

- el estado de señal de las entradas E 0.0 Y E 0.1 es ”1",
- O E 0.2 = 0.

1.9 # : Conector

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando>	BOOL	E, A, M, D, *L	El operando indica el bit al que se asigna el RLO.

* Para poder utilizar un operando de la pila de datos locales se tiene que haber declarado el operando en la sección TEMP de la tabla de declaración de variables de un bloque lógico (FC, FB, OB).

Descripción

La operación **Conector** es un elemento de asignación intermedio que almacena el RLO. En concreto, este elemento memoriza la operación lógica de bits del último ramal abierto hasta anterior al elemento de asignación.

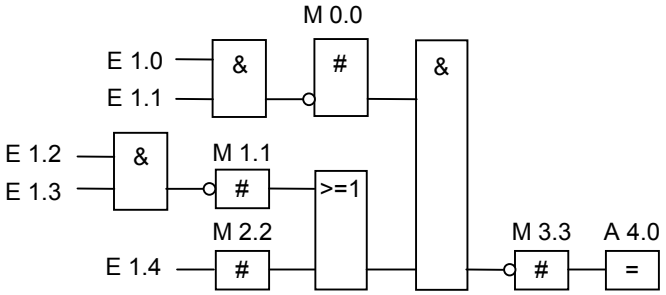
La operación Conector es afectada por el MCR (Master Control Relay).

Para crear un conector inverso invierta la entrada del conector.

Palabra de estado

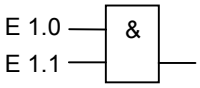
	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	0	X	-	1

Ejemplo

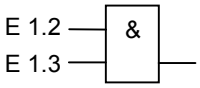


Los conectores memoriza los siguientes resultados lógicos:

M 0.0 memoriza el RLO inverso de



M 1.1 memoriza el RLO inverso de

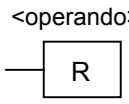


M 2.2 memoriza el RLO de E1.4

M 3.3 memoriza el RLO inverso de la operación de bits en su conjunto.

1.10 R : Desactivar salida

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando>	BOOL	E, A, M, T, Z, D, L	El operando indica qué bit se debe desactivar.

Descripción

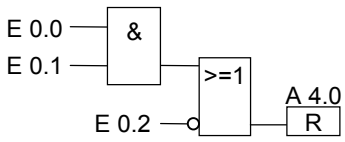
La operación **Desactivar salida** se ejecuta solamente si el RLO = 1. Si el RLO es ”1", la operación pone el operando indicado a ”0". Si el RLO es ”0", la operación no afecta al operando y éste permanece inalterado.

La operación **Desactivar salida** es afectada por el MCR (Master Control Relay).

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	0	X	-	0

Ejemplo



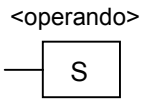
El estado de señal de la salida A 4.0 se pone a ”0" únicamente cuando:

- el estado de señal de las entradas E 0.0 Y E 0.1 es ”1"
- O el estado de señal de la entrada E 0.2 = 0

Si el RLO de la rama = 0, el estado de señal de A 4.0 no cambia.

1.11 S : Activar salida

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando>	BOOL	E, A, M, D, L	El operando indica el bit que se debe activar.

Descripción

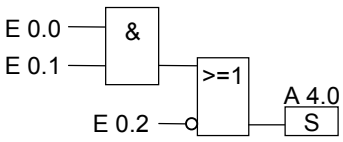
La operación **Activar salida** se ejecuta solamente si RLO = 1. Si el RLO es ”1", la operación pone el operando indicado a 1. Si el RLO es ”0", la operación no afecta al operando y éste permanece inalterado.

La operación **Activar salida** es afectada por el MCR (Master Control Relay).

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	0	X	-	0

Ejemplo



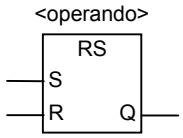
El estado de señal de la salida A 4.0 se pone a ”1" únicamente si:

- el estado de señal de las entradas E 0.0 Y E 0.1 es ”1",
- O el estado de señal de la entrada E 0.2 = 0.

Si el RLO de la rama = 0, el estado de señal de A 4.0 no cambia.

1.12 RS : Flipflop de desactivación/activación

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando>	BOOL	E, A, M, D, L	El operando indica el bit que va a ser activado o desactivado.
S	BOOL	E, A, M, D, L, T, Z	Operación de desactivación habilitada.
R	BOOL	E, A, M, D, L, T, Z	Operación de activación habilitada.
Q	BOOL	E, A, M, D, L	Estado de señal del <operando>.

Descripción

La operación **Flipflop de desactivación/activación** ejecuta las operaciones Activar (S) y Desactivar (R) solamente si el RLO es ”1". Un "0" en el RLO no afecta dichas operaciones y el operando indicado en la operación permanece inalterado.

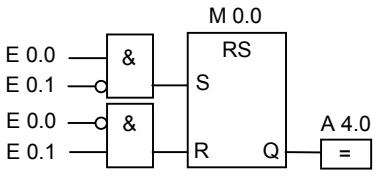
La operación **Flipflop de desactivación/activación** se desactiva si el estado de señal en la entrada R es ”1" y en la entrada S es ”0". En otro caso, si el estado de señal en la entrada R es ”0" y en la entrada S es ”1", el flipflop se activa. Si el RLO es ”1" en ambas entradas, el flipflop se activa.

La operación **Flipflop de desactivación/activación** es afectada por el MCR (Master Control Relay).

Palabra de estado

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

Ejemplo

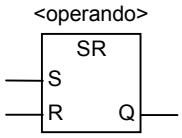


Si E 0.0 = 1 y E 0.1 = 0, se desactiva la marca M 0.0 y la salida A 4.0 es ”0". Si E 0.0 = 0 y E 0.1 = 1, se activa la marca M 0.0 y la salida A 4.0 es ”1".

Si el estado de señal de ambas entradas es ”0" no se produce ningún cambio. Si el estado de señal de ambas entradas es ”1" se impone la operación Activar en razón del orden establecido. M 0.0 se activa y A 4.0 es ”1".

1.13 SR : Flipflop de activación/desactivación

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando>	BOOL	E, A, M, D, L	El operando indica el bit que va a ser activado o desactivado.
S	BOOL	E, A, M, D, L, T, Z	Operación de activación habilitada.
R	BOOL	E, A, M, D, L, T, Z	Operación de desactivación habilitada.
Q	BOOL	E, A, M, D, L	Estado de señal del <operando>.

Descripción

La operación **Flipflop de activación/desactivación** ejecuta las operaciones Activar (S) y Desactivar (R) solamente si el RLO es ”1". Un "0" en el RLO no afecta a dichas operaciones y el operando indicado en la operación permanece inalterado.

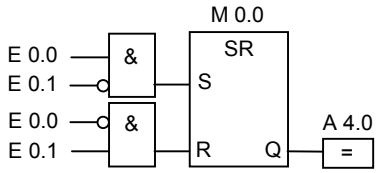
La operación **Flipflop de activación/desactivación** se activa si el estado de señal en la entrada S es ”1" y en la entrada R es ”0". En otro caso, si el estado de señal en la entrada S es ”0" y en la entrada R es ”1", el flipflop se desactiva. Si el RLO es ”1" en ambas entradas, el flipflop se desactiva.

La operación **Flipflop de activación/desactivación** es afectada por el MCR (Master Control Relay).

Palabra de estado

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

Ejemplo

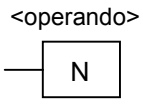


Si E 0.0 = 1 y E 0.1 = 0, se activa la marca M 0.0 y la salida A 4.0 es ”1". Si E 0.0 = 0 y E 0.1 = 1, se desactiva la marca M 0.0 y la salida A 4.0 es ”0".

Si el estado de señal de ambas entradas es ”0" no se produce ningún cambio. Si el estado de señal de ambas entradas es ”1" se impone la operación Desactivar en razón del orden establecido. M 0.0 se desactiva y la salida A 4.0 es ”0".

1.14 N : Detectar flanco negativo (1 -> 0)

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando>	BOOL	E, A, M, D, L	El operando indica la marca de flancos donde se memoriza el anterior RLO.

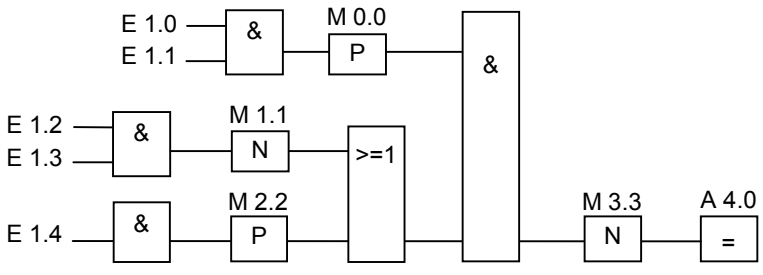
Descripción

La operación **Detectar flanco negativo RLO (1 -> 0)** reconoce un cambio del estado de señal del RLO de "1" a "0" (flanco negativo) y tras la operación lo indica poniendo el RLO a "1". El estado actual de la señal de RLO se compara con el estado de la señal del operando, la marca de flanco. Si el estado de la señal del operando es ”0" y el RLO anterior a la operación "1", tras ésta se pone el RLO a "1" (impulso) y en el resto de los casos, a "0". El RLO anterior a la operación se memoriza en el operando.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	0	X	X	1

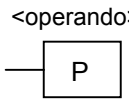
Ejemplo



La marca de flancos M 3.3 memoriza el estado de señal del RLO anterior de la operación de bits en su conjunto.

1.15 P : Detectar flanco positivo (0 -> 1)

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando>	BOOL	E, A, M, D, L	El operando indica la marca de flancos donde se memoriza el anterior RLO.

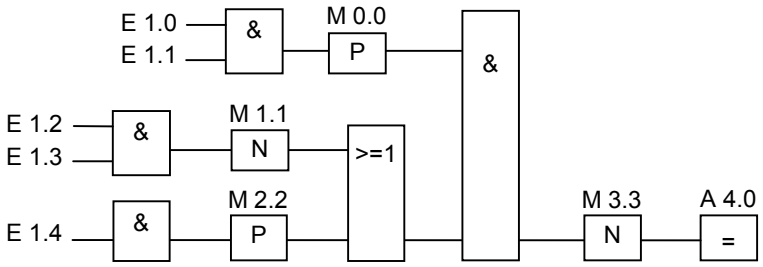
Descripción

La operación **Detectar flanco positivo RLO (0 -> 1)** reconoce un cambio de "0" a "1" en el operando dado y tras la operación lo indica poniendo el RLO a "1". El estado actual de la señal de RLO se compara con el estado de la señal del operando, la marca de flanco. Si el estado de la señal del operando es ”0" y el RLO anterior a la operación "1", tras ésta se pone el RLO a "1" (impulso) y en el resto de los casos, a "0". El RLO anterior a la operación se memoriza en el operando.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	0	X	X	1

Ejemplo



La marca de flancos M 3.3 memoriza el estado de señal del RLO anterior.

1.16 SAVE : Cargar resultado lógico (RLO) en el registro RB

Símbolo

Descripción

La operación **Cargar resultado lógico (RLO) en registro RB** memoriza el RLO en el bit RB de la palabra de estado, sin que el bit de primera consulta /ER se ponga a 0.

Por esta razón, en el siguiente segmento donde haya una Y lógica también se combinará el estado del bit RB.

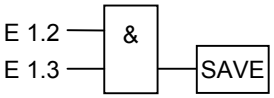
El uso de SAVE con una consulta del bit RB en el mismo bloque o en bloques subordinados no es recomendable, puesto que el bit RB puede ser modificado por numerosas operaciones intercaladas. La operación SAVE resulta especialmente útil antes de salir de un bloque, puesto que con ella la salida ENO (bit RB) se pone al valor del bit RLO, lo cual permite añadir un tratamiento de error a continuación del bloque.

Con la operación **Cargar el resultado lógico en el registro RB** se puede combinar el RLO de un segmento en un bloque subordinado. La instrucción CALL del bloque invocante pone a 0 el bit de primera consulta.

Palabra de estado

	RB	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLO**	**/ER**
se escribe	X	-	-	-	-	-	-	-	-

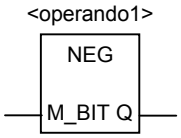
Ejemplo



El resultado lógico (RLO) se memoriza en el registro RB.

1.17 NEG : Detectar flanco de señal 1 -> 0

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando1>	BOOL	E, A, M, D, L	Señal en la que se va a detectar si ha producido un cambio negativo de flanco.
M_BIT	BOOL	A, M, D	El operando indica la marca de flancos que memoriza el estado precedente de señal de M_BIT NEG. Utilizar el área de memoria. Imagen de proceso de las entradas (E) para el M_BIT solamente si este operando no está ya ocupado por un módulo de entrada.
Q	BOOL	E, A, M, D, L	Salida del cambio único de señal.

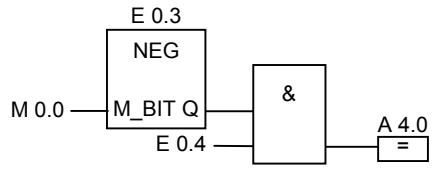
Descripción

La operación **Detectar flanco de señal 1 -> 0** compara el estado de señal del <operando1> con el estado de señal de la consulta precedente, almacenado en el parámetro M_BIT. Si se produce un cambio de "1" a "0", la salida Q será "1"; en el resto de los casos, "0".

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	0	1	X	1

Ejemplo

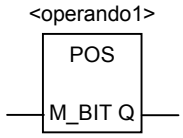


La salida A 4.0 es ”1" cuando:

- la entrada E 0.3 tiene un flanco descendente
- Y el estado de señal de la entrada E 0.4 es ”1".

1.18 POS : Detectar flanco de señal 0 -> 1

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
<Operando1>	BOOL	E, A, M, D, L	Señal en la que se va a detectar si ha producido un cambio positivo de flanco.
M_BIT	BOOL	A, M, D	El operando indica la marca de flancos que memoriza el estado precedente de señal de M_BIT POS. Utilice el área de memoria Imagen de proceso de las entradas (E) para el M_BIT solamente si este operando no está ya ocupado por un módulo de entrada.
Q	BOOL	E, A, M, D, L	Salida del cambio único de señal.

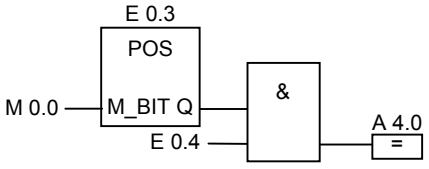
Descripción

La operación **Detectar flanco de señal 0 -> 1** compara el estado de señal del <operando1> con el estado de señal de la consulta precedente, que se ha almacenado en el parámetro M_BIT. Si se produce un cambio de "0" a "1", la salida Q será "1"; en el resto de los casos, "0".

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	0	1	X	1

Ejemplo



La salida A 4.0 es ”1" cuando:

- la entrada E 0.3 tiene un flanco ascendente
- Y el estado de señal de la entrada E 0.4 es ”1".

2 Operaciones de comparación

2.1 Lista de operaciones de comparación

Descripción

Las operaciones comparan las entradas IN1 e IN2 según los tipos de comparación siguientes:

== IN1 es igual a IN2
<> IN1 es diferente a IN2
> IN1 es mayor que IN2
< IN1 es menor que IN2
>= IN1 es mayor o igual a IN2
<= IN1 es menor o igual a IN2

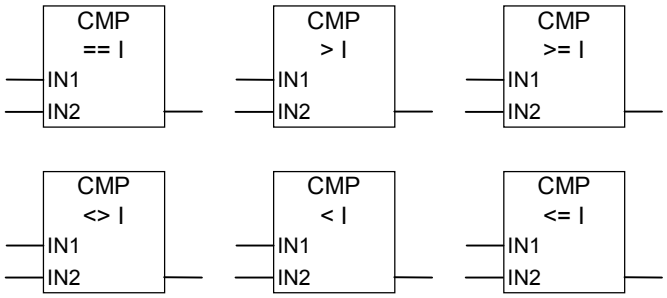
Si la comparación es verdadera, el resultado lógico (RLO) de la función es "1", en otro caso "0". La negación del resultado de comparación no existe ya que éste puede obtenerse con la función de comparación inversa.

Se dispone de las operaciones de comparación siguientes:

- CMP ? I : Comparar enteros
- CMP ? D : Comparar enteros dobles
- CMP ? R : Comparar números en coma flotante

2.2 CMP ? I : Comparar enteros

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
IN1	INT	E, A, M, D, L o constante	Primer valor a comparar
IN2	INT	E, A, M, D, L o constante	Segundo valor a comparar
Salida del cuadro	BOOL	E, A, M, D, L	Resultado de la comparación

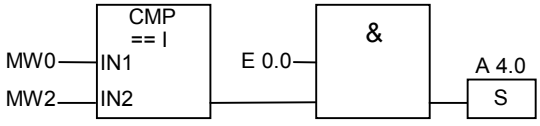
Descripción

La operación **Comparar enteros** ejecuta una operación de comparación cuya base es un entero en coma fija de 16 bits. La operación compara las entradas IN1 e IN2 según el tipo de comparación seleccionado en el cuadro.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	0	-	0	X	X	1

Ejemplo

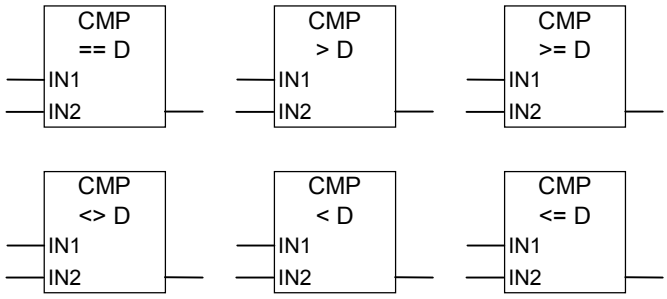


A 4.0 se activa cuando:

- MW0 = MW2
- Y en la entrada E 0.0 el estado de señal es ”1".

2.3 CMP ? D : Comparar enteros dobles

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
IN1	DINT	E, A, M, D, L o constante	Primer valor a comparar
IN2	DINT	E, A, M, D, L o constante	Segundo valor a comparar
Salida del cuadro	BOOL	E, A, M, D, L	Resultado de la comparación

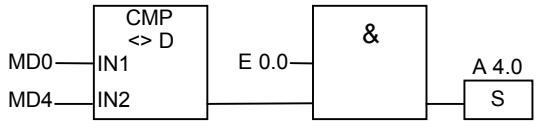
Descripción

La operación **Comparar enteros dobles** ejecuta una operación de comparación cuya base es un entero en coma fija de 32 bits. La operación compara las entradas IN1 e IN2 según el tipo de comparación seleccionado en el cuadro.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	X	X	0	-	0	X	X	1

Ejemplo

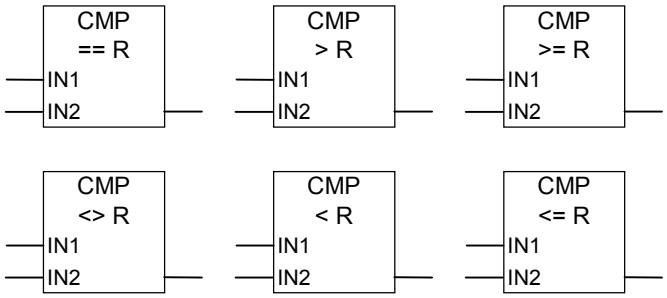


A 4.0 se activa cuando:

- MD0 es diferente de MD4
- Y en la entrada E 0.0 el estado de señal es ”1".

2.4 CMP ? R : Comparar números en coma flotante

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
IN1	REAL	E, A, M, D, L o constante	Primer valor a comparar
IN2	REAL	E, A, M, D, L o constante	Segundo valor a comparar
Salida del cuadro	BOOL	E, A, M, D, L	Resultado de la comparación

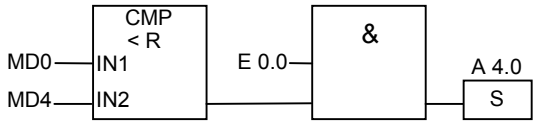
Descripción

La operación **Comparar números en coma flotante** ejecuta una operación de comparación cuya base son número en coma flotante. Esta compara las entradas IN1 e IN2 según el tipo de comparación seleccionado en el cuadro.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	X	X	X	X	0	X	X	1

Ejemplo



A 4.0 se activa cuando:

- MD0 < MD4
- Y en las entradas E 0.0 el estado de señal es ”1"

3 Operaciones de conversión

3.1 Lista de operaciones de conversión

Descripción

Las operaciones de conversión leen el contenido del parámetro IN y lo convierten o le cambian el signo. El resultado se puede recoger en el parámetro OUT. Las siguientes operaciones se utilizan para convertir números decimales codificados en binario y enteros a otros tipos de números:

- BCD_I : Convertir de BCD a entero
- I_BCD : Convertir de entero a BCD
- BCD_DI : Convertir de BCD a entero doble
- I_DI : Convertir de entero a entero doble
- DI_BCD : Convertir de entero doble a BCD
- DI_R : Convertir de entero doble a número en coma flotante

Para formar complementos de números enteros o para cambiar el signo de un número en coma flotante se utilizan las siguientes operaciones:

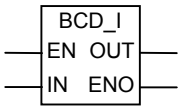
- INV_I : Complemento a 1 de entero
- INV_DI : Complemento a 1 de entero doble
- NEG_I : Complemento a 2 de entero
- NEG_DI : Complemento a 2 de entero doble
- NEG_R : Cambiar el signo de un número en coma flotante

Para convertir un número en coma flotante de 32 bits IEEE 754 en un entero de 32 bits (entero doble) se utilizan las operaciones descritas a continuación. Las operaciones difieren en el método de redondeo.

- ROUND : Redondear a entero doble
- TRUNC : Truncar a entero doble
- CEIL : Redondear número en coma flotante a entero superior
- FLOOR : Redondear número en coma flotante a entero inferior

3.2 BCD_I : Convertir de BCD a entero

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	WORD	E, A, M, D, L o constante	Número en formato BCD
OUT	INT	E, A, M, D, L	Valor entero del número BCD
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

La operación **Convertir de BCD a entero** lee el número de tres dígitos en formato decimal codificado en binario (BCD, ± 999) que contiene el parámetro de entrada IN y convierte este número en un entero. El resultado se emite en el parámetro de salida OUT.

ENO y EN siempre tienen el mismo estado de señal.

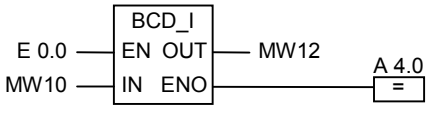
Si una parte del número BCD se encuentra en el área no válida de 10 a 15, al intentar realizar la conversión se produce un error BCDF:

- La CPU pasa a modo STOP. En el búfer de diagnóstico se registra el aviso "BCD Conversion Error" con el número de identificación de evento 2521.
- Se llama al OB 121 si éste está programado.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	1	-	-	-	-	0	1	1	1

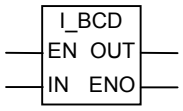
Ejemplo



La conversión se lleva a cabo cuando E 0.0 =1. El contenido de la palabra de marcas MW10 se lee como número BCD de tres dígitos y es convertido en un número entero. El resultado se almacena en MW12. Si se lleva a cabo la conversión A 4.0 = 1 (ENO = EN).

3.3 I_BCD : Convertir de entero a BCD

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	INT	E, A, M, D, L o constante	Número entero
OUT	WORD	E, A, M, D, L	Valor entero del número BCD
ENO	BOOL	E, A, M, D, L	Salida de habilitación

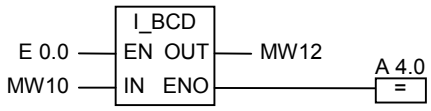
Descripción

La operación **Convertir de entero a BCD** lee el valor entero contenido en el parámetro de entrada IN y convierte este valor en un número de tres dígitos en formato decimal codificado en binario (BCD, ± 999) El resultado se emite en el parámetro de salida OUT. En caso de producirse un desbordamiento, ENO = 0.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	-	-	X	X	0	X	X	1

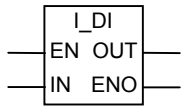
Ejemplo



La conversión se lleva a cabo cuando E 0.0 =1. El contenido de la palabra de marcas MW10 se lee como número entero y es convertido en un número BCD de tres dígitos. El resultado se almacena en MW12. Si se produce un desbordamiento A 4.0 = 1. Si el estado de señal de la entrada EN = 0 (es decir, si no se realiza la conversión), el estado de señal de la salida A 4.0 también será ”0".

3.4 I_DI : Convertir de entero a entero doble

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	INT	E, A, M, D, L o constante	Valor a convertir
OUT	DINT	E, A, M, D, L	Resultado
ENO	BOOL	E, A, M, D, L	Salida de habilitación

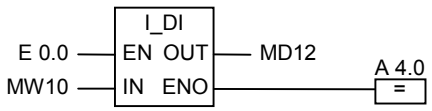
Descripción

La operación **Convertir de entero a entero doble** lee el entero contenido en el parámetro de entrada IN y lo convierte en un entero doble. El resultado se emite en el parámetro de salida OUT. ENO y EN tienen siempre el mismo estado de señal.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	1	-	-	-	-	0	1	1	1

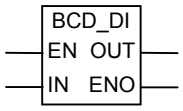
Ejemplo



La conversión se lleva a cabo cuando E 0.0 =1. El contenido de la palabra de marcas MW10 se lee como número entero y es convertido en un número entero doble. El resultado se almacena en MW12. Si se ejecuta la operación A 4.0 = 1 (ENO = EN).

3.5 BCD_DI : Convertir de BCD a entero doble

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	DWORD	E, A, M, D, L o constante	Número en formato BCD
OUT	DINT	E, A, M, D, L	Valor entero doble del número BCD
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

La operación **Convertir de BCD a entero doble** lee el número de siete dígitos en formato decimal codificado en binario (BDC, ± 9 999 999) contenido en el parámetro de entrada IN y lo convierte en un entero doble. El resultado se emite en el parámetro de salida OUT.

ENO y EN siempre tienen el mismo estado de señal.

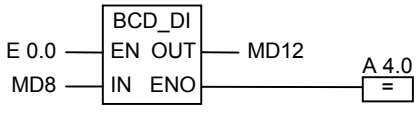
Si una parte del número BCD se encuentra en el área no válida de 10 a 15, al intentar realizar la conversión se produce un error BCDF.

- La CPU pasa a modo STOP. En el búfer de diagnóstico se registra el aviso "BCD Conversion Error" con el número de identificación de evento 2521.
- Se llama al OB 121 si éste está programado.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	1	-	-	-	-	0	1	1	1

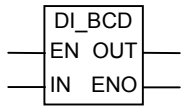
Ejemplo



La conversión se lleva a cabo cuando E 0.0 =1. El contenido de la palabra doble de marcas MD8 se lee como número BCD de siete dígitos y es convertido en un número entero doble. El resultado se almacena en MD12. Si se lleva a cabo la conversión A 4.0 = 1 (ENO = EN).

3.6 DI_BCD : Convertir de entero doble a BCD

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	DINT	E, A, M, D, L o constante	Número entero doble
OUT	DWORD	E, A, M, D, L	Valor BCD del entero doble
ENO	BOOL	E, A, M, D, L	Salida de habilitación

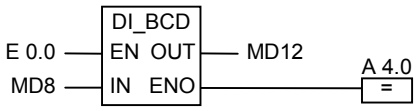
Descripción

La operación **Convertir de entero doble a BCD** lee el entero doble contenido en el parámetro de entrada IN y lo convierte en un número de siete dígitos en formato decimal codificado en binario (BCD, ± 9 999 999). El resultado se emite en el parámetro de salida OUT. En caso de desbordamiento, ENO = 0.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	-	-	X	X	0	X	X	1

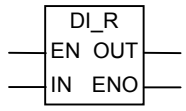
Ejemplo



La conversión se lleva a cabo cuando E 0.0 =1. El contenido de la palabra doble de marcas MD8 se lee como número entero doble y es convertido en un número BCD de siete dígitos. El resultado se almacena en MD12. Si se produce un desbordamiento A 4.0 = 0. Si el estado de señal de la entrada EN = 0 (es decir, si no se realiza la conversión), el estado de señal de la salida A 4.0 también será ”0".

3.7 DI_R : Convertir de entero doble a número en coma flotante

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	DINT	E, A, M, D, L o constante	Valor a convertir
OUT	REAL	E, A, M, D, L	Resultado
ENO	BOOL	E, A, M, D, L	Salida de habilitación

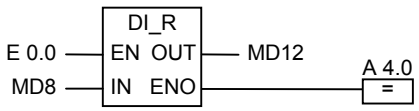
Descripción

La operación **Convertir de entero doble a número en coma flotante** lee el entero doble contenido en el parámetro de entrada IN y convierte este número en un número en coma flotante. El resultado se encuentra en el parámetro de salida OUT. ENO y EN siempre tienen el mismo estado de señal.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	1	-	-	-	-	0	1	1	1

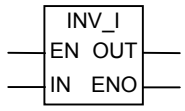
Ejemplo



La conversión se lleva a cabo cuando E 0.0 =1. El contenido de la palabra doble de marcas MD8 se lee como número entero doble y es convertido en un número en coma flotante. El resultado se almacena en MD12. Si no se ejecuta la operación A 4.0 = 0 (ENO = EN).

3.8 INV_I : Complemento a 1 de entero

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	INT	E, A, M, D, L o constante	Valor de entrada
OUT	INT	E, A, M, D, L	Complemento a 1 de entero
ENO	BOOL	E, A, M, D, L	Salida de habilitación

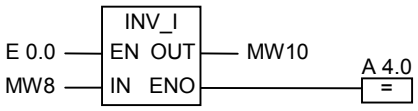
Descripción

La operación **Complemento a 1 de entero** lee el contenido del parámetro de entrada IN y ejecuta la operación lógica O-exclusiva con palabras enmascarada por FFFFH, de modo que cada bit obtiene su valor opuesto. El resultado se emite en el parámetro de salida OUT. ENO y EN siempre tienen el mismo estado de señal.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	1	-	-	-	-	0	1	1	1

Ejemplo



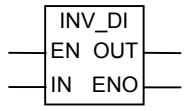
La conversión se lleva a cabo cuando E 0.0 =1. Se invierten todos los bits de MW8:

MW8 = 01000001 10000001 →
MW10 = 10111110 01111110

La conversión no se ejecuta cuando E 0.0 = 0 y A 4.0 = 0 (ENO = EN).

3.9 INV_DI : Complemento a 1 de entero doble

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	DINT	E, A, M, D, L o constante	Valor de entrada
OUT	DINT	E, A, M, D, L	Complemento a 1 de entero doble
ENO	BOOL	E, A, M, D, L	Salida de habilitación

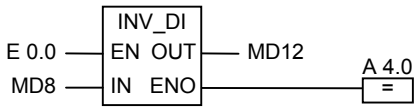
Descripción

La operación **Complemento a 1 de entero doble** lee el contenido del parámetro de entrada IN y ejecuta la operación lógica O-exclusiva con palabras con la plantilla hexadecimal FFFF FFFFH, de modo que se invierte el valor de cada bit. El resultado se emite en el parámetro de salida OUT. ENO y EN siempre tienen el mismo estado de señal.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	1	-	-	-	-	0	1	1	1

Ejemplo



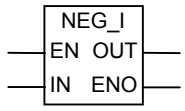
La conversión se lleva a cabo cuando E 0.0 =1. Se invierten todos los bits de la palabra doble de marcas MD8:

MD8 = F0FF FFF0 $\rightarrow$ MD12 = 0F00 000F

La conversión no se lleva a cabo cuando E0.0 = 0 y A 4.0 = 0 (ENO = EN).

3.10 NEG_I : Complemento a 2 de entero

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	INT	E, A, M, D, L o constante	Valor de entrada
OUT	INT	E, A, M, D, L	Complemento a 2 de entero
ENO	BOOL	E, A, M, D, L	Salida de habilitación

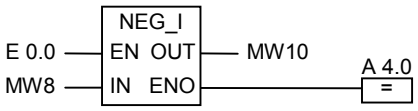
Descripción

La operación **Complemento a 2 de entero** lee el contenido del parámetro de entrada IN e invierte el signo (por ejemplo, de un valor positivo a un valor negativo). El resultado se emite en el parámetro de salida OUT. Si el estado de señal de EN es "0", el estado de señal de ENO es "0". Si el estado de señal de EN es "1" y se produce un desbordamiento, el estado de señal de ENO es "0".

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

Ejemplo



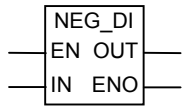
La conversión se lleva a cabo cuando E 0.0 =1. El valor de la palabra de marcas MW8 se emite con el signo invertido en MW10 al parámetro OUT. Ejemplo:

MW8 = +10 $\rightarrow$ MW10 = -10

Cuando EN = 1 y se produce un desbordamiento, ENO = 0 y el estado de señal de A 4.0 es ”0". Si no se lleva a cabo la conversión A 4.0 = 0 (ENO = EN).

3.11 NEG_DI : Complemento a 2 de entero doble

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	DINT	E, A, M, D, L o constante	Valor de entrada
OUT	DINT	E, A, M, D, L	Complemento a 2 de entero doble
ENO	BOOL	E, A, M, D, L	Salida de habilitación

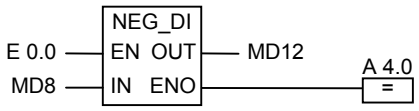
Descripción

La operación **Complemento a 2 de entero doble** lee el contenido del parámetro de entrada IN e invierte el signo (por ejemplo de un valor positivo a un valor negativo). El resultado se emite en el parámetro de salida OUT. Si el estado de señal de EN es "0", el estado de señal de ENO es "0". Si el estado de señal de EN es "1" y se produce un desbordamiento, el estado de señal de ENO es "0".

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

Ejemplo



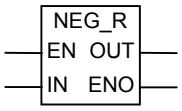
La conversión se lleva a cabo cuando E 0.0 =1. El contenido de la palabra doble de marcas MD8 se emite con el signo invertido en MD12 al parámetro OUT. Ejemplo:

MD8 = + 60.000 → MD12 = - 60.000.

Cuando EN = 1 y se produce un desbordamiento, ENO = 0 y el estado de señal de A 4.0 es ”0". Si no se lleva a cabo la conversión A 4.0 = 0 (ENO = EN).

3.12 NEG_R : Cambiar el signo de un número en coma flotante

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Valor de entrada
OUT	REAL	E, A, M, D, L	El resultado es el valor de entrada invertido
ENO	BOOL	E, A, M, D, L	Salida de habilitación

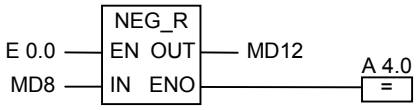
Descripción

La operación **Cambiar el signo de un número en coma flotante** lee el contenido del parámetro de entrada IN e invierte el bit de signo, es decir, la operación cambia el signo del número (por ejemplo de "0" para positivo a "1" para negativo). Los bits del exponente y de la mantisa no se modifican. El resultado se emite en el parámetro de salida OUT. ENO y EN tienen siempre el mismo estado de señal, excepto en el caso siguiente: Si el estado de señal de EN es ”1" y se produce un desbordamiento, el estado de señal de ENO es ”0".

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	-	-	-	-	0	X	X	1

Ejemplo



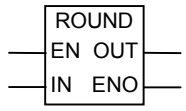
La conversión se lleva a cabo cuando E 0.0 =1. El contenido de la palabra doble de marcas MD8 se emite con signo invertido en MD12 al parámetro OUT. Ejemplo:

MD8 = + 6,234 $\rightarrow$ MD12 = - 6,234

Si no se lleva a cabo la conversión A 4.0 = 0 (ENO = EN).

3.13 ROUND : Redondear a entero doble

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Valor a redondear
OUT	DINT	E, A, M, D, L	IN redondeado al próximo número entero
ENO	BOOL	E, A, M, D, L	Salida de habilitación

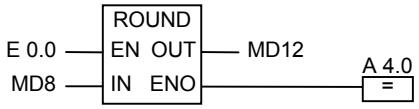
Descripción

La operación **Redondear a entero doble** lee el número en coma flotante contenido en el parámetro de entrada IN y convierte este valor en un entero doble redondeándolo al número entero más próximo. El resultado es el componente entero más próximo (es decir, el entero más próximo). El resultado se encuentra en el parámetro de salida OUT. Si el componente fraccionario = x,5 se devuelve el número par (ejemplo: 2,5 $\rightarrow$ 2, 1,5 $\rightarrow$ 2). Si se produce un desbordamiento, ENO = 0. Si la entrada no es un número en coma flotante, los bits OV y OS toman el valor ”1", y ENO el valor ”0".

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	-	-	X	X	0	X	X	1

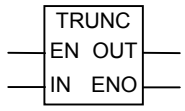
Ejemplo



La conversión se lleva a cabo cuando E 0.0 =1. El contenido de la palabra doble de marcas MD8 se lee como número en coma flotante y es convertido en un en un número entero doble siguiendo el principio ”round to nearest". El resultado de esta operación se almacena en MD12. Si se produce un desbordamiento A 4.0 = 0. Si el estado de señal de la entrada EN = 0 (es decir, si no se lleva a cabo la conversión), el estado de señal de la salida A 4.0 también será ”0".

3.14 TRUNC : Truncar a entero doble

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Valor a redondear
OUT	DINT	E, A, M, D, L	Parte entera del valor IN
ENO	BOOL	E, A, M, D, L	Salida de habilitación

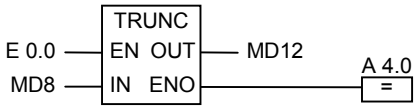
Descripción

La operación **Truncar a entero doble** lee el número en coma flotante contenido en el parámetro de entrada IN y lo convierte en un entero doble (ejemplo: 1,5 -> 1). El resultado es el componente entero del número en coma flotante indicado. El resultado se emite en el parámetro de salida OUT. En caso de desbordamiento, ENO = 0. Si la entrada no es un número en coma flotante, los bits OV y OS toman el valor ”1", y ENO el valor ”0".

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	-	-	X	X	0	X	X	1

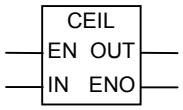
Ejemplo



La conversión se lleva a cabo cuando E 0.0 =1. El contenido de la palabra doble de marcas MD8 se lee como número en coma flotante y es convertido en un en un número entero doble siguiendo el principio ”round to zero". El resultado es la parte entera del número en coma flotante, que se almacena en MD12. Si se produce un desbordamiento A 4.0 = 0. Si el estado de señal de la entrada EN = 0 (es decir, si no se lleva a cabo la conversión), el estado de señal de la salida A 4.0 también será ”0".

3.15 CEIL : Redondear número en coma flotante a entero superior

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Valor a convertir
OUT	DINT	E, A, M, D, L	Resultado
ENO	BOOL	E, A, M, D, L	Salida de habilitación

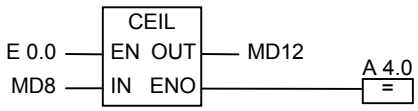
Descripción

La operación **Redondear número en coma flotante a entero superior** lee el número en coma flotante contenido en el parámetro de entrada IN y convierte este número en un entero doble (ejemplo: +1,5 → +2; -1,5 → -1). El resultado es entero más pequeño, mayor o igual al número en coma flotante indicado. El resultado se encuentra en el parámetro de salida OUT. En caso de desbordamiento, ENO = 0. Si la entrada no es un número en coma flotante, los bits OV y OS toman el valor ”1", y ENO el valor ”0".

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	-	-	X	X	0	X	X	1

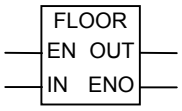
Ejemplo



La conversión se lleva a cabo cuando E 0.0 =1. El contenido de la palabra doble de marcas MD8 se lee como número en coma flotante y es convertido en un en un número entero doble siguiendo el principio ”round to + infinity". El resultado de esta operación se almacena en MD12. Si se produce un desbordamiento A 4.0 = 0. Si el estado de señal de la entrada EN = 0 (es decir, si no se lleva a cabo la conversión), el estado de señal de la salida A 4.0 también será ”0".

3.16 FLOOR : Redondear número en coma flotante a entero inferior

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Valor a convertir
OUT	DINT	E, A, M, D, L	Resultado
ENO	BOOL	E, A, M, D, L	Salida de habilitación

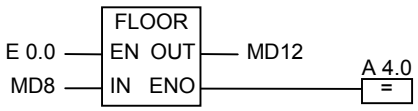
Descripción

La operación **Redondear número en coma flotante a entero inferior** lee el número en coma flotante contenido en el parámetro de entrada IN y convierte este número en un entero doble (ejemplo: +1,5 $\rightarrow$ +1; -1,5 $\rightarrow$ -2). El resultado es el entero más grande, menor o igual al número en coma flotante indicado. El resultado se encuentra en el parámetro de salida OUT. En caso de desbordamiento, ENO = 0. Si la entrada no es un número en coma flotante, los bits OV y OS toman el valor ”1", y ENO el valor ”0".

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	-	-	X	X	0	X	X	1

Ejemplo



La conversión se lleva a cabo cuando E 0.0 =1. El contenido de la palabra doble de marcas MD8 se lee como número en coma flotante y es convertido en un en un número entero doble siguiendo el principio ”round to + infinity". El resultado de esta operación se almacena en MD12. Si se produce un desbordamiento A 4.0 = 0. Si el estado de señal de la entrada EN = 0 (es decir, si no se lleva a cabo la conversión), el estado de señal de la salida A 4.0 también será ”0".

4 Operaciones de contaje

4.1 Lista de operaciones de contaje

Área de memoria

Los contadores tienen reservada un área de memoria en la CPU. Esta área de memoria reserva una palabra de 16 bits para cada contador. FUP asiste 256 contadores. Las operaciones de contaje son las únicas funciones que tienen acceso al área de memoria reservada para contadores.

Valor de contaje

Los bits 0 a 9 de la palabra de contaje contienen el valor de contaje en código binario. El valor fijado por el usuario se transfiere del acumulador al contador al activarse éste. El valor de contaje puede estar comprendido entre 0 y 999.

Dentro de este margen se puede variar dicho valor utilizando las operaciones siguientes:

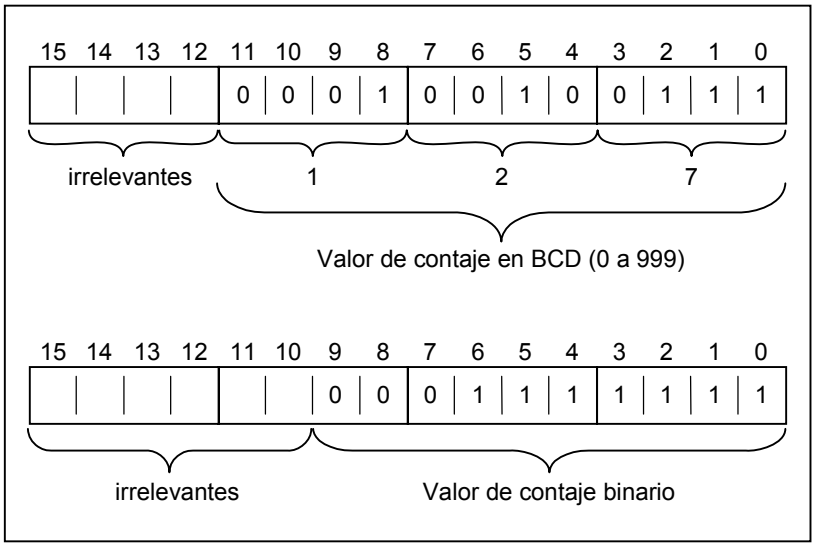
- ZAEHLER : Parametrizar e incrementar / decrementar contador
- Z_VORW : Parametrizar e incrementar contador
- Z_RUECK : Parametrizar y decrementar contador
- SZ : Posicionar el contador en preselección
- ZV : Incrementar contador
- ZR : Decrementar contador

Configuración binaria en el contador

Para poner el contador a un valor determinado hay que introducir un número de 0 a 999, por ejemplo 127, en el siguiente formato: C# 127. C# sirve para indicar el formato decimal codificado en binario.

Los bits 0 a 11 del contador contienen el valor de contaje en formato BCD (formato BCD: cada conjunto de cuatro bits contiene el código binario de un valor decimal).

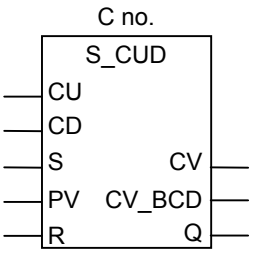
La figura muestra el contenido del contador después de haber cargado el valor de contaje 127 y el contenido de la palabra de contaje después de haber activado el contador.



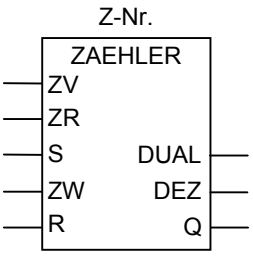
4.2 ZAEHLER : Parametrizar e incrementar / decrementar contador

Símbolo

Inglés



Alemán



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Área de memoria	Descripción
C no.	Z Nr.	COUNTER	Z	Número de identificación del contador. El área depende de la CPU utilizada.
CU	ZV	BOOL	E, A, M, D, L, T, Z	Entrada ZV: Incrementar contador
CD	ZR	BOOL	E, A, M, D, L	Entrada ZR: Decrementar contador
S	S	BOOL	E, A, M, D, L	Entrada para poner un contador a un valor de preselección
PV	ZW	WORD	E, A, M, D, L o constante	Valor comprendido entre 0 y 999 o valor introducido como C#<valor> en formato BCD
R	R	BOOL	E, A, M, D, L,T, Z	Entrada de desactivación
CV	DUAL	WORD	E, A, M, D, L	Valor de contaje actual (formato hexadecimal)
CV_BCD	DEZ	WORD	E, A, M, D, L	Valor de contaje actual (formato BCD)
Q	Q	BOOL	E, A, M, D, L	Estado del contador

Descripción

Cuando se produce un cambio de flanco de "0" a "1" en la entrada S de la operación **Parametrizar e incrementar / decrementar contador**, el contador se inicializa con el valor de contaje ZW. Si dicho valor es menor de 999, al producirse un flanco positivo en la entrada ZV se incrementa en "1" el valor del contador. Si el valor de contaje es mayor que "0", al producirse un flanco positivo en la entrada ZR el valor del contador decrementa en "1". Si ambas entradas tienen un flanco positivo, se procesan las dos operaciones y el valor de contaje se mantiene invariable.

Si se inicializa el contador y las entradas ZV/ZR tienen el RLO = 1, el contador contará una vez en el ciclo **siguiente** aunque no haya ningún cambio de flanco.

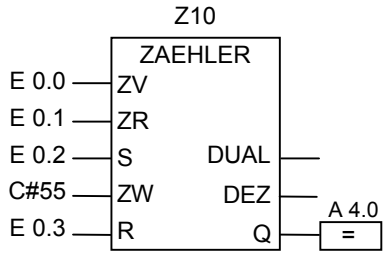
El contador se desactiva cuando aparece un 1 en la entrada R. Al desactivarse el contador el valor de contaje queda ajustado a "0".

La consulta sobre si el estado de la señal de la salida Q es "1" produce un resultado de "1" si el valor de contaje es mayor de "0". La consulta produce "0" si dicho valor es "0".

Palabra de estado

	RB	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLO**	**/ER**
se escribe	-	-	-	-	-	X	X	X	1

Ejemplo



Si el estado de señal de E 0.2 cambia de "0" a "1", el contador Z10 queda ajustado con el valor 55. Si el estado de señal de E 0.0 cambia de "0" a "1" se incrementará en 1 el valor del contador Z10, siempre que no tuviera el valor 999. Si E 0.1 cambia de "0" a "1", el valor de Z10 se decrementa en 1, a no ser que su valor fuera 0. Si E 0.3 cambia de "0" a "1", el valor de Z10 se pone a 0. La salida A 4.0 es "1" cuando el valor de Z10 es distinto de "0".

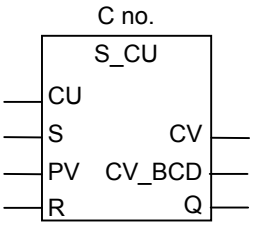
Nota

Procure no utilizar un contador en distintos puntos del programa, sino en uno solo (peligro de errores de contaje).

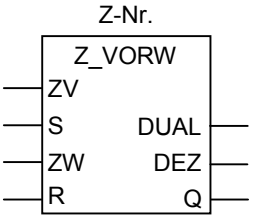
4.3 Z_VORW : Parametrizar e incrementar contador

Símbolo

Inglés



Alemán



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Área de memoria	Descripción
C no.	Z Nr.	COUNTER	Z	Número de identificación del contador. El rango depende de la CPU utilizada.
CU	ZV	BOOL	E, A, M, D, L, T, Z	Entrada ZV: Incrementar contador
S	S	BOOL	E, A, M, D, L	Entrada para poner un contador a un valor de preselección
PV	ZW	WORD	E, A, M, D, L	Valor introducido como C#<valor> en el rango entre 0 y 999
R	R	BOOL	E, A, M, D, L o constante	Entrada de desactivación
CV	DUAL	WORD	E, A, M, D, L,T, Z	Valor de contaje actual (formato hexadecimal)
CV_BCD	DEZ	WORD	E, A, M, D, L	Valor de contaje actual (formato BCD)
Q	Q	BOOL	E, A, M, D, L	Estado del contador

Descripción

Al producirse un cambio de flanco de "0" a "1" en la entrada S de la operación **Parametrizar e incrementar contador**, el contador es inicializado con el valor de contaje ZW. Si dicho valor es menor de 999, al producirse un flanco positivo en la entrada ZV se incrementa en 1 el valor del contador.

Si se inicializa el contador y las entradas ZV tienen el RLO = 1, el contador contará una vez en el ciclo **siguiente** aunque no haya ningún cambio de flanco.

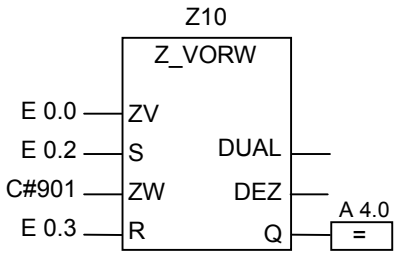
El contador se desactiva cuando aparece un 1 en la entrada R. La desactivación del contador pone el valor de contaje a 0.

La consulta sobre si el estado de la señal de la salida Q es "1" produce un resultado de "1" si el valor de contaje es mayor de "0". La consulta da "0" si dicho valor es "0".

Palabra de estado

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

Ejemplo



Si el estado de señal de E 0.2 cambia de "0" a "1", se ajusta el valor 901 para el contador Z10. Si el estado de señal de E 0.0 cambia de "0" a "1" se incrementa en "1" el valor del contador Z10, siempre que éste no tuviera el valor 999. Si E 0.3 cambia de "0" a "1", el valor de contaje de Z10 se pone a "0". La salida A 4.0 es "1" cuando el valor de Z10 es distinto de "0".

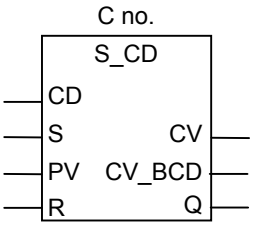
Nota

Procure no utilizar un contador en distintos puntos del programa, sino en uno solo (peligro de errores de contaje).

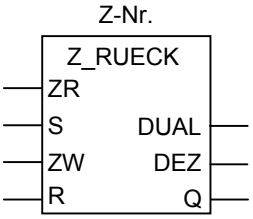
4.4 Z_RUECK : Parametrizar y decrementar contador

Símbolo

Inglés



Alemán



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Área de memoria	Descripción
C no.	Z Nr.	COUNTER	Z	Número de identificación del contador. El rango depende de la CPU utilizada.
CD	ZR	BOOL	E, A, M, D, L	Entrada ZR: Decrementar contador
S	S	BOOL	E, A, M, D, L, T, Z	Entrada para poner un contador a un valor de preselección
PV	ZW	WORD	E, A, M, D, L o constante	Valor introducido como C#<valor> en el rango entre 0 y 999
R	R	BOOL	E, A, M, D, L, T, Z	Entrada de desactivación
CV	DUAL	WORD	E, A, M, D, L	Valor de contaje actual (formato hexadecimal)
CV_BCD	DEZ	WORD	E, A, M, D, L	Valor de contaje actual (formato BCD)
Q	Q	BOOL	E, A, M, D, L	Estado del contador

Descripción

Al producirse un cambio de flanco de "0" a "1" en la entrada S de la operación **Parametrizar y decrementar contador**, el contador es inicializado con el valor de contaje ZW. Si dicho valor es menor de 999, al producirse un flanco positivo en la entrada ZR se decrementa en 1 el valor del contador.

Si se inicializa el contador y las entradas ZR tienen el RLO = 1, el contador contará una vez en el ciclo **siguiente** aunque no haya ningún cambio de flanco.

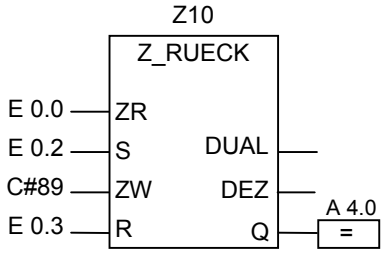
El contador se desactiva cuando aparece un 1 en la entrada R. La desactivación del contador pone el valor de contaje a 0.

La consulta sobre si el estado de la señal de la salida Q es "1" produce un resultado de "1" si el valor de contaje es mayor de "0". La consulta da "0" si dicho valor es "0".

Palabra de estado

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

Ejemplo



Si el estado de señal de E 0.2 cambia de "0" a "1", se ajusta el valor 89 para el contador Z10. Si el estado de señal de E 0.0 cambia de "0" a "1" se decrementa en "1" el valor del contador Z10, siempre que éste no tuviera el valor "0". Si E 0.3 cambia de "0" a "1", el valor de contaje de Z10 se pone a "0".

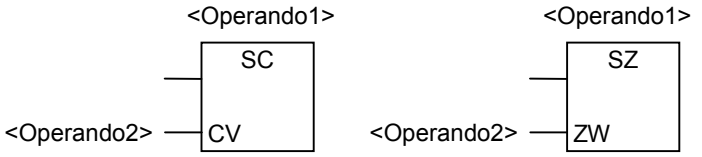
Nota

Procure no utilizar un contador en distintos puntos del programa, sino en uno solo (peligro de errores de contaje).

4.5 SZ : Posicionar el contador en preselección

Símbolo

Inglés **Alemán**



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Área de memoria	Descripción
Nº	Nº	COUNTER	Z	El operando1 indica el número del contador que se debe preajustar con un valor determinado.
CV	ZW	WORD	E, A, M, D, L o constante	El valor a preajustar (operando2) puede encontrarse entre 0 y 999. Al introducir una constante tiene que indicarse C#, p.ej., C#100, antes del valor que indica el formato BCD.

Descripción

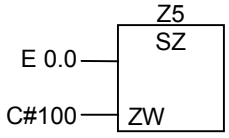
Con la operación **Posicionar el contador en preselección** se asigna un preajuste al contador definido. Esta operación sólo se ejecuta cuando el RLO dispone de un flanco ascendente (cambio de "0" a "1" en el RLO).

El cuadro **Posicionar el contador en preselección** sólo se puede disponer en el extremo derecho de la cadena de conexión. Sin embargo, puede utilizar varios cuadros **Posicionar el contador en preselección**.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	0	-	-	0

Ejemplo



El contador Z5 se preajusta con el valor 100 cuando el estado de señal de E 0.0 cambia de "0" a "1" (flanco positivo en el RLO). C# indica que se introduce un valor en el formato BCD.

Si no se dispone de ningún flanco positivo, el valor del contador Z5 no se modificará.

4.6 ZV : Incrementar contador

Símbolo

Inglés **Alemán**

Parámetro	Tipo de datos	Área de memoria	Descripción
Nº	COUNTER	Z	El operando indica el número del contador que se debe incrementar.

Descripción

La operación **Incrementar contador** incrementa el valor de un contador determinado en "1", siempre y cuando el RLO disponga de un flanco positivo (cambio de "0" a "1") y el valor del contador sea menor que 999. Si el RLO no dispone de ningún flanco positivo o el contador ya ha alcanzado el valor 999, el contador no se incrementará.

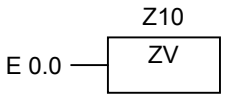
La operación **Posicionar el contador en preselección** posiciona el valor del contador.

El cuadro **Incrementar contador** sólo se puede disponer en el extremo derecho de la cadena de conexión. Sin embargo, puede utilizar varios cuadros **Incrementar contador**.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	0	-	-	0

Ejemplo



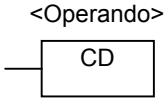
Si el estado de señal de E 0.0 cambia de "0" a "1" (flanco positivo en el RLO), el valor del contador Z10 se incrementa en "1" (a no ser que el valor de Z10 sea igual a 999).

Si no se dispone de flanco positivo, el valor del contador Z10 no se modificará.

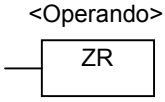
4.7 ZR : Decrementar contador

Símbolo

Inglés



Alemán



Parámetro	Tipo de datos	Área de memoria	Descripción
Nº	COUNTER	Z	El operando indica el número del contador que se debe decrementar.

Descripción

La operación **Decrementar contador** decrementa el valor de un contador determinado en "1", si el RLO dispone de un flanco positivo (cambio de "0" a "1") y el valor del contador es mayor que "0". Si el RLO no dispone de flanco positivo o el contador ya ha alcanzado el valor "0", el contador no se decrementará.

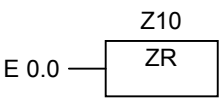
La operación **Posicionar el contador en preselección** posiciona el valor del contador.

El cuadro **Decrementar contador** sólo se puede disponer en el extremo derecho de la cadena de conexión. Sin embargo, puede utilizar varios cuadros **Decrementar contador**.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	0	-	-	0

Ejemplo



Si el estado de señal de E 0.0 cambia de "0" a "1" (flanco positivo en el RLO), el valor del contador Z10 se decrementará en "1" (a no ser que el valor de Z10 sea igual a "0").

Si no se dispone de flanco positivo, el valor del contador Z10 no se modificará.

5 Operaciones con bloques de datos

5.1 OPN : Abrir bloque de datos

Símbolo

<número del DB> o
<número del DI>

Parámetro	Tipo de datos	Area de memoria	Descripción
Número del DB o DI	-	-	El área de valores del DB o DI depende de la CPU utilizada

Descripción

La operación **Abrir bloque de datos** abre un bloque de datos como bloque de datos global (DB) o como bloque de datos de instancia (DI). El número del bloque de datos se transfiere al registro del DB o del DI. Los comandos DB y DI subsiguientes acceden a los correspondientes bloques en función del contenido que tenga el registro.

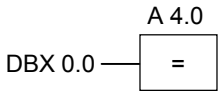
Palabra de estado

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

Ejemplo

Segmento 1

Segmento 2



DB10 es el bloque de datos abierto actualmente. Por ello, la consulta en DBX.0 se dirige al bit 0 del byte de datos 0 del DB10. El estado de señal de este bit se asigna a la salida A 4.0.

6 Operaciones de salto

6.1 Lista de operaciones de salto

Descripción

Esta operación la puede aplicar en todos los bloques lógicos, p. ej., en bloques de organización (OB), en bloques de función (FB) y en funciones (Fcs).

Se dispone de las operaciones de salto siguientes:

- JMP : Salto incondicionado
- JMP : Salto condicionado a 1 en el bloque
- JMPN : Salto condicionado a 0

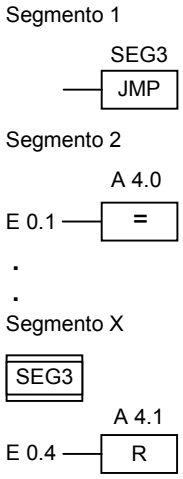
Meta como operando

El operando de una operación de salto es una meta. La meta indica el destino a donde se desea saltar en el programa.

La meta se introduce encima del cuadro JMP. Una meta se compone de cuatro caracteres como máximo. El primer carácter ha de ser una letra alfabética; los restantes caracteres pueden ser letras o números (p.ej. SEG3).

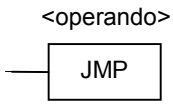
Meta como destino

La meta de destino ha de encontrarse siempre al principio de un segmento. Para introducirla hay que seleccionar LABEL en el cuadro FUP. En seguida aparece un cuadro vacío. Introducir en el cuadro el nombre de la meta.



6.2 JMP : Salto incondicionado

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
Nombre de meta	-	-	El operando señala la meta a la cual se salta sin condiciones.

Descripción

La operación **Salto incondicionado** equivale a una operación "Saltar a meta". No se ejecuta ninguna de las operaciones entre la operación de salto y la meta.

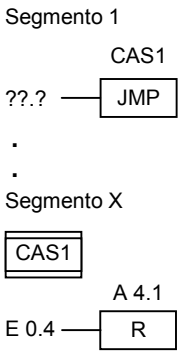
Esta operación la puede aplicar en todos los bloques lógicos, p. ej., en bloques de organización (OB), en bloques de función (FB) y en funciones (Fcs).

El cuadro **Salto incondicionado** no debe tener ninguna combinación lógica previa.

Palabra de estado

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

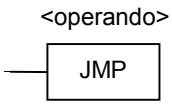
Ejemplo



El salto se ejecuta siempre. No se ejecuta ninguna de las operaciones entre la operación de salto y la meta.

6.3 JMP : Salto condicionado a 1 en el bloque

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
Nombre de meta	-	-	El operando señala la meta a la cual se salta cuando RLO = 1.

Descripción

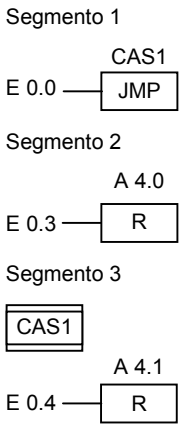
La operación **Salto condicionado a 1** equivale a una operación **Saltar a meta** cuando el RLO es 1. Para esta operación se usan los elementos FUP de **salto incondicionado**, aunque con las operaciones lógicas precedentes. El **salto condicionado** sólo se ejecutará cuando el RLO de dicha operación lógica sea 1. No se ejecutará ninguna de las operaciones entre la operación de salto y la meta.

Esta operación la puede aplicar en todos los bloques lógicos, p. ej., en bloques de organización (OB), en bloques de función (FB) y en funciones (Fcs).

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	0	1	1	0

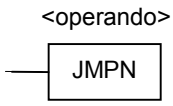
Ejemplo



Si el estado de señal de la entrada E 0.0 es "1" se ejecuta el salto a la meta CAS1. La operación que pone a "0" la salida A 4.0 no se ejecuta, incluso cuando el estado de señal de la entrada E 0.3 = 1.

6.4 JMPN : Salto condicionado a 0

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
Nombre de meta	-	-	El operando señala la meta a la cual se salta cuando RLO = 0.

Descripción

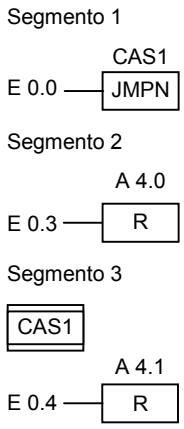
La operación **Salto condicionado a 0** equivale a una operación "Saltar a meta" que se ejecuta si RLO = 0.

Esta operación la puede aplicar en todos los bloques lógicos, p. ej., en bloques de organización (OB), en bloques de función (FB) y en funciones (FCs).

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	0	1	1	0

Ejemplo



Si el estado de señal de la entrada E 0.0 es 1 se ejecuta el salto a la meta CAS1. La operación que pone a 0 la salida A 4.0 no se ejecuta, incluso cuando el estado de señal de la entrada E 0.3 = 1.

No se ejecuta ninguna de las operaciones entre la operación de salto y la meta.

6.5 LABEL : Meta del salto

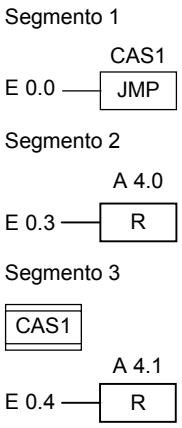
Símbolo

Descripción

La meta del salto marca la meta de una operación de salto. Una meta se compone de cuatro caracteres como máximo. El primer carácter ha de ser una letra alfabética; los restantes caracteres pueden ser letras o números (p.ej. SEG3).

Para cada salto condicionado (**JMP** o **JMPN**) tiene que haber siempre una meta del salto (LABEL).

Ejemplo



Si el estado de señal de la entrada E 0.0 es 1 se ejecuta el salto a la meta CAS1.

A causa del salto, la operación "poner a 0 la salida" no se ejecuta en la salida A 4.0, incluso cuando E 0.3 = 1.

7 Operaciones aritméticas con enteros

7.1 Lista de operaciones aritméticas con enteros

Descripción

Las operaciones aritméticas con enteros sirven para ejecutar las siguientes operaciones aritméticas con **dos** enteros (16 y 32 bits):

- ADD_I : Sumar enteros
- SUB_I : Restar enteros
- MUL_I : Multiplicar enteros
- DIV_I : Dividir enteros
- ADD_DI : Sumar enteros dobles
- SUB_DI : Restar enteros dobles
- MUL_DI : Multiplicar enteros dobles
- DIV_DI : Dividir enteros dobles
- MOD_DI : Obtener el resto de división de enteros dobles

Evaluar los bits de la palabra de estado en operaciones en coma fija

7.2 Evaluar los bits de la palabra de estado en operaciones en coma fija

Descripción

Las operaciones aritméticas básicas influyen sobre los siguientes bits de la palabra de datos:

- A1 y A0
- OV
- OS

Las tablas siguientes muestran el estado de señal de los bits de la palabra de estado para los resultados de las operaciones con números en coma fija (16 bit, 32 bit).

Margen válido	**A1**	**A0**	**OV**	**OS**
0 (cero)	0	0	0	*
enteros: -32 768 < =resultado < 0 (número negativo) enteros dobles: -2 147 483 648 <=resultado < 0 (número negativo)	0	1	0	*
enteros: 32 767 > =resultado > 0 (número positivo) enteros dobles: 2 147 483 647 > = resultado > 0 (número positivo)	1	0	0	*

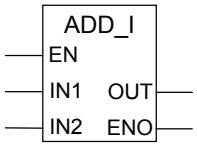
* El bit OS no se ve influido por el resultado de la operación.

Margen no válido	**A1**	**A0**	**OV**	**OS**
Desbordamiento negativo en la suma enteros: resultado = -65536 enteros dobles: resultado = -4 294 967 296	0	0	1	1
Desbordamiento negativo en la multiplicación enteros: resultado < -32 768 (número negativo) enteros dobles: resultado < -2 147 483 648 (número negativo)	0	1	1	1
Desbordamiento positivo en la suma, resta enteros: resultado > 32 767 (número positivo) enteros dobles: resultado > 2 147 483 647 (número positivo)	0	1	1	1
Desbordamiento positivo en la multiplicación, división enteros: resultado > 32 767 (número positivo) enteros dobles: resultado > 2 147 483 647 (número positivo)	1	0	1	1
Desbordamiento negativo en la suma, resta enteros: resultado < -32 768 (número negativo) enteros dobles: resultado < -2 147 483 648 (número negativo)	1	0	1	1
División por cero	1	1	1	1

Operación	**A1**	**A0**	**OV**	**OS**
+D: resultado = -4 294 967 296	0	0	1	1
/D o MOD: división por cero	1	1	1	1

7.3 ADD_I : Sumar enteros

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	INT	E, A, M, D, L o constante	Primer valor a sumar (= primer sumando)
IN2	INT	E, A, M, D, L o constante	Segundo valor a sumar (= segundo sumando)
OUT	INT	E, A, M, D, L	Resultado de la suma
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

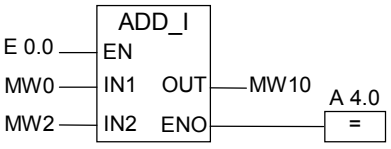
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Sumar enteros**. Esta operación suma las entradas IN1 e IN2. El resultado se determina consultando OUT. Si el resultado se encuentra fuera del área válida para enteros, los bits OV y OS tienen el valor "1" y ENO el valor "0".

Consulte también Evaluar los bits de la palabra de estado en operaciones en coma fija.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

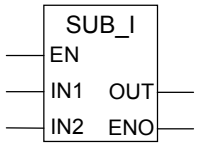
Ejemplo



El cuadro ADD_I se activa cuando E 0.0 =1. El resultado de la suma de MW0 + MW2 se almacena en la palabra de marcas MW10. Si el resultado queda fuera del área permitida para números enteros, o si el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0" y la operación no se realiza.

7.4 SUB_I : Restar enteros

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	INT	E, A, M, D, L o constante	Primer valor (del que se resta) (= minuendo)
IN2	INT	E, A, M, D, L o constante	Valor a restar del primer valor (= substraendo)
OUT	INT	E, A, M, D, L	Resultado de la sustracción (= diferencia)
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

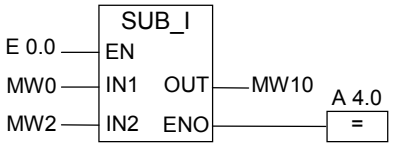
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Restar enteros**. Esta operación resta la entrada IN2 de IN1. El resultado puede determinarse consultando OUT. Si el resultado se encuentra fuera del área válida para enteros, los bits OV y OS tienen el valor "1" y ENO el valor "0".

Consulte también Evaluar los bits de la palabra de estado en operaciones en coma fija.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

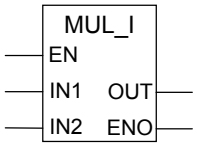
Ejemplo



El cuadro SUB_I se activa cuando E 0.0 = 1. El resultado de la resta de MW0 - MW2 se almacena en la palabra de marcas MW10. Si el resultado queda fuera del área permitida para números enteros, o si el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0" y la operación no se realiza.

7.5 MUL_I : Multiplicar enteros

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	INT	E, A, M, D, L o constante	Primer valor a multiplicar (= multiplicando)
IN2	INT	E, A, M, D, L o constante	Segundo valor a multiplicar (= multiplicando)
OUT	INT	E, A, M, D, L	Resultado de la multiplicación (= producto)
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

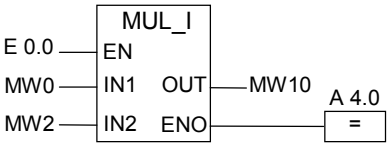
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Multiplicar enteros**. Esta operación multiplica las entradas IN1 e IN2. El resultado es un entero doble y puede determinarse consultando OUT. Si el resultado se encuentra fuera del área válida para enteros de 16 bits, los bits OV y OS tienen el valor "1" y ENO el valor "0".

Consulte también Evaluar los bits de la palabra de estado en operaciones en coma fija.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

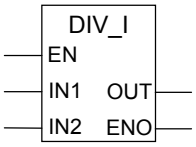
Ejemplo



El cuadro MUL_I se activa cuando E 0.0 = 1. El resultado de la multiplicación de MW0 x MW2 se almacena en la palabra de marcas MW10. Si el resultado queda fuera del área permitida para números enteros (16 bits), o si el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0" y la operación no se realiza.

7.6 DIV_I : Dividir enteros

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	INT	E, A, M, D, L o constante	Dividendo
IN2	INT	E, A, M, D, L o constante	Divisor
OUT	INT	E, A, M, D, L	Resultado de la división (= cociente)
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

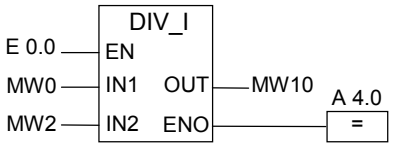
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Dividir enteros**. Esta operación divide la entrada IN1 por IN2. El cociente (fracción entera) puede determinarse consultando O. El resto no puede ser determinado. Si el cociente se encuentra fuera del área válida para enteros, los bits OV y OS tienen el valor "1" y ENO el valor "0".

Consulte también Evaluar los bits de la palabra de estado en operaciones en coma fija.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

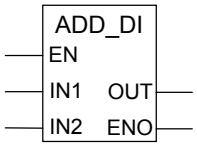
Ejemplo



El cuadro DIV_I se activa cuando E 0.0 = 1. El cociente de la división de MW0 entre MW2 se almacena en la palabra de marcas MW10. Si el cociente queda fuera del área permitida para números enteros, o si el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0" y la operación no se realiza.

7.7 ADD_DI : Sumar enteros dobles

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	DINT	E, A, M, D, L o constante	Primer valor a sumar (= primer sumando)
IN2	DINT	E, A, M, D, L o constante	Segundo valor a sumar (= segundo sumando)
OUT	DINT	E, A, M, D, L	Resultado de la suma
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

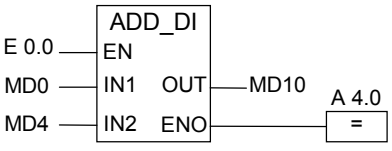
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Sumar enteros dobles**. Esta operación suma las entradas IN1 e IN2. El resultado se determina consultando OUT. Si el resultado se encuentra fuera del área válida para enteros dobles, los bits OV y OS tienen el valor "1" y ENO el valor "0".

Consulte también Evaluar los bits de la palabra de estado en operaciones en coma fija.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

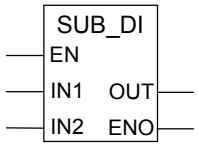
Ejemplo



El cuadro ADD_DI se activa cuando la entrada E 0.0 = 1. El resultado de la suma de MD0 + MD4 se deposita en la palabra doble de marcas MD10. Si el resultado se encuentra fuera del margen permitida para números enteros dobles o el estado de señal de la entrada E 0.0 = 0, se le asignará la salida A 4.0 señal "0".

7.8 SUB_DI : Restar enteros dobles

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	DINT	E, A, M, D, L o constante	Primer valor (del que se resta) (= minuendo)
IN2	DINT	E, A, M, D, L o constante	Valor a restar del primer valor (= substraendo)
OUT	DINT	E, A, M, D, L	Resultado de la sustracción (= diferencia)
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

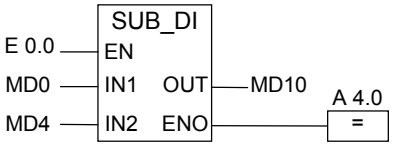
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Restar enteros dobles**. Esta operación resta la entrada IN2 de IN1. El resultado puede determinarse consultando OUT. Si el resultado se encuentra fuera del área válida para enteros dobles, los bits OV y OS tienen el valor "1" y ENO el valor "0".

Consulte también Evaluar los bits de la palabra de estado en operaciones en coma fija.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

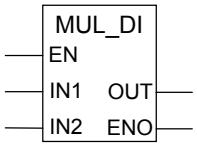
Ejemplo



El cuadro SUB_DI se activa cuando E 0.0 = 1. El resultado de la resta de MD0 - MD4 se almacena en la palabra doble de marcas MD10. Si el resultado queda fuera del área permitida para números enteros dobles, o si el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0" y la operación no se realiza.

7.9 MUL_DI : Multiplicar enteros dobles

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	DINT	E, A, M, D, L o constante	Primer valor a multiplicar (= multiplicando)
IN2	DINT	E, A, M, D, L o constante	Segundo valor a multiplicar (= multiplicando)
OUT	DINT	E, A, M, D, L	Resultado de la multiplicación (= producto)
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

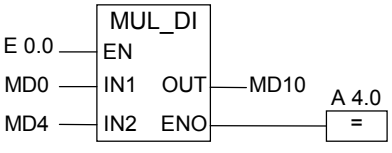
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Multiplicar enteros dobles**. Esta operación multiplica las entradas IN1 e IN2. El resultado puede determinarse consultando OUT. Si el resultado se encuentra fuera del área válida para enteros dobles, los bits OV y OS tienen el valor "1" y ENO el valor "0".

Consulte también Evaluar los bits de la palabra de estado en operaciones en coma fija.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

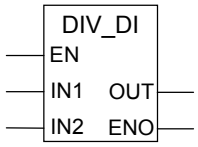
Ejemplo



El cuadro MUL_DI se activa cuando E 0.0 = 1. El resultado de la multiplicación de MD0 x MD4 se almacena en la palabra doble de marcas MD10. Si el resultado queda fuera del área permitida para números enteros dobles, o si el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0" y la operación no se realiza.

7.10 DIV_DI : Dividir enteros dobles

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	DINT	E, A, M, D, L o constante	Dividendo
IN2	DINT	E, A, M, D, L o constante	Divisor
OUT	DINT	E, A, M, D, L	Resultado de la división (= cociente)
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

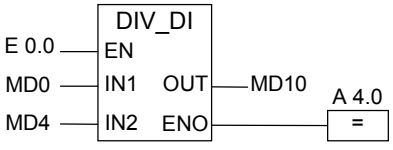
Un estado de señal de ”1” en la entrada de habilitación (EN) activa la operación **Dividir enteros dobles**. Esta operación divide la entrada IN1 por IN2. El cociente (fracción entera) puede determinarse consultando OUT. La operación Dividir enteros dobles almacena el cociente como valor de 32 bits en formato DINT. Esta operación no produce ningún resto. Si el cociente se encuentra fuera del área válida para enteros dobles, los bits OV y OS tienen el valor ”1” y ENO el valor ”0”.

Consulte también Evaluar los bits de la palabra de estado en operaciones en coma fija.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

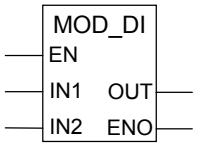
Ejemplo



El cuadro DIV_DI se activa cuando E 0.0 = 1. El cociente de la división de MD0 entre MD4 se almacena en la palabra doble de marcas MD10. Si el cociente queda fuera del área permitida para números enteros dobles, o si el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0” y la operación no se realiza.

7.11 MOD_DI : Obtener el resto de división de enteros dobles

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	DINT	E, A, M, D, L o constante	Dividendo
IN2	DINT	E, A, M, D, L o constante	Divisor
OUT	DINT	E, A, M, D, L	Resto de la división
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

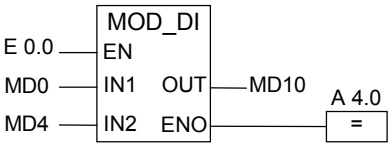
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Obtener el resto de división de enteros dobles**. Esta operación divide la entrada IN1 por IN2. El resto (fracción) puede determinarse consultando OUT. Si el resultado se encuentra fuera del área válida para enteros dobles, los bits OV y OS tienen el valor "1 "y ENO el valor "0".

Consulte también Evaluar los bits de la palabra de estado en operaciones en coma fija.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

Ejemplo



El cuadro MOD_DI se activa cuando E 0.0 = 1. El resto de la división de MD0 entre MD4 se almacena en la palabra doble de marcas MD10. Si el resultado queda fuera del área permitida para números enteros dobles, o si el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0" y la operación no se realiza.

8 Operaciones aritméticas con números en coma flotante

8.1 Lista de operaciones aritméticas con números en coma flotante

Descripción

Los números de 32 bits IEEE en coma flotante pertenecen al tipo de datos denominado ”REAL”. Las operaciones aritméticas con números en coma flotante sirven para ejecutar las siguientes operaciones aritméticas con **dos** números en coma flotante IEEE de 32 bits:

- ADD_R : Sumar números en coma flotante
- SUB_R : Restar números en coma flotante
- MUL_R : Multiplicar números en coma flotante
- DIV_R : Dividir números en coma flotante

Con las operaciones aritméticas de números en coma flotante se pueden ejecutar las siguientes funciones con **un** número en coma flotante (32 bit, IEEE 754):

- ABS : Calcular el valor absoluto de un número en coma flotante
- SQR : Calcular el cuadrado de un número en coma flotante
- SQRT : Calcular la raíz cuadrada de un número en coma flotante
- EXP : Calcular el valor exponencial de un número en coma flotante
- LN : Calcular el logaritmo natural de un número en coma flotante
- Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante
  - seno (SIN) y arcoseno (ASIN)
  - coseno (COS) y arcocoseno (ACOS)
  - tangente (TAN) y tangente (ATAN)

Consulte también Evaluar los bits de la palabra de estado en operaciones aritméticas

8.2 Evaluar los bits de la palabra de estado en operaciones en coma flotante

Descripción

Las operaciones aritméticas básicas afectan a los siguientes bits de la palabra de estado:

- A1 y A0
- OV
- OS

Las tablas siguientes muestran el estado de señal de los bits de la palabra de estado para los resultados de operaciones con números en coma flotante (32 bits).

Margen válido	**A1**	**A0**	**OV**	**OS**
+0, -0 (Cero)	0	0	0	*
-3.402823E+38 < Resultado < -1.175494E-38 (número negativo)	0	1	0	*
+1.175494E-38 < Resultado < +3.402823E+38 (número positivo)	1	0	0	*

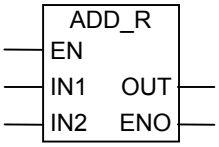
* El bit OS no es afectado por el resultado de la operación.

Margen no inválido	**A1**	**A0**	**OV**	**OS**
Desbordamiento negativo -1.175494E-38 < Resultado < -1.401298E-45 (número negativo)	0	0	1	1
Desbordamiento negativo +1.401298E-45 < Resultado < +1.175494E-38 (número positivo)	0	0	1	1
Desbordamiento Resultado < -3.402823E+38 (número negativo)	0	1	1	1
Desbordamiento Resultado > 3.402823E+38 (número positivo)	1	0	1	1
Número en coma flotante no válido u operación no permitida (valor de entrada fuera del margen válido de valores)	1	1	1	1

8.3 Operaciones bàsicas

8.3.1 ADD_R : Sumar números en coma flotante

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	REAL	E, A, M, D, L o constante	Primer valor a sumar (= primer sumando)
IN2	REAL	E, A, M, D, L o constante	Segundo valor a sumar (= segundo sumando)
OUT	REAL	E, A, M, D, L	Resultado de la suma
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

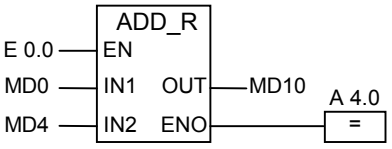
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Sumar números en coma flotante**. Esta operación suma las entradas IN1 e IN2. El resultado se puede consultar en la salida OUT. Si una de las entradas o el resultado no son números en coma flotante, los bits OV y OS tienen el valor "1" y ENO el valor "0".

En el apartado Evaluar encontrará informaciones sobre la evaluación de lo indicado en la palabra de estado.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

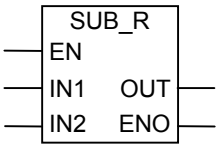
Ejemplo



El cuadro ADD_R se activa cuando E 0.0 = 1. El resultado de la suma de MD0 + MD4 se almacena en la palabra doble de marcas MD10. Si una de las entradas o el resultado no son números en coma flotante, y el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0" y la operación no se realiza.

8.3.2 SUB_R : Restar números en coma flotante

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	REAL	E, A, M, D, L o constante	Primer valor (del que se resta) (= minuendo)
IN2	REAL	E, A, M, D, L o constante	Valor a restar del primer valor (= substraendo)
OUT	REAL	E, A, M, D, L	Resultado de la sustracción (= diferencia)
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

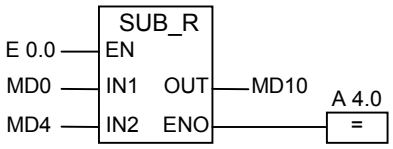
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Restar números en coma flotante**. Esta operación resta la entrada IN2 de IN1. El resultado se puede consultar en la salida OUT. Si una de las entradas o el resultado no son números en coma flotante, los bits OV y OS tienen el valor "1" y ENO el valor "0".

En el apartado Evaluar encontrará informaciones sobre la evaluación de lo indicado en la palabra de estado.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

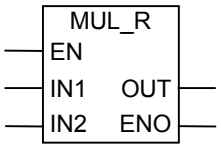
Ejemplo



El cuadro SUB_R se activa cuando E 0.0 = 1. El resultado de la resta de MD0 - MD4 se almacena en la palabra doble de marcas MD10. Si una de las entradas o el resultado no son números en coma flotante, y el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0" y la operación no se realiza.

8.3.3 MUL_R : Multiplicar números en coma flotante

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	REAL	E, A, M, D, L o constante	Primer valor a multiplicar (= multiplicador)
IN2	REAL	E, A, M, D, L o constante	Segundo valor a multiplicar (= multiplicando)
OUT	REAL	E, A, M, D, L	resultado de la multiplicación (= producto)
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

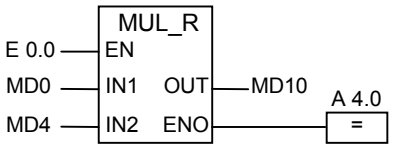
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Multiplicar números en coma flotante**. Esta operación multiplica las entradas IN1 e IN2. El resultado se puede consultar en la salida OUT. Si una de las entradas o el resultado no son números en coma flotante, los bits OV y OS tienen el valor "1" y ENO el valor "0".

En el apartado Evaluar encontrará informaciones sobre la evaluación de lo indicado en la palabra de estado.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

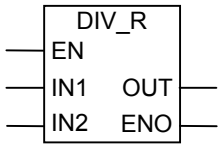
Ejemplo



El cuadro MUL_R se activa cuando E 0.0 = 1. El resultado de la multiplicación de MD0 x MD4 se almacena en la palabra doble de marcas MD10. Si una de las entradas o el resultado no son números en coma flotante, y el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0" y la operación no se realiza.

8.3.4 DIV_R : Dividir números en coma flotante

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	REAL	E, A, M, D, L o constante	Dividendo
IN2	REAL	E, A, M, D, L o constante	Divisor
OUT	REAL	E, A, M, D, L	Resultado de la división (= cociente)
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

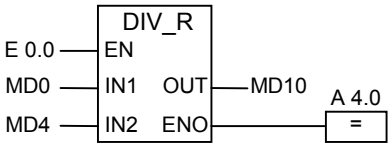
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Dividir números en coma flotante**. Esta operación divide la entrada IN1 por IN2. El resultado se puede consultar en la salida OUT. Si una de las entradas o el resultado no son números en coma flotante, los bits OV y OS tienen el valor "1" y ENO el valor "0".

En el apartado Evaluar encontrará informaciones sobre la evaluación de lo indicado en la palabra de estado.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

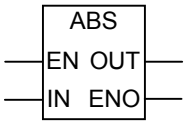
Ejemplo



El cuadro DI_R se activa cuando E 0.0 = 1. El resultado de la división de MD0 entre MD4 se almacena en la palabra doble de marcas MD10. Si una de las entradas o el resultado no son números en coma flotante, y el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0" y la operación no se realiza.

8.3.5 ABS : Calcular el valor absoluto de un número en coma flotante

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Valor de entrada: número en coma flotante
OUT	REAL	E, A, M, D, L	Valor de salida: valor absoluto del número en coma flotante
ENO	BOOL	E, A, M, D, L	Salida de habilitación

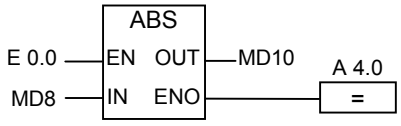
Descripción

Con la operación **Calcular el valor absoluto de un número en coma flotante** se puede calcular el valor absoluto de un número en coma flotante.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	-	-	-	-	0	X	X	1

Ejemplo



Si E 0.0 = 1, MD12 proporciona el valor absoluto de MD8.

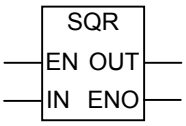
MD8 = +6,234 da como resultado MD12 = 6,234.

La salida A 4.0 es ”0" cuando no se ejecuta la conversión (ENO = EN = 0).

8.4 Operaciones ampliadas

8.4.1 SQR : Calcular el cuadrado de un número en coma flotante

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Número
OUT	REAL	E, A, M, D, L	Cuadrado del número
ENO	BOOL	E, A, M, D, L	Salida de habilitación

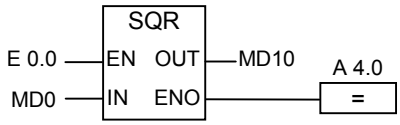
Descripción

Con la operación **Calcular el cuadrado de un número en coma flotante** se puede elevar éste al cuadrado. Si una de las entradas o el resultado no son números en coma flotante, el valor de los bits OV y OS es ”1", y el valor de ENO es ”0".

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

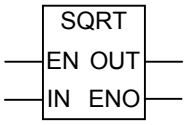
Ejemplo



El cuadro SQR se activa cuando E 0.0 = 1. El resultado de SQRT (MDO) se almacena en la palabra doble de marcas MD10. Si MD0 < 0, o si una de las entradas o el resultado no son números en coma flotante y el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0".

8.4.2 SQRT : Calcular la raíz cuadrada de un número en coma flotante

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Número
OUT	REAL	E, A, M, D, L	Raíz cuadrada del número
ENO	BOOL	E, A, M, D, L	Salida de habilitación

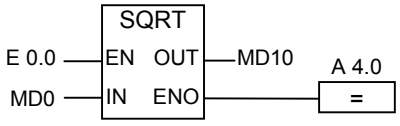
Descripción

Con la operación **Calcular la raíz cuadrada de un número en coma flotante** se puede calcular la raíz cuadrada de un número en coma flotante. Esta operación dará un resultado positivo cuando el operando sea mayor que ”0". Si una de las entradas o el resultado no son números en coma flotante, el valor de los bits OV y OS es ”1", y el valor de ENO es ”0".

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

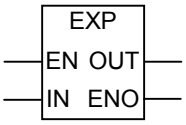
Ejemplo



El cuadro SQRT se activa cuando E 0.0 = 1. El resultado de SQRT (MDO) se almacena en la palabra doble de marcas MD10. Si MD0 < 0, o si una de las entradas o el resultado no son números en coma flotante y el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0".

8.4.3 EXP : Calcular el valor exponencial de un número en coma flotante

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Número
OUT	REAL	E, A, M, D, L	Exponente del número
ENO	BOOL	E, A, M, D, L	Salida de habilitación

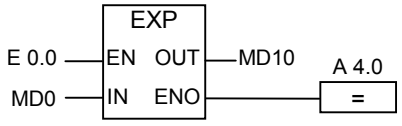
Descripción

Con la operación **Calcular el valor expotencial de un número en coma flotante** se puede calcular el valor exponencial en base e (= 2,71828...) de un número en coma flotante. Si una de las entradas o el resultado no son números en coma flotante, el valor de los bits OV y OS es ”1", y el valor de ENO es ”0".

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

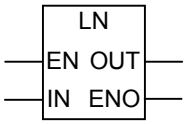
Ejemplo



El cuadro EXP se activa cuando E 0.0 = 1. El resultado de EXP (MD0) se almacena en la palabra doble de marcas MD10. Si una de las entradas o el resultado no son números en coma flotante, y si el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0".

8.4.4 LN : Calcular el logaritmo natural de un número en coma flotante

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Número
OUT	REAL	E, A, M, D, L	Logaritmo natural del número
ENO	BOOL	E, A, M, D, L	Salida de habilitación

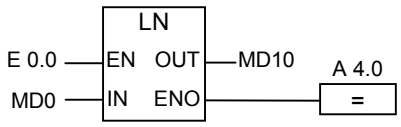
Descripción

Con la operación **Calcular el logaritmo natural de un número en coma flotante** se puede calcular el logaritmo natural de un número de este tipo. Si una de las entradas o el resultado no son números en coma flotante, el valor de los bits OV y OS es ”1", y el valor de ENO es ”0".

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

Ejemplo



El cuadro LN se activa cuando E 0.0 = 1. El resultado de LN (MD0) se almacena en la palabra doble de marcas MD10. Si MD0 < 0, o si una de las entradas o el resultado no son números en coma flotante, y el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0".

8.4.5 Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante

Descripción

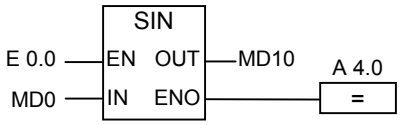
Con las siguientes operaciones puede calcular funciones trigonométricas de los ángulos que están representados por números en coma flotante (32 bits, IEEE 754):

Operación	Significado
SIN	Calcula el seno del número en coma flotante de un ángulo indicado por la medida de su arco.
ASIN	Calcula el arco seno del número en coma flotante. El resultado es un ángulo indicado por la medida de su arco. El valor queda dentro del siguiente rango de valores: π / 2 <= arcoseno <= + π / 2, siendo π = 3.14...
COS	Calcula el coseno del número en coma flotante de un ángulo indicado por la medida de su arco.
ACOS	Calcula el arco coseno del número en coma flotante. El resultado es un ángulo indicado por la medida de su arco. El valor queda dentro del siguiente rango de valores: 0 <= arcocoseno <= + π, siendo π = 3.14...
TAN	Calcula la tangente del número en coma flotante de un ángulo indicado por la medida de su arco. El ángulo se guarda como número en coma flotante.
ATAN	Calcula el arco tangente del número en coma flotante. El resultado es un ángulo indicado por la medida de su arco. El valor queda dentro del siguiente rango de valores: -π / 2 <= arcotangente <= + π / 2, siendo π = 3.14...

Palabra de estado

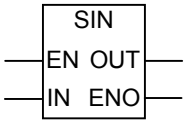
	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	X	X	X	X	X	0	X	X	1

Ejemplo



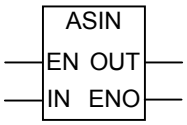
El cuadro SIN se activa cuando E 0.0 = 1. El resultado de SIN (MD0) se almacena en la palabra doble de marcas MD10. Si una de las entradas o el resultado no son números en coma flotante, y si el estado de señal de E 0.0 = 0, a la salida A 4.0 se le asigna la señal ”0".

Símbolo



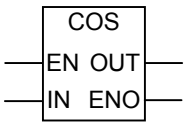
Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Número
OUT	REAL	E, A, M, D, L	Seno del número
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Símbolo



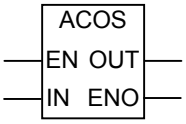
Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Número
OUT	REAL	E, A, M, D, L	Arco seno del número
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Símbolo



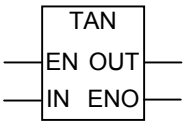
Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Número
OUT	REAL	E, A, M, D, L	Coseno del número
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Símbolo



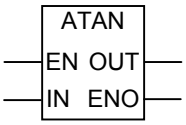
Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Número
OUT	REAL	E, A, M, D, L	Arco coseno del número
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Número
OUT	REAL	E, A, M, D, L	Tangente del número
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Símbolo

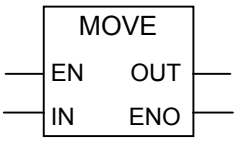


Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	REAL	E, A, M, D, L o constante	Número
OUT	REAL	E, A, M, D, L	Arco tangente del número
ENO	BOOL	E, A, M, D, L	Salida de habilitación

9 Operaciones de transferencia

9.1 MOVE : Transferir un valor

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN	Todos los tipos de datos simples de 8, 16 y 32 bits de longitud	E, A, M, D, L o constante	Valor fuente
OUT	Todos los tipos de datos simples de 8, 16 y 32 bits de longitud	E, A, M, D, L	Dirección de destino
ENO	BOOL	E, A, M, D, L	Salida de habilitación

Descripción

La operación **Transferir un valor** permite inicializar variables con valores determinados.

El valor indicado en la entrada IN se copia en el operando indicado de la salida OUT. ENO y EN tienen el mismo estado de señal.

La operación **Transferir un valor** puede copiar con el cuadro MOVE todos los tipos de datos simples de 8, 16 ó 32 bits de longitud. Los tipos de datos definidos por el usuario tales como campos o estructuras tienen que copiarse con la función de sistema SFC 20 "BLKMOV".

La operación **Transferir un valor** es afectada por el Master Control Relay (MCR). Para más información sobre el funcionamiento del MCR v. también apto. MCR on/off.

Palabra de estado

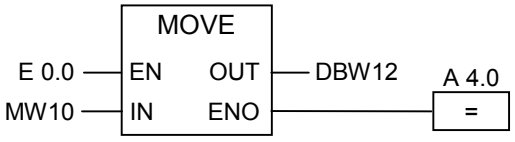
	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	1	-	-	-	-	0	1	1	1

Nota

Al transferir un valor a un tipo de datos de longitud diferente se cortan o truncan si es preciso los bytes más significativos o bien se rellenan con ceros:

Ejemplo: palabra doble	**1111 1111**	**0000 1111**	**1111 0000**	**0101 0101**
Transferencia	**Resultado**			
a una palabra doble:	1111 1111	0000 1111	1111 0000	0101 0101
a un byte:				0101 0101
a una palabra:			1111 0000	0101 0101
Ejemplo: byte:				**1111 0000**
Transferencia	**Resultado**			
a un byte:				1111 0000
a una palabra:			0000 0000	1111 0000
a una palabra doble:	0000 0000	0000 0000	0000 0000	1111 0000

Ejemplo



La operación se ejecuta cuando E 0.0 =1.

El contenido de MW10 se copia en la palabra de datos 12 del DB abierto.

Si se ejecuta la operación, A 4.0 = 1.

10 Operaciones de control del programa

10.1 Lista de operaciones de control del programa

Descripción

Se dispone de las operaciones de control del programa siguientes:

- CALL : Abrir FC/SFC sin parámetros
- CALL_FB Abrir FB
- CALL_FC Abrir FC
- CALL_SFB Abrir SFB
- CALL_SFC Abrir SFC
- Abrir multiinstancias
- Llamar a un bloque de una librería

- Operaciones Master Control Relay
- Notas importantes sobre el uso de la función MCR
- MCR< / MCR> : Conectar/Desconectar Master Control Relay
- MCRA / MCRD : Inicio/Fin Master Control Relay

- RET : Retorno

10.2 CALL : Llamar FC/SFC sin parámetros

Símbolo

<número de la FC/SFC>

Parámetro	Tipo de datos	Area de memoria	Descripción
Nº	BLOCK_FC	-	Número de la FC o SFC (p. ej.: FC10 o SFC59). Las SFCs disponibles varían en función de la CPU que se utilice. Las llamadas condicionadas introduciendo parámetros del tipo BLOCK_FC como operandos sólo se pueden efectuar para abrir bloques de función (FB); ello no es posible cuando se está trabajando con funciones (FC).

Descripción

La operación **Abrir FC/SFC sin parámetros** se utiliza para abrir una función (FC) o una función de sistema (SFC) que no tenga parámetros. Dependiendo de la combinación precedente, la llamada es condicionada o incondicionada (absoluta) (v. ejemplo).

La llamada condicionada no permite introducir parámetros del tipo BLOCK_FC en la parte que contiene las instrucciones de una función (FC). Sin embargo, en un bloque de función (FB) sí que es posible introducir como operando un parámetro del tipo BLOCK_FC.

La llamada condicionada se ejecuta solamente si RLO es "1". Si no se ejecuta una llamada condicionada, el RLO después de la llamada es "0". Si la operación se ejecuta, se efectúan las siguientes funciones:

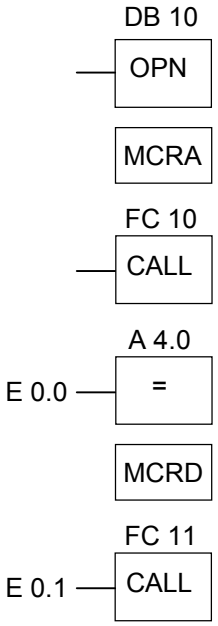
- Memoriza la dirección que se necesita para retornar al bloque que ha efectuado la llamada.
- Memoriza ambos registros de bloque de datos (DB y DI).
- Sustituye el área de datos locales previa por el área de datos locales actual.
- Crea el nuevo área de datos locales para la FC o la SFC abierta.
- Desplaza el bit MA (bit activo MCR) a la pila de bloques (BSTACK).

Después de ejecutar dichas funciones, la ejecución del programa continúa en el bloque abierto.

Palabra de estado

		RB	BI1	BI0	DEB	DM	OU	ETAT	RLG	/PI
condicionada	se escribe	-	-	-	-	0	0	1	1	0
incondicionada	se escribe	-	-	-	-	0	0	1	-	0

Ejemplo



Si se ejecuta una llamada incondicionada de la FC10, la operación CALL opera del siguiente modo:

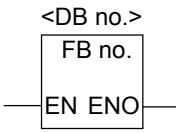
- Almacena la dirección de retorno del FB actual.
- Almacena los selectores para DB10 y el DB de instancia del FB.
- Desplaza el bit MA (que la operación MCRA había puesto a "1" a la pila de bloques (BSTACK), y lo vuelve a poner a "0" para la FC10 que se ha llamado.

La ejecución del programa continúa en la FC10. Si desea utilizar la función MCR en la FC10 tendrá que activarla allí otra vez. Una vez terminada la FC10, la ejecución del programa vuelve al FB que está efectuando la llamada. Se reestablece el bit MA. El DB10 y el DB de instancia del FB de usuario son de nuevo los DB actuales (activos), y ello dependiendo de cuáles sean los DB que había utilizado la FC10.

Después de retornar de la FC10 se le asigna a la salida A 4.0 el estado de señal de E 0.0. Al abrir la FC11 se está efectuando una llamada condicionada, que sólo se ejecutará si la entrada E 0.0 = 1. Caso de ejecutarse la llamada, la función será la misma que al abrir la FC10.

10.3 CALL_FB Llamar FB

Símbolo



El símbolo varía según el bloque de función (dependiendo de si hay parámetros o de cuántos parámetros existen). EN, ENO y el nombre o número del FB tienen que estar siempre presentes.

Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, L, D	Entrada de habilitación
ENO	BOOL	E, A, M, L, D	Salida de habilitación
Nº FB Nº DB	BLOCK_FB BLOCK_DB	- -	Número del FB/DB, el área depende de la CPU utilizada

Descripción de la operación

CALL_FB (Abrir FB) se ejecuta si EN es 1. Al ejecutarse la operación CALL_FB sucede lo siguiente:

- Memoriza la dirección que se necesita para retornar al bloque que ha efectuado la llamada.
- Memoriza ambos registros de bloque de datos (bloque de datos y bloque de datos de instancia).
- Sustituye el área de datos locales previa por el área de datos locales actual.
- Crea el nuevo área de datos locales para la función FC abierta.
- Desplaza el bit MA (bit activo MCR) a la pila de bloques (BSTACK).

Palabra de estado

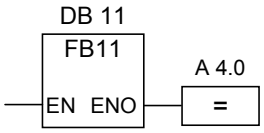
		RB	BI1	BI0	DEB	DM	OU	ETAT	RLG	/PI
condicionada	se escribe	X	-	-	-	0	0	X	X	X
incondicionada	se escribe	-	-	-	-	0	0	X	X	X

Ejemplo

Segmento 1

Segmento 2

Segmento 3



Segmento 4

Los circuitos del esquema de contactos arriba representados son elementos del programa de un bloque de función escrito por el usuario. En este bloque de función se abre DB10 y se activa el MCR. Si se ejecuta la llamada absoluta al FB11 sucede lo siguiente:

Se memorizan la dirección de retorno del bloque de función que efectúa la llamada, los datos seleccionados para el DB10 y los datos para el bloque de datos de instancia perteneciente al bloque de función que efectúa la llamada. El bit MA que fue puesto a "1" por la función MCRA se desplaza a la pila BSTACK y seguidamente es puesto a "0" para el bloque FB11 llamado. La ejecución del programa continúa en FB11. Si el FB11 necesita el MCR, hay que volver a activar el MCR en el bloque de función. El estado del RLO tiene que almacenarse a través de la operación [SAVE] en el bit RB para poder evaluar los posibles errores en el FB que efectúa la llamada. Una vez finalizada la ejecución del FB11, el programa vuelve al bloque de función que efectúa la llamada. El bit MA se restablece y el bloque de datos de instancia perteneciente al bloque de función escrito por el usuario se vuelve a convertir en el DB actual. Si el FB11 es ejecutado correctamente, ENO es 1 y, por tanto, A 4.0 es 1.

Nota

El número del bloque de datos abierto anteriormente se pierde al llamar FB/SFB. Habrá que volver a abrir el DB que se necesite.

10.4 CALL_FC Llamar FC

Símbolo

El símbolo varía según la función (dependiendo de si hay parámetros o de cuántos parámetros existen). EN, ENO y el nombre o número de la FC tienen que estar siempre presentes.

Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, L, D	Entrada de habilitación
ENO	BOOL	E, A, M, L, D	Salida de habilitación
Nº FC	BLOCK_FC	-	Número del FC, el área depende de la CPU utilizada

Descripción de la operación

CALL_FC (Abrir FC) llama a una función (FC). La llamada se ejecuta si EN es 1. Al ejecutarse la operación CALL_FC sucede lo siguiente:

- Memoriza la dirección que se necesita para retornar al bloque que ha efectuado la llamada.
- Sustituye el área de datos locales previa por el área de datos locales actual.
- Crea el nuevo área de datos locales para la función FC abierta.
- Desplaza el bit MA (bit activo MCR) a la pila de bloques (BSTACK).

Seguidamente, la ejecución del programa continúa en la función que se ha llamado.

Para determinar ENO se consulta el bit RB, el usuario tiene que asignarle a éste, en el bloque llamado, con [SAVE] el estado deseado (evaluación de errores).

Si llama a una FC y la tabla de declaración de variables del bloque llamado contiene declaraciones del tipo IN, OUT e IN_OUT, dichas variables aparecerán en la lista de parámetros formales del bloque que ha efectuado la llamada.

Al llamar las FCs **es absolutamente necesario** asignar parámetros actuales (reales) a los parámetros formales en el punto de llamada. Los valores iniciales que pueda contener la declaración de variables carecen de significado.

Palabra de estado

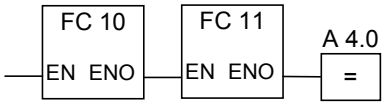
		RB	BI1	BI0	DEB	DM	OU	ETAT	RLG	/PI
condicionada	se escribe	X	-	-	-	0	0	X	X	X
incondicionada	se escribe	-	-	-	-	0	0	X	X	X

Ejemplo

Segmento 1

Segmento 2

Segmento 3



Los circuitos del esquema de contactos representados en el ejemplo son elementos del programa de un bloque de función escrito por el usuario. En este bloque de función se abre DB10 y se activa el MCR. Si se ejecuta la llamada absoluta a la FC10 sucede lo siguiente:

Se memorizan la dirección de retorno del bloque de función que efectúa la llamada, los datos seleccionados para el DB10 y los datos para el bloque de datos de instancia perteneciente al bloque de función que efectúa la llamada. El bit MA que fue puesto a "1" por la operación MCRA se desplaza a la pila BSTACK y seguidamente es puesto a "0" para el bloque FC10 que se ha llamado. La ejecución del programa continúa en FC10. Si FC10 necesita el MCR, hay que volver a activar el MCR en FC10. El estado del RLO tiene que almacenarse a través de la operación [SAVE] en el bit RB para poder realizar una evaluación de errores en el FB que ejecuta la llamada. Una vez finalizada la ejecución de la FC10, el programa vuelve al bloque de función que efectúa la llamada. El bit MA se restablece. Al finalizar la ejecución de la FC10 el programa continúa, en función de la señal de ENO, en el FB que efectúa la llamada:

ENO = 1 se ejecuta la FC11

ENO = 0 la ejecución comienza en el segmento siguiente

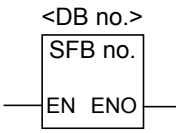
Si la ejecución de FC11 también es correcta, ENO es 1 y, por tanto, A 4.0 es 1.

Nota

Después de retornar al bloque que efectúa la llamada puede ocurrir que el DB que se había abierto anteriormente ahora ya no esté abierto. Sírvase tener en cuenta la indicación al respecto en el archivo README.

10.5 CALL_SFB Llamar SFB

Símbolo



El símbolo varía según el bloque de función de sistema (dependiendo de si hay parámetros o de cuántos parámetros existen). EN, ENO y el nombre o número del SFB tienen que estar siempre presentes.

Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, L, D	Entrada de habilitación
ENO	BOOL	E, A, M, L, D	Salida de habilitación
Nº SFB Nº DB	BLOCK_SFB BLOCK_DB	- -	Número del SFB/DB, el área depende de la CPU utilizada

Descripción de la operación

CALL_SFB (Abrir SFB) se ejecuta si EN es 1. Al ejecutarse la operación CALL_SFB sucede lo siguiente:

- Memoriza la dirección que se necesita para retornar al bloque que ha efectuado la llamada.
- Memoriza ambos registros de bloque de datos (bloque de datos y bloque de datos de instancia).
- Sustituye el área de datos locales previa por el área de datos locales actual.
- Crea el nuevo área de datos locales para la función FC abierta.
- Desplaza el bit MA (bit activo MCR) a la pila de bloques (BSTACK).

Seguidamente, la ejecución del programa continúa en el bloque de función de sistema llamado. ENO es "1" si la llamada al bloque de función de sistema (EN = 1) se ejecutó sin errores.

Palabra de estado

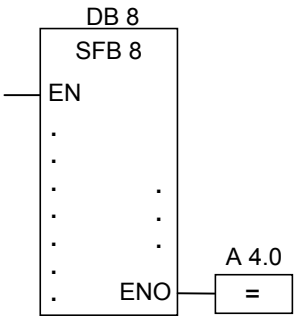
		RB	BI1	BI0	DEB	DM	OU	ETAT	RLG	/PI
condicionada	se escribe	X	-	-	-	0	0	X	X	X
incondicionada	se escribe	-	-	-	-	0	0	X	X	X

Símbolo

Segmento 1

Segmento 2

Segmento 3



Segmento 4

Los circuitos del esquema de contactos arriba representados son elementos del programa de un bloque de función escrito por el usuario. En este bloque de función se abre DB10 y se activa el MCR. Al ejecutarse la llamada absoluta al SFB8 sucede lo siguiente:

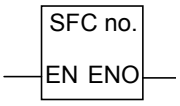
Se memorizan la dirección de retorno del bloque de función que efectúa la llamada, los datos seleccionados para el DB10 y los datos para el bloque de datos de instancia pertenciente al bloque de función que efectúa la llamada. El bit MA que fue puesto a "1" por la función MCRA se desplaza a la pila BSTACK y seguidamente puesto a "0" para el SFB8 llamado. La ejecución del programa continúa en SFB8. Una vez finalizada la ejecución de SFB8, el programa vuelve al bloque de función que efectúa la llamada. El bit MA se restablece y el bloque de datos de instancia pertenciente al bloque de función escrito por el usuario se vuelve a convertir en el DB de instancia actual. Si el SFB8 es ejecutado correctamente, ENO es 1 y, por tanto, A4.0 es 1.

Nota

El número del bloque de datos abierto anteriormente se pierde al llamar FB/SFB. Habrá que volver a abrir el DB que se necesite.

10.6 CALL_SFC Llamar SFC

Símbolo



El símbolo varía según la función de sistema (dependiendo de si hay parámetros o de cuántos parámetros existen). EN, ENO y el nombre o número de SFC tienen que estar siempre presentes.

Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, L, D	Entrada de habilitación
ENO	BOOL	E, A, M, L, D	Salida de habilitación
Nº SFC	BLOCK_SFC	-	Número del SFC, el área depende de la CPU utilizada

Descripción de la operación

CALL_SFC (Abrir SFC) llama a una función de sistema. La llamada se ejecuta si EN es 1. Al ejecutarse la operación CALL_SFC sucede lo siguiente:

- Memoriza la dirección que se necesita para retornar al bloque que ha efectuado la llamada.
- Sustituye el área de datos locales previa por el área de datos locales actual.
- Crea el nuevo área de datos locales para la función FC abierta.
- Desplaza el bit MA (bit activo MCR) a la pila de bloques (BSTACK).

Seguidamente, la ejecución del programa continúa en la función de sistema que se ha llamado. ENO es "1" si la llamada a la función (EN = 1) se produjo sin errores.

Palabra de estado

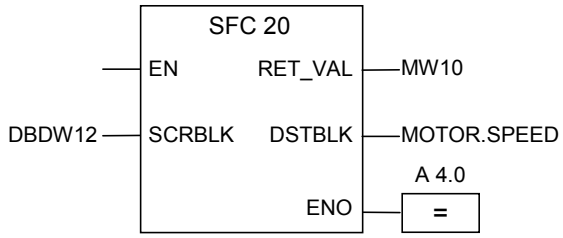
		RB	BI1	BI0	DEB	DM	OU	ETAT	RLG	/PI
condicionada	se escribe	X	-	-	-	0	0	X	X	X
incondicionada	se escribe	-	-	-	-	0	0	X	X	X

Ejemplo

Segmento 1

Segmento 2

Segmento 3



Los circuitos del esquema de contactos arriba representados son elementos del programa de un bloque de función escrito por el usuario. En este bloque se abre DB10 y se activa el MCR. Si se ejecuta la llamada absoluta a la SFC20 sucede lo siguiente:

Se memorizan la dirección de retorno del bloque de función que efectúa la llamada, los datos seleccionados para el DB10 y los datos para el bloque de datos de instancia perteneciente al bloque de función que efectúa la llamada. El bit MA que fue puesto a "1" por la operación MCRA se desplaza a la pila BSTACK y seguidamente es puesto a "0" para el bloque SFC20 que se ha llamado. La ejecución del programa continúa en SFC20. Una vez finalizada la ejecución de la SFC20, el programa vuelve al bloque de función que efectúa la llamada. El bit MA se restablece.

Una vez finalizada la ejecución de la SFC20 el programa continúa, en función de cuál sea la señal en ENO, en el FB que efectúa la llamada:

ENO = 1 A 4.0 = 1

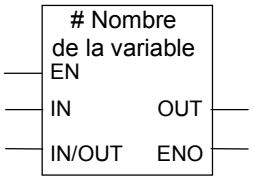
ENO = 0 A 4.0 = 0

Nota

Después de retornar al bloque que efectúa la llamada puede ocurrir que el DB que se había abierto anteriormente ahora ya no esté abierto. Sírvase tener en cuenta la indicación al respecto en el archivo README.

10.7 Abrir multiinstancias

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
ENO	BOOL	E, A, M, D, L	Salida de habilitación
# Nombre de la variable	FB/SFB	-	Nombre de la multiinstancia

Descripción

Para crear una multiinstancia debe declararse una variable estática del tipo de datos de un bloque de funciones. En el catálogo de elementos del programa sólo aparecen las multiinstancias ya declaradas. El símbolo varía según cuál sea la multiinstancia (dependiendo de si hay parámetros y, en caso afirmativo, de cuántos parámetros existen). EN, ENO y el nombre de la variable siempre tienen que estar presentes.

Palabra de estado

	RB	BI1	BI0	DEB	DM	OU	ETAT	RLG	/PI
se escribe	-	-	-	-	0	0	X	X	X

10.8 Llamar a un bloque de una librería

Se ofrecen las librerías que el Administrador SIMATIC haya encontrado. De ese conjunto de librerías, se pueden escoger:

- los bloques que están integrados en el sistema operativo de la CPU utilizada (librería "Standard Library" para proyectos de STEP-7 de la versión 3 y "stdlibs (V2)" para proyectos de STEP-7 de la versión 2),
- los bloques que el usuario mismo haya guardado en librerías con el fin de poder utilizarlas varias veces.

10.9 Operaciones Master Control Relay

Notas importantes sobre el uso de la función MCR

Definición del Master Control Relay

El Master Control Relay (MCR) (v. también apto. MCR on/off) es un relé maestro que activa y desactiva el flujo de señal. Un flujo de señal desactivado corresponde a una secuencia de operaciones que escribe un cero en lugar del valor calculado, o a una secuencia de operaciones que no modifica el valor de memoria existente.

Las operaciones **Asignación** y **Conector** escriben un 0 en la memoria si el MCR es 0. Las operaciones **Activar operando** y **Desactivar operando** no modifican el valor existente.

Las operaciones que dependen del MCR son:

- #: Conector
- =: Asignación
- S: Activar salida
- R: Desactivar salida
- SR: Flipflop de activación/desactivación
- RS: Flipflop de desactivación/activación
- MOVE: Transferir un valor

Operaciones que dependen del MCR y su reacción ante el estado de señal de MCR

Estado de señal del MCR	Asignar, Conector	Activar o desactivar operando	Transferir valor
0 ("OFF")	Escribe "0". (Imita a un relé que pasa al estado de reposo en caso de fallar la alimenta-ción.)	No escribe. (Imita a un relé que permanece en su estado actual en caso de fallar la alimentación.)	Escribe "0". (Imita a un compo-nente que da el valor "0" en caso de fallar la alimentación.)
1 ("ON")	Ejecución normal.	Ejecución normal.	Ejecución normal.

10.10 Notas importante sobre el uso de la función MCR

Precaución en bloques en los que se ha activado Master Control Relay con MCRA

- Si se ha desactivado el MCR, en seccione de programa entre conectar Master Control Relay y desconectar Master Control Relay todas las asignaciones escribirán el valor 0. Esto afecta naturalmente a **todos** los cuadros que contienen una asignación, incluida la transferencia de paráemtros a bloques
- El MCR se desactivará exactamente cuando delante de una orden de conectar Master Control Relay el RLO fue = 0.

Peligro: STOP del AS o comportamiento no definido en runtime

Para el cálculo de direcciones, el compilador también tiene acceso de escritura a los datos locales después de las variables temporales definidas en VAR_TEMP. Para ello, las siguientes secuencias de comandos ponen el PLC en STOP o provocan comportamientos indefinidos en runtime:

Acceso a parámetros formales

- Accesos a componentes de parámetros FC compuestos del tipo STRUCT, UDT, ARRAY, STRING.
- Accesos a componentes de parámetros FB compuestos del tipo STRUCT, UDT, ARRAY, STRING del área IN_OUT en un bloque apto para multiinstancia (de la versión 2).
- Accesos a parámetros de un FB multiinstancia (de la versión 2) si su dirección es mayor que 8180.0.
- El acceso en el FB multiinstancia (de la versión 2) a un parámetro del tipo BLOCK_DB abre el DB 0. Los siguientes accesos a datos ponen la CPU en STOP. Con TIMER, COUNTER, BLOCK_FC, BLOCK_FB se utiliza siempre T 0, Z 0, FC 0 o FB 0.

Transferencia de parámetros

- Calls en las que se transfieren parámetros.

KOP/FUP

- Las ramas T y los conectores en KOP o FUP arrancan con RLO = 0.

Remedio

Active las órdenes mencionadas en función del MCR:

1º **Desactive** el Master Control Relay con Fin Master Control Relay **antes** de la instrucción correspondiente o antes del segmento involucrado.

2º **Active** nuevamente el Master Control Relay con Inicio Master Control Relay **después de** la instrucción correspondiente o después del segmento involucrado.

10.11 MCR< / MCR> : Conectar/Desconectar Master Control Relay

Notas importantes sobre el uso de la función MCR

Símbolo

Conectar MCR

Con la operación **Conectar Master Control Relay (MCR<)** se memoriza el RLO en la pila MCR y queda abierta un área MCR. El RLO que acaba de ser memorizado en la pila MCR al abrir el área MCR afecta dentro de dicha área a las operaciones enumeradas en Oper. MCR. La pila MCR opera como una memoria intermedia LIFO (LIFO = last in, first out), que como máximo contiene 8 entradas. Si la pila está llena, la operación Conectar Master Control Relay genera un error en la pila MCR (MCRF).

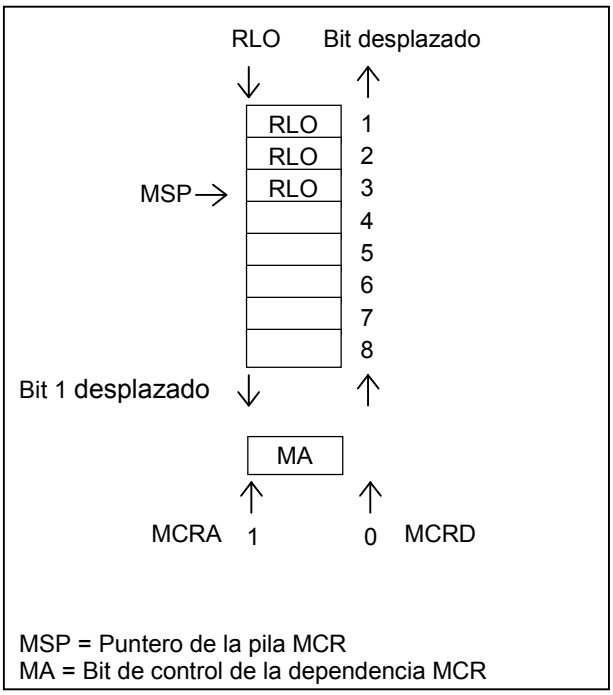
Símbolo

Desconectar MCR

Con la operación **Desconectar Master Control Relay (MCR>)** se cierra la última área abierta. Esto tiene lugar mediante el borrado de la entrada RLO en la pila MCR, a donde había sido desplazada por la operación Conectar Master Control Relay. La pila MCR opera como una memoria intermedia LIFO (LIFO = last in, first out). La entrada que queda libre al final de dicha pila se pone a 1. Si la pila ya está vacía la operación Desconectar Master Control Relay genera un error en la pila MCR (MCRF).

Pila MCR

El MCR es controlado por una pila de un bit de ancho y de ocho entradas de profundidad. El MCR se activa a condición de que las ocho entradas de la pila estén a "1". La operación MCR< copia el resultado lógico en la pila MCR; la operación MCR> retira la última entrada de la pila y pone la posición vacante a 1. En caso de error, es decir, cuando hay más de ocho operaciones MCR> consecutivas o cuando se intenta ejecutar una operación MCR> estando vacía la pila MCR, entonces se dispara el mensaje de error MCRF. La vigilancia de la pila MCR sigue al puntero de la pila (MSP: 0 = vacío, 1 = una entrada, 2 = dos entradas, ...8 = ocho entradas).



La operación **MCR<** toma el estado de señal del RLO y lo copia en el bit MCR.

La operación **MCR>** pone el bit MCR a "1" independientemente de las condiciones. (Por este motivo, las operaciones restantes entre las operaciones MCRA y MCRD funcionan independientemente del bit MCR.

Anidar operaciones (MCR<) y (MCR>)

Las operaciones **MCR<** y **MCR>** pueden anidarse. La profundidad máxima de anidado es de 8 niveles, es decir, solamente puede anidarse un máximo de ocho operaciones MCR< seguidas antes de insertar una operación MCR>. Hay que programar siempre la misma cantidad de operaciones MCR< que operaciones MCR>.

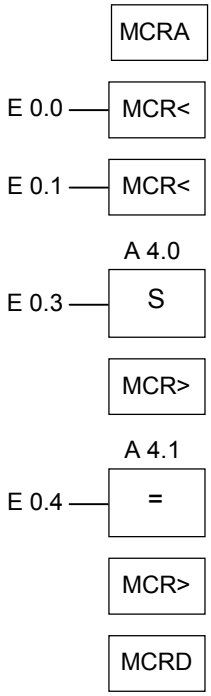
Cuando se anidan operaciones MCR<, el bit del nivel de anidado más profundo se forma cuando la operación MCR< se combina el RLO actual con el bit MCR actual, de acuerdo con la tabla de verdad Y.

Cuando una operación MCR> termina un nivel de anidado, toma el bit MCR del nivel más alto.

Palabra de estado

	RB	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLO**	**/ER**
se escribe	-	-	-	-	-	0	1	-	0

Ejemplo



Cuando la operación MCRA activa la función MCR puede crear hasta ocho áreas MCR anidadas. En nuestro ejemplo hay dos áreas MCR. La primera operación MCR> opera junto con la segunda operación MCR<. Todas las operaciones situadas entre el segundo grupo de anidamiento MCR (MCR< MCR>) pertenecen a la segunda área MCR. Las operaciones se ejecutan del siguiente modo:

- E 0.0 = 1: el estado de señal de E 0.4 se asigna a la salida A 4.1.
- E 0.0 = 0: la salida A 4.1 = 0, independientemente de cuál sea el estado de señal de E 0.4. La salida A 4.0 no varía, independientemente de cuál sea el estado de señal de E 0.3.
- E 0.0 y E 0.1 = 1: la salida A 4.1 se pone a "1" si E 0.3 =1 y A 4.1 = E 0.4.
- E 0.1 = 0: la salida A 4.0 no varía, independientemente del estado de señal de E 0.3 y de E 0.0.

10.12 MCRA / MCRD : Inicio/Fin Master Control Relay

Notas importantes sobre el uso de la función MCR

Símbolo

Inicio MCR

Con la operación **Inicio Master Control Relay** se conecta la dependencia de los comandos siguientes de MCR. Tras el comando se pueden programar las zonas MCR con las operaciones **Conectar y Desconectar MCR**. Cuando el programa activa un área MCR, todas las acciones del MCR dependen del contenido de la pila MCR.

Símbolo

Fin MCR

Con la operación **Fin Master Control Relay** se desconecta la dependencia de los comandos siguientes de MCR. Tras el comando no se pueden programar más zonas MCR. Cuando el programa desactiva un área MCR, independientemente de los registros de la pila MCR, MCR siempre conduce corriente.

La pila MCR y el bit que controla su dependencia (el bit MA) están referidos a un nivel concreto y tienen que salvaguardarse y restablecerse cada vez que se conmute el nivel de secuencia. Se preajustan al principio de cada nivel (los bits de entrada MCR 1 a 8 se ponen a "1", el puntero de la pila MCR = 0 y el bit MA = 0).

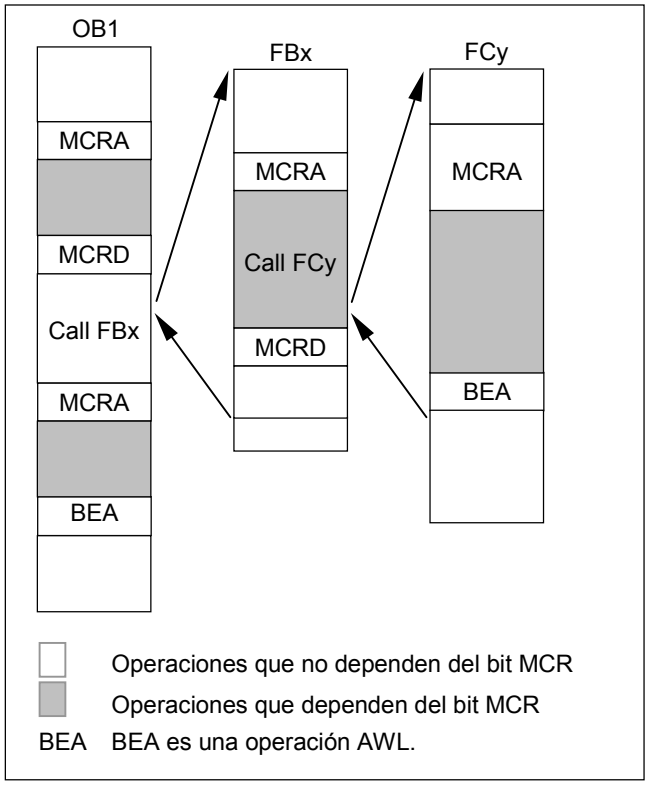
La pila MCR se pasa de bloque a bloque. El bit MA se salvaguarda cada vez que se llama al bloque y se pone a "0". Al final del bloque se vuelve a sacar.

El MCR puede ser implementado de forma que se optimice el tiempo de ejecución de las CPUs que generan código, ya que la dependencia del MCR no se pasa al bloque sino que debe ser activada explícitamente por la operación MCRA. La CPU que genera el código detecta esta operación y genera el código adicional para evaluar la pila MCR hasta detectar una operación MCRD o hasta alcanzar el final del bloque. Para las operaciones que se encuentran fuera de la zona MCRA/MCRD no aumenta el tiempo de ejecución.

En caso de emplear **MCRA** y **MCRD** en el programa, hay que tener en cuenta que deben utilizarse siempre por pares.

Activación y desactivación de un área Master Control Relay

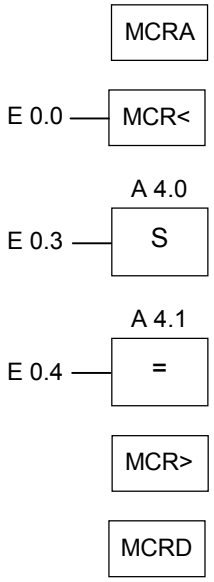
Las operaciones programadas entre MCRA y MCRD dependen del estado de señal del bit MCR. Las operaciones programadas fuera de una secuencia MCRA–MCRD no dependen del estado de señal del bit MCR. Si falta la operación MCRD, las operaciones programadas entre la operación MCRA y la operación BE dependerán del bit MCR.



Palabra de estado

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

Ejemplo



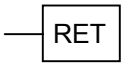
La operación MCRA activa la función MCR hasta el próximo MCRD. Las operaciones situadas entre MCR < y MCR> se procesan en función del bit MA (en este caso E 0.0).

- Si E 0.0 = 1, entonces:
  - A 4.0 se pone a "1", siempre que E 0.3 = 1
  - A 4.0 no varía, siempre que E 0.3 = 0
  - el estado de señal de E 0.4 se asigna a la salida A 4.1.
- Si E 0.0 = 0, entonces:
  - A 4.0 no varía, independientemente de cuál sea el estado de señal de E 0.3
  - A 4.1 toma el valor "0", independientemente de cuál sea el estado de señal de E 0.4.

La dependencia de las funciones (FC) y de los bloques de funciones (FB) del bit MCR ha de programarse en los bloques mismos, es decir, si se llama una FC o un FB desde una secuencia MCRA–MCRD, las instrucciones que contiene la secuencia no dependen automáticamente del bit MCR. Para que esto sí ocurra, hay que utilizar la operación MCRA en el bloque llamado.

10.13 RET : Retorno

Símbolo



Descripción

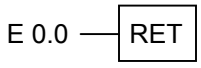
La operación **Retorno** se utiliza para abandonar bloques. Los bloques pueden abandonarse de forma condicionada.

Palabra de estado

	RB	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLO**	**/ER**
se escribe	*	-	-	-	0	0	1	1	0

* La operación **RET** se representa internamente en la secuencia "SAVE; BEB;", lo que influye también sobre el bit RB.

Ejemplo



Se sale del bloque cuando E 0.0 =1.

11 Operaciones de desplazamiento y de rotación

11.1 Operaciones de desplazamiento

11.1.1 Lista de operaciones de desplazamiento

Descripción

Las operaciones de desplazamiento sirven para desplazar bit a bit el contenido de la entrada IN, a la izquierda o a la derecha (consulte Registros de la CPU). El desplazamiento a la izquierda multiplica el contenido de la entrada IN por potencias de 2; el desplazamiento a la derecha divide el contenido de la entrada IN por potencias de 2. Por ejemplo, desplazando el equivalente binario del valor decimal 3 tres bits a la izquierda se obtiene en el acumulador el equivalente binario del valor decimal 24. Desplazando el equivalente binario del valor decimal 16 dos bits a la derecha se obtiene en el acumulador el equivalente binario del valor decimal 4.

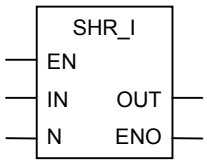
El número que se introduce en el parámetro de entrada N indica el número de bits que se va a desplazar. Las posiciones que quedan vacantes después de ejecutar la operación de desplazamiento se rellenan con ceros o con el estado de señal del bit de signo ("0" significa positivo y "1" significa negativo). El estado de señal del último bit desplazado se carga en el bit A1 de la palabra de estado. Los bits A0 y OV de la palabra de estado se ponen a "0". Para interpretar el bit A1 pueden utilizarse las operaciones de salto.

Se dispone de las siguientes operaciones de desplazamiento:

- SHR_I : Desplazar entero a la derecha
- SHR_DI : Desplazar entero doble a la derecha
- SHL_W : Desplazar palabra a la izquierda
- SHR_W : Desplazar palabra a la derecha
- SHL_DW : Desplazar palabra doble a la izquierda
- SHR_DW : Desplazar palabra doble a la derecha

11.1.2 SHR_I : Desplazar entero a la derecha

Símbolo

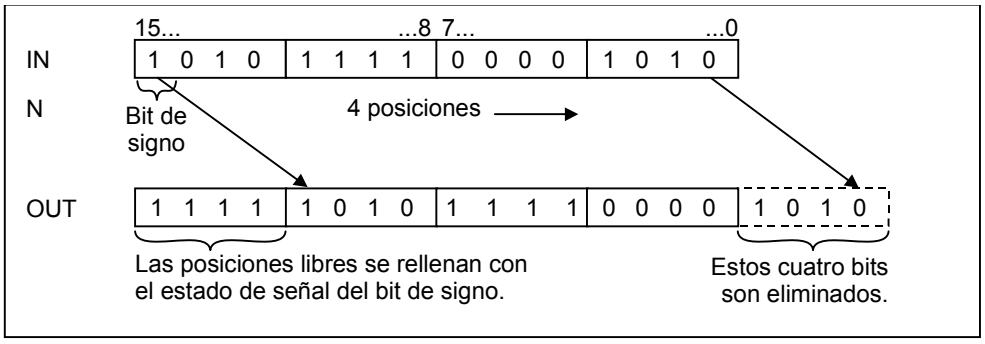


Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, L, D, T, Z	Entrada de habilitación
IN	INT	E, A, M, L, D	Valor a desplazar
N	WORD	E, A, M, L, D	Número de posiciones (bits) a desplazar
OUT	INT	E, A, M, L, D	Resultado de la operación de desplazamiento
ENO	BOOL	E, A, M, L, D	Salida de habilitación

Descripción

Un estado de señal de 1 en la entrada de habilitación (EN) activa la operación **Desplazar entero a la derecha**. Esta operación desplaza los bits 0 a 15 de la entrada IN bit a bit a la derecha. La entrada N indica el número de bits a desplazar. Si N es superior a 16, la operación se comporta como si N fuera 16. Las posiciones que quedan libres a la izquierda se rellenan con el estado de señal del bit 15 (que corresponde al signo de un entero), es decir, se rellenan con ceros si el número es positivo o se rellenan con unos si es negativo. El resultado de la operación de desplazamiento se determina consultando la salida OUT.

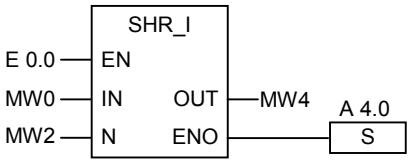
La operación activada pone los bits A0 y OV de la palabra de estado siempre a "0" cuando N es diferente de 0. El estado de señal de ENO es el mismo que el de EN.



Palabra de estado

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

Ejemplo



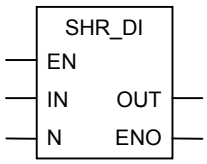
La operación se activa cuando E 0.0 = 1.

La palabra de marcas MW0 se desplaza a la derecha tantos bits como se hayan indicado en MW2.

El resultado se almacena en MW4. La salida A 4.0 se pone a ”1".

11.1.3 SHR_DI : Desplazar entero doble a la derecha

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, L, D, T, Z	Entrada de habilitación
IN	DINT	E, A, M, L, D	Valor a desplazar
N	WORD	E, A, M, L, D	Número de posiciones (bits) a desplazar
OUT	DINT	E, A, M, L, D	Resultado de la operación de desplazamiento
ENO	BOOL	E, A, M, L, D	Salida de habilitación

Descripción

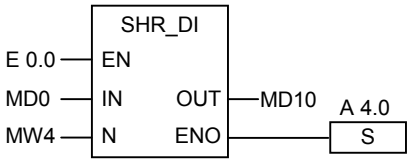
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Desplazar entero doble a la derecha**. Esta operación desplaza todo el contenido de la entrada IN bit a bit a la derecha. La entrada N indica el número de bits a desplazar. Si es superior a 32, la operación se comporta como si N fuera 32. Las posiciones que quedan libres a la izquierda se rellenan con el estado de señal del bit 15 (que es el signo de un número entero), es decir, se rellenan con ceros si el número es positivo y con unos si es negativo. El resultado de la operación de desplazamiento se determina consultando la salida OUT.

La operación activada pone los bits A0 y OV de la palabra de estado a "0" cuando N es diferente de 0. El estado de señal de ENO es el mismo que el de EN.

Palabra de estado

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

Ejemplo



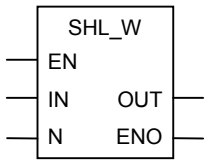
La operación se activa cuando E 0.0 = 1.

La palabra doble de marcas MD0 se desplaza a la derecha tantos bits como se hayan indicado en MW4.

El resultado se almacena en MD10. La salida A 4.0 se pone a ”1".

11.1.4 SHL_W : Desplazar palabra a la izquierda

Símbolo



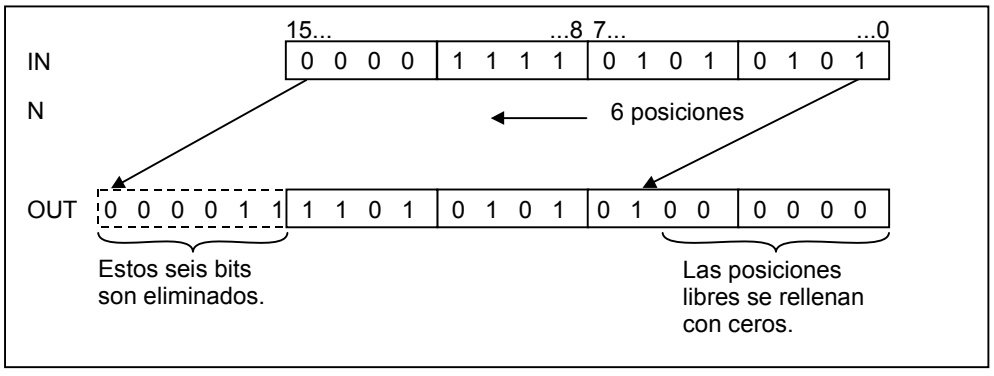
Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, L, D, T, Z	Entrada de habilitación
IN	WORD	E, A, M, L, D	Valor a desplazar
N	WORD	E, A, M, L, D	Número de posiciones (bits) a desplazar
OUT	WORD	E, A, M, L, D	Resultado de la operación de desplazamiento
ENO	BOOL	E, A, M, L, D	Salida de habilitación

Descripción

Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Desplazar palabra a la izquierda**. Esta operación desplaza los bits 0 a 15 de la entrada IN bit a bit a la izquierda.

La entrada N indica el número de bits a desplazar. Si N es superior a 16, la operación escribe un 0 en la la salida OUT y pone los bits A0 y OV de la palabra de estado a "0". Las posiciones que quedan libre a la derecha se rellenan con ceros. El resultado de la operación de desplazamiento se determina consultando la salida O.

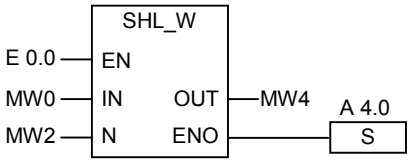
La operación activada pone los bits A0 y OV de la palabra de estado a "0" cuando N es diferente de 0. El estado de señal de ENO es el mismo que el de EN.



Palabra de estado

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

Ejemplo



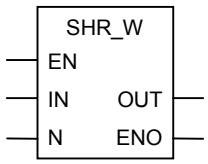
La operación se activa si el estado de señal de la entrada E 0.0 es 1.

La palabra de marcas MW0 se desplaza a la izquierda tantas posiciones como indique la palabra de marcas MW2.

El resultado se deposita en la palabra doble de marcas Mw4. La salida A 4.0 se pone a "1".

11.1.5 SHR_W : Desplazar palabra a la derecha

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, L, D, T, Z	Entrada de habilitación
IN	WORD	E, A, M, L, D	Valor a desplazar
N	WORD	E, A, M, L, D	Número de posiciones (bits) a desplazar
OUT	WORD	E, A, M, L, D	Resultado de la operación de desplazamiento
ENO	BOOL	E, A, M, L, D	Salida de habilitación

Descripción

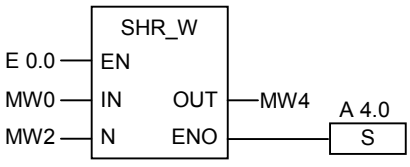
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Desplazar palabra a la derecha**. Esta operación desplaza los bits 0 a 15 de la entrada IN bit a bit a la derecha. Los bits 16 a 31 no son afectados. La entrada N indica el número de bits a desplazar. Si N es superior a 16, la operación escribe un 0 en la salida OUT y pone los bits A0 y OV de la palabra de estado a 0. Las posiciones que quedan libres a la izquierda se rellenan con ceros. El resultado de la operación de desplazamiento se determina consultando la salida OUT.

La operación activada pone los bits A0 y OV de la palabra de estado siempre a "0" cuando N es diferente de 0. El estado de señal de ENO es el mismo que el de EN.

Palabra de estado

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

Ejemplo



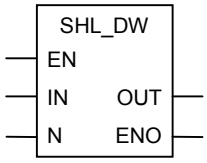
La operación se activa cuando E 0.0 = 1.

La palabra de marcas MW0 se desplaza a la derecha tantos bits como se hayan indicado en MW2.

El resultado se almacena en MW4. La salida A 4.0 se pone a ”1".

11.1.6 SHL_DW : Desplazar palabra doble a la izquierda

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, L, D, T, Z	Entrada de habilitación
IN	DWORD	E, A, M, L, D	Valor a desplazar
N	WORD	E, A, M, L, D	Número de posiciones (bits) a desplazar
OUT	DWORD	E, A, M, L, D	Resultado de la operación de desplazamiento
ENO	BOOL	E, A, M, L, D	Salida de habilitación

Descripción

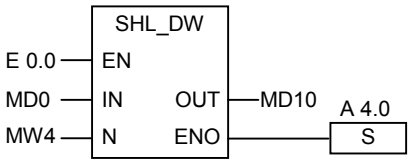
Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Desplazar palabra doble a la izquierda**. Esta operación desplaza los bits 0 a 31 de la entrada IN bit a bit a la izquierda. La entrada N indica el número de bits a desplazar. Si N es superior a 32, la operación escribe un 0 en la salida OUT y pone los bits A0 y OV de la palabra de estado a "0". Las posiciones que quedan libres a la derecha se rellenan con ceros. El resultado de la operación de desplazamiento se determina consultando la salida OUT.

La operación activada pone los bits A0 y OV de la palabra de estado a "0" cuando N es diferente de 0. El estado de señal de ENO es el mismo que el de EN.

Palabra de estado

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

Ejemplo



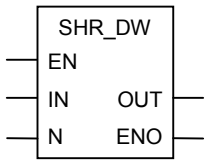
La operación se activa cuando E 0.0 = 1.

La palabra doble de marcas MD0 se desplaza a la izquierda tantos bits como se hayan indicado en MW4.

El resultado se almacena en MD10. La salida A 4.0 se pone a "1".

11.1.7 SHR_DW : Desplazar palabra doble a la derecha

Símbolo

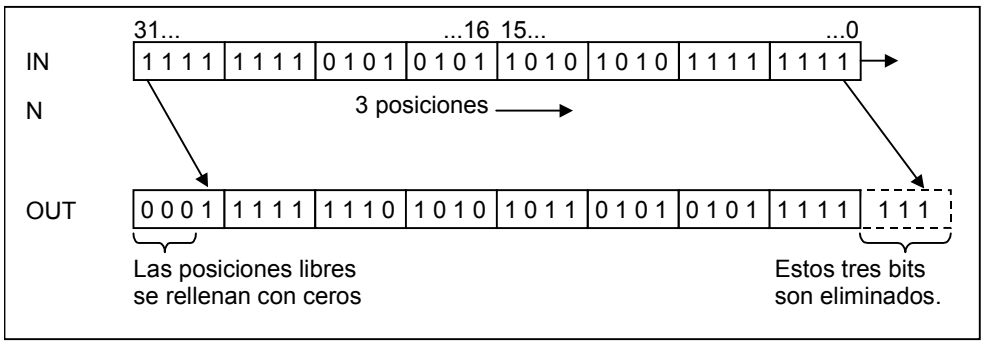


Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, L, D, T, Z	Entrada de habilitación
IN	DWORD	E, A, M, L, D	Valor a desplazar
N	WORD	E, A, M, L, D	Número de posiciones (bits) a desplazar
OUT	DWORD	E, A, M, L, D	Resultado de la operación de desplazamiento
ENO	BOOL	E, A, M, L, D	Salida de habilitación

Descripción

Un estado de señal de "1" en la entrada de habilitación (EN) activa la operación **Desplazar palabra doble a la derecha**. Esta operación desplaza los bits 0 a 31 de la entrada IN bit a bit a la derecha. La entrada N indica el número de bits a desplazar. Si N es superior a 32, la operación escribe un 0 en la salida OUT y pone los bits A0 y OV de la palabra de estado a "0". Las posiciones que quedan libres a la izquierda se rellenan con ceros. El resultado de la operación de desplazamiento se determina consultando la salida OUT.

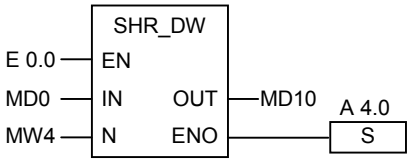
La operación activada pone el bit OV de la palabra de estado siempre a "0" cuando N es diferente de 0. El estado de señal de ENO es el mismo que el de EN.



Palabra de estado

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

Ejemplo



La operación se activa cuando E 0.0 = 1.

La palabra doble de marcas MD0 se desplaza a la derecha tantos bits como se hayan indicado en MW4.

El resultado se almacena en MD10. La salida A 4.0 se pone a ”1".

11.2 Operaciones de rotación

11.2.1 Lista de operaciones de rotación

Descripción

Las operaciones de rotación sirven para rotar bit a bit todo el contenido de la entrada IN, a la izquierda o a la derecha (consulte Registros de la CPU). Las posiciones vacantes de los bits se rellenan con los estados de señal de los bits que se desplazan fuera de la entrada IN.

El número que se introduce en el Parámetro de entrada N indica el número de bits que se va a rotar.

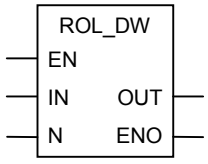
Dependiendo de la operación que se haya seleccionado, la rotación tiene lugar vía el bit A1 de la palabra de estado. El bit A0 de la palabra de estado se pone a "0".

Se dispone de las siguientes operaciones de rotación:

- ROL_DW : Rotar palabra doble a la izquierda
- ROR_DW : Rotar palabra doble a la derecha

11.2.2 ROL_DW : Rotar palabra doble a la izquierda

Símbolo

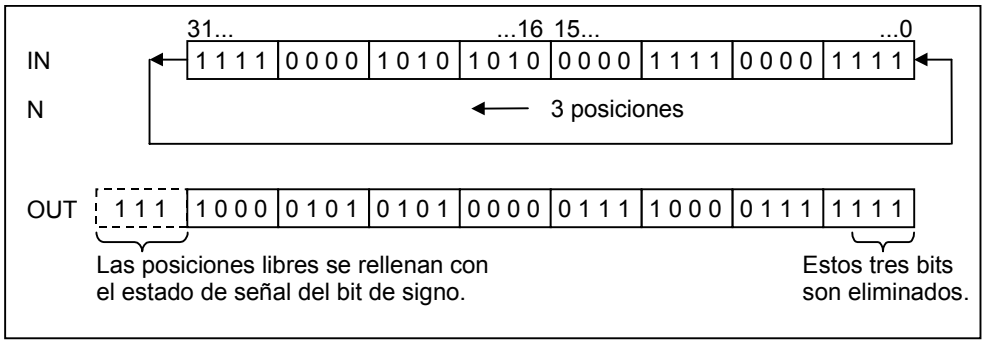


Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, L, D, T, Z	Entrada de habilitación
IN	DWORD	E, A, M, L, D	Valor a rotar
N	WORD	E, A, M, L, D	Número de posiciones (bits) a rotar
OUT	DWORD	E, A, M, L, D	Resultado de la operación de rotación
ENO	BOOL	E, A, M, L, D	Salida de habilitación

Descripción

Un estado de señal de 1 en la entrada de habilitación (EN) activa la operación **Rotar palabra doble a la izquierda**. Esta operación rota todo el contenido de la salida IN bit a bit a la izquierda. La entrada N indica el número de bits a rotar. Si N es superior a 32, la palabra doble rota (((N-1) módulo 32) + 1) posiciones. Las posiciones que quedan libres a la derecha se rellenan con los estados de señal de los bits rotados. El resultado de la operación de rotación se determina consultando la salida OUT.

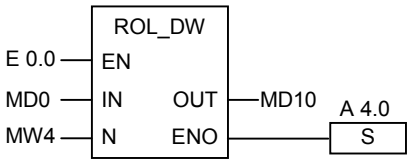
La operación activada pone los bits A0 y OV de la palabra de estado a "0" cuando N es diferente de 0. El estado de señal de ENO es el mismo que el de EN.



Palabra de estado

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

Ejemplo



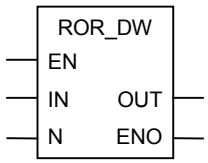
La operación se activa cuando E 0.0 = 1.

La palabra doble de marcas MD0 se rota a la izquierda tantos bits como se hayan indicado en MW4.

El resultado se almacena en MD10. La salida A 4.0 se pone a ”1".

11.2.3 ROR_DW : Rotar palabra doble a la derecha

Símbolo

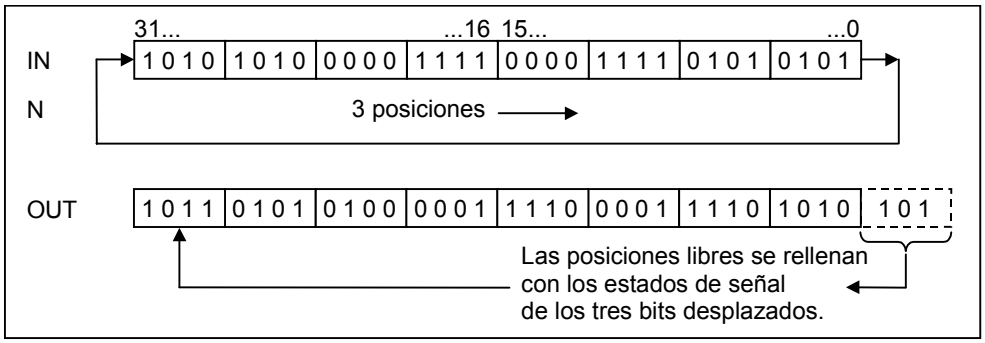


Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, L, D, T, Z	Entrada de habilitación
IN	DWORD	E, A, M, L, D	Valor a rotar
N	WORD	E, A, M, L, D	Número de posiciones (bits) a rotar
OUT	DWORD	E, A, M, L, D	Resultado de la operación de rotación
ENO	BOOL	E, A, M, L, D	Salida de habilitación

Descripción

Un estado de señal de 1 en la entrada de habilitación (EN) activa la operación Rotar palabra doble a la derecha. Esta operación hace rotar todo el contenido de la entrada IN bit a bit a la derecha. La entrada N indica el número de bits a rotar. Si N es superior a 32, la palabra doble rota (((N-1) modulo 32) +1) posiciones. Las posiciones que quedan libres a la izquierda se rellenan con los estados de señal de los bits rotados. El resultado de la operación de rotación se determina consultando la salida OUT.

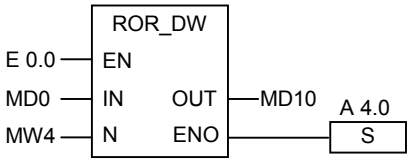
La operación activada pone los bits A0 y OV de la palabra de estado a "0" cuando N es diferente de 0. El estado de señal de ENO es el mismo que el de EN.



Palabra de estado

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

Ejemplo



La operación se activa cuando E 0.0 = 1.

La palabra doble de marcas MD0 se rota a la derecha tantos bits como se hayan indicado en MW4.

El resultado se almacena en MD10. La salida A 4.0 se pone a ”1".

12 Operaciones con bits de la palabra de estado

12.1 Lista de operaciones con bits de la palabra de estado

Descripción

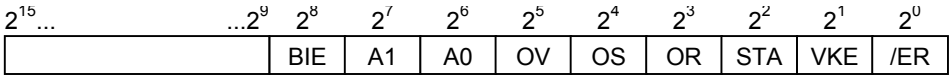
Las operaciones con bits de la palabra de estado son operaciones lógicas, que trabajan con los bits de la palabra de estado. Estas operaciones reaccionan ante una de las condiciones expuestas a continuación, representadas por uno o más bits de la palabra de estado:

- El bit de resultado binario (RB) está activado (es decir, su estado de señal es "1").
- El resultado de una función aritmética en relación a 0 puede ser: == 0, <> 0, > 0, < 0, >= 0, <= 0.
- El resultado de una función aritmética no es admisible (UO).
- Una función aritmética ha causado un desbordamiento (OV) o un desbordamiento memorizado (OS).

En una Y lógica, las operaciones con bits de la palabra de estado combinan el resultado de su consulta con el resultado lógico precedente según la tabla de verdad Y. Cuando se trata de una O lógica la combinación se realiza conforme a la tabla de verdad O.

Palabra de estado

La palabra de estado es un registro localizado en la memoria de la CPU; este registro contiene bits que pueden direccionarse en los operandos de las operaciones lógicas con bits y con palabras. En la siguiente figura se muestra la estructura de la palabra de estado.

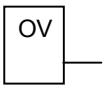


Los bits de la palabra de estado se pueden evaluar

- en operaciones en coma fija,
- en operaciones aritméticas.

12.2 OV : Bit de anomalía "desbordamiento"

Símbolo



Descripción

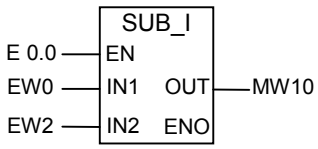
La operación **Bit de anomalía "desbordamiento"** se utiliza para detectar un desbordamiento (OV) en la función aritmética ejecutada en último lugar. Si tras una función aritmética el resultado se encuentra fuera de las áreas negativa o positiva admisibles, se activa el bit OV de la palabra de estado (v. apto. Registros de la CPU). La operación consulta el estado de señal de este bit. Este bit se desactiva una vez eliminada la causa del error. Operaciones aritméticas ejecutadas sin error desactivan este bit.

Palabra de estado

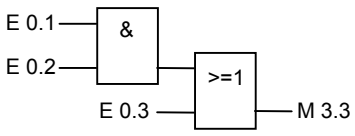
	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	X	X	X	1

Ejemplo

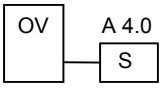
Segmento 1



Segmento



Segmento 3



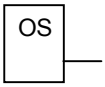
El cuadro SUB_I se activa cuando E 0.0 =1. Si el resultado de la operación aritmética EW0 - EW2 queda fuera del área válida para un número entero, entonces se activa el bit de desbordamiento OV.

La consulta en OV sobre el estado de señal da como resultado 1. La salida A 4.0 se activa si la consulta en OV es 1 y el RLO del segmento 2 también es 1 (es decir, si el RLO anterior a la salida A 4.0 = 1).

Si el estado de señal de E 0.0 = 0 (desactivada), el estado de señal de EN y de ENO es 0. Si el estado de señal de EN = 1 (activado) y el resultado de la operación aritmética queda fuera del área válida, entonces el estado de señal de ENO = 0.

12.3 OS : Bit de anomalía "desbordamiento memorizado"

Símbolo



Descripción

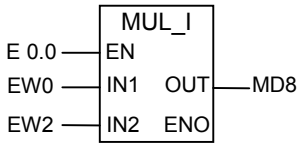
La operación **Bit de anomalía "desbordamiento memorizado"** se utiliza para detectar un desbordamiento permanente (desbordamiento memorizado, OS) en una función aritmética. Si al terminar la función aritmética el resultado se encuentra fuera de las áreas negativa o positiva admisibles, se activa el bit OS de la palabra de estado (v. apto. Registros de la CPU). La operación consulta el estado de señal de este bit. A diferencia del bit OV (bit de desbordamiento), el bit OS permanece activado incluso tras la ejecución sin error de operaciones.

Palabra de estado

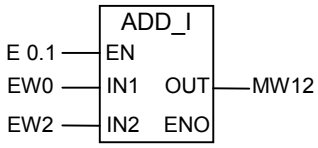
	RB	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLO**	**/ER**
se escribe	-	-	-	-	-	X	X	X	1

Ejemplo

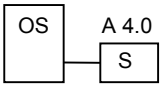
Segmento 1



Segmento 2



Segmento 3



El cuadro MUL_I se activa cuando E 0.0 =1 y se activa el cuadro BOX_ADD, si E 0.1 = 1. Si uno de los dos resultados queda fuera del área válida para un número entero, entonces se activa el bit de desbordamiento memorizado OS.

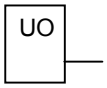
La consulta del estado de señal en OS da como resultado 1 y la salida A 4.0 se activa.

Segmento 1: Si el estado de señal de E 0.0 = 0 (desactivada), el estado de señal de EN y de ENO es 0. Si el estado de señal de EN = 1 (activado) y el resultado de la operación aritmética queda fuera del área válida, entonces el estado de señal de ENO = 0.

Segmento 2: Si el estado de señal de E 0.1 = 0 (desactivada), el estado de señal de EN y de ENO es 0. Si el estado de señal de EN = 1 (activado) y el resultado de la operación aritmética queda fuera del área válida, entonces el estado de señal de ENO = 0.

12.4 UO : Bit de anomalía "operación no válida"

Símbolo



Descripción

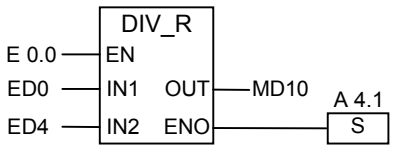
La operación **Bit de anomalía "operación no válida"** se utiliza para determinar si el resultado de una función aritmética de números coma flotante es o no admisible (es decir, si uno de los valores de la función aritmética no es un número en coma flotante válido). Para ello se evalúan los códigos de condición de la palabra de estado (A1 y A0). Si el resultado de la operación aritmética no es válida, la consulta sobre el estado de la señal produce "1". Si la combinación en A1 y A0 no indica "no válida", el resultado de la consulta es "0".

Palabra de estado

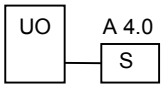
	RB	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**RLO**	**/ER**
se escribe	-	-	-	-	-	X	X	X	1

Ejemplo

Segmento 1



Segmento 2



El cuadro DIV_R se activa cuando E 0.0 =1. Si el valor de ED0 o de ED4 no es un número en coma flotante válido, la operación aritmética no es válida.

Si el estado de señal de EN = 1 (activado) y se produce un error mientras se está ejecutando la función DIV_R; entonces el estado de señal de ENO = 0.

La salida A 4.0 se activa cuando la función DIV_R se ha ejecutado pero uno de los valores de la operación aritmética no es un número en coma flotante válido. Si el estado de señal de la entrada E 0.0 = 0 (desactivada), el estado de señal de EN y de ENO es "0".

12.5 RB : Bit de anomalía "registro RB"

Símbolo

Inglés **Alemán**

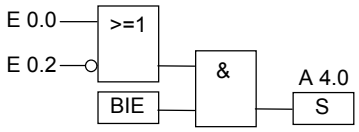
Descripción

La operación **Bit de anomalía "registro RB"** sirve para consultar el estado de señal del bit RB (resultado binario).

Palabra de estado

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

Ejemplo



La salida A 4.0 se activa cuando E 0.0 =1 O E 0.2 = 0 y, además de este RLO, el estado de señal del bit RB = 1.

12.6 <> 0 : Bits de resultado

Símbolo

== 0 —	La operación **Bit de resultado igual a 0** comprueba si el resultado de una función aritmética es igual a 0.
<> 0 —	La operación **Bit de resultado diferente de 0** comprueba si el resultado de una función aritmética es distinto que 0.
> 0 —	La operación **Bit de resultado mayor que 0** determina si el resultado de una función aritmética es mayor que 0.
< 0 —	La operación **Bit de resultado menor que 0** determina si el resultado de una función aritmética es menor que 0.
>= 0 —	La operación **Bit de resultado mayor o igual a 0** determina si el resultado de una función aritmética es mayor o igual a 0.
<= 0 —	La operación **Bit de resultado menor o igual a 0** determina si el resultado de una función aritmética es menor o igual a 0.

Descripción

Las operaciones **Bits de resultado** se utilizan para determinar la relación entre el resultado de una función aritmética y 0, es decir, si el resultado es == 0, <> 0, > 0, < 0, >= 0 o <= 0. Para ello se evalúan los códigos de condición de la palabra de estado (A1 y A0). Si se cumple la condición indicada en el operando, el resultado de la consulta del estado de señal es "1".

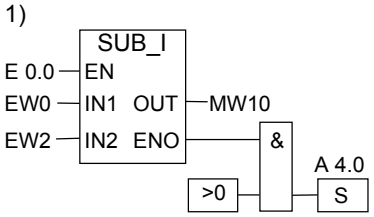
En una Y lógica, las operaciones con bits de la palabra de estado combinan el resultado de su consulta con el resultado lógico precedente según la tabla de verdad Y. Cuando se trata de una O lógica la combinación se realiza conforme a la tabla de verdad O.

Palabra de estado

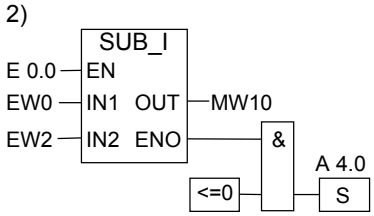
	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	-	-	-	-	-	X	X	X	1

Ejemplo

1)



2)



El cuadro SUB_I se activa cuando E 0.0 =1. Si el valor de EW0 es mayor que el valor de EW2, el resultado de la operación aritmética EW0 - EW2 es mayor que 0. Si el estado de señal de EN = 1 y se produce un error mientras se está ejecutando la función SUB_I; entonces el estado de señal de ENO = 0.

1) La salida A 4.0 se activa cuando la función se ha ejecutado correctamente y el resultado es mayor que 0. Si el estado de señal de la entrada E 0.0 = 0 (desactivada), el estado de señal de EN y de ENO es "0".

2) La salida A 4.0 se activa cuando la función se ha ejecutado correctamente y el resultado es menor o igual que 0. Si el estado de señal de la entrada E 0.0 = 0 (desactivada), el estado de señal de EN y de ENO es "0".

13 Operaciones de temporización

13.1 Lista de operaciones de temporización

Descripción

Bajo "Posición de un temporizador en la memoria y sus componentes" encontrará información sobre cómo ajustar y seleccionar los temporizadores.

Se dispone de las siguientes operaciones de temporización:

- S_IMPULS : Parametrizar y arrancar temporizador como impulso
- S_VIMP : Parametrizar y arrancar temporizador como impulso prolongado
- S_EVERZ : Parametrizar y arrancar temporizador como retardo a la conexión
- S_SEVERZ : Parametrizar y arrancar temporizador como retardo a la conexión con memoria
- S_AVERZ : Parametrizar y arrancar temporizador como retardo a la desconexión

- SI : Arrancar temporizador como impulso
- SV : Arrancar temporizador como impulso prolongado
- SE : Arrancar temporizador como retardo a la conexión
- SS : Arrancar temporizador como retardo a la conexión con memoria
- SA : Arrancar temporizador como retardo a la desconexión

13.2 Posición de un temporizador en la memoria y sus componentes

Area de memoria

Los temporizadores tienen un área reservada en la memoria de la CPU. Este área de memoria reserva una palabra de 16 bits para cada operando de temporizador. La programación con FUP asiste 256 temporizadores. Consulte los datos técnicos de la CPU para saber de cuántas palabras de temporización dispone ésta.

Las siguientes funciones tienen acceso al área de memoria de temporizadores:

- Operaciones de temporización
- Actualización por reloj de palabras de temporización. Esta función de su CPU en el estado RUN decrementa en una unidad un valor de temporización dado con el intervalo designado por la base de tiempo hasta alcanzar el valor "0".

Valor de temporización

Los bits 0 a 9 de la palabra de temporización contienen el valor de temporización en código binario. Este valor indica un número de unidades. La actualización decrementa el valor de temporización en una unidad con el intervalo designado por la base de tiempo hasta alcanzar el valor "0". Se puede cargar un valor de temporización en formato dual, hexadecimal o decimal codificado binario (BCD). El margen de temporización puede variar de 0 a 9 990 segundos.

Para cargar un valor de temporización predefinido, se observarán las siguientes reglas sintácticas.

- S5T#a**H**_b**M**_c**S**_d**MS**
  - H (horas), M (minutos), S (segundos), MS (milisegundos);
    a, b, c, d los define el usuario.
  - La base de tiempo se selecciona automáticamente y el valor de temporización se redondea al próximo número inferior con esa base de tiempo.

El valor de temporización máximo que puede introducirse es 9 900 segundos ó 2H_46M_30S.
Ejemplos:
S5T#4s --> 4 segundos
S5T#1h_15m --> 1 hora y15 minuto
S5T#2h_46m_30s-->2 horas, 46 minutos y 30 segundos

Base de tiempo

Los bits 12 y 13 de la palabra de temporización contienen la base de tiempo en código binario. La base de tiempo define el intervalo en que se decrementa en una unidad el valor de temporización. La base de tiempo más pequeña es de 10 ms, la más grande de 10 s.

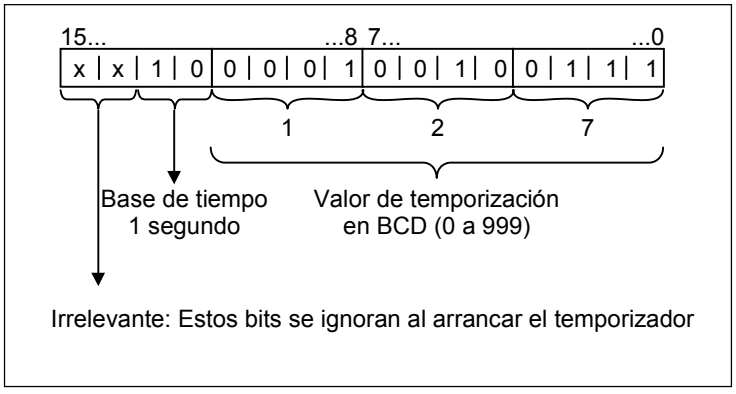
Base de tiempo	Base de tiempo en código binario
10 ms	00
100 ms	01
1 s	10
10 s	11

Puesto que los valores de temporización se almacenan con sólo un intervalo de tiempo, los valores que no son exactamente múltiplos de un intervalo de tiempo se truncan. Los valores cuya resolución es demasiado alta para el área deseada se redondean para alcanzar el área deseada aunque no la resolución deseada. La siguiente tabla muestra las resoluciones posibles y las áreas correspondientes.

Resolución	Base de tiempo
0,01 segundos	10MS a 9S_990MS
0,1 segundos	100MS a 1M_39S_900MS
1 segundo	1S a 16M_39S
10 segundos	10S a 2HR_46M_30S

Configuración en la célula de temporización

Cuando se dispara un temporizador, el contenido de la célula de temporización 1 se utiliza como valor de temporización. Los bits 0 a 11 de la célula de temporización almacenan el valor de temporización en formato decimal codificado en binario (formato BCD: cada grupo de cuatro bits contiene el código binario de un valor decimal). Los bits 12 a 13 almacenan la base de tiempo en código binario. La siguiente figura muestra el contenido de la célula de temporización cargado con el valor de temporización 127 y una base de tiempo de 1 segundo.

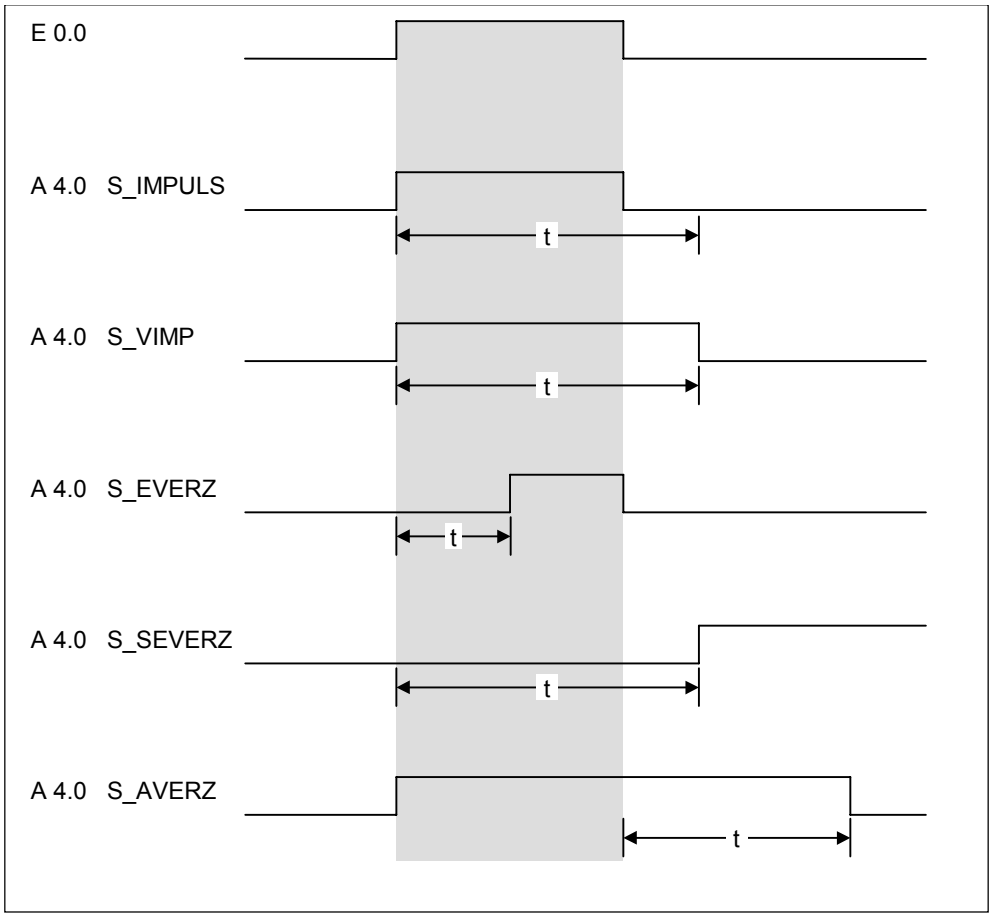


Leer el temporizador y la base de tiempo

Todos los cuadros de temporizadores tienen dos salidas, DUAL y DEZ, para las que se puede indicar una dirección de palabra. En la salida DUAL el valor de temporización está codificado en formato binario, no indicándose la base de tiempo. En la salida DEZ la base de tiempo y el valor de temporización están codificados en formato decimal codificado en binario (BCD).

Elegir el temporizador apropiado

El resumen breve de los cinco tipos de temporizadores sirve de ayuda para la elección del temporizador que se adapte mejor a sus necesidades.

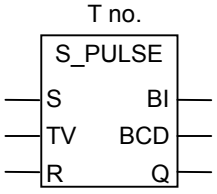


Temporizadores	Descripción
S_IMPULS Temporizador de impulso	El tiempo máximo que la señal de salida permanece a 1 corresponde al valor de temporización t programado. La señal de salida permanece a 1 durante un tiempo inferior si la señal de entrada cambia a 0.
S_VIMP Temporizador de impulso prolongado	La señal de salida permanece a 1 durante el tiempo programado, independientemente del tiempo en que la señal de entrada esté a 1.
S_EVERZ Temporizador de retardo a la conexión	La señal de salida es 1 solamente si ha finalizado el tiempo programado y la señal de entrada sigue siendo 1.
S_SEVERZ Temporizador de retardo a la conexión con memoria	La señal de salida cambia de 0 a 1 solamente si ha finalizado el tiempo programado, independientemente del tiempo en que la señal de salida esté a 1.
S_AVERZ Temporizador de retardo a la desconexión	La señal de salida es 1 cuando la señal de entrada es 1 o cuando el temporizador está en marcha. El temporizador arranca cuando la señal de entrada cambia de 1 a 0.

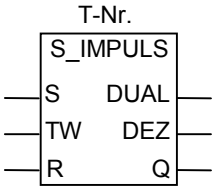
13.3 S_IMPULS : Parametrizar y arrancar temporizador como impulso

Símbolo

Inglés



Alemán



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Area de memoria	Descripción
Nº	Nº	TIMER	T	Número de identificación del temporizador. El área depende de la CPU utilizada.
S	S	BOOL	E, A, M, D, L, T, Z	Entrada de arranque
TV	TW	S5TIME	E, A, M, D, L o constante	Valor de temporización preseleccionado (margen: 0 - 999)
R	R	BOOL	E, A, M, D, L, T, Z	Entrada de desactivación
BI	DUAL	WORD	E, A, M, D, L	Tiempo restante (formato de entero)
BCD	DEZ	WORD	E, A, M, D, L	Tiempo restante (formato BCD)
Q	Q	BOOL	E, A, M, D, L	Estado del temporizador

Descripción

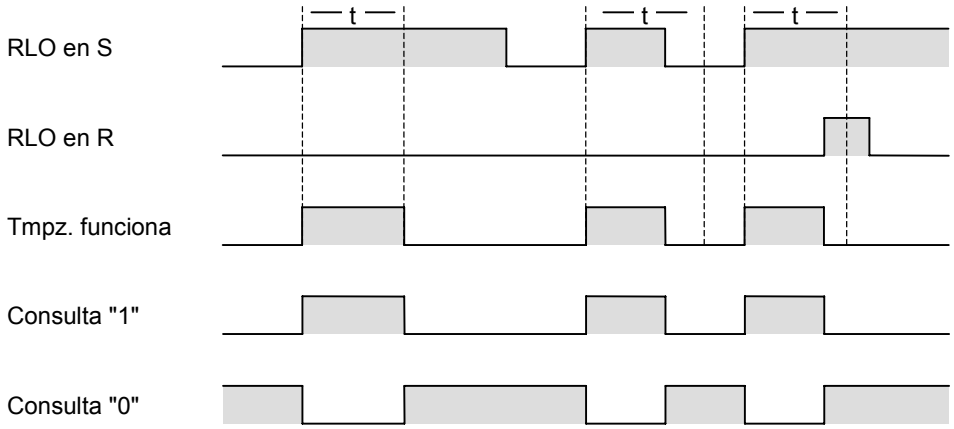
La operación **Parametrizar y arrancar temporizador como impulso** arranca un determinado temporizador cuando se produce un flanco positivo (es decir, cuando el estado de señal cambia de "0" a "1") en la entrada Activar (S). Para habilitar un temporizador tiene que producirse necesariamente un cambio de señal. El temporizador continúa funcionando con el tiempo indicado en la entrada Valor de temporización (TW) hasta que el tiempo programado transcurra y mientras que el estado de señal de la entrada S sea "1". Mientras el temporizador está en marcha, la consulta sobre si el estado de señal de la salida Q es "1" da un "1" como resultado. Si el estado de señal de la entrada S cambia de "1" a "0" antes de finalizar el tiempo, el temporizador se para. En este caso, la consulta si el estado de señal de la salida Q es "1" produce un "0" como resultado.

Si el temporizador está en marcha y el estado de señal de la entrada Desactivar (R) cambia de "0" a "1", entonces se desactiva el temporizador, es decir, se pone a "0". Este cambio también pone el valor de temporización y la base de tiempo a "0". Un estado de señal de "1" en la entrada R del temporizador no tiene efecto alguno si el temporizador no está en marcha.

El valor de temporización actual puede determinarse consultando las salidas DUAL y DEZ. El valor de DUAL es en código binario; el de DEZ es en formato decimal codificado en binario.

Diagrama de temporización

Características del temporizador como impulso:

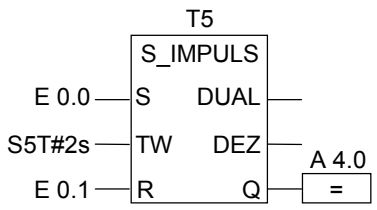


t = tiempo programado

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
Ecriture	-	-	-	-	-	X	X	X	1

Ejemplo

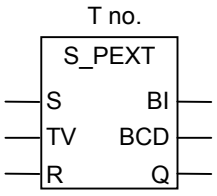


Si el estado de señal de la entrada E 0.0 cambia de ”0" a ”1" (flanco positivo en el RLO) se activa el temporizador T5. El temporizador continúa en marcha con el tiempo especificado de 2 segundos mientras la entrada E 0.0 =1. Si el estado de señal de la entrada E 0.0 cambia de ”1" a ”0" antes de que hayan transcurrido los 2 segundos, el temporizador se detiene. Si el estado de señal de la entrada E 0.1 cambia de ”0" a ”1" mientras el temporizador está en marcha, el temporizador se pone a 0. El estado de señal de la salida A 4.0 se mantiene en ”1" mientras funcione el temporizador.

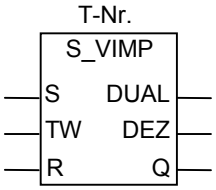
13.4 S_VIMP : Parametrizar y arrancar temporizador como impulso prolongado

Símbolo

Inglés



Alemán



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Area de memoria	Descripción
Nº	Nº	TIMER	T	Número de identificación del temporizador. El área depende de la CPU utilizada.
S	S	BOOL	E, A, M, D, L, T, Z	Entrada de arranque
TV	TW	S5TIME	E, A, M, D, L o constante	Valor de temporización preseleccionado (margen: 0 - 999)
R	R	BOOL	E, A, M, D, L, T, Z	Entrada de desactivación
BI	DUAL	WORD	E, A, M, D, L	Tiempo restante (formato de entero)
BCD	DEZ	WORD	E, A, M, D, L	Tiempo restante (formato BCD)
Q	Q	BOOL	E, A, M, D, L	Estado del temporizador

Descripción

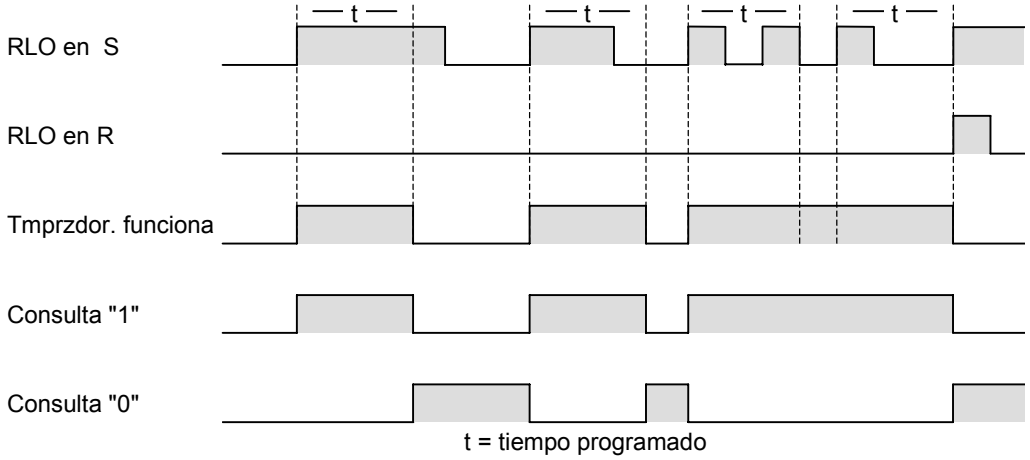
La operación **Parametrizar y arrancar temporizador como impulso prolongado** arranca el temporizador especificado cuando se produce un flanco positivo en la entrada Activar (S), es decir, cuando el estado de señal cambia de ”0" a ”1". Para habilitar el temporizador tiene que producirse necesariamente un cambio de señal. El temporizador continúa en marcha con el tiempo indicado en la entrada Valor de temporización (TW) si el estado de señal de la entrada S cambia a 0 antes de que finalice el tiempo. La consulta sobre si el estado de señal de la salida Q es 1 es 1 mientras el temporizador esté en marcha. El temporizador arranca nuevamente con el tiempo indicado si el estado de señal de la entrada S cambia de 0 a 1 mientras el temporizador está en marcha.

Si se produce un cambio de "0" a "1" en la entrada del temporizador Desactivar (R) mientras el temporizador está funcionando, éste se pone a "0". Este cambio también pone el valor de temporización y la base de tiempo a "0".

La temporización actual puede determinarse consultando las salidas DUAL y DEZ. El valor de DUAL es en código binario; el valor de DEZ es en formato decimal codificado en binario.

Diagrama de temporización

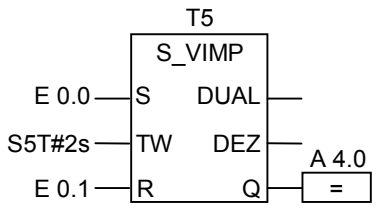
Características del temporizador como impulso prolongado:



Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
Ecriture	-	-	-	-	-	X	X	X	1

Ejemplo

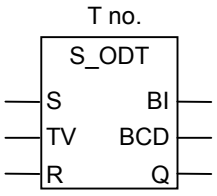


Si el estado de señal de la entrada E 0.0 cambia de ”0" a ”1" (flanco positivo en el RLO) se arranca el temporizador T5. El temporizador continúa en marcha, independientemente de que haya un flanco negativo en la entrada S, con el tiempo especificado de 2 segundos. Si el estado de señal de la entrada E 0.0 cambia de ”1" a ”0" antes de que hayan transcurrido los 2 segundos, el temporizador arranca de nuevo. Si el estado de señal de la entrada E 0.1 cambia de ”0" a ”1" mientras el temporizador está en marcha, el temporizador se arranca de nuevo. El estado de señal de la salida A 4.0 se mantiene en ”1" mientras funcione el temporizador.

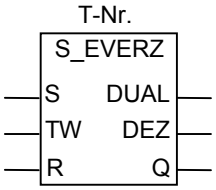
13.5 S_EVERZ : Parametrizar y arrancar temporizador como retardo a la conexión

Símbolo

Inglés



Alemán



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Area de memoria	Descripción
Nº	Nº	TIMER	T	Número de identificación del temporizador. El área depende de la CPU utilizada.
S	S	BOOL	E, A, M, D, L, T, Z	Entrada de arranque
TV	TW	S5TIME	E, A, M, D, L o constante	Valor de temporización preseleccionado (margen: 0 - 999)
R	R	BOOL	E, A, M, D, L, T, Z	Entrada de desactivación
BI	DUAL	WORD	E, A, M, D, L	Tiempo restante (formato de entero)
BCD	DEZ	WORD	E, A, M, D, L	Tiempo restante (formato BCD)
Q	Q	BOOL	E, A, M, D, L	Estado del temporizador

Descripción

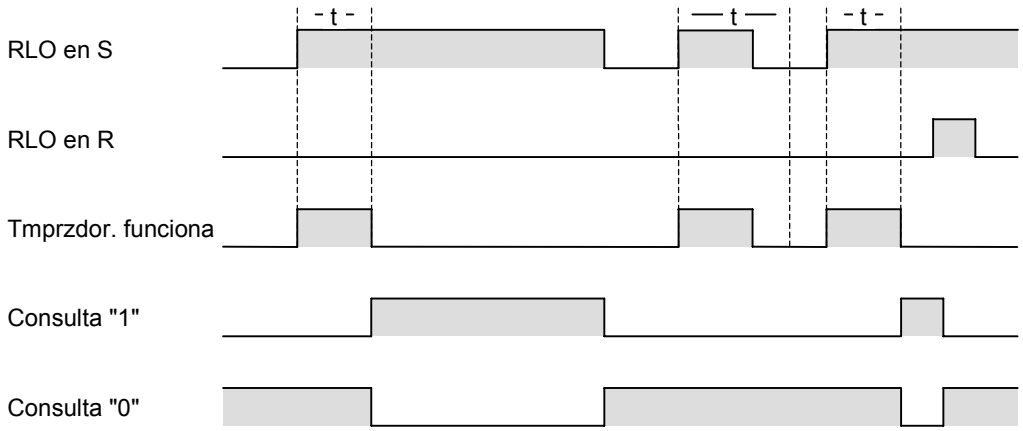
La operación **Parametrizar y arrancar temporizador como retardo a la conexión** arranca un temporizador determinado si se produce un flanco positivo (es decir, si el estado de señal cambia de "0" a "1") en la entrada Activar (S). Para habilitar un temporizador se tiene que producirse necesariamente un cambio de señal. El temporizador continúa funcionando con el tiempo indicado en la entrada Valor de temporización (TW) mientras el estado de señal de la entrada S sea "1". La consulta si el estado de señal de la salida Q es "1" produce un "1" como resultado si el tiempo ha transcurrido, si el estado de señal de la entrada S sigue siendo "1" y si la entrada Desactivar (R) se mantiene a "0". Si el estado de señal de la entrada S cambia de "1" a "0" mientras está en marcha el temporizador, éste se para. En este caso, la consulta sobre si el estado de señal es "1" produce un "0" como resultado.

Cuando la entrada Desactivar (R) cambia de "0" a "1", el temporizador se desactiva. Este cambio también pone el valor de temporización y la base de tiempo a "0". El temporizador también se borra si el estado señal de la entrada R es "1" mientras el temporizador está parado.

El valor actual se determina consultado las salidas DUAL y DEZ. El valor de temporización de DUAL está codificado en formato binario; el de DEZ está en formato BCD.

Diagrama de temporización

Características del temporizador de retardo a la conexión:

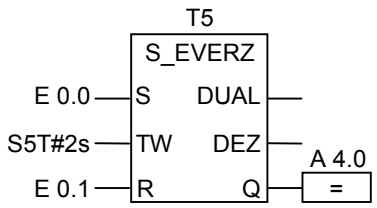


t = tiempo programado

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
Ecriture	-	-	-	-	-	X	X	X	1

Ejemplo

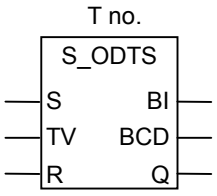


Si el estado de señal de la entrada E 0.0 cambia de ”0" a ”1" (flanco positivo en el RLO) se arranca el temporizador. Si transcurre el tiempo especificado de 2 segundos (2s) y el estado de señal de la entrada E 0.0 sigue siendo ”1", el estado de señal de la salida A 4.0 es ”1". Si cambia el estado de señal de E 0.0 de ”1" a ”0", se detiene el temporizador y A 4.0 es ”0". Si el estado de señal de E 0.0 cambia de ”0" a ”1" mientras está transcurriendo el tiempo especificado, el temporizador se arranca de nuevo.

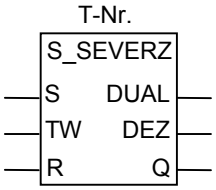
13.6 S_SEVERZ : Parametrizar y arrancar temporizador como retardo a la conexión con memoria

Símbolo

Inglés



Alemán



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Area de memoria	Descripción
Nº	Nº	TIMER	T	Número de identificación del temporizador. El área depende de la CPU utilizada.
S	S	BOOL	E, A, M, D, L, T, Z	Entrada de arranque
TV	TW	S5TIME	E, A, M, D, L o constante	Valor de temporización preseleccionado (margen: 0 - 999)
R	R	BOOL	E, A, M, D, L, T, Z	Entrada de desactivación
BI	DUAL	WORD	E, A, M, D, L	Tiempo restante (formato de entero)
BCD	DEZ	WORD	E, A, M, D, L	Tiempo restante (formato BCD)
Q	Q	BOOL	E, A, M, D, L	Estado del temporizador

Descripción

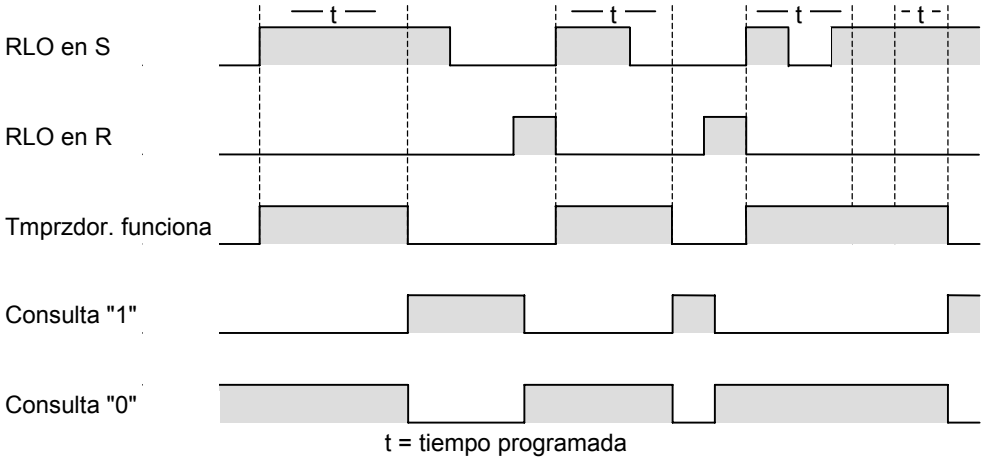
La operación **Parametrizar y arrancar temporizador como retardo a la conexión con memoria** arranca un determinado temporizador si se produce un flanco positivo (es decir, si el estado de la señal cambia de "0" a "1") en la entrada Activar (S). Para habilitar un temporizador se tiene que producir necesariamente un cambio del estado de señal. El temporizador continúa funcionando con el tiempo indicado en la entrada Valor de temporización (TW) si el estado de señal de la entrada S cambie a "0" antes de finalizar el tiempo. La consulta sobre si el estado de señal de la salida Q es "1" produce un resultado de "1" al finalizar el tiempo, independientemente del estado de señal de la entrada S, si la entrada Desactivar (R) se mantiene a "0". El temporizador arranca nuevamente con el tiempo indicado si el estado de señal de la entrada S cambia de "0" a "1" mientras está funcionando el temporizador.

Si el estado de señal de la entrada del temporizador Desactivar (R) cambia de "0" a "1", el temporizador se pone a "0" independientemente del RLO de la entrada S.

El valor de temporización actual puede determinarse consultando las salidas DUAL y DEZ. El valor de DUAL está en formato binario; el valor de DEZ está en formato BCD.

Diagrama de temporización

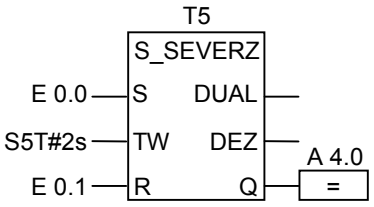
Características del temporizador de retardo a la conexión con memoria:



Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
Ecriture	-	-	-	-	-	X	X	X	1

Ejemplo

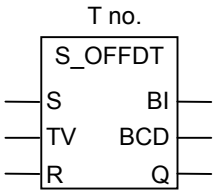


Si el estado de señal de la entrada E 0.0 cambia de ”0" a ”1" (flanco positivo en el RLO) se arranca el temporizador T5. El temporizador continúa en marcha, independientemente de que en la entrada E 0.0 se produzca un cambio de señal de ”1" a ”0". Si el estado de señal de la entrada E 0.0 cambia de ”0" a ”1" antes de que haya transcurrido el tiempo indicado, el temporizador arranca de nuevo. Si el estado de señal de la entrada E 0.0 cambia de ”0" a ”1" mientras el temporizador está en marcha, el temporizador se arranca de nuevo. El estado de señal de la salida A 4.0 es ”1" si ha transcurrido el tiempo y la entrada E 0.1 se mantiene en ”0".

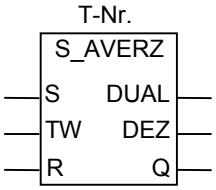
13.7 S_AVERZ : Parametrizar y arrancar temporizador como retardo a la desconexión

Símbolo

Inglés



Alemán



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Area de memoria	Descripción
Nº	Nº	TIMER	T	Número de identificación del temporizador. El área depende de la CPU utilizada.
S	S	BOOL	E, A, M, D, L, T, Z	Entrada de arranque
TV	TW	S5TIME	E, A, M, D, L o constante	Valor de temporización preseleccionado (margen: 0 - 999)
R	R	BOOL	E, A, M, D, L, T, Z	Entrada de desactivación
BI	DUAL	WORD	E, A, M, D, L	Tiempo restante (formato de entero)
BCD	DEZ	WORD	E, A, M, D, L	Tiempo restante (formato BCD)
Q	Q	BOOL	E, A, M, D, L	Estado del temporizador

Descripción

La operación **Parametrizar y arrancar temporizador como retardo a la desconexión** arranca un determinado temporizador cuando se produce un flanco negativo en la entrada Activar (S). Para habilitar un temporizador se tiene que producir necesariamente un cambio del estado de señal. La consulta sobre si el estado de señal de la salida Q es "1" produce un "1" como resultado si el estado de señal de la entrada S es "1" o si el temporizador está en marcha. El temporizador se pone a "0" si el estado de señal de la entrada S cambia de "0" a "1" mientras está funcionando el temporizador. El temporizador no arranca de nuevo hasta que el estado de señal de la entrada S cambie nuevamente de "1" a "0".

Si el estado de señal de la entrada del temporizador Desactivar (R) cambia de "0" a "1" mientras el temporizador está funcionando, éste se pone a "0".

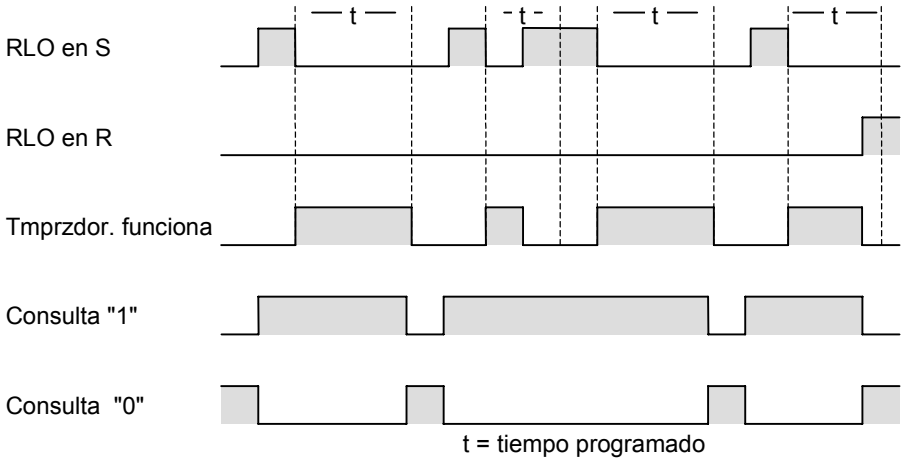
El valor de temporización actual puede determinarse consultando las salidas DUAL y DEZ. El valor de DUAL está en formato binario; el valor de DEZ está en formato BCD.

Palabra de estado

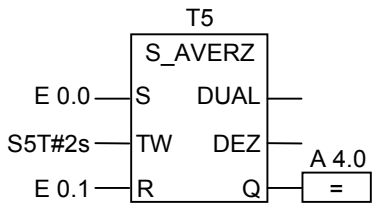
	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
Ecriture	-	-	-	-	-	X	X	X	1

Diagrama de temporización

Características del temporizador como retardo a la desconexión:



Ejemplo

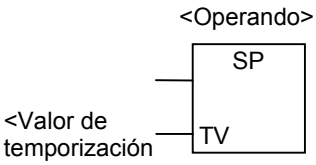


Si el estado de señal de la entrada E 0.0 cambia de ”1" a ”0" se arranca el temporizador T5. La salida A 4.0 es ”1" cuando E 0.0 =1 o mientras el temporizador está en marcha. Si el estado de señal de la entrada E 0.0 cambia de ”0" a ”1" mientras está transcurriendo el tiempo, el temporizador se pone a ”0".

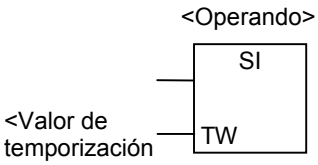
13.8 SI : Arrancar temporizador como impulso

Símbolo

Inglés



Alemán



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Area de memoria	Descripción
Número del temporizador	Número del temporizador	TIMER	T	El operando indica el número del temporizador que se debe arrancar.
TV	TW	S5TIME	E, A, M, D, L o constante	Valor de temporización (formato S5TIME)

Descripción

La operación **Arrancar temporizador como impulso** arranca un temporizador con un valor determinado, siempre y cuando el RLO disponga de un flanco ascendente (cambio de "0" a "1"). Mientras el RLO sea positivo, el temporizador se seguirá ejecutando con el valor indicado. La consulta sobre si el estado de señal es "1" produce un "1" mientras esté funcionando el temporizador. Si el RLO cambia de "1" a "0" mientras esté funcionando el temporizador, éste se detendrá. En este caso la consulta sobre si el estado de señal es "1" produce un "0".

El cuadro **Arrancar temporizador como impulso** sólo se puede disponer en el extremo derecho de la cadena de conexión. Sin embargo, puede utilizar varios cuadros **Arrancar temporizador como impulso**.

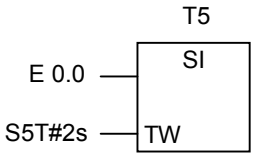
Consulte también Posición de un temporizador en la memoria y sus componentes.

Palabra de estado

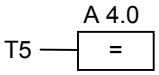
	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
Ecriture	-	-	-	-	-	0	-	-	0

Ejemplo

Segmento 1



Segmento 2



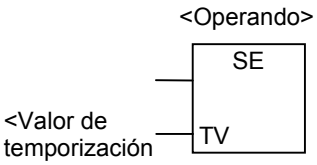
Si el estado de la señal en E 0.0 cambia de "0" a "1" (flanco positivo en el RLO), se arrancará el temporizador T5. Mientras el estado de la señal sea 1, el temporizador funcionará con el valor preajustado de 2 segundos. Si el estado de la señal en E 0.0 cambia de "1" a "0" mientras esté funcionando el temporizador, éste se detendrá.

Mientras funcione el temporizador, el estado de la señal en la salida A 4.0 = 1.

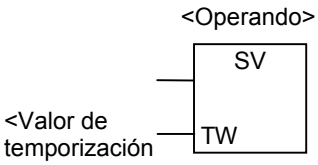
13.9 SV : Arrancar temporizador como impulso prolongado

Símbolo

Inglés



Alemán



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Area de memoria	Descripción
Número del temporizador	Número del temporizador	TIMER	T	El operando indica el número del temporizador que se debe arrancar.
TV	TW	S5TIME	E, A, M, D, L o constante	Valor de temporización (formato S5TIME)

Descripción

La operación **Arrancar temporizador como impulso prolongado** arranca un temporizador con un valor determinado, siempre y cuando el RLO disponga de un flanco ascendente (cambio de "0" a "1"). El temporizador seguirá funcionando con el valor indicado, incluso si el RLO cambia a "0" con el temporizador en marcha. La consulta sobre si el estado de señal es "1" produce un "1", mientras el temporizador esté funcionando. El temporizador se arrancará de nuevo (se redisparará) con el valor de temporización indicado, si el RLO cambia de "0" a "1" con el temporizador en marcha.

El cuadro **Arrancar temporizador como impulso prolongado** sólo se puede disponer en el extremo derecho de la cadena de conexión. Sin embargo, puede utilizar varios cuadros **Arrancar temporizador como impulso prolongado**.

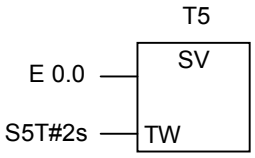
Consulte también Posición de un temporizador en la memoria y sus componentes.

Palabra de estado

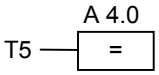
	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
Ecriture	-	-	-	-	-	0	-	-	0

Ejemplo

Segmento 1



Segmento 2



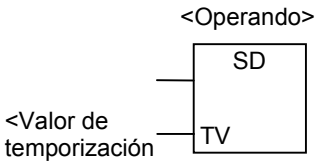
Si el estado de señal en E 0.0 cambia de "0" a "1" (flanco positivo en el RLO), se arrancará el temporizador T5. El temporizador seguirá funcionando, sin verse afectado por un flanco negativo en el RLO. Si el estado de señal en E 0.0 cambia de "0" a "1" antes de que haya transcurrido el valor de temporización indicado, el temporizador se redisparará.

Mientras funcione el temporizador, el estado de señal en la salida A 4.0 = 1.

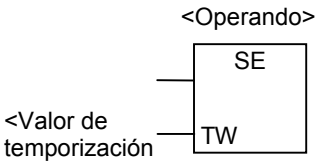
13.10 SE : Arrancar temporizador como retardo a la conexión

Símbolo

Inglés



Alemán



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Area de memoria	Descripción
Número del temporizador	Número del temporizador	TIMER	T	El operando indica el número del temporizador que se debe arrancar.
TV	TW	S5TIME	E, A, M, D, L o constante	Valor de temporización (formato S5TIME)

Descripción

La operación **Arrancar temporizador como retardo a la conexión** arranca un temporizador determinado, siempre y cuando el RLO disponga de un flanco ascendente (cambio de "0" a "1"). La consulta sobre si el estado de señal es "1" produce un "1", si el temporizador ha funcionado sin problemas y el RLO sigue indicando "1". Si el RLO cambia de "1" a "0" mientras esté funcionando el temporizador, éste se detendrá. En este caso la consulta sobre si el estado de señal es "1" producirá siempre un "0".

El cuadro **Arrancar temporizador como retardo a la conexión** sólo se puede disponer en el extremo derecho de la cadena de conexión. Sin embargo, puede utilizar varios cuadros **Arrancar temporizador como retardo a la conexión**.

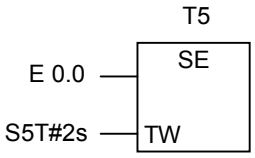
Consulte también Posición de un temporizador en la memoria y sus componentes.

Palabra de estado

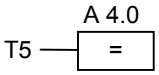
	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
Ecriture	-	-	-	-	-	0	-	-	0

Ejemplo

Segmento 1



Segmento 2

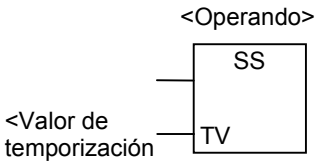


Si el estado de la señal en E 0.0 cambia de "0" a "1" (flanco positivo en el RLO), se arrancará el temporizador T5. Si el tiempo indicado ya ha transcurrido y el estado de la señal continúa siendo "1", la salida A 4.0 = 1. Si el estado de la señal cambia de "1" a "0", se detendrá el temporizador.

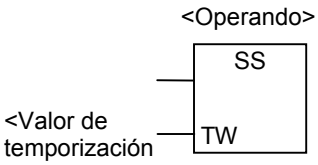
13.11 SS : Arrancar temporizador como retardo a la conexión con memoria

Símbolo

Inglés



Alemán



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Area de memoria	Descripción
Número del temporizador	Número del temporizador	TIMER	T	El operando indica el número del temporizador que se debe arrancar.
TV	TW	S5TIME	E, A, M, D, L o constante	Valor de temporización (formato S5TIME)

Descripción

La operación **Arrancar temporizador como retardo a la conexión con memoria** arranca un temporizador determinado, siempre y cuando el RLO disponga de un flanco ascendente (cambio de "0" a "1"). El temporizador seguirá funcionando con el valor de temporización indicado, aunque el RLO cambie a "0" antes de que haya transcurrido el tiempo indicado. La consulta sobre si el estado de señal es "1" produce un "1" independientemente del RLO. Si el RLO cambia de "0" a "1" mientras el temporizador esté funcionando, éste se arrancará de nuevo (se redisparará) con el valor indicado.

El cuadro **Arrancar temporizador como retardo a la conexión con memoria** sólo se puede disponer en el extremo derecho de la cadena de conexión. Sin embargo, puede utilizar varios cuadros **Arrancar temporizador como retardo a la conexión con memoria**.

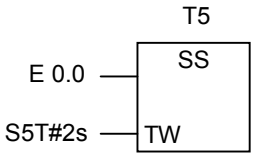
Consulte también Posición de un temporizador en la memoria y sus componentes.

Palabra de estado

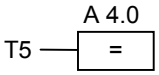
	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
Ecriture	-	-	-	-	-	0	-	-	0

Ejemplo

Segmento 1



Segmento 2



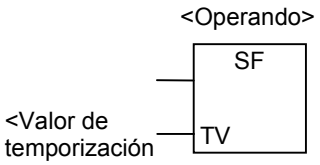
Si el estado de la señal en E 0.0 cambia de "0" a "1" (flanco positivo en el RLO),se arrancará el temporizador T5. El temporizador seguirá funcionando independientemente de si el estado de señal en E 0.0 cambia de "1" a "0". Si el estado de señal cambia de "0" a "1" antes de que haya transcurrido el valor de temporización, se redisparará el temporizador.

La salida A 4.0 = 1 cuando haya transcurrido el tiempo indicado.

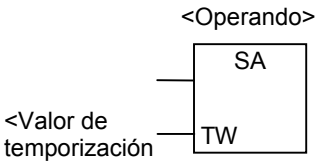
13.12 SA : Arrancar temporizador como retardo a la desconexión

Símbolo

Inglés



Alemán



Parámetro Inglés	Parámetro Alemán	Tipo de datos	Area de memoria	Descripción
Número del temporizador	Número del temporizador	TIMER	T	El operando indica el número del temporizador que se debe arrancar.
TV	TW	S5TIME	E, A, M, D, L o constante	Valor de temporización (formato S5TIME)

Descripción

La operación **Arrancar temporizador como retardo a la desconexión** arranca un temporizador determinado, siempre y cuando el RLO disponga de un flanco descendente (cambio de "1" a "0"). La consulta sobre si el estado de la señal es "1" produce un "1" si el RLO = 1 o si el temporizador funciona. El temporizador se pone a "0" si el RLO cambia de "0" a "1" mientras esté funcionando el temporizador. El temporizador se arrancará de nuevo si el RLO cambia de "1" a "0".

EL cuadro **Arrancar temporizador como retardo a la desconexión** sólo se puede disponer en el extremo derecho de la cadena de conexión. Sin embargo, puede utilizar varios cuadros **Arrancar temporizador como retardo a la desconexión**.

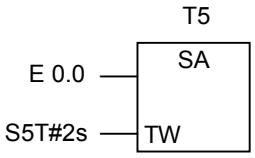
Consulte también Posición de un temporizador en la memoria y sus componentes.

Palabra de estado

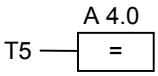
	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
Ecriture	-	-	-	-	-	0	-	-	0

Ejemplo

Segmento 1



Segmento 2



El temporizador se arranca cuando el estado de la señal en E 0.0 cambia de "1" a "0".

Si el estado de la señal cambia de "0" a "1", el temporizador se pondrá a "0".

El estado de la señal en la salida A 4.0 es "1", si el estado de la señal en la entrada E 0.0 = 1 o si el temporizador funciona.

14 Operaciones lógicas con palabras

14.1 Lista de operaciones lógicas con palabras

Descripción

Las operaciones lógicas con palabras comparan bit a bit pares de palabras (16 bits) y palabras dobles (32 bits) según la lógica de Boole. Estos valores se interpretan como puras configuraciones binarias. El resultado se determina consultando la salida OUT. ENO y EN tienen el mismo estado de señal.

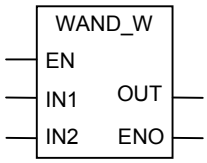
Si el resultado en la salida OUT es diferente de 0, el bit A1 de la palabra de estado se pone a "1". Si el resultado en la salida OUT es igual a 0, el bit A1 de la palabra de estado se pone a "0".

Se dispone de las operaciones lógicas con palabras siguientes:

- WAND_W : Y con palabras
- WOR_W : O con palabras
- WXOR_W : O-exclusiva con palabras
- WAND_DW : Y con palabras dobles
- WOR_DW : O con palabras dobles
- WXOR_DW : O-exclusiva con palabras dobles

14.2 WAND_W : Y con palabras

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	WORD	E, A, M, D, L o constante	Primer valor de la operación lógica
IN2	WORD	E, A, M, D, L o constante	Segundo valor de la operación lógica
OUT	WORD	E, A, M, D, L	Resultado lógico
ENO	BOOL	E, A, M, D, L	Salida de habilitación

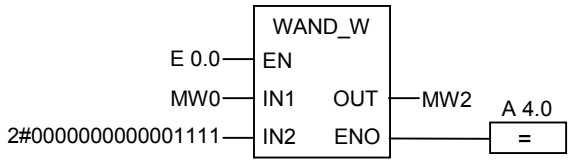
Descripción

Un 1 en la entrada de habilitación (EN) activa la operación **Y con palabras**. Esta operación combina bit a bit los dos valores digitales indicados en las entradas IN1 e IN2, según la tabla de verdad Y. Estos valores se interpretan como puras configuraciones binarias. El resultado se determina consultando la salida OUT. ENO y EN tienen el mismo estado de señal.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	1	X	0	0	-	X	1	1	1

Ejemplo



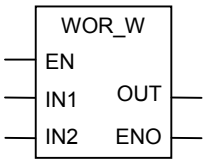
La operación se activa, si E 0.0 = 1. Los únicos bits relevantes son los bits 0 a 3, el resto de la palabra de marcas MW0 está enmascarado.

IN1 = 0101010101010101
IN2 = 0000000000001111
OUT = 0000000000000101

A 4.0 es "1", si la operación se ejecuta.

14.3 WOR_W : O con palabras

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	WORD	E, A, M, D, L o constante	Primer valor de la operación lógica
IN2	WORD	E, A, M, D, L o constante	Segundo valor de la operación lógica
OUT	WORD	E, A, M, D, L	Resultado lógico
ENO	BOOL	E, A, M, D, L	Salida de habilitación

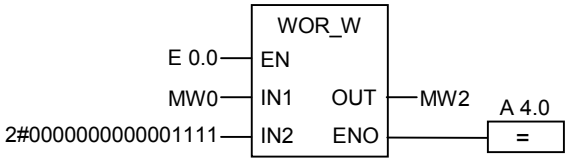
Descripción

Un 1 en la entrada de habilitación (EN) activa la operación **O con palabras dobles**. Esta operación combina bit a bit los dos valores digitales indicados en las entradas IN1 e IN2, según la tabla de verdad O.Estos valores se interpretan como puras configuraciones binarias. El resultado se puede consultar en la salida OUT. ENO y EN tienen el mismo estado de señal.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	1	X	0	0	-	X	1	1	1

Ejemplo



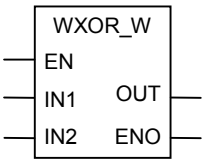
Si el estado de señal de la entrada E 0.0 = 1 se activa la operación. Se efectúa una O lógica con los bits de MW0 y de las constantes, de forma que los bits 0 a 3 se ponen a ”1"; MW2 toma los demás bits de MW0 tal como estaban.

IN1 = 0101010101010101
IN2 = 0000000000001111
OUT = 0101010101011111

A 4.0 es "1", si la operación se ejecuta.

14.4 WXOR_W : O-exclusiva con palabras

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	WORD	E, A, M, D, L o constante	Primer valor de la operación lógica
IN2	WORD	E, A, M, D, L o constante	Segundo valor de la operación lógica
OUT	WORD	E, A, M, D, L	Resultado lógico
ENO	BOOL	E, A, M, D, L	Salida de habilitación

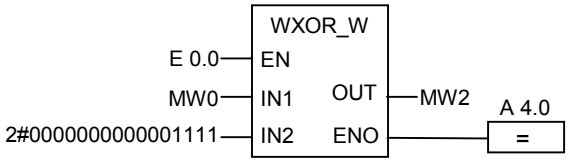
Descripción

Un 1 en la entrada de habilitación (EN) activa la operación **O-exclusiva con palabras**. Esta operación combina bit a bit los dos valores digitales indicados en las entradas IN1 e IN2, según la tabla de verdad O-exclusiva. Estos valores se interpretan como puras configuraciones binarias. El resultado se puede consultar en la salida OUT. ENO y EN tienen el mismo estado de señal.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	1	X	0	0	-	X	1	1	1

Ejemplo



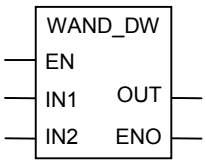
La operación se activa si la entrada E 0.0 = "1".

IN1 = 0101010101010101
IN2 = 0000000000001111
OUT = 0101010101011010

A 4.0 es ”1”, si la operación se ejecuta.

14.5 WAND_DW : Y con palabras dobles

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	DWORD	E, A, M, D, L o constante	Primer valor de la operación lógica
IN2	DWORD	E, A, M, D, L o constante	Segundo valor de la operación lógica
OUT	DWORD	E, A, M, D, L	Resultado lógico
ENO	BOOL	E, A, M, D, L	Salida de habilitación

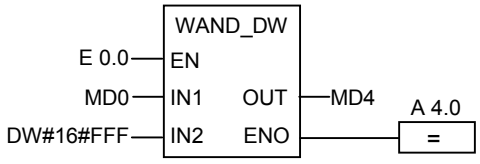
Descripción

Un 1 en la entrada de habilitación (EN) activa la operación **Y con palabras dobles**. Esta operación combina bit a bit los dos valores digitales indicados en las entradas IN1 e IN2, según la tabla de verdad Y. Estos valores se interpretan como puras configuraciones binarias. El resultado se determina consultando la salida OUT. ENO y EN tienen el mismo estado de señal.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	1	X	0	0	-	X	1	1	1

Ejemplo



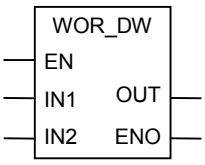
La operación se activa, si E 0.0 = 1. Los únicos bits relevantes son los bits 0 a 11, el resto de la palabra doble MD4 está enmascarado.

IN1	=	0101010101010101	0101010101010101
IN2	=	0000000000000000	0000111111111111
OUT	=	0000000000000000	0000010101010101

A 4.0 es "1", si la operación se ejecuta.

14.6 WOR_DW : O con palabras dobles

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	DWORD	E, A, M, D, L o constante	Primer valor de la operación lógica
IN2	DWORD	E, A, M, D, L o constante	Segundo valor de la operación lógica
OUT	DWORD	E, A, M, D, L	Resultado lógico
ENO	BOOL	E, A, M, D, L	Salida de habilitación

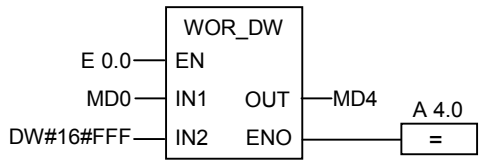
Descripción

Un 1 en la entrada de habilitación (EN) activa la operación **O con palabras dobles**. Esta operación combina bit a bit los dos valores digitales indicados en las entradas IN1 e IN2, según la tabla de verdad O.Estos valores se interpretan como puras configuraciones binarias. El resultado se puede consultar en la salida OUT. ENO y EN tienen el mismo estado de señal.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	1	X	0	0	-	X	1	1	1

Ejemplo



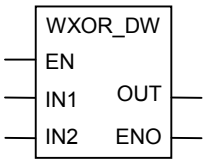
Si el estado de señal de la entrada E 0.0 = 1 se activa la operación. Se efectúa una O lógica con los bits de MW0 y de las constantes, de forma que los bits 0 a 11 se ponen a ”1"; MW2 toma los demás bits de MW0 tal como estaban.

IN1	=	0101010101010101	0101010101010101
IN2	=	0000000000000000	0000111111111111
OUT	=	0101010101010101	0101111111111111

A 4.0 es "1", si la operación se ejecuta.

14.7 WXOR_DW : O-exclusiva con palabras dobles

Símbolo



Parámetro	Tipo de datos	Area de memoria	Descripción
EN	BOOL	E, A, M, D, L, T, Z	Entrada de habilitación
IN1	DWORD	E, A, M, D, L o constante	Primer valor de la operación lógica
IN2	DWORD	E, A, M, D, L o constante	Segundo valor de la operación lógica
OUT	DWORD	E, A, M, D, L	Resultado lógico
ENO	BOOL	E, A, M, D, L	Salida de habilitación

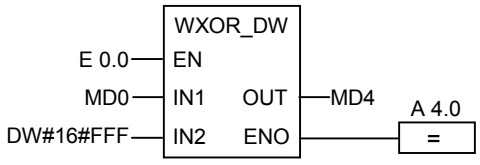
Descripción

Un 1 en la entrada de habilitación (EN) activa la operación **O-exclusiva con palabras dobles**. Esta operación combina bit a bit los dos valores digitales indicados en las entradas IN1 e IN2, según la tabla de verdad O-exclusiva. Estos valores se interpretan como puras configuraciones binarias. El resultado se puede consultar en la salida OUT. ENO y EN tienen el mismo estado de señal.

Palabra de estado

	RB	A1	A0	OV	OS	OR	STA	RLO	/ER
se escribe	1	X	0	0	-	X	1	1	1

Ejemplo



La operación se activa si la entrada E 0.0 = 1.

IN1 = 0101010101010101 0101010101010101
IN2 = 0000000000000000 0000111111111111
OUT = 0101010101010101 0101101010101010

A 4.0 es "1", si la operación se ejecuta.

A Sinopsis de las operaciones FUP

A.1 Operaciones FUP ordenadas según las abreviaturas nemotécnicas alemanas (SIMATIC)

Nemotécnica alemana	Nemotécnica inglesa	Catálogo de elementos de programa	Descripción
&	&	Operaciones lógicas con bits	Operación lógica Y
>=1	>=1	Operaciones lógicas con bits	Operación lógica O
=	=	Operaciones lógicas con bits	Asignación
#	#	Operaciones lógicas con bits	Conector
---|	---|	Operaciones lógicas con bits	Insertar una entrada binaria
---o|	---o|	Operaciones lógicas con bits	Invertir una entrada binaria
==0	==0	Bits de estado	Bits de resultado
>0	>0	Bits de estado	Bits de resultado
>=0	>=0	Bits de estado	Bits de resultado
<0	<0	Bits de estado	Bits de resultado
<=0	<=0	Bits de estado	Bits de resultado
<>0	<>0	Bits de estado	Bits de resultado
ABS	ABS	Función en coma flotante	Calcular el valor absoluto de un número en coma flotante
ACOS	ACOS	Función en coma flotante	Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante
ADD_DI	ADD_DI	Función en coma fija	Sumar enteros dobles
ADD_I	ADD_I	Función en coma fija	Sumar enteros
ADD_R	ADD_R	Función en coma flotante	Sumar números en coma flotante
ASIN	ASIN	Función en coma flotante	Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante
ATAN	ATAN	Función en coma flotante	Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante
BCD_DI	BCD_DI	Convertidor	Convertir de BCD a entero doble
BCD_I	BCD_I	Convertidor	Convertir de BCD a entero
BIE	BR	Bits de estado	Bit de anomalía "registro RB"
CALL	CALL	Control del programa	Llamar FC/SFC sin parámetros
CALL_FB	CALL_FB	Control del programa	CALL_FB Llamar FB
CALL_FC	CALL_FC	Control del programa	CALL_FC Llamar FC
CALL_SFB	CALL_SFB	Control del programa	CALL_SFB Llamar SFB
CALL_SFC	CALL_SFC	Control del programa	CALL_SFC Llamar SFC
CEIL	CEIL	Convertidor	Redondear número en coma flotante a entero superior
CMP >=D	CMP >=D	Comparador	Comparar enteros dobles
CMP >=I	CMP >=I	Comparador	Comparar enteros
CMP >=R	CMP >=R	Comparador	Comparar números en coma flotante

Nemotécnica alemana	Nemotécnica inglesa	Catálogo de elementos de programa	Descripción
COS	COS	Función en coma flotante	Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante
DI_BCD	DI_BCD	Convertidor	Convertir de entero doble a BCD
DI_R	DI_R	Convertidor	Convertir de entero doble a número en coma flotante
DIV_DI	DIV_DI	Función en coma fija	Dividir enteros dobles
DIV_I	DIV_I	Función en coma fija	Dividir enteros
DIV_R	DIV_R	Función en coma flotante	Dividir números en coma flotante
EXP	EXP	Función en coma flotante	Calcular el valor exponencial de un número en coma flotante
FLOOR	FLOOR	Convertidor	Redondear número en coma flotante a entero inferior
I_BCD	I_BCD	Convertidor	Convertir de entero a BCD
I_DI	I_DI	Convertidor	Convertir de entero a entero doble
INV_I	INV_I	Convertidor	Complemento a 1 de entero
INV_DI	INV_DI	Convertidor	Complemento a 1 de entero doble
JMP	JMP	Saltos	Salto incondicionado
JMP	JMP	Saltos	Salto condicionado a 1 en el bloque
JMPN	JMPN	Saltos	Salto condicionado a 0
LABEL	LABEL	Saltos	Meta del salto
LN	LN	Función en coma flotante	Calcular el logaritmo natural de un número en coma flotante
MCR>	MCR>	Control del programa	Conectar/Desconectar Master Control Relay
MCR<	MCR<	Control del programa	Conectar/Desconectar Master Control Relay
MCRA	MCRA	Control del programa	Inicio/Fin Master Control Relay
MCRD	MCRD	Control del programa	Inicio/Fin Master Control Relay
MOD_DI	MOD_DI	Función en coma fija	Obtener el resto de división de enteros dobles
MOVE	MOVE	Desplazar	Transferir un valor
MUL_DI	MUL_DI	Función en coma fija	Multiplicar enteros dobles
MUL_I	MUL_I	Función en coma fija	Multiplicar enteros
MUL_R	MUL_R	Función en coma flotante	Multiplicar números en coma flotante
N	N	Operaciones lógicas con bits	Detectar flanco negativo (1 -> 0)
NEG	NEG	Operaciones lógicas con bits	Detectar flanco de señal 1 -> 0
NEG_DI	NEG_DI	Convertidor	Complemento a 2 de entero doble
NEG_I	NEG_I	Convertidor	Complemento a 2 de entero
NEG_R	NEG_R	Convertidor	Cambiar el signo de un número en coma flotante
OPN	OPN	Llamada DB	Abrir bloque de datos
OS	OS	Bits de estado	Bit de anomalía "desbordamiento memorizado"
OV	OV	Bits de estado	Bit de anomalía "desbordamiento"
P	P	Operaciones lógicas con bits	Detectar flanco positivo (0 -> 1)
POS	POS	Operaciones lógicas con bits	Detectar flanco de señal 0 -> 1
R	R	Operaciones lógicas con bits	Desactivar salida
RET	RET	Control del programa	Retorno
ROL_DW	ROL_DW	Desplazar/rotar	Rotar palabra doble a la izquierda
ROUND	ROUND	Convertidor	Redondear a entero doble
ROR_DW	ROR_DW	Desplazar/rotar	Rotar palabra doble a la derecha
RS	RS	Operaciones lógicas con bits	Flipflop de desactivación/activación
S	S	Operaciones lógicas con bits	Activar salida

Nemotécnica alemana	Nemotécnica inglesa	Catálogo de elementos de programa	Descripción
SA	SF	Temporizadores	Arrancar temporizador como retardo a la desconexión
SAVE	SAVE	Operaciones lógicas con bits	Cargar resultado lógico (RLO) en el registro RB
S_AVERZ	S_OFFDT	Temporizadores	Parametrizar y arrancar temporizador como retardo a la desconexión
SE	SD	Temporizadores	Arrancar temporizador como retardo a la conexión
S_EVERZ	S_ODT	Temporizadores	Parametrizar y arrancar temporizador como retardo a la conexión
SHL_DW	SHL_DW	Desplazar/rotar	Desplazar palabra doble a la izquierda
SHL_W	SHL_W	Desplazar/rotar	Desplazar palabra a la izquierda
SHR_DI	SHR_DI	Desplazar/rotar	Desplazar entero doble a la derecha
SHR_DW	SHR_DW	Desplazar/rotar	Desplazar palabra doble a la derecha
SHR_I	SHR_I	Desplazar/rotar	Desplazar entero a la derecha
SHR_W	SHR_W	Desplazar/rotar	Desplazar palabra a la derecha
SI	SP	Temporizadores	Arrancar temporizador como impulso
S_IMPULS	S_PULSE	Temporizadores	Parametrizar y arrancar temporizador como impulso
SIN	SIN	Función en coma flotante	Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante
SQR	SQR	Función en coma flotante	Calcular el cuadrado de un número en coma flotante
SQRT	SQRT	Función en coma flotante	Calcular la raíz cuadrada de un número en coma flotante
SR	SR	Operaciones lógicas con bits	Flipflop de activación/desactivación
SS	SS	Temporizadores	Arrancar temporizador como retardo a la conexión con memoria
S_SEVERZ	S_ODTS	Temporizadores	Parametrizar y arrancar temporizador como retardo a la conexión con memoria
SUB_DI	SUB_DI	Función en coma fija	Restar enteros dobles
SUB_I	SUB_I	Función en coma fija	Restar enteros
SUB_R	SUB_R	Función en coma flotante	Restar números en coma flotante
SV	SE	Temporizadores	Arrancar temporizador como impulso prolongado
S_VIMP	S_PEXT	Temporizadores	Parametrizar y arrancar temporizador como impulso prolongado
SZ	SC	Temporizadores	Posicionar el contador en preselección
TAN	TAN	Función en coma flotante	Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante
TRUNC	TRUNC	Convertidor	Truncar a entero doble
UO	UO	Bits de estado	Bit de anomalía "operación no válida"
WAND_DW	WAND_DW	Operaciones lógicas con palabras	Y con palabras dobles
WAND_W	WAND_W	Operaciones lógicas con palabras	Y con palabras
WOR_DW	WOR_DW	Operaciones lógicas con palabras	O con palabras dobles
WOR_W	WOR_W	Operaciones lógicas con palabras	O con palabras
WXOR_DW	WXOR_DW	Operaciones lógicas con palabras	O-exclusiva con palabras dobles
WXOR_W	WXOR_W	Operaciones lógicas con palabras	O-exclusiva con palabras

Nemotécnica alemana	Nemotécnica inglesa	Catálogo de elementos de programa	Descripción
XOR	XOR	Operaciones lógicas con bits	Operación lógica O-exclusiva
ZAEHLER	S_CUD	Contadores	Parametrizar e incrementar / decrementar contador
ZR	CD	Contadores	Decrementar contador
Z_RUECK	S_CD	Contadores	Parametrizar y decrementar contador
ZV	CU	Contadores	Incrementar contador
Z_VORW	S_CU	Contadores	Parametrizar e incrementar contador

A.2 Operaciones FUP ordenadas según las abreviaturas nemotécnicas inglesas (internacional)

Nemotécnica inglesa	Nemotécnica alemana	Catálogo de elementos de programa	Descripción
&	&	Operaciones lógicas con bits	Operación lógica Y
>=1	>=1	Operaciones lógicas con bits	Operación lógica O
=	=	Operaciones lógicas con bits	Asignación
#	#	Operaciones lógicas con bits	Conector
---|	---|	Operaciones lógicas con bits	Insertar una entrada binaria
---o|	---o|	Operaciones lógicas con bits	Invertir una entrada binaria
==0	==0	Bits de estado	Bits de resultado
>0	>0	Bits de estado	Bits de resultado
>=0	>=0	Bits de estado	Bits de resultado
<0	<0	Bits de estado	Bits de resultado
<=0	<=0	Bits de estado	Bits de resultado
<>0	<>0	Bits de estado	Bits de resultado
ABS	ABS	Función en coma flotante	Calcular el valor absoluto de un número en coma flotante
ACOS	ACOS	Función en coma flotante	Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante
ADD_DI	ADD_DI	Función en coma fija	Sumar enteros dobles
ADD_I	ADD_I	Función en coma fija	Sumar enteros
ADD_R	ADD_R	Función en coma flotante	Sumar números en coma flotante
ASIN	ASIN	Función en coma flotante	Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante
ATAN	ATAN	Función en coma flotante	Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante
BCD_DI	BCD_DI	Convertidor	Convertir de BCD a entero doble
BCD_I	BCD_I	Convertidor	Convertir de BCD a entero
BR	BIE	Bits de estado	Bit de anomalía "registro RB"
CALL	CALL	Control del programa	Llamar FC/SFC sin parámetros
CALL_FB	CALL_FB	Control del programa	CALL_FB Llamar FB
CALL_FC	CALL_FC	Control del programa	CALL_FC Llamar FC
CALL_SFB	CALL_SFB	Control del programa	CALL_SFB Llamar SFB
CALL_SFC	CALL_SFC	Control del programa	CALL_SFC Llamar SFC
CD	ZR	Contadores	Decrementar contador
CEIL	CEIL	Convertidor	Redondear número en coma flotante a entero superior
CMP >=D	CMP >=D	Comparador	Comparar enteros dobles
CMP >=I	CMP >=I	Comparador	Comparar enteros
CMP >=R	CMP >=R	Comparador	Comparar números en coma flotante
COS	COS	Función en coma flotante	Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante

Nemotécnica inglesa	Nemotécnica alemana	Catálogo de elementos de programa	Descripción
CU	ZV	Contadores	Incrementar contador
DI_BCD	DI_BCD	Convertidor	Convertir de entero doble a BCD
DI_R	DI_R	Convertidor	Convertir de entero doble a número en coma flotante
DIV_DI	DIV_DI	Función en coma fija	Dividir enteros dobles
DIV_I	DIV_I	Función en coma fija	Dividir enteros
DIV_R	DIV_R	Función en coma flotante	Dividir números en coma flotante
EXP	EXP	Función en coma flotante	Calcular el valor exponencial de un número en coma flotante
FLOOR	FLOOR	Convertidor	Redondear número en coma flotante a entero inferior
I_BCD	I_BCD	Convertidor	Convertir de entero a BCD
I_DI	I_DI	Convertidor	Convertir de entero a entero doble
INV_I	INV_I	Convertidor	Complemento a 1 de entero
INV_DI	INV_DI	Convertidor	Complemento a 1 de entero doble
JMP	JMP	Saltos	Salto incondicionado
JMP	JMP	Saltos	Salto condicionado a 1 en el bloque
JMPN	JMPN	Saltos	Salto condicionado a 0
LABEL	LABEL	Saltos	Meta del salto
LN	LN	Función en coma flotante	Calcular el logaritmo natural de un número en coma flotante
MCR>	MCR>	Control del programa	Conectar/Desconectar Master Control Relay
MCR<	MCR<	Control del programa	Conectar/Desconectar Master Control Relay
MCRA	MCRA	Control del programa	Inicio/Fin Master Control Relay
MCRD	MCRD	Control del programa	Inicio/Fin Master Control Relay
MOD_DI	MOD_DI	Función en coma fija	Obtener el resto de división de enteros dobles
MOVE	MOVE	Desplazar	Transferir un valor
MUL_DI	MUL_DI	Función en coma fija	Multiplicar enteros dobles
MUL_I	MUL_I	Función en coma fija	Multiplicar enteros
MUL_R	MUL_R	Función en coma flotante	Multiplicar números en coma flotante
N	N	Operaciones lógicas con bits	Detectar flanco negativo (1 -> 0)
NEG	NEG	Operaciones lógicas con bits	Detectar flanco de señal 1 -> 0
NEG_DI	NEG_DI	Convertidor	Complemento a 2 de entero doble
NEG_I	NEG_I	Convertidor	Complemento a 2 de entero
NEG_R	NEG_R	Convertidor	Cambiar el signo de un número en coma flotante
OPN	OPN	Llamada DB	Abrir bloque de datos
OS	OS	Bits de estado	Bit de anomalía "desbordamiento memorizado"
OV	OV	Bits de estado	Bit de anomalía "desbordamiento"
P	P	Operaciones lógicas con bits	Detectar flanco positivo (0 -> 1)
POS	POS	Operaciones lógicas con bits	Detectar flanco de señal 0 -> 1
R	R	Operaciones lógicas con bits	Desactivar salida
RET	RET	Control del programa	Retorno
ROL_DW	ROL_DW	Desplazar/rotar	otar palabra doble a la izquierda

Nemotécnica inglesa	Nemotécnica alemana	Catálogo de elementos de programa	Descripción
ROUND	ROUND	Convertidor	Redondear a entero doble
ROR_DW	ROR_DW	Desplazar/rotar	Rotar palabra doble a la derecha
RS	RS	Operaciones lógicas con bits	Flipflop de desactivación/activación
S	S	Operaciones lógicas con bits	Activar salida
SAVE	SAVE	Operaciones lógicas con bits	Cargar resultado lógico (RLO) en el registro RB
SC	SZ	Contadores	Posicionar el contador en preselección
S_CD	Z_RUECK	Contadores	Parametrizar y decrementar contador
S_CU	Z_VORW	Contadores	Parametrizar e incrementar contador
S_CUD	ZAEHLER	Contadores	Parametrizar e incrementar / decrementar contador
SD	SE	Temporizadores	Arrancar temporizador como retardo a la conexión
SE	SV	Temporizadores	Arrancar temporizador como impulso prolongado
SF	SA	Temporizadores	Arrancar temporizador como retardo a la desconexión
SHL_DW	SHL_DW	Desplazar/rotar	Desplazar palabra doble a la izquierda
SHL_W	SHL_W	Desplazar/rotar	Desplazar palabra a la izquierda
SHR_DI	SHR_DI	Desplazar/rotar	Desplazar entero doble a la derecha
SHR_DW	SHR_DW	Desplazar/rotar	Desplazar palabra doble a la derecha
SHR_I	SHR_I	Desplazar/rotar	Desplazar entero a la derecha
SHR_W	SHR_W	Desplazar/rotar	Desplazar palabra a la derecha
SIN	SIN	Función en coma flotante	Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante
S_ODT	S_EVERZ	Temporizadores	Parametrizar y arrancar temporizador como retardo a la conexión
S_ODTS	S_SEVERZ	Temporizadores	Parametrizar y arrancar temporizador como retardo a la conexión con memoria
S_OFFDT	S_AVERZ	Temporizadores	Parametrizar y arrancar temporizador como retardo a la desconexión
SP	SI	Temporizadores	Arrancar temporizador como impulso
S_PEXT	S_VIMP	Temporizadores	Parametrizar y arrancar temporizador como impulso prolongado
S_PULSE	S_IMPULS	Temporizadores	Parametrizar y arrancar temporizador como impulso
SQR	SQR	Función en coma flotante	Calcular el cuadrado de un número en coma flotante
SQRT	SQRT	Función en coma flotante	Calcular la raíz cuadrada de un número en coma flotante
SR	SR	Operaciones lógicas con bits	Flipflop de activación/desactivación
SS	SS	Temporizadores	Arrancar temporizador como retardo a la conexión con memoria
SUB_DI	SUB_DI	Función en coma fija	Restar enteros dobles
SUB_I	SUB_I	Función en coma fija	Restar enteros
SUB_R	SUB_R	Función en coma flotante	Restar números en coma flotante
TAN	TAN	Función en coma flotante	Calcular funciones trigonométricas de ángulos representados mediante números en coma flotante
TRUNC	TRUNC	Convertidor	Truncar a entero doble

Nemotécnica inglesa	Nemotécnica alemana	Catálogo de elementos de programa	Descripción
UO	UO	Bits de estado	Bit de anomalía "operación no válida"
WAND_DW	WAND_DW	Operaciones lógicas con palabras	Y con palabras dobles
WAND_W	WAND_W	Operaciones lógicas con palabras	Y con palabras
WOR_DW	WOR_DW	Operaciones lógicas con palabras	O con palabras dobles
WOR_W	WOR_W	Operaciones lógicas con palabras	O con palabras
WXOR_DW	WXOR_DW	Operaciones lógicas con palabras	O-exclusiva con palabras dobles
WXOR_W	WXOR_W	Operaciones lógicas con palabras	O-exclusiva con palabras
XOR	XOR	Operaciones lógicas con bits	Operación lógica O-exclusiva

B Ejemplos de programación

B.1 Lista de ejemplos de programación

Aplicaciones prácticas

Todas las instrucciones FUP activan una operación determinada. Combinando estas operaciones en un programa se puede llevar a cabo una gran variedad de tareas de automatización. Este capítulo contiene los siguientes ejemplos:

- Controlar una cinta transportadora usando operaciones lógicas con bits
- Detectar el sentido de marcha de una cinta transportadora usando operaciones lógicas con bits
- Generar un impulso de reloj usando operaciones de temporización
- Supervisión del depósito usando operaciones de contaje y de comparación
- Resolver un problema usando operaciones aritméticas con enteros
- Ajustar el tiempo de calentamiento de una caldera

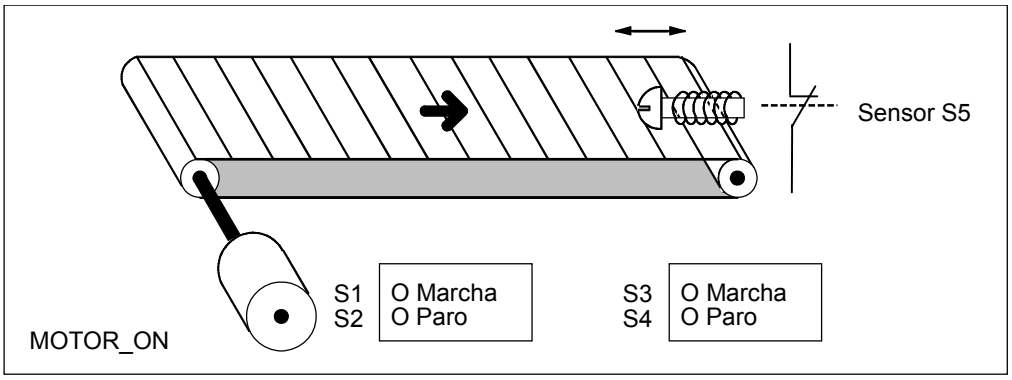
Operaciones utilizadas

Nemotécnica alemana	Operación	Descripción
WAND_W	Lógica de palabras	Y con palabras
WOR_W	Lógica de palabras	O con palabras
Z_RUECK	Contadores	Decrementar contador
Z_VORW	Contadores	Incrementar contador
R	Operaciones lógicas con bits	Desactivar salida
S	Operaciones lógicas con bits	Activar salida
P	Operaciones lógicas con bits	Detectar flanco positivo 0 $\rightarrow$ 1
ADD_I	Función en coma fija	Sumar enteros
DIV_I	Función en coma fija	Dividir enteros
MUL_I	Función en coma fija	Multiplicar enteros
CMP >=I	Comparadores	Comparar enteros
CMP <=I	Comparadores	Comparar enteros
&	Operaciones lógicas con bits	Operación Y lógica
>=1	Operaciones lógicas con bits	Operación O lógica
=	Operaciones lógicas con bits	Asignación
JMPN	Saltos	Saltar si es 0 (condicional)
RET	Control del programa	Retorno
MOVE	Desplazamiento	Asignar un valor
SV	Temporizadores	Temporizador de impulso prolongado

B.2 Ejemplos: Operaciones lógicas con bits

Ejemplo 1: Controlar una cinta transportadora

La figura muestra una cinta transportadora que se pone en marcha eléctricamente. Al principio de la cinta (es decir, en el extremo izquierdo) se encuentran dos pulsadores: S1 para MARCHA (start) y S2 para PARO (stop). Al final de la cinta, es decir, en el extremo derecho se encuentran otros dos pulsadores: S3 para MARCHA y S4 para PARO. La cinta puede ponerse en marcha o pararse desde cualesquiera de ambos extremos. Asimismo, el sensor S5 detiene la cinta cuando un paquete alcanza el final de la cinta.



Programación absoluta y simbólica

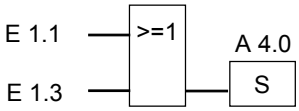
Se puede escribir un programa que controle la cinta transportadora usando **valores absolutos** o **símbolos** para representar los distintos componentes del sistema de transporte.

Los símbolos los define el usuario en la tabla de símbolos (v. la Ayuda en pantalla de STEP 7).

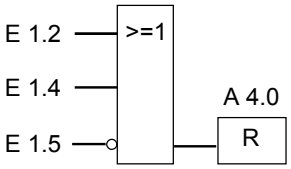
Componente del sistema	Dirección absoluta	Símbolo	Tabla de símbolos
Pulsador de marcha	E 1.1	S1	E 1.1 S1
Pulsador de paro	E 1.2	S2	E 1.2 S2
Pulsador de marcha	E 1.3	S3	E 1.3 S3
Pulsador de paro	E 1.4	S4	E 1.4 S4
Sensor	E 1.5	S5	E 1.5 S5
Motor	A 4.0	MOTOR_ON	A 4.0 MOTOR_ON

Diagrama de funciones para controlar una cinta transportadora

Segmento 1: Pulsando cualquiera de los pulsadores de marcha se pone el motor en marcha.

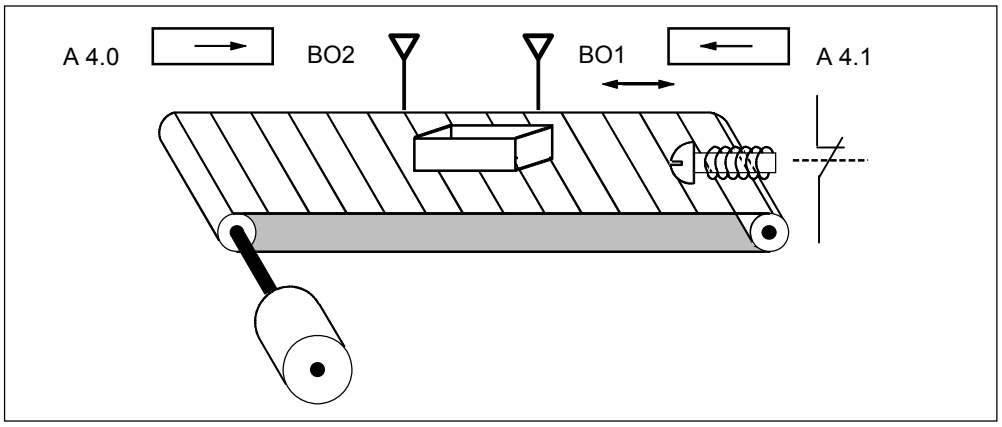


Segmento 2: Pulsando cualquiera de los pulsadores de paro o abriendo el contacto normalmente cerrado al final de la cinta se desconecta el motor.



Ejemplo 2: Detectar el sentido de marcha de una cinta transportadora

La figura muestra una cinta transportadora equipada con dos barreras ópticas (BO1 y BO2) concebidas para detectar el sentido de marcha de la cinta transportadora. Cada barrera óptica funciona igual que un contacto normalmente abierto.



Programación absoluta y simbólica

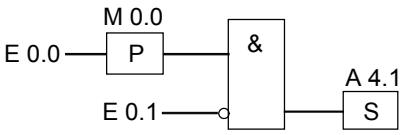
Se puede escribir un programa que controle la cinta transportadora usando **valores absolutos** o **símbolos** para representar los distintos componentes del sistema de transporte.

Los símbolos los define el usuario en la tabla de símbolos (v. la Ayuda en pantalla de STEP 7).

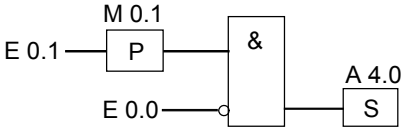
Componente del sistema	Dirección absoluta	Símbolo	Tabla de símbolos
Barrera óptica 1	E 0.0	BO1	E 0.0 BO1
Barrera óptica 2	E 0.1	BO2	E 0.1 BO2
Indicador de movimiento a la derecha	A 4.0	DER	A 4.0 DER
Indicador de movimiento a la izquierda	A 4.1	IZQ	A 4.1 IZQ
Marca de impulso 1	M 0.0	MI1	M 0.0 MI1
Marca de impulso 2	M 0.1	MI2	M 0.1 MI2

Diagrama de funciones para detectar el sentido de marcha de una cinta transportadora

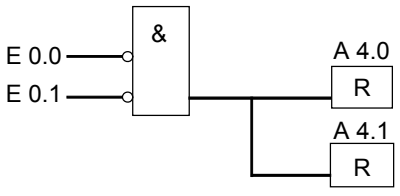
Segmento 1: Si el estado de señal de la entrada E 0.0 cambia de 0 a 1 (flanco positivo) y al mismo tiempo el estado de señal de la entrada E 0.1 es 0, entonces el paquete se está moviendo a la izquierda.



Segmento 2: Si el estado de señal de la entrada E 0.1 cambia de 0 a 1 (flanco positivo) y al mismo tiempo el estado de señal de la entrada E 0.0 es 0, entonces el paquete se esta moviendo a la derecha. Si se interrumpe una de las barreras ópticas, ésto significa que hay un paquete entre las barreras.



Segmento 3: Si una de las barreras ópticas es interrumpida, ésto significa que un paquete se encuentra entre las barreras. El indicador de sentido de marcha se desactiva.



B.3 Ejemplo: Operaciones de temporización

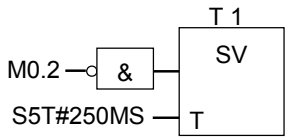
Reloj

Para generar una señal que se repita periódicamente se puede utilizar un reloj o un relé intermitente. Los relojes se suelen utilizar en sistemas de señalización que controlan la intermitencia de lámparas indicadoras.

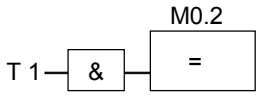
En el S7-300 se puede implementar la función Reloj usando un procesamiento tem-porizado en bloques de organización especiales. El ejemplo siguiente de un programa FUP muestra el uso de funciones temporizadas para generar un reloj.

Diagrama de funciones para generar un impulso de reloj (relación impulso-pausa 1:1)

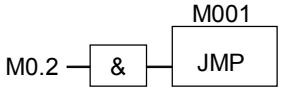
Segmento 1: Si el estado de señal del temporizador T1 es 0, se carga el valor de temporización 250 ms en T1 y T1 arranca como temporizador de impulso prolongado.



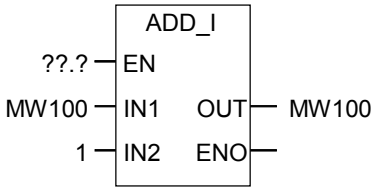
Segmento 2: El estado de señal del temporizador se almacena temporalmente en una marca auxiliar.



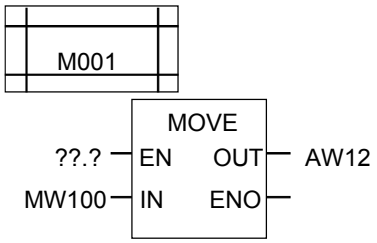
Segmento 3: Si el estado de señal del temporizador T1 es 1, salta a la meta M001.



Segmento 4: Cada vez que transcurre el tiempo programado en el temporizador T1 se incrementa en 1 la palabra de marcas 100.

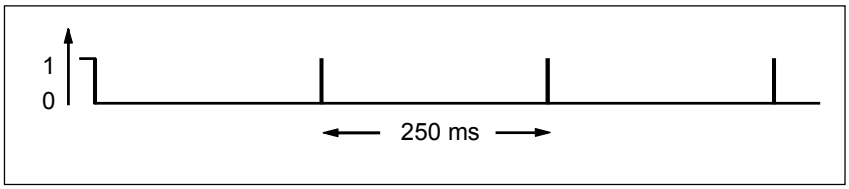


Segmento 5: La operación **MOVE** permite ver las distintas frecuencias de reloj en las salidas A 12.0 a A 13.7.



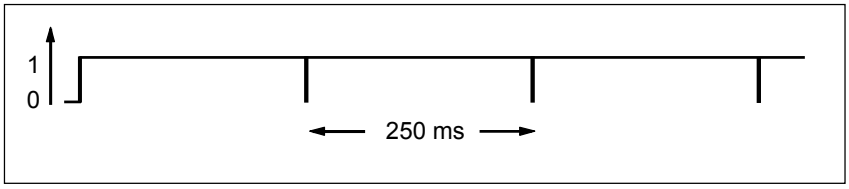
Consulta del estado de señal

La consulta de la señal del temporizador T1 arroja el siguiente resultado lógico para el parámetro de entrada invertido de operación Y lógica (M0.2):



En cuanto finaliza el tiempo programado, el temporizador se vuelve a poner en marcha. Por este motivo, la consulta produce sólo brevemente un estado de señal de 1.

La figura muestra el aspecto de un bit RLO negado (invertido):



El bit RLO es 0 cada 250 ms. El salto se ignora y el contenido de la palabra de marcas MW100 se incrementa en 1.

Programar una frecuencia determinada

Con los bits de los bytes de marcas MB101 y MB100 se consiguen las frecuencias siguientes:

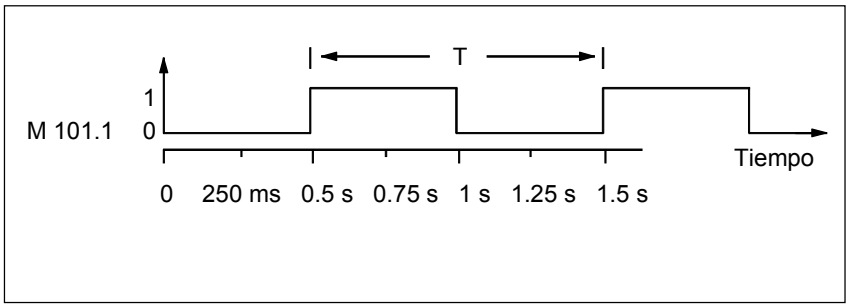
MB101, MB100	Frecuencia en hertzios	Duración
M 101.0	2.0	0.5 s (250 ms on / 250 ms off)
M 101.1	1.0	1 s (0.5 s on / 0.5 s off)
M 101.2	0.5	2 s (1 s on / 1 s off)
M 101.3	0.25	4 s (2 s on / 2 s off)
M 101.4	0.125	8 s (4 s on / 4 s off)
M 101.5	0.0625	16 s (8 s on / 8 s off)
M 101.6	0.03125	32 s (16 s on / 16 s off)
M 101.7	0.015625	64 s (32 s on / 32 s off)
M 100.0	0.0078125	128 s (64 s on / 64 s off)
M 100.1	0.0039062	256 s (128 s on / 128 s off)
M 100.2	0.0019531	512 s (256 s on / 256 s off)
M 100.3	0.0009765	1024 s (512 s on / 512 s off)
M 100.4	0.0004882	2048 s (1024 s on / 1024 s off)
M 100.5	0.0002441	4096 s (2048 s on / 2048 s off)
M 100.6	0.000122	8192 s (4096 s on / 4096 s off)
M 100.7	0.000061	16384 s (8192 s on / 8192 s off)

Estados de señal de los bits del byte de marcas MB101

Ciclo	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en ms
0	0	0	0	0	0	0	**0**	0	250
1	0	0	0	0	0	0	**0**	1	250
2	0	0	0	0	0	0	1	0	250
3	0	0	0	0	0	0	1	1	250
4	0	0	0	0	0	1	**0**	0	250
5	0	0	0	0	0	1	**0**	1	250
6	0	0	0	0	0	1	1	0	250
7	0	0	0	0	0	1	1	1	250
8	0	0	0	0	1	0	**0**	0	250
9	0	0	0	0	1	0	**0**	1	250
10	0	0	0	0	1	0	1	0	250
11	0	0	0	0	1	0	1	1	250
12	0	0	0	0	1	1	**0**	0	250

Estado de señal del bit 1 de MB101 (M 101.1)

Frecuencia = 1/T = 1/1 s = 1 Hz



B.4 Ejemplo: Operaciones de contaje y comparación

Area de almacenamiento con contador y comparador

La figura muestra un sistema con dos cintas transportadoras y un área de almacenamiento temporal colocada entre ambas. La cinta transportadora 1 transporta paquetes al área de almacenamiento. Una barrera óptica situada al final de la cinta 1 junto al área de almacenamiento determina cuántos paquetes se transportan a dicha área. La cinta transportadora 2 transporta paquetes desde el área de almacenamiento a una plataforma de carga donde llegan camiones y los recogen para suministrarlos a los clientes. Una barrera óptica situada al final de la cinta transportadora 2 junto al área de almacenamiento determina cuántos paquetes abandonan el área de almacenamiento para ser transportados a la plataforma de carga. Un panel indicador con cinco lámparas señala el nivel del área de almacenamiento temporal.

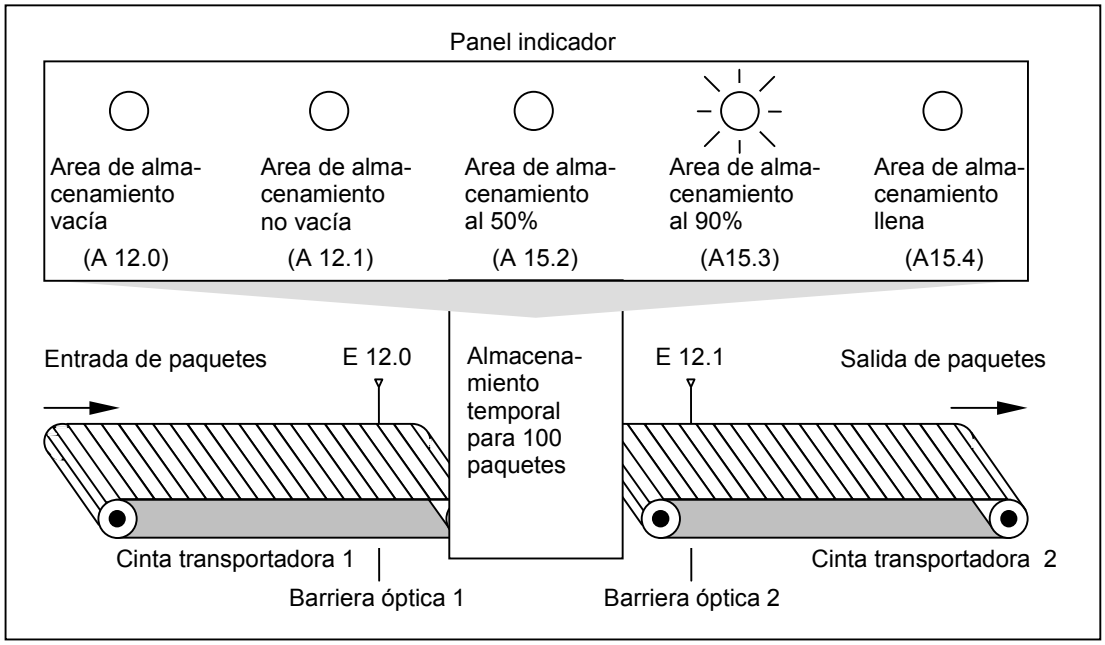
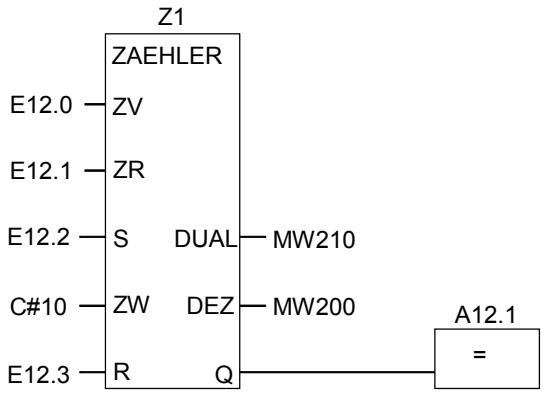
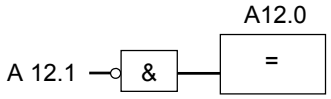


Diagrama de funciones para activar las lámparas del panel indicador

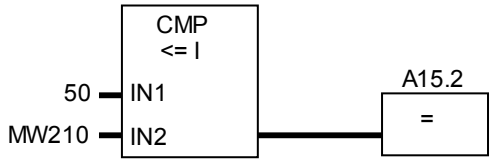
Segmento 1: El contador Z1 incrementa con un flanco de señal de "0" a "1" en la entrada ZV y decrementa con un flanco de señal de "0" a "1"en la entrada ZR. Con un flanco de señal de "0" a "1" en la entrada S el valor del contador se pone en el valor de ZW. Con un flanco de señal de "0" a "1" en la entrada R el valor del contador se pone a "0". En el MW200 está depositado el valor actual del contador de Z1. A12.1 marca "Area de almacenamiento no vacía".



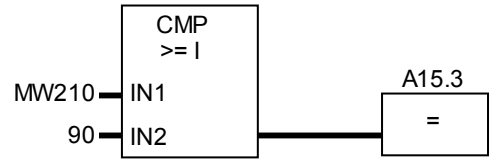
Segmento 2: A12.0 señaliza "área de almacenamiento vacía".



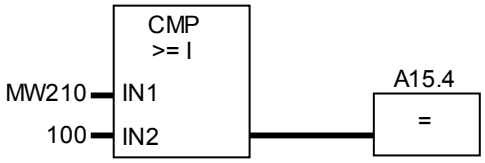
Segmento 3:Si 50 es menor o igual al valor del contador (o si el estado actual del contador es mayor igual que 50), se enciende la lámpara "Area de almacenamiento al 50%.



Segmento 4: Si el valor del contador es mayor o igual a 90 se enciende la lámpara "Area de almacenaciemto al 90%.



Segmento 5: Si el valor del contador es mayor o igual a 100 se enciende la lámpara "Area de almacenamiento llena".



B.5 Ejemplo: Operaciones de aritmétca con enteros

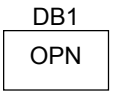
Resolver un Problema aritmético

El programa de ejemplo siguiente muestra cómo obtener con tres operaciones aritméticas para enteros el mismo resultado que la ecuación:

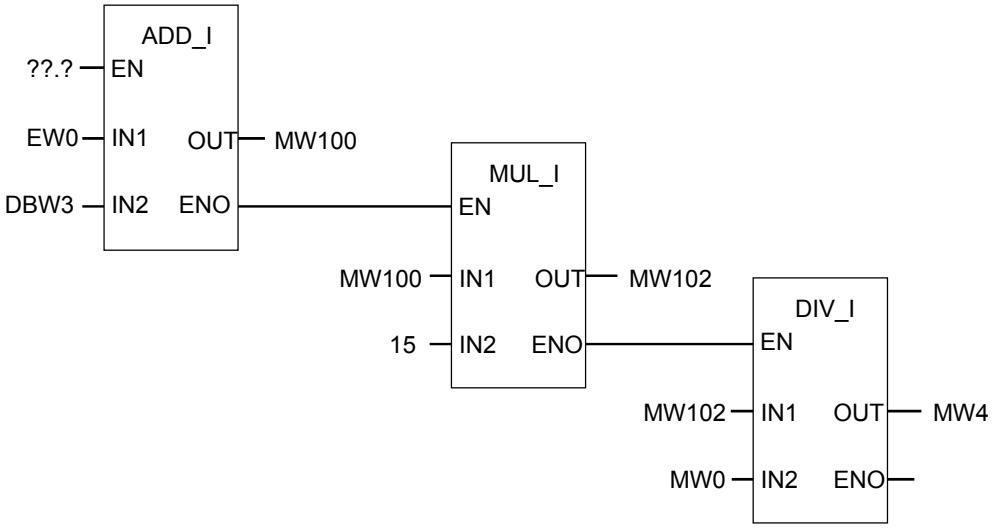
MW4 = ((EW0 + DBW3) x 15) / MW0

Diagrama de funciones

Segmento 1: Abrir bloque de datos DB1.



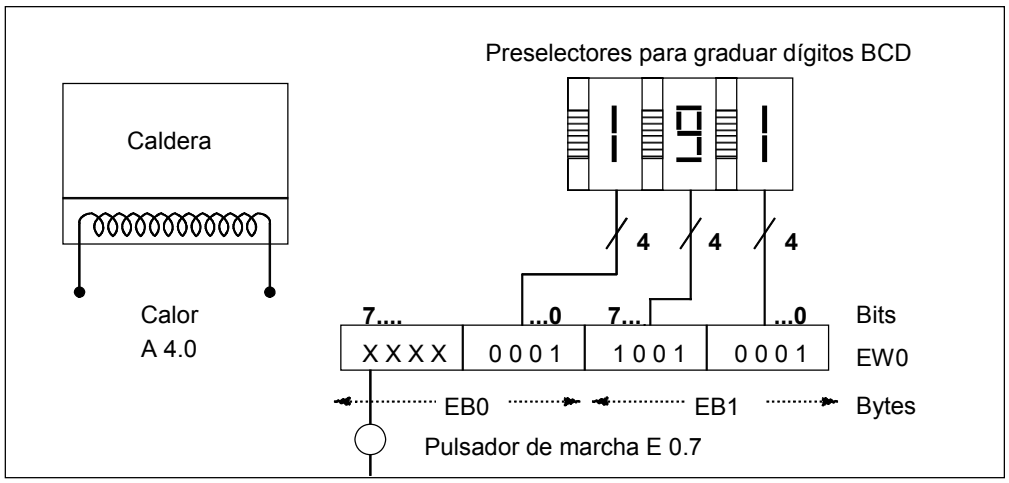
Segmento 2: La palabra de entrada EW0 se suma a la palabra de datos globales DBW3 (el bloque de datos tiene que estar definido y abierto) y la suma se carga en la palabra de marcas MW100. Después se multiplica MW100 por 15 y el resultado se deposita en la palabra doble de marcas MW102. Luego se divide MW102 entre MW0. Es resultado se guarda en MW4.



B.6 Ejemplo: Operaciones lógicas con palabras

Calentar una caldera

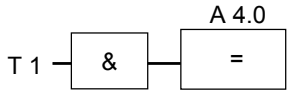
El operador de la caldera conecta la caldera accionando el pulsador de marcha. El operador puede graduar un tiempo de calentamiento utilizando los preselectores mecánicos. El valor fijado por el operador indica los segundos en formato decimal codificado en binario (BCD).



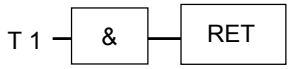
Componente del sistema	Dirección absoluta
Pulsador de marcha	E 0.7
Preselector digital para unidades	E 1.0 a E 1.3
Preselector digital para decenas	E 1.4 a E 1.7
Preselector digital para centenas	E 0.0 a E 0.3
Comienzo del proceso de calentamiento	A 4.0

Diagrama de funciones

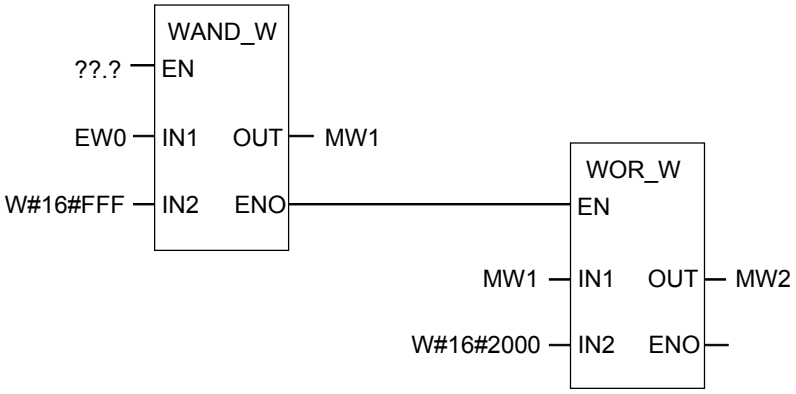
Segmento 1: Si el temporizador está en marcha, comienza el proceso de calentamiento.



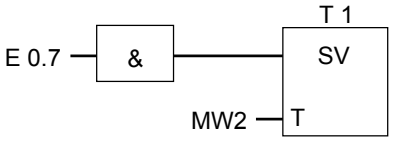
Segmento 2: Si el temporizador está en marcha, la operación Retorno finaliza aquí.



Segmento 3: Enmascarar los bits de entrada E 0.4 a E 0.7 (es decir, ponerlos a 0). Estos bits de las entradas de los preselectores no se utilizan. Los 16 bits de las entradas de los preselectores se combinan con W#16#0FFF mediante la operación **Y con palabras**. El resultado se carga en la palabra de marcas MW1. Para regular la base de tiempo en segundos se combina el valor de preselección con W#16#2000 mediante la operación **O con palabras**, poniendo el bit 13 a 1 y el bit 12 a 0.



Segmento 4: Arrancar el temporizador T1 como temporizador de impulso prolongado, cuando se oprima el pulsador de marcha, cargando la palabra de marcas MW2 como valor de preselección (derivado de la lógica arriba descrita).



C Uso de FUP

C.1 Mecanismo EN/ENO

La habilitación (EN) y la salida de habilitación (ENO) de cuadros (boxes) FUP/KOP se realiza mediante el bit RB.

Si EN y ENO están conectados, rige:

ENO = EN AND NOT (error del cuadro)

Si no se produce ningún error (error del cuadro = 0), entonces ENO = EN.

El mecanismo EN/ENO se utiliza para:

- operaciones aritméticas
- operaciones de transferencia y de conversión
- operaciones de desplazamiento y de rotación
- llamadas de bloques

Este mecanismo **no se utiliza** para:

- comparaciones
- contadores
- temporizadores

Al rededor de las verdaderas instrucciones del cuadro o box se generan instrucciones adicionales para el mecanismo EN/ENO dependiendo de las operaciones lógicas precedentes o siguientes. Los cuatro casos posibles se indican a continuación con un sumando a modo de ejemplo:

- Sumando con conexión EN y con conexión ENO
- Sumando con conexión EN y sin conexión ENO
- Sumando sin conexión EN y con conexión ENO
- Sumando sin conexión EN y sin conexión ENO

Advertencia para la creación de bloques propios

Si desea escribir bloques y llamarlos después en FUP/KOP deberá vigilar que el bit RB esté activado al salir del bloque. El cuarto ejemplo demuestra que ésto no ocurre automáticamente. El RB no puede ser utilizado igual que una marca, ya que es sobreescrito continuamente por el mecanismo EN/ENO. Utilice en su lugar una variable temporal para guardar los errores que se hayan producido. Inicialice la variable con el valor 0. Active esta variable sirviéndose del mecanismo EN/ENO en cualquier punto del bloque donde crea que una operación fracasada pueda representar un error para todo el bloque. Para ello basta un NOT y una bobina de activación. Al final del bloque programa un segmento:

```
ende:  UN fehler

       SAVE
```

Tome las medidas necesarias para que este segmento se ejecute en cualquier caso, es decir, no utilice nunca una operación BEB dentro de este bloque e impida que se salte este segmento.

C.1.1 Sumando con conexión EN y ENO

Si el sumando tiene una conexión EN y una ENO se añaden las siguientes instrucciones AWL:

```
1           U   E   0.0   // Conexión EN
2           SPBNB _001    // Desplazar RLO al RB y saltar si RLO == 0
3           L     in1     // Parámetro del cuadro
4           L     in2     // Parámetro del cuadro
5           +I            // Suma
6           T     out     // Parámetro del cuadro
7           UN    OV      // Detección de errores
8           SAVE          // Guardar error en RB
9           CLR           // Primera consulta
10    _001: U     RB      // Desplaza RB al RLO
11          =     A   4.0
```

Después de la línea1 el RLO contiene el resultado de la combinación lógica precedente. La instrucción SPBNB copia el RLO en el RB y activa el bit de primera consulta.

- Si el RLO es 0, se salta a la línea 10 y se continúa con U RB. La suma no se lleva a cabo. En la línea 10 se copia el RB en el RLO y por lo tanto se le asigna un 0 a la salida.
- Si el RLO es 1, no se salta, es decir, se realiza la suma. En la línea 7 se determina si se ha producido un error al sumar, lo cual se almacena en el bit RB en la línea 8. La línea 9 activa el bit de primera consulta. En la línea 10 se vuelve a copiar el bit RB en el RLO, con lo cual en la salida se indica si la suma es correcta.
  El bit RB no se modifica en las líneas 10 y 11, por lo que también indica si la suma es correcta.

C.1.2 Sumando con conexión EN y sin conexión ENO

Si el sumando tiene una conexión EN, pero no tiene una conexión ENO, se añaden las siguientes instrucciones AWL:

```
1           U     E      0.0  // Conexión EN
2           SPBNB _001        // Desplazar RLO al RB y saltar si RLO == 0
3           L     in1         // Parámetro del cuadro
4           L     in2         // Parámetro del cuadro
5           +I                // Suma
6           T     out         // Parámetro del cuadro
7    _001:  NOP   0
```

Después de la línea 1 el RLO contiene el resultado de la combinación lógica precedente. La instrucción SPBNB copia el RLO en el RB y activa el bit de primera consulta.

- Si el RLO es 0, se salta a la línea 7, no se realiza la suma y el RLO y RB son 0.
- Si el RLO era 1, no se salta, es decir, se realiza la suma. No se determina si se ha producido un error al sumar. El RLO y el RB son 1.

C.1.3 Sumando sin conexión EN y con conexión ENO

Si el sumando no tiene ninguna conexión EN, pero tiene una ENO, se añaden las siguientes instrucciones AWL:

```
1           L     in1     // Parámetro del cuadro
2           L     in2     // Parámetro del cuadro
3           +I            // Suma
4           T     out     // Parámetro del cuadro
5           UN    OV      // Detección de errores
6           SAVE          // Guardar error en RB
7           CLR           // Primera consulta
8           U     RB      // Desplazar RB a RLO
9           =     A      4.0
```

La suma se realiza en cualquier caso. En la línea 5 se determina si se ha producido un error al sumar, lo cual se guarda en el RB en la línea 6. La línea 7 activa el bit de primera consulta. En la línea 8 se vuelve a copiar el bit RB en el RLO, con lo cual se indica en la salida si la suma ha tenido éxito.

El bit RB no se modifica en las líneas 8 y 9, es decir que también indica si la suma ha tenido éxito.

C.1.4 Sumando sin conexión EN y sin conexión ENO

Si el sumando no tiene ninguna conexión EN y tampoco una ENO se añaden las siguientes instrucciones AWL:

```
1           L     in1     // Parámetro del cuadro
2           L     in2     // Parámetro del cuadro
3           +I            // Suma
4           T     out     // Parámetro del cuadro
5           NOP 0
```

La suma se ejecuta. El RLO y el bit RB no cambian.

C.2 Transferencia de parámetros

Los parámetros de un bloque se entregan o transfieren en forma de valores. En el caso de los bloques de función (FB), el bloque llamado utiliza una copia del valor del parámetro actual (real) que se encuentra en el DB de instancia. En el caso de las funciones (FC), la pila de datos locales contiene una copia del valor actual (real). Antes de la llamada se copian los valores INPUT en el DB de instancia, o en la pila LSTACK, según el caso. Después de la llamada se copian los valores OUTPUT en las variables. Dentro del bloque llamado se opera solamente con una copia. Las instrucciones AWL necesarias se encuentran en el bloque que efectúa la llamada y quedan ocultos para el usuario.

Nota

En caso de utilizar en una función marcas, entradas, salidas y entradas o salidas de la periferia como operandos actuales, éstas recibirán un tratamiento distinto del de los demás operandos. La actualización no se llevará a cabo a través de la pila LSTACK, sino directamente.

Excepción:

Si el parámetro formal correspondiente es un parámetro de entrada del tipo de datos BOOL, la actualización de los parámetros actuales tiene lugar por LSTACK.

Atención

Vigile que al programar el bloque llamado también se escriban los parámetros declarados como OUTPUT, de lo contrario los valores emitidos serán arbitrarios. En el caso de los bloques de función se obtiene el valor del DB de instancia memorizado en la última llamada, mientras que en el caso de las funciones se obtiene aquel valor que se encuentre casualmente en la pila de datos locales (LSTACK).

Tenga en cuenta los puntos siguientes:

- Inicialice en lo posible todos los parámetros OUTPUT.
- Evite utilizar instrucciones de activación (set) y desactivación (reset). Estas instrucciones dependen del RLO. Si el RLO es 0 se mantiene el valor casual.
- Si programa un salto en el bloque, tenga cuidado de no saltarse ninguna parte en la que se escriban parámetros OUTPUT. No se olvide del BEB y del efecto que tienen las instrucciones MCR.

Índice alfabético

D

E

F

I

J

L

M

N

O

P

R

S

T

U

V

W

X

Y

Z