

CHƯƠNG 3: NGÔN NGỮ LẬP TRÌNH VÀ ỨNG DỤNG

3.1. Giới thiệu các ngôn ngữ lập trình:

Lập trình cho S7 200 và các PLC khác của hãng Siemens dựa trên 3 phương pháp cơ bản:

Phương pháp hình thang (Ladder logic _ LAD).

Phương pháp khối hàm (Function Block Diagram _ FBD).


Phương pháp liệt kê câu lệnh (Statement List _ STL).


Chương này sẽ giới thiệu các thành phần cơ bản của ba phương pháp và cách sử dụng chúng trong lập trình.

Nếu chương trình được viết theo ngôn ngữ LAD (hoặc FBD) thì có thể chuyển sang ngôn ngữ STL hay FBD (hoặc LAD) tương ứng. Nhưng không phải bất cứ chương trình viết theo STL nào cũng chuyển sang ngôn ngữ LAD hay FBD được. Bộ tập lệnh STL được trình bày trong giáo án này đều có một chức năng như các tiếp điểm, cuộn dây, các hộp (trong LAD) hay IC số trong FBD.

Những lệnh này phải phối hợp được trạng thái các tiếp điểm để quyết định về giá trị trạng thái đầu ra hoặc giá trị logic cho phép hoặc không cho phép thực chức năng của một (hay nhiều) cuộn dây hoặc hộp. Trong lập trình logic thường hay sử dụng hai ngôn ngữ LAD và STL vì nó gần gũi hơn đối với chuyên ngành điện. Sau đây là những định nghĩa cần phải nắm khi bắt tay vào thiết kế một chương trình:

3.1.1. Định nghĩa về LAD: LAD là ngôn ngữ lập trình bằng đồ họa. Những thành phần cơ bản dùng trong LAD tương ứng với những thành phần cơ bản dùng trong bảng mạch role.

+ Tiếp điểm có hai loại: Thường đóng 

Thường hở 

+ Cuộn dây (coil):



+ Hộp (box): Mô tả các hàm khác nhau, nó làm việc khi có tín hiệu đưa đến hộp. Có các nhóm hộp sau: hộp các bộ định thời, hộp các bộ đếm, hộp di chuyển dữ liệu, hộp các hàm toán học, hộp trong truyền thông mạng...

+ Mạng LAD: Là mạch nối các phần tử thành một mạng hoàn thiện, các phần tử như cuộn dây hoặc các hộp phải được mắc đúng chiều. Nguồn điện có hai đường chính, một đường bên trái thể hiện dây nóng, một đường bên phải là dây trung tính (neutral) nhưng không được thể hiện trên giao diện lập trình. Một mạch làm việc được khi các phần tử được mắc đúng chiều và kín mạch.

3.1.2. Định nghĩa về STL: Là phương pháp thể hiện chương trình dưới dạng tập hợp các câu lệnh. Để tạo ra một chương trình bằng STL, người lập trình cần phải hiểu rõ phương thức sử dụng 9 bit trong ngăn xếp (stack) logic của S7 200.

Ngăn xếp là một khối 9 bit chồng lên nhau từ S0÷S8, nhưng tất cả các thuật toán liên quan đến ngăn xếp đều làm việc với bit đầu tiên và bit thứ hai (S0 và S1) của ngăn xếp. giá trị logic mới có thể được gởi hoặc nối thêm vào ngăn xếp. Hai bit S0 và S1 phối hợp với nhau thì ngăn xếp được kéo lên một bit.

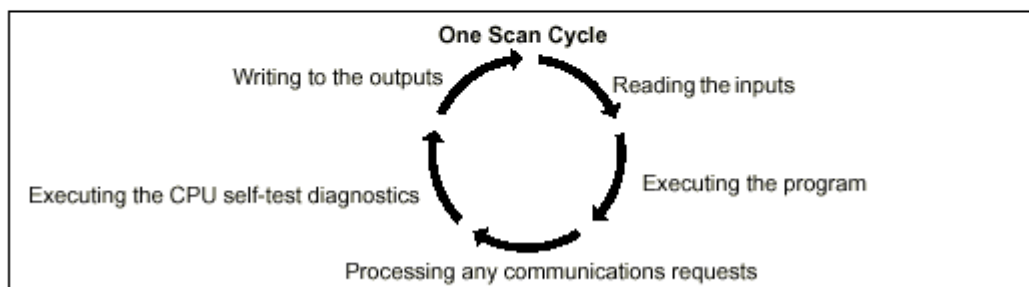
Ngăn xếp của S7 200 (logic stack):

S0	Stack0 bit đầu tiên của ngăn xếp.
S1	Stack1 bit thứ hai của ngăn xếp.
S2	Stack2 bit thứ ba của ngăn xếp.
S3	Stack3 bit thứ tư của ngăn xếp.
S4	Stack4 bit thứ năm của ngăn xếp.
S5	Stack5 bit thứ sáu của ngăn xếp.
S6	Stack6 bit thứ bảy của ngăn xếp.
S7	Stack7 bit thứ tám của ngăn xếp.
S8	Stack8 bit thứ chín của ngăn xếp.

3.2. Vòng quét (thực hiện chương trình) và cấu trúc của một chương trình:

PLC thực hiện chương trình theo vòng lặp. Mỗi vòng lặp được gọi là vòng quét (scan). Các giai đoạn của vòng quét:

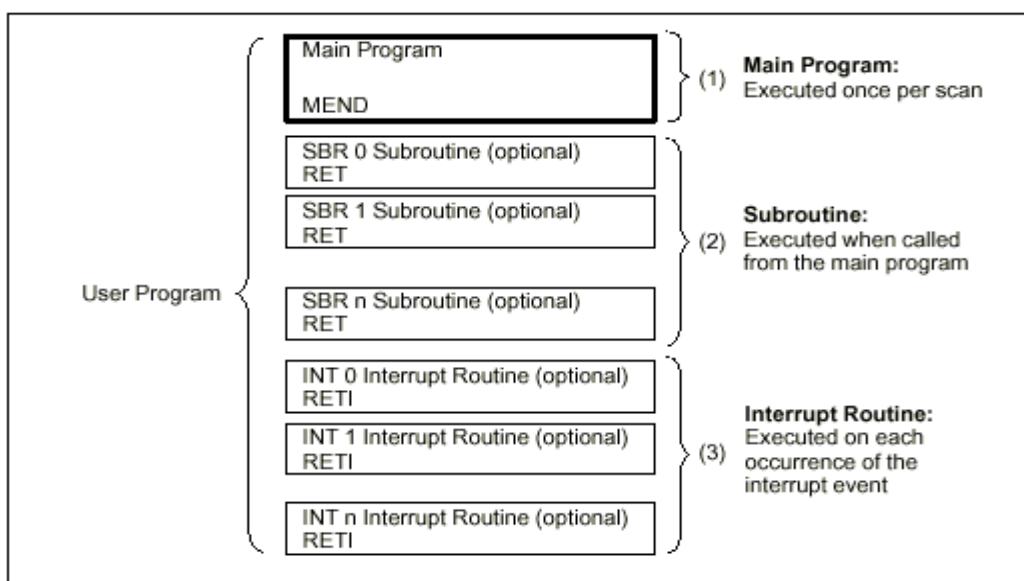
- Reading the inputs
- Executing the program
- Processing any communication requests
- Executing the CPU self-test diagnostics
- Writing to the outputs



Hình 3.1: Chu kỳ vòng quét của CPU S7-200.

Khi gặp lệnh vào/ra tức thời ngay lập tức hệ thống dừng tất cả mọi công việc khác, ngay cả chương trình xử lý ngắt để thực hiện chương trình này trực tiếp với cổng vào/ra.

Nếu sử dụng các chế độ ngắt, chương trình con tương ứng với từng tín hiệu ngắt được soạn thảo và cài đặt như một bộ phận của chương trình. Chương trình xử lý ngắt chỉ được thực hiện trong vòng quét khi xuất hiện tín hiệu báo ngắt và có thể xảy ra ở bất cứ thời điểm nào trong vòng quét.



Hình 3.2: Cấu trúc của chương trình S7-200

3.3. Tập lệnh S7-200:

Tập lệnh của S7-200 được chia làm 3 nhóm:

1. Các lệnh mà khi thực hiện thì làm việc độc lập không phụ thuộc vào giá trị logic của bit đầu tiên trong ngăn xếp (gọi là nhóm lệnh không điều kiện).
2. Các lệnh chỉ thực hiện khi bit đầu tiên trong ngăn xếp có giá trị bằng 1 (gọi là nhóm lệnh có điều kiện).
3. Các nhãn lệnh đánh dấu vị trí trong tập lệnh (gọi là nhóm lệnh điều khiển chương trình).

! Các ngôn ngữ sử dụng chữ *I* (Immediately) để chỉ ý nghĩa tức thời.

Instructions	Cây lệnh
Bit Logic	Tập lệnh Bit
Clock	Tập lệnh can thiệp vào thời gian hệ thống
Communications	Tập lệnh truyền thông
Compare	Tập lệnh so sánh
Convert	Tập lệnh biến đổi
Counters	Tập các bộ đếm
Floating-Point Math	Tập lệnh toán học
Integer Math	Tập lệnh toán học
Interrupt	Tập lệnh điều khiển ngắt
Logical Operations	Tập lệnh các phép tính logic biến đổi
Move	Tập lệnh di chuyển dữ liệu
Program Control	Tập lệnh điều khiển chương trình
Shift/Rotate	Tập lệnh thao tác với thanh ghi (dịch/quay vòng thanh ghi)
String	Tập lệnh làm việc với chuỗi
Table	Tập lệnh làm việc với bảng dữ liệu
Timers	Tập các bộ định thời
Subroutine	Tập lệnh gọi chương trình con và chương trình ngắt

Hình 3.3: Mô tả cây lệnh với SIMATIC S7-200

	Bit Logic	
1	LD BIT	Load
1	A BIT	AND
1	O BIT	OR
1	LDN BIT	Load Not
1	AN BIT	AND Not
1	ON BIT	OR Not
1	LDI BIT	Load Immediate
1	AI BIT	AND Immediate
1	OI BIT	OR Immediate
1	LDNI BIT	Load Not Immediate
1	ANI BIT	AND Not Immediate
1	ONI BIT	OR Not Immediate
1	NOT	Logical Not
1	EU	Edge Up
1	ED	Edge Down
1	ALD	AND Load
1	OLD	OR Load
1	LPS	Logic Push
1	LDS N	Load Stack
1	LRD	Logic Read
1	LPP	Logic Pop
1	= BIT	Output
1	=I BIT	Output Immediate
2	S BIT, N	Set
2	SI BIT, N	Set Immediate
2	R BIT, N	Reset
2	RI BIT, N	Reset Immediate
2	AENO	AND ENO
3	NOP N	No Operation

Hình 3.4: Mô tả cây lệnh bit








































	Clock	
2	TODR T	Time of Day Clock Read
2	TODW T	Time of Day Clock Write

Hình 3.5: Mô tả cây lệnh can thiệp vào thời gian hệ thống

	Communications	
2	XMT TABLE, PORT	Transmit Message
2	RCV TABLE, PORT	Receive Message
2	NETR TABLE, PORT	Network Read
2	NETW TABLE, PORT	Network Write
2	GPA ADDR, PORT	Get Port Address
2	SPA ADDR, PORT	Set Port Address

Hình 3.6: Mô tả cây lệnh truyền thông

Compare		
1	LDB= IN1, IN2	Load Byte Equal
1	AB= IN1, IN2	AND Byte Equal
1	OB= IN1, IN2	OR Byte Equal
1	LDB<> IN1, IN2	Load Byte Not Equal
1	AB<> IN1, IN2	AND Byte Not Equal
1	OB<> IN1, IN2	OR Byte Not Equal
1	LDB>= IN1, IN2	Load Byte Greater Than or Equal
1	AB>= IN1, IN2	AND Byte Greater Than or Equal
1	OB>= IN1, IN2	OR Byte Greater Than or Equal
1	LDB<= IN1, IN2	Load Byte Less Than or Equal
1	AB<= IN1, IN2	AND Byte Less Than or Equal
1	OB<= IN1, IN2	OR Byte Less Than or Equal
1	LDB> IN1, IN2	Load Byte Greater Than
1	AB> IN1, IN2	AND Byte Greater Than
1	OB> IN1, IN2	OR Byte Greater Than
1	LDB< IN1, IN2	Load Byte Less Than
1	AB< IN1, IN2	AND Byte Less Than
1	OB< IN1, IN2	OR Byte Less Than
1	LDW= IN1, IN2	Load Word Equal
1	AW= IN1, IN2	AND Word Equal
1	OW= IN1, IN2	OR Word Equal
1	LDW<> IN1, IN2	Load Word Not Equal
1	AW<> IN1, IN2	AND Word Not Equal
1	OW<> IN1, IN2	OR Word Not Equal
1	LDW>= IN1, IN2	Load Word Greater Than or Equal
1	AW>= IN1, IN2	AND Word Greater Than or Equal
1	OW>= IN1, IN2	OR Word Greater Than or Equal
1	LDW<= IN1, IN2	Load Word Less Than or Equal
1	AW<= IN1, IN2	AND Word Less Than or Equal
1	OW<= IN1, IN2	OR Word Less Than or Equal
1	LDW> IN1, IN2	Load Word Greater Than
1	AW> IN1, IN2	AND Word Greater Than
1	OW> IN1, IN2	OR Word Greater Than
1	LDW< IN1, IN2	Load Word Less Than
1	AW< IN1, IN2	AND Word Less Than
1	OW< IN1, IN2	OR Word Less Than
1	LDD= IN1, IN2	Load Double Word Equal
1	AD= IN1, IN2	AND Double Word Equal
1	OD= IN1, IN2	OR Double Word Equal

1		LDD<> IN1, IN2	Load Double Word Not Equal
1		AD<> IN1, IN2	AND Double Word Not Equal
1		OD<> IN1, IN2	OR Double Word Not Equal
1		LDD>= IN1, IN2	Load Double Word Greater Than or Equal
1		AD>= IN1, IN2	AND Double Word Greater Than or Equal
1		OD>= IN1, IN2	OR Double Word Greater Than or Equal
1		LDD<= IN1, IN2	Load Double Word Less Than or Equal
1		AD<= IN1, IN2	AND Double Word Less Than or Equal
1		OD<= IN1, IN2	OR Double Word Less Than or Equal
1		LDD> IN1, IN2	Load Double Word Greater Than
1		AD> IN1, IN2	AND Double Word Greater Than
1		OD> IN1, IN2	OR Double Word Greater Than
1		LDD< IN1, IN2	Load Double Word Less Than
1		AD< IN1, IN2	AND Double Word Less Than
1		OD< IN1, IN2	OR Double Word Less Than
1		LDR= IN1, IN2	Load Real Equal
1		AR= IN1, IN2	AND Real Equal
1		OR= IN1, IN2	OR Real Equal
1		LDR<> IN1, IN2	Load Real Not Equal
1		AR<> IN1, IN2	AND Real Not Equal
1		OR<> IN1, IN2	OR Real Not Equal
1		LDR>= IN1, IN2	Load Real Greater Than or Equal
1		AR>= IN1, IN2	AND Real Greater Than or Equal
1		OR>= IN1, IN2	OR Real Greater Than or Equal
1		LDR<= IN1, IN2	Load Real Less Than or Equal
1		AR<= IN1, IN2	AND Real Less Than or Equal
1		OR<= IN1, IN2	OR Real Less Than or Equal
1		LDR> IN1, IN2	Load Real Greater Than
1		AR> IN1, IN2	AND Real Greater Than
1		OR> IN1, IN2	OR Real Greater Than
1		LDR< IN1, IN2	Load Real Less Than
1		AR< IN1, IN2	AND Real Less Than
1		OR< IN1, IN2	OR Real Less Than
1		LDS= IN1, IN2	Load String Equal
1		AS= IN1, IN2	AND String Equal
1		OS= IN1, IN2	OR String Equal
1		LDS<> IN1, IN2	Load String Not Equal
1		AS<> IN1, IN2	AND String Not Equal
1		OS<> IN1, IN2	OR String Not Equal

Hình 3.7: Mô tả cây lệnh so sánh

	Convert	
2	BTI IN, OUT	Byte to Integer
2	ITB IN, OUT	Integer to Byte
2	ITD IN, OUT	Integer to Double Integer
2	ITS IN, OUT, FMT	Integer to String
2	DTI IN, OUT	Double Integer to Integer
2	DTR IN, OUT	Double Integer to Real
2	DTS IN, OUT, FMT	Double Integer to String
2	ROUND IN, OUT	Round
2	TRUNC IN, OUT	Truncate
2	RTS IN, OUT, FMT	Real to String
2	BCDI OUT	Binary Coded Decimal to Integer
2	IBCD OUT	Integer to Binary Coded Decimal
2	ITA IN, OUT, FMT	Integer to ASCII
2	DTA IN, OUT, FMT	Double Integer to ASCII
2	RTA IN, OUT, FMT	Real to ASCII
2	ATH IN, OUT, LEN	ASCII to Hexadecimal
2	HTA IN, OUT, LEN	Hexadecimal to ASCII
2	STI IN, INDX, OUT	String to Integer
2	STD IN, INDX, OUT	String to Double Integer
2	STR IN, INDX, OUT	String to Real
2	DECO IN, OUT	Decode
2	ENCO IN, OUT	Encode
2	SEG IN, OUT	Seven Segment Display Conversion

Hình 3.8: Mô tả cây lệnh biến đổi

	Counter	
2	CTU C, PV	Count Up
2	CTD C, PV	Count Down
2	CTUD C, PV	Count Up/Down
2	HDEF HSC, MODE	High Speed Counter Definition
2	HSC N	High-Speed Counter
2	PLS Q	Pulse Output

Hình 3.9: Mô tả cây lệnh các bộ đếm

	Timer	
2	TON T, PT	On-Delay Timer
2	TONR T, PT	Retentive On-Delay Timer
2	TOF T, PT	Off-Delay Timer

Hình 3.10: Mô tả cây lệnh các bộ định thời

	Interrupt	
2	CRETI	Conditional Return from Interrupt (INT)
2	ENI	Enable Interrupt
2	DISI	Disable Interrupt
2	ATCH INT, EVENT	Attach Interrupt
2	DTCH EVENT	Detach Interrupt

Hình 3.11: Mô tả cây lệnh điều khiển ngắt

Floating-Point Math		
2	+R IN1, OUT	Add Real Numbers
2	-R IN1, OUT	Subtract Real Numbers
2	*R IN1, OUT	Multiply Real Numbers
2	/R IN1, OUT	Divide Real Numbers
2	SQRT IN1, OUT	Square Root
2	SIN IN, OUT	Sine
2	COS IN, OUT	Cosine
2	TAN IN, OUT	Tangent
2	LN IN, OUT	Natural Logarithm
2	EXP IN, OUT	Natural Exponential
2	PID TABLE, LOOP	PID Calculation

Hình 3.12: Mô tả cây lệnh học kiểu Floating-Point

Integer Math		
2	+I IN1, OUT	Add Integer
2	+D IN1, OUT	Add Double Integer
2	-I IN1, OUT	Subtract Integer
2	-D IN1, OUT	Subtract Double Integer
2	MUL IN1, OUT	Multiply Integer to Double Integer
2	*I IN1, OUT	Multiply Integer
2	*D IN1, OUT	Multiply Double Integer
2	DIV IN1, OUT	Divide Integer to Quotient/Remainder
2	/I IN1, OUT	Divide Integer
2	/D IN1, OUT	Divide Double Integer
2	INCB OUT	Increment Byte
2	INCW OUT	Increment Word
2	INCD OUT	Increment Double Word
2	DECB OUT	Decrement Byte
2	DECW OUT	Decrement Word
2	DECD OUT	Decrement Double Word

Hình 3.13: Mô tả cây lệnh toán học kiểu Integer

Logical Operations		
2	INVB OUT	Invert Byte
2	INVW OUT	Invert Word
2	INVD OUT	Invert Double Word
2	ANDB IN1, OUT	AND Byte
2	ANDW IN1, OUT	AND Word
2	ANDD IN, OUT	AND Double Word
2	ORB IN1, OUT	OR Byte
2	ORW IN1, OUT	OR Word
2	ORD IN1, OUT	OR Double Word
2	XORB IN1, OUT	Exclusive OR Byte
2	XORW IN1, OUT	Exclusive OR Word
2	XORD IN1, OUT	Exclusive OR Double Word

Hình 3.14: Mô tả cây lệnh phép tính logic biến đổi

Move		
2	MOVB IN, OUT	Move Byte
2	MOVW IN, OUT	Move Word
2	MOVD IN, OUT	Move Double Word
2	MOVR IN, OUT	Move Real
2	BMB IN, OUT, N	Block Move Bytes
2	BMW IN, OUT, N	Block Move Words
2	BMD IN, OUT, N	Block Move Double Words
2	SWAP IN	Swap Bytes
2	BIR IN, OUT	Move Byte Immediate from Inputs
2	BIW IN, OUT	Move Byte Immediate to Outputs

Hình 3.15: Mô tả cây lệnh di chuyển dữ liệu

Program Control		
3	FOR INDEX, INITIAL, FINAL	For/Next Loop
3	NEXT	For/Next Loop
3	JMP N	Jump to Label
3	LBL N	Label
3	LSCR N	Load Sequential Control Relay
3	SCRT N	Sequential Control Relay Transition
3	SCRE	Sequential Control Relay End
3	CSCRE	Conditional Sequential Control Relay End
3	CRET	Conditional Return from Subroutine (SBR)
3	END	Conditional End of Program (OB1)
3	STOP	Transition to Stop Mode
3	WDR	Watchdog Reset

Hình 3.16: Mô tả cây lệnh điều khiển chương trình

Shift/Rotate		
2	SLB OUT, N	Shift Left Byte
2	SLW OUT, N	Shift Left Word
2	SLD OUT, N	Shift Left Double Word
2	SRB OUT, N	Shift Right Byte
2	SRW OUT, N	Shift Right Word
2	SRD OUT, N	Shift Right Double Word
2	RLB OUT, N	Rotate Left Byte
2	RLW OUT, N	Rotate Left Word
2	RLD OUT, N	Rotate Left Double Word
2	RRB OUT, N	Rotate Right Byte
2	RRW OUT, N	Rotate Right Word
2	RRD OUT, N	Rotate Right Double Word
2	SHRB DATA, S_BIT, N	Shift Register Bit

Hình 3.17: Mô tả cây lệnh điều khiển chương trình



Hình 3.18: Mô tả cây lệnh làm việc với chuỗi



Hình 3.19: Mô tả cây lệnh làm việc với bảng dữ liệu

- ! 1_ Các lệnh không điều kiện.
- 2_ Các lệnh có điều kiện.
- 3_ Các lệnh điều khiển chương trình.

3.4. Cú pháp và cách ứng dụng SIMATIC struction S7-200:

3.4.1. Toán hạng và giới hạn cho phép:

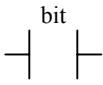
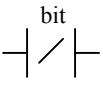
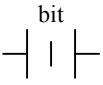
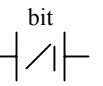



Bảng 3.1: Giới hạn toán hạng của CPU S7-200 series CPU 22x

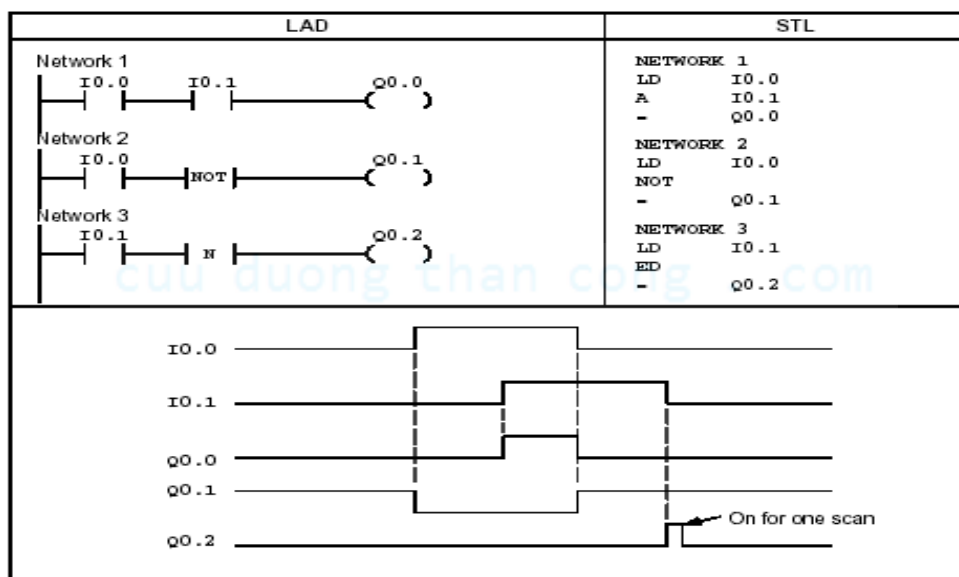
Access Method		CPU 221	CPU 222	CPU 224	CPU 224 XP	CPU 226
Bit access (byte.bit)	I	0.0 to 15.7	0.0 to 15.7	0.0 to 15.7	0.0 to 15.7	0.0 to 15.7
	Q	0.0 to 15.7	0.0 to 15.7	0.0 to 15.7	0.0 to 15.7	0.0 to 15.7
	V	0.0 to 2047.7	0.0 to 2047.7	0.0 to 8191.7	0.0 to 10239.7	0.0 to 10239.7
	M	0.0 to 31.7	0.0 to 31.7	0.0 to 31.7	0.0 to 31.7	0.0 to 31.7
	SM	0.0 to 165.7	0.0 to 299.7	0.0 to 549.7	0.0 to 549.7	0.0 to 549.7
	S	0.0 to 31.7	0.0 to 31.7	0.0 to 31.7	0.0 to 31.7	0.0 to 31.7
	T	0 to 255	0 to 255	0 to 255	0 to 255	0 to 255
	C	0 to 255	0 to 255	0 to 255	0 to 255	0 to 255
	L	0.0 to 63.7	0.0 to 63.7	0.0 to 63.7	0.0 to 63.7	0.0 to 63.7
Byte access	IB	0 to 15	0 to 15	0 to 15	0 to 15	0 to 15
	QB	0 to 15	0 to 15	0 to 15	0 to 15	0 to 15
	VB	0 to 2047	0 to 2047	0 to 8191	0 to 10239	0 to 10239
	MB	0 to 31	0 to 31	0 to 31	0 to 31	0 to 31
	SMB	0 to 165	0 to 299	0 to 549	0 to 549	0 to 549
	SB	0 to 31	0 to 31	0 to 31	0 to 31	0 to 31
	LB	0 to 63	0 to 63	0 to 63	0 to 63	0 to 63
	AC	0 to 3	0 to 3	0 to 3	0 to 255	0 to 255
	KB (Constant)	KB (Constant)	KB (Constant)	KB (Constant)	KB (Constant)	KB (Constant)
Word access	IW	0 to 14	0 to 14	0 to 14	0 to 14	0 to 14
	QW	0 to 14	0 to 14	0 to 14	0 to 14	0 to 14
	VW	0 to 2046	0 to 2046	0 to 8190	0 to 10238	0 to 10238
	MW	0 to 30	0 to 30	0 to 30	0 to 30	0 to 30
	SMW	0 to 164	0 to 298	0 to 548	0 to 548	0 to 548
	SW	0 to 30	0 to 30	0 to 30	0 to 30	0 to 30
	T	0 to 255	0 to 255	0 to 255	0 to 255	0 to 255
	C	0 to 255	0 to 255	0 to 255	0 to 255	0 to 255
	LW	0 to 62	0 to 62	0 to 62	0 to 62	0 to 62
	AC	0 to 3	0 to 3	0 to 3	0 to 3	0 to 3
	AIW	0 to 30	0 to 30	0 to 62	0 to 62	0 to 62
	AQW	0 to 30	0 to 30	0 to 62	0 to 62	0 to 62
	KW (Constant)	KW (Constant)	KW (Constant)	KW (Constant)	KW (Constant)	KW (Constant)
Double word access	ID	0 to 12	0 to 12	0 to 12	0 to 12	0 to 12
	QD	0 to 12	0 to 12	0 to 12	0 to 12	0 to 12
	VD	0 to 2044	0 to 2044	0 to 8188	0 to 10236	0 to 10236
	MD	0 to 28	0 to 28	0 to 28	0 to 28	0 to 28
	SMD	0 to 162	0 to 296	0 to 546	0 to 546	0 to 546
	SD	0 to 28	0 to 28	0 to 28	0 to 28	0 to 28
	LD	0 to 60	0 to 60	0 to 60	0 to 60	0 to 60
	AC	0 to 3	0 to 3	0 to 3	0 to 3	0 to 3
	HC	0 to 5	0 to 5	0 to 5	0 to 5	0 to 5
	KD (Constant)	KD (Constant)	KD (Constant)	KD (Constant)	KD (Constant)	KD (Constant)

3.4.2. SIMATIC instructions:

1. SIMATIC Bit Logic Instructions:

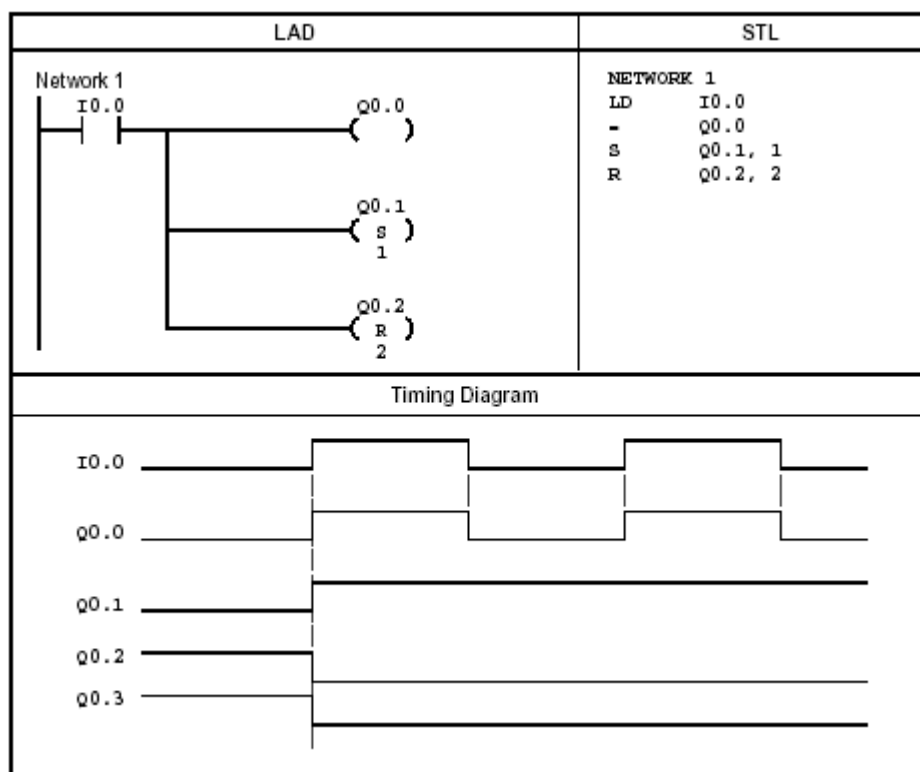
STL	LAD	Mô tả (Description)	Toán hạng (Operands)	Kiểu dữ liệu (Data Types)
-----	-----	------------------------	-------------------------	------------------------------------

LD		Tiếp điểm thường mở sẽ được đóng khi bit = 1	bit: I, Q, M, V, SM, T, C, S, L	Bool
A				
O				
LDN		Tiếp điểm thường đóng sẽ được mở khi bit = 1	bit: I, Q, M, V, SM, T, C, S, L	Bool
AN				
ON				
LDI		Tiếp điểm thường mở sẽ đóng tức thời (không phụ thuộc vào chu kỳ vòng quét)	bit: I	Bool
AI				
OI				
LDNI		Tiếp điểm thường đóng sẽ mở tức thời (không phụ thuộc vào chu kỳ vòng quét)	bit: I	Bool
AIN				
OIN				
NOT		Đảo giá trị logic của bit đầu tiên trong ngăn xếp	Không	Không
EU		Bit đầu tiên trong ngăn xếp có giá trị bằng 1 (trong khoảng thời gian đúng bằng 1 chu kỳ vòng quét) khi phát hiện sườn lên của tín hiệu đầu vào.	bit: I, Q, M, V, SM, T, C, S, L	Bool
ED		Bit đầu tiên trong ngăn xếp có giá trị bằng 1 (trong khoảng thời gian đúng bằng 1 chu kỳ vòng quét) khi phát hiện sườn xuống của tín hiệu đầu vào.	bit: I, Q, M, V, SM, T, C, S, L	Bool



Hình 3.20: Ví dụ minh họa lệnh LD, NOT, ED trong chương trình LAD và STL

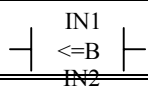
STL	LAD	Mô tả Description	Toán hạng Operands	Kiểu dữ liệu Data Types
= bit		Cuộn dây đầu ra ở trạng thái ON khi có dòng điện điều khiển đi qua.	bit: I, Q, M, V, SM, T, C, S, L	Bool
=I bit		Cuộn dây đầu ra ở trạng thái ON tức thời (không phụ thuộc vào chu kỳ vòng quét) khi có dòng điện điều khiển đi qua.	bit: Q	Bool
S bit, n		Set 1 mảng gồm n tiếp điểm, tính từ tiếp điểm "bit" (n <= 128 tiếp điểm).	bit: I, Q, M, V, SM, T, C, S, L n: IB, QB, MB, VB, SMB, SB, LB, AC, Constant, *VD, *AC, *LD	Bool
R bit, n		Reset 1 mảng gồm n tiếp điểm, tính từ tiếp điểm "bit" (n <= 128 tiếp điểm).	bit: I, Q, M, V, SM, T, C, S, L n: IB, QB, MB, VB, SMB, SB, LB, AC, Constant, *VD, *AC, *LD	Bool
SI bit, n		Set tức thời 1 mảng gồm n tiếp điểm, tính từ tiếp điểm "bit" (n <= 128 tiếp điểm).	bit: Q n: IB, QB, MB, VB, SMB, SB, LB, AC, Constant, *VD, *AC, *LD	Bool
RI bit, n		Reset tức thời 1 mảng gồm n tiếp điểm, tính từ tiếp điểm "bit" (n <= 128 tiếp điểm).	bit: Q n: IB, QB, MB, VB, SMB, SB, LB, AC, Constant, *VD, *AC, *LD	Bool
NOP		Lệnh rỗng, không hoạt động n lần.	n: 0 ÷ 255	Byte

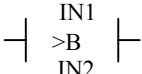
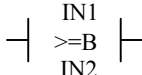
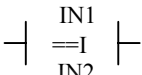
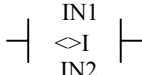
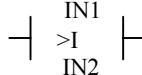
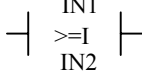
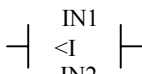
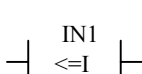


Hình 3.21: Ví dụ minh họa lệnh =, S, R trong chương trình LAD và STL

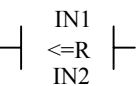
2. SIMATIC Compare Byte Instructions:

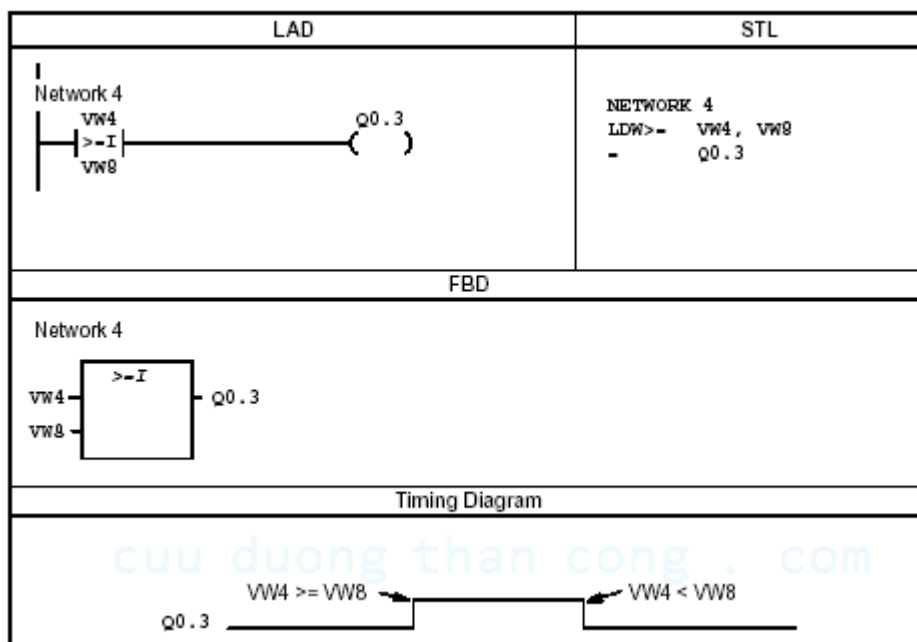
STL	LAD	Mô tả (Description)	Toán hạng (Operands)	Kiểu dữ liệu (Data Types)
COMPARE BYTE				
LDB=		Lệnh so sánh giá trị của hai byte IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1= IN2 là đúng.	IB, QB, MB, VB, SMB, SB, LB, AC, Constant, *VD, *AC, *LD	Byte
AB=				
OB=				
LDB<>		Lệnh so sánh giá trị của hai byte IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1<> IN2 là đúng.	IB, QB, MB, VB, SMB, SB, LB, AC, Constant, *VD, *AC, *LD	Byte
AB<>				
OB<>				
LDB<		Lệnh so sánh giá trị của hai byte IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1< IN2 là đúng.	IB, QB, MB, VB, SMB, SB, LB, AC, Constant, *VD, *AC, *LD	Byte
AB<				
OB<				
LDB<=		Lệnh so sánh giá trị của hai byte	IB, QB, MB,	Byte



		IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh $IN1 \leq IN2$ là đúng.	VB, SMB, SB, LB, AC, Constant, *VD, *AC, *LD	
AB \leq				
OB \leq				
LDB $>$		Lệnh so sánh giá trị của hai byte IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh $IN1 > IN2$ là đúng.	IB, QB, MB, VB, SMB, SB, LB, AC, Constant, *VD, *AC, *LD	Byte
AB $>$				
OB $>$				
LDB \geq		Lệnh so sánh giá trị của hai byte IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh $IN1 \geq IN2$ là đúng.	IB, QB, MB, VB, SMB, SB, LB, AC, Constant, *VD, *AC, *LD	Byte
AB \geq				
OB \geq				
COMPARE WORD (COPARE INTEGER)				
LDW $=$		Lệnh so sánh giá trị của hai Word IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh $IN1 = IN2$ là đúng.	IW, QW, MW, VW, SMW, SW, LW, AC, Constant, *VD, *AC, *LD	Word
AW $=$				
OW $=$				
LDW \diamond		Lệnh so sánh giá trị của hai Word IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh $IN1 \diamond IN2$ là đúng.	IW, QW, MW, VW, SMW, SW, LW, AC, Constant, *VD, *AC, *LD	Word
AW \diamond				
OW \diamond				
LDW $>$		Lệnh so sánh giá trị của hai Word IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh $IN1 > IN2$ là đúng.	IW, QW, MW, VW, SMW, SW, LW, AC, Constant, *VD, *AC, *LD	Word
AW $>$				
OW $>$				
LDW \geq		Lệnh so sánh giá trị của hai Word IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh $IN1 \geq IN2$ là đúng.	IW, QW, MW, VW, SMW, SW, LW, AC, Constant, *VD, *AC, *LD	Word
AW \geq				
OW \geq				
LDW $<$		Lệnh so sánh giá trị của hai Word IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh $IN1 < IN2$ là đúng.	IW, QW, MW, VW, SMW, SW, LW, AC, Constant, *VD, *AC, *LD	Word
AW $<$				
OW $<$				
LDW \leq		Lệnh so sánh giá trị của hai Word IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh $IN1 \leq IN2$ là đúng.	IW, QW, MW, VW, SMW, SW, LW, AC, Constant, *VD, *AC, *LD	Word
AW \leq				
OW \leq				
COMPARE DOUBLEWORD				

LDDW=	<div><div>IN1</div><div>=D</div><div>IN2</div></div>	Lệnh so sánh giá trị của hai DoubleWord IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1 = IN2 là đúng.	ID, QD, MD, VD, SMD, SD, LD, AC, Constant, *VD, *AC, *LD	Double Word
ADW=				
ODW=				
LDDW<	<div><div>IN1</div><div><D</div><div>IN2</div></div>	Lệnh so sánh giá trị của hai DoubleWord IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1 < IN2 là đúng.	ID, QD, MD, VD, SMD, SD, LD, AC, Constant, *VD, *AC, *LD	Double Word
ADW<				
ODW<				
LDDW>	<div><div>IN1</div><div>>D</div><div>IN2</div></div>	Lệnh so sánh giá trị của hai DoubleWord IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1 > IN2 là đúng.	ID, QD, MD, VD, SMD, SD, LD, AC, Constant, *VD, *AC, *LD	Double Word
ADW>				
ODW>				
LDDW>=	<div><div>IN1</div><div>>=D</div><div>IN2</div></div>	Lệnh so sánh giá trị của hai DoubleWord IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1 >= IN2 là đúng.	ID, QD, MD, VD, SMD, SD, LD, AC, Constant, *VD, *AC, *LD	Double Word
ADW>=				
ODW>=				
LDDW<	<div><div>IN1</div><div><D</div><div>IN2</div></div>	Lệnh so sánh giá trị của hai DoubleWord IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1 < IN2 là đúng.	ID, QD, MD, VD, SMD, SD, LD, AC, Constant, *VD, *AC, *LD	Double Word
ADW<				
ODW<				
LDDW<=	<div><div>IN1</div><div><=D</div><div>IN2</div></div>	Lệnh so sánh giá trị của hai DoubleWord IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1 <= IN2 là đúng.	ID, QD, MD, VD, SMD, SD, LD, AC, Constant, *VD, *AC, *LD	Double Word
ADW<=				
ODW<=				
COMPARE REAL				
LDR=	<div><div>IN1</div><div>=R</div><div>IN2</div></div>	Lệnh so sánh giá trị của hai số thực IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1 = IN2 là đúng.	ID, QD, MD, VD, SMD, SD, LD, AC, Constant, *VD, *AC, *LD	Real
AR=				
OR=				
LDR<	<div><div>IN1</div><div><R</div><div>IN2</div></div>	Lệnh so sánh giá trị của hai số thực IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1 < IN2 là đúng.	ID, QD, MD, VD, SMD, SD, LD, AC, Constant, *VD, *AC, *LD	Real
AR<				
OR<				
LDR>	<div><div>IN1</div><div>>R</div><div>IN2</div></div>	Lệnh so sánh giá trị của hai số thực IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1 > IN2 là đúng.	ID, QD, MD, VD, SMD, SD, LD, AC, Constant, *VD, *AC, *LD	Real
AR>				
OR>				
LDR>=	<div><div>IN1</div><div>>=R</div><div>IN2</div></div>	Lệnh so sánh giá trị của hai số thực IN1 và IN2. Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1 >= IN2 là đúng.	ID, QD, MD, VD, SMD, SD, LD, AC, Constant, *VD, *AC, *LD	Real
AR>=				
OR>=				
LDR<	<div><div>IN1</div><div><R</div><div>IN2</div></div>	Lệnh so sánh giá trị của hai số thực IN1 và IN2.	ID, QD, MD, VD, SMD, SD, LD, AC,	Real

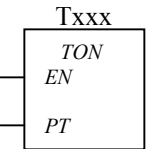
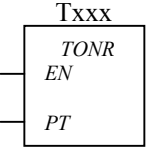
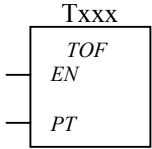
AR<		Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1 < IN2 là đúng	Constant, *VD, *AC, *LD	
OR<				
LDR<=		Lệnh so sánh giá trị của hai số thực IN1 và IN2.	ID, QD, MD, VD, SMD, SD, LD, AC, Constant, *VD, *AC, *LD	Real
AR<=		Trạng thái tiếp điểm là đóng khi lệnh so sánh IN1 <= IN2 là đúng		
OR<=				



Hình 3.22: Ví dụ minh họa lệnh so sánh trong chương trình LAD, FBD và STL

3. SIMATIC Timer Instructions:

STL	LAD	Mô tả (Description)	Toán hạng (Operands)	Kiểu dữ liệu (Data Types)
On Delay Timer (TON)				
		Đây là lệnh đếm thời gian hoạt khi tín hiệu	Txxx: Constant	word
			IN : power flow	bool

TON Txxx, PT		<p>EN là ON.</p> <p>Khi giá trị đếm tức thời trong thanh ghi CT \geq giá trị đặt trước trong thanh ghi PT thì bit trạng thái Txxx của bộ Timer là ON.</p> <p>Giá trị đếm tức thời trong thanh ghi CT = 0 và bit trạng thái về off khi tín hiệu ở đầu vào là off. Ngược lại với bộ TON, thanh ghi CV và bit trạng thái vẫn giữ nguyên trừ khi có lệnh Reset bộ TONR. Ngoài ra có thể sử dụng lệnh Reset để xóa thanh ghi tức thời cũng như bit trạng thái của bộ TON.</p>	<p>PT: IW, QW, MW, SMW, VW, LW, SW, AIW, T, C, AC, Constant, *VD, *AC, *LD</p>	INT
TONR Txxx, PT		<p>Ta có thể sử dụng toán hạng Word (INT) tương ứng với lệnh INT hay toán hạng bit tương ứng với bit trạng thái.</p>		
TOF Txxx, PT		<p>Trạng thái của bit Txxx có cùng trạng thái với tín hiệu tại chân EN ở đầu vào, tại thời điểm này giá trị trong thanh ghi CT= 0. Tại thời điểm</p>		

		khi có sườn xuống của tín hiệu ở chân EN giá trị trong thanh ghi CV bắt đầu tăng dần đến khi CT = PT bit Txxx xuống mức thấp và CT giữ nguyên giá trị này cho đến khi có tín hiệu (mức cao mới kích vào chân EN). Có thể xoá CT và Txxx bằng lệnh Reset.		
--	--	---	--	--

Bảng 3.2: Số Timer và độ phân giải

Timer Type	Resolution in milliseconds (ms)	Maximum Value in seconds (s)	Timer Number
TONR (retentive)	1 ms	32.767 s (0.546 min.)	T0, T64
	10 ms	327.67 s (0.546 min.)	T1 to T4, T65 to T68
	100 ms	3276.7 s (0.546 min.)	T5 to T31, T69 to T95
TON, TOF (non-retentive)	1 ms	32.767 s (0.546 min.)	T32, T96
	10 ms	327.67 s (0.546 min.)	T33 to T36, T97 to T100
	100 ms	3276.7 s (0.546 min.)	T37 to T63, T101 to T255

Note: Không thể cùng một lúc sử dụng cả 2 bộ TON và TOF cho cùng 1 địa chỉ (ví dụ T37).

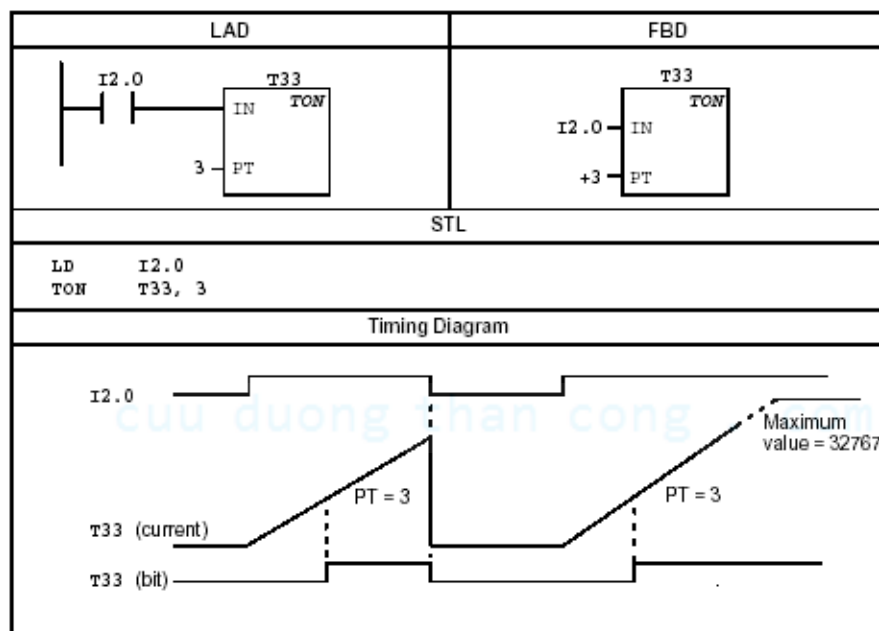
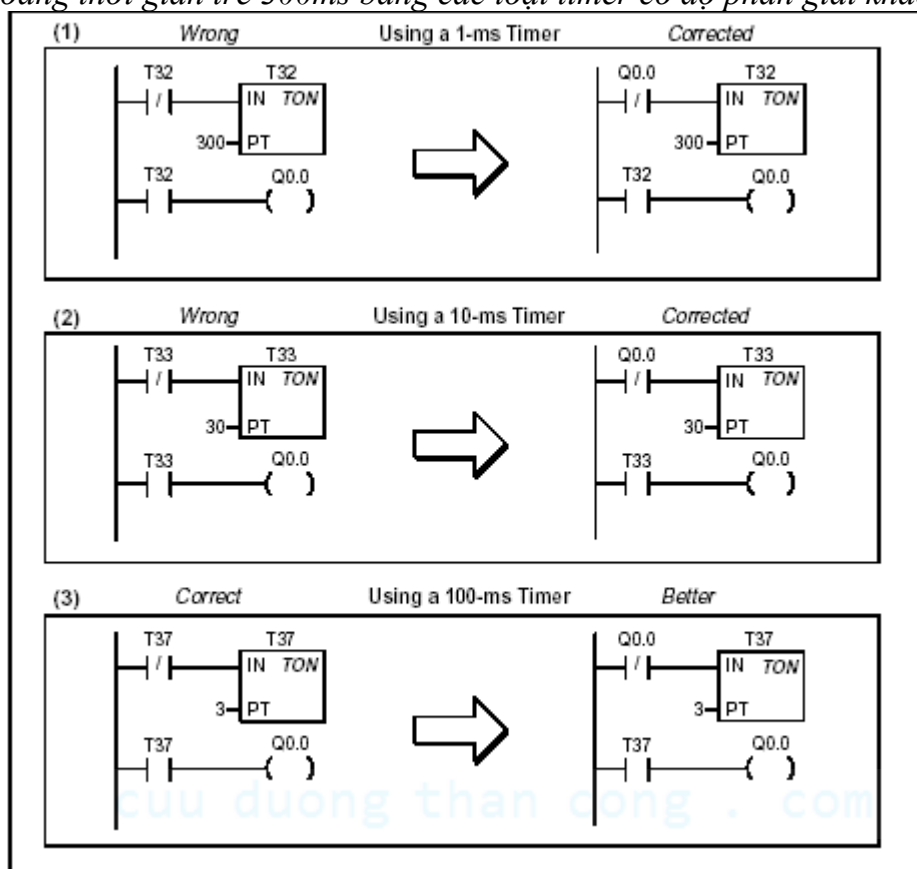
Bảng 3.3: Giá trị đặt tối đa cho từng loại và trạng thái làm việc của các loại Timer

Timer Type	Current >= Preset	Enabling Input ON	Enabling Input OFF	Power Cycle/ First Scan
TON	Timer bit ON, Current continues counting to 32,767	Current value counts time	Timer bit OFF, Current value = 0	Timer bit OFF, Current value = 0
TONR	Timer bit ON, Current continues counting to 32,767	Current value counts time	Timer bit and current value maintain last state	Timer bit OFF, Current value may be maintained ¹
TOF	Timer bit OFF, Current = Preset, stops counting	Timer bit ON, Current value = 0	Timer counts after ON to OFF transition	Timer bit OFF, Current value = 0

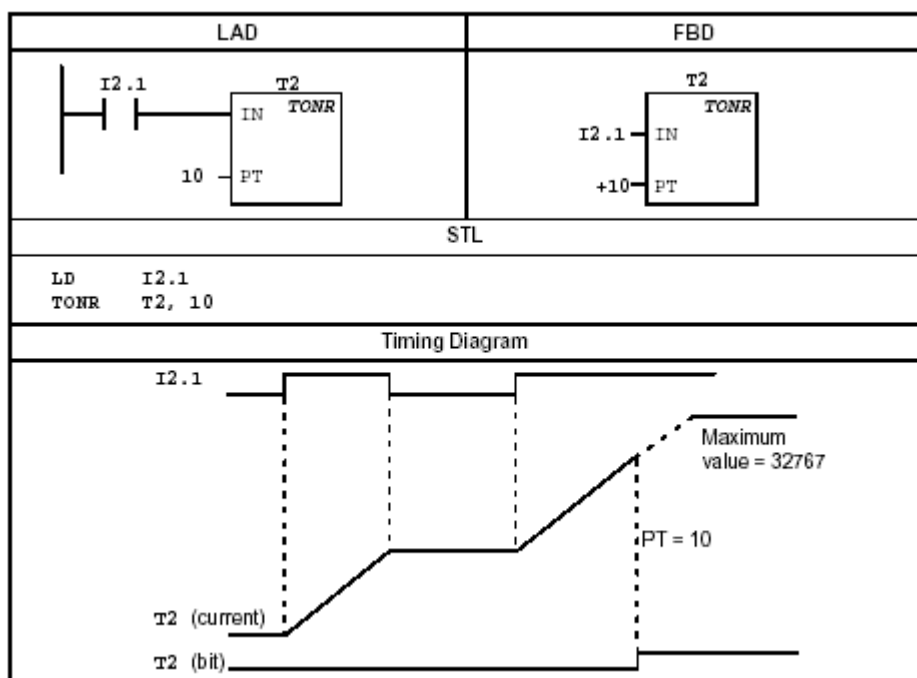
¹ The retentive timer current value can be selected for retention through a power cycle. See Section 5.3 for information about memory retention for the S7-200 CPU.

Việc sử dụng tiếp điểm thường đóng Q0.0 bên dưới để đảm làm tín hiệu đầu vào cho Timer đảm bảo cho Q0.0 sẽ có giá trị logic bằng 1 trong một vòng quét ở mỗi thời điểm mà giá trị đếm tức thời của bộ Timer đạt giá trị đặt trước PT.

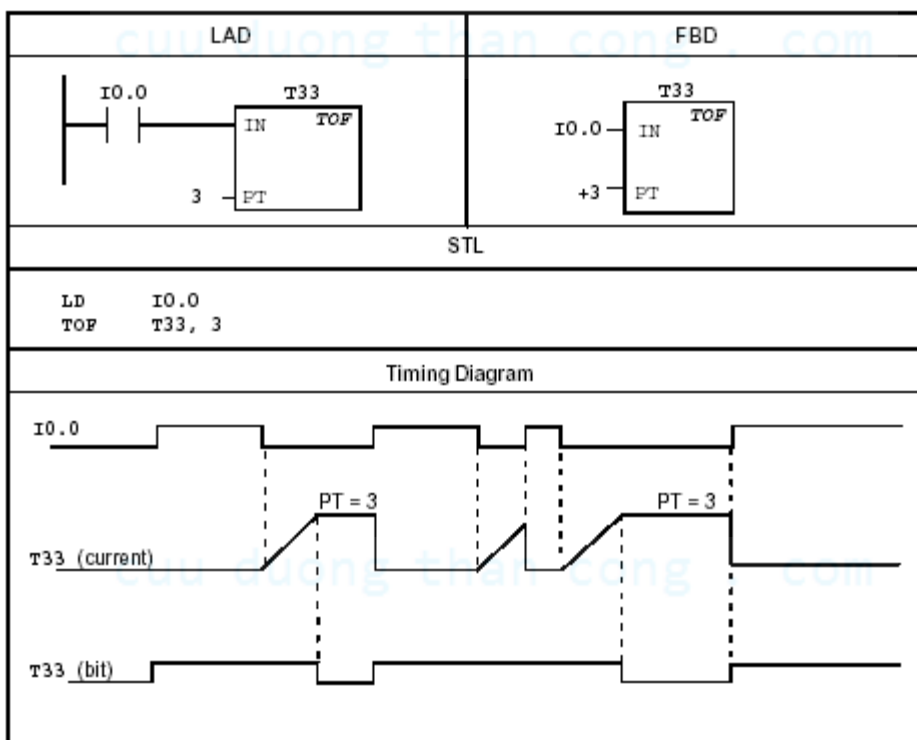
Tạo khoảng thời gian trễ 300ms bằng các loại timer có độ phân giải khác nhau:



Hình 3.23: Ví dụ cách sử dụng bộ TON

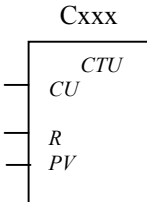
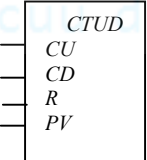
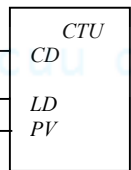


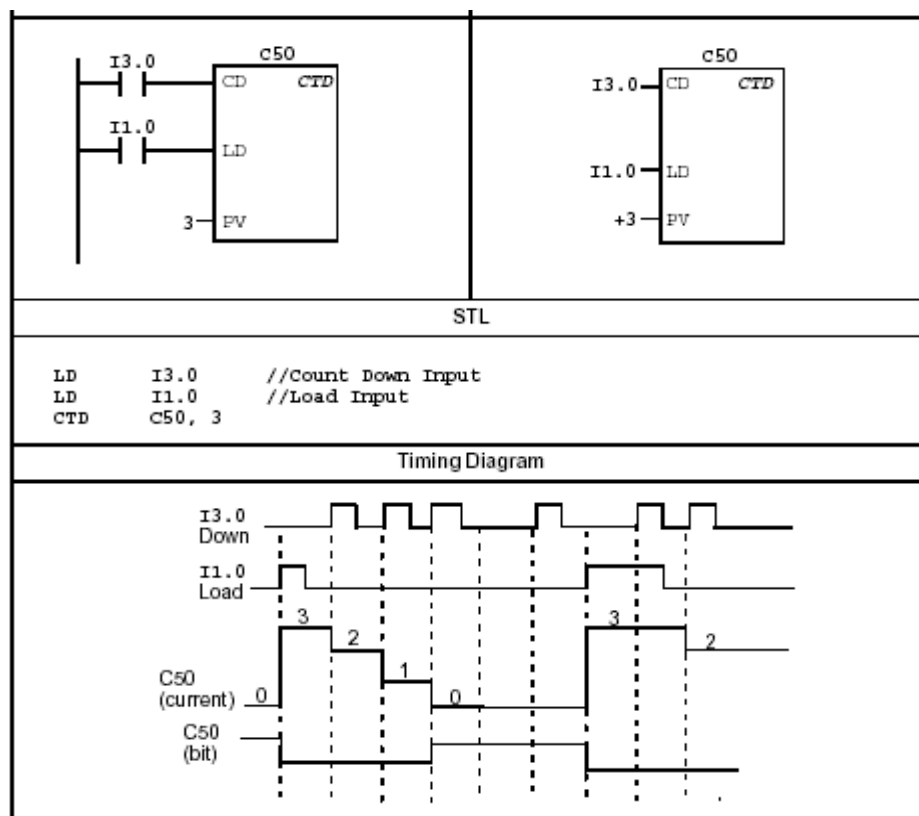
Hình 3.24: Ví dụ cách sử dụng bộ TONR



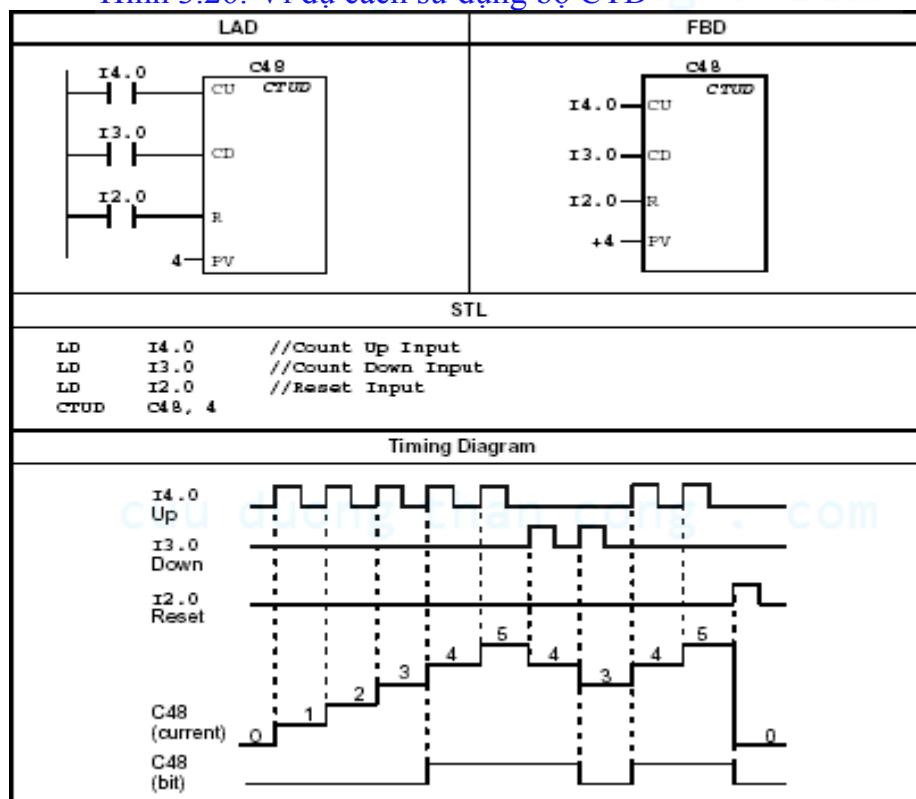
Hình 3.25: Ví dụ cách sử dụng bộ TOF

4. SIMATIC Counter Instructions (*Count Up, Count Up Down, Count Down*):

STL	LAD	Mô tả Description	Toán hạng Operands	Kiểu dữ liệu Data Types
CTU Cxxx, PV		<p>Khai báo bộ đếm tiến theo sườn lên của tín hiệu đầu vào CU. Khi giá trị đếm tức thời C-Word lớn hơn hoặc bằng giá trị đặt trước PV, thì bit trạng thái Cxxx có giá trị bằng 1. Bộ đếm được Reset khi R có giá trị logic bằng 1. Bộ đếm ngừng đếm khi giá trị đếm đạt giá trị cực đại 32767.</p>	Cxxx: Constant	word
			EU, R : power flow.	bool
			PT: IW, QW, MW, SMW, VW, LW, SW, AIW, T, C, AC, Constant, *VD, *AC, *LD	INT
CTUD Cxxx, PV		<p>Khai báo bộ đếm tiến/lùi; đếm tiến theo sườn lên của tín hiệu đầu vào CU, đếm lùi theo sườn lên của tín hiệu đầu vào CD. Khi giá trị đếm tức thời C-Word lớn hơn hoặc bằng giá trị đặt trước PV, thì bit trạng thái Cxxx có giá trị bằng 1. Bộ đếm được Reset khi R có giá trị logic bằng 1. Bộ đếm ngừng đếm tiến khi giá trị đếm đạt giá trị cực đại 32767. Bộ đếm ngừng đếm lùi khi giá trị đếm đạt giá trị cực đại -32768. CTUD reset khi đầu vào R có giá trị logic bằng 1.</p>	Cxxx: Constant	word
			EU, ED, R : power flow.	bool
			PT: IW, QW, MW, SMW, VW, LW, SW, AIW, T, C, AC, Constant, *VD, *AC, *LD	INT
CTD Cxxx, PV		<p>Khai báo bộ đếm lùi theo sườn lên của tín hiệu đầu vào CD. Khi có sườn lên tại LD, giá trị đặt trước PV được load vào thanh tức thời C-Word. Mỗi khi có sườn lên tại CD, giá trị trong C-Word giảm đi 1 đơn vị, cho đến khi C-Word = 0 thì bit trạng thái C_{bit} = 1.</p>		

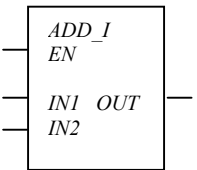
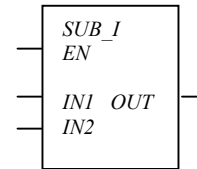
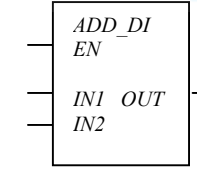
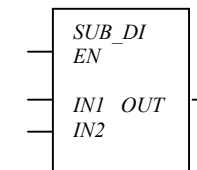
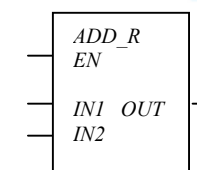


Hình 3.26: Ví dụ cách sử dụng bộ CTD

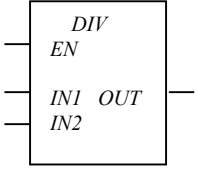
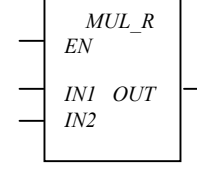
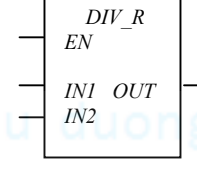
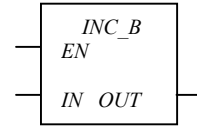
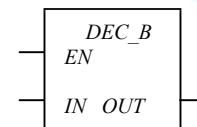


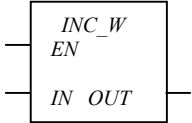
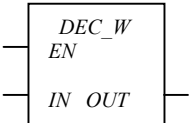
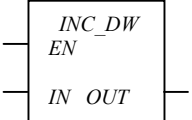
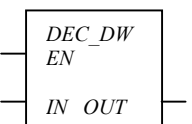
Hình 3.27: Ví dụ cách sử dụng bộ CTUD

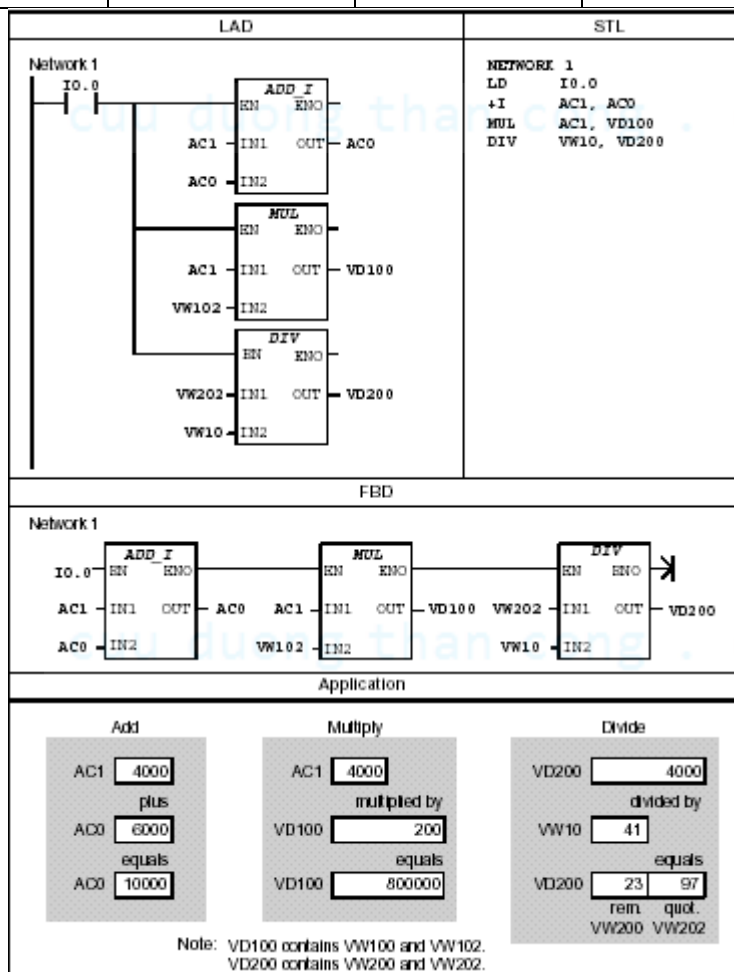
5. SIMATIC Integer Math Instructions:

STL	LAD	Mô tả Description	Toán hạng Operands	Kiểu dữ liệu Data Types
Add Integer and Subtract Integer				
MOVW IN1, OUT +I IN2, OUT hoặc +I IN1, IN2		Lệnh cộng hai số nguyên 16 bit IN1 + IN2 kết quả chứa trong OUT (16 bit)	IN1, IN2: IW, QW, MW, SMW, VW, LW, SW, AIW, T, C, AC, Constant, *VD, *AC, *LD OUT: IW, QW, MW, SMW, VW, LW, SW, T, C, AC, *VD, *AC, *LD	INT
MOVW IN1, OUT -I IN2, OUT hoặc -I IN1, IN2		Lệnh trừ hai số nguyên 16 bit IN1 - IN2 kết quả chứa trong OUT (16 bit)	IN1, IN2: IW, QW, MW, SMW, VW, LW, SW, T, C, AC, *VD, *AC, *LD OUT: IW, QW, MW, SMW, VW, LW, SW, T, C, AC, *VD, *AC, *LD	INT
Add Double Integer and Subtract Double Integer				
MOVD IN1, OUT +D IN2, OUT hoặc +D IN1, IN2		Lệnh cộng hai số nguyên 32 bit IN1 + IN2 kết quả chứa trong OUT (32 bit)	IN1, IN2: ID, QD, MD, VD, SMD, SD, LD, HC, AC, Constant, *VD, *AC, *LD OUT: ID, QD, MD, VD, SMD, SD, LD, AC, *VD, *AC, *LD	DINT
MOVD IN1, OUT -D IN2, OUT hoặc -D IN1, IN2		Lệnh trừ hai số nguyên 32 bit IN1 - IN2 kết quả chứa trong OUT (32 bit)	IN1, IN2: ID, QD, MD, VD, SMD, SD, LD, AC, *VD, *AC, *LD OUT: ID, QD, MD, VD, SMD, SD, LD, AC, *VD, *AC, *LD	DINT
Add Real and Subtract Real				
MOVR IN1, OUT +R IN2, OUT hoặc +R IN1, IN2		Lệnh cộng hai số thực 32 bit IN1 + IN2 kết quả chứa trong OUT (32 bit)	IN1, IN2: ID, QD, MD, VD, SMD, SD, LD, HC, AC, Constant, *VD, *AC, *LD OUT: ID, QD, MD, VD, SMD, SD, LD, AC, *VD, *AC, *LD	Real
MOVR IN1, OUT -R IN2, OUT		Lệnh trừ hai số thực 32 bit IN1	IN1, IN2: ID, QD, MD, VD, SMD, SD, LD,	Real

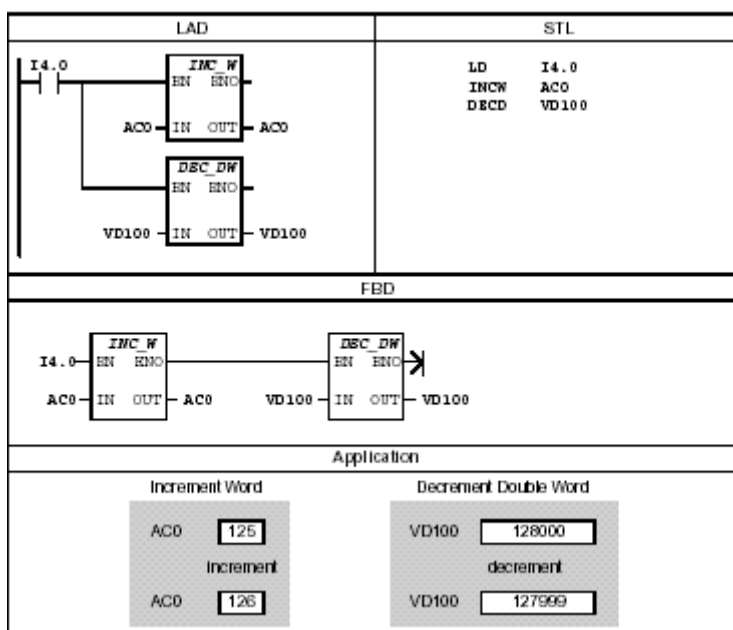
<div>hoặc -R IN1, IN2</div>	<div><div>SUB_R EN</div><div>IN1 OUT IN2</div></div>	<div>+ IN2 kết quả chứa trong OUT (32 bit)</div>	<div>HC,AC, Constant, *VD, *AC, *LD</div> <div>OUT: ID, QD, MD, VD, SMD, SD, LD,AC, *VD, *AC, *LD</div>	
<div>Multiply Integer and Divide Integer</div>				
<div>MOVW IN1, OUT *I IN2, OUT hoặc *I IN1, IN2</div>	<div><div>MULL_I EN</div><div>IN1 OUT IN2</div></div>	<div>Lệnh nhân hai số nguyên 16 bit IN1*IN2 kết quả chứa trong OUT (16 bit)</div>	<div>IN1, IN2: IW, QW, MW, SMW, VW, LW, SW, AIW, T, C, AC, Constant, *VD, *AC, *LD</div> <div>OUT: IW, QW, MW, SMW, VW, LW, SW, T, C, AC, *VD, *AC, *LD</div>	<div>INT</div>
<div>MOVW IN1, UT /I IN2, OUT hoặc /I IN1, IN2</div>	<div><div>DIV_I EN</div><div>IN1 OUT IN2</div></div>	<div>Lệnh chia hai số nguyên 16 bit IN1/IN2 kết quả chứa trong OUT (16 bit)</div>		
<div>Multiply Double Integer and Divide Double Integer</div>				
<div>MOVD IN1, OUT *D IN2, OUT hoặc *D IN1, IN2</div>	<div><div>MUL_DI EN</div><div>IN1 OUT IN2</div></div>	<div>Lệnh nhân hai số nguyên 32 bit IN1*IN2 kết quả chứa trong OUT (32 bit)</div>	<div>IN1, IN2: ID, QD, MD, VD, SMD, SD, LD, HC,AC, Constant, *VD, *AC, *LD</div> <div>OUT: ID, QD, MD, VD, SMD, SD, LD,AC, *VD, *AC, *LD</div>	<div>DINT</div>
<div>MOVD IN1, OUT /D IN2, OUT hoặc /D IN1, IN2</div>	<div><div>DIV_DI EN</div><div>IN1 OUT IN2</div></div>	<div>Lệnh chia hai số nguyên 32 bit IN1/IN2 kết quả chứa trong OUT (32 bit)</div>		
<div>Multiply Integer to Double Double Integer and Divide Integer to Double Double Integer</div>				
<div>MOVW IN1, OUT MUL IN2, OUT hoặc MUL IN1, IN2</div>	<div><div>MUL EN</div><div>IN1 OUT IN2</div></div>	<div>Lệnh nhân hai số nguyên 16 bit IN1*IN2 kết quả chứa trong OUT (32 bit)</div>	<div>IN1, IN2: IW, QW, MW, SMW, VW, LW, SW, AIW, T, C, AC, Constant, *VD, *AC, *LD</div> <div>OUT: ID, QD, MD, VD, SMD, SD, LD,AC, *VD, *AC, *LD</div>	<div>INT</div> <div>DINT</div>

MOVW IN1, OUT DIV IN2, OUT hoặc DIV IN1, IN2		Lệnh chia hai số nguyên 16 bit IN1*IN2 kết quả chứa trong OUT (32 bit)	IN1, IN2: IW, QW, MW, SMW, VW, LW, SW, AIW, T, C, AC, Constant, *VD, *AC, *LD	INT
			OUT: ID, QD, MD, VD, SMD, SD, LD, AC, *VD, *AC, *LD	DINT
Miltiply Real and Divide Real				
MOVR IN1, OUT *R IN2, OUT hoặc *R IN1, IN2		Lệnh nhân hai số thực 32 bit IN1*IN2 kết quả chứa trong OUT (32 bit)	IN1, IN2: IW, QW, MW, SMW, VW, LW, SW, AIW, T, C, AC, Constant, *VD, *AC, *LD OUT: ID, QD, MD, VD, SMD, SD, LD, AC, *VD, *AC, *LD	Real
MOVR IN1, OUT /R IN2, OUT hoặc /R IN1, IN2		Lệnh chia hai số thực 32 bit IN1/IN2 kết quả chứa trong OUT (32 bit)		
Những lệnh này làm đơn giản hoá các vòng điều khiển bên trong chương trình hoặc là các quá trình lập. Trong LAD hay trong STL các lệnh tăng hoặc giảm đều làm việc với các toán hạng có kiểu Byte, từ đơn, kiểu từ kép theo nguyên tắc cộng hoặc trừ toán hạng với số nguyên 1. Để tiết kiệm ô nhớ ta có thể sử dụng đầu vào đồng thời làm đầu ra.				
Increment Byte and Decrement Byte				
INCB OUT		Mô tả ở trên.	IN: IB, QB, MB, VB, SMB, SB, LB, AC, Constant, *VD, *AC, *LD OUT: IB, QB, MB, VB, SMB, SB, LB, AC, *VD, *AC, *LD	Byte
DECB OUT				
Increment Word and Decrement Word				

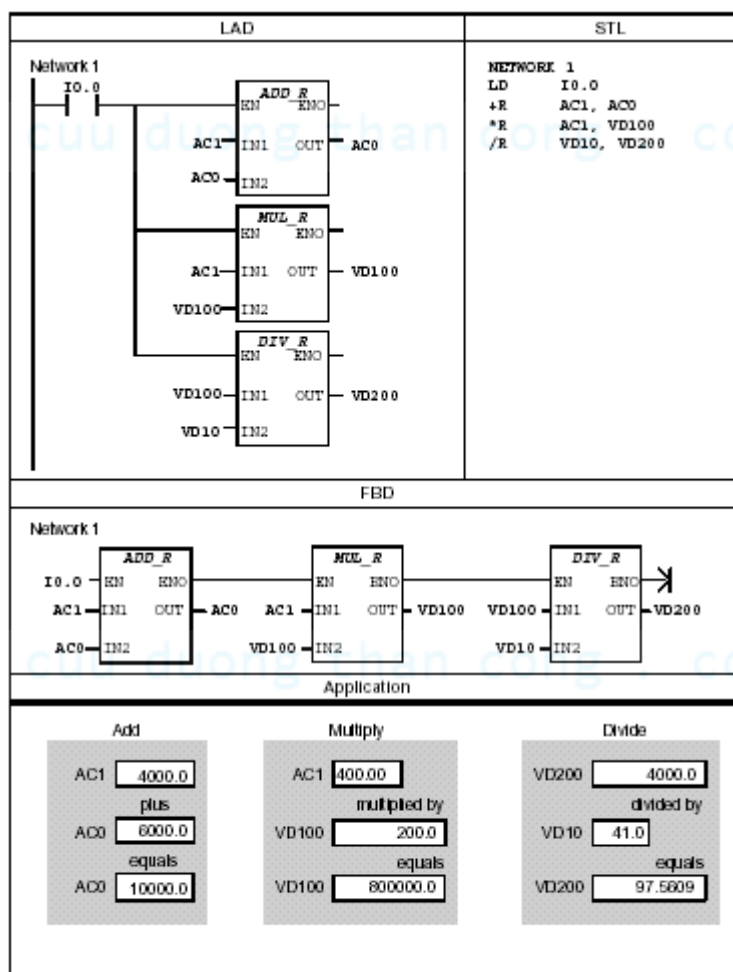
INCW OUT		Mô tả ở trên	IN: IW, QW, MW, SMW, VW, LW, SW, AIW, T, C, AC, Constant, *VD, *AC, *LD OUT: IW, QW, MW, SMW, VW, LW, SW, AIW, T, C, AC, *VD, *AC, *LD	INT
DECW OUT				
Increment Double Word and Decrement Double Word				
INCD OUT		Mô tả ở trên	IN: ID, QD, MD, VD, SMD, SD, LD, HC,AC, Constant, *VD, *AC, *LD OUT: ID, QD, MD, VD, SMD, SD, LD, HC,AC, *VD, *AC, *LD	DINT
DECB OUT				



Hình 3.28: Ví dụ về cách sử dụng lệnh MIL, DIV



Hình 3.29: Ví dụ về cách sử dụng lệnh INC, DEC



Hình 3.30: Ví dụ về cách sử dụng lệnh ADD, MUL, DIV với số thực