PROGRAMACIÓN AVANZADA DE PLC’s

OMRON®

CMP, FUN(20)/1

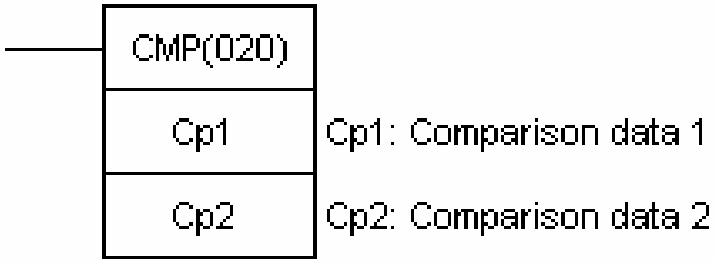
■ LA INSTRUCCIÓN CMP REALIZA LA COMPARACIÓN ENTRE DOS DATOS DE 16 BIT, CONTENIDOS EN DOS CANALES.

■ EL RESULTADO DE LA COMPRACIÓN SOLO SE REFLEJA EN UNOS RELES ESPECIALES DE ”<“, “=“, o “>”.

“P_LT” Indicador de Menor Que (LT)

“P_EQ” Indicador de iguales (EQ)

“P_GT” Indicador de Mayor Que (GT)



■ LAS ÁREAS DE DATOS UTILIZABLES EN LA COMPARACIÓN SON :

— S:#, IR, SR, HR, TIM, CNT

— D: IR, HR

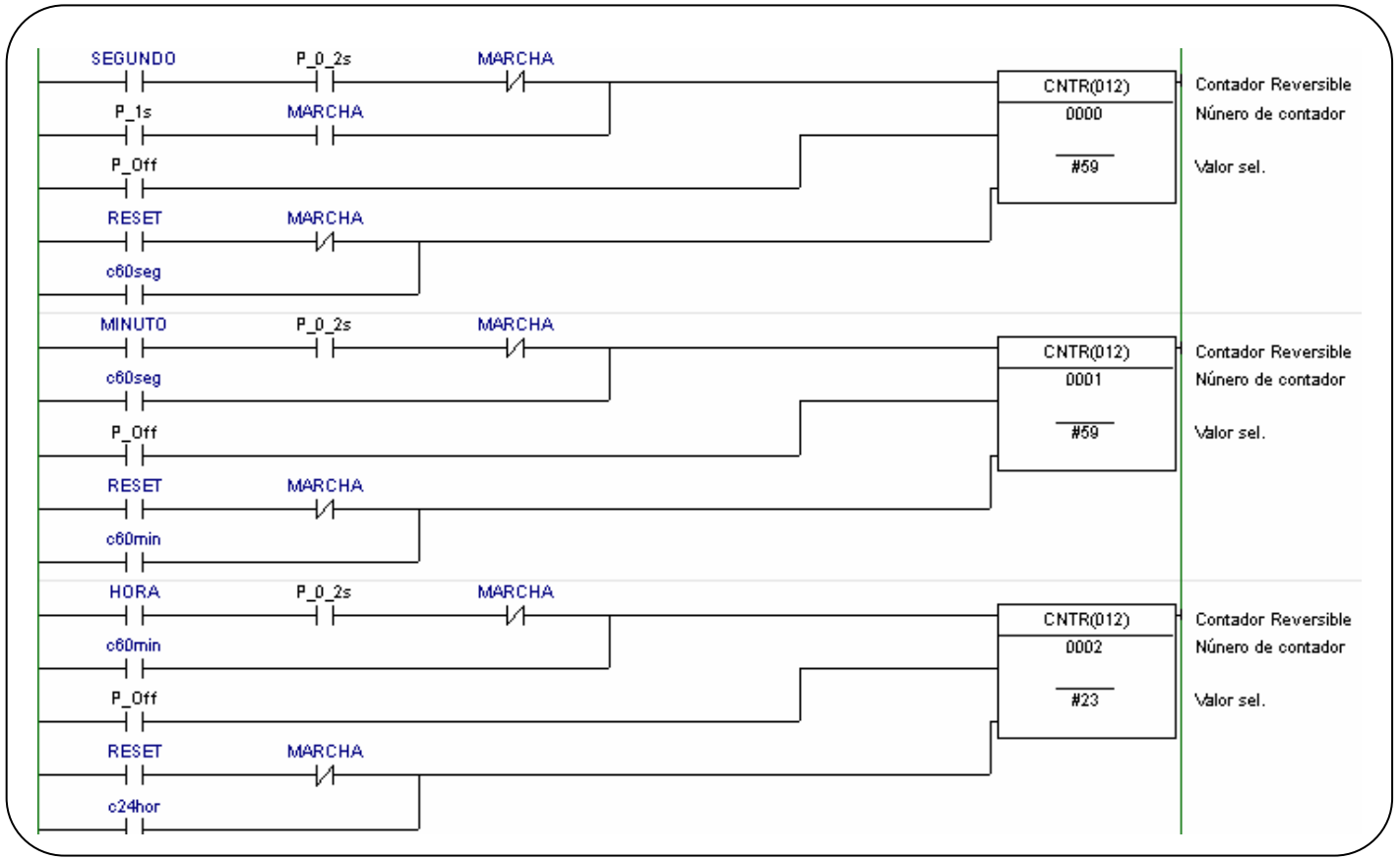
CMP, FUN(20)/2

■ SEGÚN LOS VALORES QUE TENGAN LOS DATOS A COMPARAR Cp1 Y Cp2 SE TIENEN LOS SIGUIENTES CASOS:

DATOS	RESULTADO	P_LT	P_EQ	P_GT
Cp1 < Cp2	Menor	1	0	0
Cp1 = Cp2	Igual	0	1	0
Cp1 > Cp2	Mayor	0	0	1

EJEMPLO CMP(20)/1, ALARMA DE RELOJ

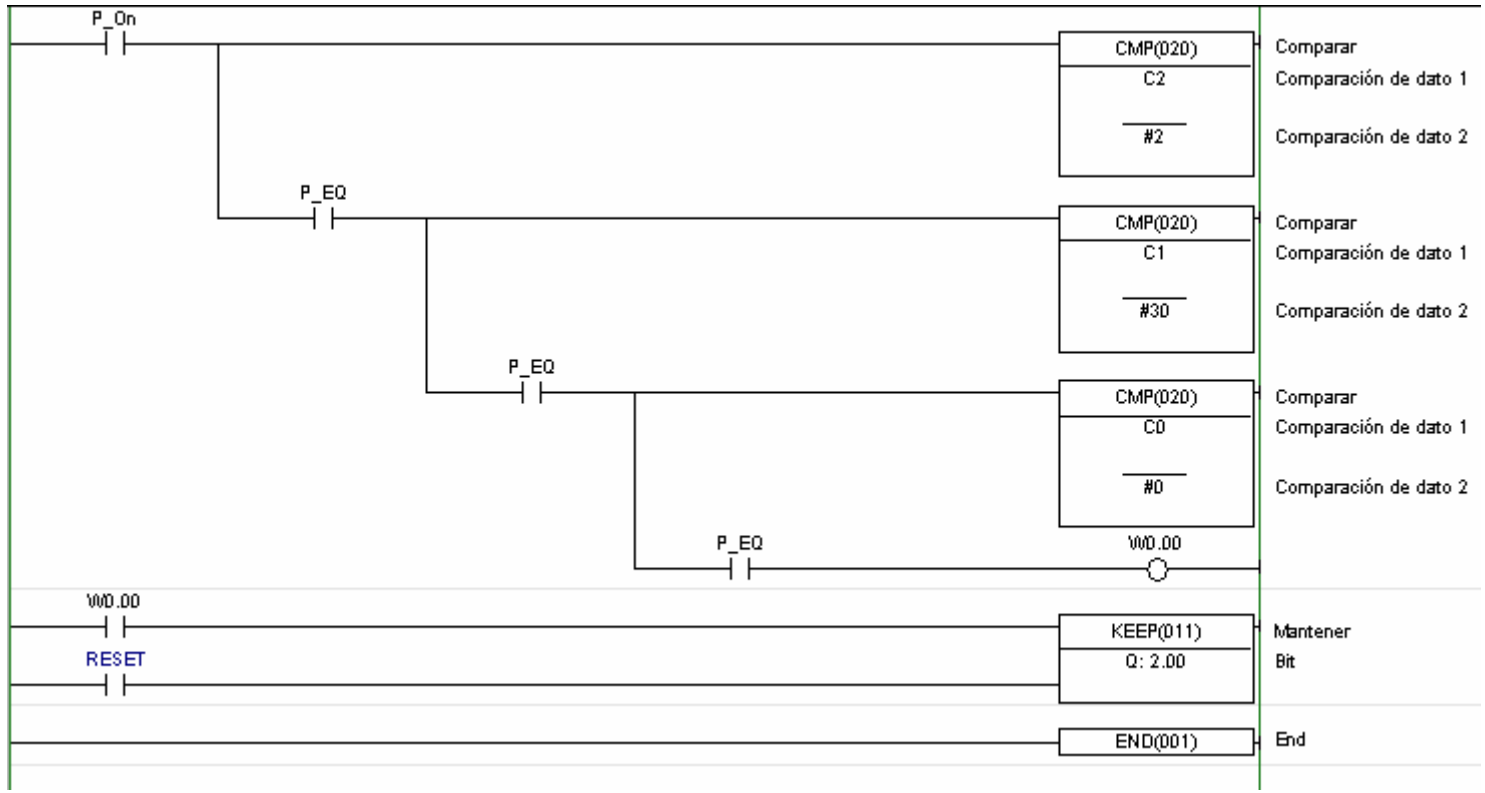
- LAS HORAS MINUTOS Y SEGUNDOS ESTAN PREVIAMENTE PROGRAMADOS EN TRE CONTADORES C2, C1 Y C0.
- CANDO SE ALCANCE LAS HORAS, MINUTOS Y SEGUNDOS AJUSTADOS POR PROGRAMA, SE ACTIVARÁ LA SALIDA “2.00”.
- LA SALIDA SE DESACTIVARÁ AL PULSAR LE ENTRADA DE REPOSICIÓN “0.04”.



OMRON®

EJEMPLO CMP(20)/2, ALARMA DE RELOJ

- SE PUEDE PROGRAMAR UNA SERIE DE COMPARACIONES EN CADENA.



COMPARACIONES EN LINEA(CS1)/2

- Comparan dos datos S1 y S2.
- Los datos a comparar pueden ser:
  - Formato: con o sin signo.
  - Longitud: de 1 (S1 con S2) ó 2 (S1 y S1+1 con S2 y S2+1) palabras.
- Son instrucciones intermedias: Se pueden conectar como LD, AND y OR.

COMPARACIONES EN LINEA (CS1)/1

•Hay disponibles un total de 24 instrucciones de comparación. Estas pueden utilizar varias combinaciones de símbolos y opciones. Si no se especifican opciones, la comparación será para un sólo canal sin signo.

Los tipos básicos son:		Cada tipo puede tener las opciones:	
=	Igual		
<>	Diferente		SIN signo
<	Menor	S	Signo
<=	Menor o igual	L	Doble Longitud
>	Mayor	SL	Doble Longitud con Signo.
>=	Mayor o igual		

(Ejemplos: LD=(300), #0, D0 AND=S(302),-2, D0)

COMPARACIONES EN LINEA(CS1)/3

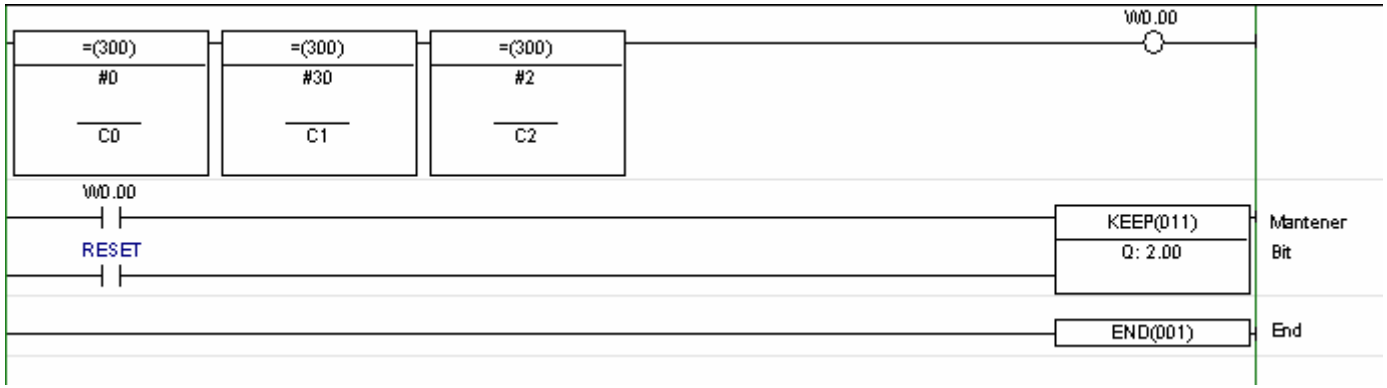
Símbolo	Formato	Longitud
=(300)	-: Sin signo	-: 1 palabra
=L(301)	-: Sin signo	L: 2 palabras
=S(302)	S: Con signo	-: 1 palabra
=SL(303)	S: Con signo	-: 2 palabras
<>(305)	-: Sin signo	-: 1 palabra
<>L(306)	-: Sin signo	L: 2 palabras
<>S(307)	S: Con signo	-: 1 palabra
<>SL(308)	S: Con signo	-: 2 palabras
<(310)	-: Sin signo	-: 1 palabra
<L(311)	-: Sin signo	L: 2 palabras
<S(312)	S: Con signo	-: 1 palabra
<SL(313)	S: Con signo	-: 2 palabras
<=(315)	-: Sin signo	-: 1 palabra
<=L(316)	-: Sin signo	L: 2 palabras
<=S(317)	S: Con signo	-: 1 palabra
<=SL(318)	S: Con signo	-: 2 palabras
>(320)	-: Sin signo	-: 1 palabra
>L(321)	-: Sin signo	L: 2 palabras
>S(322)	S: Con signo	-: 1 palabra
<SL(323)	S: Con signo	-: 2 palabras
>=(325)	-: Sin signo	-: 1 palabra
>=L(326)	-: Sin signo	L: 2 palabras
>=S(327)	S: Con signo	-: 1 palabra
>=SL(328)	S: Con signo	-: 2 palabras

.....COMPARACIONES EN LINEA (CS1)

■ CON EL CS1 SE PUEDE PROGRAMAR UNA SERIE DE COMPARACIONES EN LINEA QUE DEN COMO RESULTADO LA ACTIVACIÓN DE LA ALARMA.

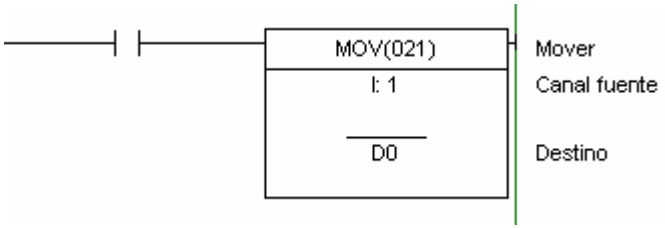
ANTES: CMP(020) SERIE C

AHORA: CS1



MOV, FUN(21)

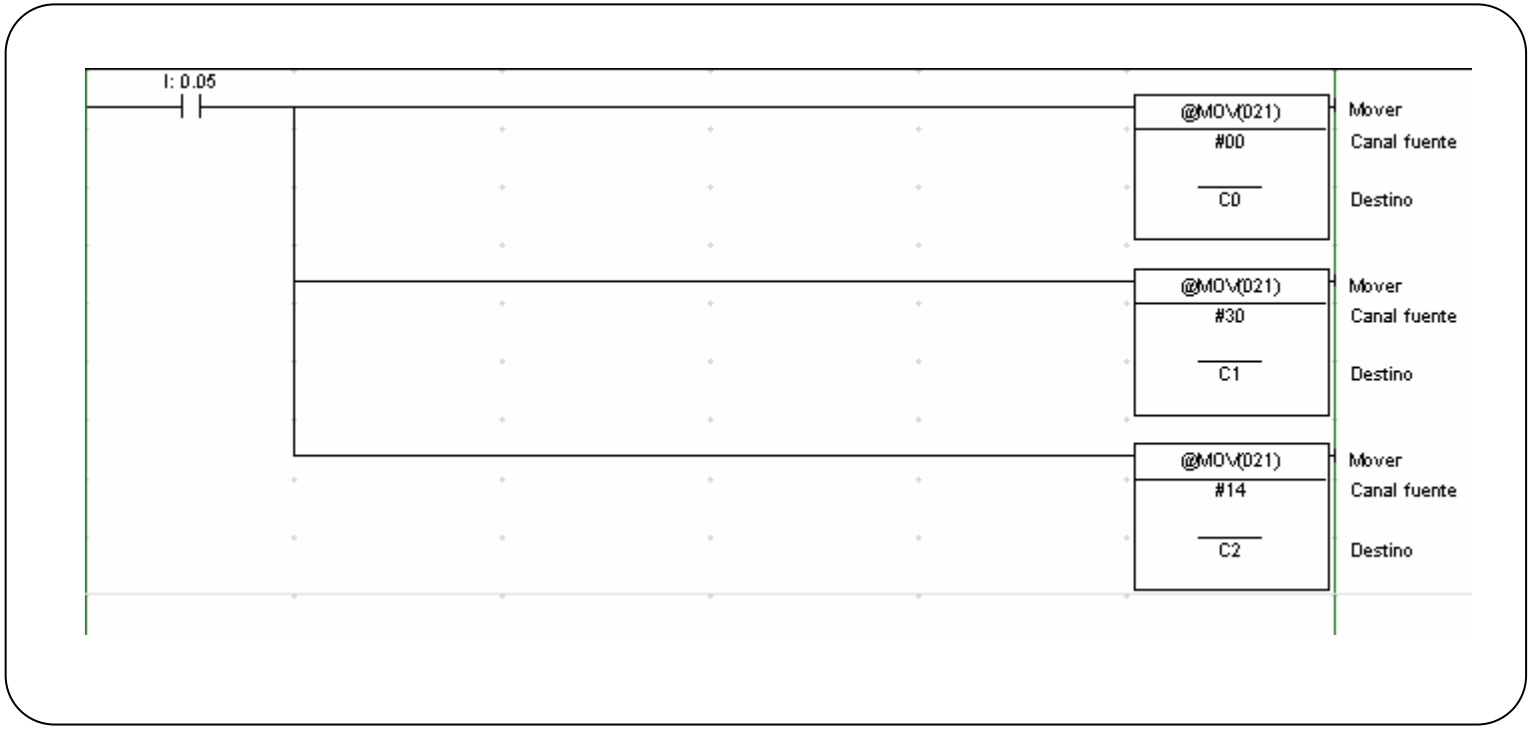
- LA INSTRUCCIÓN MOV REALIZA EL MOVIMIENTO DE UN DATO DE 16 BIT, DESDE UN CANAL A OTRO.
- EL CONTENIDO DEL CANAL FUENTE S SE TRANSFIERE AL CANAL DESTINO D.



- LAS ÁREAS DE DATOS UTILIZABLES EN LA TRANSFERENCIA SON :
  - S:Fuente: CIO, W, H, A, T, C, D, E, E?_, @D, @E, @E?_, *D, *E, *E?_, #, DR, ,IR
  - D:Destino: CIO, W, H, A, T, C, D, E, E?_, @D, @E, @E?_, *D, *E, *E?_, DR, ,IR

EJEMPLO. AJUSTE DEL RELOJ A LAS 14:30:00

■ AL PULSAR EL ENTRADA 0.05 SE PRODUCE EL MOVIMIENTO DE LOS DATOS 00, 30 Y 14 A LOS CONTADORES C0, C1 y C2.



Direccionamientos

Addessing

OMRON

Direccionamientos (Serie C y CS1)

- Existen varios tipos de direccionamientos:
  - Inmediato (#, &)
  - Directo (CIO, W, H, A, T, C, D, E, En_)
  - Indirecto (D, E, En_) sustituye a IEMS(-)
    - Dirección en BCD (*D, *E, *En_)
    - Dirección en Binario (@D, @E, @En_)
  - Indirecto Indexado (IR)
    - ,IR0
    - +234,IR0
    - DR0,IR0
    - ,IR0++

DIRECCIONAMIENTO INDIRECTO

■ Cuando para un operando se especifica el área de DM, se puede utilizar una dirección indirecta.

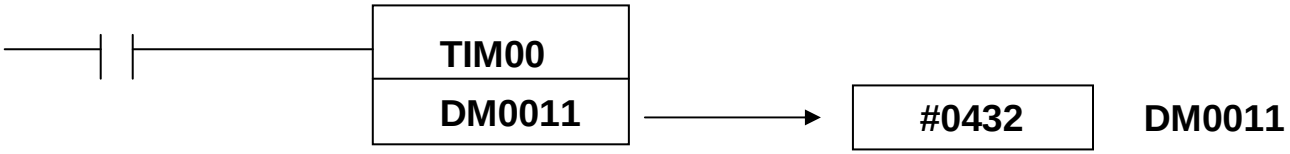
— Para diferenciar el direccionamiento de DM indirecto se coloca un asterisco delante de DM : ***DM**

■ Cuando se especifica una dirección indirecta de DM, el canal DM designado contendrá la dirección del canal DM que contiene el dato que se utilizará como operando de la instrucción.

■ Cuando se utilice direccionamiento indirecto, la dirección del canal deseado debe estar en BCD y debe especificar un canal comprendido en área de DM.

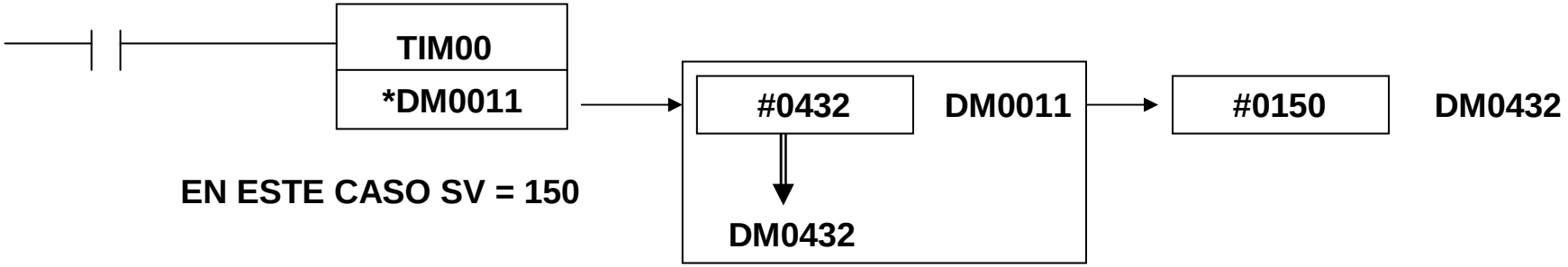
DIRECCIONAMIENTO INDIRECTO

■ Normalmente la variable especificada por una cierta instrucción es tal que la instrucción opera con el dato especificado en la variable especificada.



EN ESTE CASO SV = 432

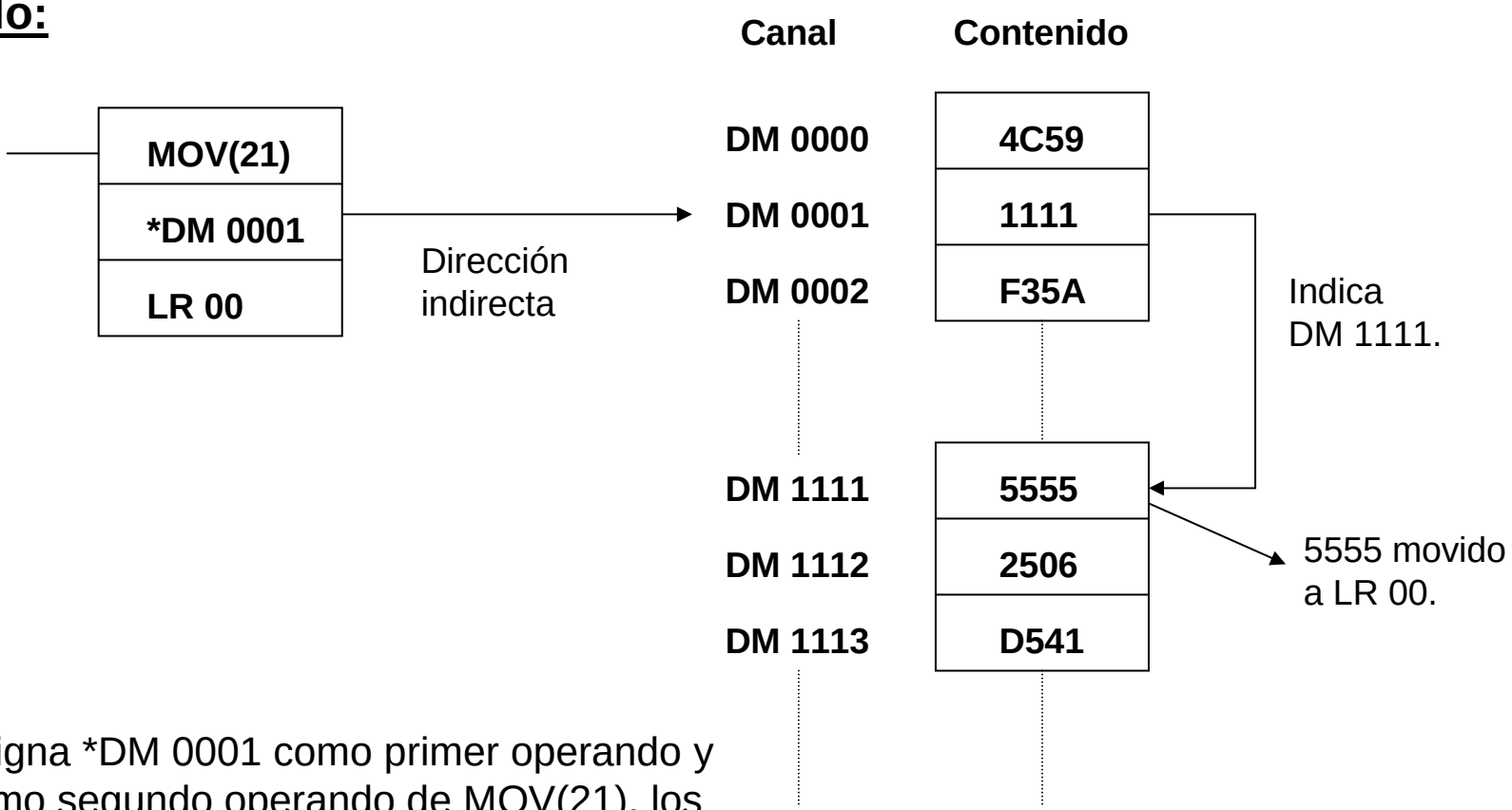
■ El direccionamiento indirecto permite especificar un dato por la dirección de DM donde ése dato está contenido (la dirección es la variable contenida).



EN ESTE CASO SV = 150

DIRECCIONAMIENTO INDIRECTO

Ejemplo:

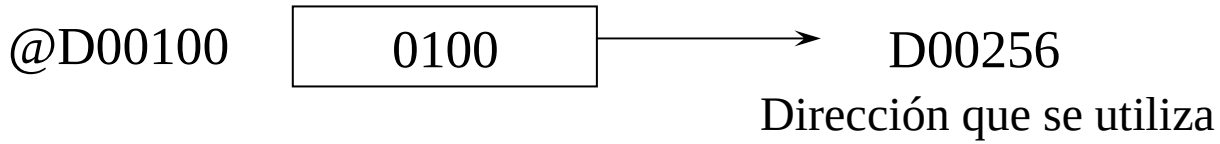


Si se designa *DM 0001 como primer operando y LR 00 como segundo operando de MOV(21), los contenidos de DM0001 son 1111 y DM 1111 contiene 5555, el valor 5555 será movido a LR 00.

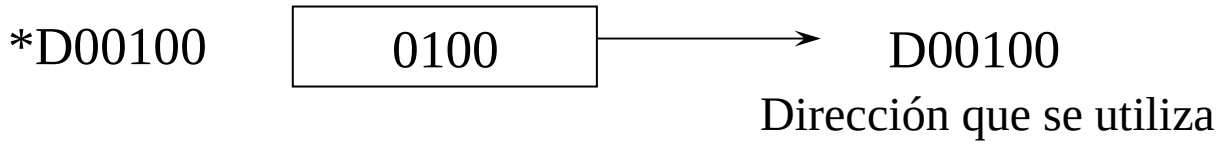
Direccionamiento Indirecto de DMs (CS1)

✓ Los DMs se pueden direccionar indirectamente de dos formas:

1) Direccionamiento en Modo-Binario (@D). Se puede direccionar todo el área de DMs (D00000 a D32767) con los valores en hexadecimal 0000 a 7FFF.



2) Direccionamiento en Modo-BCD (*D). En este modo sólo parte del área de DMs (D00000 a D09999) puede ser direccionada indirectamente con los valores BCD de 0000 a 9999.



Ejemplos

MOV(21)
&12
W000

Escribe C (12 traducido a binario) en W000
Inmediato traducido a binario
Directo

MOV(21)
#1A34
W000

Escribe 1A34 binario en W000
Inmediato binario
Directo

MOV(21)
W000
W001

Escribe el contenido de W000 en W001
Directo
Directo

MOV(21)
#1A
*D00000

Escribe 1A en la dirección(BCD)indicada en D00000
Inmediato
Indirecto BCD

Ejemplos

Escribe el contenido de W000 en la dirección (Binaria) indicada en D00000

MOV(21)
W000
@D00000

Directo

Indirecto Binario

MOV(21)
#1A
,IR0

Escribe 1A en la dirección indicada en IR0

Inmediato

Indexado

MOV(21)
#1A
+23,IR0

Escribe 1A en la dirección indicada en (IR0+23)

Inmediato

Indexado

MOV(21)
#1A
DR0,IR0

Escribe 1A en la dirección indicada en (IR0+DR0)

Inmediato

Indexado

Ejemplos

MOV(21)
#1A
,IR0+

Escribe 1A en la dirección indicada en IR0 y aumenta IR0 una unidad

Inmediato

Indexado

MOV(21)
#1A
,IR0++

Escribe 1A en la dirección indicada en IR0 y aumenta IR0 dos unidades

Inmediato

Indexado

MOV(21)
#1A
,-IR0

Escribe 1A en la dirección indicada en (IR0-1) y disminuye IR0 una unidad

Inmediato

Indexado

MOV(21)
#1A
,--IR0

Escribe 1A en la dirección indicada en (IR0-2) y disminuye IR0 dos unidades

Inmediato

Indexado

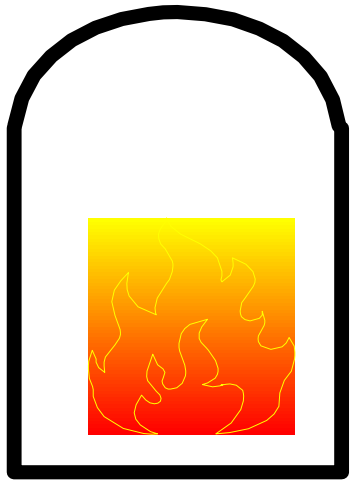
DIRECCIONAMIENTO INDIRECTO

EJEMPLO

ALMACENAR LA TEMPERATURA EN UN HORNO CADA 15 SEG. Y DURANTE 2 HRS., TIEMPO DE DURACIÓN DEL PROCESO.

DATOS

- **Tº ENTRADA ANALÓGICA: CANAL 101**
- **INICIO DATOS: DM 0001**

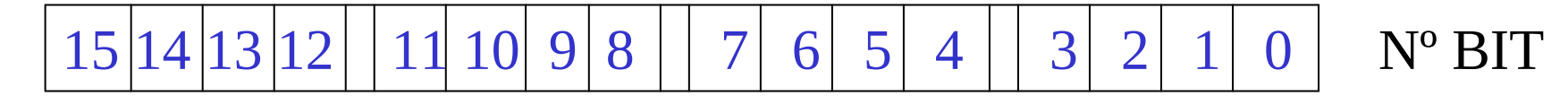


TRATAMIENTO NUMERICO

OMRON®

CONCEPTO DE REGISTRO(CANAL)

- DISPOSITIVO CAPAZ DE ALMACENAR UNA INFORMACION DIGITAL (1s y/o 0s)
- EN NUESTROS PLC's TODOS LOS REGISTROS SON DE 16 Bits (POSICIONES)



SISTEMAS DE NUMERACION

- LAS VARIABLES, EN GENERAL, PUEDEN EXPRESARSE O REPRESENTARSE SEGÚN DISTINTOS SISTEMAS DE NUMERACIÓN
- EL SISTEMA HABITUAL QUE SE EMPLEA DE FORMA COTIDIANA ES EL SISTEMA DIGITAL, QUE UTILIZA LOS SÍMBOLOS DEL 0 AL 9.
- HAY OTROS SISTEMAS DE NUMERACION QUE, AL TRABAJAR CON MÁQUINAS Y CON COMUNICACIONES, NOS APARECERÁN CONSTANTEMENTE
  - » BINARIO
  - » BCD (BINARIO CODIFICADO DECIMAL)
  - » HEXADECIMAL
  - » COMA FLOTANTE
  - » ASCII

SISTEMAS DE NUMERACION

- EN GENERAL,CUANDO UNA CANTIDAD (Nª ENTERO) SE REPRESENTA MEDIANTE UN SISTEMA DE NUMERACIÓN DE BASE B, QUIERE DECIR :

$$N_B = XXXX$$

$$N_B = X_N B^N + X_{N-1} B^{N-1} + \ldots\ldots + X_1 B^1 + X_0 B^0$$

CODIGO BINARIO

- CODIGO BINARIO

  - UTILIZA LOS SIMBOLOS (1 y 0) PAEA REPRESENTAR CUALQUIER VALOR
  - LA FORMULA DE CONVERSION DE UN NUMERO DECIMAL A UN NUMERO BINARIO ES LA SIGUIENTE :

$$Nº\ DECIMAL = Z_N \times 2^N + Z_{N-1} \times 2^{N-1} + ... + Z_0 \times 2^0$$

  - DONDE Z_N ES UNO DE LOS 2 SIMBOLOS (0 ó 1)

CODIGO BINARIO

- CODIGO BINARIO

  - **EJEMPLO:** LA REPRESENTACION DEL Nº12 EN BINARIO SERÁ :

$$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 12$$

$$1 \quad 1 \quad 0 \quad 0 \quad = \quad 12$$

  - **EJEMPLO :** REPRESENTAR EN BINARIO LOS Nº DECIMALES 16 Y 45.

		2^5	2^4	2^3	2^2	2^1	2^0
16	$\Rightarrow$		1	0	0	0	0
45	$\Rightarrow$	1	0	1	1	0	1

CODIGO HEXADECIMAL

- CODIGO HEX
  - » CODIGO MEDIANTE EL CUAL CADA NUMERO DEL SISTEMA DECIMAL (0..9) SE REPRESENTA EN BINARIO (0,1).
  - » LA CONVERSION DIRECTA ES LA SIGUIENTE :

HEXADECIMAL	BINARIO
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

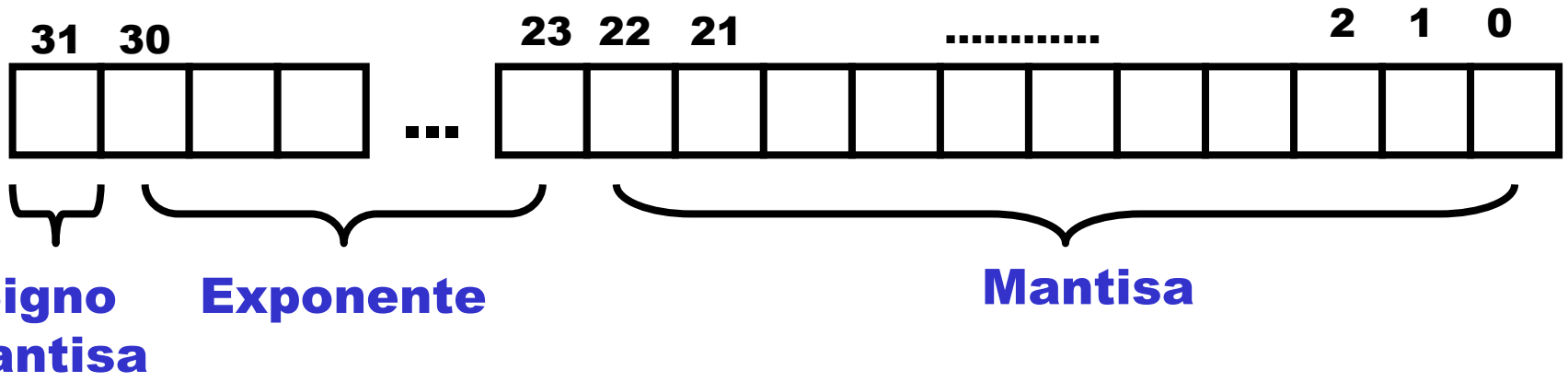
CODIGO BCD

- CODIGO BCD
  - » CODIGO MEDIANTE EL CUAL CADA NUMERO DEL SISTEMA DECIMAL (0..9) SE REPRESENTA EN BINARIO (0,1).
  - » LA CONVERSION DIRECTA ES LA SIGUIENTE :

DECIMAL	BINARIO(BCD)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Números en COMA FLOTANTE

- COMA FLOTANTE



$$Nº\ DECIMAL = (-1)^{Signo} \times 2^{e-127} (1 + Mantisa \times 2^{-23})$$

- » **Signo (s) ⇒ 1: negativo , 0: positivo (bit 31)**
- » **Mantisa (M) ⇒ La mantisa incluye 23 bits (bit 0.. 22). Representa la parte derecha de número decimal.**
- » **Exponente (e) ⇒ El exponente incluye 8 bits (bit 23..30).**

Números en COMA FLOTANTE

- Se pueden expresar los números:
  - $-\infty$ (e=255, M=0, s=0)
  - $-3.402823 \cdot 10^{38} \div -1.175494 \cdot 10^{-38}$
  - 0 (e=0)
  - $1.175494 \cdot 10^{-38} \div 3.402823 \cdot 10^{38}$
  - ∞ (e=255, M=0, s=1)
  - NaN (e=255, M≠0): Número no válido.

- **No es necesario conocer el formato de estos números, sólo que ocupan 32 bits.**

Precauciones COMA FLOTANTE

- Las operaciones indeterminadas 0.0/0.0, ∞/∞, ∞-∞ dan como resultado NaN.
- Overflow (±∞) y Underflow (±0). Es más peligroso el Overflow al convertir el resultado a entero (binario con signo).
- Los decimales se truncan al convertirlos a entero (binario con signo).
- Cualquier operación con un NaN como operando da como resultado NaN.

IEEE754

- Expresan números reales en 32 bits conforme al estándar IEEE754:

- $(-1)^{\textbf{signo}} \cdot 2^{\textbf{exponente}-127} \cdot (1+\textbf{Mantisa} \cdot 2^{-23})$

- 1#10000000#11000000000000000000000
  - Signo: $(-1)^{1} = -1$
  - Exponente: $2^{128-127} = 2^{1} = 2$
  - Mantisa: $1+6291456 \cdot 2^{-23} = 1+0.75 = 1.75$
  - Resultado: $-1.75 \cdot 2 = \textbf{-3.5}$

CODIGO ASCII

- CODIGO INTERNACIONAL CUYAS SIGLAS RESPONDEN A AMERICAN STANDAR CODE INFORMATION INTERCHANGE.
- HOY UTILIZADO EN COMUNICACIONES E INTERCAMBIO DE DATOS.
- EN ESTE CODIGO SE UTILIZAN 8 BIT's PARA LA REPRESENTACION.
- Ejemplo :

A = 41 = 0100 0001

5 = 35 = 0011 0101

> = 3E = 0011 1110

Tipos de variables en CX-P

BOOL	Variable de un bit, los posibles estados son 0-OFF y 1-ON.
UINT	Variable de una palabra en binario sin signo.
INT	Variable de una palabra en binario con signo.
UINT_BCD	Variable de una palabra en formato BCD (4 dígitos).
UDINT	Variable de dos palabras en binario sin signo.
DINT	Variable de dos palabras en binario con signo..
UDINT_BCD	Variable de dos palabras en formato BCD (8 dígitos).
ULINT	Variable de cuatro palabras en binario sin signo.
LINT	Variable de cuatro palabras en binario con signo.
ULINT_BCD	Variable de cuatro palabras en formato BCD (16 dígitos).

Tipos de variables en CX-P

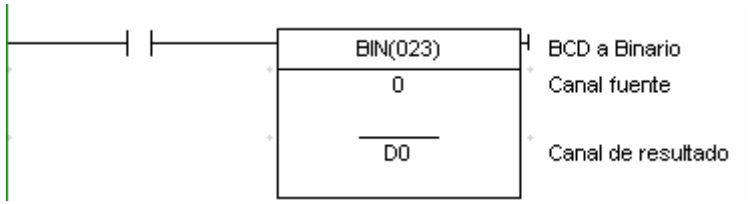
REAL	Variable de 2 palabras (32Bit) con formato en coma flotante (formato IEEE). Este formato se utiliza para las operaciones en coma flotante del del CVM1-V2 y del CS1.
NUMBER	Constante numérica en formato decimal. El valor puede ser con signo o en coma flotante. No se trata de una variable, sino de un valor numérico a utilizar por la función.
CHANNEL	Variable de una palabra. Se utiliza para compatibilizar con anteriores programas y hace referencia a cualquier variable no booleana. CX-P no puede chequear si la variable está siendo utilizada para valores en BCD o en binario.

CONVERSION
DE FORMATO DE DATOS

OMRON®

BIN(023) - BCD a Binario

•Convierte el contenido BCD de S a su equivalente en binario y lo envía a R. Sólo cambia el contenido de R, el contenido de S permanece inalterable.

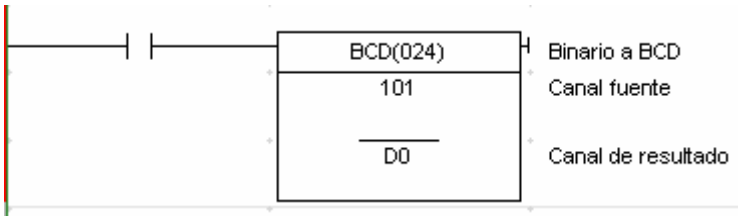


Rango:

S:Canal fuente: CIO, W, H, A, T, C, D, E, E?_, @D @E, @E?_, *D, *E, *E?_, DR, ,IR
R: Canal de resultado:CIO, W, H, A, T, C, D, E, E?_, @D @E, @E?_, *D, *E, *E?_, DR, ,IR

BCD(024) - Binario a BCD

•BCD(24) convierte el contenido binario (hexadecimal) de S a su equivalente en BCD y lo envía a R. Sólo cambia el contenido de R; el contenido de S permanece inalterable.



Rango:

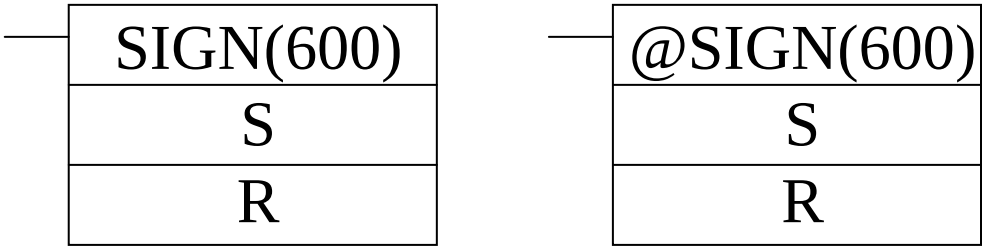
S:Canal fuente: CIO, W, H, A, T, C, D, E, E?_, @D @E, @E?_, *D, *E, *E?_, DR, ,IR
R: Canal de resultado:CIO, W, H, A, T, C, D, E, E?_, @D @E, @E?_, *D, *E, *E?_, DR, ,IR

SIGN(600)-(CS1)

- Convierte un valor de 16 bits, S, a su equivalente de 32 bits binario con signo, R.
  - S: Palabra fuente de 16 bits.
  - R: Primera palabra del resultado.

8000 ⑨ FFFF 8000

7000 ⑨ 0000 7000



BINS(470) -(CS1)

- Convierte una palabra de BCD con signo S, a binario con signo R. La palabra de control indica el formato del signo en BCD.
  - C: Palabra de control: 0, 1, 2 ó 3
  - S: Palabra en BCD.
  - R: Palabra en binario.

BINS(470)
C
S
R

@BINS(470)
C
S
R

BINS(470) - (CS1)

C=0000 (-999 a 999)

000	1	xxxx	xxxx	xxxx

El bit 12 de S indica el signo (1 negativo).
Los bits 13 a 15 deben ser 0.

C=0001 (-7999 a 7999)

1	xxx	xxxx	xxxx	xxxx

El bit 15 de S indica el signo (1 negativo).

C=0002 (-999 a 9999)

F,0-9	xxxx	xxxx	xxxx

El signo se indica en S: F negativo o 0-9 positivo
Los valores A-E dan error.

C=0003 (-1999 a 9999)

FA,0-9	xxxx	xxxx	xxxx

El signo se indica en S: F=- y A=-1 o 0-9 positivo
Los valores B-E dan error.

BISL(472) - (CS1)

- Convierte un dato de 32 bits BCD con signo (S+1 S) a binario con signo de 32 bits (R+1 R). C indica el formato de signo en BCD.
  - C: Palabra de control: 0, 1, 2 ó 3
  - S: Primera palabra de BCD.
  - R: Primera palabra en binario.

BISL(472)
C
S
R

@BISL(472)
C
S
R

BISL(472) - (CS1)

C=0000 (-999 9999 a 999 9999)

000	1	xxxx	xxxx	xxxx

El bit 12 de S+1 indica el signo (1 negativo).
Los bits 13 a 15 deben ser 0.

C=0001 (-7999 9999 a 7999 9999)

1	xxx	xxxx	xxxx	xxxx

El bit 15 de S+1 indica el signo (1 negativo).

C=0002 (-999 9999 a 9999 9999)

F,0-9	xxxx	xxxx	xxxx

El signo se indica en S+1: F negativo o 0-9 positivo
Los valores A-E dan error.

C=0003 (-1999 9999 a 9999 9999)

FA,0-9	xxxx	xxxx	xxxx

El signo se indica en S+1: F=- y A=-1 o 0-9 positivo
Los valores B-E dan error.

BCDS(471) - (CS1)

- Convierte una palabra de binario con signo a BCD con signo. C indica el formato de signo en BCD.
  - C: Palabra de control: 0, 1, 2 ó 3
  - S: Palabra en binario.
  - R: Palabra en BCD.
- Esta es la instrucción inversa a BINS(470)

BCDS(471)
C
S
R

@BCDS(471)
C
S
R

BDSL(473) - (CS1)

- Convierte un dato de binario con signo 32 bits a BCD con signo 32 bits. C indica el formato de signo en BCD.
  - C: Palabra de control: 0, 1, 2 ó 3
  - S: Palabra en binario.
  - R: Palabra en BCD.
- Esta es la instrucción inversa a BINS(470)

BDSL(473)
C
S
R

@BDSL(473)
C
S
R

Limitaciones BCDS y BDSL - (CS1)

- En BCDS el dato S está limitada según C:
  - C=0 FC19 a FFFF y 0000 a 03E7
  - C=1 F0C1 a FFFF y 0000 a 1F3F
  - C=2 FC19 a FFFF y 0000 a 270F
  - C=3 F831 a FFFF y 0000 a 270F
- En BDSL el dato S está limitada según C:
  - C=0 FF67 6981 a FFFF FFFF y 0 a 0098 967F
  - C=1 FB3B 4C01 a FFFF FFFF y 0 a 04C4 B3FF
  - C=2 FF67 6981 a FFFF FFFF y 0 a 05F5 E0FF
  - C=3 FECE D301 a FFFF FFFF y 0 a 05F5 E0FF

OPERACIONES ARITMETICAS

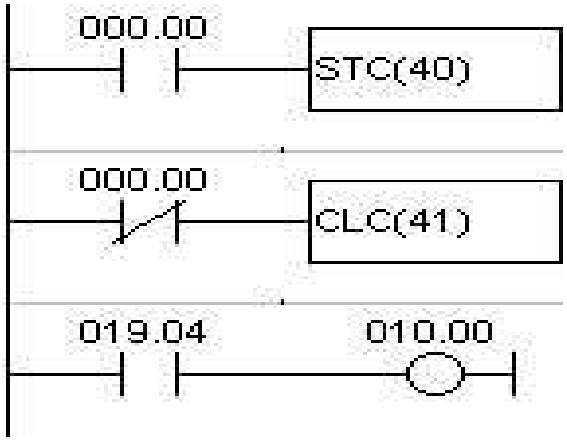
OMRON

STC / CLC, FUN 40 / 41

- **FUNCIÓN:** ESTAS INSTRUCCIONES GESTIONAN EL FLAG DE ACARREO, O FLAG **CY.**

  EL FLAG CY SE UTILIZA EN LAS OPERACIONES MATEMÁTICAS, PARA DETECTAR:

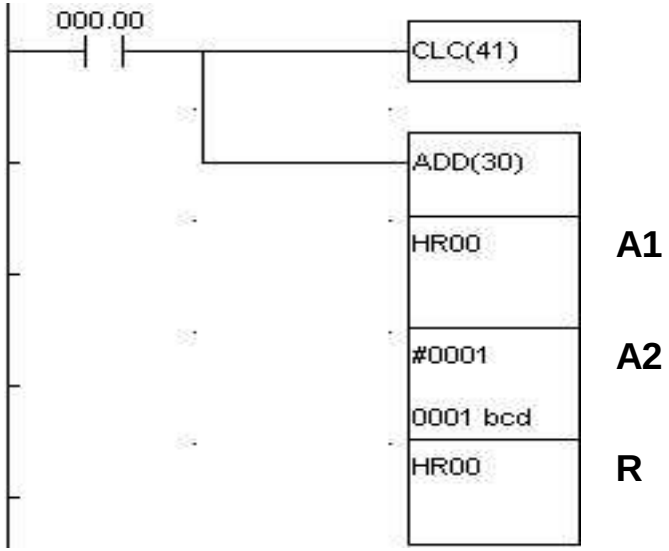
  - EXISTENCIA DE OVERFLOW EN EL RESULTADO DE UNA SUMA **(ADD)**
  - EXISTENCIA DE RESULTADO NEGATIVO EN UNA SUBSTRACCIÓN **(SUB)**




```
LD          0000
STC(40)
LD  NOT     0000
CLC(41)
LD          1904
OUT         1000
```

ADD, FUN(30) /1

- LA INSTRUCCIÓN **ADD** EJECUTA LA SUMA ENTRE DOS DATOS DE 16 BIT (CANALES Y/O CONSTANTES) EN FORMATO BCD
- AL RESULTADO SE LE SUMA EL ACARREO SUMANDO 1 SI CY= ON
- LOS PARÁMETROS DE LAS INSTRUCCIONES SON 3:
  - **A1,A2 =** SUMANDOS (#, IR, SR, HR, TIM, CNT)
  - **R =** RESULTADO (IR, HR) = A1+A2+CY




```
LD          0000
STC(40)
ADD(30)
            HR00
            #0001
            HR00
```

ADD, FUN(30) /2

- EN EL CASO DE ACARREO (CY = 1), LA SUMA DE LOS DOS SUMANDOS HA SUPERADO EL LÍMITE 9999. EL RESULTADO EFECTIVO ES ENTONCES 10000+R

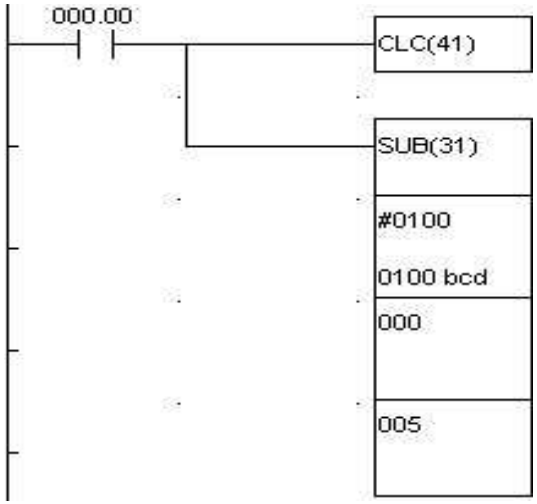
- EN EL CASO DE SUMA CON 8 O MÁS CIFRAS BCD, (SUMA COMBINADA) SE DEBERÁ RESETEAR CY SÓLO PARA LA PRIMERA SUMA.

- SI EL RESULTADO DE LA OPERACIÓN SUMA ES = 0, ENTONCES EL FLAG EQ = 1

- SI LOS SUMANDOS NO ESTÁN EN FORMATO BCD, LA OPERACIÓN NO SE EJECUTA Y ENTONCES ER = 1

SUB, FUN(31) /1

- LA INSTRUCCIÓN **SUB** EJECUTA LA SUBSTRACCIÓN DE DOS DATOS DE 16 BIT EN FORMATO BCD
- AL RESULTADO SE LE RESTA EL ACARREO O BIEN SE LE SUMA -1 SI CY = ON
- LOS PARÁMETROS DE LA INSTRUCCIÓN SON 3:
  - **Mi =** MINUENDO (#, IR, SR, HR, TIM, CNT)
  - **Su =** SUSTRAENDO (#, IR, SR, HR, TIM, CNT)
  - **R =** RESULTADO (IR, HR) = Mi - Su - CY



Mi

Su

R

LD	**0000**
STC(40)	
SUB(31)	
	#0100
	00
	05

SUB, FUN(31) /2

- SEGÚN LOS VALORES QUE TENGAN Mi Y Su, SE TIENEN LOS SIGUIENTES CASOS:

DATOS	RESULTADO	CY	EQ
Mi > Su	R= Mi - Su	0	0
Mi = Su	R= 0	0	1
Mi < Su	R= Mi + (10000-Su)	1	0

- SI Mi Y Su NO ESTÁN EN EL FORMATO BCD, LA OPERACIÓN NO SE EJECUTA, Y ER = 1

EJEMPLO DE PROGRAMACIÓN

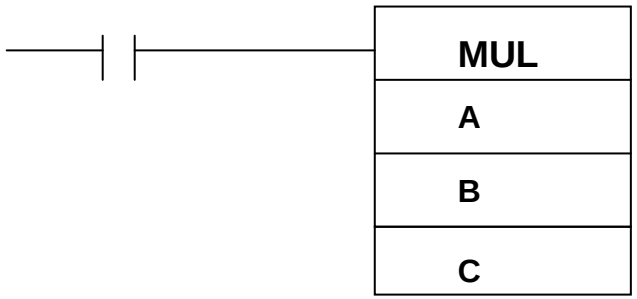
SUMA DE DATOS DE HASTA 8 DÍGITOS

- **APLICACIÓN:** SE TRATA DE SUMAR DOS DATOS QUE PUEDEN TENER UNA LONGITUD DE HASTA 8 DÍGITOS.
- ESTE PROGRAMA PUEDE LLEVARSE A CABO UTILIZANDO DIRECTAMENTE LA INSTRUCCIÓN DE SUMA DE DOBLE LONGITUD PERO LO HAREMOS USANDO LA INSTRUCCIÓN ADD NORMAL.
- LOS DATOS A SUMAR SON LOS SIGUIENTES:
  - DATO A: 4 DÍGITOS MAYORES → DM1
    4 DÍGITOS MENORES → DM0
  - DATO B: 4 DÍGITOS MAYORES → DM3
    4 DÍGITOS MENORES → DM2
- EL RESULTADO SE GUARDARÁ EN :
  - DÍGITO NUM 9 → DM6
  - 4 DÍGITOS MAYORES → DM5
  - 4 DÍGITOS MENORES → DM4

➔ El programa debe poder detectar si alguno de los datos A o B no está en formato
BCD. Utilizar el CARRY en las instrucciones suma.

MULTIPLICACIÓN BCD, FUN(32) @FUN(32)

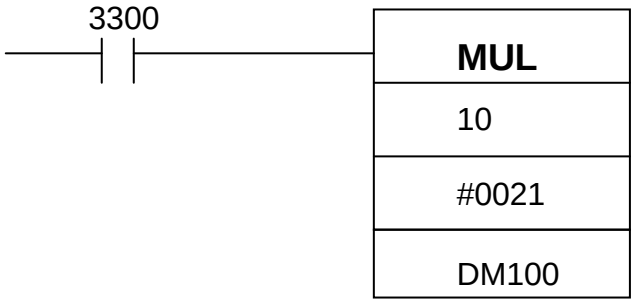
- **FUNCIÓN:** MULTIPLICA EL CONTENIDO DE LOS DATOS ESPECIFICADOS EN LA INSTRUCCIÓN (EN BCD) Y EL RESULTADO SE TRANSFIERE A UN REGISTRO.



A, B = CANALES /CONSTANTES

R = REGISTRO RESULTADO

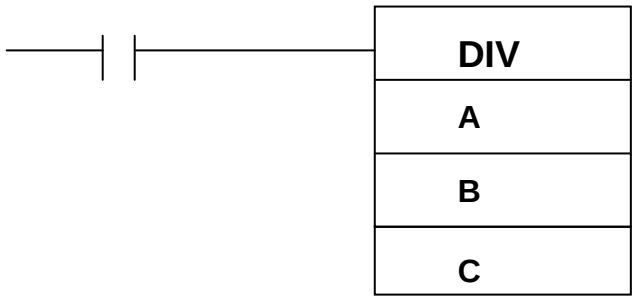
MULTIPLICACIÓN BCD, FUN(32) @FUN(32)



0034	CH 10
X	
0021	
II	
714	DM100= (CH 10) X 21

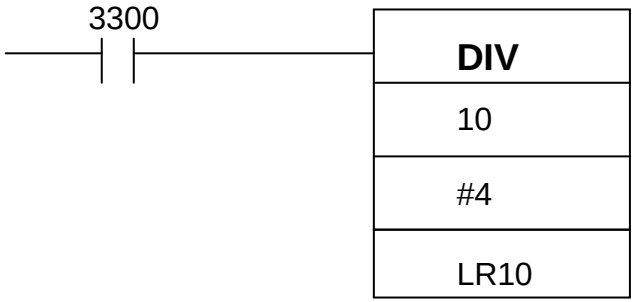
DIVISIÓN BCD, FUN(33) @FUN(33)

- **FUNCIÓN:** DIVIDE EL CONTENIDO DE LOS DATOS ESPECIFICADOS EN LA INSTRUCCIÓN (EN BCD) Y EL RESULTADO SE TRANSFIERE A DOS REGISTROS (COCIENTE Y RESTO).



A = DIVIDENDO
B = DIVISOR
R = COCIENTE
R+1 = RESTO
A, B = CANAL / CONSTANTE

DIVISIÓN BCD, FUN(33) @FUN(33)



133 CH 10

(CH 10) = (LR 10) X 4 + LR 11

4

33 LR 10

1 LR 11

OMRON

OPERACIONES ARITMETICAS (CS1)/1

+(400), +L(401), +C(402), +CL(403) - Suma Binaria

+B(404), +BL(405), +BC(406), +BCL(407) - Suma BCD

+F(454) - Suma Coma Flotante
+D(845) - Doble Suma en coma flotante

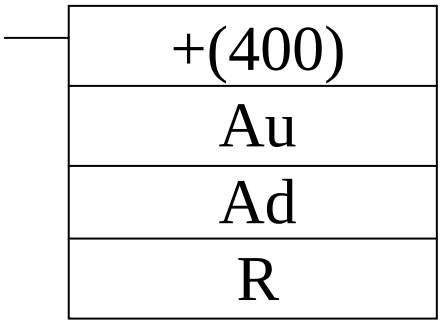
-(410), -L(411), -C(412), -CL(413) - Resta Binaria

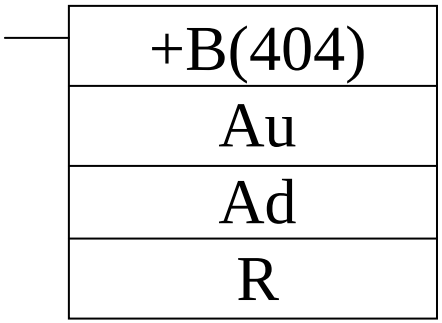
-B(414), -BL(415), -BC(416), -BCL(417) - Resta BCD

-F(455) - Resta Coma Flotante
-D(846) - Resta en coma flotante de doble precisión

+(400) y +B(404)

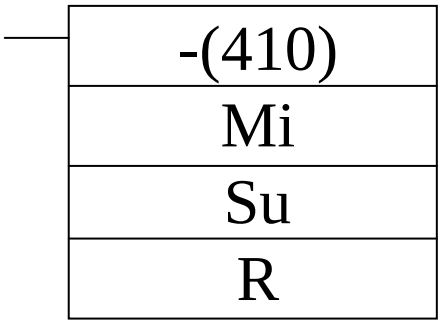
- Suma 2 números binarios o BCD de 16 bits.
  - Au: Palabra del primer sumando.
  - Ad: Plabra del segundo sumando.
  - R: Palabra del resultado.
- El rango para +(400) es de -32.768 a 32.767. Para +B(404) es de 0000 a 9999

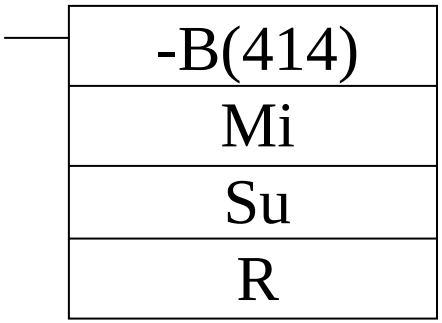




-(410) y -B(414)

- Resta 2 números binarios o BCD de 16 bits.
  - Mi: Palabra del Minuendo.
  - Su: Palabra del Sustraendo.
  - R: Palabra del resultado.
- El rango para +(400) es de -32.768 a 32.767.
  Para +B(404) es de 0000 a 9999.
- Si el resultado es negativo, el acarreo se activa y el complemento a 10 del resultado actual es puesto en R





+, - Ejemplo

D00000	0000 0000 0110 0000	96
D00001	0000 0000 0010 0000	32

+(400)
D00000
D00001
D00100

D00100	0000 0000 1000 0000	128

-(410)
D00000
D00001
D00100

D00100	0000 0000 0100 0000	64

+B, -B Ejemplo

	0	0	6	0	
D00000	0000	0000	0110	0000	60
D00001	0000	0000	0010	0000	20
	0	0	2	0	

+B(404)
D00000
D00001
D00100

	0	0	8	0	
D00100	0000	0000	1000	0000	80

-B(414)
D00000
D00001
D00100

	0	0	4	0	
D00100	0000	0000	0100	0000	40

OPERACIONES ARITMETICAS (CS1)/2

*(420), *L(421), *U(422), *UL(423) - Multiplicación Binaria

*B(424), *BL(425) - Multiplicación BCD

*F(456) - Multiplicación Coma Flotante
*D(847) - Multiplicación en coma flotante doble

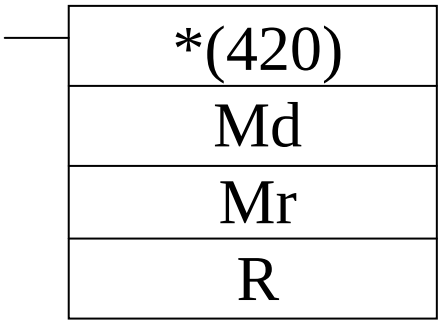
/(430), /L(431), /U(432), /UL(433) - División Binaria

/B(434), /BL(435) - División BCD

/F(457) - División Coma Flotante
/D(848) - División en coma flotante doble

*(420) y *B(424)

- Multiplica 2 números binarios o BCD de 16 bits.
  - Md: Palabra del Multiplicando.
  - Mr: Palabra del Multiplicador.
  - R: Palabra del Resultado.
- El resultado ocupa dos Palabras R y R+1



/(430) y /B(434)

- Multiplica números binarios o BCD de 16 bits.
  - Dd: Primera palabra del Dividendo.
  - Dr: Primera palabra del Divisor.
  - R: Primera palabra del Resultado.
- El resultado ocupa dos Palabras R y R+1
- Palabra R: Cociente
  Palabra R+1: Resto

*, / Ejemplo

D00000	0000 0000 0110 0000	96
D00001	0000 0000 0010 0000	32

*(420)
D00000
D00001
D00100

D00100	0000 1100 0110 0000	3072
D00101	0000 0000 0000 0000	0

/(430)
D00000
D00001
D00100

D00100	0000 0000 0000 0011	3
D00101	0000 0000 0010 0000	0

*B, /B Ejemplo

	0	0	6	0	
D00000	0000	0000	0110	0000	60
D00001	0000	0000	0010	0000	20
	0	0	2	0	

*B(424)
D00000
D00001
D00100

	1	2	0	0	
D00100	0001	0010	0000	0000	1200
D00101	0000	0000	0000	0000	0

/B(434)
D00000
D00001
D00100

	0	0	0	3	
D00100	0000	0000	0000	0011	3
D00101	0000	0000	0010	0000	0

Instrucciones en Coma Flotante

Números reales en coma flotante.

OMRON

Números en Coma Flotante

- Se pueden expresar los números:
  - $-\infty$ (e=255, f=0, s=0)
  - $-3.402823 \cdot 10^{38} \div -1.175494 \cdot 10^{-38}$
  - 0 (e=0)
  - $1.175494 \cdot 10^{-38} \div 3.402823 \cdot 10^{38}$
  - ∞ (e=255, f=0, s=1)
  - NaN (e=255, f≠0): Número no válido.
- **No es necesario conocer el formato de estos números, sólo que ocupan 32 bits.**

Precauciones Coma Flotante

- Las operaciones indeterminadas 0.0/0.0, ∞/∞, ∞-∞ dan como resultado NaN.
- Overflow (±∞) y Underflow (±0). Es más peligroso el Overflow al convertir el resultado a entero (binario con signo).
- Los decimales se truncan al convertirlos a entero (binario con signo).
- Cualquier operación con un NaN como operando da como resultado NaN.

IEEE754

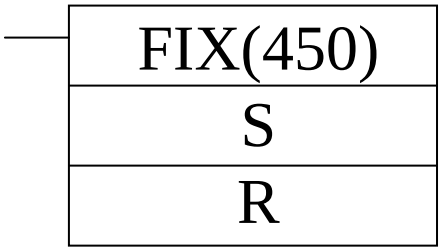
- Expresan números reales en 32 bits conforme al estándar IEEE754:
  - f: Mantisa 23 bits bit 0 al 22
  - e: Exponente 8 bits bit 23 al 30
  - s: Signo 1 bit bit 31
- $(-1)^{s} \cdot 2^{e-127} \cdot (1+f \cdot 2^{-23})$
- 1#10000000#11000000000000000000000
  - Signo: $(-1)^{1} = -1$
  - Exponente: $2^{128-127} = 2^{1} = 2$
  - Mantisa: $1+6291456 \cdot 2^{-23} = 1+0.75 = 1.75$
  - Resultado: $-1.75 \cdot 2 =$ **-3.5**

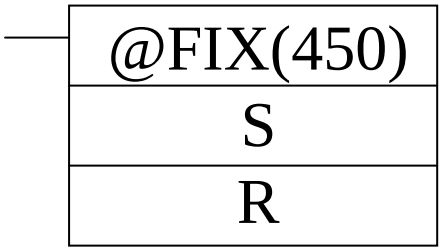
Instrucciones Coma Flotante

- Se pueden realizar las siguientes:
  - Conversión: FIX, FIXL, FLT, FLTL
  - Operaciones: +F, -F, *F, /F, SQRT, PWR
  - Conversiones angulares: RAD, DEG
  - Angulares: SIN, COS, TAN, ASIN, ACOS, ATAN
  - En base e: LOG, EXP
- **Los operandos deben ser Nºs en formato coma flotante IEEE754.**
- **No es necesario conocer este formato, sólo que ocupan 32 bits.**

FIX(450)

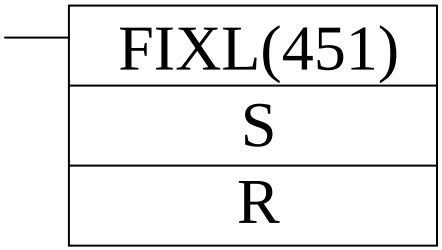
- Convierte un número en coma flotante a un entero (binario con signo) de 16 bits.
  - S: Primera palabra de número en coma flotante.
  - R: Palabra donde se guarda el resultado.
- La parte decimal es truncada (3.5 ≙ 3)
- Rango de -32.768 ÷ 32.767.

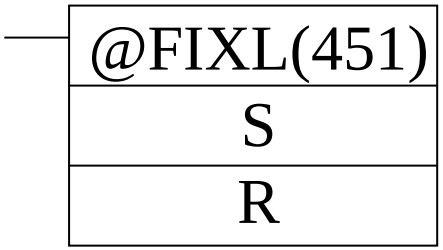




FIXL(451)

- Convierte un número en coma flotante a un entero (binario con signo) de 32 bits.
  - S: Primera palabra de número en coma flotante.
  - R: Primera palabra donde se guarda el resultado.
- La parte decimal es truncada (-3.5 ≙ -3)
- Rango de -2.147.482.648÷2.147.482.647.





FIX, FIXL Ejemplo

FIX(450)
D00000
D00100

D00000	0000000000000000
D00001	0100000001100000
D00100	0000000000000011

3.5

3

FIXL(451)
D00002
D00102

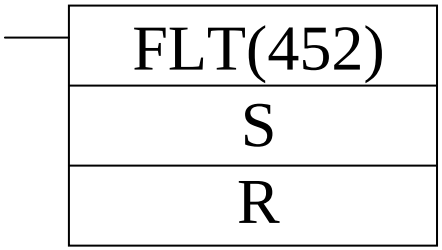
D00002	0000000000000000
D00003	1100110001100000
D00102	0000000000000000
D00103	1111110010000000

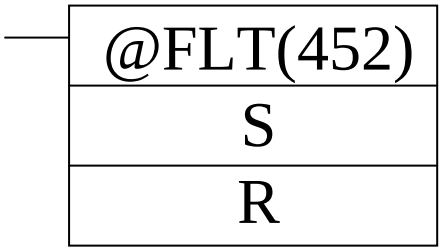
-58.720.256

-58.720.256

FLT(452)

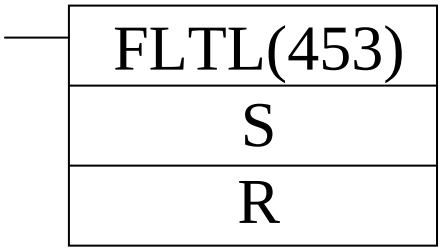
- Convierte un número entero (binario con signo) de 16 bits en formato de coma flotante de 32 bits.
  - S: Palabra del número entero.
  - R: Primera palabra del resultado.
- Rango de -32.768 ÷ 32.767.

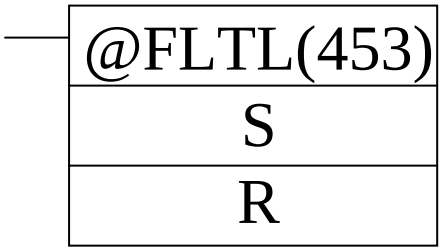




FLTL(453)

- Convierte un número entero (binario con signo) de 32 bits en formato de coma flotante de 32 bits.
  - S: Primera palabra del número entero.
  - R: Primera palabra del resultado.
- Rango de -2.147.482.648÷2.147.482.647.
- Números > 16.777.215 pierden precisión.





FLT, FLTL Ejemplo

FLT(452)
D00100
D00000

D00000	0000000000000000
D00001	0100000001000000
D00100	0000000000000011

3

3

FLTL(453)
D00102
D00002

D00002	0000000000000000
D00003	1100110001100000
D00102	0000000000000000
D00103	1111110010000000

-58.720.256

-58.720.256

OMRON®

+F(454)

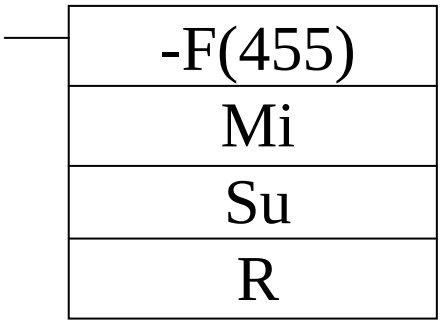
- Suma 2 números en coma flotante de 32 bits.
  - Au: Primera palabra del primer sumando.
  - Ad: Primera palabra del segundo sumando.
  - R: Primera palabra del resultado.
- El resultado puede ser ∞, -∞, 0, NaN.
- Tener en cuenta: ∞ - ∞ = NaN y que NaN+número=NaN.

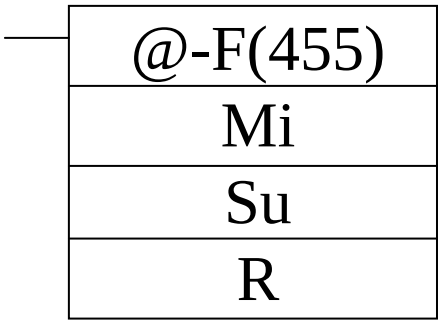
+F(454)
Au
Ad
R

@+F(454)
Au
Ad
R

-F(455)

- Resta 2 números en coma flotante de 32 bits.
  - Mi: Primera palabra del Minuendo.
  - Su: Primera palabra del Sustraendo.
  - R: Primera palabra del resultado.
- El resultado puede ser ∞, -∞, 0, NaN.
- Tener en cuenta: ∞ - ∞ = NaN y que NaN-número=NaN, número-NaN=NaN.





+F, -F Ejemplo

D00000	0000 0000 0000 0000
D00001	0100 0000 0110 0000
D00002	0000 0000 0000 0000
D00003	0100 0000 1100 0000

3.5

6

+F(454)
D00000
D00002
D00100

D00100	0000 0000 0000 0000
D00101	0100 0001 1110 0000

3.5

+6

9.5

-F(455)
D00000
D00002
D00100

D00100	0000 0000 0000 0000
D00101	1100 0000 0010 0000

3.5

- 6

-2.5

*F(456)

- Multiplica 2 números en coma flotante de 32 bits.
  - Md: Primera palabra del Multiplicando.
  - Mr: Primera palabra del Multiplicador.
  - R: Primera palabra del Resultado.
- El resultado puede ser ∞, -∞, 0, NaN.
- Tener en cuenta: 0·∞ = NaN y que NaN·número=NaN.

/F(457)

- Multiplica 2 números en coma flotante de 32 bits.
  - Dd: Primera palabra del Dividendo.
  - Dr: Primera palabra del Divisor.
  - R: Primera palabra del Resultado.
- El resultado puede ser ∞, -∞, 0, NaN.
- Tener en cuenta: 0/0 = NaN, ∞/∞ = NaN y que NaN/número=NaN, número/NaN=NaN.

/F(457)
Dd
Dr
R

@/F(457)
Dd
Dr
R

*F, /F Ejemplo

D00000	0000 0000 0000 0000
D00001	0100 0000 0110 0000
D00002	0000 0000 0000 0000
D00003	0100 0000 1100 0000

*F(456)
D00000
D00002
D00100

D00100	0000 0000 0000 0000
D00101	0100 0001 1011 0000

```
  3.5
* 6
 21
```

/F(457)
D00000
D00002
D00100

D00100	0101 0101 0101 0101
D00101	0011 1111 0101 1010

```
  3.5
÷ 6
 0.583333
```