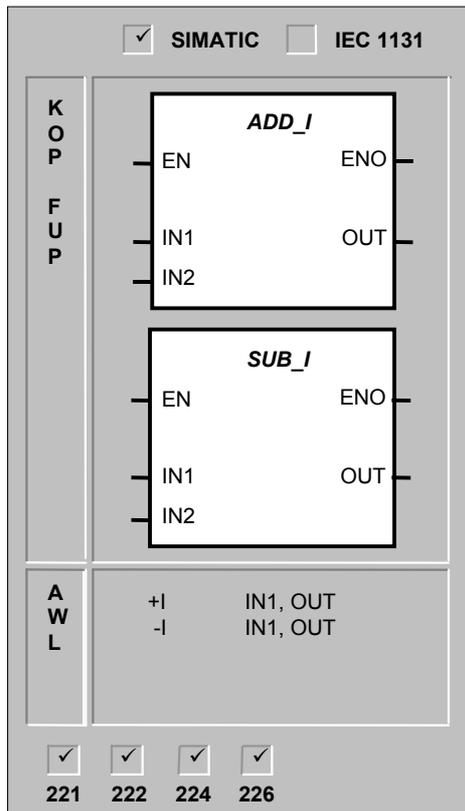


## Sumar y Restar enteros de 16 bits fija

## Operaciones aritméticas en coma fija

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VW, IW, QW, MW, SW, SMW, T, C, AC, LW, AIW, constante, *VD, *AC, *LD	INT
OUT	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, *VD, *AC, *LD	INT

[Acceder a la memoria de la CPU](#)     
 [ENO](#)     
 [Errores](#)     
 [Operaciones soportadas por las CPUs S7-200](#)     
 [Nemotécnica SIMATIC/Internacional](#)



Las operaciones **Sumar enteros de 16 bits** y **Restar enteros de 16 bits** suman/restan dos enteros de 16 bits, arrojando un resultado de 16 bits (OUT).

En KOP y FUP:  $IN1 + IN2 = OUT$   
 $IN1 - IN2 = OUT$

En AWL:  $IN1 + OUT = OUT$   
 $OUT - IN1 = OUT$

### Condiciones de error que ponen ENO a 0:

0006 (direccionamiento indirecto), SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución)

Estas operaciones afectan a las siguientes marcas especiales:  
 SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

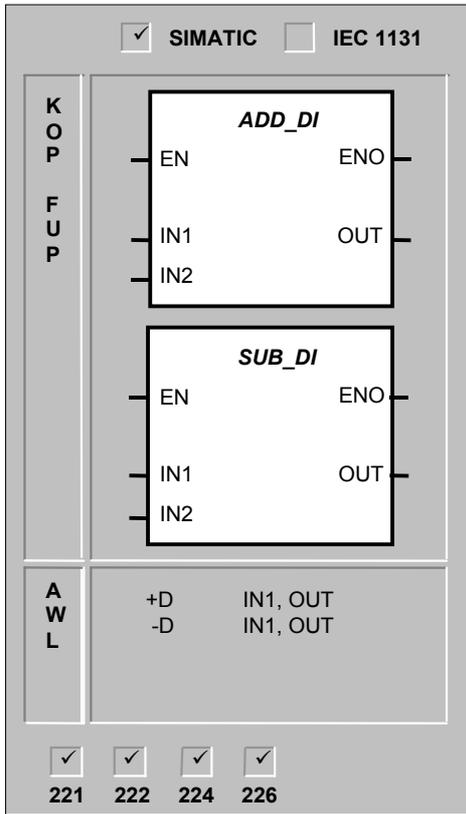
### Ejemplo

# Sumar y Restar enteros de 32 bits fija

## Operaciones aritméticas en coma fija

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VD, ID, QD, MD, SMD, SD, LD, AC, HC, constante, *VD, *AC, *LD	DINT
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	DINT

[Acceder a la memoria de la CPU](#)    
 [ENO](#)    
 [Errores](#)    
 [Operaciones soportadas por las CPUs S7-200](#)    
 [Nemotécnica SIMATIC/Internacional](#)



Las operaciones **Sumar enteros de 32 bits** y **Restar enteros de 32 bits** suman/restan dos enteros de 32 bits, arrojando un resultado de 32 bits (OUT).

En KOP y FUP:  $IN1 + IN2 = OUT$   
 $IN1 - IN2 = OUT$

En AWL:  $IN1 + OUT = OUT$   
 $OUT - IN1 = OUT$

**Condiciones de error que ponen ENO a 0:**

SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales: SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

## Multiplicar y Dividir enteros de 16 bits a enteros de 32 bits

### Operaciones aritméticas en coma fija



#### Entradas/salidas

IN1, IN2

OUT

[Acceder a la memoria de la CPU](#)

#### Operandos

VW, IW, QW, MW, SW, SMW, T, C, LW, AC, AIW, constante, \*VD, \*AC, \*LD

VD, ID, QD, MD, SMD, SD, LD, AC, \*VD, \*LD, \*AC

[ENO](#)

[Errores](#)

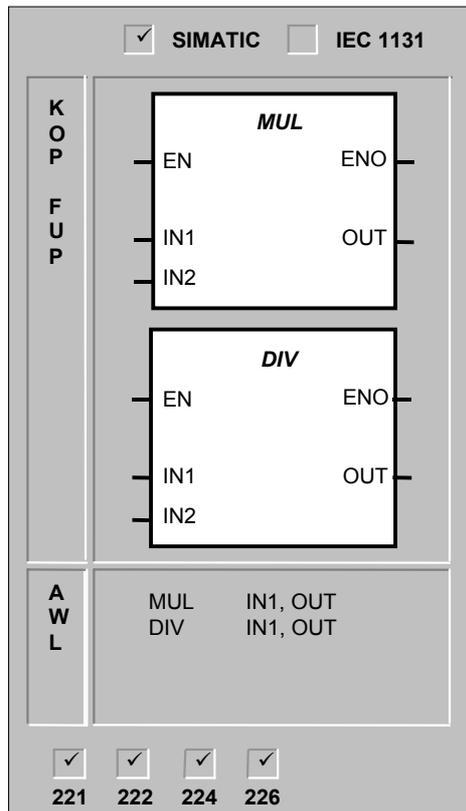
[Operaciones soportadas por las CPUs S7-200](#)

#### Tipos de datos

INT

DINT

[Nemotécnica SIMATIC/Internacional](#)



La operación **Multiplicar enteros de 16 bits a enteros de 32 bits** multiplica dos números enteros de 16 bits, arrojando un producto de 32 bits.

La operación **Dividir enteros de 16 bits a enteros de 32 bits** divide dos números enteros de 16 bits, arrojando un resultado de 32 bits compuesto de un cociente de 16 bits (los menos significativos) y un resto de 16 bits (los más significativos).

En la operación AWL de multiplicación, la palabra menos significativa (16 bits) del OUT de 32 bits se utiliza como uno de los factores.

En la operación AWL de división, la palabra menos significativa (16 bits) del OUT de 32 bits se utiliza como dividendo.

En KOP y FUP:  $IN1 * IN2 = OUT$   
 $IN1 / IN2 = OUT$

En AWL:  $IN1 * IN2 = OUT$   
 $OUT / IN1 = OUT$

#### Condiciones de error que ponen ENO a 0:

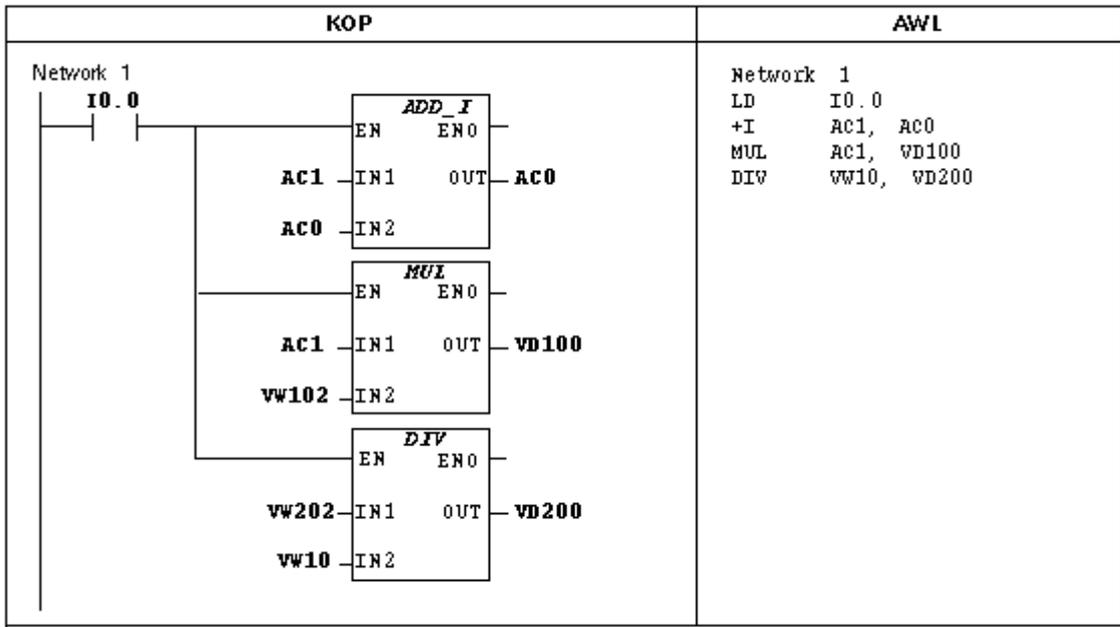
SM1.1 (desbordamiento), SM1.3 (división por cero), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales:

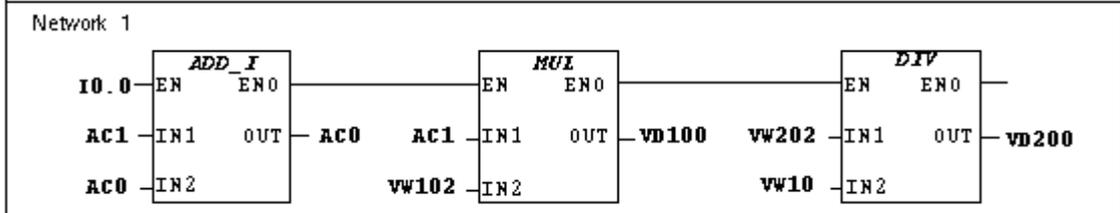
SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo); SM1.3 (división por cero)

Si se activa SM1.3 (división por cero) durante una operación de división, permanecerán inalterados los demás bits aritméticos de estado, así como los operandos de entrada originales. En otro caso, todos los bits aritméticos de estado asistidos contendrán el estado válido al finalizar la operación aritmética.

#### Ejemplo



**FUP**



**Aplicación**

Sumar	Multiplicar	Dividir
AC1 <input style="width: 50px;" type="text" value="4000"/> más AC0 <input style="width: 50px;" type="text" value="6000"/> igual a AC0 <input style="width: 50px;" type="text" value="10000"/>	AC1 <input style="width: 50px;" type="text" value="4000"/> multiplicado por VD100 <input style="width: 50px;" type="text" value="200"/> igual a VD100 <input style="width: 50px;" type="text" value="800000"/>	VD200 <input style="width: 50px;" type="text" value="4000"/> dividido por VW10 <input style="width: 50px;" type="text" value="41"/> igual a VD200 <input style="width: 50px;" type="text" value="23"/> <input style="width: 50px;" type="text" value="97"/> resto cociente VW200 VW202

Nota: VD100 contiene VW100 y VW102.  
VD200 contiene VW200 y VW202.

## Incrementar y Decrementar byte Operaciones aritméticas en coma fija

### Entradas/salidas

IN  
OUT

### Operandos

VB, IB, QB, MB, SB, SMB, LB, AC, constante, \*VD, \*AC, \*LD  
VB, IB, QB, MB, SB, SMB, LB, AC, \*VD, \*AC, \*LD

### Tipos de datos

BYTE  
BYTE

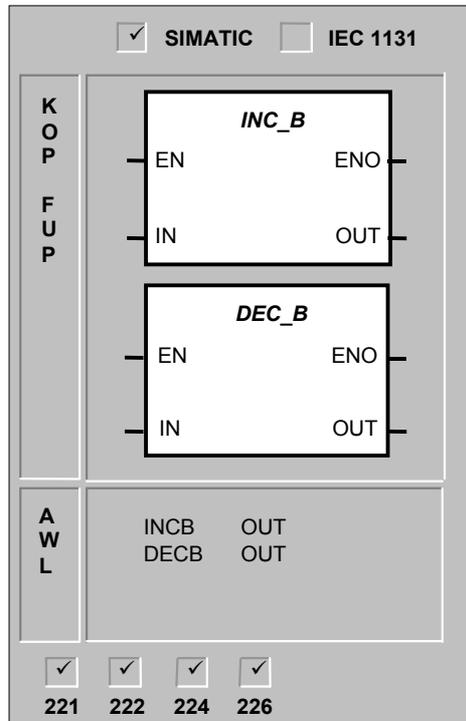
[Acceder a la memoria de la CPU](#)

[ENO](#)

[Errores](#)

[Operaciones soportadas por las CPUs S7-200](#)

[Nemotécnica SIMATIC/Internacional](#)



Las operaciones **Incrementar byte** y **Decrementar byte** suman/restan 1 al byte de entrada (IN) y depositan el resultado en la variable indicada por OUT.

Las operaciones Incrementar byte y Decrementar byte no llevan signo.

En KOP y FUP:  $IN + 1 = OUT$   
 $IN - 1 = OUT$

En AWL:  $OUT + 1 = OUT$   
 $OUT - 1 = OUT$

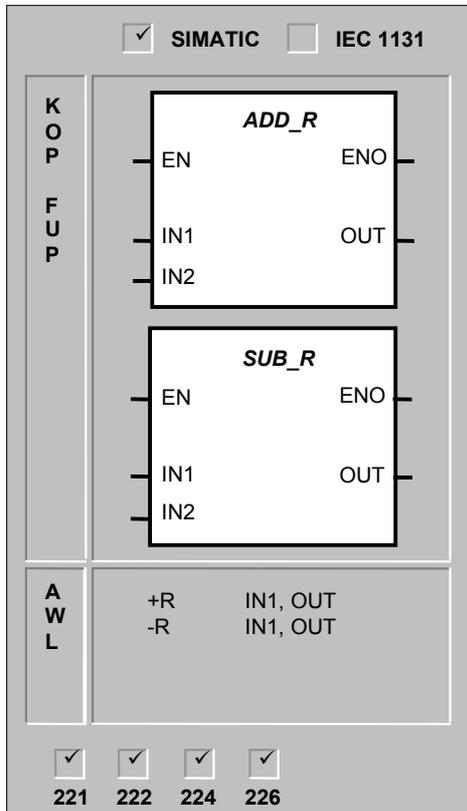
#### Condiciones de error que ponen ENO a 0:

SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales:  
SM1.0 (cero); SM1.1 (desbordamiento)

## Sumar y Restar reales Operaciones aritméticas en coma flotante

Entradas/salidas	Operandos	Tipos de datos
IN1, IN2	VD, ID, QD, MD, SD, SMD, AC, LD, constante, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SD, SMD, AC, LD, *VD, *AC, *LD	REAL
<a href="#">Acceder a la memoria de la CPU</a>	<a href="#">Errores</a>	<a href="#">Nemotécnica SIMATIC/Internacional</a>
	<a href="#">Operaciones soportadas por las CPUs S7-200</a>	



Las operaciones **Sumar reales** y **Restar reales** suman/restan dos números reales de 32 bits, dando como resultado un número real de 32 bits (OUT).

En KOP y FUP:  $IN1 + IN2 = OUT$   
 $IN1 - IN2 = OUT$

En AWL:  $IN1 + OUT = OUT$   
 $OUT - IN1 = OUT$

### Condiciones de error que ponen ENO a 0:

SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales:  
 SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

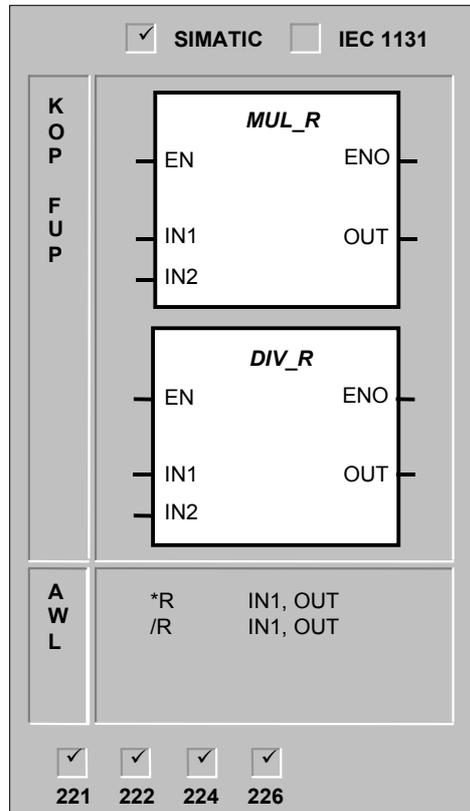
SM1.1 se utiliza para indicar errores de desbordamiento y valores no válidos. Si se activa SM1.1, el estado de SM1.0 y de SM1.2 no será válido y no se alterarán los operandos de entrada originales. Si SM1.1 y SM1.2 no se activan durante una operación de división, la operación aritmética habrá finalizado con un resultado válido, y tanto SM1.0 como SM1.2 contendrán un estado válido.

**Nota:** Los números reales (o números en coma flotante) se representan en el formato descrito en la norma ANSI/IEEE 754-1985 (precisión sencilla). Para obtener más información al respecto, consulte dicha norma.

### Ejemplo

## Multiplicar y Dividir reales Operaciones aritméticas en coma flotante

Entradas/salidas	Operandos	Tipos de datos	
IN1, IN2	VD, ID, QD, MD, SMD, SD, LD, AC, constante, *VD, *AC, *LD	REAL	
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	REAL	
<a href="#">Acceder a la memoria de la CPU</a>	<a href="#">ENO</a> <a href="#">Errores</a>	<a href="#">Operaciones soportadas por las CPUs S7-200</a>	<a href="#">Nemotécnica SIMATIC/Internacional</a>



La operación **Multiplicar reales** multiplica dos números reales de 32 bits, dando como resultado un número real de 32 bits (OUT).

La operación **Dividir reales** divide dos números reales de 32 bits, dando como resultado un cociente de número real de 32 bits.

En KOP y FUP:  $IN1 * IN2 = OUT$   
 $IN1 / IN2 = OUT$

En AWL:  $IN1 * IN2 = OUT$   
 $OUT / IN1 = OUT$

### Condiciones de error que ponen ENO a 0:

SM1.1 (desbordamiento), SM1.3 (división por cero), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales:

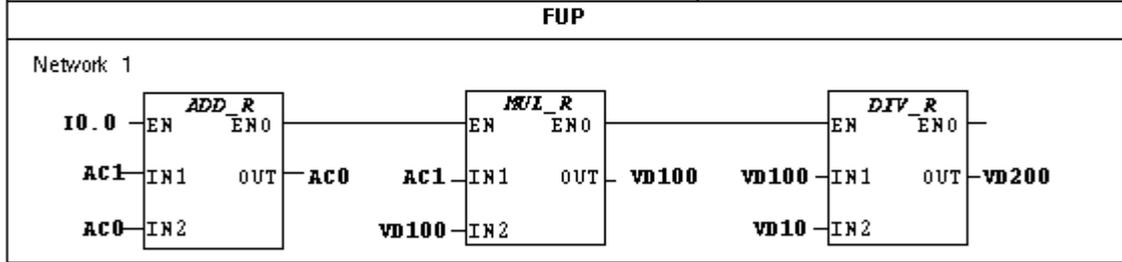
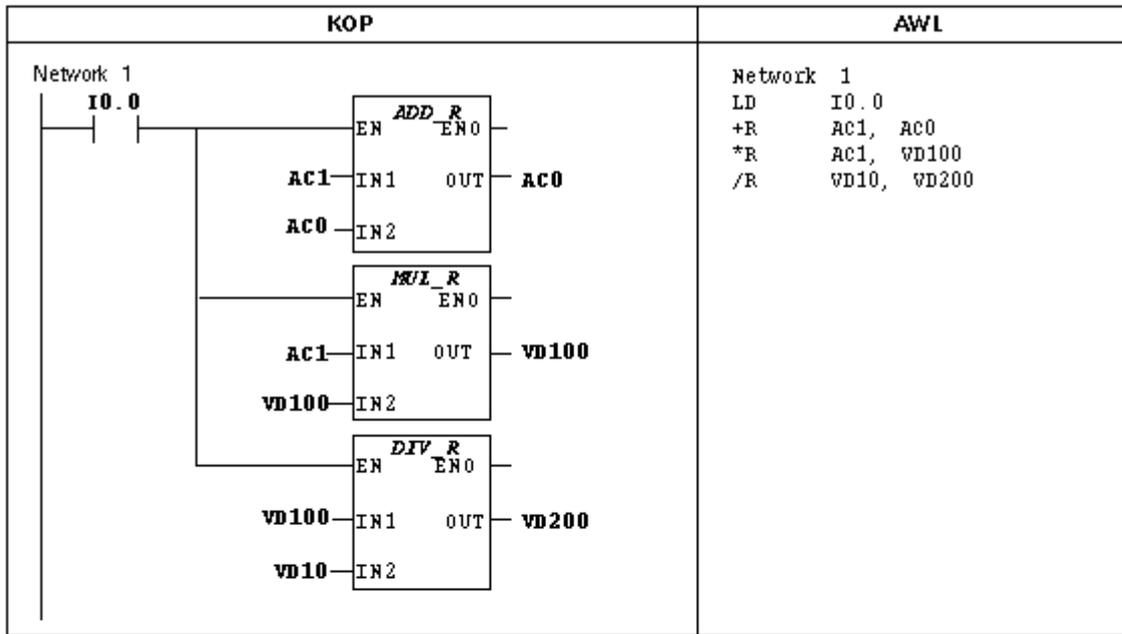
SM1.0 (cero); SM1.1 (desbordamiento o valor no válido generado durante la operación o parámetro de entrada no válido); SM1.2 (negativo); SM1.3 (división por cero)

Si se activa SM1.3 durante una operación de división, permanecerán inalterados los demás bits aritméticos de estado, así como los operandos de entrada originales. SM1.1 se utiliza para indicar errores de desbordamiento y valores no válidos. Si se activa SM1.1, el estado de SM1.0 y de SM1.2 no será válido y no se alterarán los operandos de entrada originales. Si SM1.1 y SM1.3 no se activan durante una operación de división, la operación aritmética habrá finalizado con un resultado válido, y tanto SM1.0 como SM1.2 contendrán un estado válido.

### Nota

Los números reales (o números en coma flotante) se representan en el formato descrito en la norma ANSI/IEEE 754-1985 (precisión sencilla). Para obtener más información al respecto, consulte dicha norma.

### Ejemplo



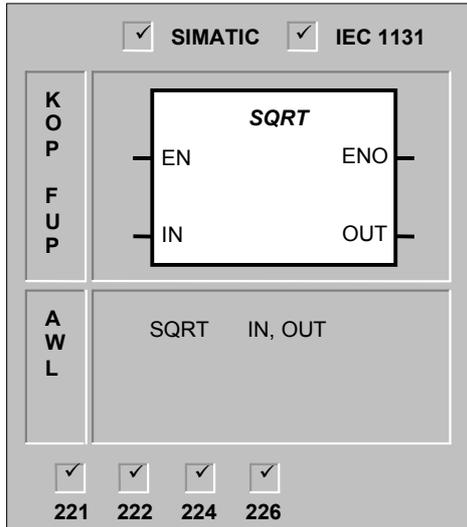
**Aplicación**

<p><b>Sumar</b></p> <p>AC1 <input type="text" value="4000.0"/></p> <p>más</p> <p>AC0 <input type="text" value="6000.0"/></p> <p>igual a</p> <p>AC0 <input type="text" value="10000.0"/></p>	<p><b>Multiplicar</b></p> <p>AC1 <input type="text" value="400.00"/></p> <p>multiplicado por</p> <p>VD100 <input type="text" value="200.0"/></p> <p>igual a</p> <p>VD100 <input type="text" value="800000.0"/></p>	<p><b>Dividir</b></p> <p>VD200 <input type="text" value="4000.0"/></p> <p>dividido por</p> <p>VD10 <input type="text" value="41.0"/></p> <p>igual a</p> <p>VD200 <input type="text" value="97.5609"/></p>
---	--	---

## Raíz cuadrada Operaciones aritméticas en coma flotante (SIMATIC)

### Funciones numéricas (IEC)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SMD, SD, LD, AC, constante, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	REAL
<a href="#">Acceder a la memoria de la CPU</a>	<a href="#">ENO</a> <a href="#">Errores</a> <a href="#">Operaciones soportadas por las CPUs S7-200</a>	<a href="#">Nemotécnica SIMATIC/Internacional</a>



La operación **Raíz cuadrada** extrae la raíz cuadrada de un número real de 32 bits (IN), dando como resultado un número real de 32 bits (OUT), como muestra la ecuación:

$$\sqrt{IN} = OUT$$

#### Condiciones de error que ponen ENO a 0:

SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales:

SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

SM1.1 se utiliza para indicar errores de desbordamiento y valores no válidos. Si se activa SM1.1, el estado de SM1.0 y de SM1.2 no será válido y no se alterarán los operandos de entrada originales. Si SM1.1 y SM1.2 no se activan durante una operación de división, la operación aritmética habrá finalizado con un resultado válido, y tanto SM1.0 como SM1.2 contendrán un estado válido.

## Regulación PID Operaciones aritméticas en coma flotante (SIMATIC)

### Funciones numéricas (IEC)

#### Entradas/salidas

TBL

LOOP

[Acceder a la memoria de la CPU](#)

#### Operandos

VB

Constante(0 a 7)

[ENO](#)

[Errores](#)

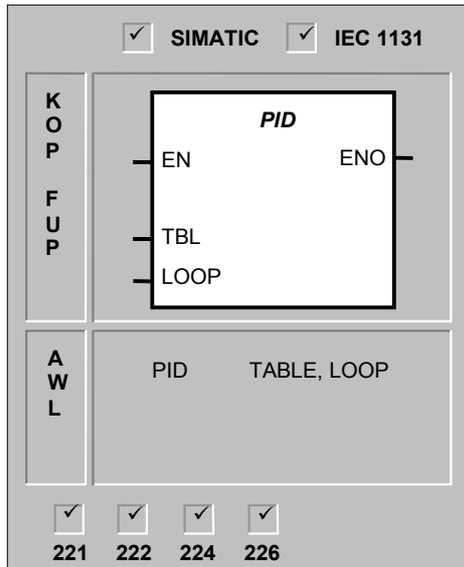
#### Tipos de datos

BYTE

BYTE

[Operaciones soportadas por las CPUs S7-200](#)

[Nemotécnica SIMATIC/Internacional](#)



La operación **Regulación PID** ejecuta el cálculo de un lazo de regulación PID en el LOOP referenciado en base a las informaciones de entrada y configuración definidas en TABLE (TBL).

La operación PID (lazo de regulación con acción proporcional, integral, derivada) sirve para ejecutar el cálculo PID. Para habilitar el cálculo PID, el primer nivel de la pila lógica (TOS) deberá estar ON (circulación de corriente). Esta operación tiene dos operandos: una dirección TBL que constituye la dirección inicial de la tabla del lazo y un número LOOP que es una constante comprendida entre 0 y 7. Un programa sólo admite ocho operaciones PID. Si se utilizan dos o más operaciones PID con el mismo número de lazo (aunque tengan diferentes direcciones de tabla), los dos cálculos PID se interferirán mutuamente siendo impredecible la salida resultante.

La tabla del lazo almacena nueve parámetros que sirven para controlar y supervisar la operación del mismo. Incluye el valor actual y previo de la variable del proceso (valor real), la consigna, la salida o magnitud manipulada, la ganancia, el tiempo de muestreo, el tiempo de acción integral, el tiempo de acción derivada y la suma integral (bias).

#### Condiciones de error que ponen ENO a 0:

SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

El compilador de la CPU generará un error de compilación (error de margen) y la compilación fallará si los operandos de la dirección inicial o del número de lazo PID en la tabla del lazo están fuera de margen.

La operación PID no verifica si ciertos valores de entrada en la tabla del lazo son conformes con los límites de margen. Es decir, el usuario debe asegurarse de que la variable del proceso y la consigna (al igual que la suma integral y la variable del proceso previa, si se utilizan como entradas) sean números reales comprendidos entre 0.0 y 1.0.

Si se detecta algún error al ejecutar las operaciones aritméticas del cálculo PID se activa la marca SM1.1 (desbordamiento o valor no válido) y se finaliza la ejecución de la operación PID. (La actualización de los valores de salida en la tabla del lazo puede ser incompleta por lo que deberán descartarse dichos valores y corregir el valor de entrada que ha causado el error matemático antes de efectuar la siguiente ejecución de la operación de regulación PID).

#### Intervalo de muestreo:

Para poder realizar el cálculo PID con el intervalo de muestreo deseado, la operación PID deberá ejecutarse bien dentro de una rutina de interrupción temporizada o desde el programa principal, a intervalos controlados por un temporizador. El tiempo de muestreo debe definirse en calidad de entrada para la operación PID a través de la tabla del lazo.

#### Uso del Asistente PID

STEP 7-Micro/WIN 32 incorpora el Asistente PID que le ayuda a definir un algoritmo PID para un proceso de control de bucle cerrado.– Seleccione el comando de menú **Herramientas>Asistente de operaciones** y elija "PID" en la ventana del Asistente.

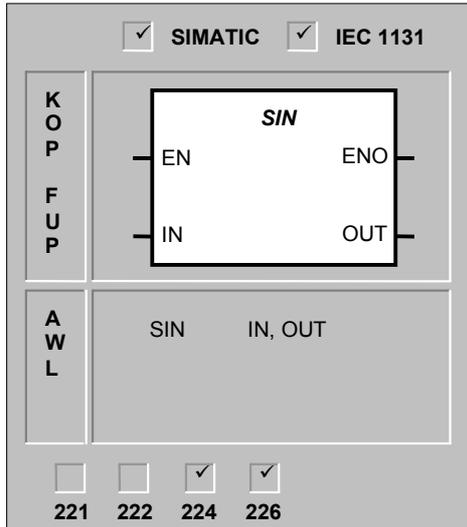
[Asistente PID](#)

[Algoritmo PID](#)

## Seno Operaciones aritméticas en coma flotante (SIMATIC)

### Funciones numéricas (IEC)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SMD, SD, LD, AC, constante, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	REAL
<a href="#">Acceder a la memoria de la CPU</a>	<a href="#">ENO</a> <a href="#">Errores</a>	<a href="#">Operaciones soportadas por las CPUs S7-200</a> <a href="#">Nemotécnica SIMATIC/Internacional</a>



La operación **Seno** calcula el seno del valor del ángulo IN y deposita el resultado en OUT. El ángulo de entrada se indica en radianes. Si el ángulo se indica en grados, es preciso multiplicarlo por  $\text{PI}()/180$  para convertirlo a radianes.

#### Condiciones de error que ponen ENO a 0:

SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

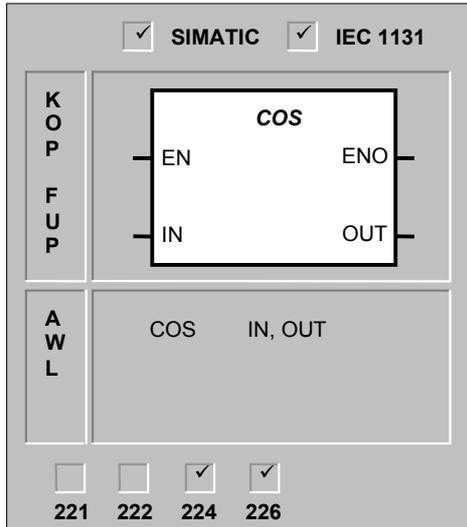
Estas operaciones afectan a las siguientes marcas especiales:

SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

## Coseno Operaciones aritméticas en coma flotante (SIMATIC)

### Funciones numéricas (IEC)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SMD, SD, LD, AC, constante, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	REAL
<a href="#">Acceder a la memoria de la CPU</a>	<a href="#">ENO</a> <a href="#">Errores</a>	<a href="#">Operaciones soportadas por las CPUs S7-200</a> <a href="#">Nemotécnica SIMATIC/Internacional</a>



La operación **Coseno** calcula el coseno del valor del ángulo IN y deposita el resultado en OUT. El ángulo de entrada se indica en radianes. Si el ángulo se indica en grados, es preciso multiplicarlo por  $\text{PI}()/180$  para convertirlo a radianes.

#### Condiciones de error que ponen ENO a 0:

SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

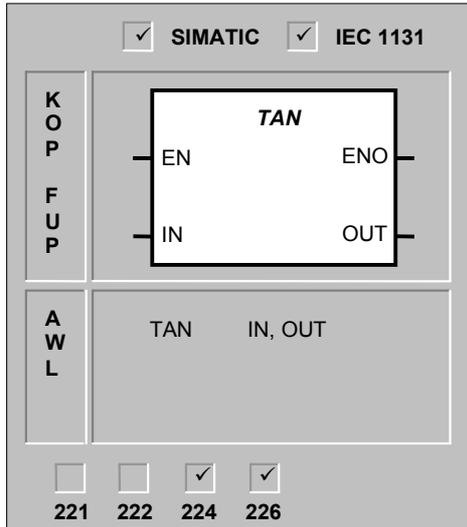
Estas operaciones afectan a las siguientes marcas especiales:

SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

## Tangente Operaciones aritméticas en coma flotante (SIMATIC)

### Funciones numéricas (IEC)

Entradas/salidas	Operandos	Tipos de datos
IN	VD, ID, QD, MD, SMD, SD, LD, AC, constante, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *AC, *LD	REAL
<a href="#">Acceder a la memoria de la CPU</a>	<a href="#">ENO</a> <a href="#">Errores</a>	<a href="#">Operaciones soportadas por las CPUs S7-200</a> <a href="#">Nemotécnica SIMATIC/Internacional</a>



La operación **Tangente** calcula la tangente del valor del ángulo IN y deposita el resultado en OUT. El ángulo de entrada se indica en radianes. Si el ángulo se indica en grados, es preciso multiplicarlo por  $\text{PI}()/180$  para convertirlo a radianes.

#### Condiciones de error que ponen ENO a 0:

SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales:

SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

## Exponencial natural (SIMATIC) Operaciones aritméticas en coma flotante

### Funciones numéricas (IEC)

#### Entradas/salidas

IN

OUT

[Acceder a la memoria de la CPU](#)

[ENO](#)

#### Operandos

VD, ID, QD, MD, SMD, SD, LD, AC, constante, \*VD, \*AC, \*LD

VD, ID, QD, MD, SMD, SD, LD, AC, \*VD, \*AC, \*LD

[Errores](#)

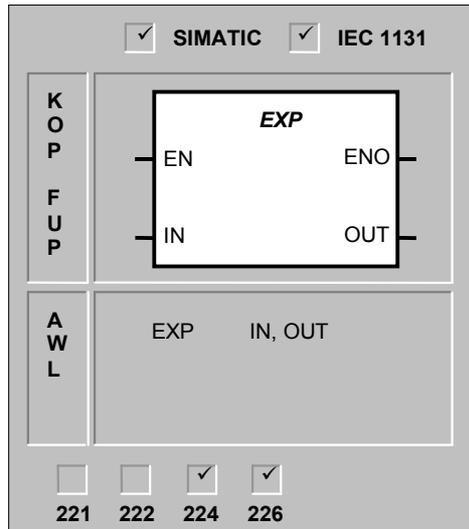
[Operaciones soportadas por las CPUs S7-200](#)

#### Tipos de datos

REAL

REAL

[Nemotécnica SIMATIC/Internacional](#)



La operación **Exponencial natural** ejecuta el cálculo exponencial de la constante e elevada a la potencia del valor de IN y deposita el resultado en OUT. La constante e equivale a 2,71828182845904, siendo ésta la base del logaritmo natural.

#### Condiciones de error que ponen ENO a 0:

SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales:

SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)

## Logaritmo natural (SIMATIC)

## Operaciones aritméticas en coma flotante

## Funciones numéricas (IEC)

### Entradas/salidas

IN

OUT

[Acceder a la memoria de la CPU](#)

[ENO](#)

### Operandos

VD, ID, QD, MD, SMD, SD, LD, AC, constante, \*VD, \*AC, \*LD

VD, ID, QD, MD, SMD, SD, LD, AC, \*VD, \*AC, \*LD

[Errores](#)

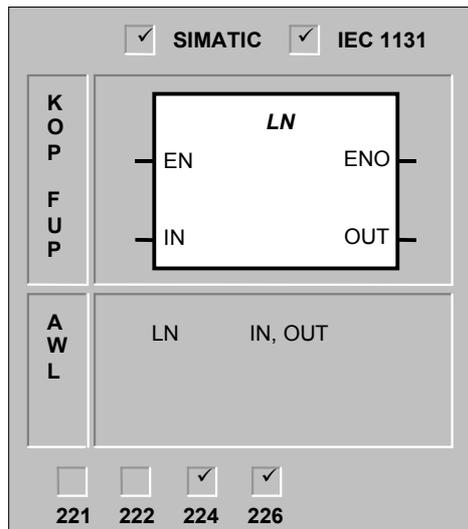
[Operaciones soportadas por las CPUs S7-200](#)

### Tipos de datos

REAL

REAL

[Nemotécnica SIMATIC/Internacional](#)



La operación **Logaritmo natural** calcula el logaritmo natural del valor de IN y deposita el resultado en OUT. El logaritmo natural se basa en la constante e (2,71828182845904).

### Condiciones de error que ponen ENO a 0:

SM1.1 (desbordamiento), SM4.3 (tiempo de ejecución), 0006 (direccionamiento indirecto)

Estas operaciones afectan a las siguientes marcas especiales:

SM1.0 (cero); SM1.1 (desbordamiento); SM1.2 (negativo)