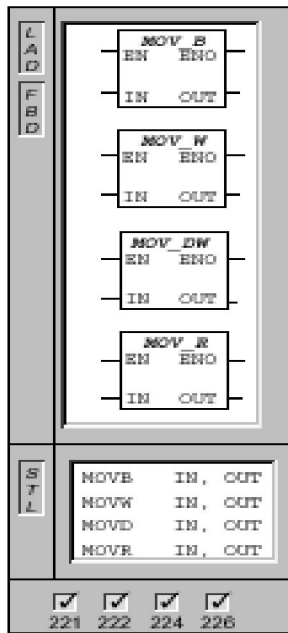


Chương 3: CÁC PHÉP TOÁN SỐ CỦA PLC**BÀI 3.1: CHỨC NĂNG TRUYỀN DẪN****1. Truyền Byte, Word, Doubleword**

Move Byte, Move Word, Move Double Word, Move Real

Chỉ dẫn Move Byte chuyển byte đầu vào (IN) tới byte đầu ra (OUT). Byte đầu vào không được biến đổi bởi việc chuyển này.

Chỉ dẫn Move Word chuyển từ đầu vào (IN) tới từ đầu ra (OUT). Từ đầu vào không được biến đổi bởi việc chuyển này.

Chỉ dẫn Move Double Word chuyển từ kép đầu vào (IN) tới từ kép đầu ra (OUT). Từ kép đầu vào không được biến đổi bởi việc chuyển này.

Chỉ dẫn Move Real chuyển một từ kép đầu vào thực 32 bit (IN) tới từ kép đầu ra (OUT). Từ kép đầu vào không được biến đổi bởi việc chuyển này.

Điều kiện lỗi khi đặt ENO = 0; SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

Move	Các đầu vào/ra	Các toán hạng	Các kiểu dữ liệu
Byte	IN	VB, IB, QB, MB, SB, SMB, LB, AC, Hằng số, *VD, *AC, *LD	BYTE
	OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE
Từ	IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, Hằng số, AC *VD, *AC, *LD	WORD, INT
	OUT	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD	WORD, INT
Từ kép	IN	VD, ID, QD, MD, SD, SMD, LD, HC, &VB, &IB, &QB, &MB, &SB, &T, &C, AC, Hằng số, *VD, *AC, *LD	DWORD, DINT
	OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	DWORD, DINT
Số thực	IN	VD, ID, QD, MD, SD, SMD, LD, AC, Hằng số, *VD, *AC, *LD	REAL

	OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	REAL
--	-----	---	------

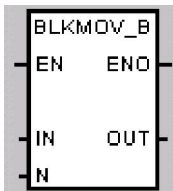
2. Truyền một vùng nhớ dữ liệu (mảng dữ liệu)

Các lệnh làm việc với mảng dùng để điều hành một mảng dữ liệu lớn

a. BLKMOV_B (LAD) - BMB (STL)

Lệnh di chuyển nội dung của một mảng byte. Độ dài của mảng được xác định bởi N có kiểu byte, do vậy lớn nhất mảng chỉ có thể có 255 byte. Byte đầu của mảng là IN (kiểu byte) nơi đến được xác định bằng byte đầu là OUT.

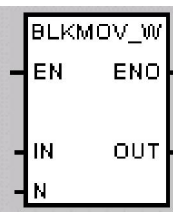
Cú pháp dùng lệnh di chuyển BMB trong STL hay BLKMOV_B trong LAD như sau:

LAD	STL	Toán hạng
	BMB IN, OUT, N	IN, OUT: (byte) VB, IB, MB, SMB, *VD, *AD N (byte): VB, IB, MB, SMB, *VD, *AD

b. BLKMOV_W (LAD) - BMW (STL)

Lệnh di chuyển nội dung của một mảng từ đơn. Độ dài của mảng được xác định bởi N có kiểu byte, do vậy lớn nhất mảng chỉ có thể có 255 byte. Từ đơn đầu của mảng là IN (kiểu byte) nơi đến được xác định bằng byte đầu là OUT.

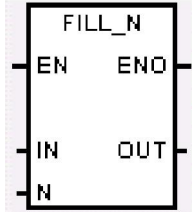
Cú pháp dùng lệnh di chuyển BMW trong STL hay BLKMOV_W trong LAD như sau:

LAD	STL	Toán hạng
	BMW IN OUT, N	IN, OUT (word): VW, T, C, IW, QW, MW, SMW, *VD, *AD N (byte): VB, IB, MB, SMB, *VD, *AD, hằng số

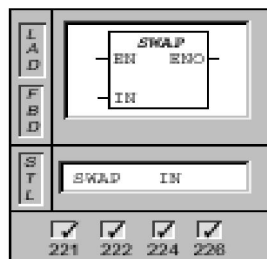
c. FILL_N (LAD) - FILL (STL)

Lệnh ghi vào một mảng từ đơn giá trị của từ IN. Độ dài của mảng được xác định bởi N có kiểu byte, do vậy lớn nhất mảng chỉ có thể có 255 từ. Từ đơn đầu của mảng là OUT (kiểu từ).

Cú pháp dùng lệnh di chuyển FIIL_N trong LAD hay FILL trong STL như sau:

LAD	STL	Toán hạng
	FILL IN OUT N	IN, OUT (word) VW,T, C, IW, QW, MW, SMW, *VD, *AD N (byte): VB, IB, MB, SMB, *VD, *AD, hằng số

d. Swap Byte

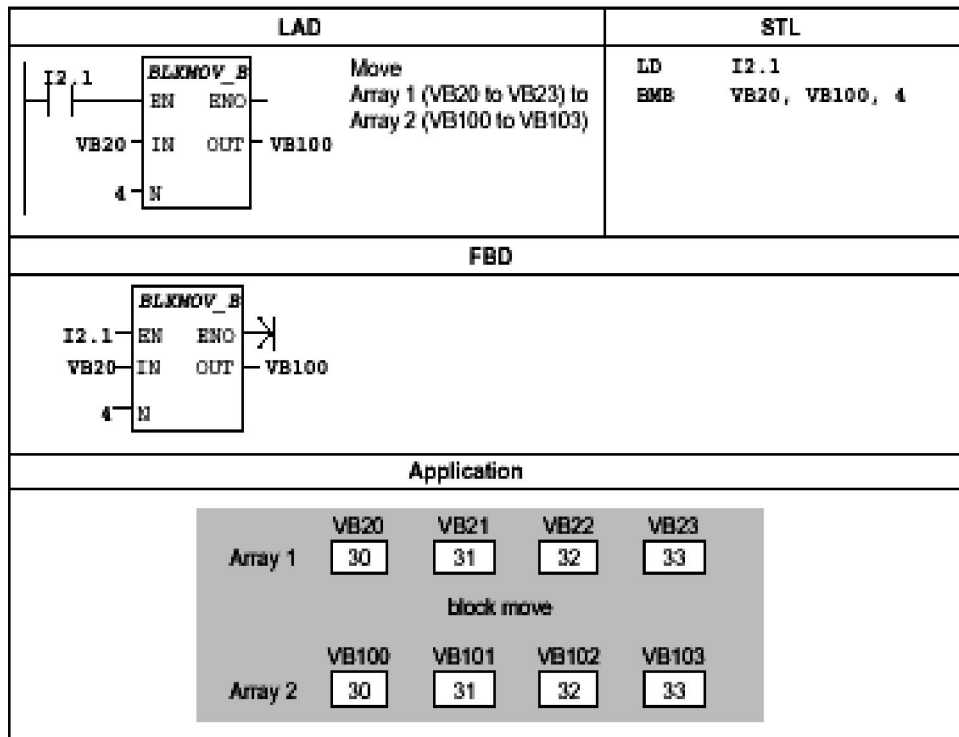


Chỉ dẫn Swap Byte trao đổi byte có trọng số lớn nhất với byte có trọng số nhỏ nhất của từ (IN).
Điều kiện lỗi khi đặt ENO = 0; SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

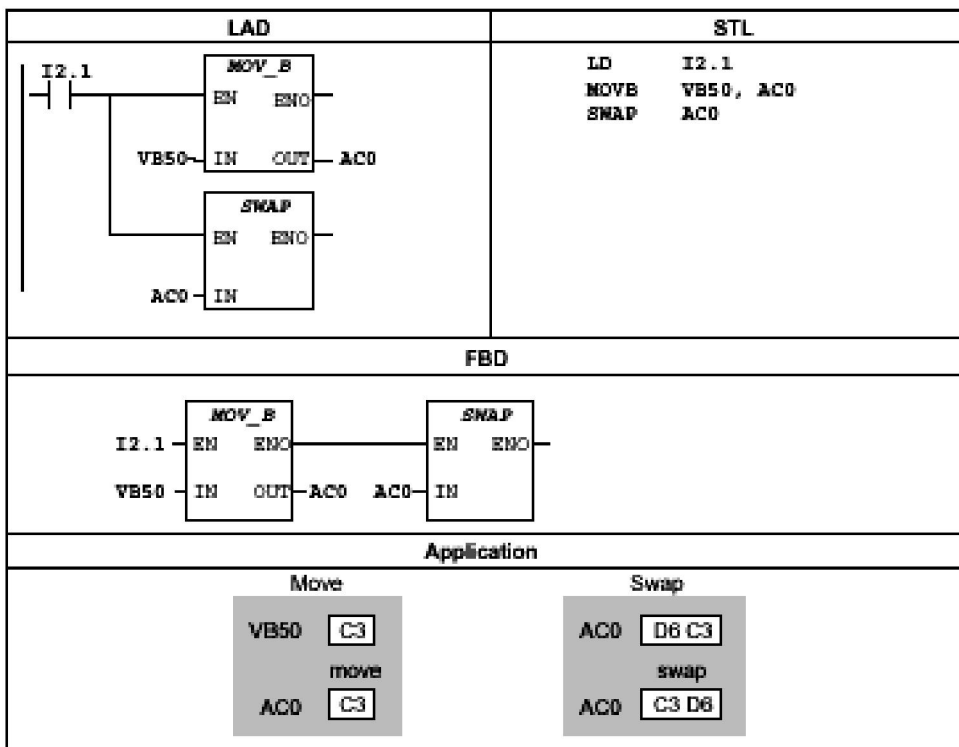
Các đầu vào/ra	Các toán hạng	Các kiểu dữ liệu
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	WORD

3. Bài tập

a. Lệnh di chuyển nội dung mảng

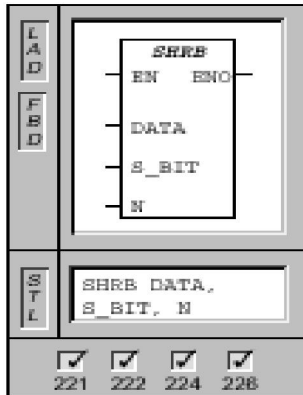


b. Ví dụ về Move và Swap



BÀI 3.2: CHỨC NĂNG SO SÁNH

I. Chức năng dịch chuyển (Thanh ghi dịch)



Chỉ dẫn Shift Register (SHRB) dịch giá trị của DATA tới thanh ghi dịch. S_BIT xác định bit trọng nhỏ nhất của thanh ghi dịch. N xác định độ dài của thanh ghi dịch và điều khiển dịch (Dịch cộng = N, dịch trừ = -N).

Mỗi khi bit được dịch ngoài bởi chỉ dẫn SHRB là được đặt trong bit nhớ tràn (SM1.1).

Điều kiện lỗi khi đặt ENO = 0; SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp), 0091 (toán hạng ngoài phạm vi), 0092 (lỗi trong trường đếm).

Chỉ dẫn này ảnh hưởng bit nhớ đặc biệt như sau: SM1.1 (tràn).

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
DATA, S_BIT	I, Q, M, SM, T, C, V, S, L	BOOL
N	VB, IB, QB, MB, SMB, LB, AC, Hằng số *VD, *AC, SB, *LD	BYTE

Những hiểu biết về thanh ghi dịch Bit

Chỉ dẫn thanh ghi dịch bit cung cấp một cách dễ dàng cho sắp xếp và điều khiển cung cấp sản phẩm hoặc dữ liệu. Sử dụng chỉ dẫn thanh ghi dịch bit là được định nghĩa bởi cả bit dấu nhỏ nhất (S_BIT) và số của các bit được xác định bởi độ dài (N). Hình 9-41 minh họa một ví dụ của chỉ dẫn thanh ghi dịch bit.

Địa chỉ của bit dấu trọng số lớn nhất thanh ghi dịch (MSB.b) có thể được tính bởi công thức sau:

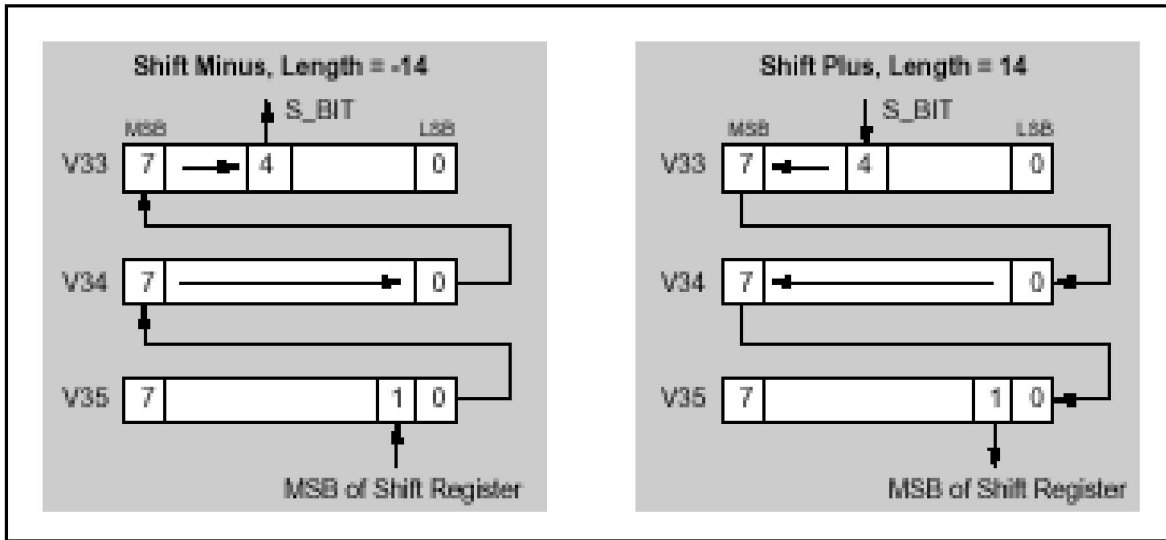
$MSB.b = [(Byte \text{ của } S_BIT) + ([N] - 1 + (bit \text{ của } S_BIT))/8]$ [duy trì việc chia cho 8]

Bạn phải trừ 1 bit bởi vì S_BIT là một của các bit của thanh ghi dịch.

Cho ví dụ, nếu S_BIT là V33.4 và N là 14, khi đó MSB.b là V35.1, hoặc:

$$\begin{aligned}
 MSB.b &= V33 + ([14] - 1 + 4)/8 \\
 &= V33 + 17/8 \\
 &= V33 + 2 \text{ với một sự duy trì của } 1 \\
 &= V35.1
 \end{aligned}$$

Trên một dịch trừ, được chỉ thị bởi một giá trị âm của độ dài (N), dữ liệu đầu vào dịch tới bit trọng số lớn nhất của thanh ghi dịch, và dịch ngoài của bit trọng số nhỏ nhất (S_BIT).

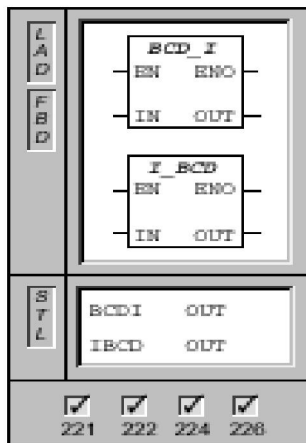


Trên một dịch cộng, được chỉ thị bởi một giá trị dương của độ dài (N), dữ liệu đầu vào (DATA) dịch tới bit trọng số nhỏ nhất của thanh ghi dịch, được xác định bởi S_BIT và nằm ngoài của bit trọng số lớn nhất của thanh ghi dịch.

Dữ liệu được dịch ngoài khi đó được đặt trong bit nhớ tràn (SM1.1). Độ dài max của thanh ghi dịch là 64 bit, dương hoặc âm. Hình 9-40 minh họa bit dịch đối với các giá trị âm hoặc dương của N.

II. Chức năng chuyển đổi

1. BCD to Integer, Integer to BCD



Chỉ dẫn BCD to Integer chuyển đổi đầu vào thập phân mã nhị phân (IN) tới một giá trị số nguyên và tải kết quả tới biến được xác định bởi OUT. Phạm vi hiệu lực cho IN là từ 0 tới 9999 BCD.

Chỉ dẫn Integer to BCD chuyển đổi đầu vào giá trị số nguyên (IN) tới một thập phân mã nhị phân và tải kết quả tới biến được xác định bởi OUT. Phạm vi hiệu lực cho IN là từ 0 tới 9999 số nguyên.

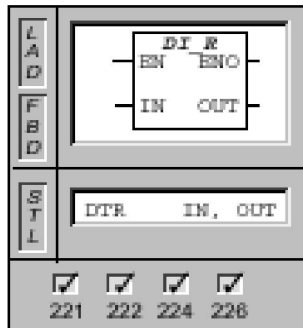
Điều kiện lỗi khi đặt ENO = 0; SM1.6 (Lỗi BCD), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp). Các chỉ dẫn ảnh hưởng các bit nhớ đặc biệt như sau: SM1.6 (vô

hiệu hoá BCD).

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN	VW, T, C, IW, QW, MW, SMW, LW, AC, AIW, hằng số, *VD, *AC, *LD	WORD

OUT	VW, T, C, IW, QW, MW, SMW, LW, AC, hằng số, *VD, *AC, *LD	WORD
-----	---	------

2. Double Integer to Real

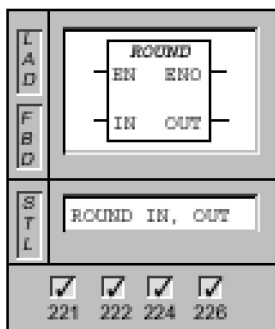


Chỉ dẫn Double Integer to Real chuyển đổi giá trị 32 bit, số nguyên có dấu (IN) tới một số thực 32 bit và đặt kết quả tới biến được xác định bởi OUT.

Điều kiện lỗi khi đặt ENO = 0; SM1.6 (Lỗi BCD), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN	VD, ID, QD, MD, SMD, AC, LD, HC, hằng số, *VD, *AC, SD, *LD	DINT
OUT	VD, ID, QD, MD, SMD, LD, AC, *VD, *AC, SD, *LD	REAL

3. Round



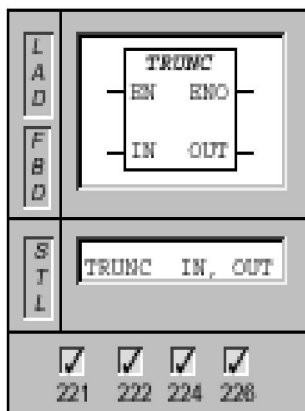
Chỉ dẫn Round chuyển đổi giá trị thực (IN) tới một giá trị số nguyên kép và đặt kết quả tới biến được xác định bởi OUT. Nếu phần thập phân là 0.5 hoặc lớn hơn, số là được làm tròn lên.

Điều kiện lỗi khi đặt ENO = 0; SM1.1 (tràn), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

Các chỉ dẫn ảnh hưởng tới các bit nhớ đặc biệt như sau: SM1.1 (tràn).

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN	VD, ID, QD, MD, SMD, AC, LD, hằng số, *VD, *AC, SD, *LD	REAL
OUT	VD, ID, QD, MD, SMD, LD, AC, *VD, *AC, SD, *LD	DINT

4. Truncate.



Chỉ dẫn Truncate chuyển đổi một số thực 32 bit (IN) tới một giá trị số nguyên có dấu 32 bit và đặt kết quả tới biến được xác định bởi OUT. Chỉ phần số nguyên của số thực là được chuyển đổi, và phần thập phân là được loại bỏ.

Nếu giá trị bạn đang chuyển đổi không là một số thực có hiệu lực hoặc có phạm vi rộng được đáp ứng ở

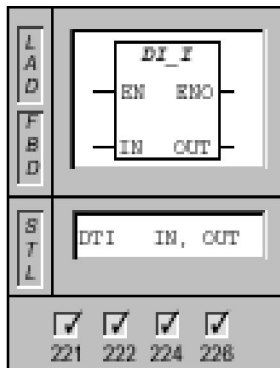
đầu ra, khi đó bit tràn là thiết lập và đầu ra không bị ảnh hưởng.

Điều kiện lỗi khi đặt ENO = 0; SM1.1 (tràn), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

Các chỉ dẫn ảnh hưởng tới các bit nhớ đặc biệt như sau: SM1.1 (tràn).

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN	VD, ID, QD, MD, SMD, LD, AC, hằng số, *VD, *AC, SD, *LD	REAL
OUT	VD, ID, QD, MD, SMD, LD, AC, *VD, *AC, SD, *LD	DINT

5. Double Integer to Integer



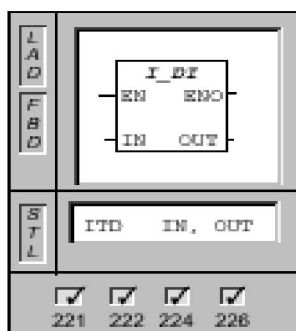
Chỉ dẫn Double Integer to Integer chuyển đổi giá trị nguyên kép (IN) tới một giá trị số nguyên và đặt kết quả tới biến được xác định bởi OUT. Nếu giá trị bạn đang chuyển đổi có phạm vi rộng được đáp ứng ở đầu ra, khi đó bit tràn là thiết lập và đầu ra không bị ảnh hưởng.

Điều kiện lỗi khi đặt ENO = 0; SM1.1 (tràn), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

Các chỉ dẫn ảnh hưởng tới các bit nhớ đặc biệt như sau: SM1.1 (tràn).

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN	VD, ID, QD, MD, SMD, AC, LD, HC, hằng số, *VD, *AC, SD, *LD	DINT
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	INT

6. Integer to Double Integer



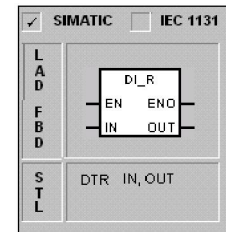
Chỉ dẫn Integer to Double Integer chuyển đổi giá trị nguyên (IN) tới một giá trị số nguyên kép và đặt kết quả tới biến được xác định bởi OUT. Dấu là được mở rộng.

Điều kiện lỗi khi đặt ENO = 0; SM1.1 (tràn), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

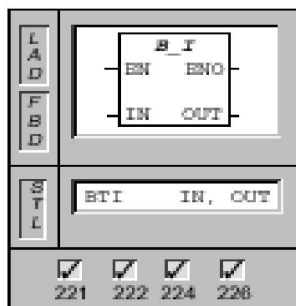
Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, Hằng số , *AC *VD, *LD	INT
OUT	VD, ID, QD, MD, SMD, LD, AC, *VD, *AC, *LD	DINT

7. Integer to Real

Để chuyển đổi một số nguyên 32 bit tới một số thực, sử dụng chỉ dẫn Integer to Double Integer



8. Byte to Integer

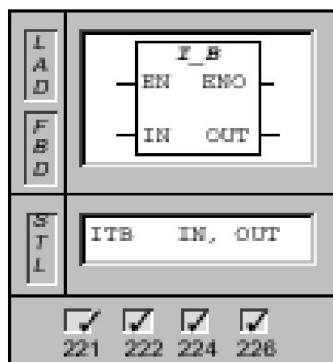


Chỉ dẫn Byte to Integer chuyển đổi giá trị byte (IN) tới một giá trị số nguyên và đặt kết quả tới biến được xác định bởi OUT. Byte là không dấu, thật vậy dấu là không được mở rộng.

Điều kiện lỗi khi đặt ENO = 0; SM1.1 (tràn), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN	VB, IB, QB, MB, SB, SMB, LB, AC, Hằng số, *AC *VD, *LD	BYTE
OUT	VW, IW, QW, MW, SMW, LW, T, C, AC, *VD, *AC, *LD	INT

9. Integer to Byte



Chỉ dẫn Integer to Byte chuyển đổi giá trị từ (IN) tới một giá trị byte và đặt kết quả tới biến được xác định bởi OUT.

Giá trị 0 tới 255 là được chuyển đổi. Kết quả tất cả các giá trị khác đặt ở tràn và đầu ra là không bị ảnh hưởng.

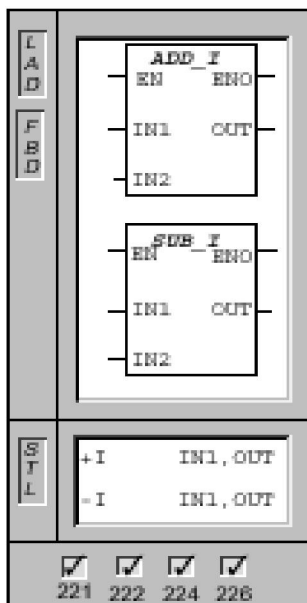
Điều kiện lỗi khi đặt ENO = 0; SM1.1 (tràn), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

Các chỉ dẫn ảnh hưởng tới các bit nhớ đặc biệt như sau: SM1.1 (trần)

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN	VW, IW, QW, MW, SMW, LW, T, C, AIW, AC, Hằng số, *VD, *AC, *LD	INT
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *AC *VD, *LD	BYTE

III. Chức năng toán học

1. Số nguyên SIMATIC



a. Add Integer và Subtract Integer

Chỉ dẫn Add Integer và Subtract Integer cộng hoặc trừ hai số nguyên 16 bit và cung cấp một kết quả 16 bit (Ra).

Trong LAD và FBD: $IN1 + IN2 = OUT$

$IN1 - IN2 = OUT$

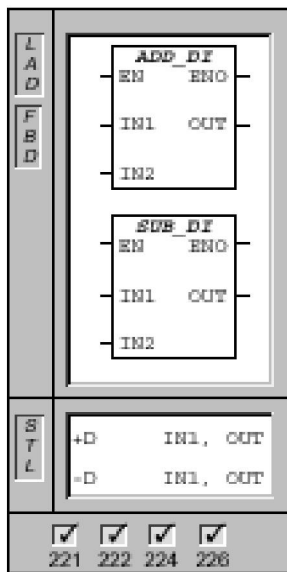
Trong STL: $IN1 + OUT = OUT$

$OUT - IN1 = OUT$

Các điều kiện lỗi khi đặt ENO = 0: SM1.1 (Trần), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN1, IN2	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, hằng số, *VD, *AC, *LD	INT
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	INT

b. Add Double Integer và Subtract Double Integer



Chỉ dẫn Add Double Integer và Subtract Double Integer cộng hoặc trừ hai số nguyên 32 bit và cung cấp một kết quả 32 bit (Ra).

Trong LAD và FBD: $IN1 + IN2 = OUT$

$IN1 - IN2 = OUT$

Trong STL: $IN1 + OUT = OUT$

$OUT - IN1 = OUT$

Các điều kiện lỗi khi đặt ENO = 0: SM1.1 (Tràn), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

Các chỉ dẫn ảnh hưởng các bit bộ nhớ đặc biệt như sau: SM1.0 (zero); SM1.1 (tràn); SM1.2 (âm).

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN1, IN2	VD, ID, QD, MD, SD, SMD, LD, AC, HC, hằng số, *VD, *AC, *LD	DINT
OUT	VD, ID, QD, MD, SMW, SD, LD, AC, *VD, *AC, *LD	DINT

c. Multiply Integer và Divide Integer

Chỉ dẫn Multiply Integer nhân hai số nguyên 16 bit và cung cấp một sản phẩm 16 bit.

Chỉ dẫn Divide Integer chia hai số nguyên 16 bit và cung cấp một thương số 16 bit. Phần còn lại không giữ.

Bit tràn là thiết lập nếu kết quả là lớn hơn một đầu ra từ.

Trong LAD và FBD: $IN1 * IN2 = OUT$

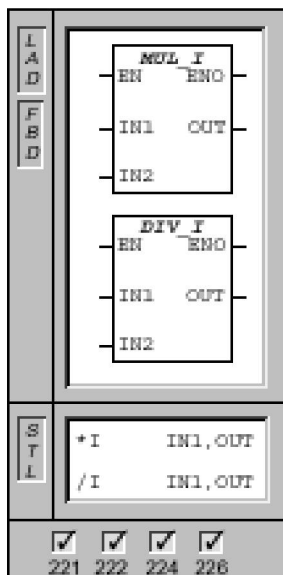
$IN1 / IN2 = OUT$

Trong STL: $IN1 * OUT = OUT$

$OUT / IN1 = OUT$

Các điều kiện lỗi khi đặt ENO = 0: SM1.1 (Tràn), SM1.3 (chia cho 0), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

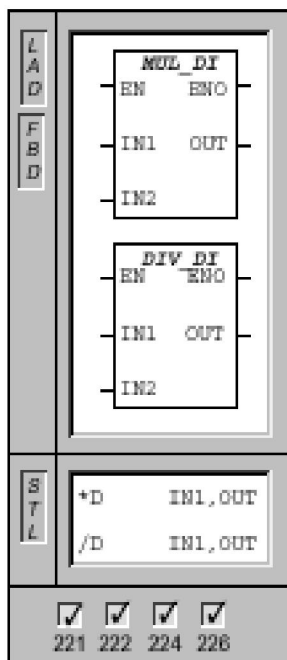
Các chỉ dẫn ảnh hưởng các bit bộ nhớ đặc biệt như sau: SM1.0 (zero); SM1.1 (tràn); SM1.2 (âm); SM1.3 (chia cho 0).



Nếu SM1.1 (tràn) là thiết lập trong khi hoạt động nhân hoặc chia, khi đó các bit trạng thái phép toán khác là không được thay đổi trái và các toán hạng đầu vào ban đầu là không biến đổi. Nếu không thì, tất cả các bit trạng thái phép toán được hỗ trợ bao gồm trạng thái có hiệu lực trên sự hoàn thiện của hoạt động phép toán.

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN1, IN2	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, hằng số, *VD, *AC, *LD	INT
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	INT

d. Multiply Double Integer và Divide Double Integer



Chỉ dẫn Multiply Double Integer nhân hai số nguyên 32 bit và cung cấp một sản phẩm 32 bit.

Chỉ dẫn Divide Double Integer chia hai số nguyên 32 bit và cung cấp một thương số 32 bit. Phần còn lại không giữ.

Bit tràn là thiết lập nếu kết quả là lớn hơn một đầu ra từ.

Trong LAD và FBD: $IN1 * IN2 = OUT$
 $IN1 / IN2 = OUT$

Trong STL: $IN1 * OUT = OUT$
 $OUT / IN1 = OUT$

Các điều kiện lỗi khi đặt ENO = 0: SM1.1 (Tràn), SM1.3 (chia cho 0), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

Các chỉ dẫn ảnh hưởng các bit bộ nhớ đặc biệt như sau: SM1.0 (zero); SM1.1 (tràn); SM1.2 (âm); SM1.3 (chia cho 0).

Nếu SM1.1 (tràn) là thiết lập trong khi hoạt động nhân hoặc chia, đầu ra là không ghi và tất cả các bit trạng thái phép toán khác là đặt tới zero.

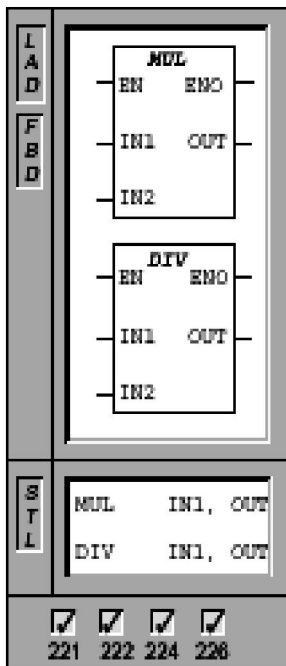
Nếu SM1.3 (chia cho 0) là thiết lập trong khi hoạt động chia, khi đó các bit trạng thái phép toán khác là không được thay đổi trái và các toán hạng đầu vào ban đầu là không biến đổi. Nếu không thì, tất cả các bit trạng thái phép toán được hỗ trợ bao gồm trạng thái có hiệu lực trên sự hoàn thiện của hoạt động phép toán.

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN1, IN2	VD, ID, QD, MD, SMD, SD, LD, AC, HC, hằng số, *VD, *AC, *LD	DINT
OUT	VD, ID, QD, MD, SMW, SD, LD, AC, *VD, *AC, *LD	DINT

e. Multiply Integer To Double Integer và Divide Integer To Double Integer

Chỉ dẫn Multiply Integer to Double Integer nhân hai số nguyên 16 bit và cung cấp một sản phẩm 32 bit.

Chỉ dẫn Divide Integer to Double Integer chia hai số nguyên 32 bit và cung cấp một kết quả 32 bit bao gồm một phần còn lại 16 bit (có nghĩa lớn nhất) và một thương số 16 bit (có nghĩa nhỏ nhất).



Trong chỉ dẫn nhân STL, từ có nghĩa nhỏ nhất (16 bit) của OUT 32 bit là được sử dụng như số bị chia.

Trong LAD và FBD: $IN1 * IN2 = OUT$

$IN1 / IN2 = OUT$

Trong STL: $IN1 * OUT = OUT$

$OUT / IN1 = OUT$

Các điều kiện lỗi khi đặt ENO = 0: SM1.1 (Tràn), SM1.3 (chia cho 0), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

Các chỉ dẫn ảnh hưởng các bit bộ nhớ đặc biệt như sau: SM1.1 (tràn); SM1.3 (chia cho 0), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp).

Các chỉ dẫn ảnh hưởng tới các bit bộ nhớ đặc biệt như sau: SM1.0 (zero); SM1.1 (tràn); SM1.2 (âm); SM1.3 (chia cho 0).

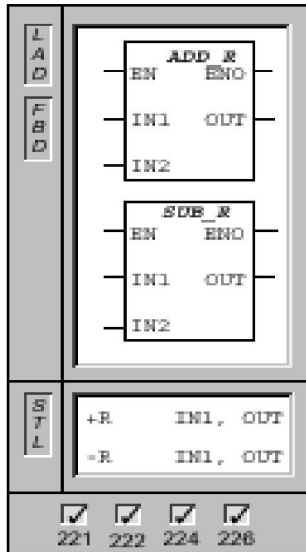
Nếu SM1.3 (chia cho 0) là thiết lập trong khi hoạt động chia, khi đó các bit trạng thái phép toán khác là không được thay đổi trái và các toán hạng đầu vào ban đầu là không biến đổi. Nếu không thì, tất cả các bit trạng thái phép toán được hỗ trợ bao gồm trạng thái có hiệu lực trên sự hoàn thiện của hoạt động phép toán.

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
--------------------	---------------	------------------

IN1, IN2	VW, IW, QW, MW, SW, SMW, LW, AC, AIW, T, C, hằng số, *VD, *AC, *LD	INT
OUT	VD, ID, QD, MD, SMW, SD, LD, AC, *VD, *AC, *LD	DINT

2. Các phép toán số thực:

a. Add Real, Subtract Real



Chỉ dẫn Add Real và Add Subtract cộng hoặc trừ hai số thực 32 bit và cung cấp một kết quả số thực 32 bit (OUT).

Trong LAD và FBD: $IN1 + IN2 = OUT$

$$IN1 - IN2 = OUT$$

Trong STL:

$$IN1 + OUT = OUT$$

$$OUT - IN1 = OUT$$

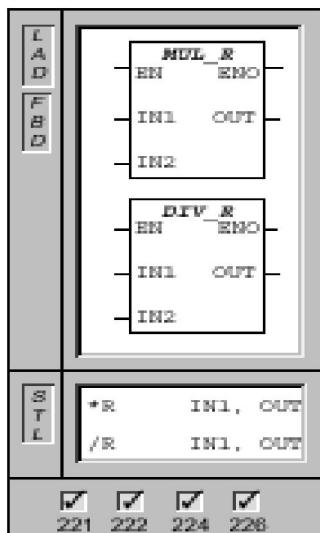
Điều kiện lỗi khi đặt ENO = 0: SM1.1 (tràn), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp)

Các chỉ dẫn ảnh hưởng tới các bit bộ nhớ đặt biệt như sau: SM1.0 (zero); SM1.1 (tràn); SM1.2 (âm).

SM1.1 là được sử dụng để chỉ thị các lỗi tràn và các giá trị phạm luật. Nếu SM1.0 là thiết lập, khi đó trạng thái của SM1.0 và SM1.2 là không có hiệu lực và các toán hạng đầu vào ban đầu là không biến đổi. Nếu SM1.1 là không thiết lập, khi đó hoạt động phép toán được hoàn thiện với một kết quả có hiệu lực và SM1.0 và SM1.2 chứa trạng thái có hiệu lực.

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN1, IN2	VD, ID, QD, MD, SD, SMD, AC, LD, hằng số, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SD, SMD, AC, LD, *VD, *AC, *LD	REAL

b. Multiply Real, Divide Real



Chỉ dẫn Multiply Real nhân hai số thực 32 bit và cung cấp một kết quả số thực 32 bit (OUT).

Chỉ dẫn Divide Real chia hai số thực 32 bit và cung cấp một kết quả số thực 32 bit (OUT).

Trong LAD và FBD: $IN1 * IN2 = OUT$

$IN1 / IN2 = OUT$

Trong STL: $IN1 * OUT = OUT$

$OUT / IN1 = OUT$

Điều kiện lỗi khi đặt ENO = 0: SM1.1 (tràn), SM1.3 (chia cho 0), SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp)

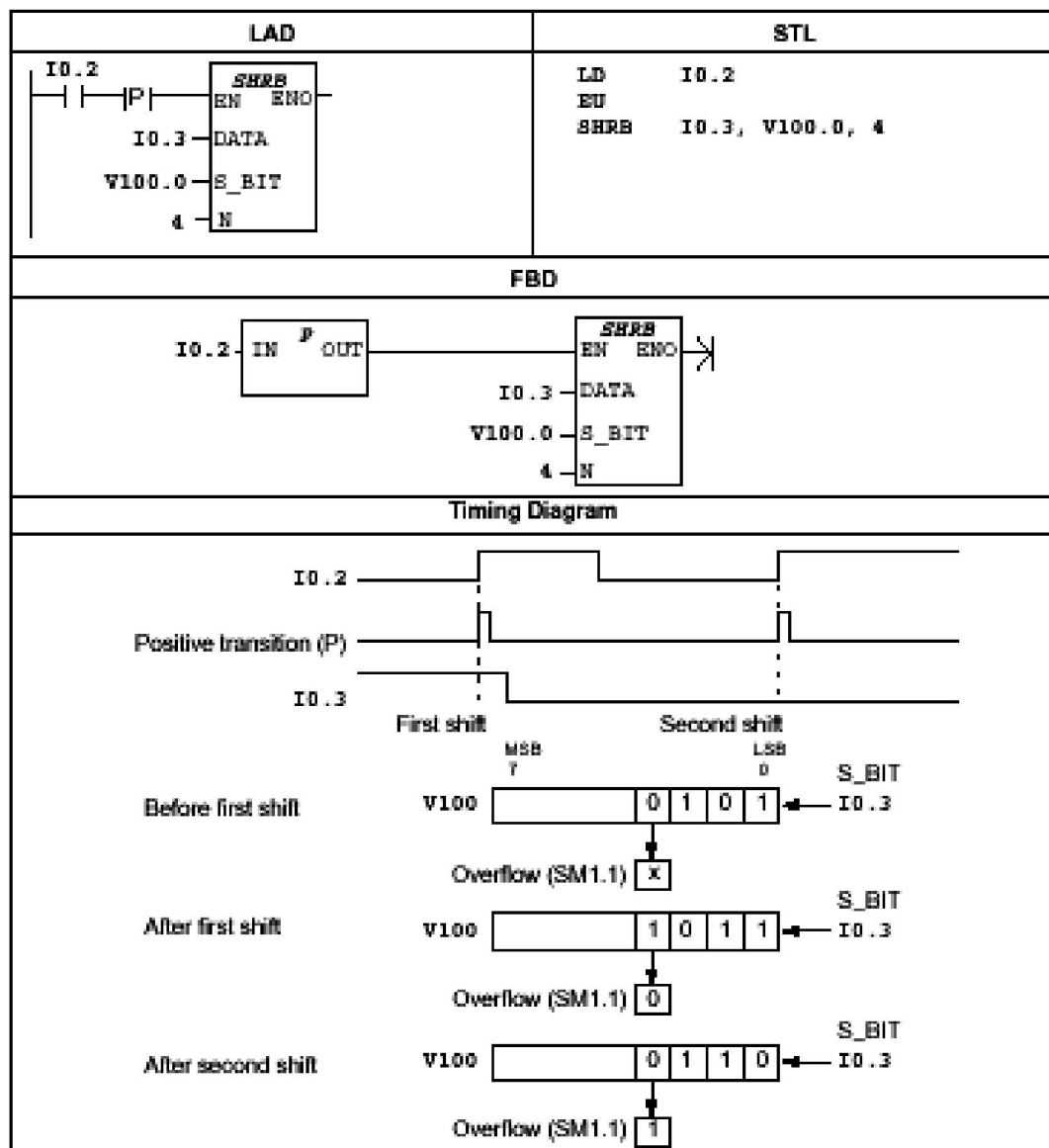
Các chỉ dẫn ảnh hưởng tới các bit bộ nhớ đặt biệt như sau: SM1.0 (zero); SM1.1 (tràn hoặc giá trị phạm luật được sinh ra trong khi hoạt động hoặc đầu vào vi phạm thông số cơ sở); SM1.2 (âm).

SM1.3 là thiết lập trong khi hoạt động chia, khi đó các bit trạng thái phép toán khác là không thay đổi trái và các toán hạng đầu vào ban đầu không biến đổi. SM1.1 được sử dụng để chỉ thị các lỗi tràn và các giá trị phạm luật. Nếu SM1.1 là thiết lập, khi đó trạng thái của SM1.0 và SM1.2 là không có hiệu lực và các toán hạng đầu vào ban đầu là không biến đổi. Nếu SM1.1 và SM1.3 (đang hoạt động chia) là không thiết lập, khi đó hoạt động phép toán được hoàn thiện với một kết quả có hiệu lực và SM1.0 và SM1.2 chứa trạng thái có hiệu lực.

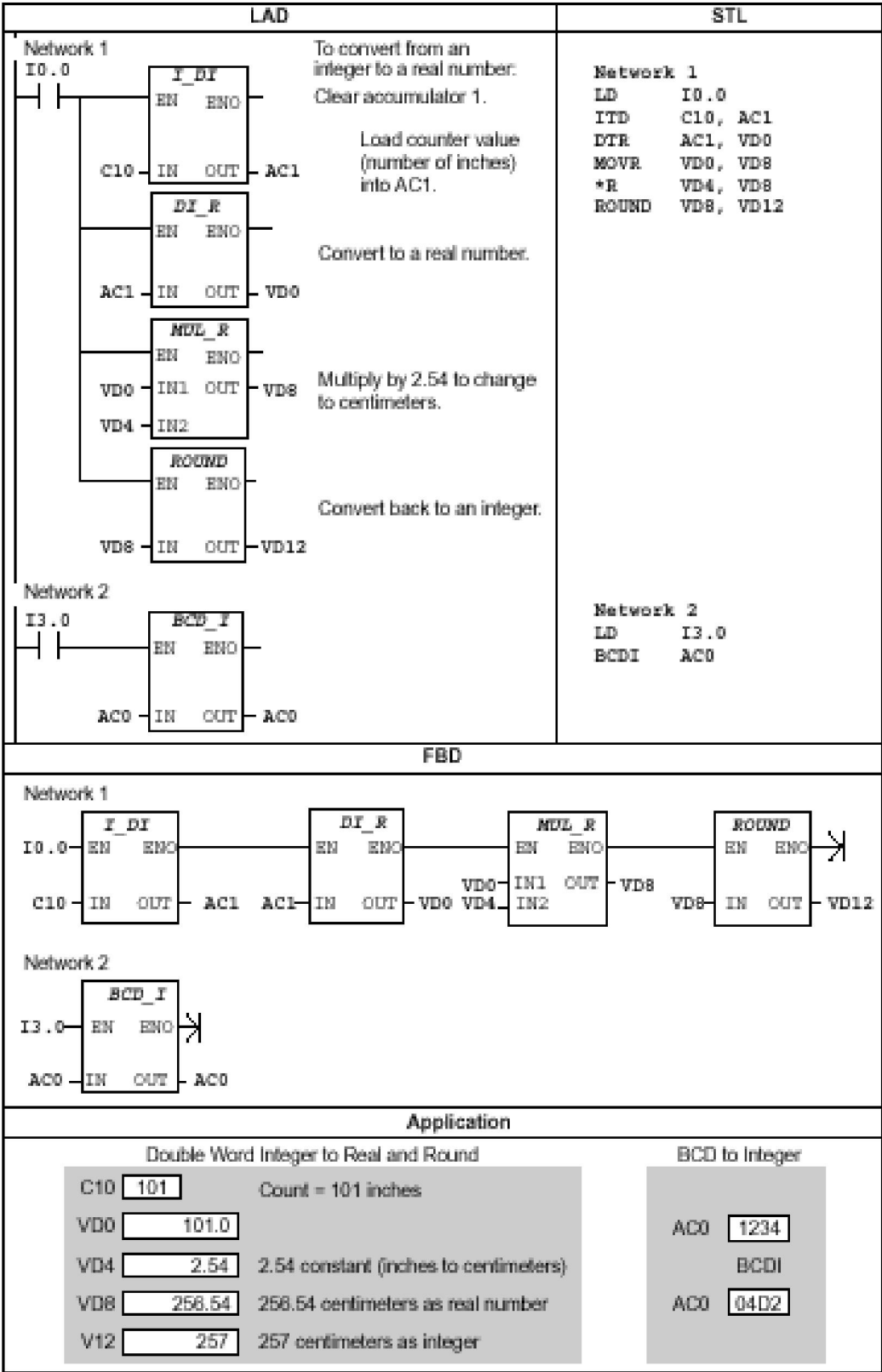
Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
IN1, IN2	VD, ID, QD, MD, SD, SMD, AC, LD, hằng số, *VD, *AC, *LD	REAL
OUT	VD, ID, QD, MD, SD, SMD, AC, LD, *VD, *AC, *LD	REAL

IV. Bài tập:

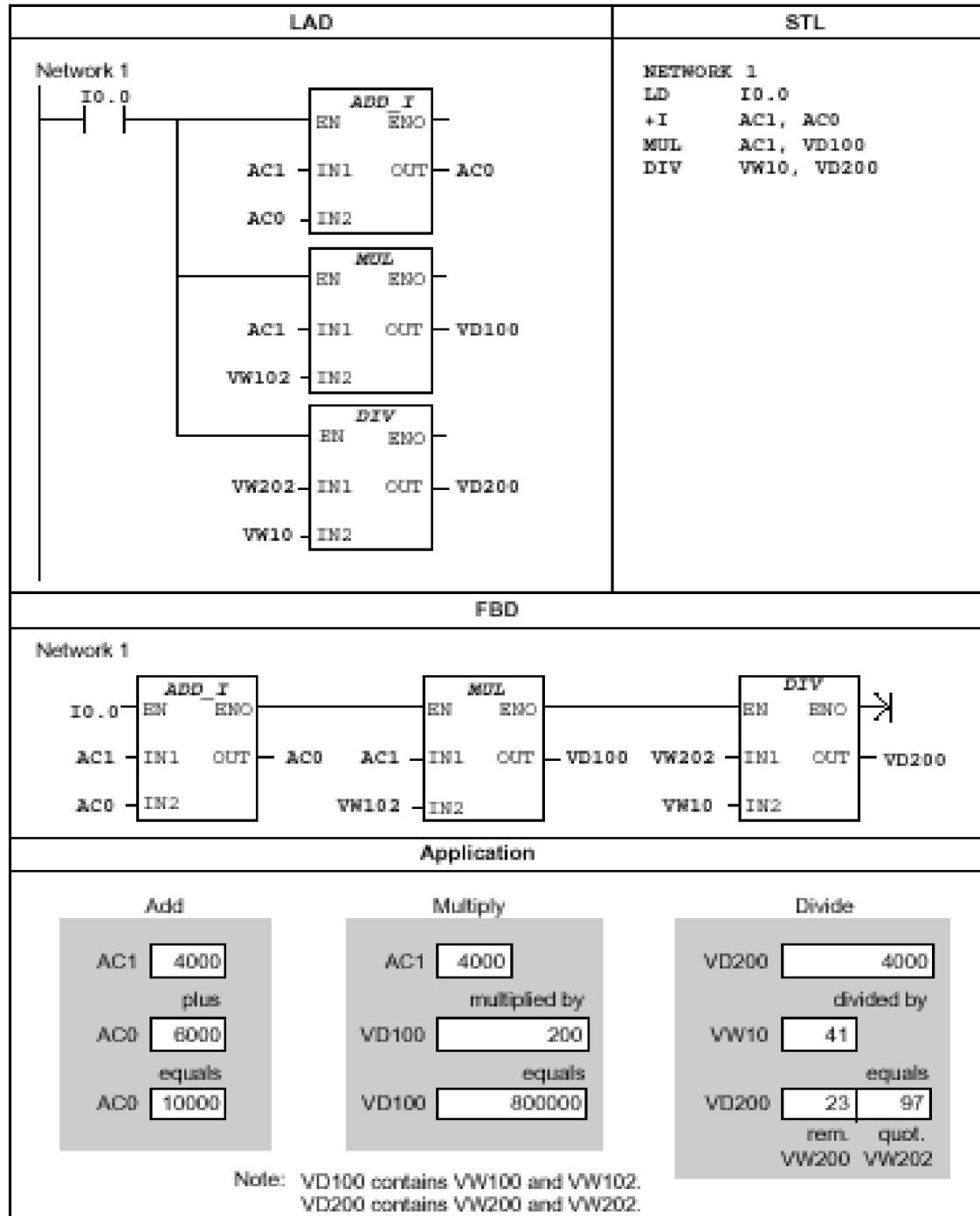
1. Thanh ghi dịch bit



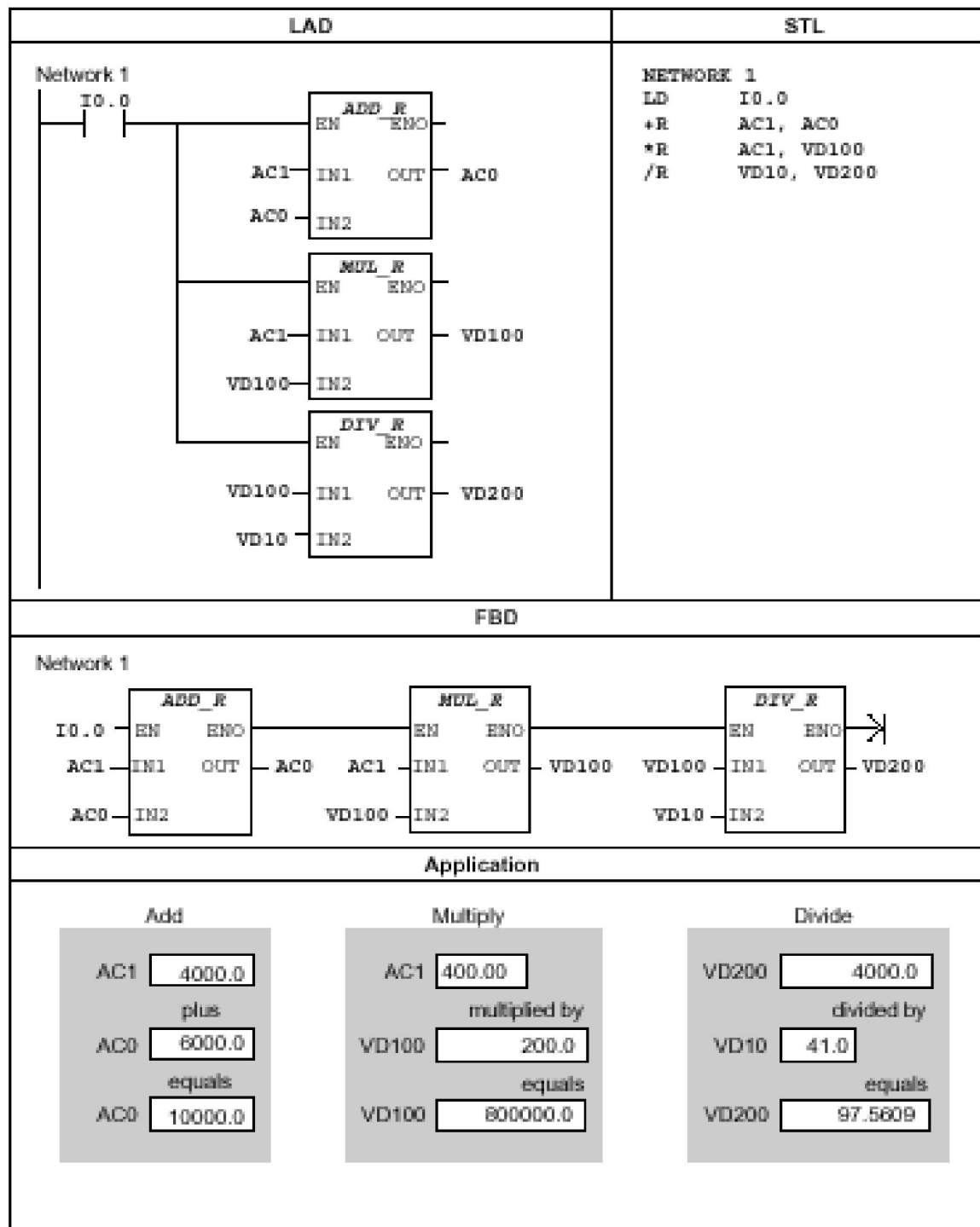
2. Ví dụ về chuyển đổi



3. Các ví dụ phép toán 1

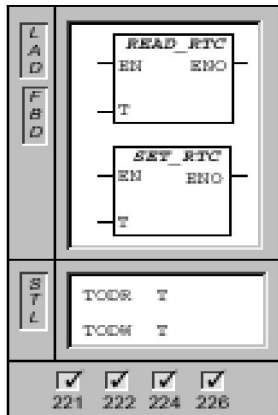


4. Các ví dụ phép toán 2



BÀI 3.3: ĐỒNG HỒ THỜI GIAN THỰC

1. Read Real-Time Clock, Set Real-Time Clock



Chỉ dẫn Read Real-Time Clock đọc thời gian hiện tại và ngày từ xung nhịp và tải nó trong một khởi động bộ đệm 8 byte tại địa chỉ T.

Chỉ dẫn Set Real-Time Clock ghi thời gian hiện tại và ngày tới xung nhịp bắt đầu tại bộ đệm 8 byte địa chỉ riêng bởi T.

Trong giản đồ STL, các chỉ dẫn TODR và TODW là được miêu tả như đọc thời gian của ngày (TODR) và ghi thời gian của ngày (TODW).

TODR: các điều kiện lỗi khi đặt ENO = 0: SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp), 000C (trục quay xung nhịp không hiện diện).

TODW: các điều kiện lỗi khi đặt ENO = 0: SM4.3 (đang chạy), 0006 (địa chỉ gián tiếp), 0007 (lỗi dữ liệu TOD), 000C (trục quay xung nhịp không hiện diện).

Các đầu vào/đầu ra	Các toán hạng	Các kiểu dữ liệu
T	VB, IB, QB, MB, SMB, SB, LB, *VD, *AC, *LD	BYTE

T	T+1	T+2	T+3	T+4	T+5	T+6	T+7
Năm	Tháng	ngày	giờ	phút	giây	0	ngày của tuần

Hình 3.3. Định dạng của bộ đệm thời gian

Khởi tạo xung nhịp thời gian của ngày, thời gian và ngày sau khi nguồn nuôi mở rộng bị yếu điện hoặc bộ nhớ bị mất như sau:

Date: 01-Jan-90
Time: 00:00:00
Day of Week Sunday