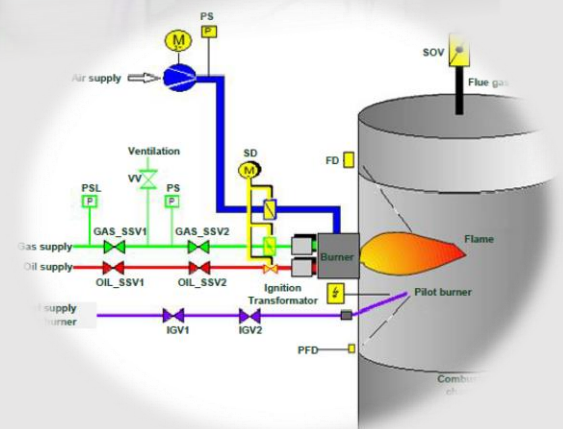
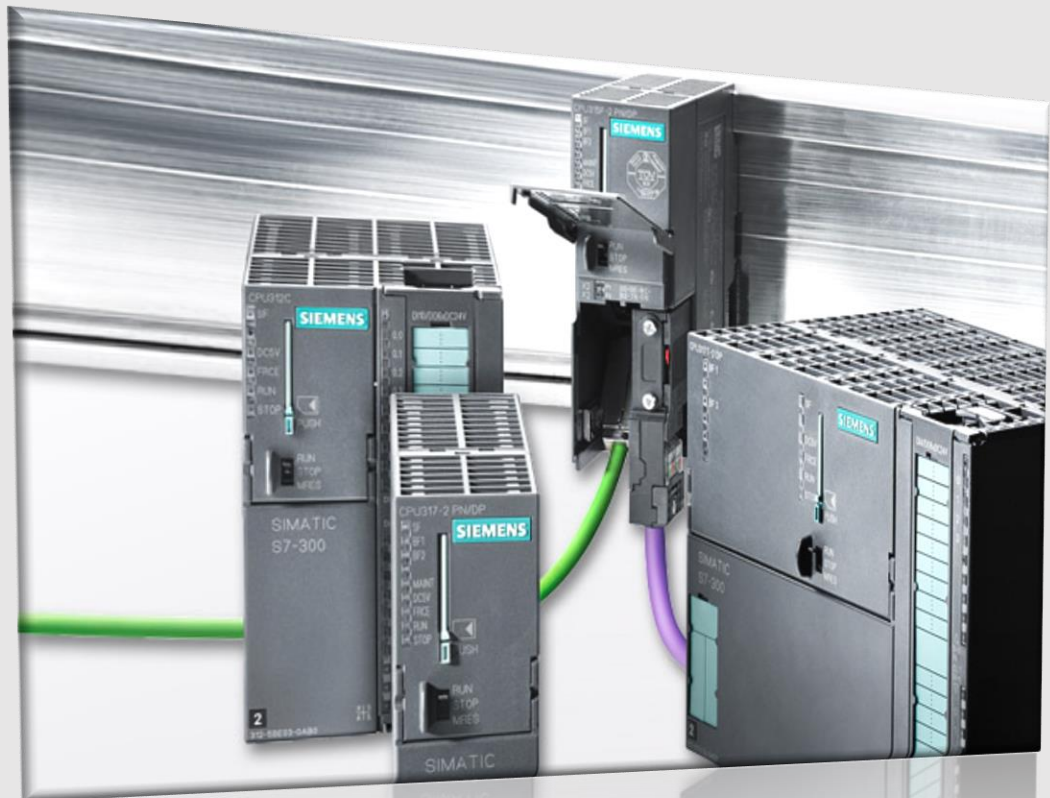


PLC S7-300.400

LEVEL(I)



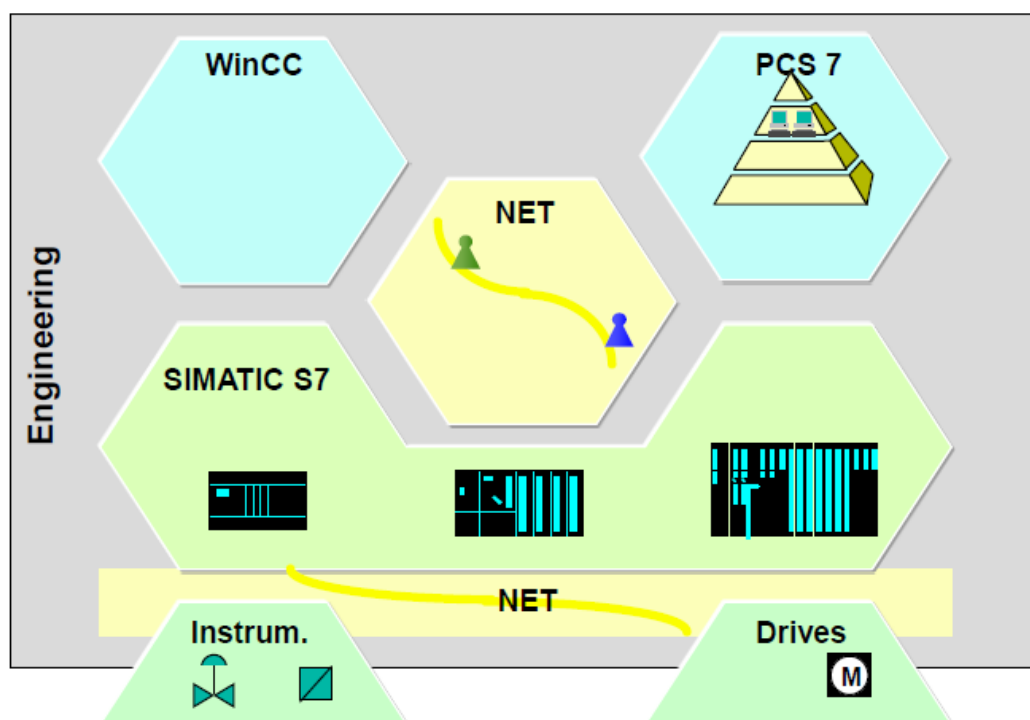
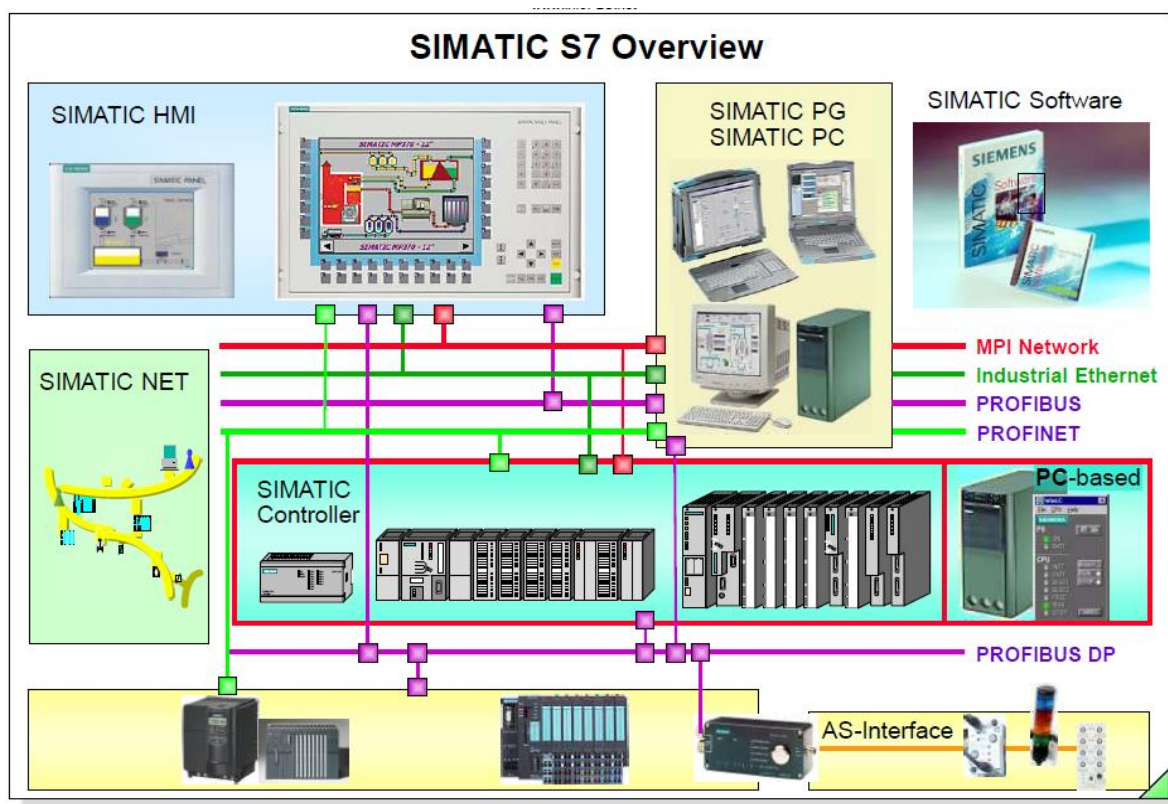
مهندس نازنین زنجیریان

SIEMENS

سخت افزار PLC S7-300,400

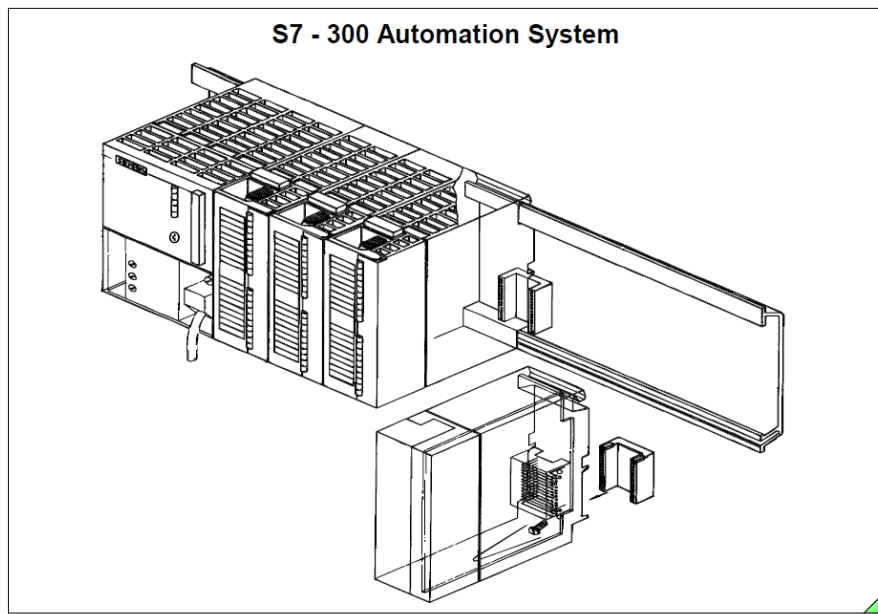


محصولات SIMATIC



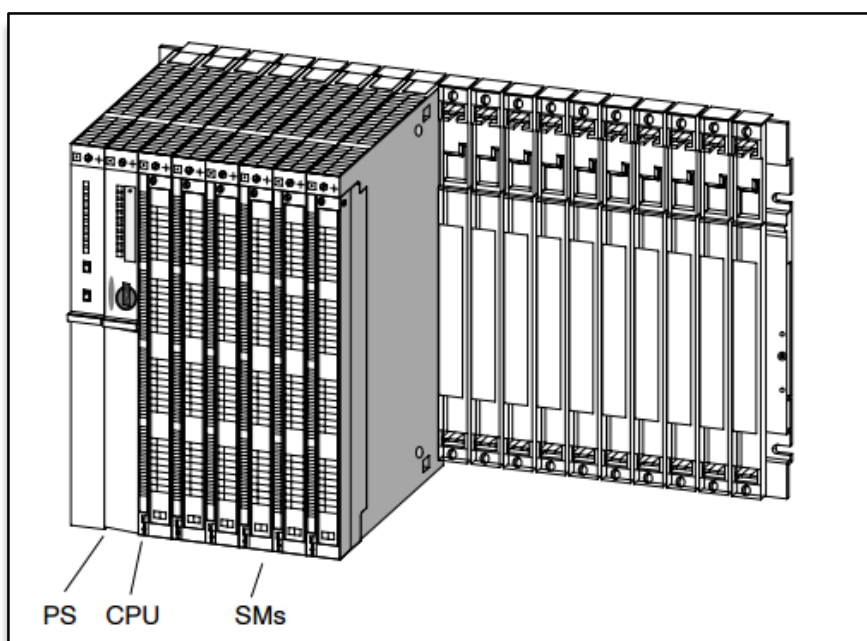
نحوه نصب ماژول های S7-300

در PLC S7-300، ماژول ها توسط Bus Connector به صورت سری به هم متصل می شوند.

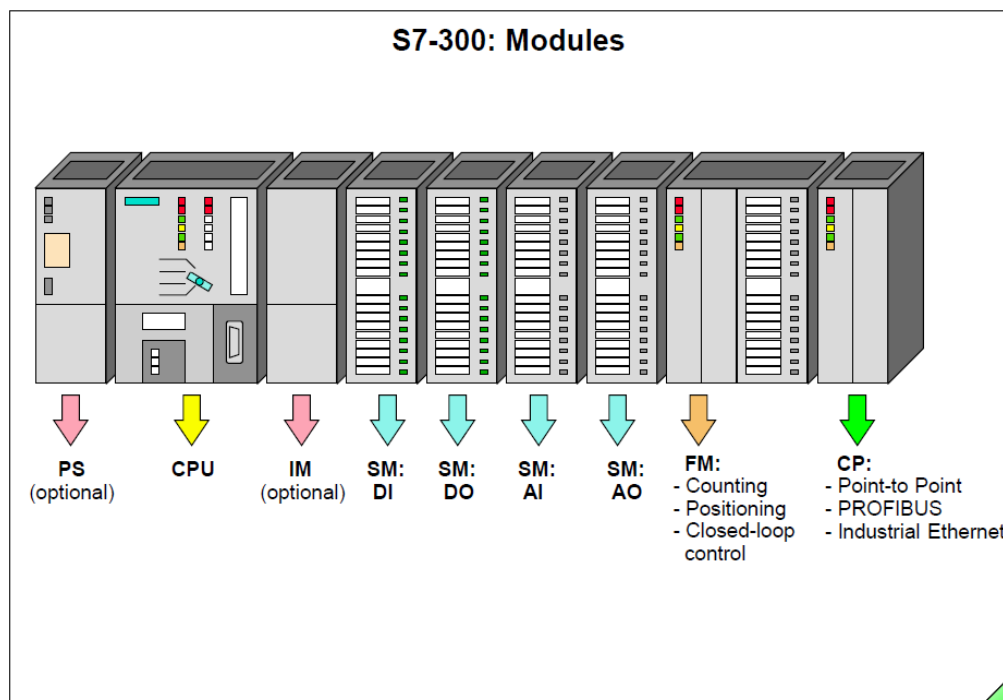


نحوه ی قرارگرفتن ماژول های S7-400

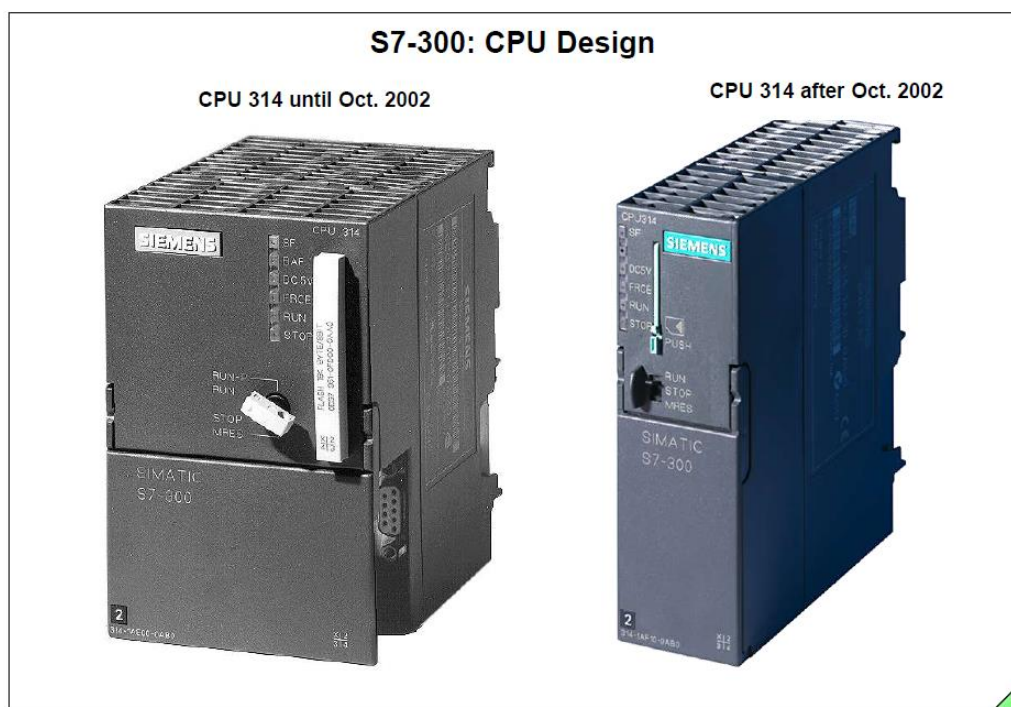
در S7-400 تمامی ماژول ها بر روی رک قرار می گیرند. در واقع ارتباط بین CPU با سایر ماژول ها توسط رک برقرار می شود. در این ساختار تغذیه ماژول ها نیز در رک وجود دارد.



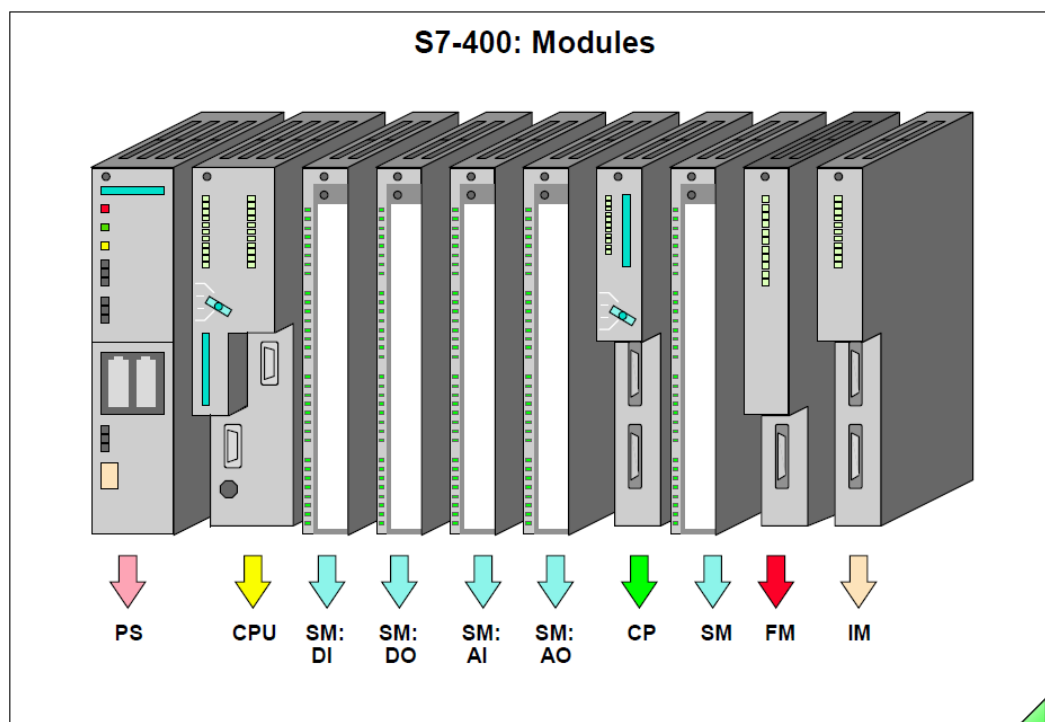
ماژول های S7-300



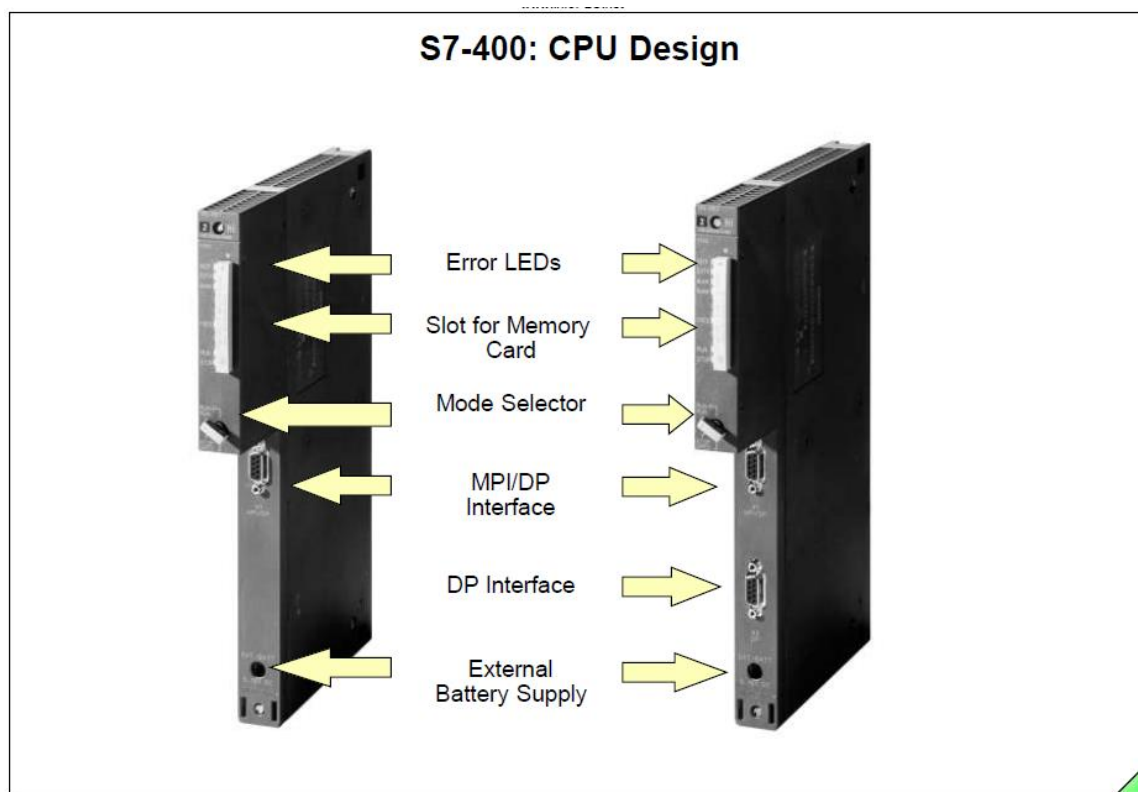
CPU های قدیم و جدید S7-300



ماژول های S7-400

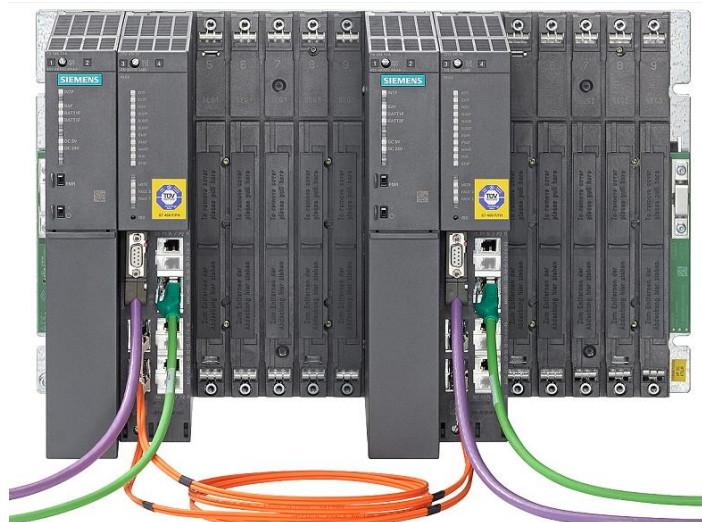


ساختار CPU در S7-400

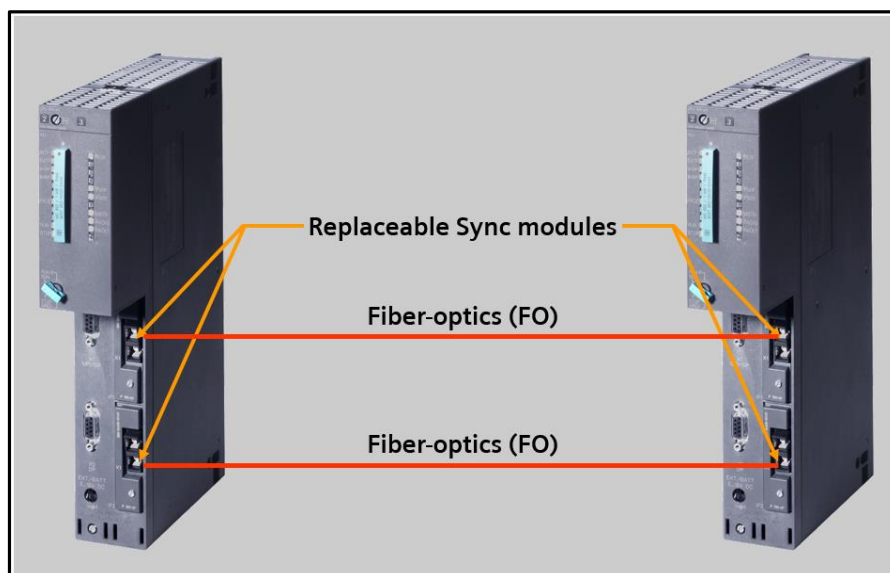


ساختار سیستم H در S7-400

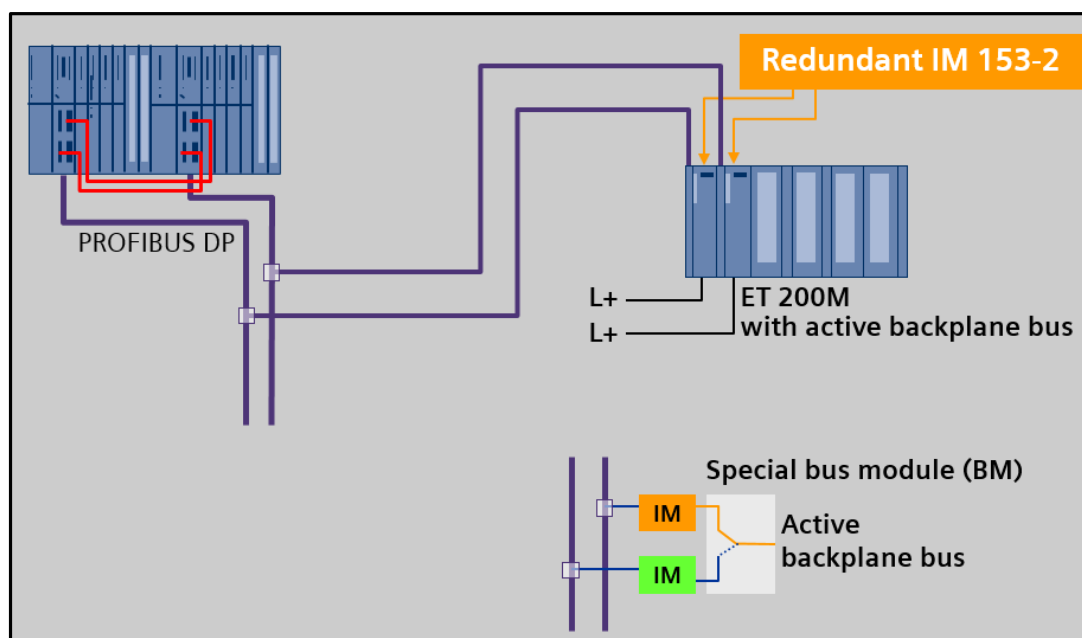
در این ساختار از دو CPU در یک رک استفاده می شود. یک CPU در حال اجرای برنامه و کنترل پروسه می باشد. این کنترلر Master می باشد. زمانی که بر روی این CPU خطایی رخ دهد، سیستم رزرو یا Standby در زمان کوتاهی وارد مدار میشود و کار کنترل را انجام میدهد. این زمان در سیستم Redundant سخت افزاری بسیار کوتاه و در حد چند میلی ثانیه می باشد. سیستم افزونگی سخت افزاری را می توان در برخی از CPU های S7-400 که دارای پسوند H می باشند، پیاده سازی نمود. ایستگاه I/O در این حالت می تواند یک ایستگاه ET200M همانند ET200M با قابلیت اتصال به سیستم Redundant باشد.



در این ساختار دو CPU توسط فیبر به یکدیگر متصل میشوند. عملیات سنکرون سازی بین دو CPU توسط ماژول های سنکرون سازی یا Sync انجام میشود.



اتصال سیستم Redundant به ایستگاه ET200M



آشنایی با ماژول پاور در S7-300

کارت پاور زیمنس تیپ S7-300 دارای تغذیه ی ورودی 220VAC و خروجی 24VDC می باشد. جهت تغذیه ماژول های S7-300 الزامی به استفاده از منبع تغذیه با برند زیمنس نمی باشد و منبع تغذیه با هر برندی با داشتن شرایط لازم برای تغذیه کارت می تواند مورد استفاده قرار گیرد. در صورتی که بخواهیم منبع تغذیه روی ریل استاندارد S7، در کنار سایر ماژول ها قرار گیرد، می توان از ماژول پاور زیمنس و تیپ S7-300 استفاده می کنیم. روی این منبع تغذیه سوئیچ ON/OFF برای روشن و خاموش کردن پاور تعبیه شده است. همچنین LED مربوط به نمایش صحت سطح سیگنال خروجی با رنگ سبز وجود دارد که در شرایط کار عادی روشن می باشد. ترمینال های ورودی و خروجی کارت پاور در شکل زیر نشان داده شده است.



آشنایی با CPU های خانواده S7-300

SIMATIC S7-300 Compact-CPU

S7-300 CPUs module overview

Overview

- ✓ Description
- ✓ SIMATIC S7-300 CPU 312C
- ✓ SIMATIC S7-300 CPU 313C
- ✓ SIMATIC S7-300 CPU 313C-2 PtP
- ✓ SIMATIC S7-300 CPU 313C-2 DP
- ✓ SIMATIC S7-300 CPU 314C-2 PtP
- ✓ SIMATIC S7-300 CPU 314C-2 DP
- ✓ SIMATIC S7-300 CPU 314C-2 PN/DP

-مشخصات CPU های کامپکت

❖ دارای پسوند C می باشند.

❖ دارای تعدادی I/O به صورت Onboard می باشند.

❖ در این CPU ها نیز امکان توسعه وجود دارد.

SIMATIC S7-300 CPU 312C

For small applications with high requirements in terms of processing power

- MPI-Schnittstelle onboard
- Technological functions:
 - Counting
 - Frequency measurement
 - Pulse width modulation
 - Pulse generator
- 10 digital inputs
- 6 digital outputs



SIMATIC S7-300 CPU 313C

CPU for installations with high requirements in terms of processing power and response time.

- MPI interface onboard
- Technological functions:
 - Counting
 - Closed loop control
 - Frequency measurement
 - Pulse width modulation
 - Pulse generator
- 24 digital inputs
- 16 digital outputs
- 4 analog inputs
- 2 analog outputs



SIMATIC S7-300 CPU 313C-2 PtP

CPU for installations with high requirements in terms of processing power and response time.

- MPI- interface onboard
- Point to point link interface
- Technological functions:
 - Counting
 - Closed loop control
 - Frequency measurement
 - Pulse width modulation
 - Pulse generator
- 16 digital inputs
- 16 digital outputs



SIMATIC S7-300 CPU 313C-2 DP

CPU for installations with high requirements in terms of processing power and response time.

- MPI interface onboard
- PROFIBUS DP master/slave interface
- Technological functions:
 - Counting
 - Closed loop control
 - Frequency measurement
 - Pulse width modulation
 - Pulse generator
- 16 digital inputs
- 16 digital outputs



SIMATIC S7-300 CPU 314C-2 PtP

CPU for installations with high requirements in terms of processing power and response time.

- MPI interface onboard
- Point to point link interface
- Technological functions:
 - Counting
 - Closed loop control
 - Frequency measurement
 - Pulse width modulation
 - Pulse generator
 - Positioning
- 24 digital inputs
- 16 digital outputs
- 4 analog inputs
- 2 analog outputs



SIMATIC S7-300 CPU 314C-2 PN/DP

CPU for machines with high demands of computing power and reaction rate

- combined MPI-PROFIBUS DP-interface onboard
- 2 PROFINET Ports for line structures
- Technology functions:
 - Counting
 - Controlling
 - Detecting frequencies
 - Pulse width modulation
 - Pulse generator
 - Positioning
- 24 digital inputs
- 16 digital outputs
- 4 analog inputs
- 2 analog outputs



در CPU های کامپکت:

Counting : یعنی مجهز بودن CPU به ورودی های HSC (شمارنده های سرعت بالا) برای شمارش

پالس های فرکانس بالا

Detecting Frequency : یعنی CPU مجهز به فانکشن هایی جهت اندازه گیری فرکانس پالس می باشد.

Pulse Width Modulation : یعنی CPU مجهز به فانکشن، جهت تولید پالس PWM در خروجی

می باشد.

Pulse Generator : یعنی CPU مجهز به خروجی پالس با فرکانس بالا برای کنترل موتور پله ای یا سرو

می باشد.

Positioning : یعنی CPU مجهز به فانکشن جهت استفاده در کاربردهای کنترل موقعیت می باشد.

Positioning می تواند به صورت آنالوگ یا دیجیتال باشد.

در S7-300 همانطور که بیان شد، CPU های کامپکت در Order های مختلفی در دسترس می باشد. CPU

314C-2PN/DP یکی از کامل ترین CPU های کامپکت می باشد.

Technical data: Compact CPUs

N = Product innovations

NEW

CPU	CPU 312C ¹⁾	CPU 313C ¹⁾	CPU 313C-2 PtP	CPU 313C-2 DP ¹⁾	CPU 314C-2 PtP ¹⁾	CPU 314C-2 DP ¹⁾	CPU 314C-2 PN/DP
Dimensions (mm)	80 x 125 x 130	120 x 125 x 130	80 x 125 x 130 N	80 x 125 x 130 N	120 x 125 x 130		
Required front connector	1 x 40-pin	2 x 40-pin	1 x 40-pin		2 x 40-pin		
Order No. group: 6ES7	312-5BF.	313-5BG.	313-6BG.	313-6CG.	314-6BH.	314-6CH.	314-6EH.
Firmware N	V3.3	V3.3	V3.3	V3.3	V3.3	V3.3	V3.3
Memory							
Work memory N	64 KB	128 KB			192 KB		
Instructions N	21 K	42 K			64 K		
Processing times							
Bit operation N	0.1 µs	0.07 µs			0.06 µs		
Word operations N	0.24 µs	0.15 µs			0.12 µs		
Fixed-point operations N	0.32 µs	0.2 µs			0.16 µs		
Floating-point operations N	1.1 µs	0.72 µs			0.59 µs		
Bit memories/timers/counters							
Bit memory	256 bytes						
S7 timers/counters	256 / 256						
IEC timers/counters	● ¹⁾						
Address ranges							
Number of I/Os (bytes)	1 024 / 1 024			2 048 / 2 048	1 024 / 1 024	2 048 / 2 048	
I/O process image	1 024 / 1 024			2 048 / 2 048	1 024 / 1 024	2 048 / 2 048	
Digital channels (central)	266	1 016	1 008		1 016		
Analog channels (central)	64	253	248		253		
Integrated functions							
Counter (incremental enc.)	2, 24 V/10 kHz	3, 24 V/30 kHz			4, 24 V/60 kHz		
Pulse outputs (PCM)	2 channels, max. 2.5 kHz	3 channels, max. 2.5 kHz			4 channels, max. 2.5 kHz		
Frequency measurement	2 channels max. 10 kHz	3 channels, max. 30 kHz			4 channels max. 60 kHz		
Controlled positioning					SFB for positioning, 1 axis via 2 DO, AO		
Integrated "Control" FB		PID controllers					
Integrated inputs/outputs							
Digital inputs	10 x 24 V DC; all channels can be used for process interrupts	24 x 24 V DC; all channels can be used for process interrupts	16 x 24 V DC; all channels can be used for process interrupts		24 x 24 V DC; all channels can be used for process interrupts		
Digital outputs	6 x 24 V DC, 0.5 A	16 x 24 V DC, 0.5 A					
Analog inputs		4 : ± 10 V, 0 ... 10 V, ± 20 mA, 0 / 4 ... 20 mA; 1 : 0 ... 600 Ω, PT100			4 : ± 10 V, 0 ... 10 V, ± 20 mA, 0 / 4 ... 20 mA; 1 : 0 ... 600 Ω, PT100		
Analog outputs		2 : ± 10 V, 0 ... 10 V, ± 20 mA, 0 / 4 ... 20 mA			2 : ± 10 V, 0 ... 10 V, ± 20 mA, 0 / 4 ... 20 mA		
DP interface							
DP master systems int./ CP 342-5	○ / ●			● / ●	○ / ●	● / ●	● / ●
DP slave				●		●	●
Data set gateway ²⁾				●		●	●
PROFINET interface							
PROFINET CBA							●
PROFINET IO							●
PROFINET with IRT							●
Open User Communication (OUC)							
• TCP/IP							●
• UDP							●
• ISO-on-TCP (RFC 1006)							●
Web server							●
PtP interface							
Physics			RS485/422		RS485/422		
Protocol driver			3964 (R), RK512, ASCII		3964 (R), RK512, ASCII		

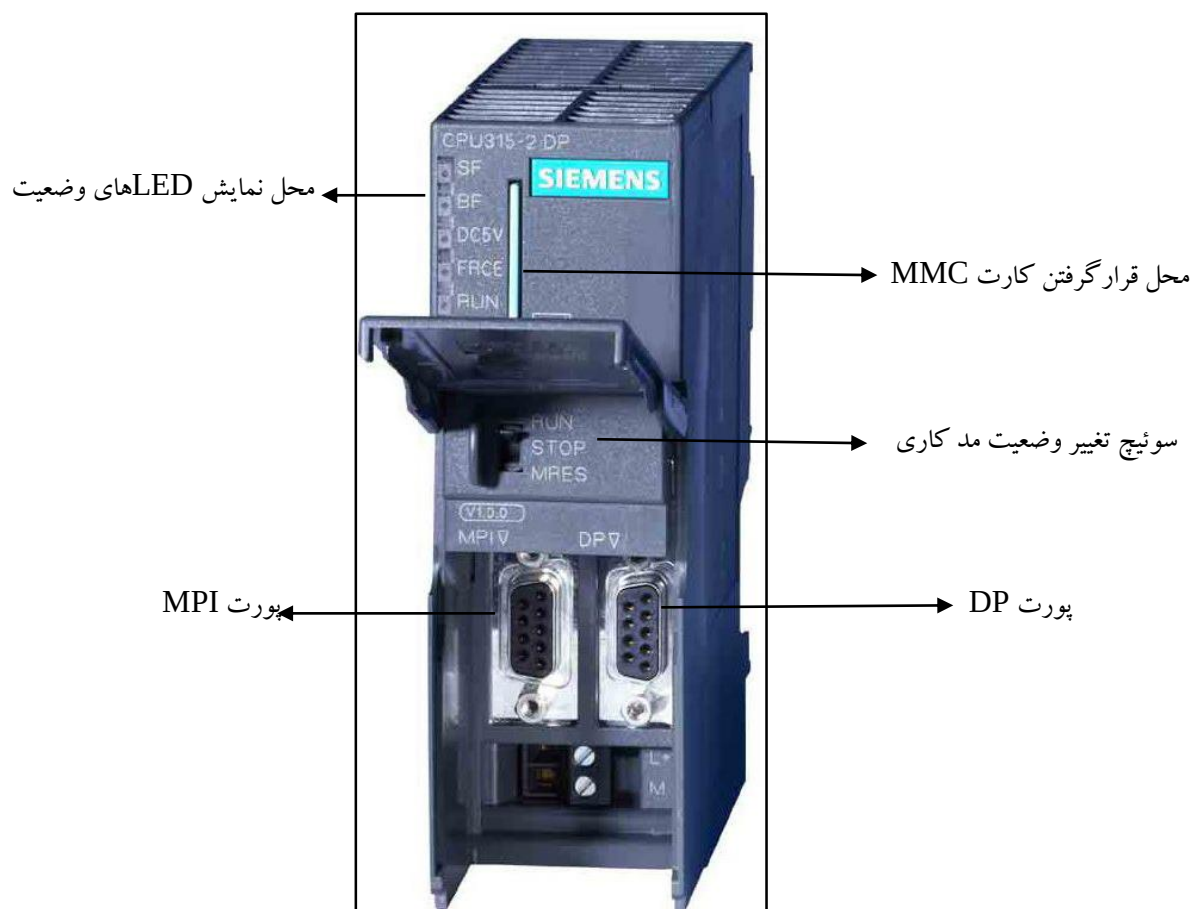
-مشخصات CPU های ماژولار

- ✓ SIMATIC S7-300 CPU 312
- ✓ SIMATIC S7-300 CPU 314
- ✓ SIMATIC S7-300 CPU 315-2 DP
- ✓ SIMATIC S7-300 CPU 315-2 PN/DP
- ✓ SIMATIC S7-300 CPU 317-2 DP
- ✓ SIMATIC S7-300 CPU 317-2 PN/DP
- ✓ SIMATIC S7-300 CPU 319-3 PN/DP

Technical data: Standard CPUs **N** = Product innovations

CPU	CPU 312	CPU 314 ¹⁾	CPU 315-2 DP ¹⁾	CPU 315-2 PN/DP ¹⁾	CPU 317-2 DP	CPU 317-2 PN/DP ¹⁾	CPU 319-3 PN/DP
Dimensions (mm)	40 x 125 x 130	40 x 125 x 130	40 x 125 x 130	40 x 125 x 130	40 x 125 x 130 N	40 x 125 x 130	120 x 125 x 130
Order No. group: 6ES7	312-1AE.	314-1AG.	315-2AH.	315-2EH.	317-2AK.	317-2EK.	318-3EL.
Firmware	V3.0			V3.2	V3.3 N	V3.2	V3.2
Memory							
Work memory	32 KB	128 KB	256 KB	384 KB	1 MB N	1 MB	2 MB
Instructions	10 K	42 K	85 K	128 K	340 K	340 K	680 K
Processing times							
Bit operation	0.1 µs	0.06 µs	0.05 µs		0.025 µs N	0.025 µs	0.004 µs
Word operation	0.24 µs	0.12 µs	0.09 µs		0.03 µs N	0.03 µs	0.01 µs
Fixed-point operation	0.32 µs	0.16 µs	0.12 µs		0.04 µs N	0.04 µs	0.01 µs
Floating-point operation	1.1 µs	0.59 µs	0.45 µs		0.16 µs N	0.16 µs	0.04 µs
Bit memories/timers/counters							
Bit memory	256 bytes		2 048 bytes		4 096 bytes		8 192 bytes
S7 timers/counters	256 / 256				512 / 512		2 048 / 2 048
IEC timers/counters	● ¹⁾						
Address ranges							
Number of I/Os (bytes)	1 024 / 1 024		2 048 / 2 048		8 192 / 8 192		
Process image I/O (bytes), max.	1 024 / 1 024		2 048 / 2 048		8 192 / 8 192		
Digital channels (central)	256	1 024					
Analog channels (central)	64	256					
DP interfaces							
DP master systems internal / CP 342-5	○ / ●		● / ●				
DP slaves			●				
Data set gateway ²⁾			●	●	●	●	
PROFINET interface							
PROFINET CBA				●		●	
PROFINET IO				●		●	
PROFINET with IRT				● ³⁾		● ³⁾	
Open User Communication (OUC)							
• TCP/IP				●		●	
• UDP				●		●	
• ISO-on-TCP (RFC 1006)				●		●	
Web server				●		●	

آشنایی با ماژول CPU



وضعیت LED های CPU

SF: این LED وضعیت خطاهای سیستمی شامل خطای سخت افزاری و نرم افزاری را با رنگ قرمز گزارش می کند.

BF: این LED وضعیت خطاهای مربوط به شبکه را از لحاظ سخت افزاری و نرم افزاری با رنگ قرمز گزارش می کند.

DC5V: این LED وضعیت صحت و سلامت سطح ولتاژ 5V داخلی باس را با رنگ سبز گزارش می کند.

FRCE: این LED در شرایطی که قابلیت نرم افزاری Force حداقل روی یک بیت اعمال شده باشد با رنگ نارنجی، روشن می شود.

کارت حافظه S7-300

در سری جدید PLC های S7-300 حافظه ی Load Memory داخلی وجود ندارد و این کارت به عنوان حافظه بارگذاری جهت ذخیره ی برنامه ی کاربر استفاده می شود. بنابراین وجود این کارت در CPU های جدید (از سال 2002 به بعد) الزامی می باشد. با توجه به اینکه این کارت از نوع فلش می باشد، با قطع تغذیه و خارج نمودن آن از اسلات مربوطه بر روی CPU، اطلاعات داخل آن از بین نمی رود.



کارت حافظه MMC در ظرفیت های مختلف ارائه شده است.

Bezeichnung	Order no.
SIMATIC Micro Memory Card 64 KB	6ES7953-8LF20-0AA0
SIMATIC Micro Memory Card 128 KB	6ES7953-8LG20-0AA0
SIMATIC Micro Memory Card 512 KB	6ES7953-8LJ30-0AA0
SIMATIC Micro Memory Card 2 MB	6ES7953-8LL31-0AA0
SIMATIC Micro Memory Card 4 MB	6ES7953-8LM20-0AA0
SIMATIC Micro Memory Card 8 MB	6ES7953-8LP20-0AA0

در CPU های مختلف S7-300 ماکزیمم فضای بارگذاری مقداری مشخص می باشد. در جدول زیر به برخی از این موارد اشاره شده است.

CPU	ORDER NO.	TYPE OF MEMORY CARD
CPU 312	6ES7312-1AD10-0AB0	MMC MAX. 4MB
	6ES7312-1AE13-0AB0	MMC MAX. 4MB
CPU 312 IFM	6ES7312-5AC02-0AB0	No Memory Card
	6ES7312-5AC82-0AB0	No Memory Card
CPU 313	6ES7313-1AD02-0AB0	MC MAX. 512KB
	6ES7313-1AD03-0AB0	MC MAX. 4MB

CPU 314C-2 DP	6ES7314-6CF00-0AB0	MMC MAX.4MB
	6ES7314-6CF01-0AB0	MMC MAX.8MB
CPU 315-2DP	6ES7315-2AF02-0AB0	MC MAX. 512KB
	6ES7315-2AF03-0AB0	MC MAX. 4MB
	6ES7315-2AG10-0AB0	MMC MAX. 8MB
CPU 316	6ES7316-1AG00-0AB0	MC MAX. 512KB
CPU 316-2DP	6ES7316-2AG00-0AB0	MC MAX. 4MB
CPU 317-2PN/DP	6ES7317-2FK13-0AB0	MMC MAX. 8MB
CPU 318-2	6ES7318-2AJ00-0AB0	MC MAX. 4MB
CPU 319-3 PN/DP	6ES7318-3EL00-0AB0	MMC MAX. 8MB
	6ES7318-3EL01-0AB0	MMC MAX. 8MB

بخش های مختلف حافظه

فضای حافظه در CPU به 3 قسمت تقسیم می شود:

Load Memory (حافظه بارگذاری): فضایی از حافظه می باشد که برنامه کاربر شامل بلوک های منطقی و دیتا بلوک ها، تنظیمات سیستمی و شبکه در آن قرار می گیرد. این حافظه را می توان معادل هارد کامپیوتر در نظر گرفت.

Work Memory (حافظه کاری): بخش های اجرایی برنامه از حافظه بارگذاری به حافظه کاری جهت اجرا منتقل می شوند. این حافظه را می توان معادل RAM کامپیوتر در نظر گرفت.

System Memory (حافظه سیستمی): این فضا شامل بخش های زیر می باشد:

- T (تایمرها)
- C (کانترها)
- M (حافظه)
- PII و PIO (حافظه مربوط به تصاویر ورودی و خروجی)
- L STACK (دیتاهای محلی)

فضاهای حافظه بارگذاری در S7-300 جدید و قدیم و S7-400 طراحی متفاوتی دارند که در بحث های بعدی با جزئیات این موضوع آشنا می شویم.

سوئیچ تعیین وضعیت مد کاری

توسط این سوئیچ حالت کاری CPU از حالت RUN $\rightarrow$ STOP و برعکس تغییر می کند.

مد Startup :

وقتی توسط کلید تعبیه شده بر روی CPU و یا از طریق نرم افزار، CPU به مد RUN سوئیچ می شود، در واقع ابتدا وارد مد Startup می شود. این مد در حالت نرمال، کاملاً گذرا می باشد. در این مد وضعیت سخت افزار چک شده و اگر برنامه ای در بلوکهای Startup نوشته شده باشد، اجرا می شود. در ضمن بسته به تنظیم نوع راه اندازی، ممکن است قسمت هایی یا تمام دیتای حافظه سیستمی پاک شود.

انواع Restart (راه اندازی مجدد CPU)

Warm Restart: در این نوع راه اندازی که در S7-300 و S7-400 قابل اجرایی باشد، اطلاعات فضای حافظه ی سیستمی که به صورت ناپایدار در نظر گرفته شده اند، از بین می روند و بخش های پایدار اطلاعات خود را حفظ می کنند. در ضمن در این نوع راه اندازی، برنامه کاربر از ابتدا آغاز می شود.

Cold Restart: در این نوع راه اندازی کلیه محتویات حافظه سیستمی چه از نوع پایدار و چه از نوع ناپایدار از بین می روند. این نوع راه اندازی در PLC S7-400 و CPU 318-2 در S7-300 قابل اجرا می باشد.

Hot Restart: در این راه اندازی که مختص PLC S7-400 و CPU318 در S7-300 می باشد، کلیه اطلاعات فضای حافظه ی سیستمی چه از نوع ماندگار یا غیرماندگار حفظ می شوند و برنامه کاربر از همان نقطه ای که قطع شده بود ادامه پیدا می کند.

همچنین با نگه داشتن سوئیچ در مود MRES با در نظر گرفتن شرایطی CPU ریست می شود. در واقع ریست کردن CPU روی هر سه فضای حافظه ی Load Memory، Work Memory و System Memory تاثیر می گذارد. به عبارت دیگر وقتی CPU ریست می شود، مقادیر حافظه های ذکر شده پاک می شوند. اگر کارت حافظه از نوع RAM باشد، با ریست کردن محتوای آن پاک می شود ولی اگر کارت حافظه از نوع FLASH باشد، محتوای آن پاک نمی شود.

پورت MPI

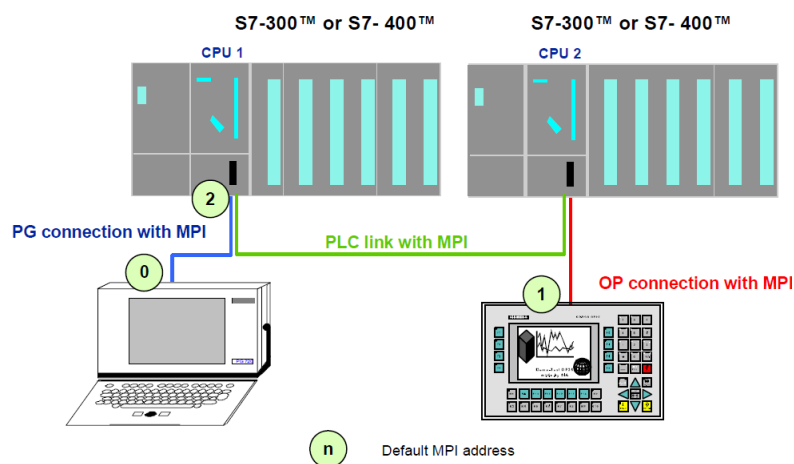
MPI شبکه استاندارد و مختص زیمنس است که روی تمام PLC های سری S7-300 و S7-400 وجود دارد. برای این درگاه ارتباطی سه کاربرد اصلی بیان می شود:

1- جهت برنامه ریزی برای اتصال کابل PC Adaptor MPI

2- جهت اتصال به پانل های صنعتی (HMI)

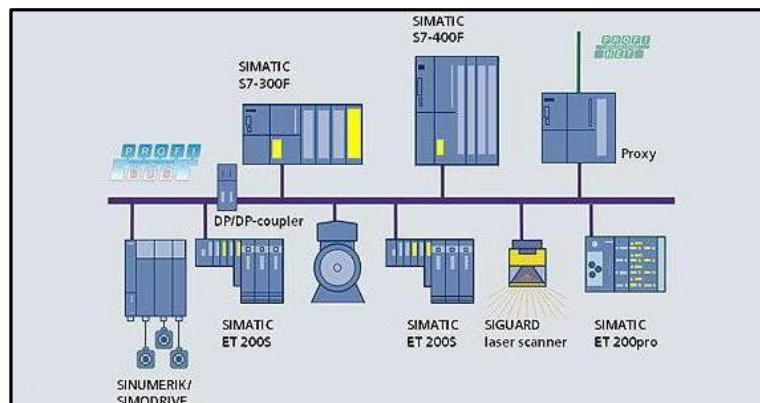
3- جهت اتصال یک PLC S7 به PLC S7 دیگر

Networking with MPI

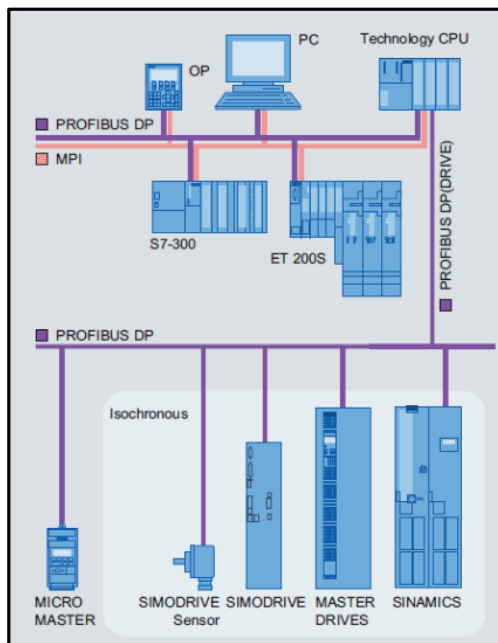


پورت DP

توسط این پورت، PLC S7 می تواند به شبکه پروفیباس (نسخه DP) متصل شود. البته شبکه پروفیباس در 3 نسخه FMS، DP و PA عرضه شده است.

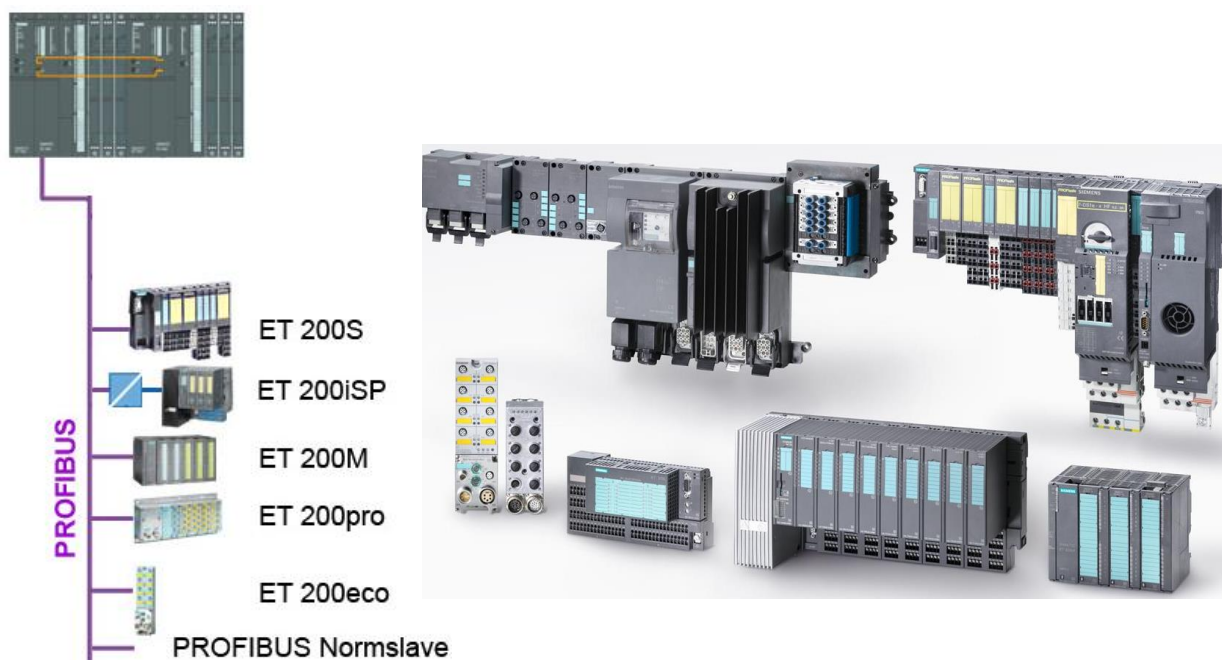


برخی از مهمترین ایستگاه هایی که در این شبکه قرار می گیرند:



- PLC
- CNC
- HMI
- PC
- DRIVE
- ET
- تجهیزات ابزار دقیق مجهز به اینترفیس شبکه
- تجهیزات قدرت مجهز به اینترفیس شبکه

ET ها یا همان Remote I/O ها یکی از مهمترین ایستگاه هایی هستند که می توانند در شبکه پروفیاس قرار گیرند. وظیفه Remote I/O ها جمع آوری سیگنال از سطح پلنت و انتقال آن به یک CPU می باشد. البته استفاده از ET در پروژه ها می تواند به دلایل مختلفی باشد. شرکت زیمنس ET های متنوعی را طراحی و عرضه نموده است.



کابل شبکه پروفیباس یک کابل دو رشته مسی می باشد.

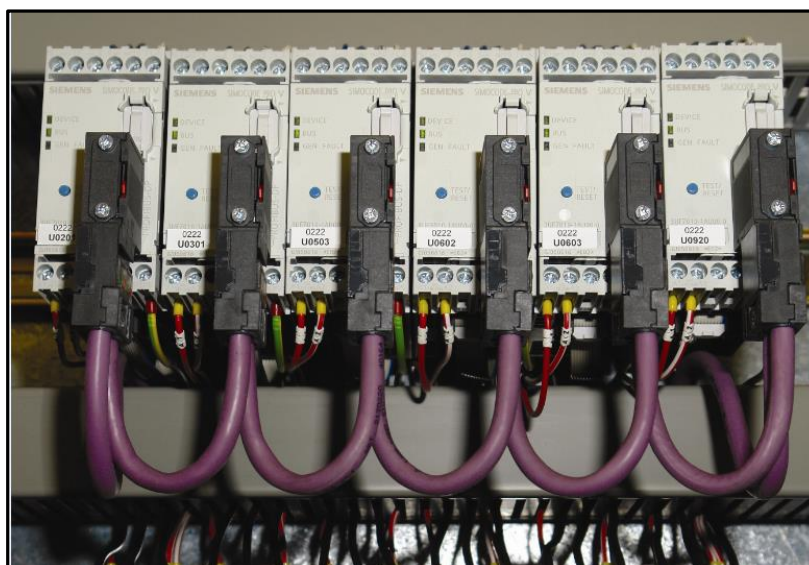


ارتباط این کابل با پورت DP توسط کانکتورهای مطابق شکل زیر برقرار میشود.

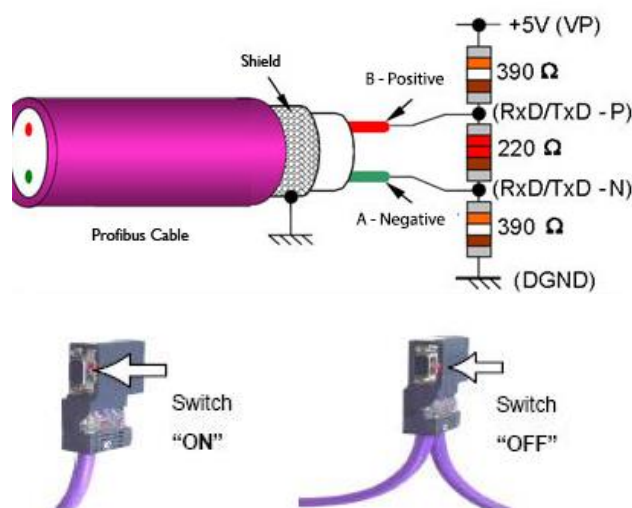


بر روی این کانکتور دو محل ورود و خروج جهت اتصال به ایستگاه قبلی و بعدی تعبیه شده است. توسط کابل مسی می توان تنها توپولوژی باس را بدون استفاده از تجهیز خاصی طراحی و اجرا نمود. در توپولوژی باس، ایستگاه ها به صورت سری به یکدیگر متصل می شوند.

در شکل زیر این توپولوژی را مشاهده می کنید.



بر روی این کانکتور، سوئیچی جهت On و Off کردن ترمیناتور تعبیه شده است. در شکل زیر مدار مقاومتی مربوط به ترمیناتور را مشاهده می کنید. با On کردن سوئیچ، مدار زیر در انتهای باس بر روی مسیر ورودی کانکتور برقرار میشود.



با On کردن این سوئیچ در ایستگاه ابتدا و انتها، باس بسته شده و از بازگشت سیگنال های ناخواسته و بازگشتی جلوگیری می کند. بحث ترمینیت کردن در سرعت های بالا و مسافت های طولانی در کیفیت سیگنال بسیار مهم می باشد.

ماکزیمم سرعت شبکه پروفیباس 12Mbps می باشد. این سرعت با توجه به مسافت محدود می شود.

Baud rate	Maximum segment length
9.6 kbit/s	1 200m
19.2 kbit/s	1 200m
45.45 kbit/s	1 200m
93.75 kbit/s	1 200m
187.5 kbit/s	1 000m
500.0 kbit/s	400m
1.5 Mbit/s	200m
3.0 Mbit/s	100m
6.0 Mbit/s	100m
12.0 Mbit/s	100m

توسط Repeater می توان مسافت کابل را با حفظ سرعت افزایش داد. البته ریپتر کاربردهای دیگری نیز دارد.



افزایش تعداد ایستگاه

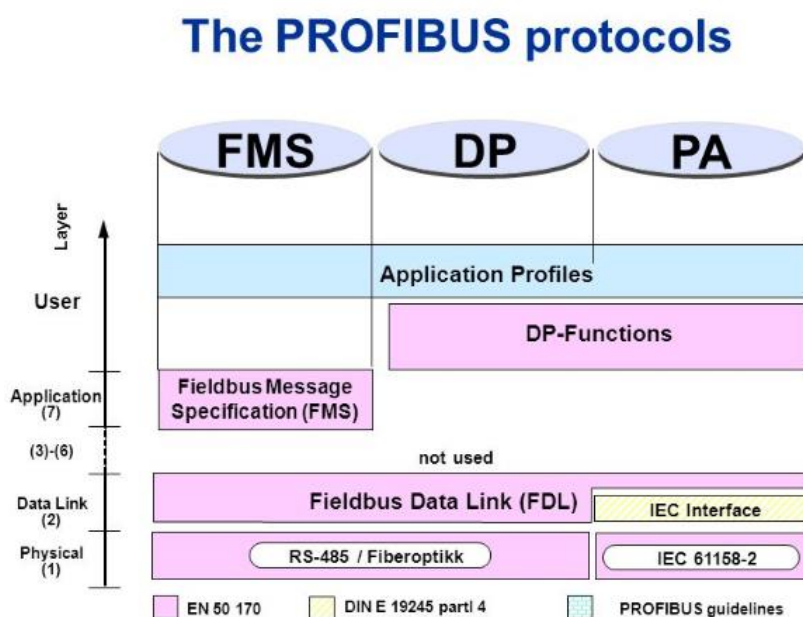
افزایش مسافت کابل با حفظ سرعت

ایزوله کردن دو سگمنت

پیاده سازی توپولوژی Tree

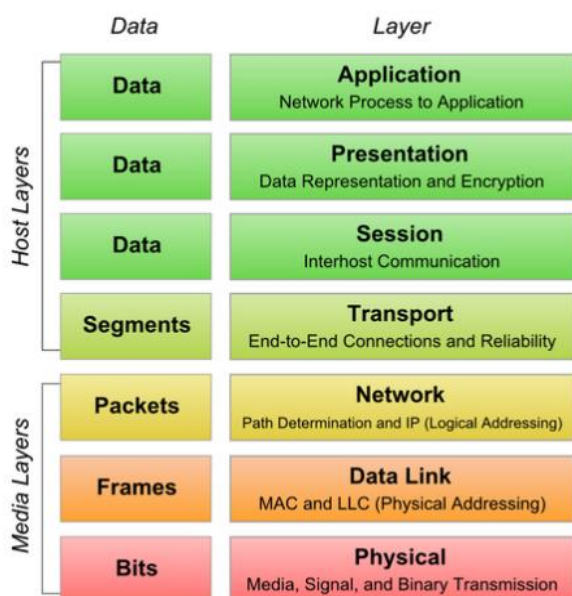
اتصال یک HMI به دو CPU

مدل OSI در شبکه پروفیباس:



همانطور که در شکل فوق استفاده می کنید، نسخه DP و PA از لایه های 1 و 2 و در صورت نیاز از یک لایه واسط استفاده می کنند. FMS از لایه های 1، 2 و 7 مدل OSI استفاده می کند. لایه های هفت گانه مدل OSI به صورت زیر می باشند.

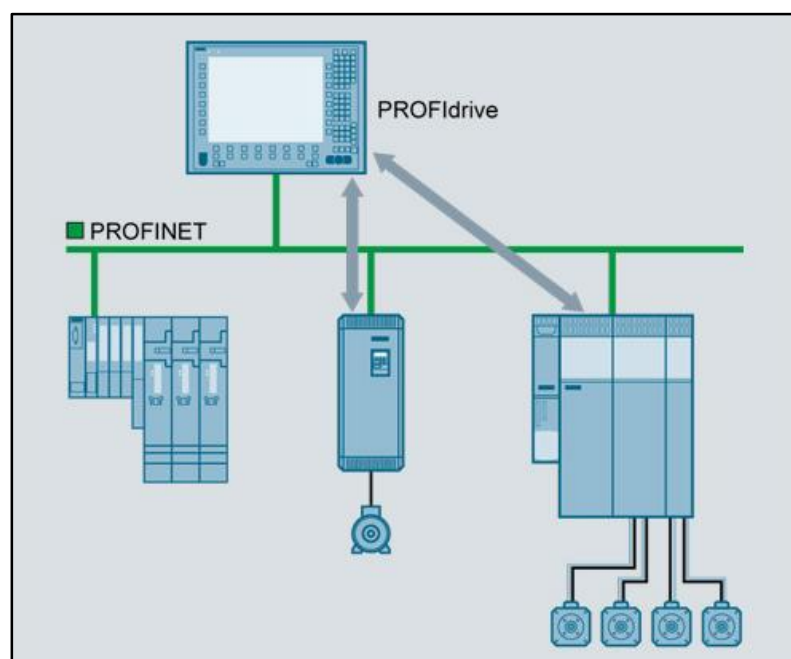
OSI Model



برخی از CPU ها نیز دارای پسوند PN می باشند. این CPU ها مجهز به رابط LAN می باشند. توسط این رابط امکان اتصال به شبکه Profinet امکان پذیر می باشد. این شبکه در بستر Ethernet شبکه ای مناسب برای ارتباط با Remote I/O های جدید می باشد. البته این شبکه به این موضوع محدود نمی شود و روز به روز در حال توسعه می باشد.



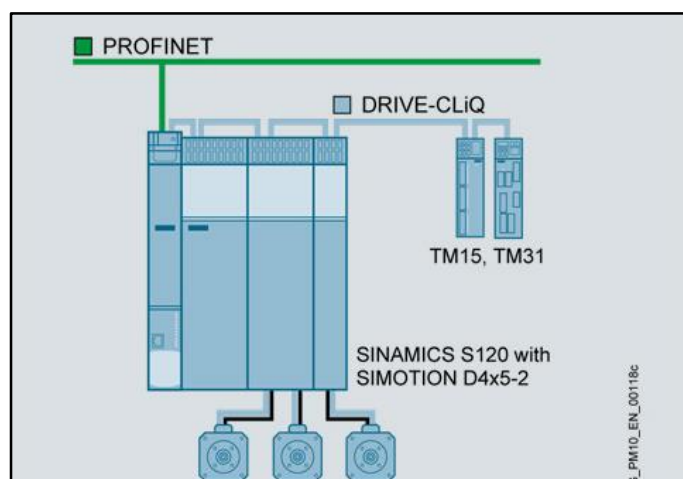
امروزه شرکت زیمنس، ET هایی را با قابلیت اتصال به شبکه Profinet طراحی و عرضه نموده است. همچنین برخی از درایوهای زیمنس نیز قابلیت اتصال به این شبکه را دارا می باشند.



در شکل زیر یک ایستگاه ET200S که مجهز به رابط Profinet می باشد را مشاهده می کنید.



در شکل زیر یک درایو SINAMICS زیمنس که مجهز به رابط Profinet می باشد را مشاهده می کنید.



آشنایی با ماژول های ورودی و خروجی S7-300



کارت های ورودی دیجیتال DI : SM321

کارت های خروجی دیجیتال DO : SM322

کارت های ورودی/خروجی دیجیتال DI/DO : SM323

کارت های ورودی آنالوگ AI : SM331

کارت های خروجی آنالوگ AO : SM332

کارت های ورودی/خروجی آنالوگ AI/AO : SM334



آشنایی با ماژول های ورودی و خروجی S7-400

کارت های ورودی دیجیتال DI : SM421

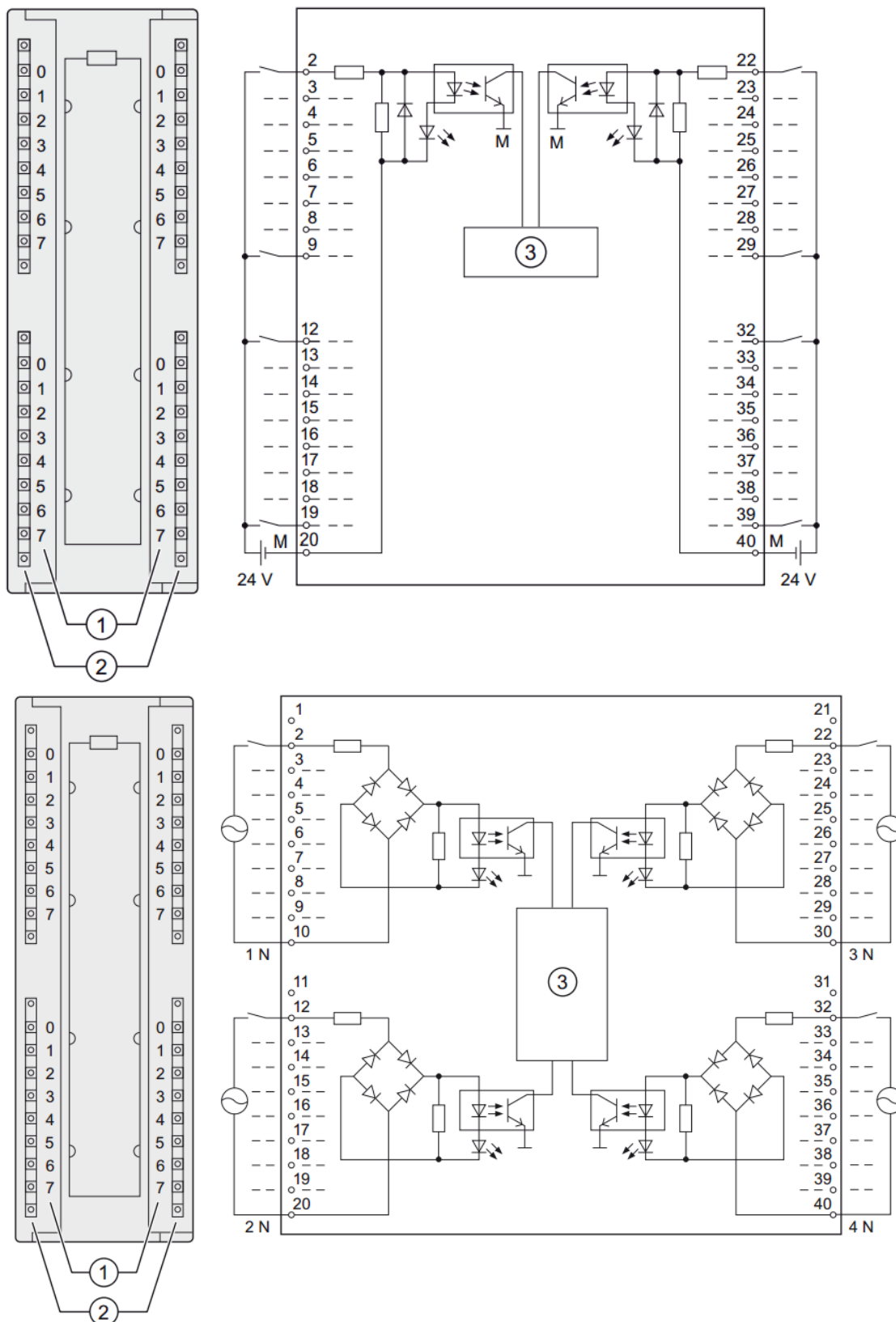
کارت های خروجی دیجیتال DO : SM422

کارت های ورودی آنالوگ AI : SM431

کارت های خروجی آنالوگ AO : SM432

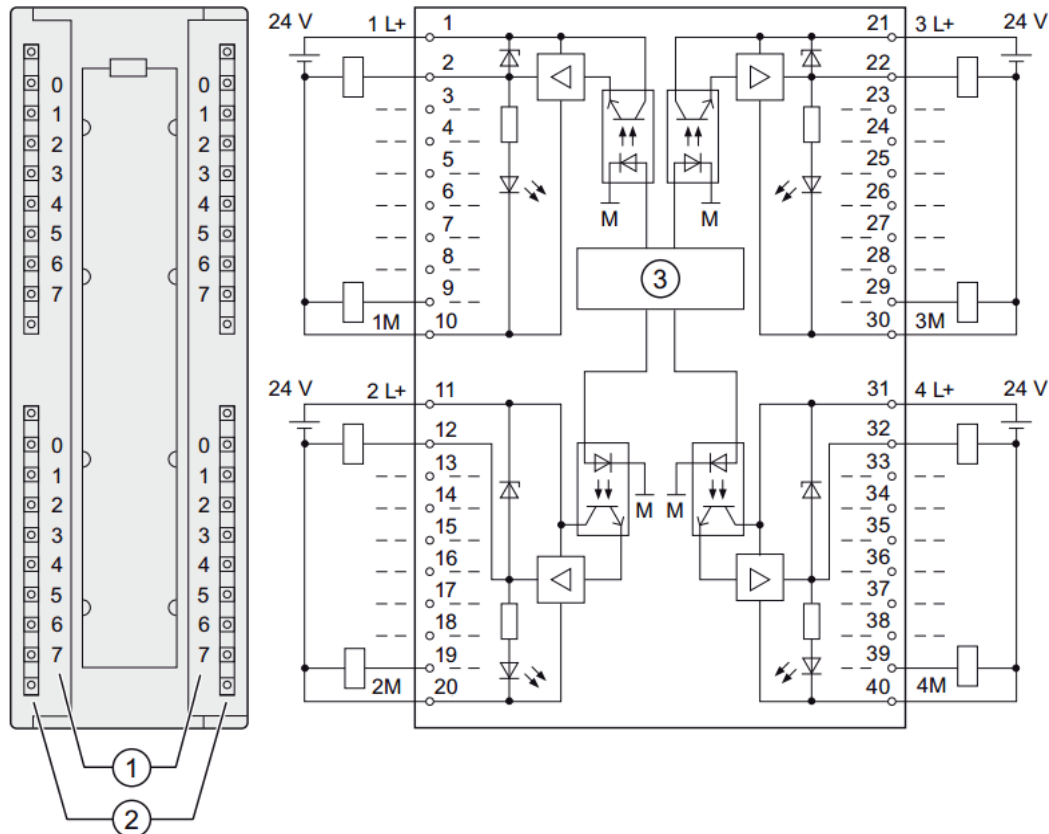
اتصالات در ماژول های DI

Wiring and block diagrams of SM 321; DI 32 x DC 24 V

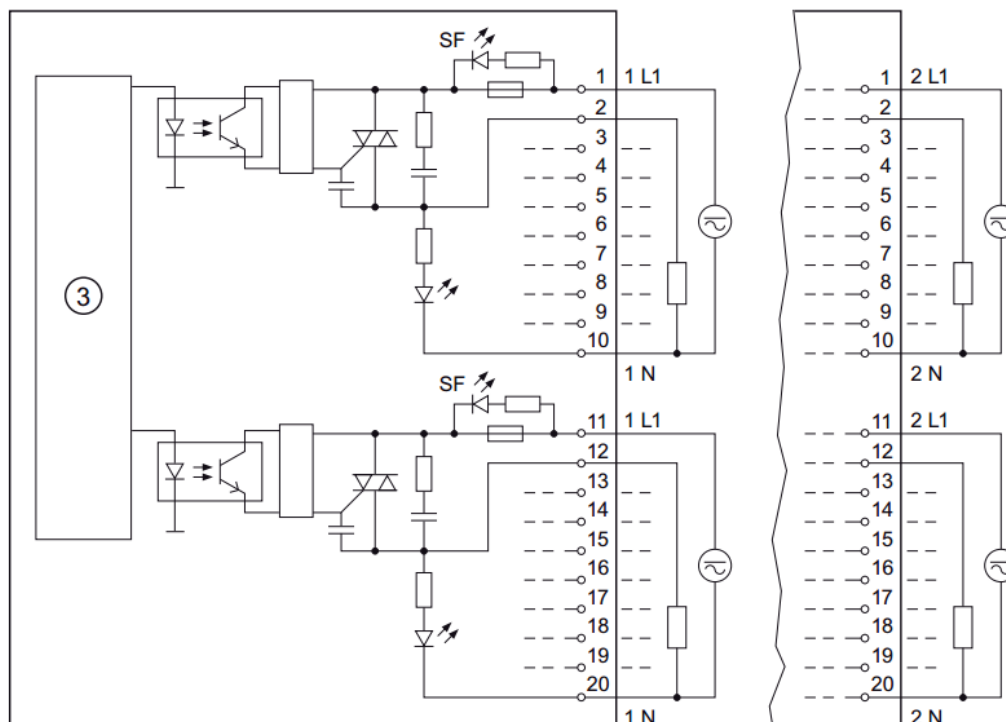


اتصالات در ماژول های DO

Wiring and block diagram of SM 322; DO 32 x DC 24 V/ 0.5 A



Wiring and block diagram of SM 322; DO 32 x AC 120/230 V/1 A

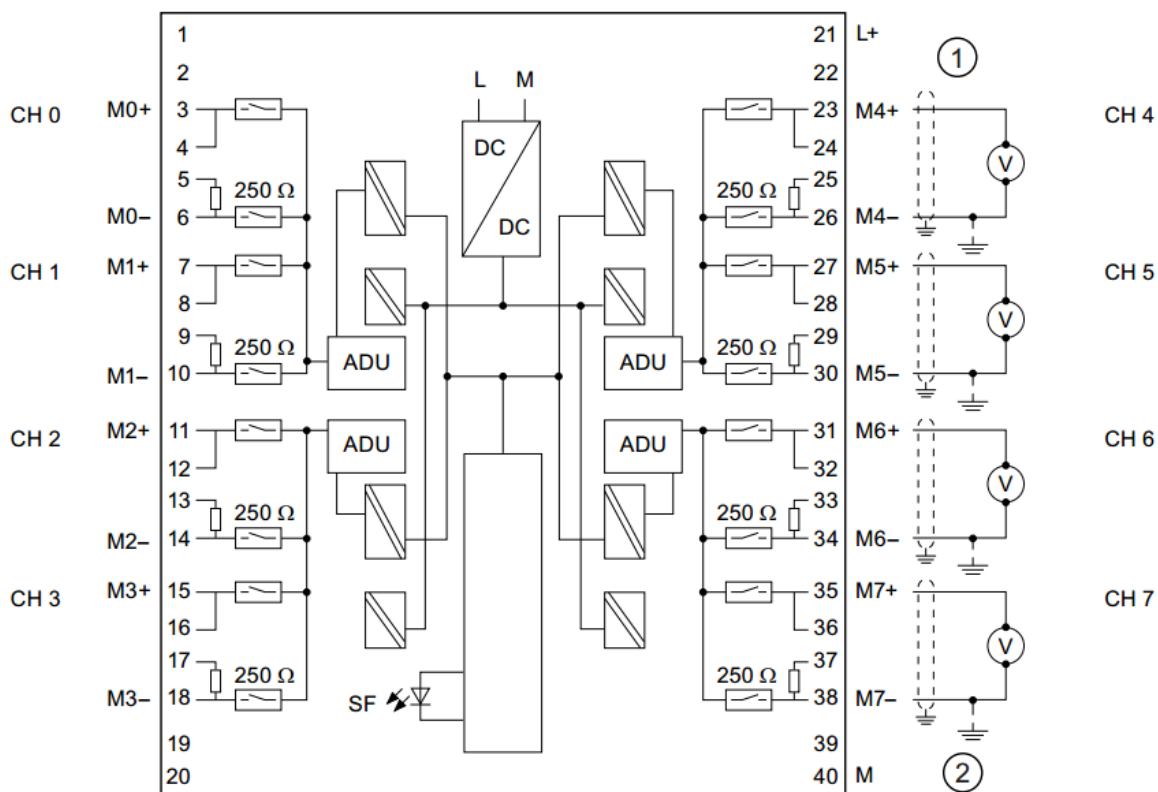


اتصالات در ماژول های AI

نحوه اتصال سیگنال های ولتاژی

Wiring: Voltage and current measurement

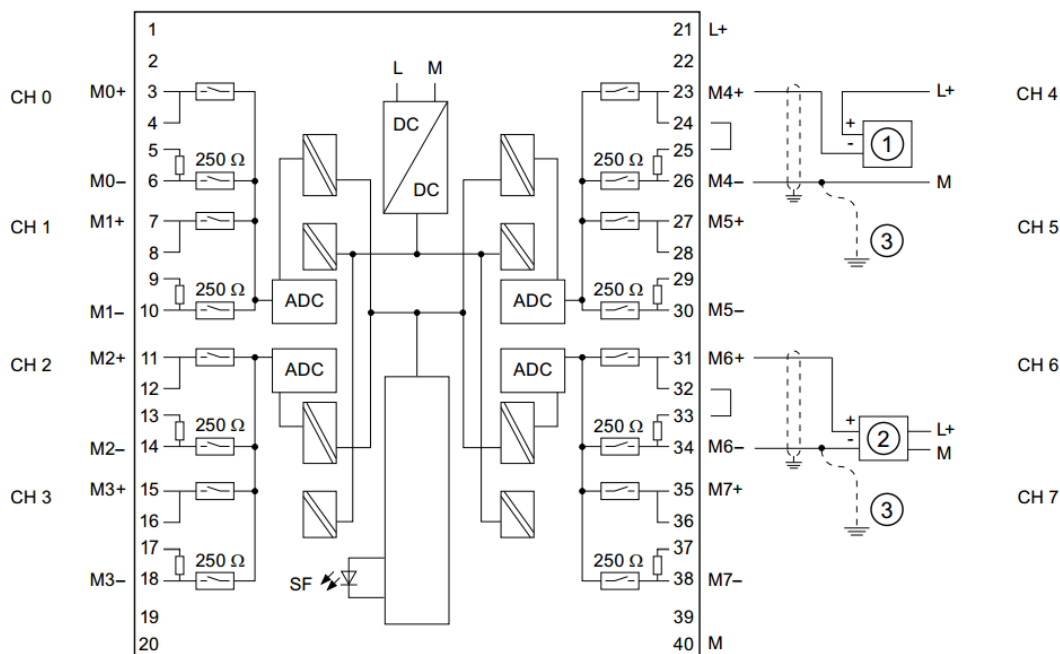
Connection possible at both sides at channels 0 to 7



نحوه اتصال سیگنال های جریانی

Wiring: 2-wire and 4-wire transducer

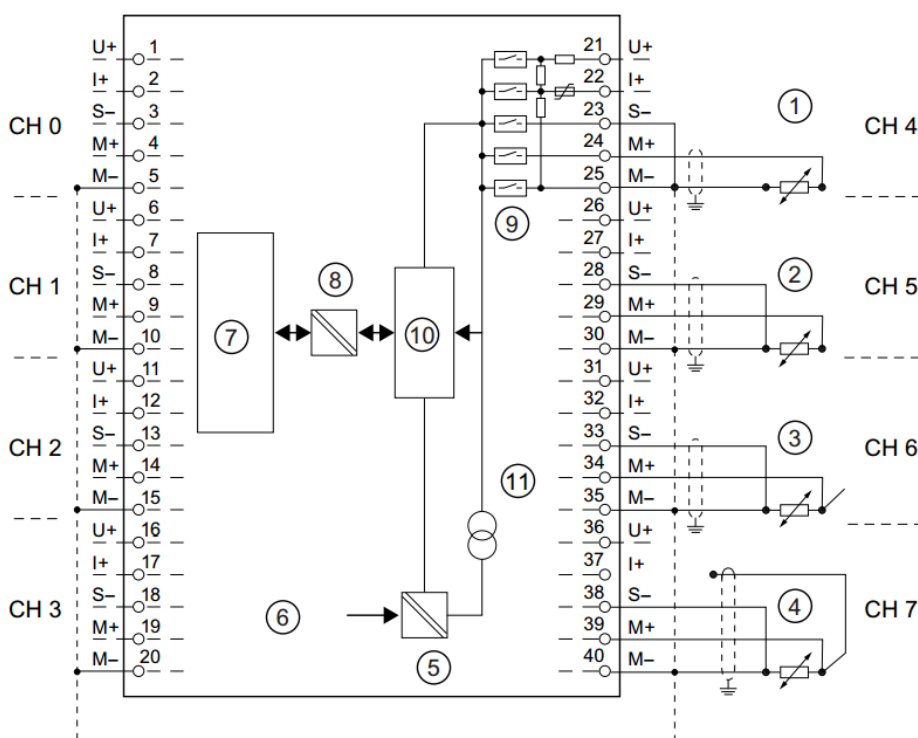
Connection possible at both sides at channels 0 to 7



نحوه اتصال سنسورهای مقاومتی RTD

Wiring: Resistance measurement with 2-, 3- and 4-wire connection

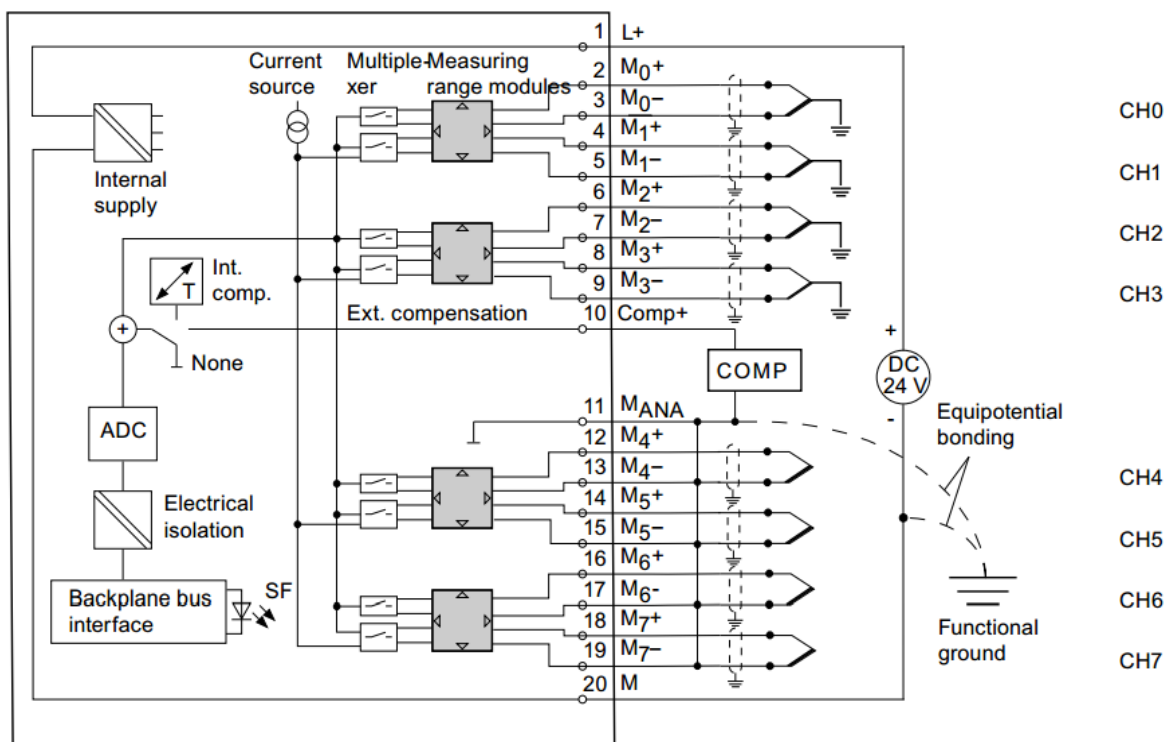
The following connection possibilities also apply to silicon temperature sensors and PTCs.



نحوه اتصال ترموکوپل

Wiring: Thermocouples with external compensation

Insert a bridge between Comp+ and M_{ANA} when using the internal compensation.



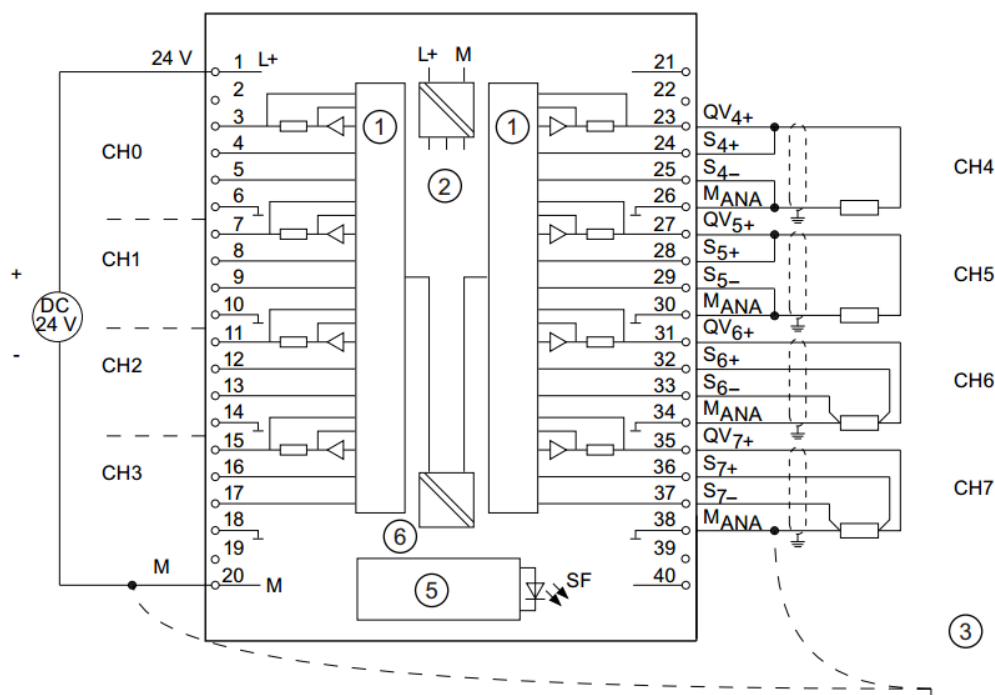
اتصالات در ماژول های AO

نحوه اتصال بارهای ولتاژی

Wiring: 2 and 4-wire connection for voltage output

The following image depicts:

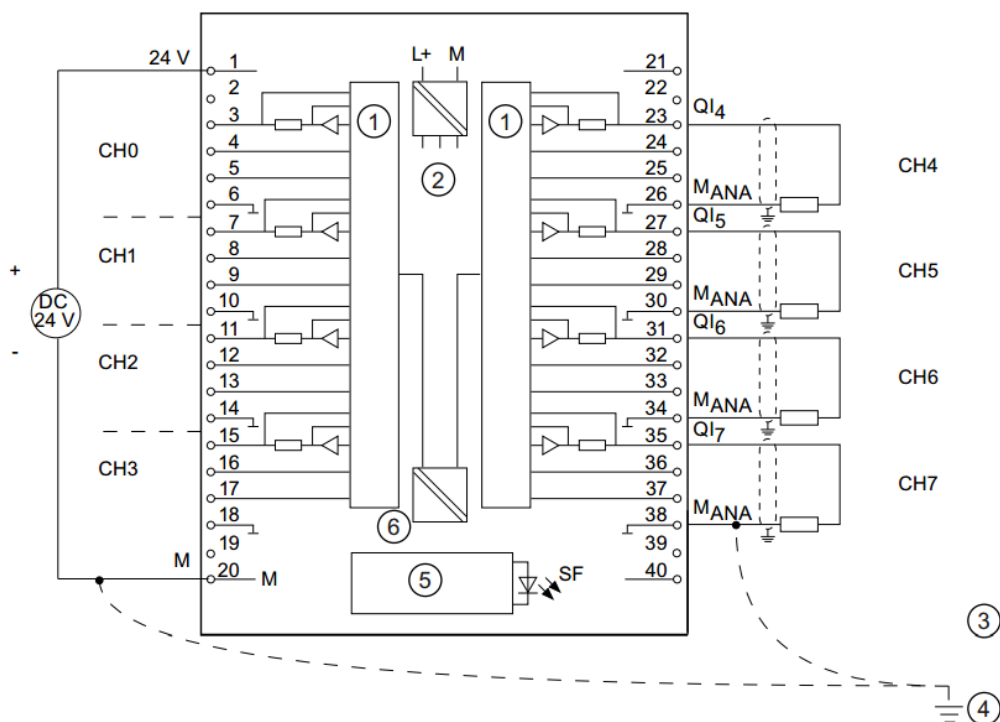
- 2-wire connection, no compensation for line impedance and
- 4-wire connection with compensation for line impedance



اتصالات در ماژول های AO

نحوه اتصال بارهای جریانی

Wiring: Current output



آشنایی با ماژول های ارتباطی IM در S7-300

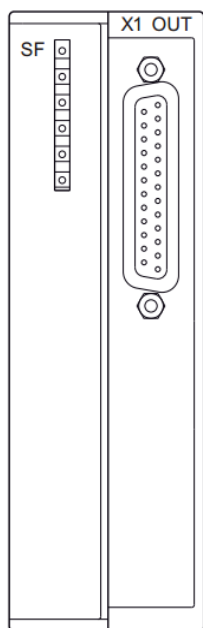
ماژول IM360S و IM361R جهت توسعه رک تا سه رک افزایشی علاوه بر رک اصلی مورد استفاده قرار می گیرد. محل قرارگیری این کارت در اسلات 3 روی ریل PLC می باشد.



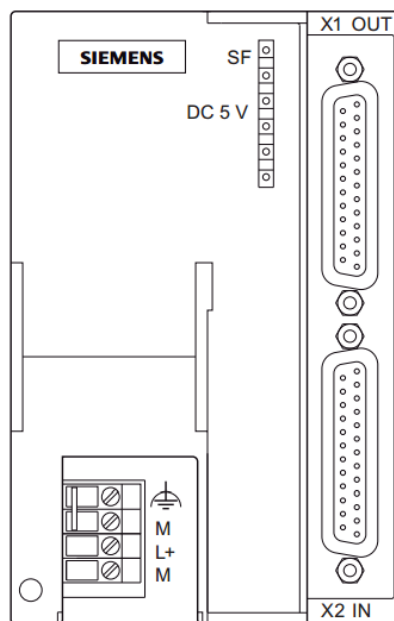
DI/DO Default Addressing in Multi-Tier Configurations

Rack 3	PS	IM (Receive)	96.0 to 99.7	100.0 to 103.7	104.0 to 107.7	108.0 to 111.7	112.0 to 115.7	116.0 to 119.7	120.0 to 123.7	124.0 to 127.7	
Rack 2	PS	IM (Receive)	64.0 to 67.7	68.0 to 70.7	72.0 to 75.7	76.0 to 79.7	80.0 to 83.7	84.0 to 87.7	88.0 to 91.7	92.0 to 95.7	
Rack 1	PS	IM (Receive)	32.0 to 35.7	36.0 to 39.7	40.0 to 43.7	44.0 to 47.7	48.0 to 51.7	52.0 to 55.7	56.0 to 59.7	60.0 to 63.7	
Rack 0	PS	CPU	IM (Send)	0.0 to 3.7	4.0 to 7.7	8.0 to 11.7	12.0 to 15.7	16.0 to 19.7	20.0 to 23.7	24.0 to 27.7	28.0 to 31.7
Slot	1	2	3	4	5	6	7	8	9	10	11

IM360S دارای یک رابط و IM361R دارای دو رابط می باشد.



IM 360



IM 361

ماژول IM365S/R جهت توسعه رک تنها تا یک رک افزایشی علاوه بر رک اصلی مورد استفاده قرار می گیرد.



مشخصات

S7-300/S7-300F

IM 360/361/365 interface modules



Overview

- For connecting mounting racks in multi-tier SIMATIC S7-300 configurations
- IM 365: For design of central controller and max. 1 expansion unit.
Limited use of modules in the expansion unit (e.g. no CPs or FMs)
- IM 360/IM 361: For design of central controller and max. 3 expansion units.
No limitation in selection of modules in the expansion unit

Area of application

IM 360/IM 361 and IM 365 modules support a multitier configuration of the S7-300 automation system (CPU 313C, CPU 314 or later), consisting of a central controller (CC) and up to three expansion racks (ERs).

The individual racks are interconnected via the interface modules.

- IM 365:
Central controller and 1 expansion rack each with max. 8 modules; Distance: 1 m
- IM 360/IM 361:
Central controller and max. 3 expansion racks each with max. 8 modules; Distance between two adjacent racks: 4 cm to 10 m

General features

All interfaces have the following common features:

- Compact configuration:
The rugged plastics housing contains the interfaces for the connecting cables.
- Simple mounting:
The interface modules are mounted on the DIN rail (slot 3) and connected to the I/O modules via a bus connector just like any other module.
- Problem-free configuration:
The interface modules are self-configuring. Address assignment is not required.
- Status and fault LEDs

IM 365

The IM 365 is the low-cost expansion for an ER. It contains the following features:

- One of the two IM 365 modules is inserted in the CC, the other into the ER. The modules are connected by a fixed cable 1 m in length.
- Limited use of modules:
The ER is not connected to the C bus (communication bus). Therefore, no C bus nodes (e.g., CPs and FMs, see configuring aids) may be inserted into the ER.
- No separate power supply:
The modules in the ER are powered by the CPU power supply.

IM 360/IM 361

IM 360 and IM 361 are the ideal solution for comprehensive expansions. They contain the following features:

- IM 360 plugs into CC
- IM 361 plugs into each ER
- Separate power supply
Each IM 361 requires an external 24 V DC power supply. This then powers all modules of the respective ER. A PS 307 load power supply can be connected using a power connector.
- No module selection restrictions:

آشنایی با ماژول های FM

از این ماژول ها به منظور انجام مقاصد خاص استفاده می شود. در واقع این ماژول ها می توانند عملیاتی مانند شمارش پالس های فرکانس بالا، تولید پالس های فرکانس بالا و سیگنال های کنترلی جهت کنترل سرو موتور و موتورپله ای، کنترلرهای PID و... پردازش کنند.

Function modules in detail

- > Counter Module FM 350-1/FM 350-2
- > Positioning Module FM 351
- > Cam Controller FM 352
- > High-Speed Boolean Processor FM 352-5
- > Positioning Module FM 353
- > Positioning Module FM 354
- > Controller Module FM 355
- > Controller Module FM 355-2
- > Positioning Module FM 357-2
- > PROFIBUS-Module IM 174
- > Position decoder module SM 338

Counter Module FM 350-1/FM 350-2

Intelligent counter module for counting tasks

Overview

- ✓ FM 350-1
- ✓ FM 350-2

FM 350-1

The intelligent FM 350-1 counter module has one counting channel for up and down counting, and is an excellent choice for universal high-frequency counting tasks up to 500 kHz. It also features measuring functions for speed, frequency and period length. In distributed configurations with S7-400 or WinAC RTX, the FM 350-1 supports isochrone mode.



FM 350-2

The FM 350-2 counter module, by contrast, is an 8-channel counter module for universal counting and measuring tasks up to 20 kHz. It can be used in the SIMATIC S7-300 and in the ET 200M (with S7-300/S7-400 as master).



Positioning Module FM 351

For controlling rapid traverse/creep speed drives

Overview

✓ Description

Description

The FM 351 is a module for adjusting and positioning mechanical axes based on the rapid traverse/creep speed principle. With this module, two axes can be positioned independently of one another. The axes to be positioned may be linear axes with limited traversing range and continuously rotating rotary axes. Incremental encoders and SSI encoders can be used for position decoding. The FM 351 is used in the SIMATIC S7-300, C7 and in the ET 200M on S7-masters as well as with WinAC Basis and WinAC Slot 41x.

**Cam Controller FM 352**

Solves position or time-dependent tasks

Overview

✓ Description

✓ Advantages

Description

The FM 352 electronic cam controller is a 1-channel module for solving position- or time-dependent tasks.

They are far superior to mechanical cam controllers, particularly thanks to their high level of flexibility.

**Advantages**

- Either an incremental encoder, an SSI encoder or a 24 V initiator can be connected.
- 32 cam tracks can be read from the automation system's CPU.
- 13 cam tracks can be output directly over integrated digital outputs
- The module can be quickly adapted to the relevant cam control application by means of parameters.
- The FM 352 off-loads the CPU thanks to autonomous, actual value-dependent setting and resetting of electronic cams
- The FM 352 is used in the SIMATIC S7-300, C7 and ET 200M on S7 masters

Positioning Module FM 353

For stepper motor control

Overview

▼ Description

Description

The FM 353 is a positioning module for controlling stepper motors. It can perform many functions for solving a wide variety of positioning tasks, and offers a high level of precision, dynamic response, and user-friendliness. In contrast to modules which control servomotors, the FM 353 requires no position decoding system

It can be used in all low-load positioning tasks and where there are no major load fluctuations. When these prerequisites are fulfilled, virtually the same degree of accuracy can be achieved with the FM 353 and stepper motors (μm -range) as with servomotors.

The FM 353 is used in the SIMATIC S7-300, C7 and ET 200M on S7 masters. Operation with WinAC Slot 41x is possible.

**Positioning Module FM 354**

For servomotor control

Overview

▼ Description

Description

The FM 354 is a positioning module for position control with servomotors. It is suitable for precise position control in the face of load fluctuations or high dynamic response.

The FM 354 can be used for simple point-to-point positioning as well as for complex velocity profiles with the very highest demands on dynamic response, precision and speed, and is thus the ideal solution for positioning tasks with high clock-pulse rates both in mini controls and in distributed configurations directly on the machine.

The FM 354 is used in the SIMATIC S7-300, C7 and ET 200M on S7 masters as well as with WinAC Slot41x.



Controller Module FM 355

The universal control solution

Overview

▼ Description

Description

The FM 355 closed-loop control module has 4 channels, and can be used universally for temperature, pressure, flow and level control.

The FM 355 controller module for SIMATIC S7-300, C7 and ET 200M (on S7-300-/S7-400 masters) can also be used in PCS7 systems.

It offers:

- A variety of pre-programmed control structures
- A self-optimisation function for fast, simple controller optimisation and
- A backup function for problem-free operation in the event of a CPU stop/failure.



The module is available in two versions for controlling different actuators

- FM 355 C as continuous controller for the control of analog actuators
- FM 355 S as step-action or pulse-action controller for the control of actuators via digital outputs

Controller Module FM 355-2

The expert in temperature control

Overview

- ✓ Description
- ✓ Advantages

Description

The FM 355-2 is the specialized, backup-capable controller module for temperature control with integrated online self-optimization for SIMATIC S7-300, C7 and ET 200M.

It has four channels, and was specially developed for the control of temperature systems, whereby controllers with heating and active cooling can also be implemented and optimized.

Other loops with similar requirements can also be controlled.

**Advantages**

- Controller structures are, for the most part, preprogrammed and configurable, i.e. function blocks can be combined and interconnected as needed.
- All functions essential to temperature control are integrated
 - o Control algorithms specifically developed for temperature
 - o Integrated online self-optimization
 - o Control parameters are available at the step response turning point
 - o Best available support for thermocouples
- Backup capability
- Enhanced precision when using thermocouples
- Type B, E, J, K, R, S thermocouple connectable
- Temperature compensation when using thermocouples:
 - o via external Pt100;
 - o internal temperature sensor;
 - o parameterized value.
- Graphic configuring with user-friendly parameter assignment screen forms
- Extensive test functions as well as simulation options are available, and simplify commissioning and diagnostics

آشنایی با ماژول های CP

توسط این ماژول ها، CPU می تواند به شبکه های مختلف متصل شود. در واقع این ماژول های کارت های شبکه برای CPU محسوب می شوند.

Description	Bus system/communication network
CP 340	Point-to-point link
CP 341	Point-to-point link
CP 342-5	PROFIBUS DP
CP 342-5 FO	PROFIBUS DP (fiber optic)
CP 343-1 Lean	PROFINET / Industrial Ethernet
CP 343 1	PROFINET / Industrial Ethernet
CP 343-1 Advanced	Advanced PROFINET / Industrial Ethernet
CP 343-2	AS-Interface (Master)
CP 343-2P	AS-Interface (Master)
CP 343-5	PROFIBUS FMS

Point-to-point link via communication processors

Overview

✓ Description

Description

Point-to-point link via communication processors (CPs) is an extremely powerful and low-cost alternative to bus systems. The advantage of point-to-point links over bus systems is especially significant when only a few (RS 485) devices are to be connected to the SIMATIC S7. The CPs can also easily link third-party systems to the SIMATIC S7. Thanks to the great flexibility of the CPs, different physical transmission media, transmission rates or even customized transmission protocols can be implemented.

The interface modules for the S7-300 are available in three versions, each with one interface for the different physical transmission media.

All point-to-point communications modules use up to 4 protocol drivers which are integrated in the operating system. The CPs, like all other S7 modules, are parameterized from a programming device or PC via the central processing unit (CPU).



SIMATIC S7-300 communication processors

- CP 340 - Low-cost entry version
- CP 341 - Powerful computer connection, loadable protocols

Communication for SIMATIC S7-300

Modular, powerful and with full PROFIBUS functionality: Communication modules for SIMATIC S7-300

CP 342-5

for connecting S7-300, the modular controller for innovative solutions in the manufacturing industry, to PROFIBUS

- Expansion of the process I/O of SIMATIC S7-300 using several PROFIBUS interfaces
- Use in the field as DP-V0 master or DP-V0 slave
- Use in the cell area with cross-network programming device/operator panel communication through S7 routing
- Extensive operator control and monitoring thanks to multiplex function with OP communication
- Flexible use of the process I/O by means of dynamic activation of DP slaves



CP 342-5 FO

Fiber-optic technology is used when

- the environment is severely affected by electromagnetic interference,
- extreme potential differences exist and
- high transmission rates are required.

AS-Interface Master for SIMATIC S7-300

The AS-i master CP343-2 (P) facilitates the connection of AS-i actuators and sensors to the SIMATIC S7-300 or to the ET 200M distributed I/O system. Multiple AS-i masters and AS-i networks can be connected to the SIMATIC S7-300. Up to 62 slaves or up to 992 I/O points can be integrated per AS-i network. Engineering is comfortably realized via drag & drop in the TIA Portal or in SIMATIC STEP 7.



Communication processors for SIMATIC-S7 300

With our comprehensive portfolio, we offer the right CP for every requirement, as lean, standard or added value versions for SIMATIC S7-300, the modular controller for innovative system solutions in the manufacturing industry. Connecting controllers via these CPs permits direct integration into Industrial Ethernet networks. The communication processors have multi-protocol capability and support the following protocols:

- PG/OP communication
- S7 communication
- open communication (SEND/RECEIVE)
- PROFINET communication
- IT communication

CP 343-1 Lean: Connects even small applications to Industrial Ethernet

The low-cost connection for programming devices and HMI devices, as well as for communication between S7 programmable controllers.

➤ Product selection and ordering

**CP 343-1: The tried and tested, rugged and secure standard versions**

The efficient, reliable and rugged communication processor that links the S7-300 via Industrial Ethernet, also as a PN IO controller for the connection of PROFINET.

➤ Product selection and ordering

**CP 343-1 Advanced: From PROFINET to IT communication**

In addition to the functionality of the CP 343-1, the CP 343-1 Advanced offers e-mail functions and user-created web pages and thus optimally supports maintenance and quality assurance. The Internet functions such as FTP even allow connection to the most diverse PC-based systems.

With its integrated security functions of firewall (Stateful Inspection), VPN protocol (IPSec) and protocols for data encryption such as FTPS and SNMPv3, the communications processor protects S7-300 stations and lower-level networks against unauthorized access, as well as protecting data transmission against manipulation and espionage by means of encryption. Therefore, for the S7-300, this CP is the bridge between the field level and the MES level.

➤ Product selection and ordering

**CP 343-1 ERPC: Connects SIMATIC S7-300 controllers with ERP / MES systems**

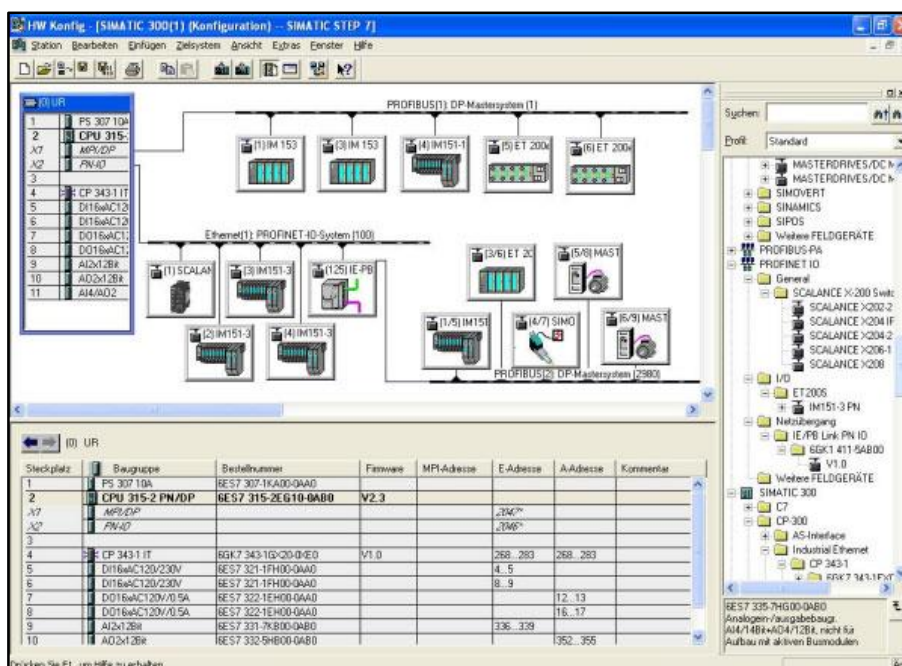
In combination with a loadable firmware expansion, this communication processor offers efficient access via Industrial Ethernet to databases across all corporate levels without laborious programming of the controller.

➤ Product selection and ordering



کار با نرم افزار

SIMATIC MANAGER



مراحل ساخت پروژه:

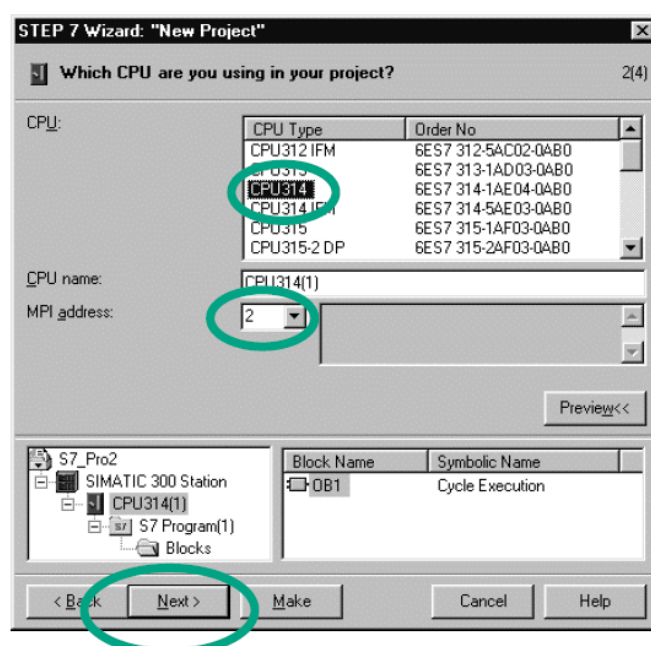
1- دابل کلیک بر روی آیکون نرم افزار



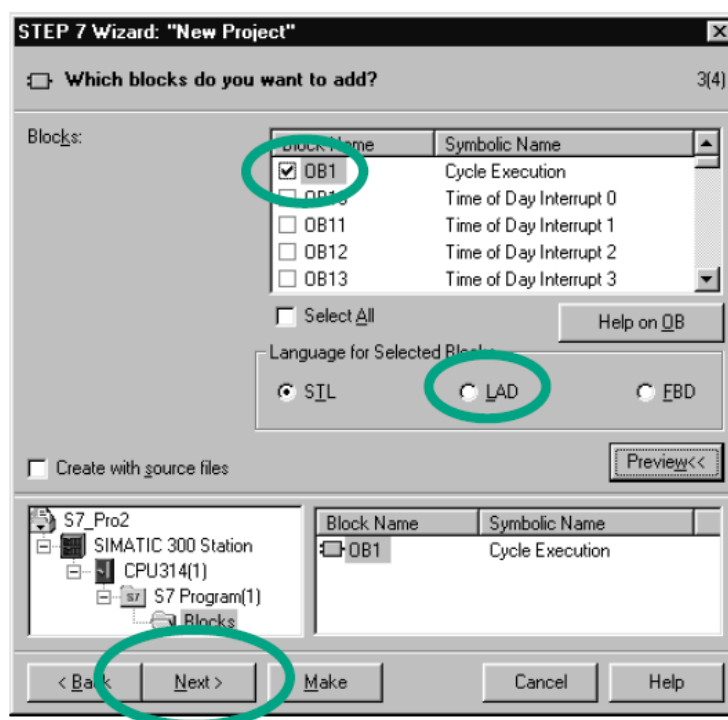
SIMATIC Manager



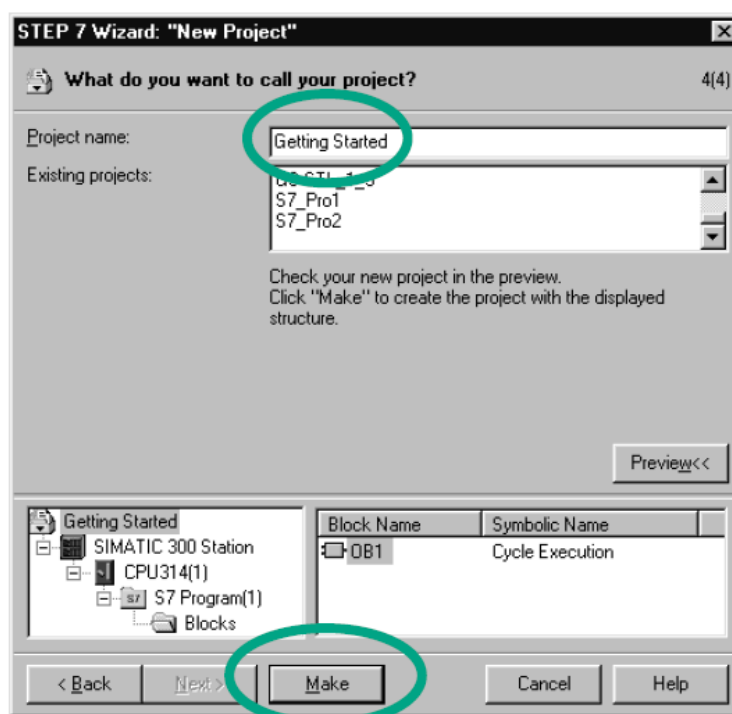
2- انتخاب نوع CPU و آدرس پورت MPI



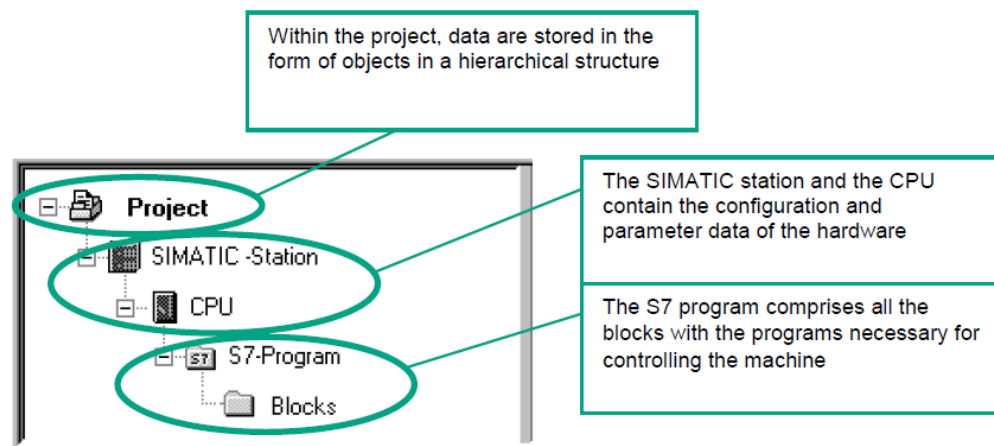
3- انتخاب بلوک برنامه نویسی و روش برنامه نویسی



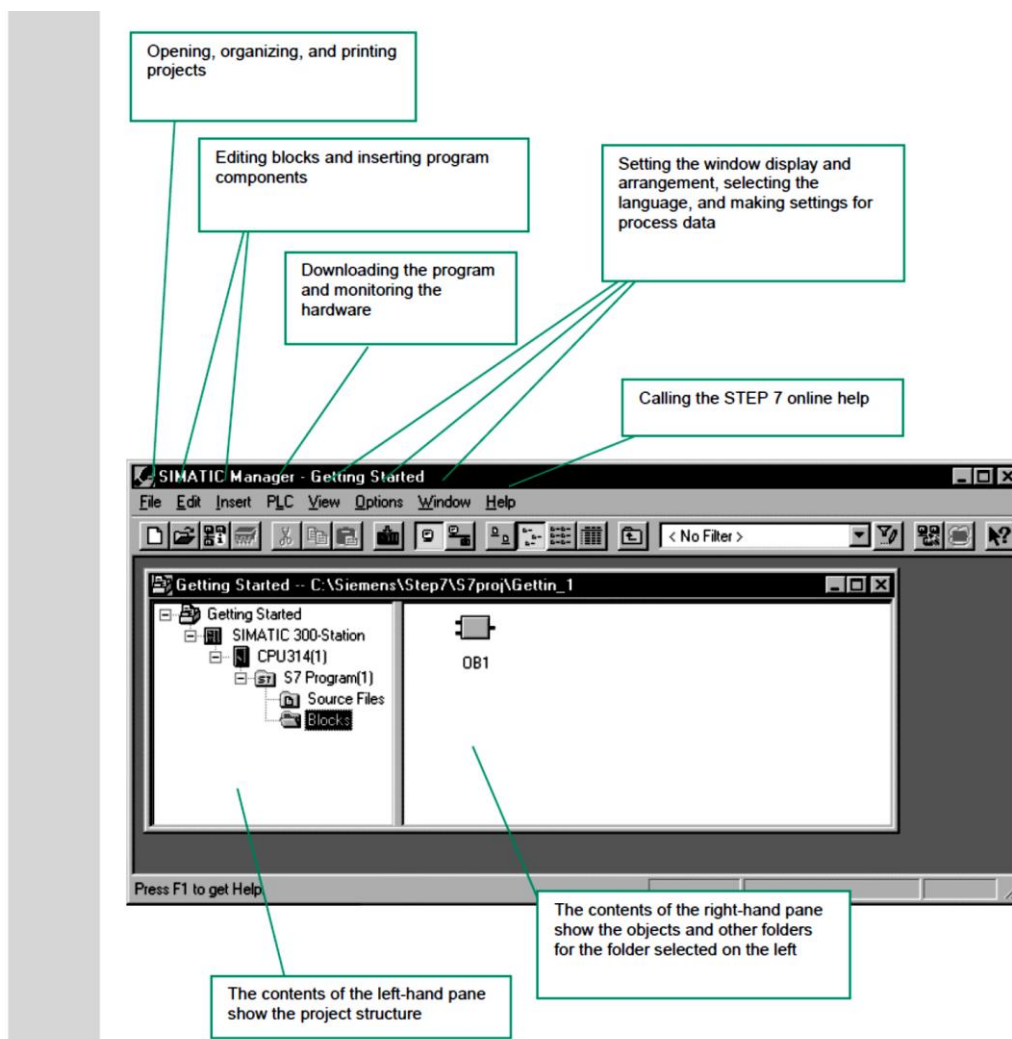
4- وارد کردن نام برای پروژه و کلیک بر روی گزینه FINISH



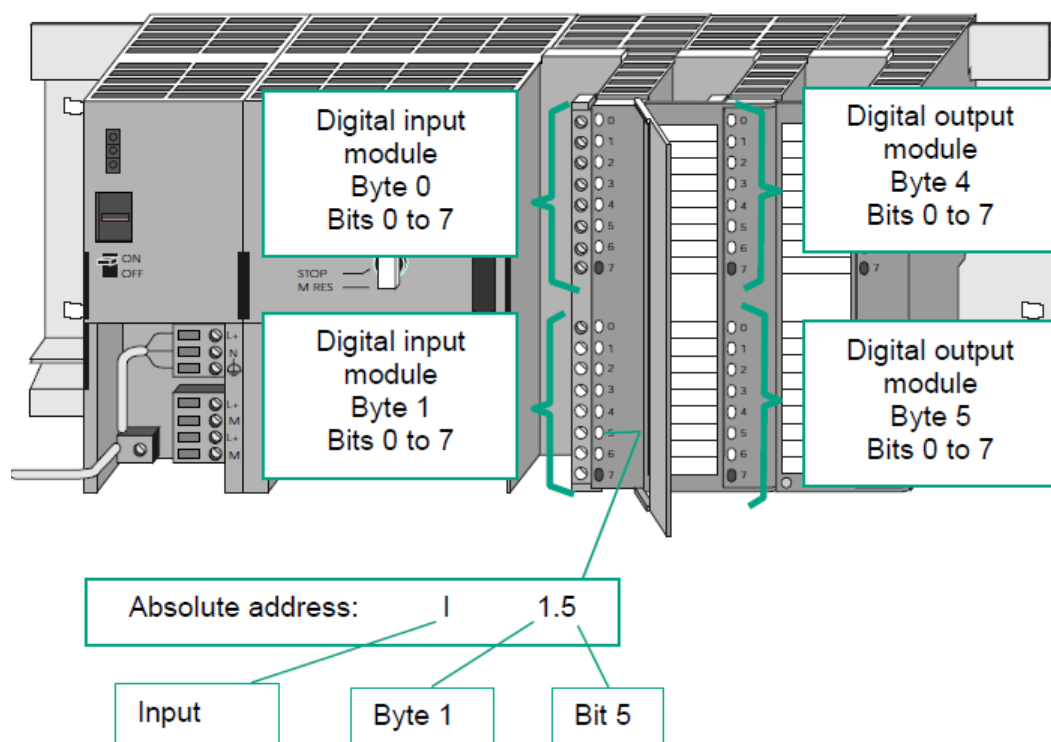
ساختار پروژه ایجاد شده



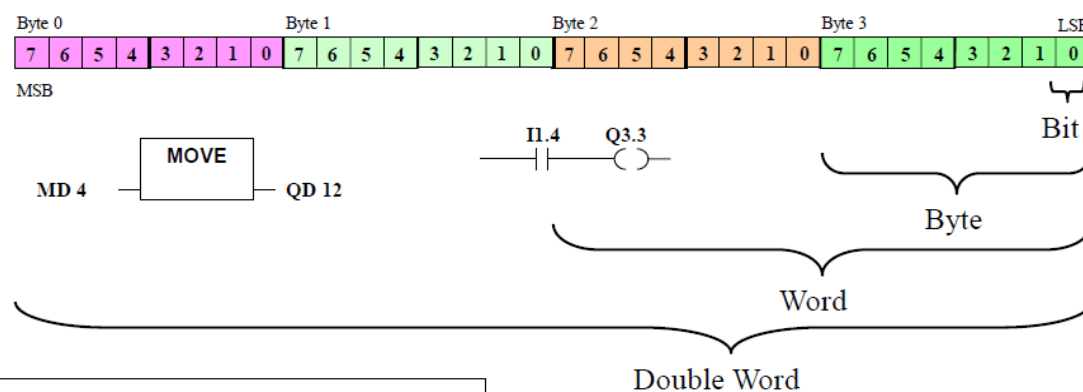
نمای کلی نرم افزار



نحوه آدرس دهی در S7



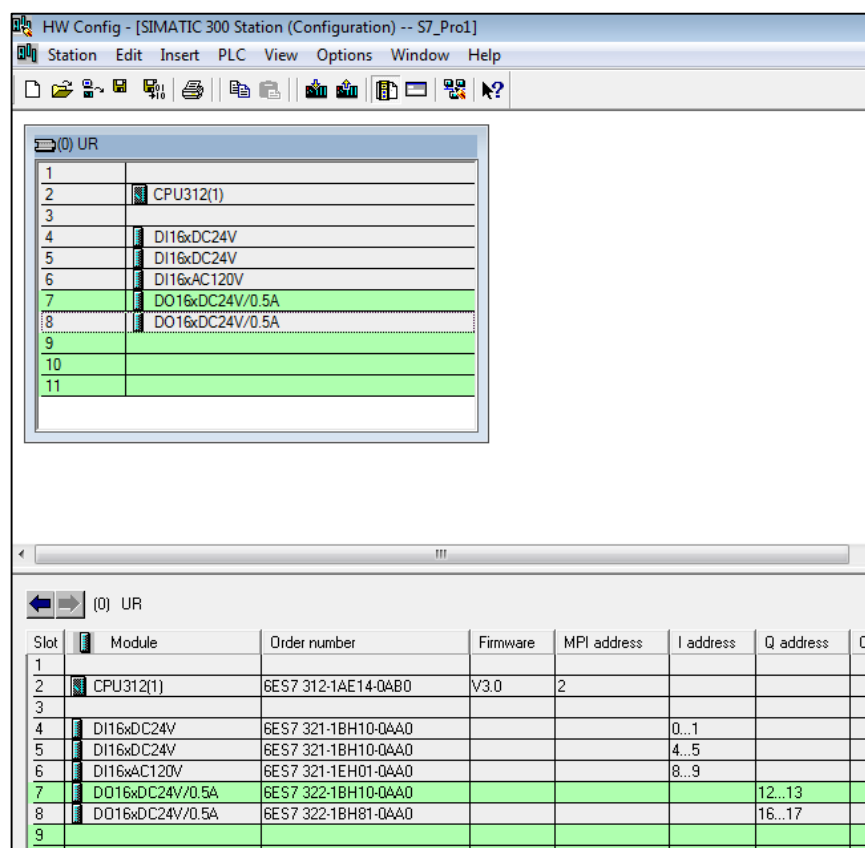
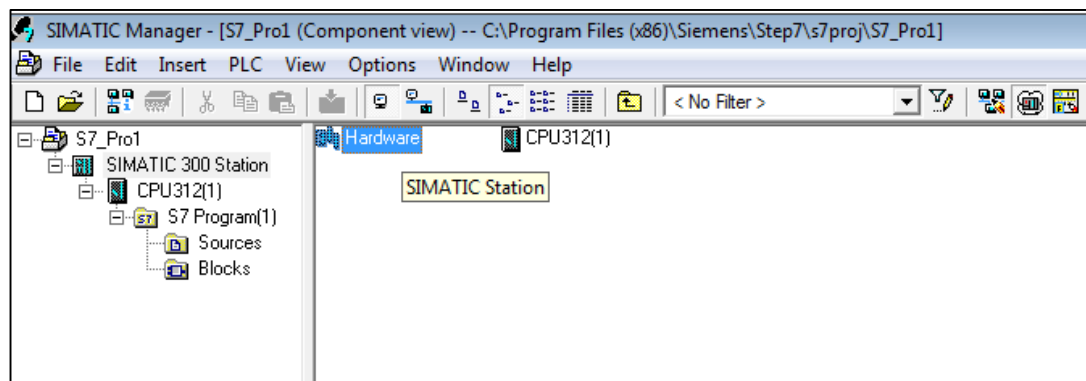
Addressing



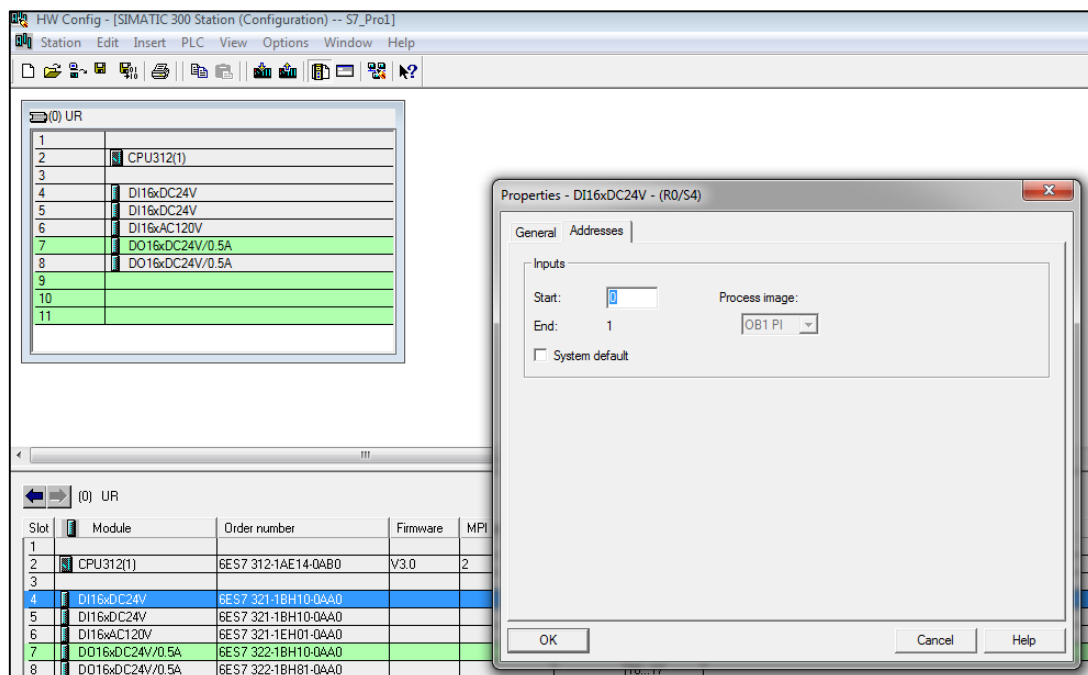
Control Statement		
A	I	3.4
Operational Part	Address Part	
A	I	3.4
	Address Identifier	Parameter
	I	3.4

Parameter = Byte Address and Bit Address

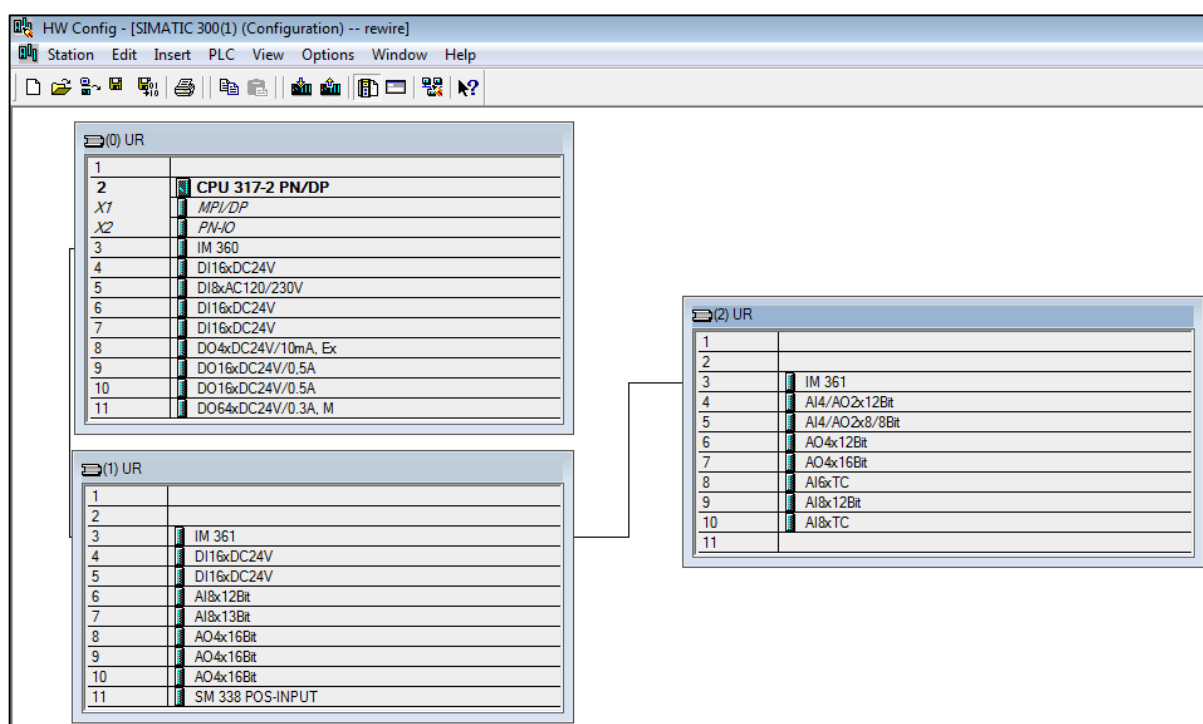
پیکربندی سخت افزار در محیط Hardware



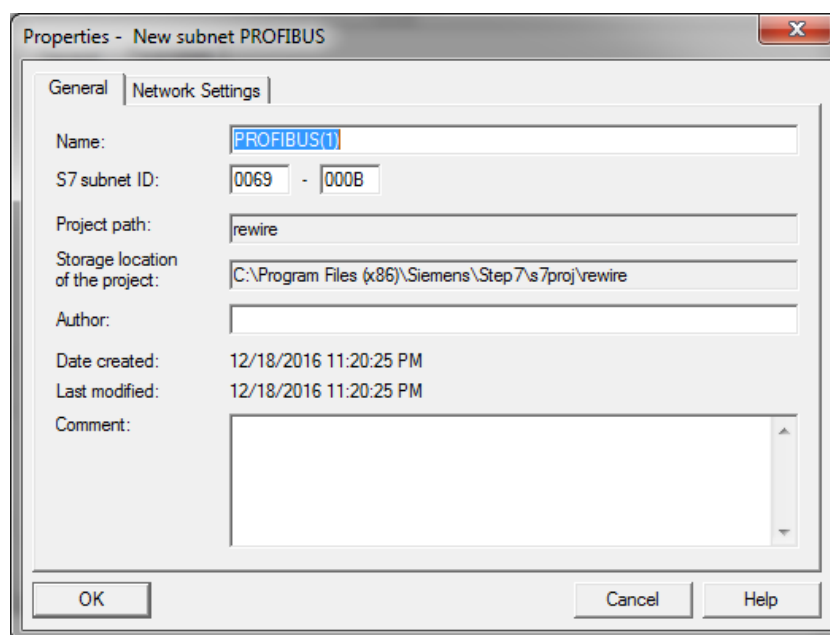
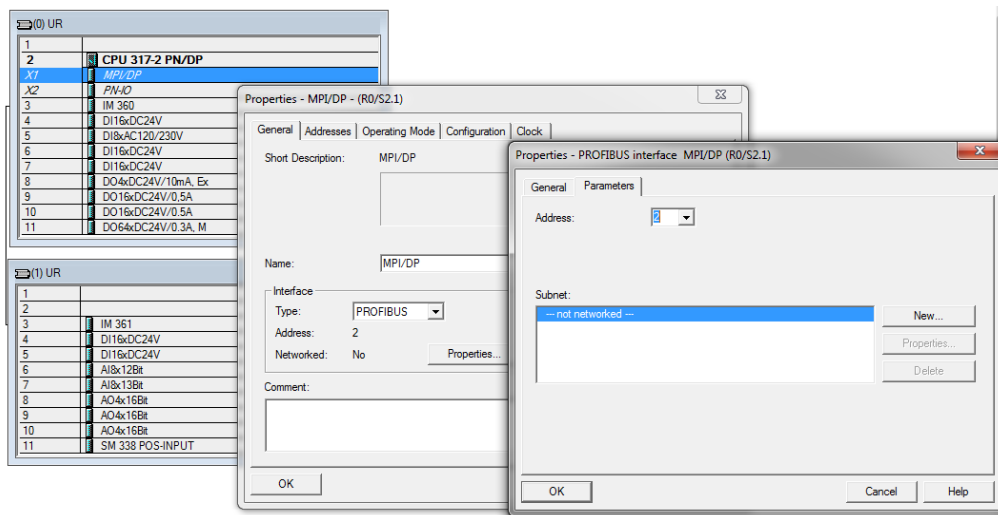
تغییر آدرس های ماژول های SM



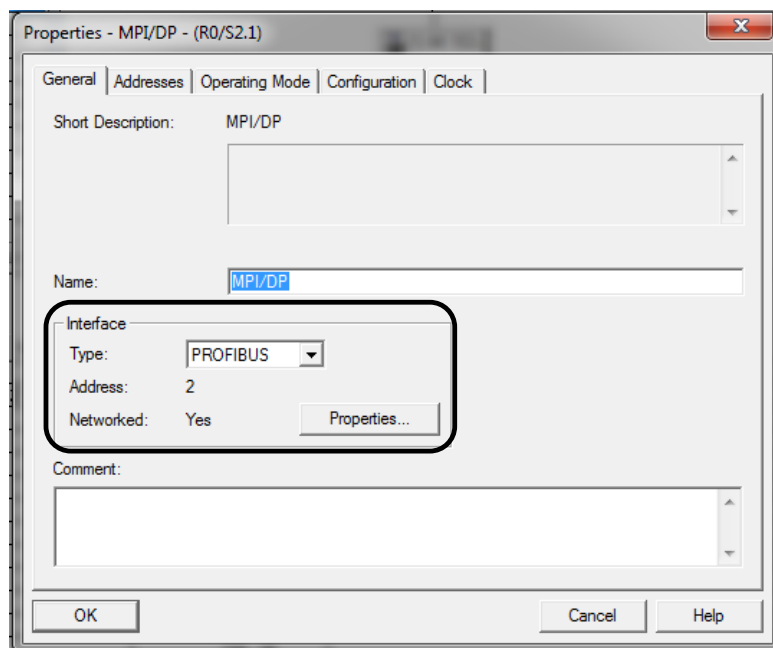
در PLC S7-300 توسط کارت IM360S و IM361R ظرفیت PLC به لحاظ تعداد کارت های SM, CP, FM تا سه رک اضافی در کنار رک اصلی می تواند توسعه پیدا کند. این تعداد ماکزیمم تعداد مجاز برای PLC S7-300 می باشد.



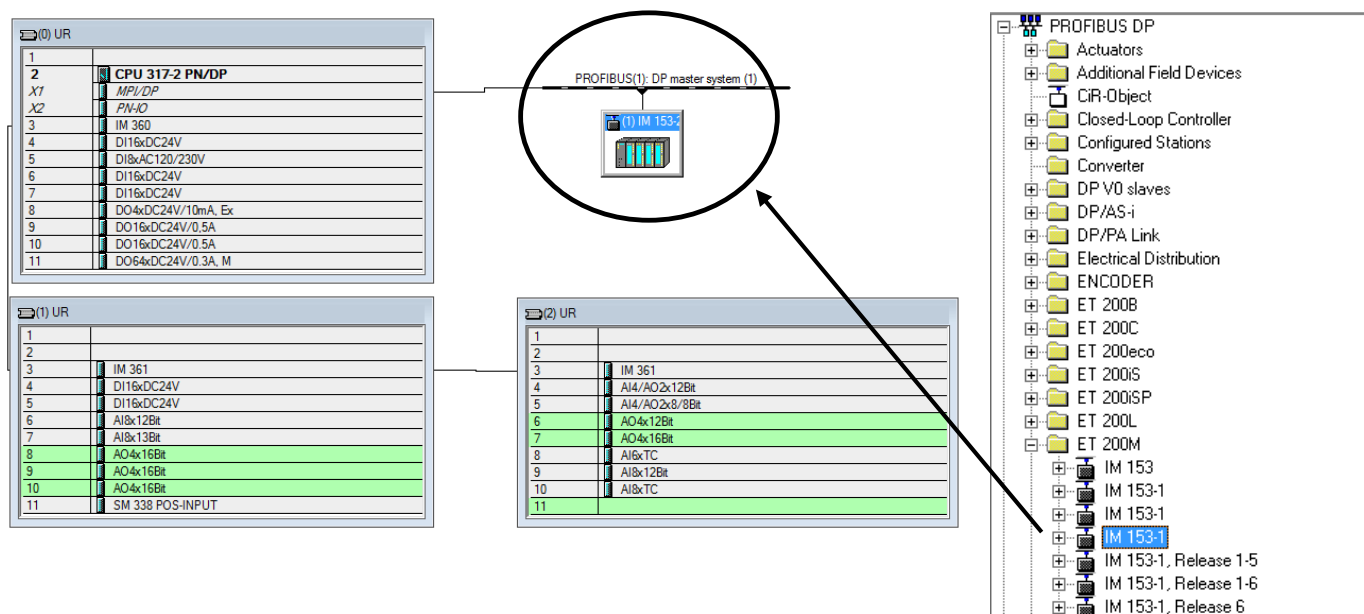
یکی از روش های افزایش تعداد بیشتر I/O و یا متمرکز کردن I/O ها در قسمت های مختلف یک سیستم، استفاده از ایستگاه های Remote I/O می باشد که در لیست تجهیزات زیمنس با نام ET معرفی می شوند. معمول ترین راه استفاده از ET ها قرار دادن این ایستگاهها در شبکه ی پروفیباس است و برقراری ارتباط با PLC توسط این شبکه است. جهت اتصال ایستگاه ET به شبکه پروفیباس ابتدا باید بستر این شبکه روی پورت شبکه PLC ایجاد شود.



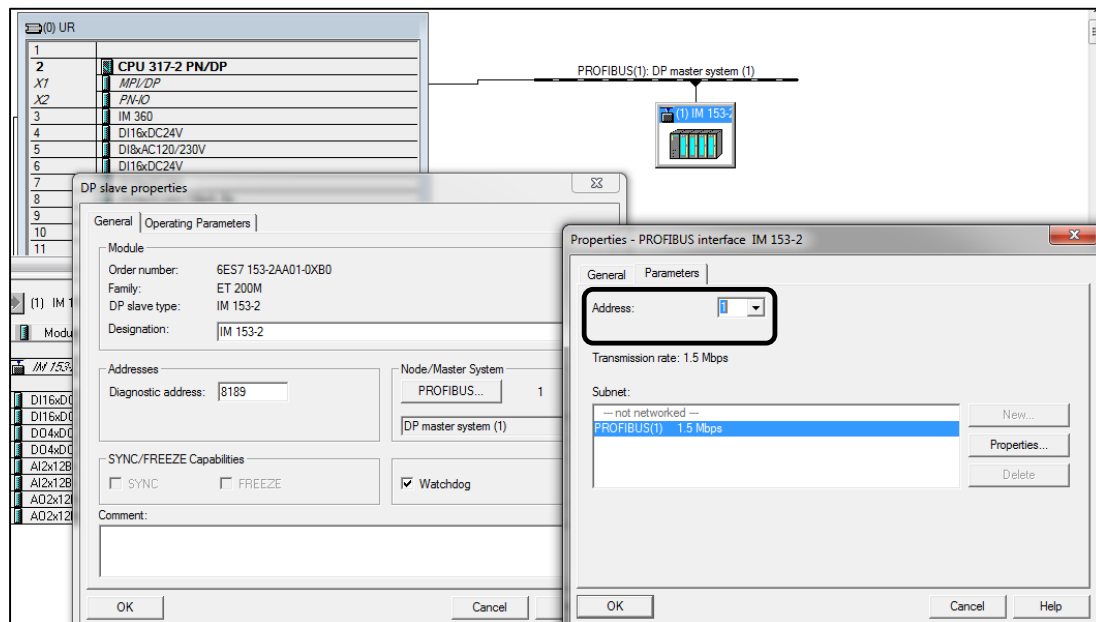
فعال شدن شبکه پروفیباس DP



در تصویر زیر نحوه ی اتصال یک ایستگاه ET200M در شبکه ی پروفیباس نشان داده شده است. جهت چیدمان کارت در ایستگاه ET از قسمت پایین نرم افزار استفاده می کنیم.

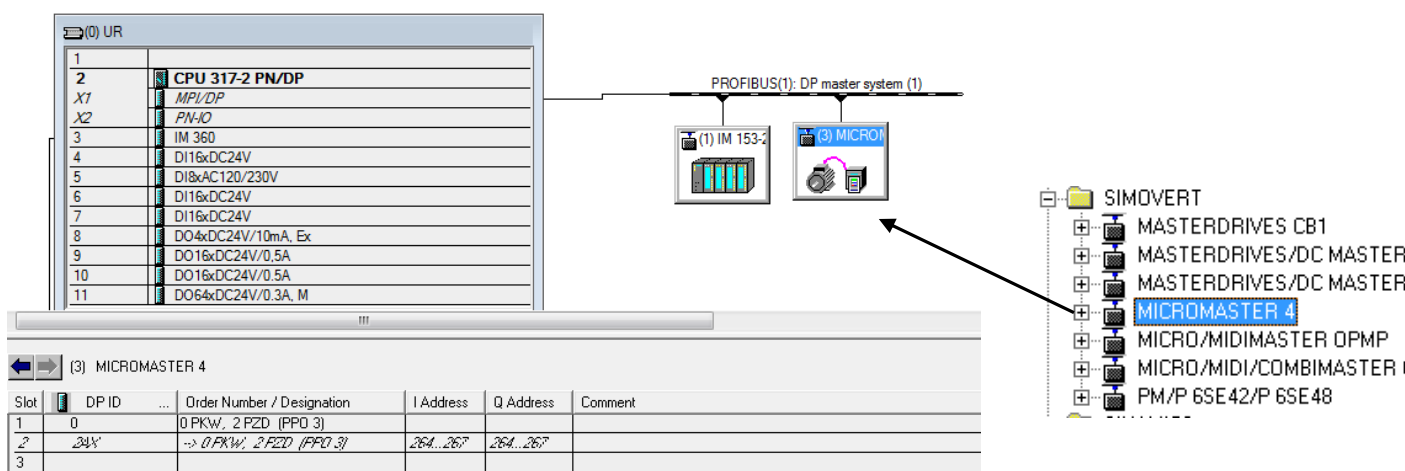


هر یک از ایستگاه های متصل شده به شبکه ی پروفیباس دارای آدرس منحصر به فرد هستند که آدرس Node شبکه نامیده می شود. جهت تعیین این آدرس داریم:



همچنین آدرس کارت های روی ET در ادامه ی کارت های ورودی و خروجی رک اصلی تنظیم می شود.

یکی دیگر از ایستگاه هایی که می تواند به شبکه پروفیباس متصل شود، درایو می باشد. به عنوان مثال جهت اتصال یک درایو میکرومستر زیمنس به شبکه ی پروفیباس از کتابخانه ی نرم افزار و در زیرمجموعه ی PROFIBUS DP گزینه MICROMASTER 4 را انتخاب می کنیم.



جهت تبادل دیتای درایو با PLC ، حتما نیاز به فضای PZD می باشد. از فضای PKW جهت دسترسی و تغییر پارامترهای درایو استفاده می شود.

تعاریف اولیه

نحوه پردازش CPU

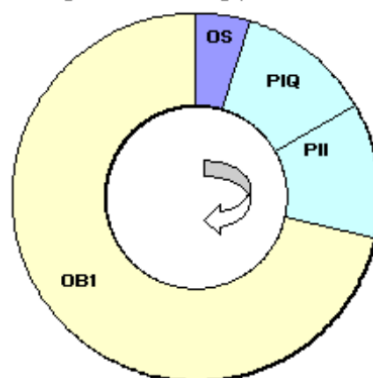
بعد از RUN کردن CPU در صورت وجود بلوک های Start Up ابتدا برنامه راه اندازی خوانده می شود. در ادامه مقادیر تصاویر PIQ به خروجی ها نوشته شده و وضعیت ورودی ها از کارت ها خوانده و در حافظه PII قرار می گیرد. پس از قرارگیری مقدار لحظه ای متغیرها در PII، برنامه نوشته شده در CPU اجرا می شود. در ضمن اجرای برنامه، CPU هر جا که لازم باشد اطلاعات را از جدول PII می خواند. مقدار فضای PII و PIQ برای هر CPU مقداری مشخص می باشد. به عنوان مثال CPU 313C دارای 128/128 بایت فضای Process Image می باشد. در ضمن اجرای برنامه اگر نیاز به ایجاد فرمانی جهت خروجی ها باشد، این فرمان در ناحیه PIQ قرار داده می شود. این فضا نیز بخشی از حافظه CPU می باشد که فرمان های ارسالی در آن ذخیره می شود. پس از اینکه اجرای برنامه تمام شد، CPU در ابتدای سیکل مقادیر موجود در PIQ را به کارت خروجی ارسال می کند.

PII مخفف Process Image Input می باشد و معرف بخشی از حافظه CPU است که ورودی ها در آن ذخیره می شوند.

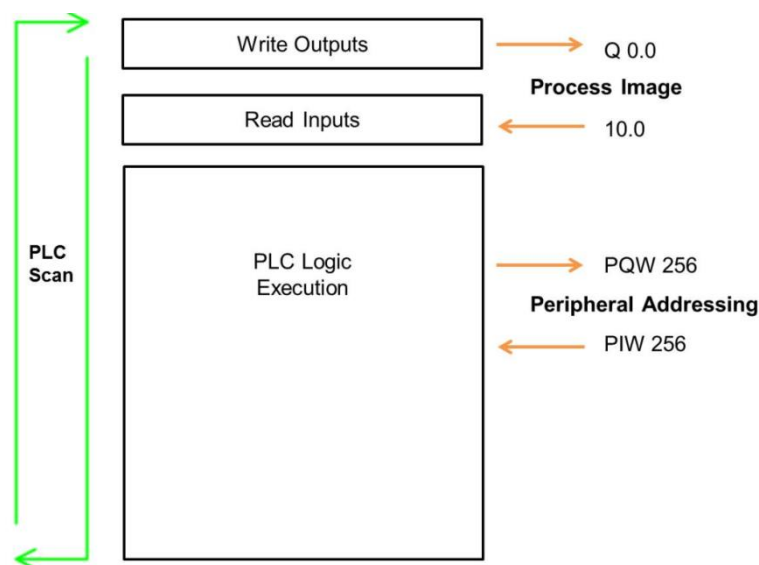
به طور مشابه برخی از خروجی ها بدون اینکه از PIQ فرمان بگیرند به طور مستقیم در صورت لزوم از CPU مقدار می گیرند. به این خروجی ها اصطلاحاً Peripheral output می گویند.

نحوه اسکن CPU بعد از سال 98 به صورت زیر تغییر یافت تا بعد از اینکه CPU از مد Startup عبور کرد، ابتدا مقادیر فضای PIQ را به کارت های خروجی ارسال کند.

Cyclic Program Processing (CPUs as of 10/98)

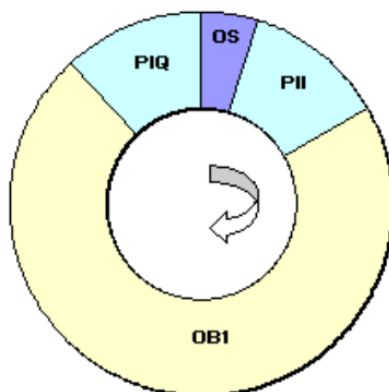


ادامه



قبل از سال 98 پروسه پردازش برنامه به صورت زیر انجام طراحی شده بود.

Cyclic Program Processing (CPUs up to 10/98)

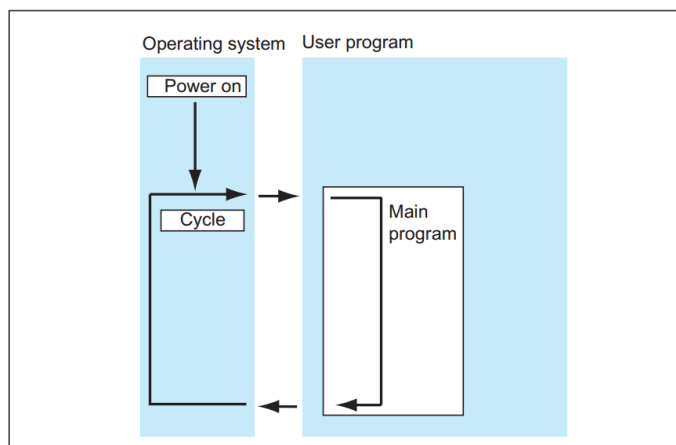


زمان سیکل اسکن

به مدت زمانی که طول می کشد تا یک سیکل کاری توسط CPU اجرا شود، زمان سیکل اسکن گفته می شود. این زمان از جمع زمان های زیر به دست می آید:

1- زمان مربوط به خواندن ورودی ها و قرار دادن آن در PII

2- زمان ارسال خروجی ها از PIQ به کارت خروجی

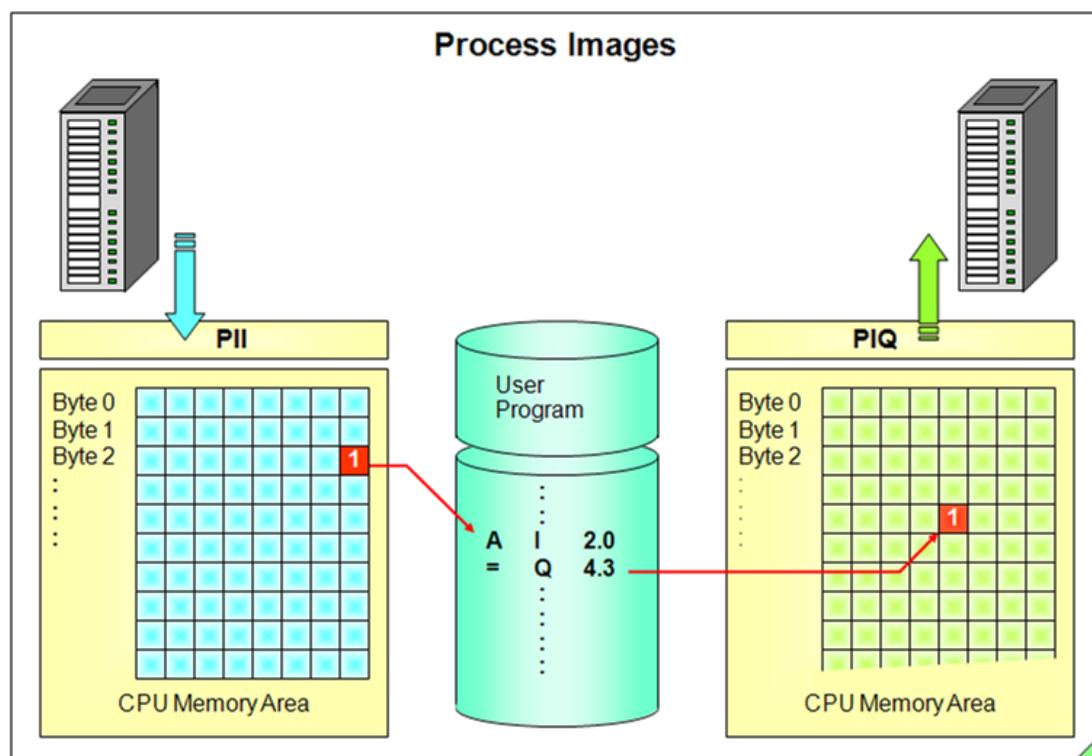


3- زمان مربوط به اجرای برنامه

4- زمان مربوط به سیستم عامل

5- زمان مربوط به تبادل دیتا با شبکه

6- وقفه های پیش آمده در زمان کار CPU



زمان مجاز سیکل اسکن

از آنجا که در طول زمان اجرای برنامه اصلی توسط CPU، مقادیر موجود در PII به روز رسانی نمی شوند، باید زمان سیکل اسکن زمان کمی باشد تا امکان به روز رسانی سریع PII و PIQ وجود داشته باشد. این زمان با توجه به پروسه های مختلف متفاوت می باشد. در هر صورت کاربر می تواند زمانی را به عنوان حداکثر زمان مجاز سیکل اسکن تعریف کند که در صورت افزایش زمان سیکل اسکن CPU از این زمان، یک وقفه به CPU اعمال شود.

این زمان به صورت پیش فرض 150 میلی ثانیه است که در محیط HW قابل تغییر می باشد.

در S7-400 امکان اختصاص زمانی وجود دارد که به عنوان حداقل زمان سیکل اسکن می باشد. در صورتی که زمان سیکل اسکن از مقدار تعریف شده کمتر باشد، OS سیکل اسکن بعدی را پس از سپری شدن زمان مورد نظر آغاز می کند. مثلاً اگر در یک PLC حداقل زمان سیکل اسکن 30 میلی ثانیه تعریف شده باشد و سیکل اسکن جاری 20 میلی ثانیه طول بکشد، OS به اندازه 10 ثانیه صبر می کند و سپس سیکل اسکن بعدی را اجرا می کند.

بخش های حافظه

Load Memory

شامل برنامه کاربر، تنظیمات پیکربندی سخت افزار و دستورات برنامه نویسی و سایر جزئیات می باشد.

Work Memory

حافظه کاری CPU بوده که برنامه اجرایی به آنجا منتقل می شود. از بین آنچه از برنامه ی کاربر که به Load Memory منتقل شده است، تنها بخش های اجرایی به Work Memory منتقل می گردند.

System Memory

در این بخش از حافظه، برنامه ذخیره نمی شود، بلکه مخصوص ذخیره سازی دیتاهای مختلف است. این بخش شامل:

❖ تصاویر PII

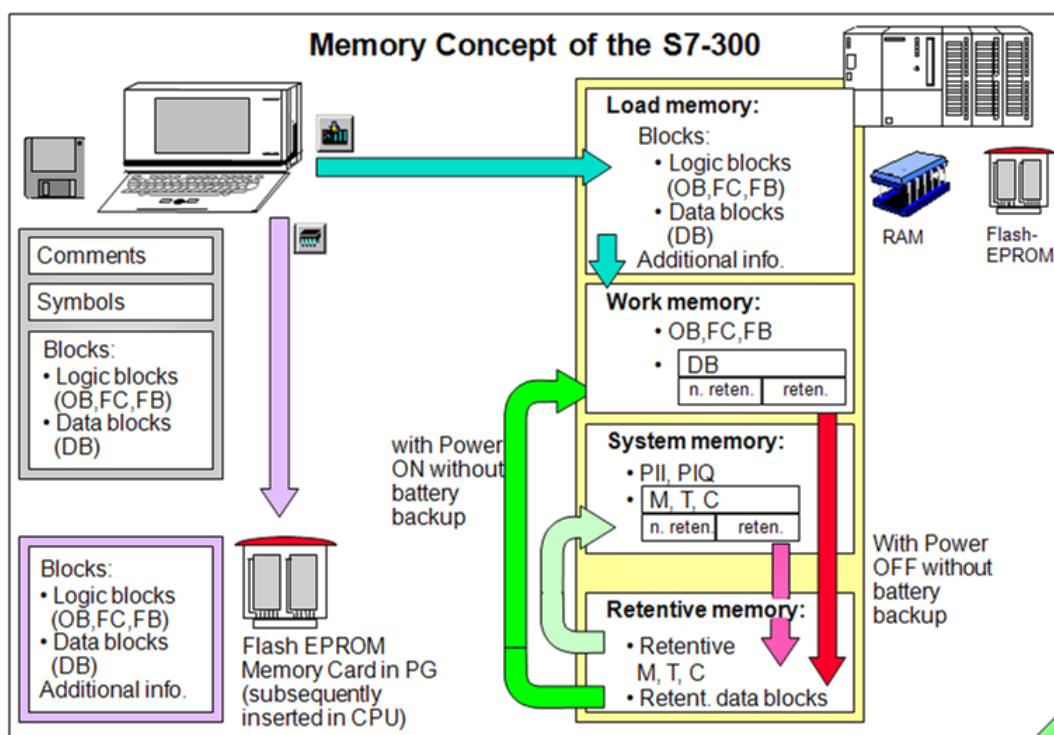
❖ تصاویر PIQ

❖ تایمرها

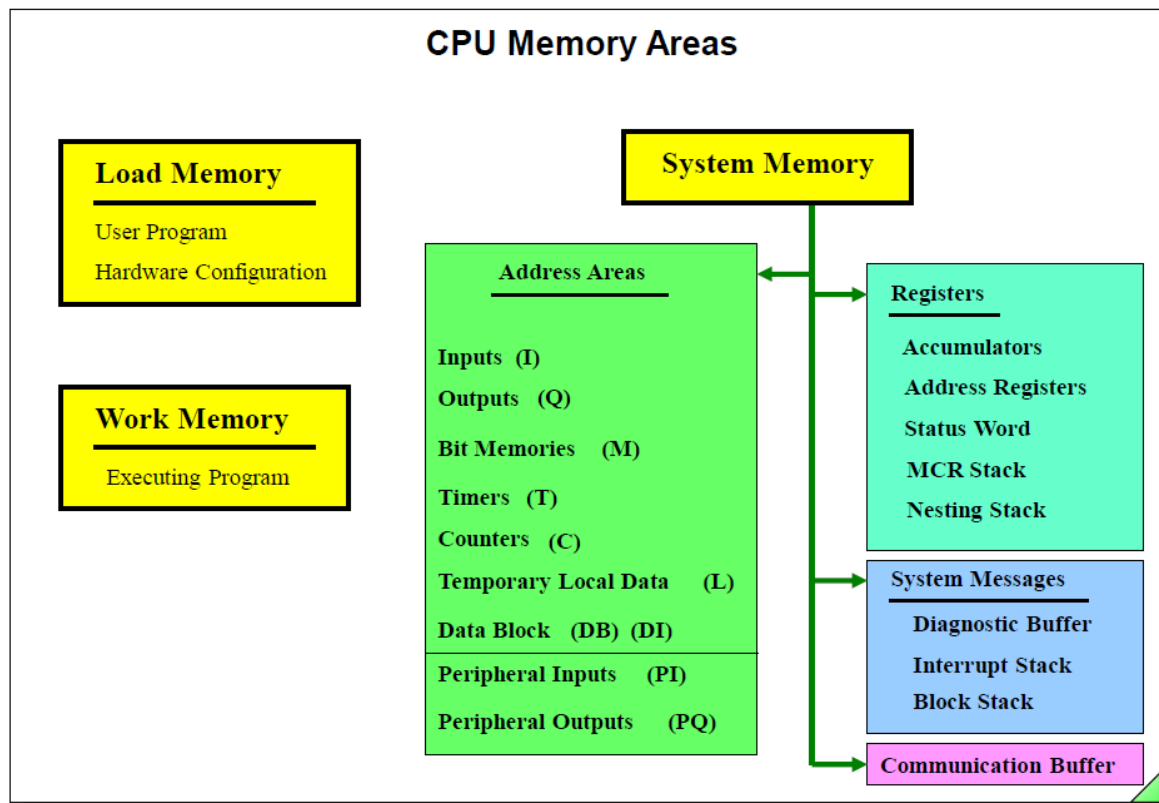
❖ کانترها

❖ حافظه M

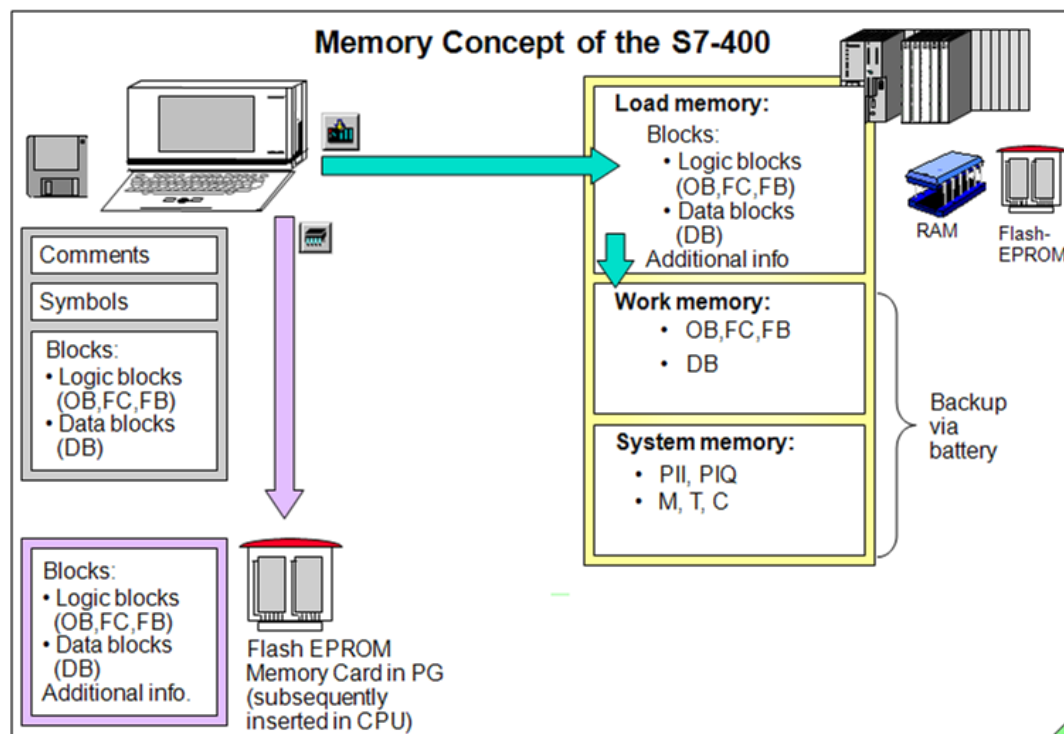
مفهوم حافظه در S7-300



فضاهای حافظه CPU



مفهوم حافظه در S7-400



بلوک های برنامه

Block Types

There are several different types of blocks you can use within an S7 user program:

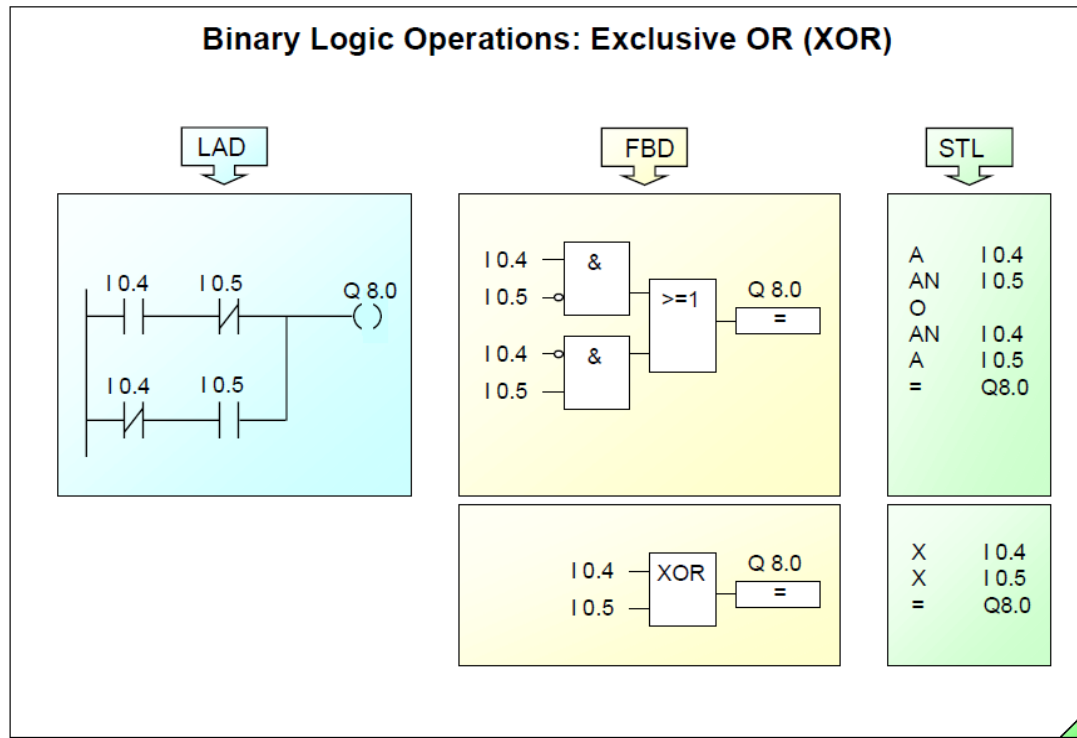
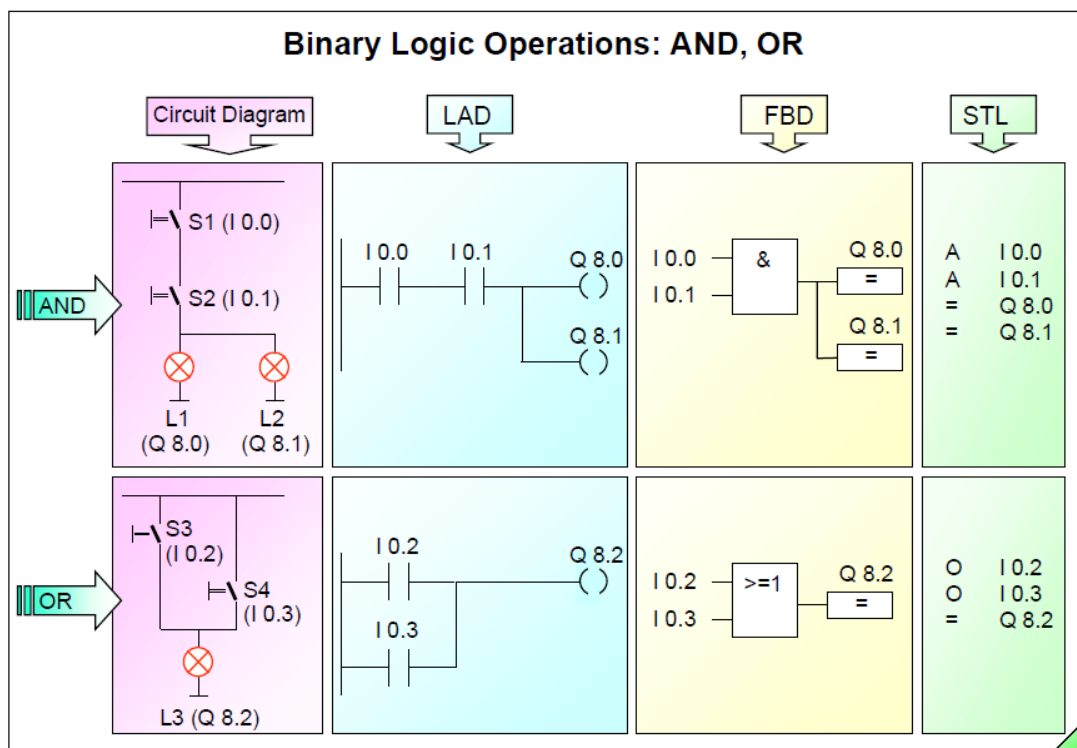
Block	Brief Description of Function	See Also
Organization blocks (OB)	OBs determine the structure of the user program.	Organization Blocks and Program Structure
System function blocks (SFB) and system functions (SFC)	SFBs and SFCs are integrated in the S7 CPU and allow you access to some important system functions.	System Function Blocks (SFB) and System Functions (SFC)
Function blocks (FB)	FBs are blocks with a "memory" which you can program yourself.	Function Blocks (FB)
Functions (FC)	FCs contain program routines for frequently used functions.	Functions (FC)
Instance data blocks (instance DB)	Instance DBs are associated with the block when an FB/SFB is called. They are created automatically during compilation.	Instance Data Blocks
Data blocks (DB)	DBs are data areas for storing user data. In addition to the data that are assigned to a function block, shared data can also be defined and used by any blocks.	Shared Data Blocks (DB)

بلوک های Interrupts

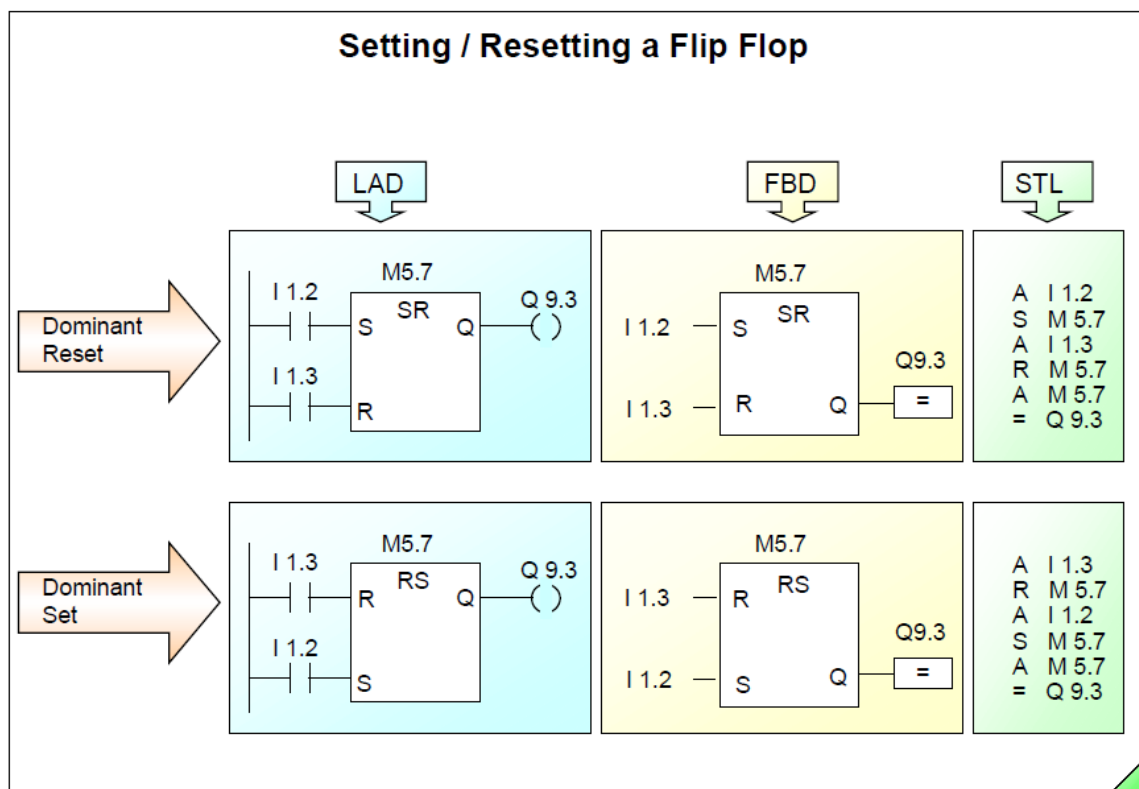
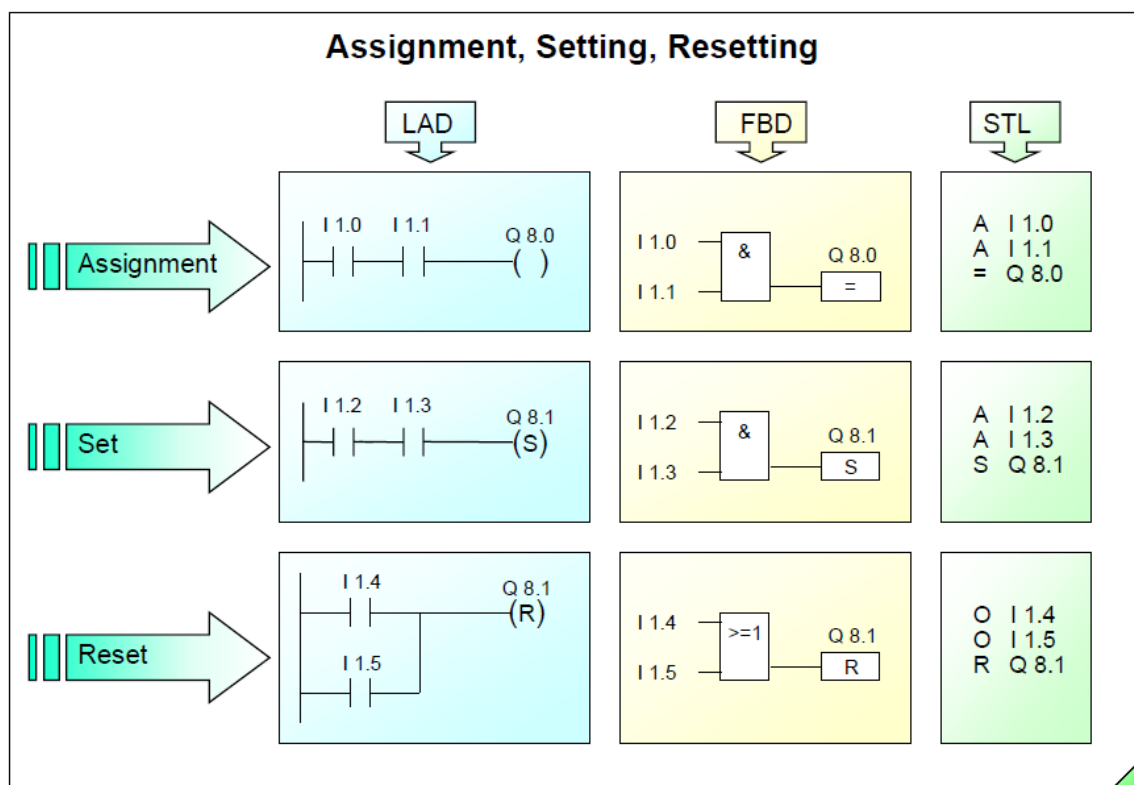
Type of Interrupt	Organization Block	Priority Class (Default)	See also
Main program scan	OB1	1	Organization Block for Cyclic Program Processing (OB1)
Time-of-day interrupts	OB10 to OB17	2	Time-of-Day Interrupt Organization Blocks (OB10 to OB17)
Time-delay interrupts	OB20 OB21 OB22 OB23	3 4 5 6	Time-Delay Interrupt Organization Blocks (OB20 to OB23)
Cyclic interrupts	OB30 OB31 OB32 OB33 OB34 OB35 OB36 OB37 OB38	7 8 9 10 11 12 13 14 15	Cyclic Interrupt Organization Blocks (OB30 to OB38)

Type of Interrupt	Organization Block	Priority Class (Default)	See also
Hardware interrupts	OB40 OB41 OB42 OB43 OB44 OB45 OB46 OB47	16 17 18 19 20 21 22 23	Hardware Interrupt Organization Blocks (OB40 to OB47)
DPV1 interrupts	OB 55 OB 56 OB 57	2 2 2	Programming DPV1 Devices
Multicomputing interrupt	OB60 Multicomputing	25	Multicomputing - Synchronous Operation of Several CPUs
Synchronous cycle interrupt	OB 61 OB 62 OB 63 OB 64	25	Configuring Short and Equal-Length Process Reaction Times on PROFIBUS-DP
Redundancy errors	OB70 I/O Redundancy Error (only in H systems) OB72 CPU Redundancy Error (only in H systems)	25 28	"Error Handling Organization Blocks (OB70 to OB87 / OB121 to OB122)"
Asynchronous errors	OB80 Time Error OB81 Power Supply Error OB82 Diagnostic Interrupt OB83 Insert/Remove Module Interrupt OB84 CPU Hardware Fault OB 85 Program Cycle Error OB86 Rack Failure OB87 Communication Error	25 (or 28 if the asynchronous error OB exists in the startup program)	Error Handling Organization Blocks (OB70 to OB87 / OB121 to OB122)
Background cycle	OB90	29 ¹⁾	Background Organization Block (OB90)
Startup	OB100 Restart (Warm start) OB101 Hot Restart OB102 Cold Restart	27 27 27	Startup Organization Blocks (OB100/OB101/OB102)
Synchronous errors	OB121 Programming Error OB122 Access Error	Priority of the OB that caused the error	Error Handling Organization Blocks (OB70 to OB87 / OB121 to OB122)
¹⁾ The priority class 29 corresponds to priority 0.29. The background cycle has a lower priority than the free cycle.			

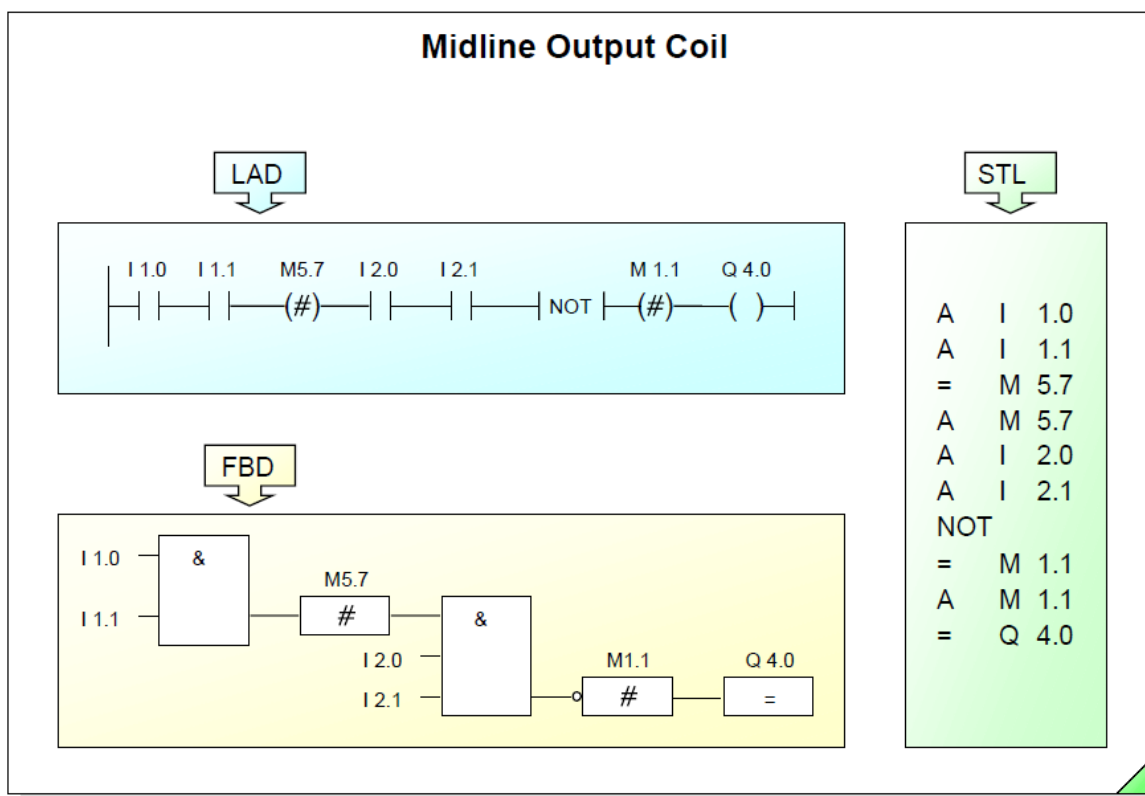
آشنایی با دستورات پرکاربرد در STEP 7



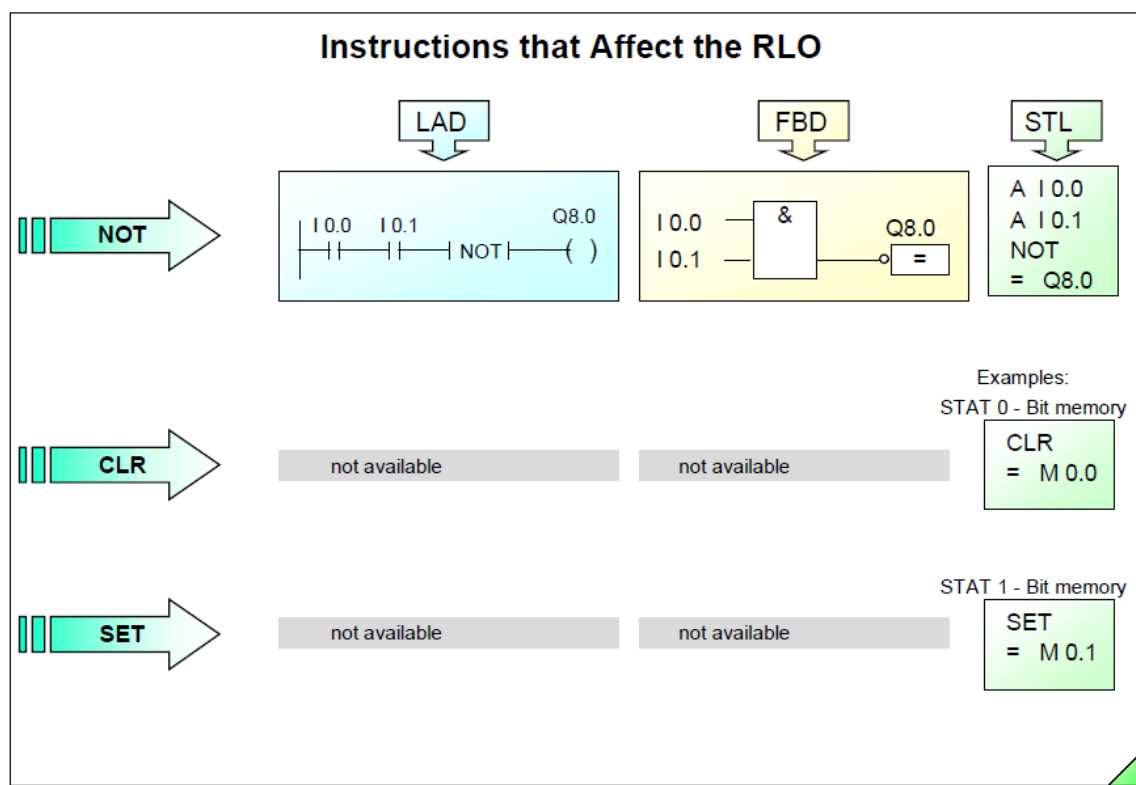
دستورات SET_RESET و فلیپ فلاپ



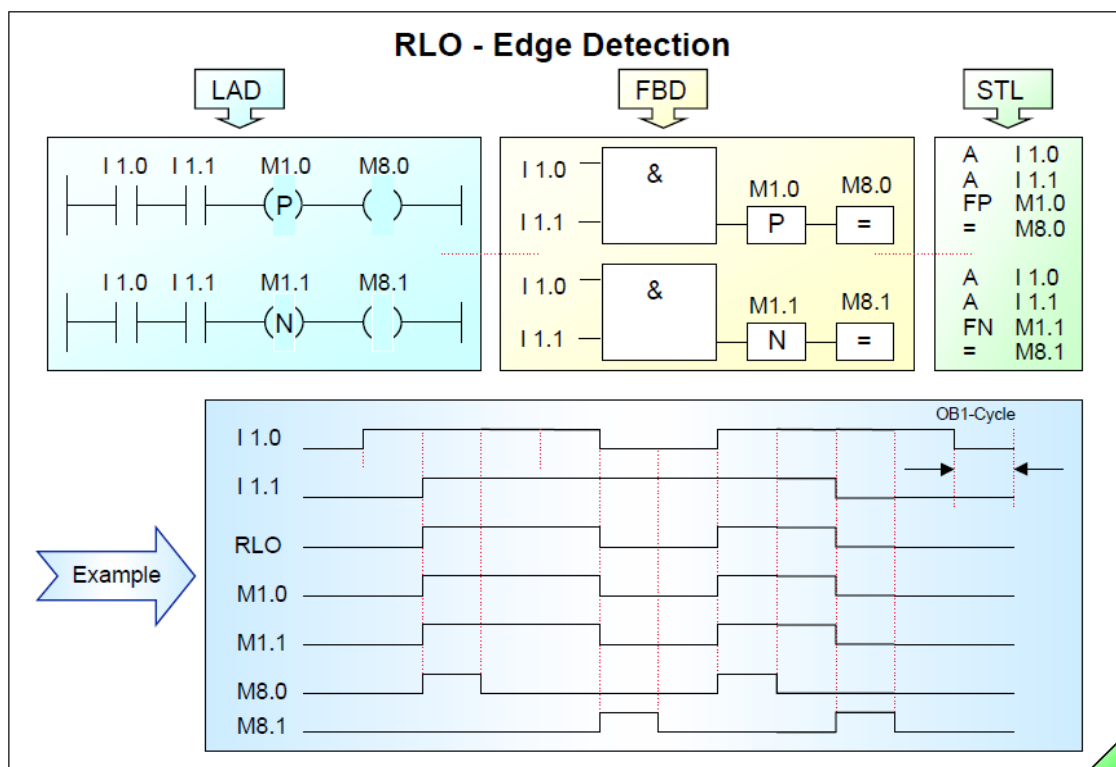
دستور کویل میانی



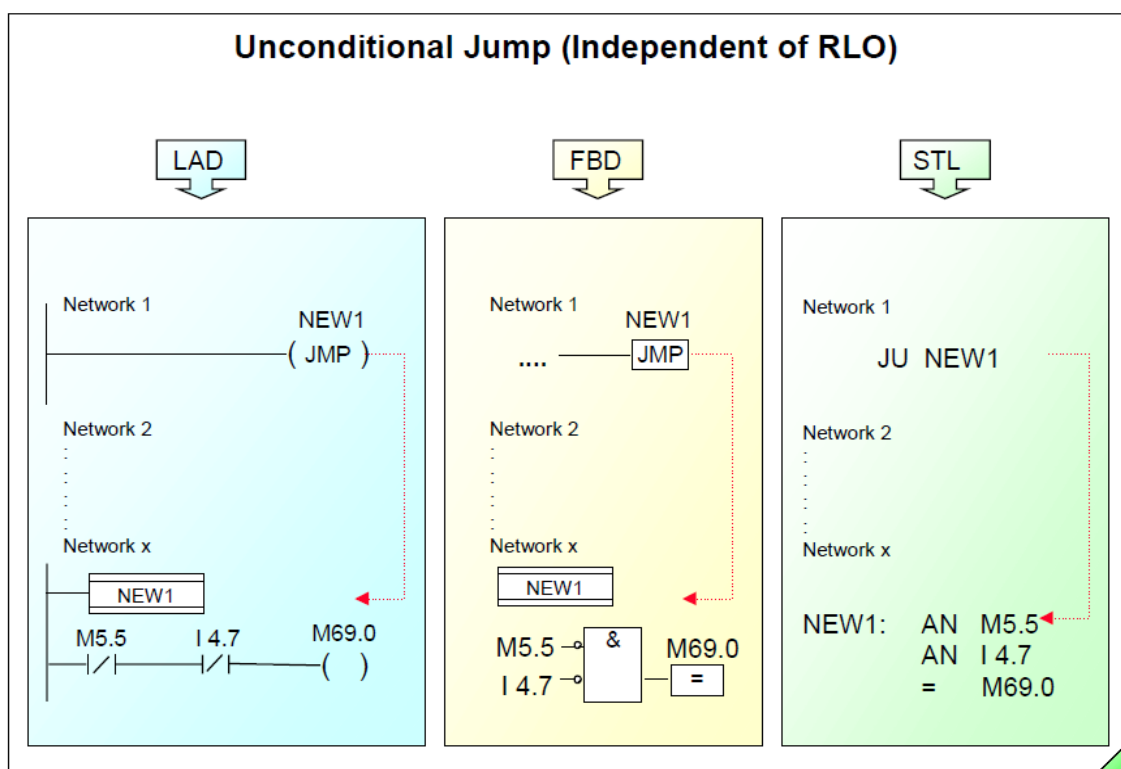
دستورات مرتبط با بیت RLO



دستورات تشخیص لبه



دستورات پرش



آشنایی با تایپ دیتاها

Integer (INT, 16-Bit Integer) Data Type

Value Range -32768 to +32767
(without sign: 0 to 65535)

Arithmetic Operations: such as + I, * I, <I, ==I

Display Formats:

DEC: + 662 **BIN.: 2#** 0 0 0 0 0 0 1 0 1 0 0 1 0 1 1 0 **HEX: W#160 2 9 6**

without sign

Sign positive numbers

6 x 16⁰ = 6
9 x 16¹ = 144
2 x 16² = 512
662

DEC: - 662 **BIN.: 2#** 1 1 1 1 1 1 0 1 0 1 0 1 0 1 0 1 **HEX: W#16 F D 6 A**

without sign

Sign negative numbers

Representation as twos complement

10 x 16⁰ = 16
6 x 16¹ = 96
13 x 16² = 3328
15 x 16³ = 61440
64874

Double Integer (DINT, 32-Bit Integer) Data Type

Value Range L# -2147483648 to L#+2147483647
(without sign: 0 to 4294967295)

Arithmetic Operations: such as + D, * D, <D, ==D

Display Formats:

DEC: L# +540809 **BIN.: 2#** 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 **HEX: DW#16#** 0 0 0 8 4 0 8 9

(without sign)

Sign positive numbers

DEC: L# -540809 **BIN.: 2#** 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 **HEX: DW#16#** F F F 7 B F 7 7

(without sign)

Sign negative numbers

Representation as twos complement

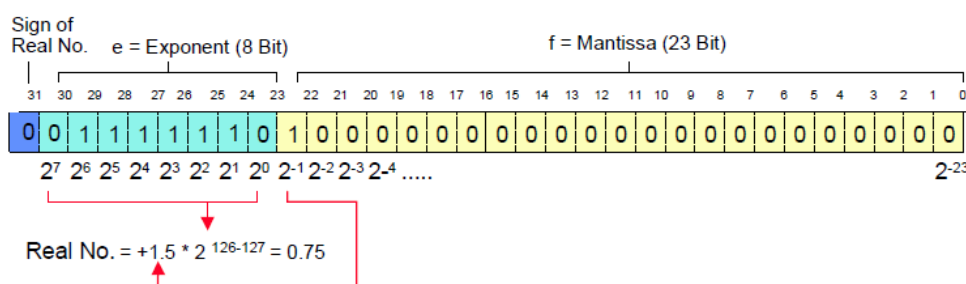
REAL (Floating-point Number, 32 Bit) Data Type

Value Range $-1.175495 \cdot 10^{-38}$ to $3.402823 \cdot 10^{+38}$

Arithmetic Operations: such as + R, * R, <R, ==R
sin, acos, ln, exp, SQR

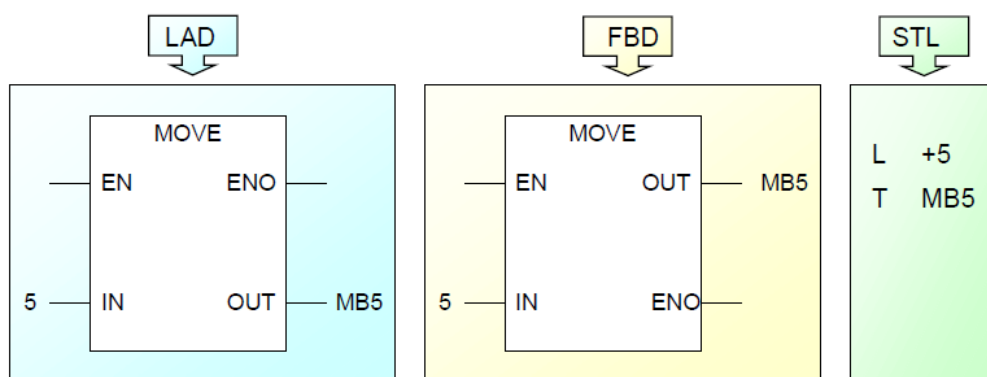
General Format of a Real Number = (Sign) • (1.f) • (2^{e-127})

Example: 7.50000e-001 ($7.5 \cdot 10^{-1} = 0.75$)



دستور انتقال MOVE

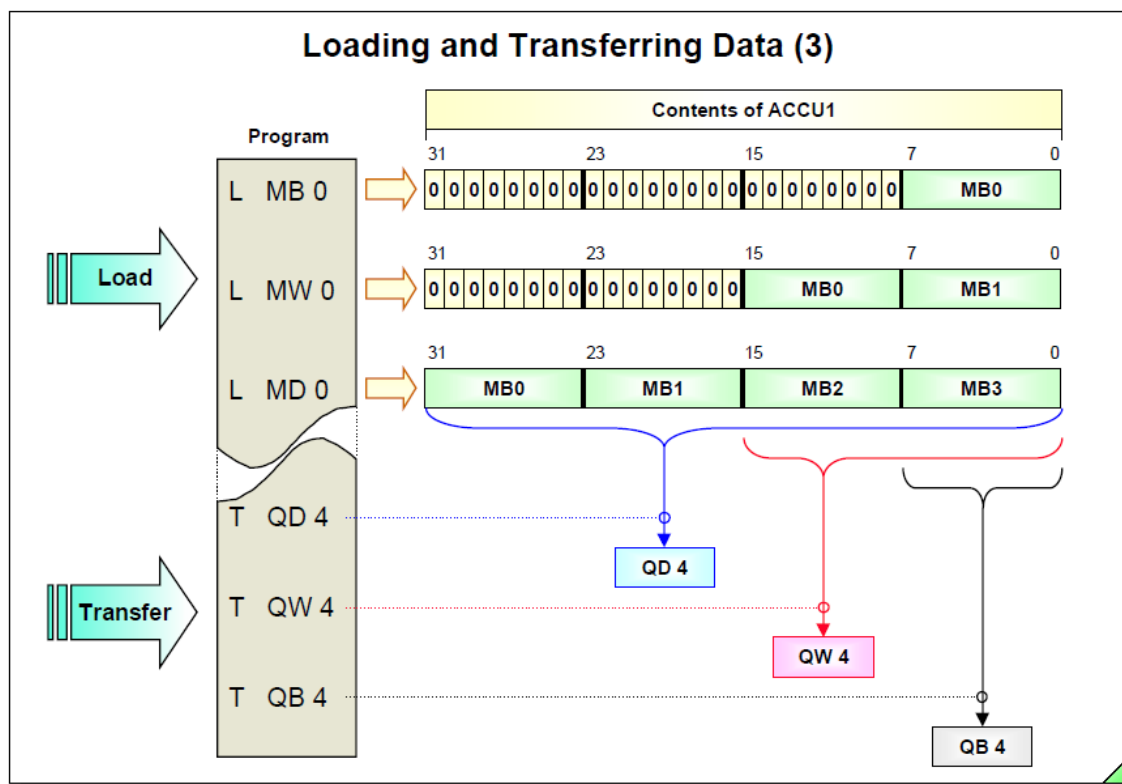
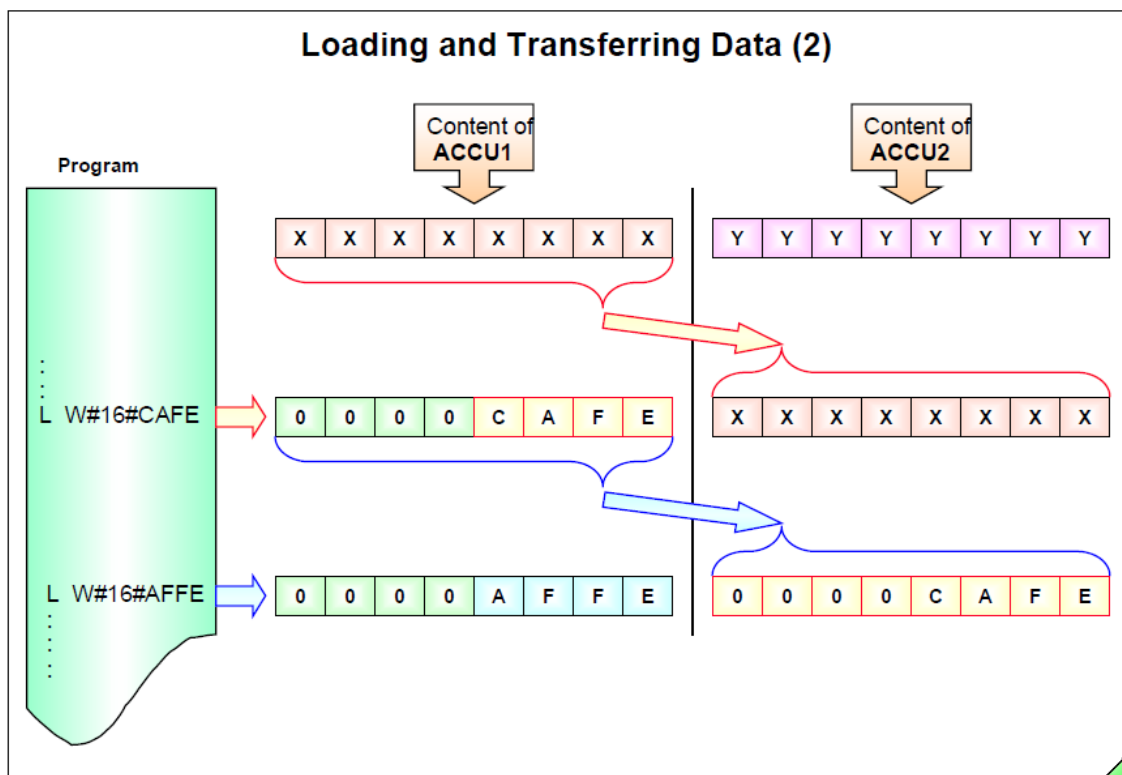
Loading and Transferring Data (1)



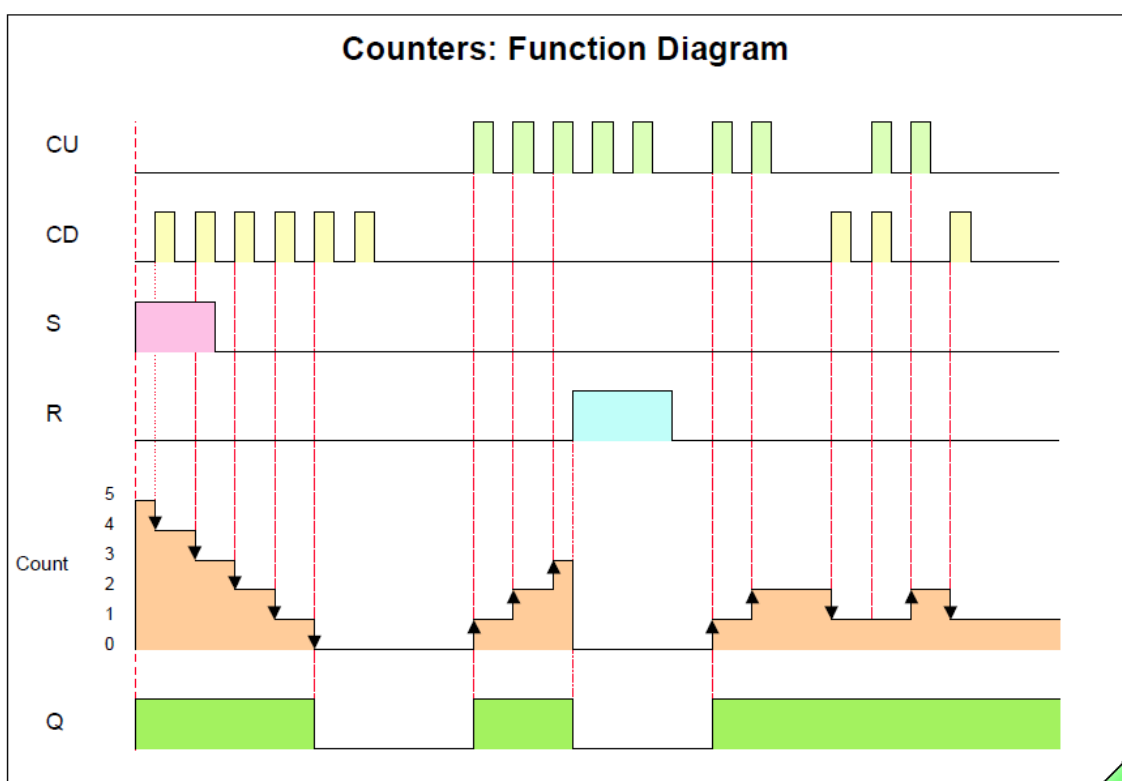
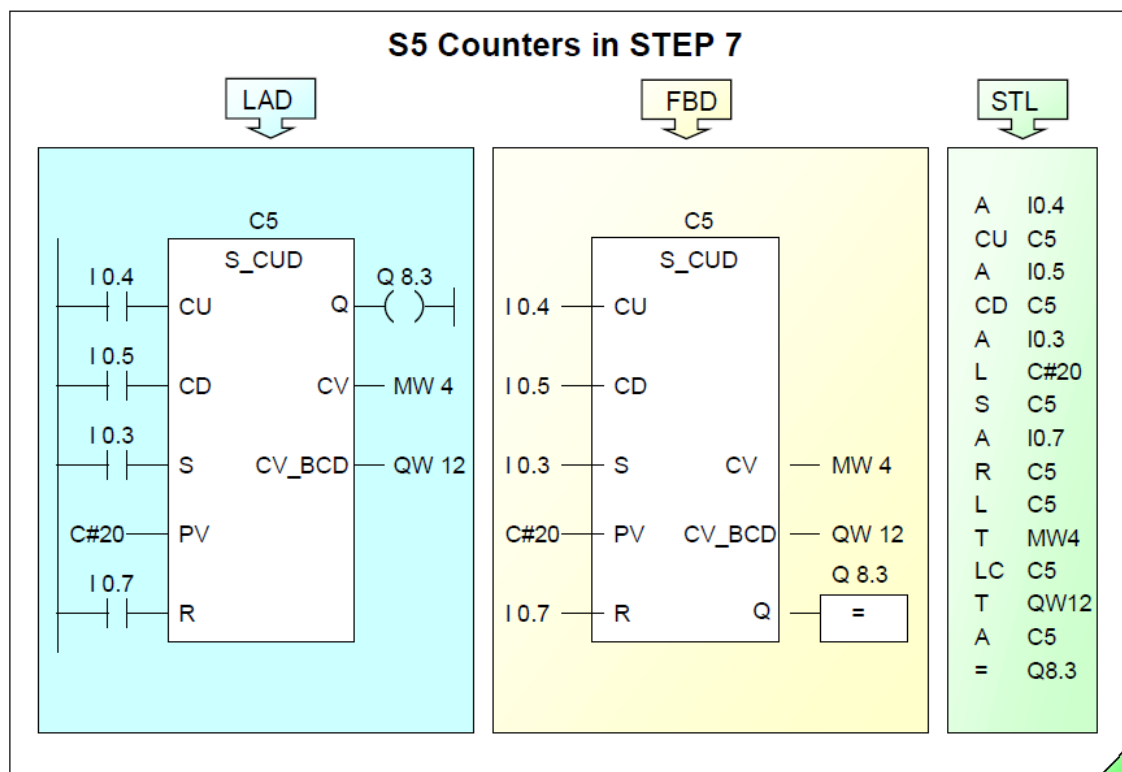
Examples of Load

L +5	//	16-bit constant (Integer)
L L#523123	//	32-bit constant (Double Integer)
L B#16#EF	//	byte in hexadecimal form
L 2#0010 0110 1110 0011	//	16-bit binary value
L 3.14	//	32-bit constant (Real)

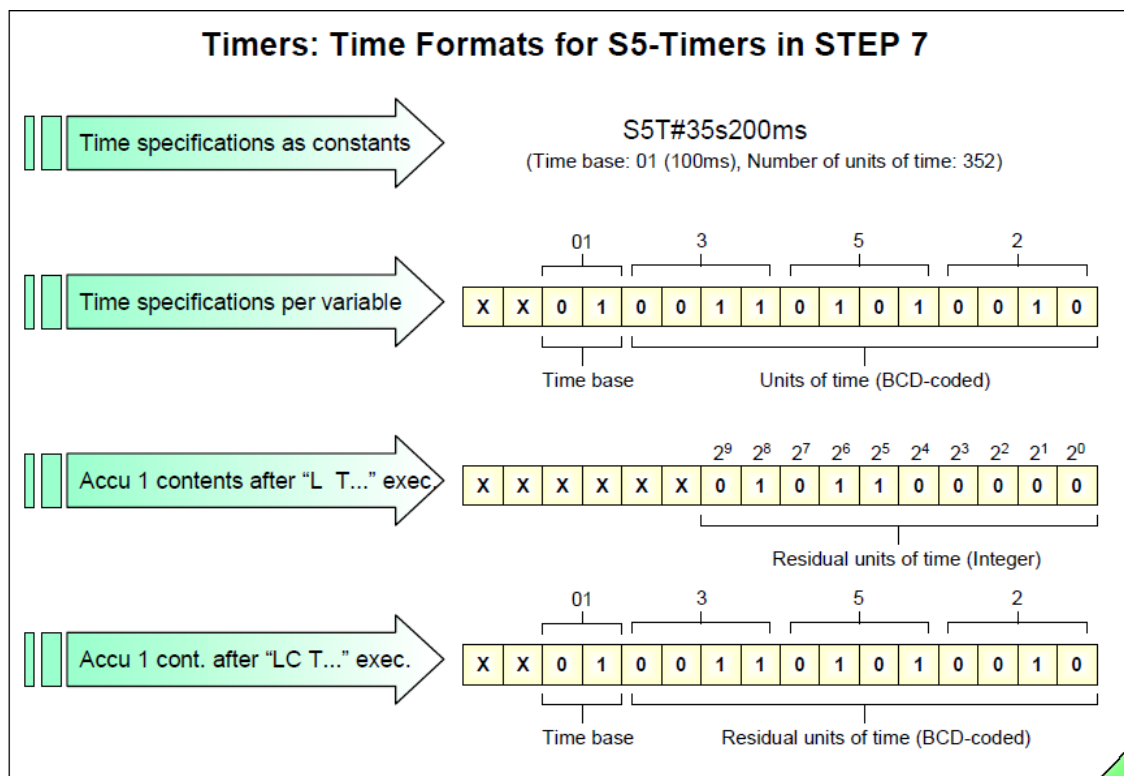
نحوه قرارگیری دیتا در آکومولاتور



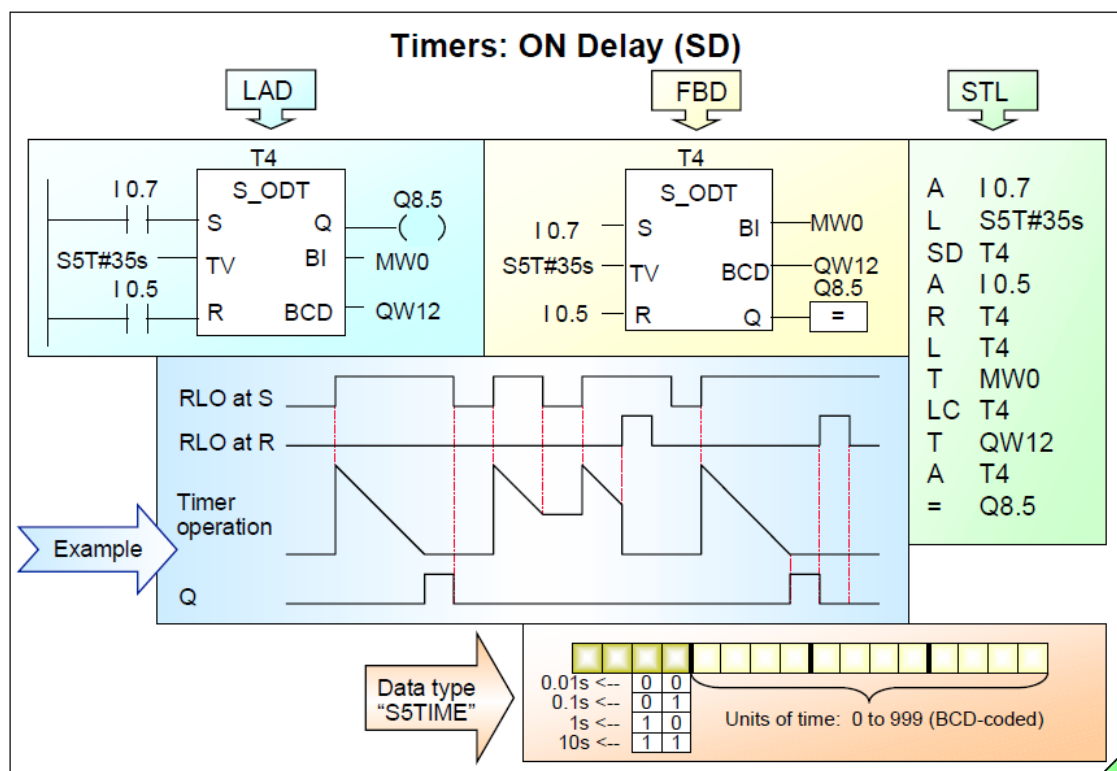
دستورات شمارنده



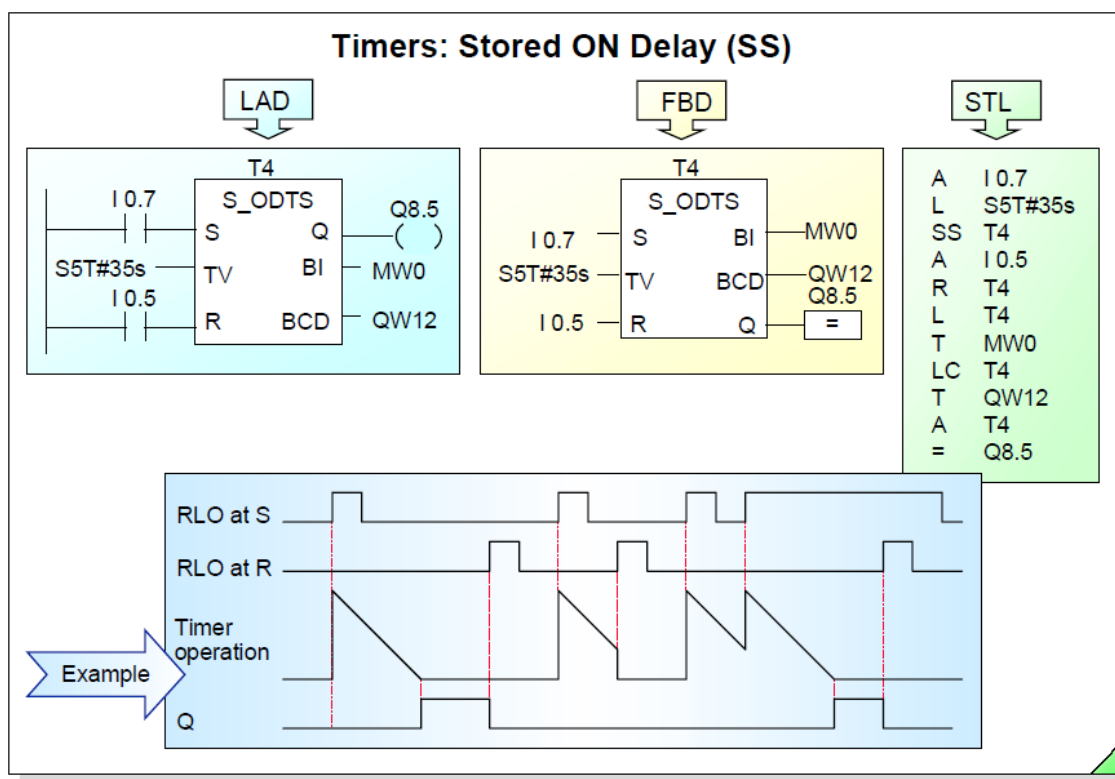
فرمت زمان در S7



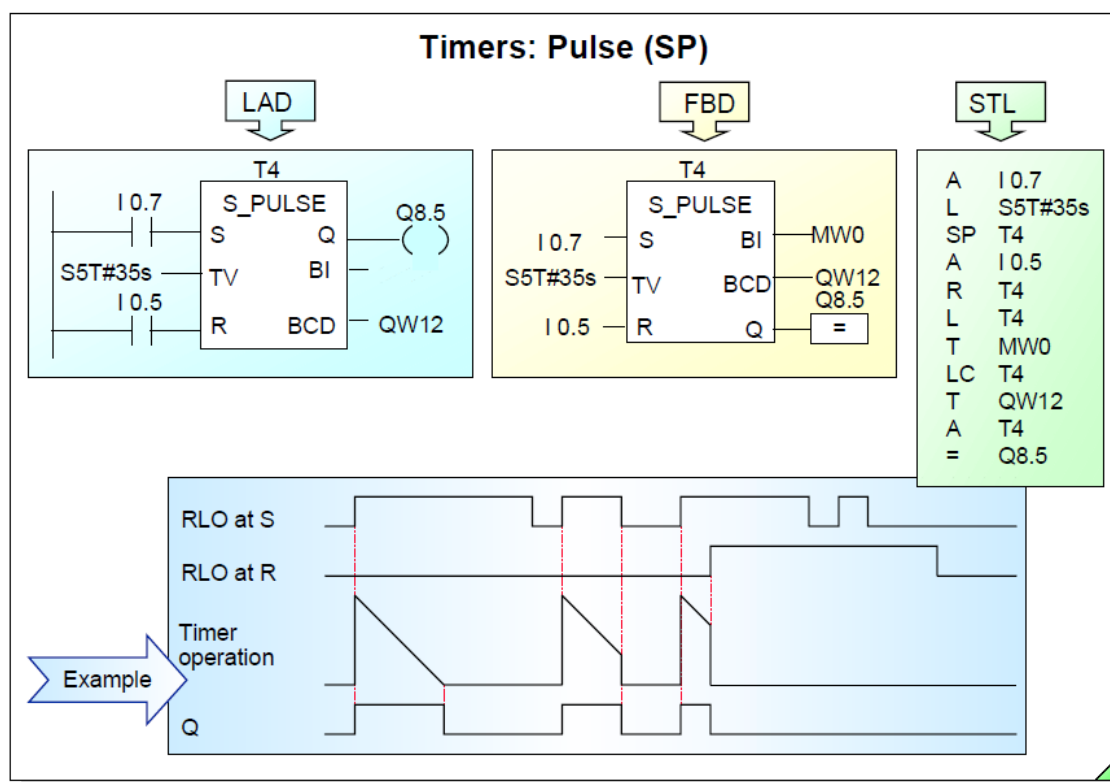
تایمر S-ODT



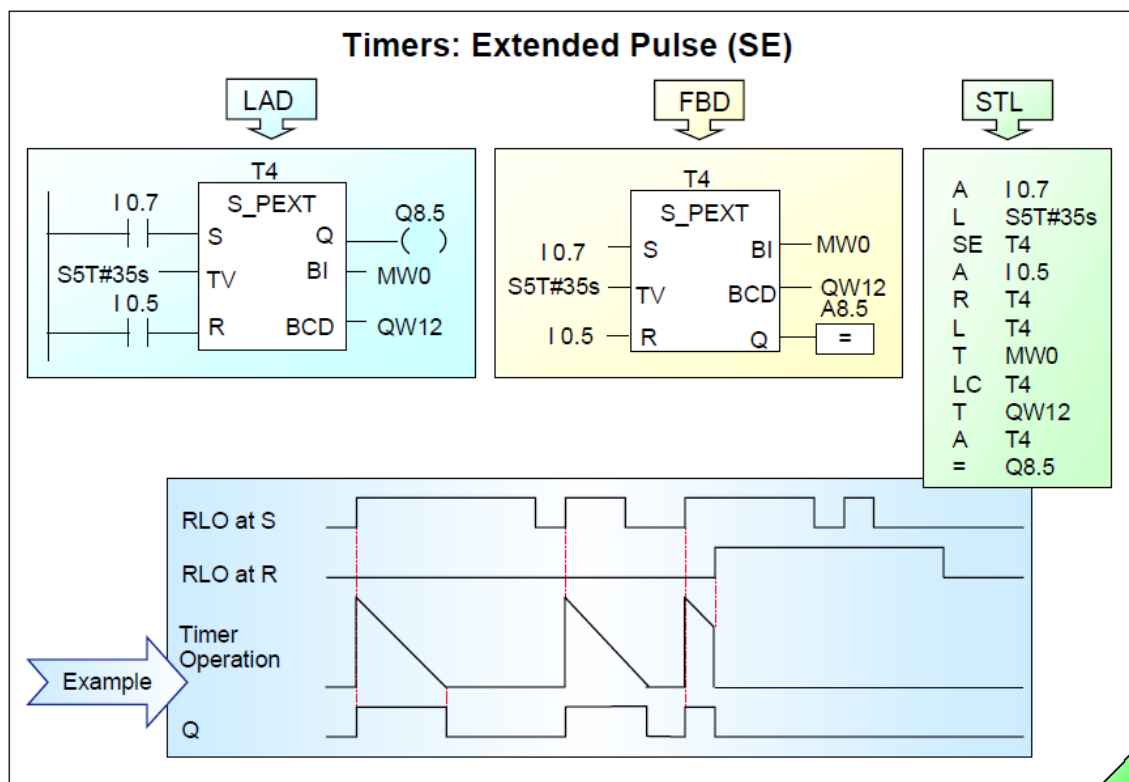
تایمر S-ODTS



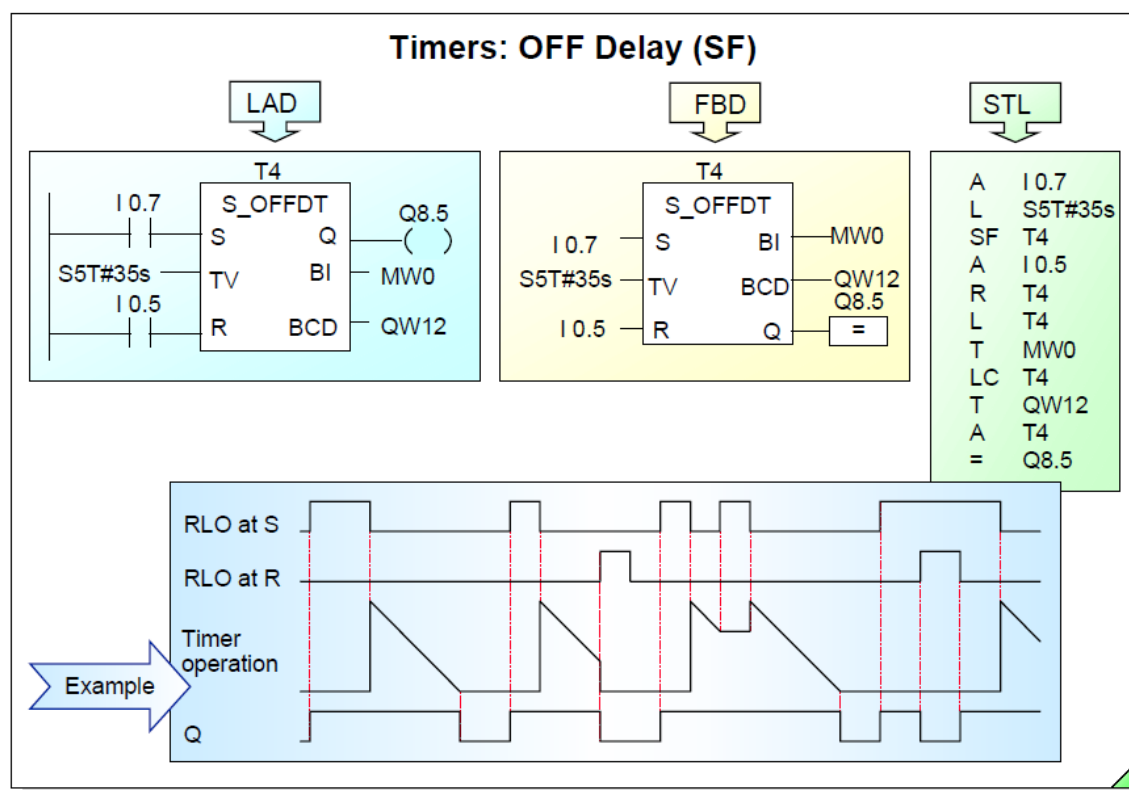
تایمر S-PULSE



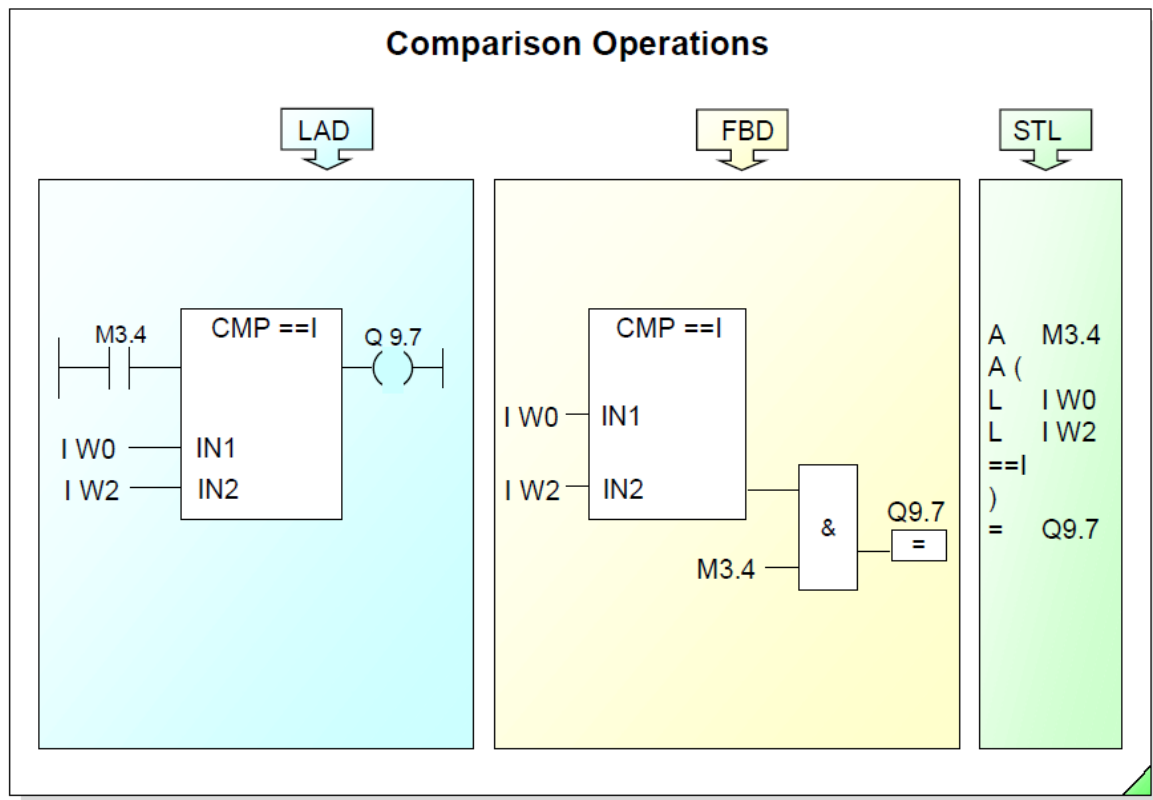
تایمر S-PEXT



تایمر S-OFFDT



دستورات مقایسه کننده



CMP

You can use comparison instructions to compare the following pairs of numerical values:

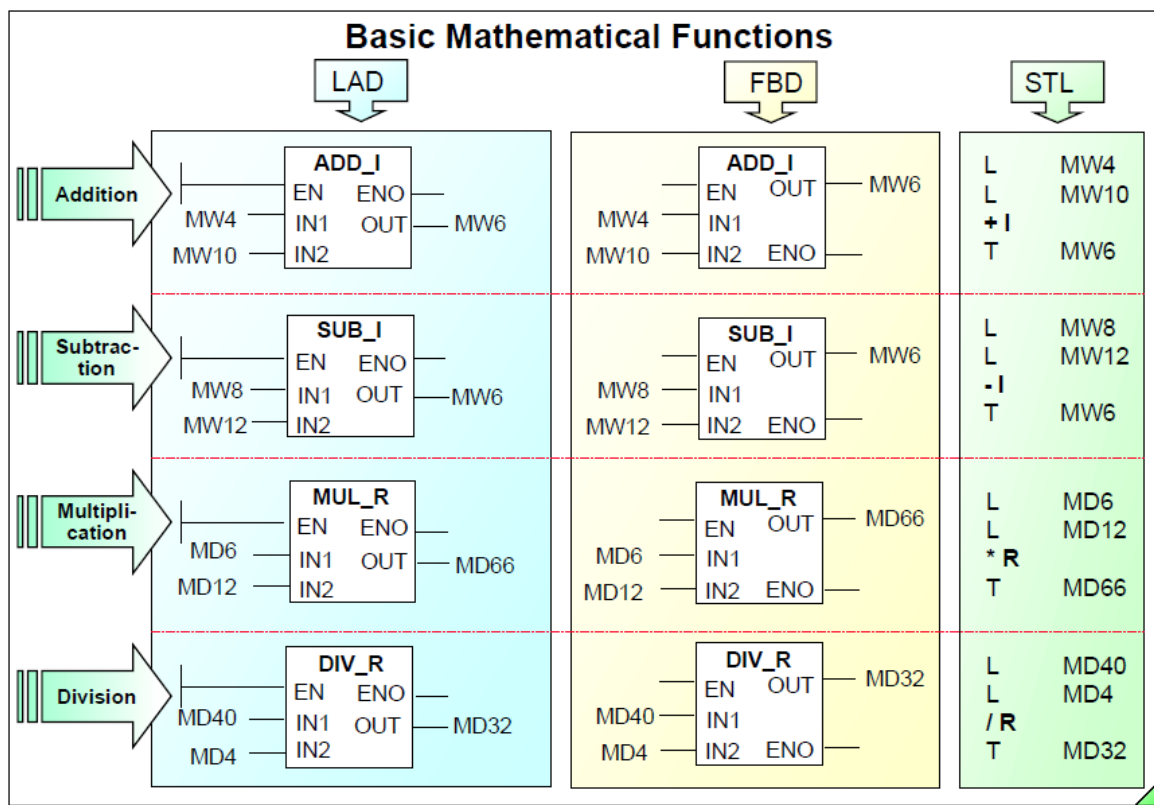
- I** Compare integers (on the basis of 16 bit fixed-point number)
- D** Compare integers (on the basis of 32 bit fixed-point number)
- R** Compare floating-point numbers (on 32 bit real number basis = IEEE floating-point numbers).

If the result of the comparison is "true", then the RLO of the operation is "1", otherwise it is "0".

The values at inputs IN1 and IN2 are compared for conformity with the specified condition:

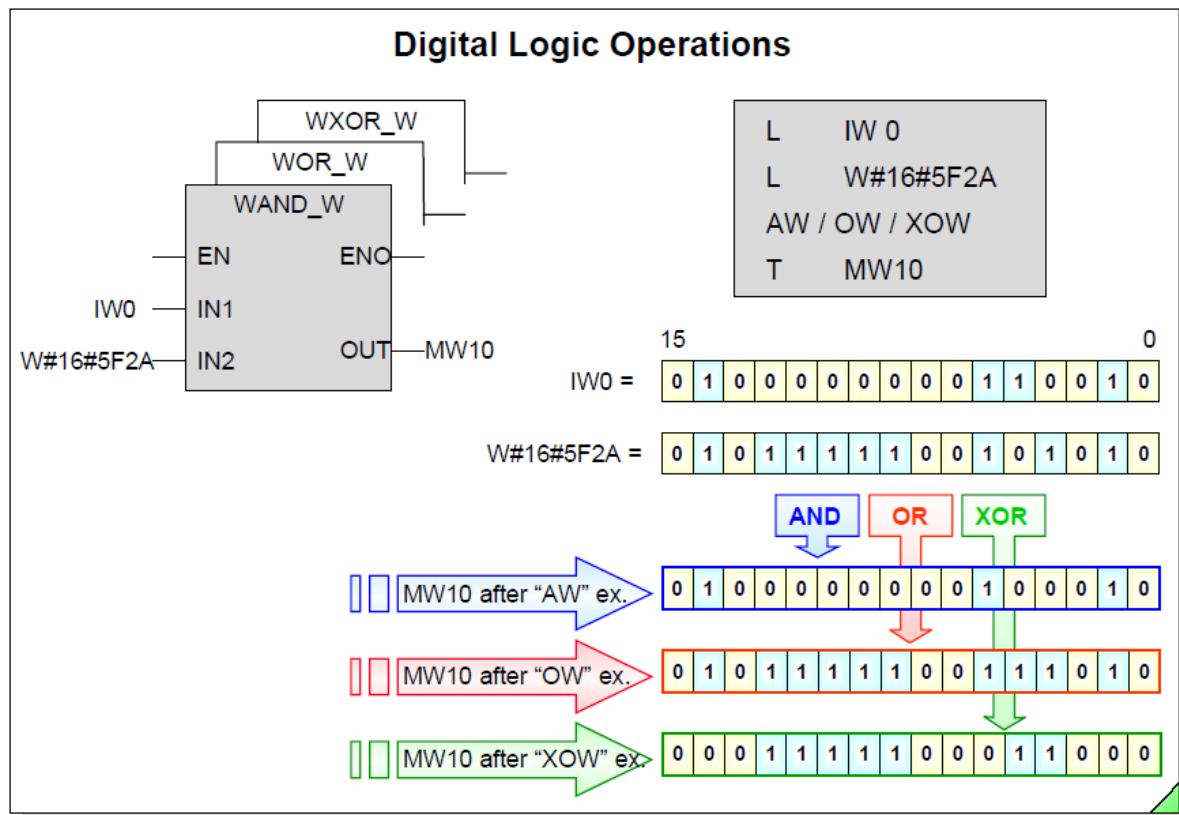
- ==** IN1 is equal to IN2
- <>** IN1 is not equal to IN2
- >** IN1 is greater than IN2
- <** IN1 is less than IN2
- >=** IN1 is greater than or equal to IN2
- <=** IN1 is less than or equal to IN2.

دستورات ریاضی



Addition:	ADD_I	Add integer
	ADD_DI	Add double integer
	ADD_R	Add real number
Subtraction:	SUB_I	Subtract integer
	SUB_DI	Subtract double integer
	SUB_R	Subtract real number
Multiplication:	MUL_I	Multiply integer
	MUL_DI	Multiply double integer
	MUL_R	Multiply real number
Division:	DIV_I	Divide integer
	DIV_DI	Divide double integer
	DIV_R	Divide real number

دستورات Logical Operations

**WAND_W**

The "AND Word" operation gates the two digital values at inputs IN1 and IN2 bit by bit in accordance with the AND truth table. The result of the AND operation is stored at the address at output OUT.

The instruction is executed when EN = 1.

Example: Masking out the 4th decade of the thumbwheel buttons :

```

IW4=      =0100 0100 1100 0100
W#16#0FFF =0000 1111 1111 1111
MW30      =0000 0100 1100 0100

```

WOR_W

The "OR Word" operation gates the two digital values at inputs IN1 and IN2 bit by bit in accordance with the OR truth table. The result of the OR operation is stored at the address at output OUT.

The instruction is executed when EN = 1.

Example: Setting bit 0 in MW32 :

```

MW32      =0100 0010 0110 1010
W#16#0001 =0000 0000 0000 0001
MW32      =0100 0010 0110 1011

```

WXOR_W

The "Exclusive OR Word" operation gates the two digital values at inputs IN1 and IN2 bit by bit in accordance with the XOR truth table. The result of the OR operation is stored at the address at output OUT. The result of the XOR operation is stored at the address at output OUT.

The instruction is executed when EN=1.

Example: detecting signal changes in IW0 :

```

IW0      =0100 0100 1100 1010
MW28     =0110 0010 1011 1001
MW24     =0010 0110 0111 0011

```

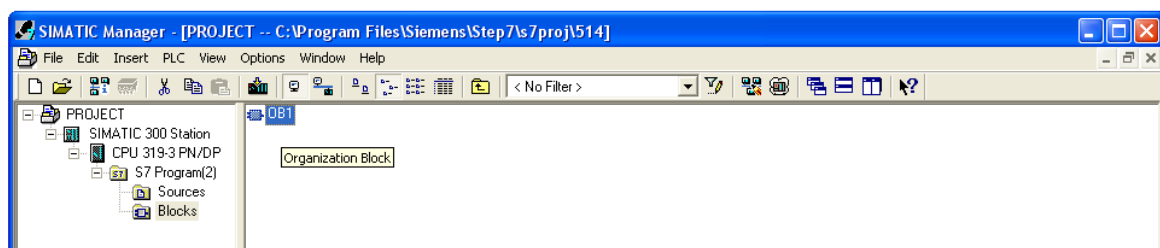
انواع داده در S7

Elementary Data Types in STEP 7

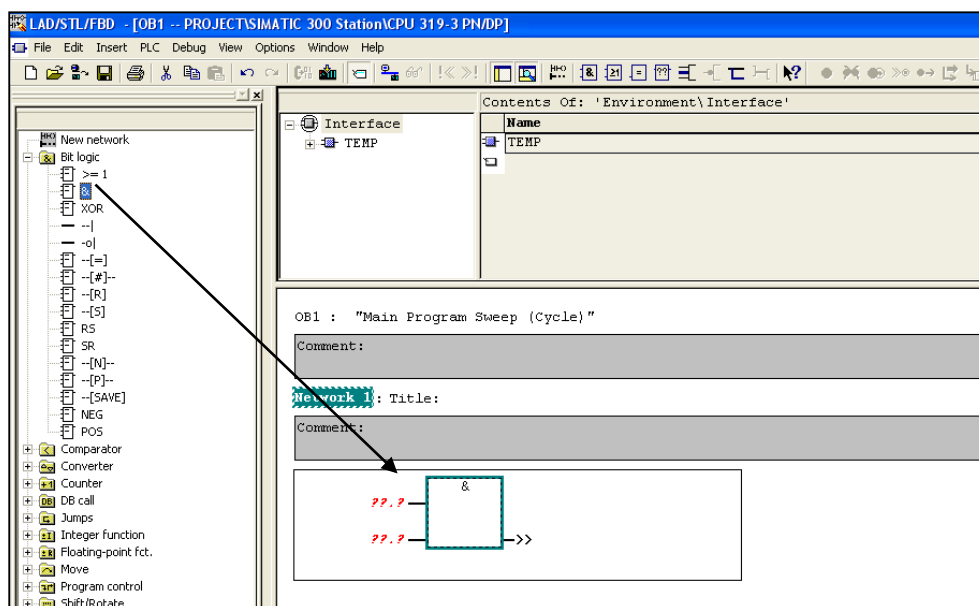
Keyword	Length (in bits)	Example of a constant of this type
BOOL	1	1 or 0
BYTE	8	B#16#A9
WORD	16	W#16#12AF
DWORD	32	DW#16#ADAC1EF5
CHAR	8	'w'
S5TIME	16	S5T#5s_200ms
INT	16	123
DINT	32	65539
REAL	32	1.2 or 34.5E-12
TIME	32	T#2D_1H_3M_45S_12MS
DATE	16	D#1993-01-20
TIME_OF_DAY	32	TOD#12:23:45.12

برنامه نویسی در محیط OB1

همانطور که می دانید بلوک اصلی برنامه کاربر OB1 می باشد. در این قسمت نیز جهت آشنایی هر چه بیشتر با دستورات برنامه نویسی تنها از بلوک OB1 به عنوان بلوک برنامه استفاده می کنیم. برای این منظور لازم است در نرم افزار پس از ساختن یک پروژه بر روی گزینه OB1 دابل کلیک تا وارد محیط برنامه نویسی این بلوک شوید.



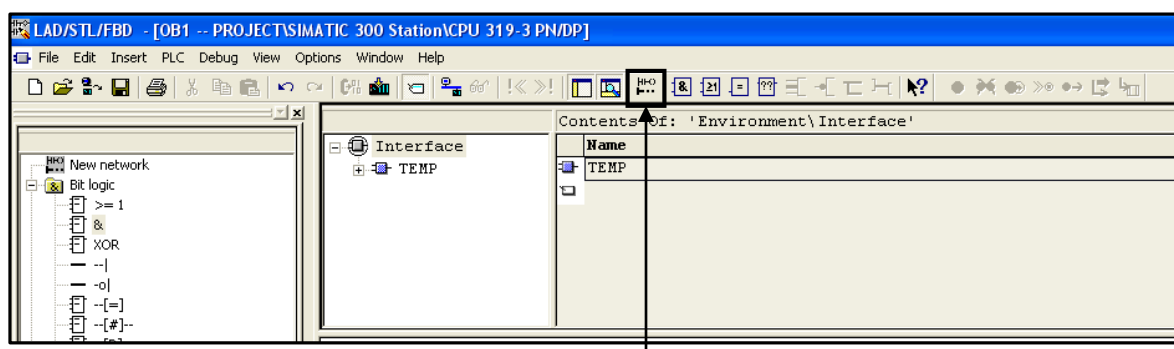
در بلوک OB1 گزینه ای تحت عنوان Bit Logic در پنجره دستورات وجود دارد که در زیر مجموعه این گزینه دستورات بیتی همانند تیغه ها، گیت های منطقی، فلیپ فلاپ ها و دستورات تشخیص لبه قرار گرفته شده اند.



البته لازم به ذکر است که پنجره دستورات در روش برنامه نویسی STL فاقد گزینه Bit Logic و سایر گزینه های مربوط به دستورات می باشد.

نحوه ساختن Network در برنامه

در بلوک های برنامه نویسی برنامه کنترلی می بایست در Network های مختلف وارد شود. در ادامه، برنامه های بررسی شده دارای چندین دستور فلیپ فلاپ می باشند که هر فلیپ فلاپ می بایست در یک Network وارد شود. این موضوع برای سایر دستورات نیز صادق می باشد. ساده ترین راه جهت ساخت Network در بلوک های برنامه استفاده از آیکون مشخص شده در شکل زیر می باشد.



New Network

با هر بار فعال کردن آیکون شکل فوق، یک Network جدید ساخته می شود.

OB1 : "Main Program Sweep (Cycle)"

Comment:

Network 1: Title:

Comment:

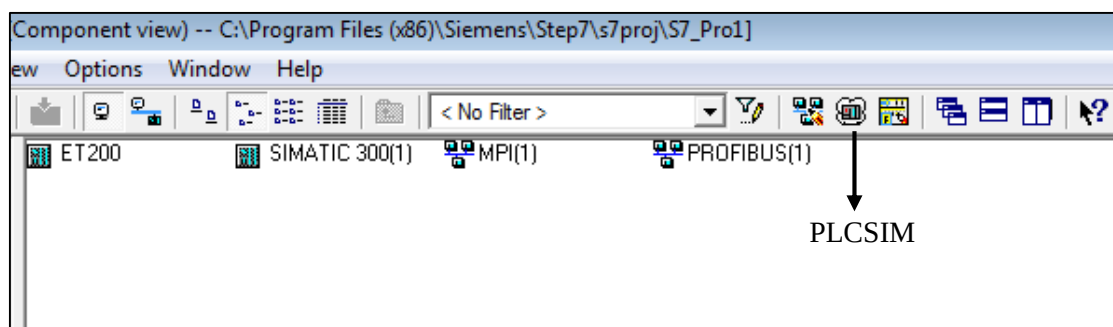
Network 2: Title:

Comment:

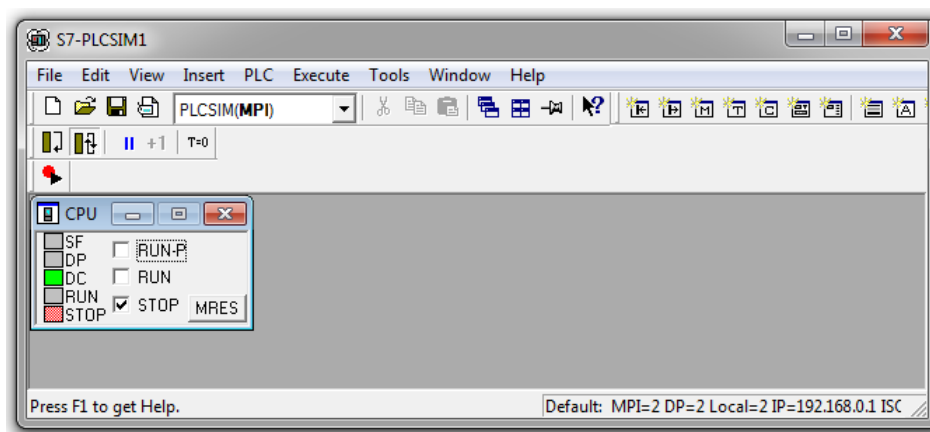
همانطور که در شکل فوق مشاهده می کنید، بلوک OB1 دارای دو Network می باشد که هر Network محل نوشتن دستورات مربوط به یک برنامه می باشد.

تست برنامه

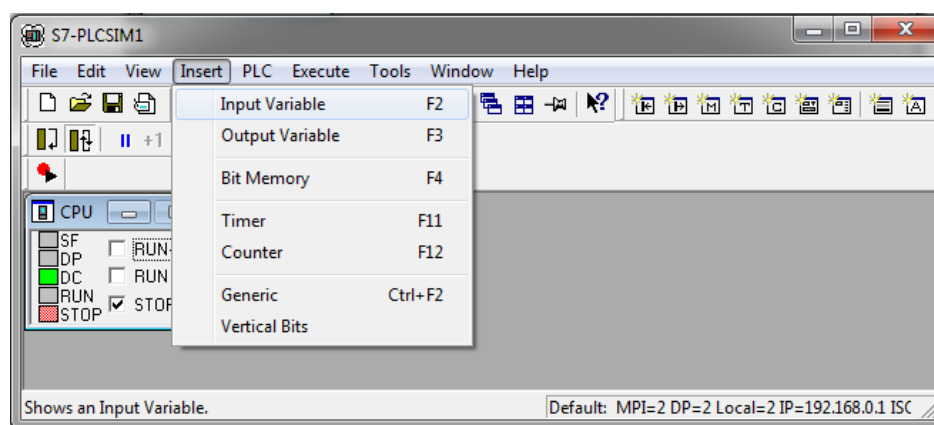
برای تست برنامه در محیط نرم افزار می بایست از نرم افزار PLCSIM استفاده کرد. برای باز کردن نرم افزار PLCSIM بر روی آیکون مشخص شده در شکل زیر کلیک می کنیم.



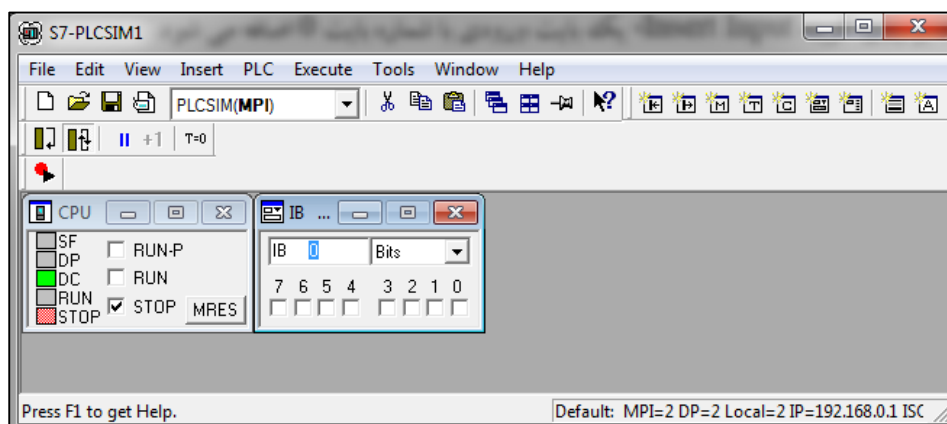
با کلیک بر روی آیکون شکل فوق، محیط شبیه ساز باز می شود.



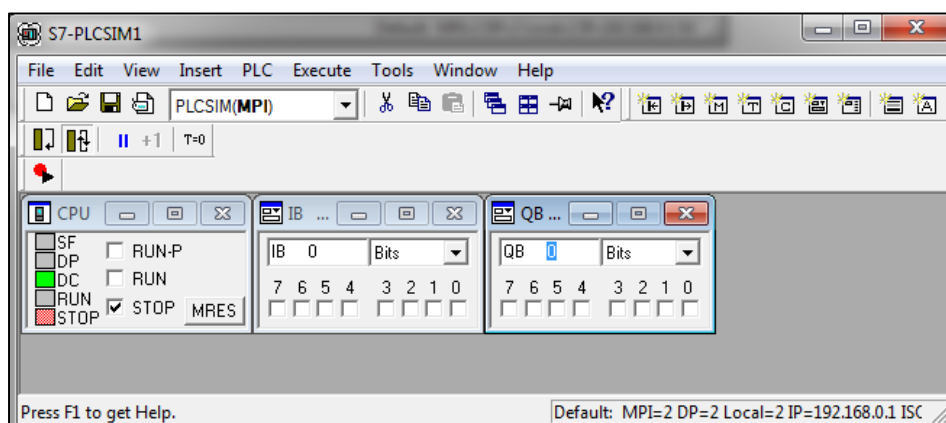
برای وارد کردن بایت های مربوط به ورودی و خروجی در محیط PLCSIM، به مسیر مشخص شده در شکل زیر مراجعه کنید.



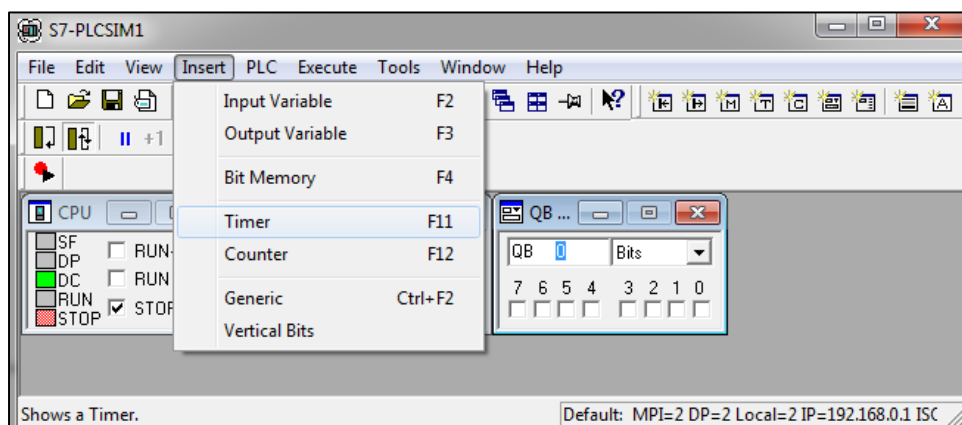
با کلیک بر روی گزینه Insert Input، یک بایت ورودی با شماره بایت 0 اضافه می شود.



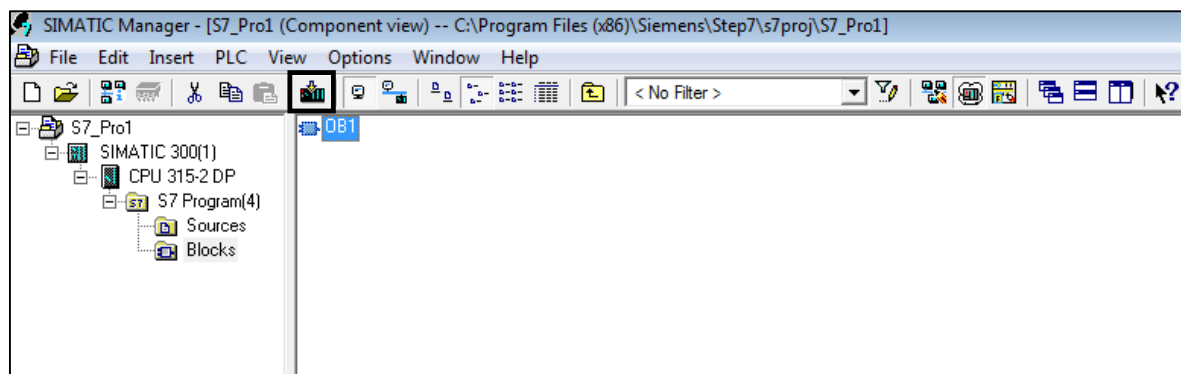
با کلیک بر روی گزینه Insert Output نیز یک بایت خروجی با شماره بایت 0 اضافه می شود.



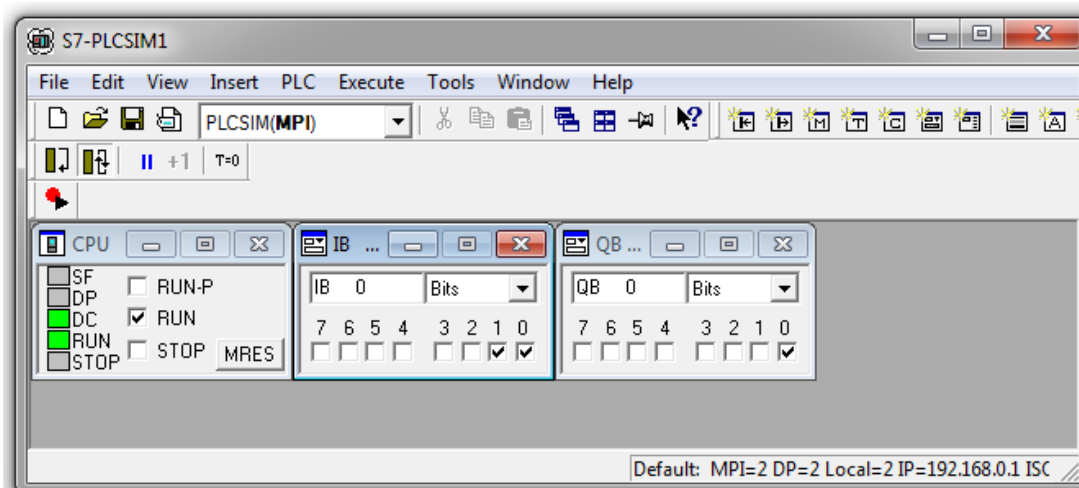
برای دیدن مقادیر تایمرها و کانترها نیز می توان از گزینه های مشخص شده در شکل استفاده نمود.



برای دانلود بلوک برنامه، زمانی که شبیه ساز در وضعیت STOP می باشد، بر روی گزینه دانلود کلیک کنید.

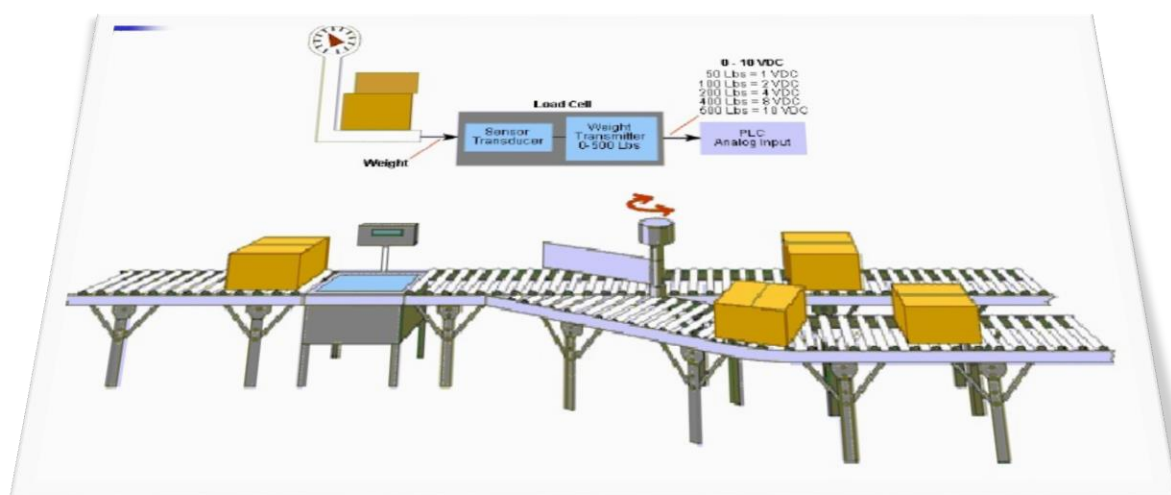


با دانلود بلوک برنامه، شبیه ساز را جهت تست در وضعیت RUN قرار می دهیم. در ادامه با فعال کردن بیت های ورودی، مطابق لاجیک نوشته شده برای برنامه، خروجی ها فعال میشوند.



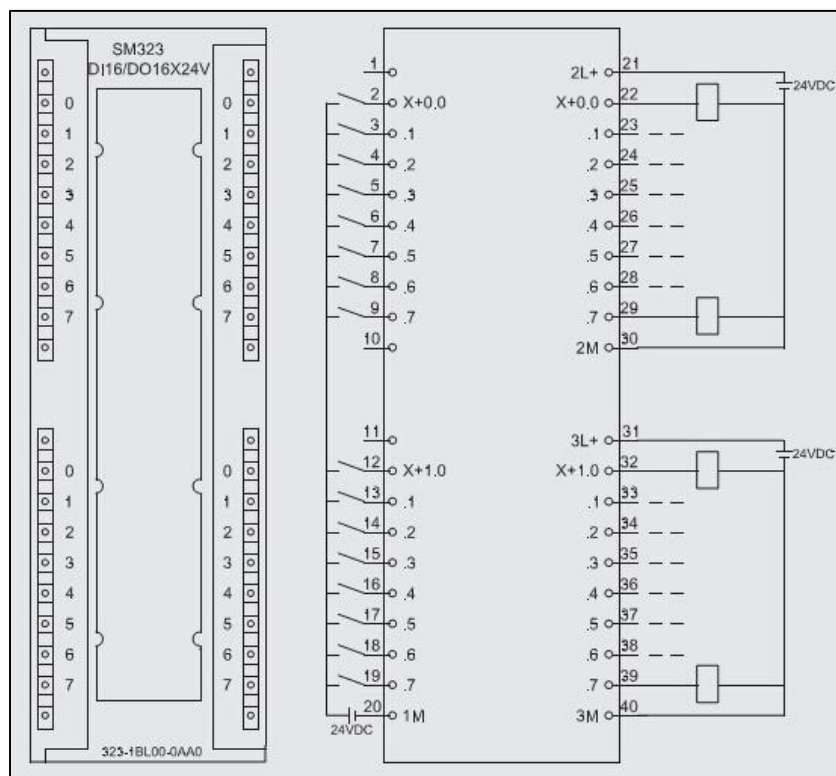
مجموعه مثال های برنامه نویسی

SIMATIC MANAGER



سخت افزار

در تمامی مثال ها فرض بر این است که ورودی های متصل شده به کارت DI تماما شاستی استارت، و خروجی های DO نیز به رله ها متصل شده اند.



روش های برنامه نویسی

STL: این زبان برنامه نویسی یک روش نوشتاری است که مجموعه دستورات به صورت کدهایی از پیش تعریف شده نوشته می شود. برخی از دستورات برنامه نویسی تنها در روش STL معتبر می باشند و معادل آنها در روش های گرافیکی تعریف نشده است. اکثر برنامه هایی که به این زبان نوشته می شوند قابل تبدیل به روش های گرافیکی نمی باشند.

LADDER و FBD زبان های برنامه نویسی گرافیکی هستند که لیست دستورات به صورت سمبل ها و یا بلوک هایی در دسترس می باشند. برنامه ای که به یکی از این روش ها نوشته می شود، قابل تبدیل شدن به زبان

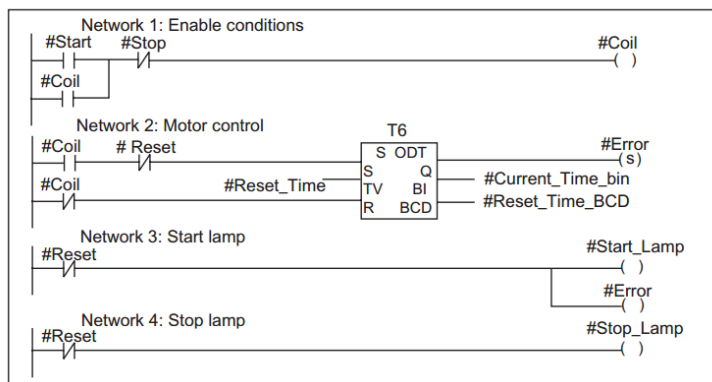
نوشتاری می باشد.

Example of Networks in Statement List

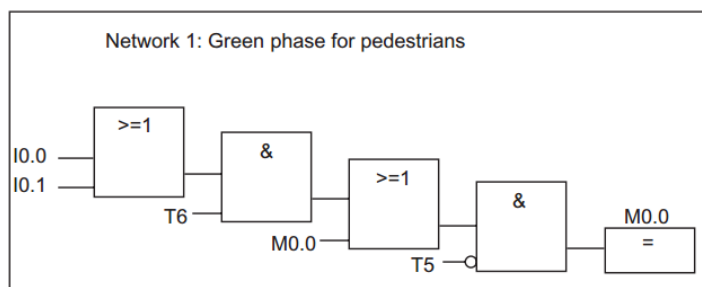
```

Network 1: Control drain valve
A(
O
O #Coil
)
AN #Close
= #Coil
Network 2: Display "Valve open"
A #Coil
= #Disp_open
Network 3: Display "Valve closed"
AN #Coil
= #Disp_closed
    
```

Example of Networks in LAD



Example of a Network in FBD



مثال 1

در این مثال با فعال شدن ورودی I0.0، خروجی Q0.0 به صورت لحظه ای فعال می گردد.

روش LAD

Network 1 : Title:

Comment:



روش FBD

Network 1 : Title:

Comment:



روش STL

Network 1 : Title:

Comment:

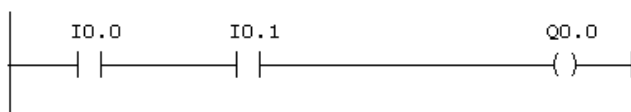
```
A      I      O.0
=      Q      O.0
```

مثال 2

در این مثال زمانی که هر دو ورودی IO.0 و IO.1 با هم فعال شوند، خروجی Q0.0 به صورت لحظه ای فعال می شود. (دستگاه کاتر)

Network 1 : Title:

Comment:



Network 1 : Title:

Comment:



Network 1 : Title:

Comment:

A	I	O.0
A	I	O.1
=	Q	O.0

مثال 3

در این مثال زمانی که هر کدام از ورودی های I0.0 یا I0.1 فعال شوند، خروجی Q0.0 به صورت لحظه ای فعال می گردد.

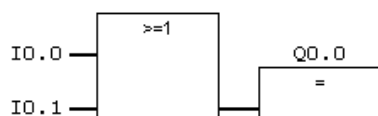
Network 1 : Title:

Comment:



Network 1 : Title:

Comment:



Network 1 : Title:

Comment:

O	I	0.0
O	I	0.1
=	Q	0.0

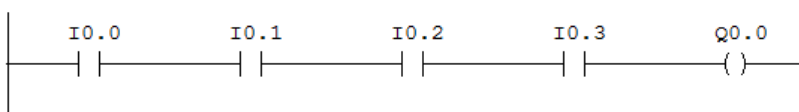
همانطور که می دانید منطق عملکرد AND معادل ضرب و منطق عملکرد OR معادل جمع در مدارات منطقی می باشد.

مثال 4

پیاده سازی رابطه $(A.B.C.D)=F$

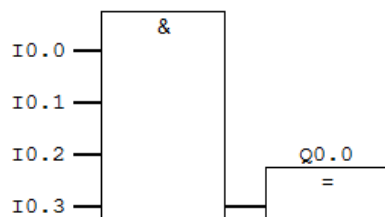
Network 1: Title:

Comment:



Network 1: Title:

Comment:



Network 1: Title:

Comment:

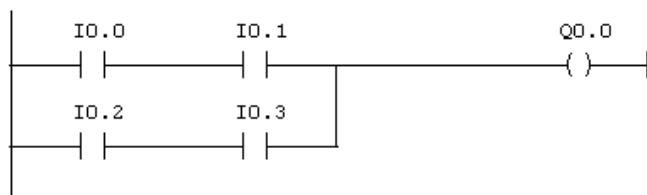
A	I	0.0
A	I	0.1
A	I	0.2
A	I	0.3
=	Q	0.0

مثال 5

پیاده سازی رابطه $(A.B)+(C.D)=F$

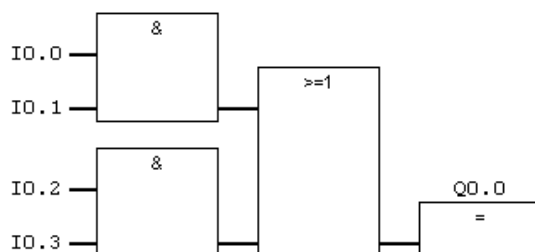
Network 1 : Title:

Comment:



Network 1 : Title:

Comment:



Network 1 : Title:

Comment:

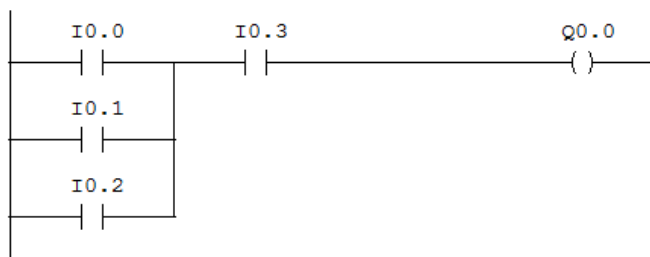
A	I	0.0
A	I	0.1
O		
A	I	0.2
A	I	0.3
=	Q	0.0

مثال 6

پیاده سازی رابطه $(A+B+C).D=F$

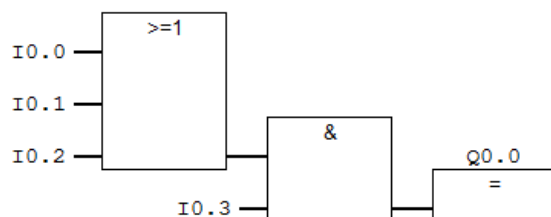
Network 1: Title:

Comment:



Network 1: Title:

Comment:



Network 1: Title:

Comment:

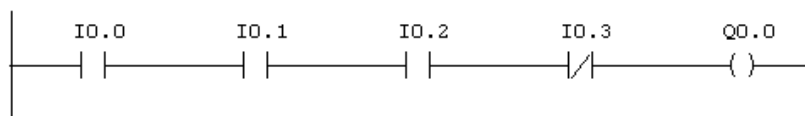
```
A(
O   I      0.0
O   I      0.1
O   I      0.2
)
A   I      0.3
=   Q      0.0
```

مثال 7

در این مثال با فرض اینکه تمامی ورودی های متصل شده به PLC شاستی استارت می باشند، زمانی که 3 ورودی اول فعال شوند، خروجی Q0.0 فعال می گردد. در این حالت زمانی که ورودی چهارم فعال شود، خروجی قطع می شود. در واقع در برنامه ورودی I0.3 به صورت NOT در نظر گرفته شده است.

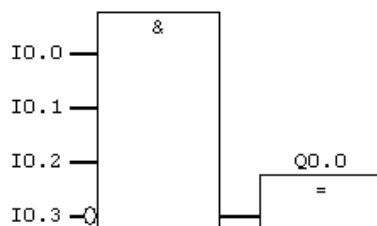
Network 1 : Title:

Comment:



Network 1 : Title:

Comment:



Network 1 : Title:

Comment:

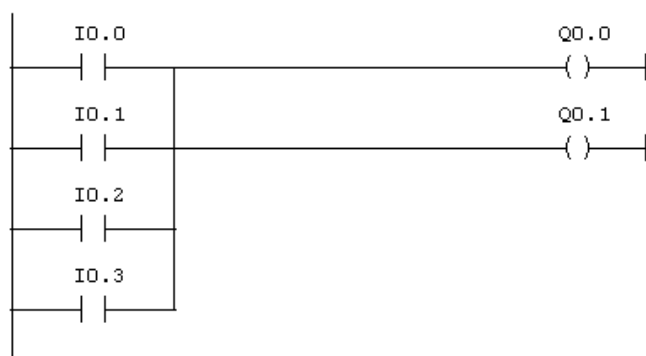
A	I	O.0
A	I	O.1
A	I	O.2
AN	I	O.3
=	Q	O.0

مثال 8

در این مثال زمانی که هر یک از چهار شاستی متصل شده به ورودی فعال شوند، خروجی های Q0.0 و Q0.1 به صورت لحظه ای فعال می شوند.

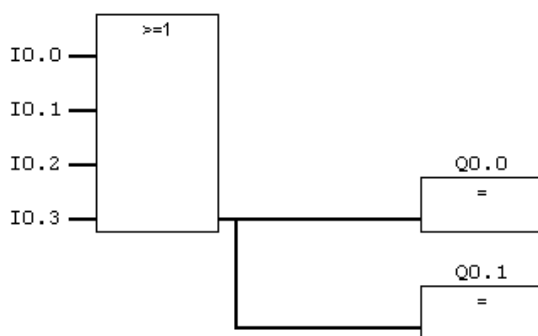
Network 1: Title:

Comment:



Network 1: Title:

Comment:



Network 1: Title:

Comment:

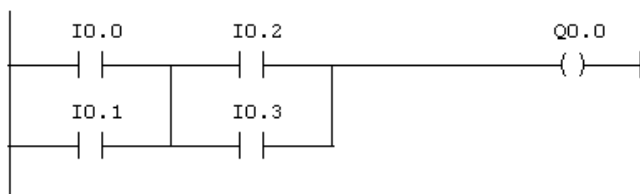
O	I	0.0
O	I	0.1
O	I	0.2
O	I	0.3
=	Q	0.0
=	Q	0.1

مثال 9

پیاده سازی رابطه $(A+B).(C+D)=F$

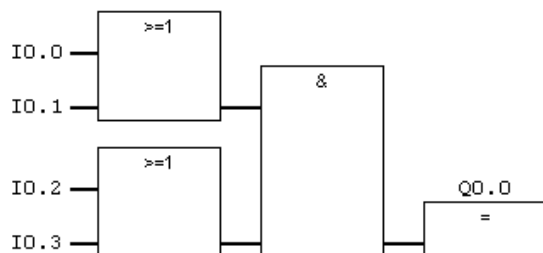
Network 1 : Title:

Comment:



Network 1 : Title:

Comment:



Network 1 : Title:

Comment:

```

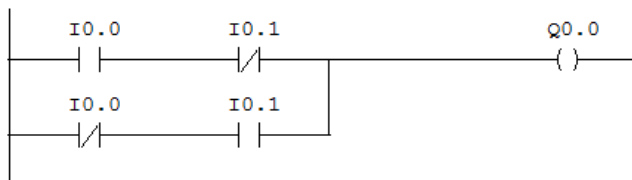
A(
O      I      0.0
O      I      0.1
)
A(
O      I      0.2
O      I      0.3
)
=      Q      0.0
    
```

مثال 10**پیاده سازی رابطه $XOR : (A'.B)+(A.B')=F$**

کاربرد این منطق مثلا برای مشخص کردن وضعیت کنترل فاز است. تحت شرایطی که هر سه فاز ورودی وصل و یا قطع باشند، بیزر خاموش است. وقتی برنامه کنترلی، دو فاز شدن و یا تک فاز شدن را تشخیص دهد، خروجی PLC (آلارم) فعال می شود.

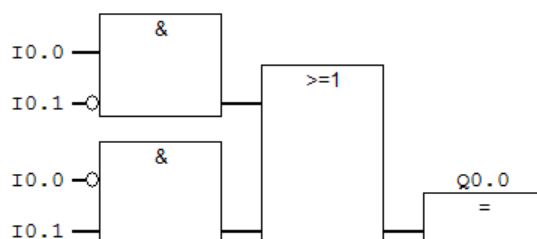
Network 1: Title:

Comment:



Network 1: Title:

Comment:



Network 1: Title:

Comment:

A	I	0.0
AN	I	0.1
O		
AN	I	0.0
A	I	0.1
=	Q	0.0

در مدارات فرمان، برای اجرای لاجیک فوق، می بایست از شاستی استپ استارت دوبل استفاده کرد.

مثال 11

در تمامی مثال بررسی شده، برنامه STL نشان داده شده، نتیجه تبدیل برنامه از روش های LAD یا FBD به روش STL بود. به عبارت دیگر این برنامه STL پیش فرض نرم افزار می باشد. در این مثال می خواهیم برنامه مثال 5 را به روش STL از ابتدا طراحی کنیم. نکته قابل توجه این است که این برنامه از لحاظ عملکرد کاملاً شبیه برنامه مثال 5 می باشد، ولی به روش های LAD یا FBD تبدیل نمی شود.

Network 1 : Title:

Comment:

A	I	0.0
A	I	0.1
=	M	0.0
A	I	0.2
A	I	0.3
=	M	0.1
O	M	0.0
O	M	0.1
=	Q	0.0

مثال 12

بازنویسی مثال 9 به روش STL

Network 1 : Title:

Comment:

O	I	0.0
O	I	0.1
=	M	0.0
O	I	0.2
O	I	0.3
=	M	0.1
A	M	0.0
A	M	0.1
=	Q	0.0

مثال 13

برنامه ای به زبان های گرافیکی بنویسید که منطق آن مشابه منطق STL شکل زیر باشد.

Network 1 : Title:

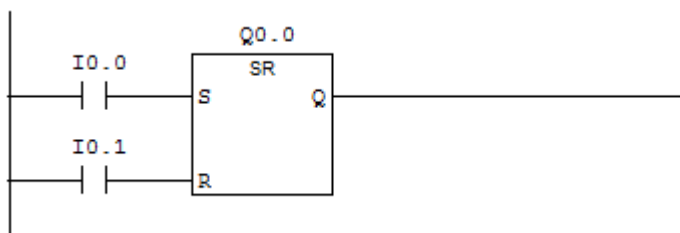
Comment:

O	I	0.0
O	I	0.1
=	M	0.0
&	M	0.0
&	I	0.2
=	M	0.1
O	M	0.1
O	I	0.3
=	Q	0.0

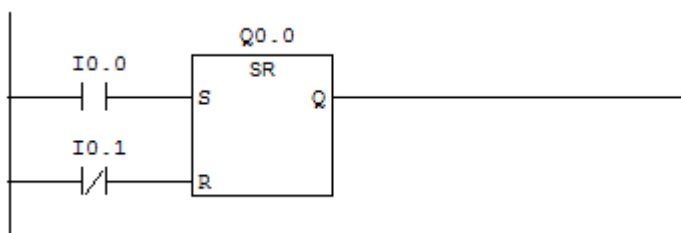
❖ تا اینجا در تمام مثال ها از خروجی لحظه ای استفاده کردیم. در این حالت با برداشتن دست از روی شاستی خروجی قطع میشود و در حالت شبیه سازی وضعیت ورودی ها را به صورت کلید نگه میداشتیم تا بتوانیم نتایج خروجی را ببینیم. اما با توجه به اینکه در عمل، اکثر مدارهای کنترلی به صورت دائم کار می باشند و از آنجایی که تمامی باتن ها به صورت شاستی در نظر گرفته می شوند، می بایست خروجی ها به صورت دائم کنترل شوند.

مثال 14

کنترل یک موتور از یک نقطه به صورت دائم



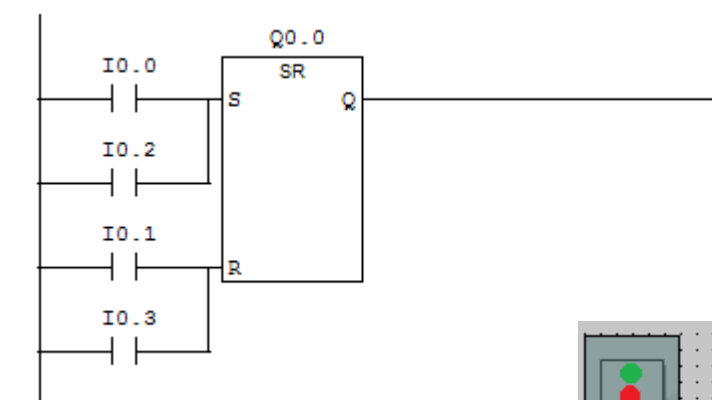
نکته ای که در استفاده از دستور فلیپ فلاپ وجود دارد این است که پایه های فلیپ فلاپ حساس به لبه بالارونده ی ورودی هستند و ما در این مثال نیز از شاستی استارت برای ریست استفاده کردیم. اگر معادل مدار فرمان از شاستی استپ استفاده کنیم که دارای تیغه ی N.C هست، برای اینکه اثر یک بودن را بر روی پایه ی ریست برداریم باید حالت تیغه بسته ی شاستی استپ را روی پایه ی ریست قرار دهیم.



در ادامه بحث فرض بر این است که کلیه ی کلید ها و شاستی ها از جنس استارت هستند.

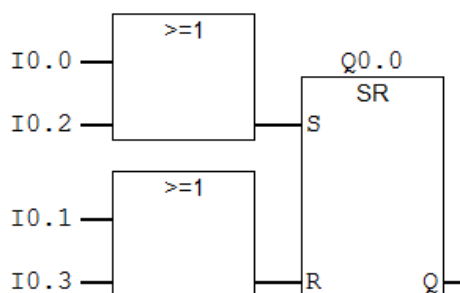
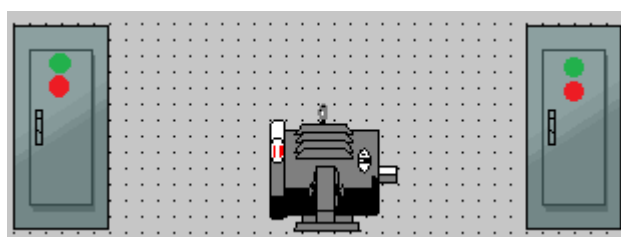
مثال 15

کنترل یک موتور از دو نقطه به صورت دائم



تابلو اول : استارت I0.0 و استپ I0.1

تابلو دوم : استارت I0.2 و استپ I0.3

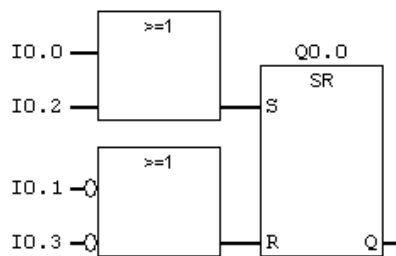


آموزش روش STL در قالب مثال های جداگانه آموزش داده می شود.

فرض کنید در مثال قبل ورودی هایی که به عنوان ریست در نظر گرفته شده اند، از جنس استپ باشند. برنامه به صورت زیر تغییر می کند.

Network 1 : Title:

Comment:

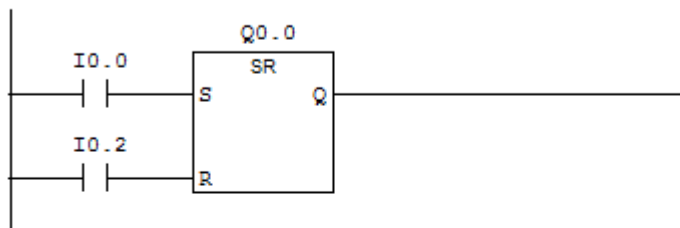


مثال 16

کنترل دو موتور از دو نقطه به صورت مجزا

Network 1: Title:

Comment:



استارت موتور اول I0.0

استارت موتور دوم I0.1

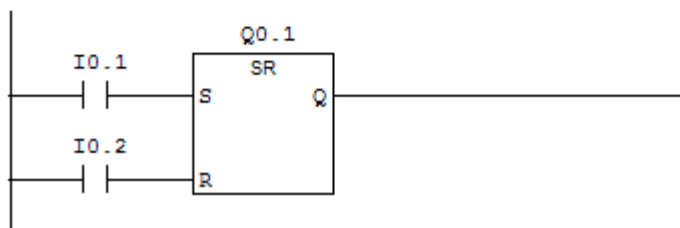
استپ کل I0.2

موتور اول Q0.0

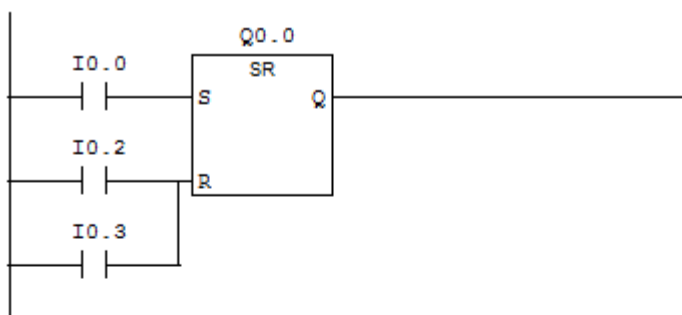
موتور دوم Q0.1

Network 2: Title:

Comment:

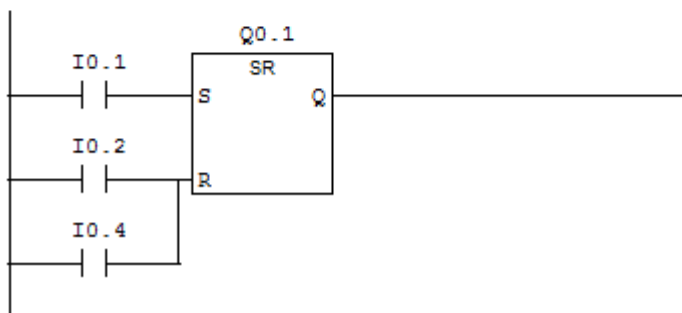


در مثال قبل با یک استپ کل هر دو موتور متوقف می شوند. در صورتی که هر یک از موتورها استپ جداگانه داشته باشند، برنامه به صورت زیر خواهد بود.



Network 2 : Title:

Comment:

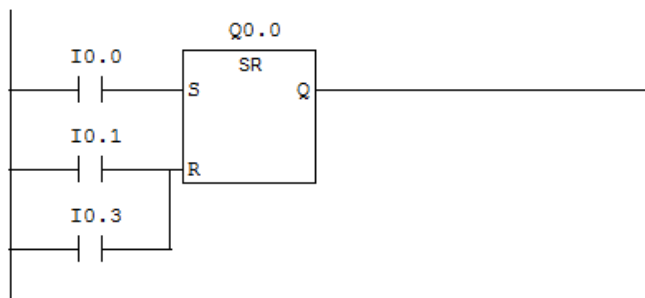


مثال 17

مدار یکی پس از دیگری

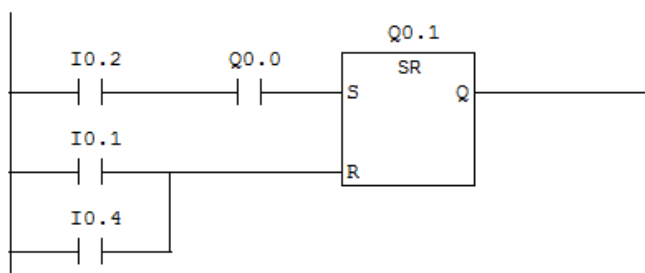
Network 1 : Title:

Comment:



Network 2 : Title:

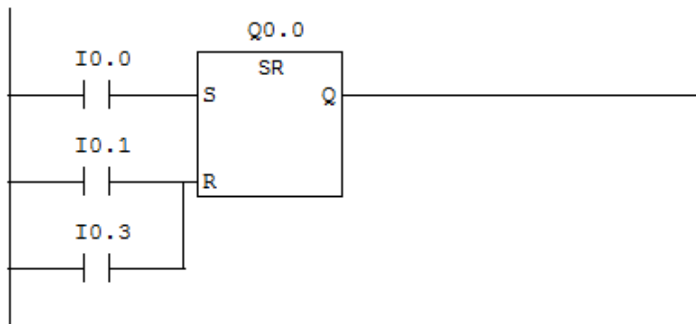
Comment:



در برنامه مثال قبل، با فشردن شاستی I0.3، موتور اول خاموش می شود، ولی موتور دوم به کار خود ادامه می دهد و هیچ گونه وابستگی بعد از روشن شدن به خروجی Q0.0 ندارد.

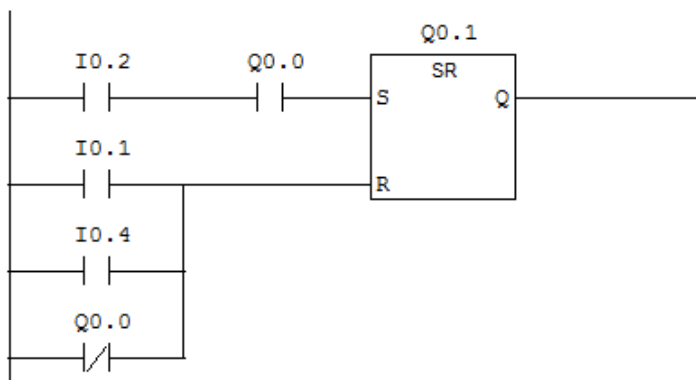
Network 1 : Title:

Comment:



Network 2 : Title:

Comment:



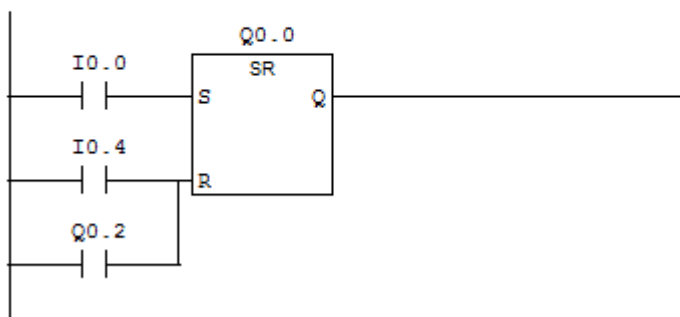
در این مثال اگر پمپ روغن به هر دلیلی خاموش شود، موتور مته نیز خاموش می شود. با زدن استپ I0.3 که مربوط به پمپ روغن می باشد، می توان عملکرد این برنامه را تست نمود.

مثال 18

در یک سیستم صنعتی سه عدد شیر وجود دارد. هریک از شیرها با شاستی استارت خود باز می شوند مشروط به اینکه شیر دوم حتما بعد شیر اول و شیر سوم بعد شیر دوم باز شود. باز شدن شیر سوم باعث بسته شدن دو شیر قبلی می شود.

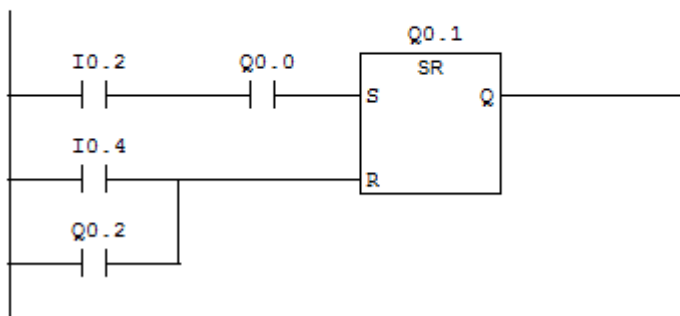
Network 1 : Title:

Comment:



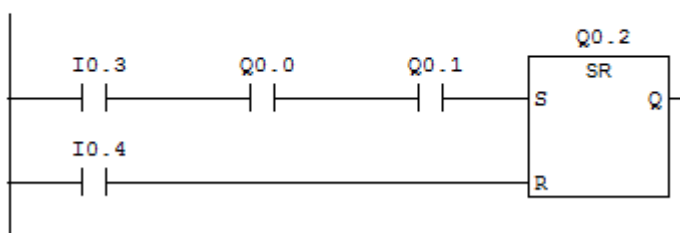
Network 2 : Title:

Comment:



Network 3 : Title:

Comment:

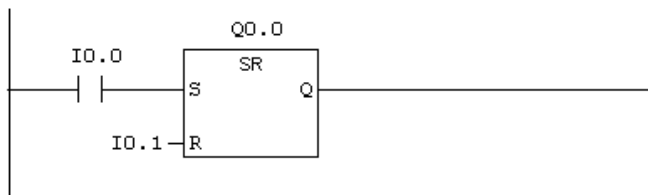


مثال 19

دو موتور داریم که با زدن شاستی اول موتور اول روشن می شود. تا زمانیکه دست روی شاستی هست، موتور اول تنها کار میکند. با برداشتن دست از روی شاستی موتور دوم نیز روشن می شود.

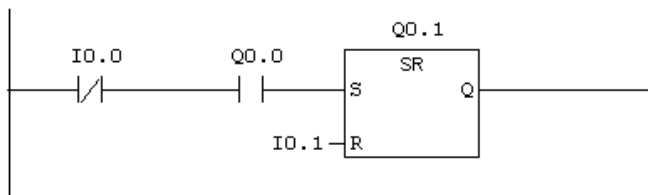
Network 1 : Title:

Comment:



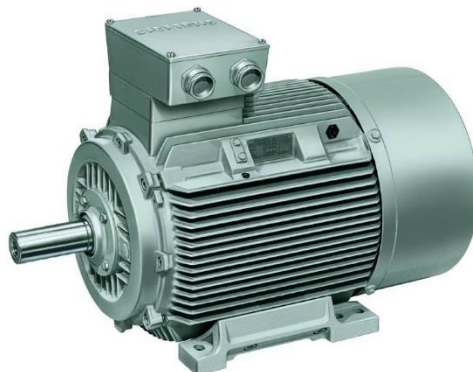
Network 2: Title:

Comment:



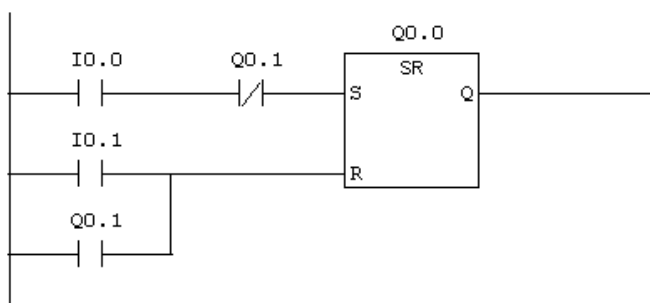
مثال 20

چپگرد راستگرد کند با حفاظت کامل



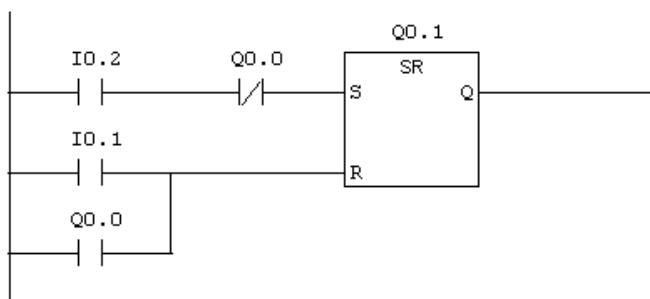
Network 1 : Title:

Comment:

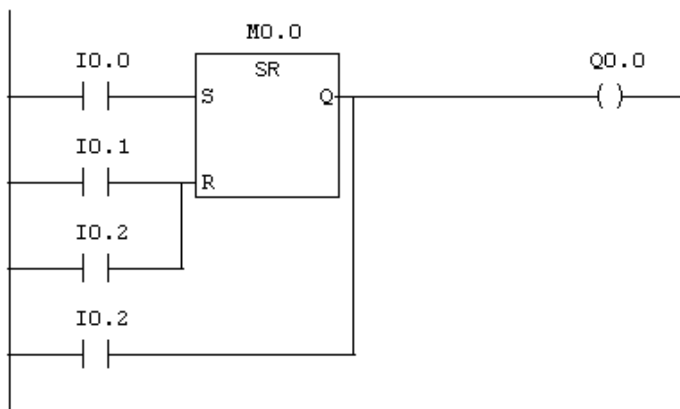


Network 2 : Title:

Comment:



مثال 21



کنترل یک موتور به صورت لحظه ای و دائم

IO.0: شاستی استارت حالت دائم

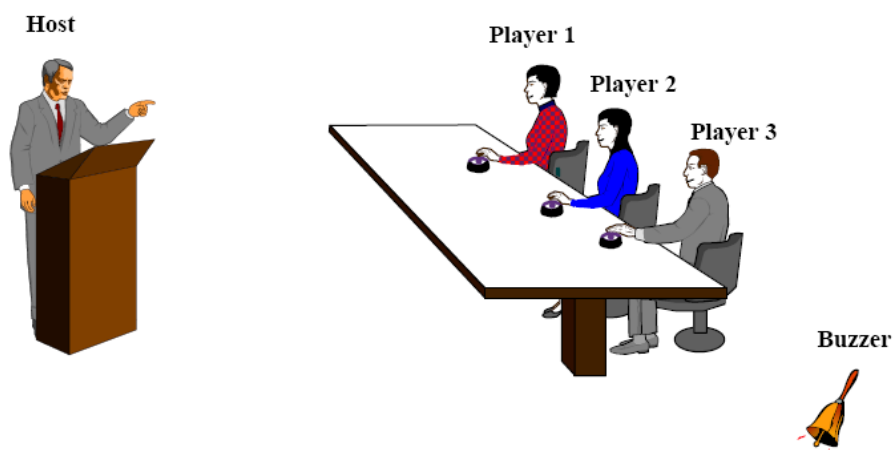
IO.1: شاستی استاپ حالت دائم

IO.2: پدال حالت لحظه ای

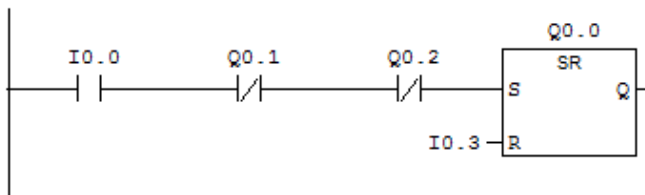
تمرین 22

میز مسابقه برای سه شرکت کننده

می خواهیم برنامه مربوط به سه میز مسابقه را طراحی نماییم. بطوری که با فشردن شاستی میز اول، لامپ آن میز روشن شود و در این لحظه اگر شاستی میز دوم و سوم فشرده شد، در مدار هیچ تغییری حاصل نگردد. این مطلب برای سایر شرکت کننده ها نیز صادق است.

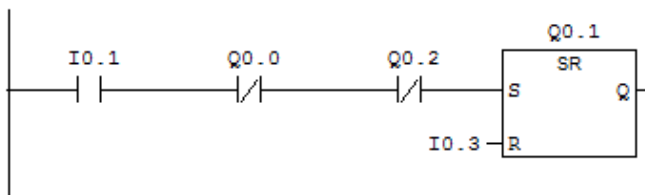


برنامه



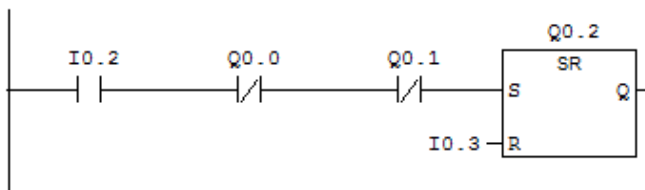
Network 2 : Title:

Comment:



Network 3 : Title:

Comment:

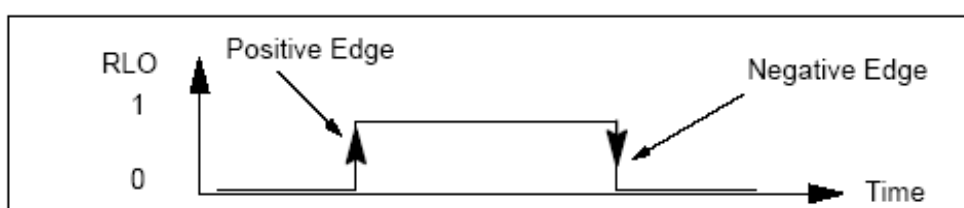


تمرین

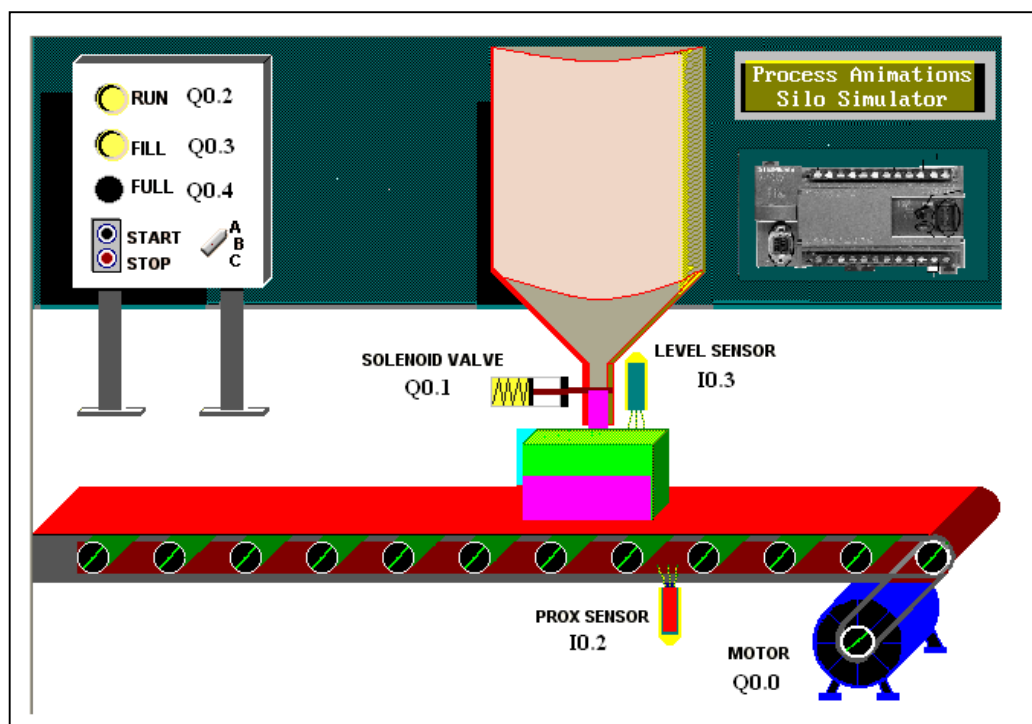
در یک سیستم صنعتی پنج مصرف کننده وجود دارد که از یک منبع، تغذیه می شوند. توان مصرفی هریک تجهیزات به شرح زیر است : P1:5 kw , P2:2 kw , P3:1 kw , P4:7 kw , P5:9 kw

در این پروژه تنها تعدادی از مصرف کننده ها می توانند همزمان روشن شوند که مجموع بار مصرفی آنها کمتر یا برابر 10 kw باشد. هر مصرف کننده دارای یک استارت و استپ مجزا می باشند.

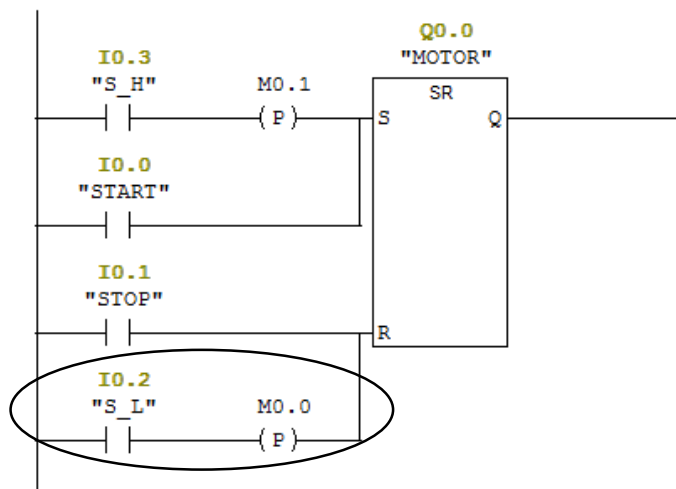
این دستورات یکی دیگر از دستورات پر کاربرد در برنامه نویسی می باشند. توسط این دستورات می توان لبه های ایجاد شده (بالا یا پایین) در یک پالس را آشکار نمود. با استفاده از این دستورات تغییر ورودی از قطع به وصل (خاموش به روشن) یا از وصل به قطع (روشن به خاموش) آشکار می شود. مثلاً هر گاه آشکارسازی قطع به وصل (لبه بالا رونده) یک حسگر مورد نظر باشد، PLC به ازای هر قطع به وصل در یک مرور اجازه عبور برنامه از طریق اتصال مربوطه را می دهد. دستور تغییر از وصل به قطع (روشن به خاموش) نیز به همین ترتیب و در جهت عکس دستور قبل عمل می کند. مثلاً یک کلید را فرض کنیم. زمانی که کلید را فشار می دهیم، اگر از دستور تغییر مثبت استفاده کرده باشیم خروجی فعال می شود. ولی اگر از دستور تغییر منفی استفاده کرده باشیم، زمانی که کلید را رها کنیم خروجی فعال می شود.



این دستورات دارای کاربردهای فراوانی در برنامه نویسی می باشند و در بسیاری از برنامه ها کار برنامه نویسی را برای کاربران ساده و سریع می کنند. در ادامه به چند کاربرد این دستورات با بررسی مثال های مختلف توجه فرمایید.

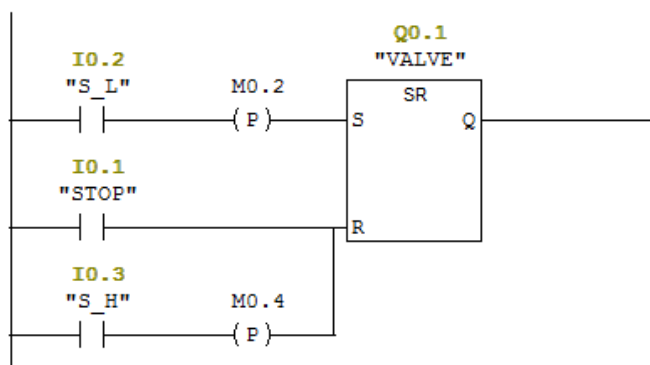


برنامه مثال 23



Network 2: Title:

Comment:



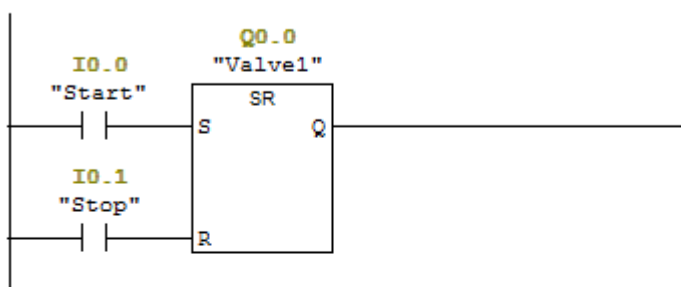
در این مثال قرار دادن دستور P بر روی ورودی I0.2 در NET1 ضروری می باشد. ولی قرار دادن دستور P برای هر دو سنسور در هر دو NET مشکلی ایجاد نمی کند.

آدرس حافظه ای که بالای دستور حساس به لبه ی بالا اختصاص می یابد، جهت انتقال نتیجه به محل دیگر کاربرد دارد.

مثال 24 (کاربرد دستور حساس به لبه پایین رونده): در یک سیستم صنعتی دو شیر وجود دارد که بسته شدن یکی منجر به باز شدن دیگری می شود. در این حالت داریم:

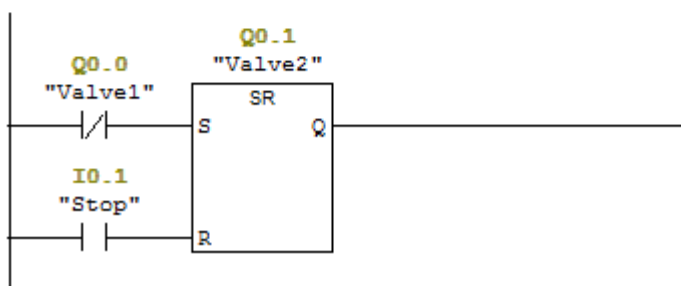
Network 1: Title:

Comment:



Network 2: Title:

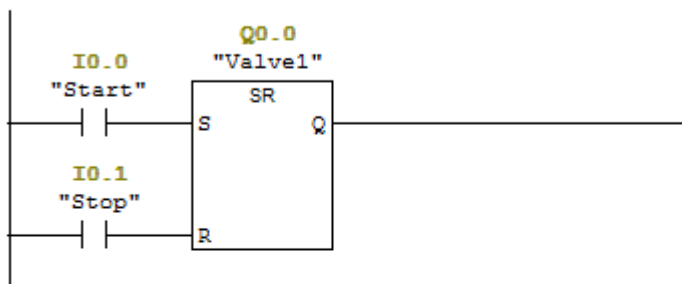
Comment:



در این روش برنامه نویسی از لحظه ی اول Valve2 باز می شود. برای اینکه حتما تغییر حالت بسته شدن شیر 1 را شرط باز شدن شیر 2 قرار دهیم، از دستور حساس به لبه ی پایین استفاده میکنیم.

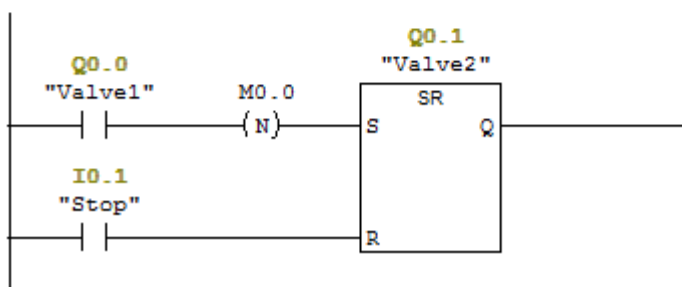
Network 1: Title:

Comment:



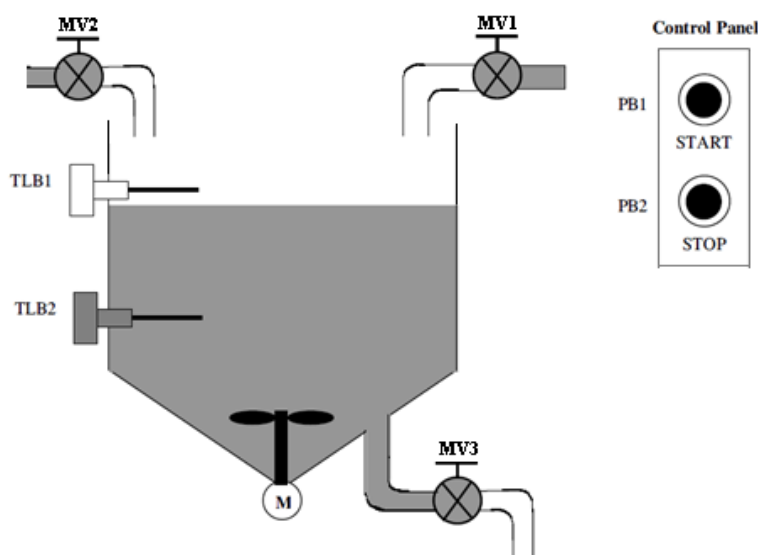
Network 2: Title:

Comment:



مثال 25- کنترل شارژ و تخلیه مخزن

در یک فرآیند صنعتی دو مایع می بایست در یک مخزن با یکدیگر میکس شوند. نحوه کار بدین صورت می باشد که با فشردن شستی استارت PB1 تعبیه شده بر روی پانل، شیرهای ورودی MV1 و MV2 باز می شوند. در همین زمان موتور مربوط به میکسر نیز روشن می گردد. وقتی سطح مایع از سنسور TLB2 عبور و به سنسور TLB1 رسید، شیرهای ورودی بسته و موتور نیز استپ می شود. در مرحله بعد شیر MV3 باز شده و عمل تخلیه صورت می گیرد. وقتی سطح مایع به زیر سنسور TBL2 رسید، شیر تخلیه MV3 و مجددا شیرهای ورودی برای شارژ جدید باز شوند. فشردن شستی استپ نیز در هر مرحله کل سیستم را متوقف می کند.



لیست I/O :

استارت PB1 : I0.0

استپ PB2 : I0.1

سنسور حد بالای مخزن TBL1 : I0.2

سنسور حد پایین مخزن TBL2 : I0.3

شیر ورودی MV1 : Q0.0

شیر ورودی MV2 : Q0.1

موتور میکسر : Q0.2

شیر تخلیه MV3 : Q0.3

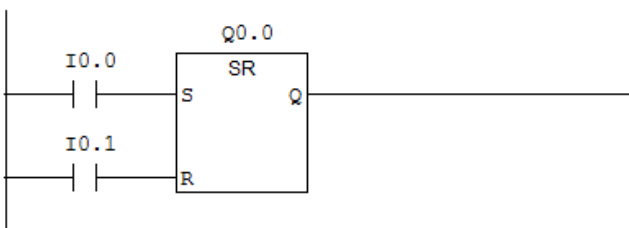
مثال 26

با فشردن شاستی استارت I0.0 پمپ 1 روشن و پس از سپری شدن 10 ثانیه ، پمپ 2 نیز روشن شود. با فشردن شاستی استپ I0.1 هر دو پمپ خاموش شوند.



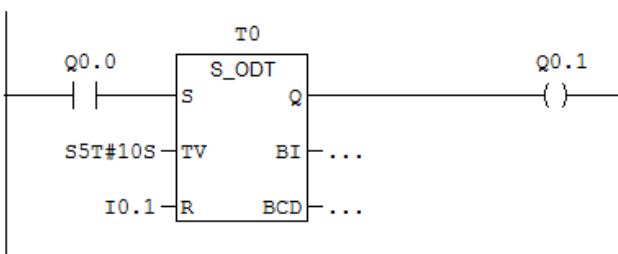
Network 1 : Title:

Comment:



Network 2 : Title:

Comment:

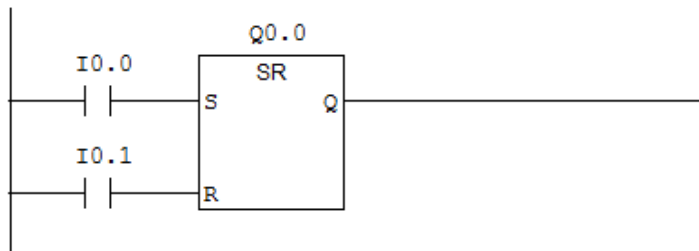


در این مثال با خاموش شدن پمپ اول، پمپ دوم نیز خاموش میشود. چون تایمر به لبه پایین رونده ورودی حساس می باشد.

در این مثال (مثال 27) می خواهیم وابستگی پمپ دوم را به پمپ اول حذف کنیم. در ضمن هدف طراحی مثال ها با تایمر S-ODT می باشد.

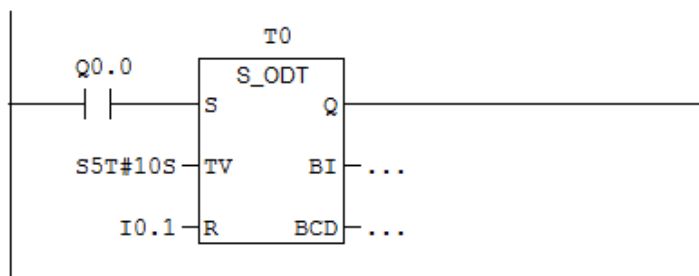
Network 1 : Title:

Comment:



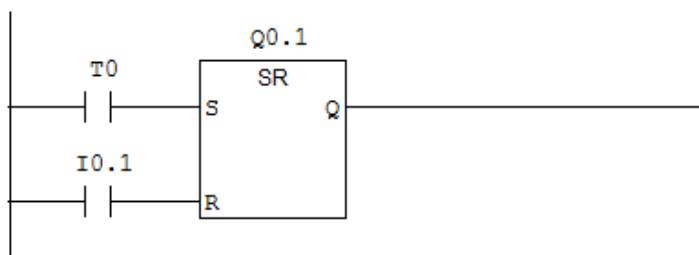
Network 2 : Title:

Comment:



Network 3 : Title:

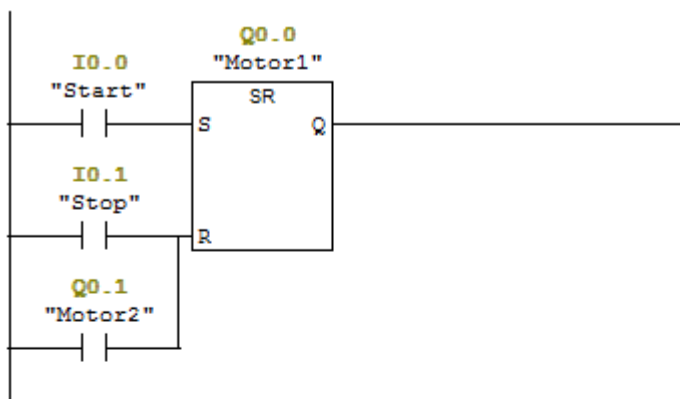
Comment:



در این مثال برای پمپ دوم از فیلیپ فلاپ جداگانه استفاده شده است.

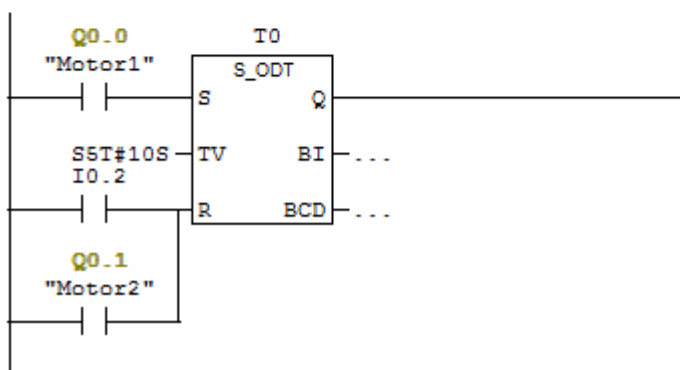
مثال 28

با زدن شاستی استارت موتور اول روشن می شود. موتور دوم با 10s تاخیر زمانی بعد از موتور اول روشن می شود. موتور دوم موتور اول را خاموش می کند.



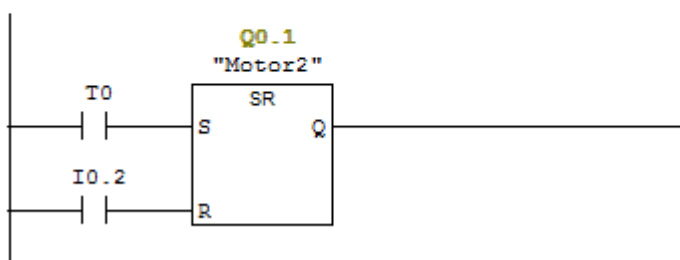
Network 2 : Title:

Comment:



Network 3: Title:

Comment:

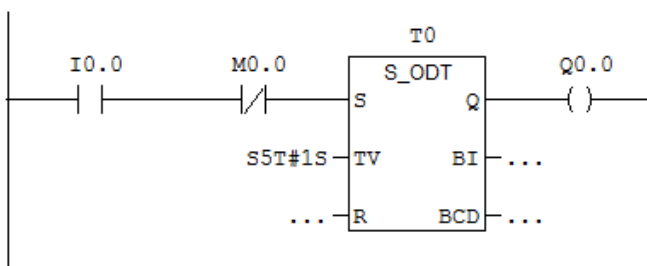


مثال 29- تولید پالس

می خواهیم برنامه‌ای بنویسیم که با فعال شدن ورودی I0.0 در Q0.0 یک قطار پالس (یک ثانیه روشن و یک ثانیه خاموش) تولید گردد.

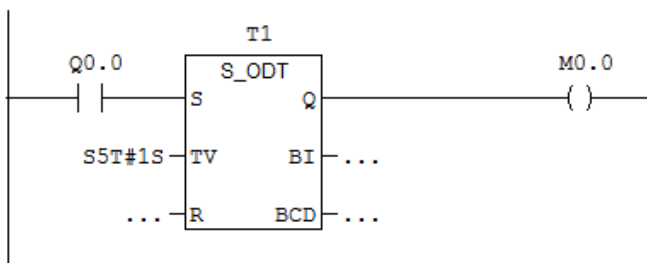
Network 1 : Title:

Comment:



Network 2 : Title:

Comment:

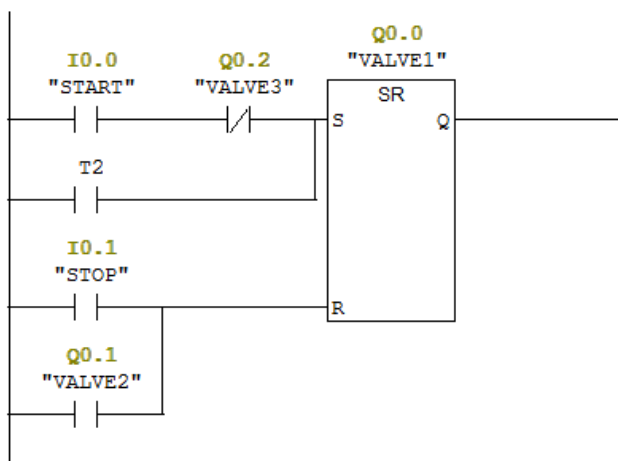


مثال 30

فرض کنید در یک پروژه صنعتی از 3 ولو در سه مسیر استفاده شده است. نحوه باز و بسته شدن ولوها بدین صورت می باشد که با فشردن شاستی استارت، ولو 1 باز و 1 دقیقه بعد ولو 2 باز و ولو 1 بسته می شود. در ادامه نیز 1 دقیقه بعد ولو 3 باز و ولو 2 بسته می شود. در مرحله پایانی نیز 1 دقیقه بعد ولو 1 مجددا باز و ولو 3 بسته می گردد. این روند تا زمانی که شاستی استپ فشرده نشده است ادامه می یابد. در ضمن لازم به ذکر است که اگر در وسط سیکل، شاستی استارت مجددا فشرده شد هیچ تاثیری در عملکرد مدار نداشته باشد.

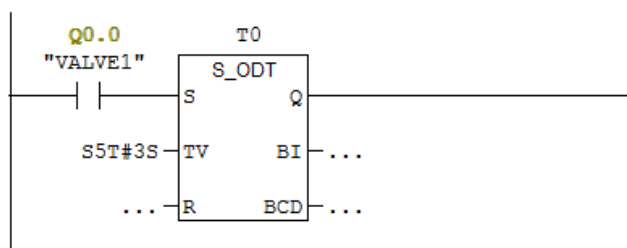
Network 1 : Title:

Comment:



Network 2 : Title:

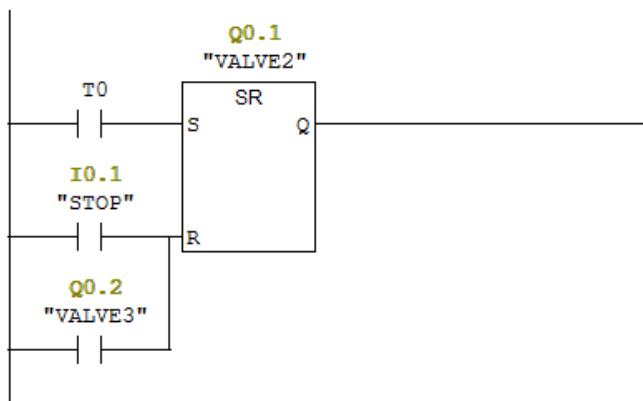
Comment:



ادامه

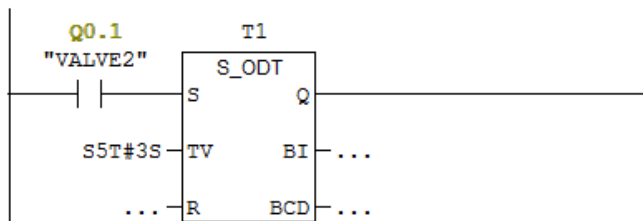
Network 3 : Title:

Comment:



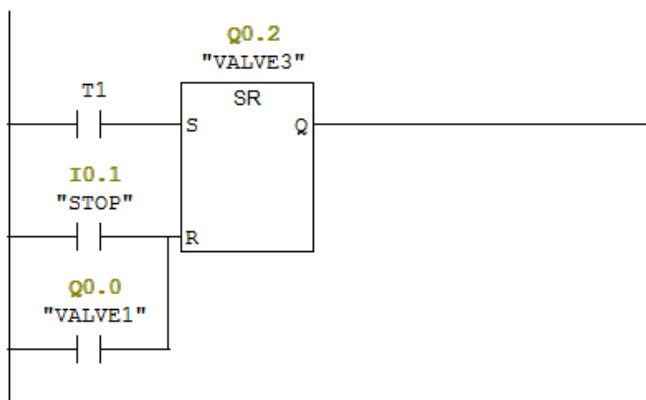
Network 4 : Title:

Comment:



Network 5 : Title:

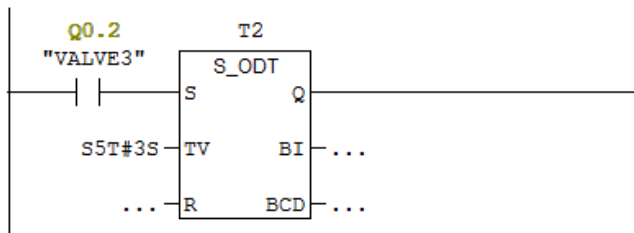
Comment:



ادامه

Network 6 : Title:

Comment:



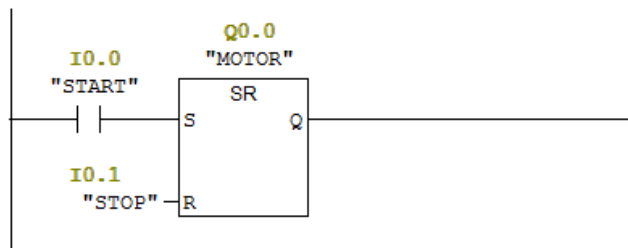
مثال 31

در یک پروژه صنعتی از یک موتور با توان بالا (Q0.0) در یک محیط بسته استفاده و در کنار این موتور نیز یک فن خنک کن (Q0.1) قرار داده شده است. نحوه کار بدین صورت می باشد که با فشردن استارت I0.0 هر دو مصرف کننده وارد مدار می شوند. اما زمانی که فرمان قطع موتور توسط I0.1 صادر گردید، موتور بلافاصله متوقف و فن خنک کن 3 دقیقه بعد از خاموش شدن موتور خاموش گردد. مطلوب است نوشتن برنامه این مثال توسط تایمر S-ODT



Network 1 : Title:

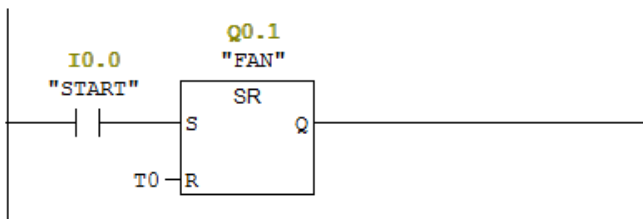
Comment:



ادامه

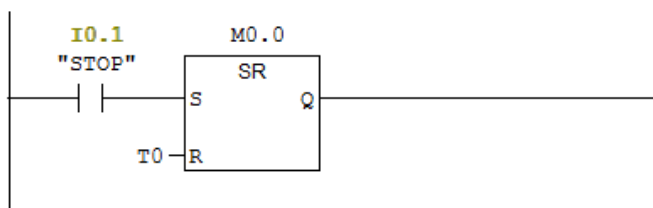
Network 2 : Title:

Comment:



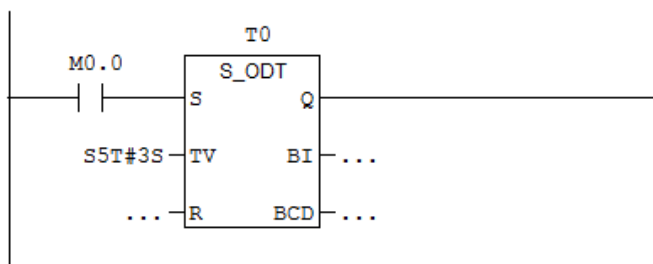
Network 3 : Title:

Comment:



Network 4 : Title:

Comment:



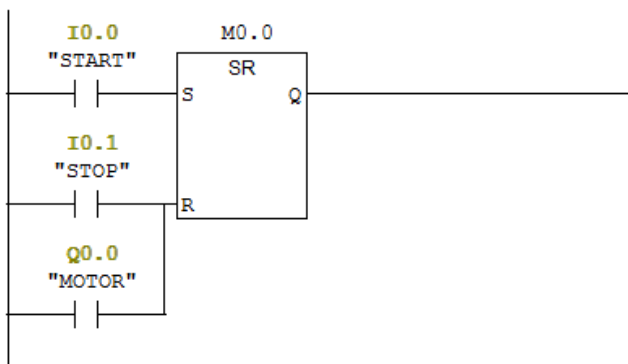
مثال 32

در این مثال می خواهیم برنامه ای طراحی کنیم که با فعال کردن ورودی I0.0 ، یک دقیقه بعد خروجی Q0.0

فعال شود. تایمر S-ODT

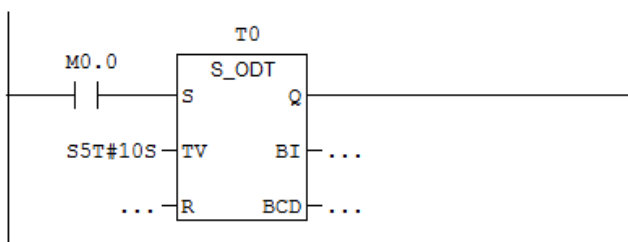
Network 1 : Title:

Comment:



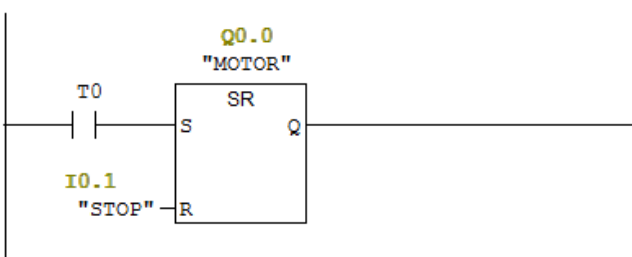
Network 2 : Title:

Comment:



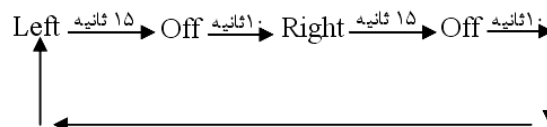
Network 3 : Title:

Comment:



تمرین 1- چپگرد / راستگرد اتوماتیک

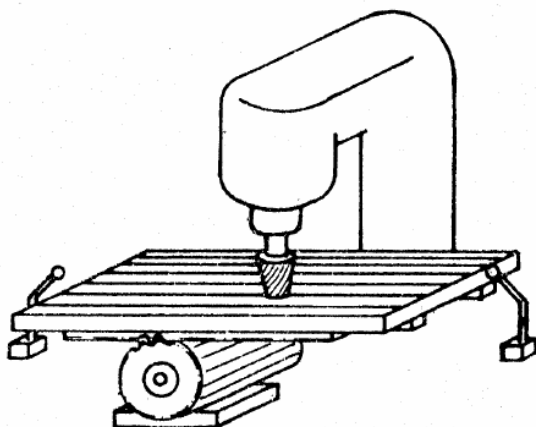
برنامه طراحی کنید که یک موتور سه فاز که به صورت چپگرد راستگرد عمل می کند به صورت زیر کنترل شود.



با توجه به شکل فوق با فرمان استارت I0.1، موتور به صورت چپگرد به مدت 15 ثانیه وارد مدار می شود. پس از سپری شدن 15 ثانیه، موتور به مدت 10 ثانیه خاموش و در ادامه کنتاکتور مربوط به راستگرد وارد مدار می شود. این روند برای حالت راستگرد نیز ادامه پیدا می کند و در پایان سیکل نیز پس از 10 ثانیه توقف مجدداً موتور به صورت چپگرد وارد مدار می شود. این سیکل تا زمانی که شاستی استپ مدار (I0.0) فعال نشود ادامه یابد.

تمرین 2- کنترل ماشین فرز کاری

یک میز مربوط به ماشین فرز کاری را در نظر بگیرید. نحوه کار بدین صورت می باشد که با روشن شدن موتور، صفحه متحرک ماشین فرز در یک جهت حرکت می کند و در پایان مسیر با برخورد به میکروسوییچ جهت حرکت موتور عکس می شود و با معکوس شدن جهت حرکت موتور جهت حرکت میز نیز تغییر می کند. این مرحله مجدداً در ابتدای مسیر با برخورد به میکروسوییچ دیگر تکرار می شود و این روند تا زمانی که شاستی استپ ماشین زده نشود ادامه می یابد.



I0.0 : استپ کل

I0.1 : استارت شروع به کار

I0.2 : میکروسوییچ ابتدای میز

I0.3 : میکروسوییچ انتهای میز

Q0.0 : کنتاکتور مربوط به چپگرد

Q0.1 : کنتاکتور مربوط به راستگرد

*هر دو میکروسوئیچ استفاده شده در این ماشین دارای یک تیغه باز می باشند.

مثال 33

در این مثال می خواهیم مثال 30 را به روش STL بازنویسی کنیم. (با استفاده از تایمر S-ODT)

Network 1: Title:

Comment:

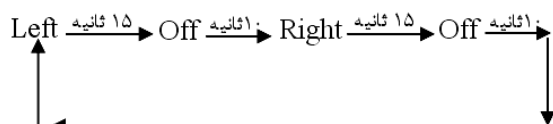
```

A      I      0.0
AN     Q      0.1
AN     Q      0.2
S      Q      0.0
A      Q      0.0
L      S5T#1M
SD     T      0
A      T      0
R      Q      0.0
S      Q      0.1
A      Q      0.1
L      S5T#1M
SD     T      1
A      T      1
R      Q      0.1
S      Q      0.2
A      Q      0.2
L      S5T#1M
SD     T      2
A      T      2
R      Q      0.2
S      Q      0.0
A      I      0.1
R      Q      0.0
R      Q      0.1
R      Q      0.2
    
```

مثال 34

در این مثال می خواهیم با زدن شاستی استارت یک موتور به صورت زیر در دو جهت کنترل شود. (طراحی توسط

تایمر S-ODT)

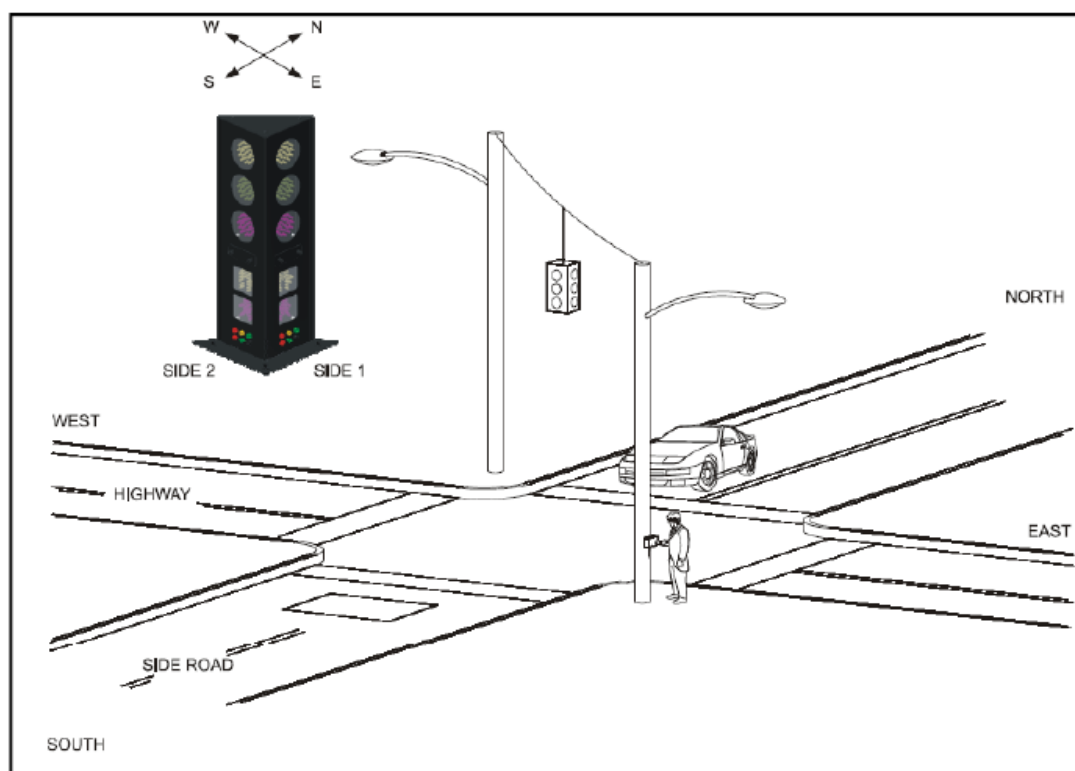


لازم به ذکر است که تا زمانی که شاستی استپ فشرده نشده است، این سیکل تکرار گردد.

A	I	0.0
AN	Q	0.1
S	Q	0.0
A	Q	0.0
L	S5T#10s	
SD	T	0
A	T	0
R	Q	0.0
S	M	0.0
A	M	0.0
L	S5T#5s	
SD	T	1
A	T	1
AN	Q	0.0
S	Q	0.1
R	M	0.0
A	Q	0.1
L	S5T#10s	
SD	T	2
A	T	2
R	Q	0.1
S	M	0.1
A	M	0.1
L	S5T#5s	
SD	T	3
A	T	3
AN	Q	0.1
S	Q	0.0
R	M	0.1
A	I	0.1
R	Q	0.0
R	Q	0.1
R	M	0.0
R	M	0.1

تمرین- در مثال فوق با اعمال شرایطی خاص، عملکرد موتور مختل می شود. مشکل را یافته و برنامه را اصلاح کنید.

تمرین: کنترل چراغ راهنمایی



کنترل چراغ راهنمایی شامل ۴ مرحله می باشد:

۱-مرحله اول: چراغ سبز اصلی به همراه قرمز فرعی

۲-مرحله دوم: چراغ زرد اصلی به همراه قرمز فرعی

۳-مرحله سوم: چراغ قرمز اصلی به همراه سبز فرعی

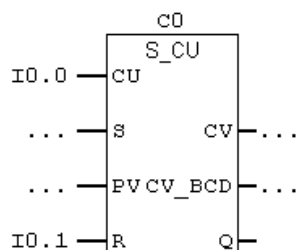
۴-مرحله چهارم: چراغ قرمز اصلی به همراه زرد فرعی

۵- بازگشت به مرحله ۱

در این تمرین جهت شروع به کار از یک کلید استارت و جهت خاموش شدن کلیه لامپ ها از یک کلید استپ استفاده می شود.

مثال 35

شمارش تعداد پالس دریافتی از یک سنسور

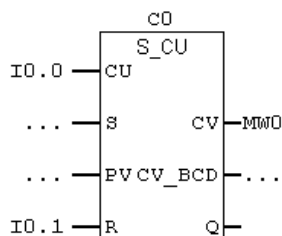


مثال 36

شمارش تعداد پالس و روشن شدن Q0.0 در صورتی که مقدار شمارنده برابر 50 شود.

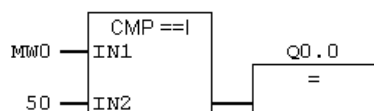
Network 1: Title:

Comment:



Network 2: Title:

Comment:

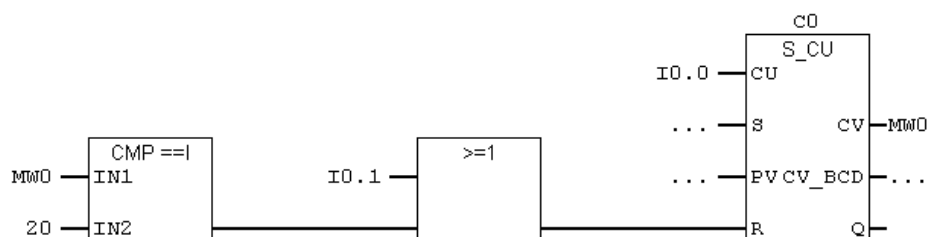


مثال 37

شمارش تعداد پالس های دریافتی از یک سنسور و روشن شدن 3 خروجی در بازه های مختلف. شمارش تا 20 ادامه یافته و مجدداً شمارنده صفر شده و این سیکل تکرار می شود.

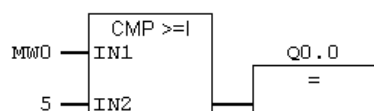
Network 1: Title:

Comment:



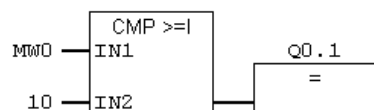
Network 2: Title:

Comment:



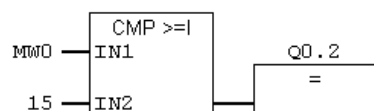
Network 3: Title:

Comment:



Network 4: Title:

Comment:



بازنویسی مثال قبل به روش STL با استفاده از دستورات SET و RESET

Network 1: Title:

Comment:

A I 0.0
CU C 0
L C 0
L 5

مثال 38

شاستی پوش باتن به روش STL

Network 1: Title:

Comment:

```

A      I      0.0
CU     C      0
L      C      0
L      1
==I
S      Q      0.0
L      C      0
L      2
==I
R      Q      0.0
R      C      0
    
```

مثال 39- خط بسته بندی

با فشردن شاستی PB1 کانوایر مربوط به جعبه ها شروع به حرکت می کند. وقتی جعبه به سنسور SE2 رسید، موتور کانوایر جعبه ها خاموش و موتور کانوایر مربوط به سیب ها روشن می شود. زمانی که تعداد 10 سیب در کارتن قرار گرفت کانوایر سیب ها متوقف و این روند برای جعبه بعدی با روشن شدن مجدد کانوایر جعبه ها تکرار می شود. تعداد سیب ها نیز توسط سنسور SE1 شمرده می شود. شاستی PB2 نیز شاستی ریست کل می باشد.

PB1: ورودی استارت I0.0

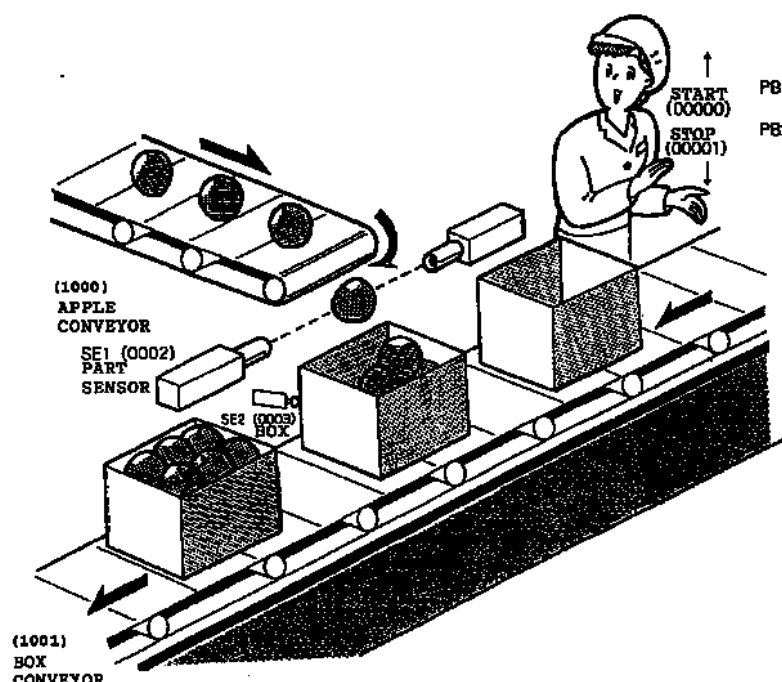
PB2: ورودی استپ I0.1

SE1: ورودی (سنسور شمارش سیب ها) I0.2

SE2: ورودی (سنسور تشخیص وجود جعبه ها) I0.3

Box Conveyor: خروجی Q0.0

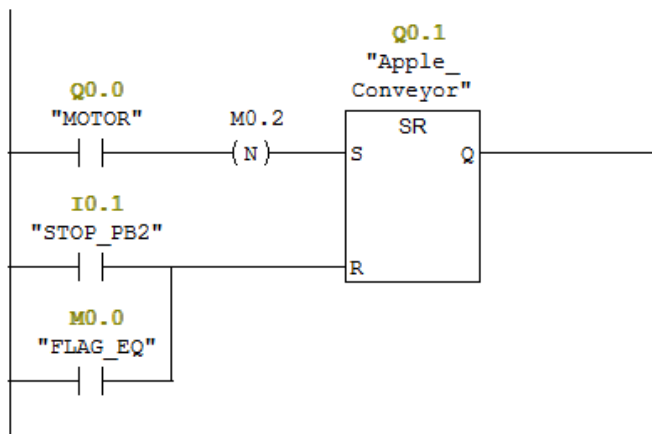
Apple Conveyor: خروجی Q0.1



برنامه

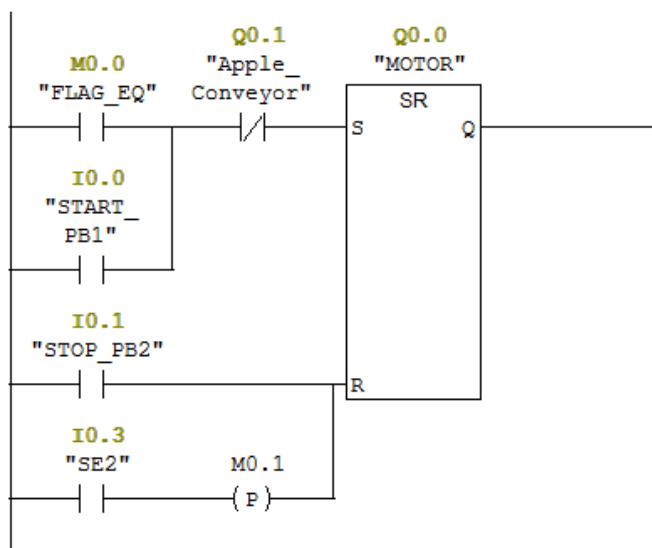
Network 2 : Title:

Comment:



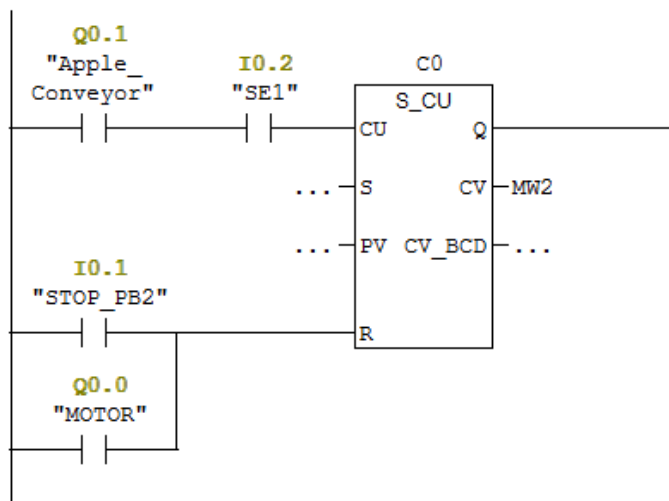
Network 1: Title:

Comment:



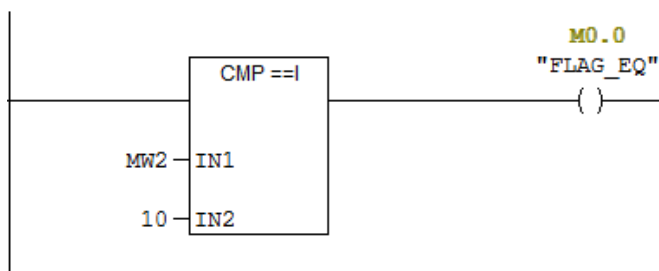
Network 3 : Title:

Comment:



Network 4 : Title:

Comment:



مثال 40

در این مثال می خواهیم تعداد پالس تولید شده از یک سنسور توسط یک شمارنده شمارش و 3 خروجی PLC در بازه های عددی مشخص شده در قسمت زیر روشن شوند. با رسیدن مقدار شمارنده به 20، هر سه خروجی PLC خاموش و این روند مجدداً با ریست شدن شمارنده تکرار شود.

* با رسیدن مقدار شمارنده به عدد 5، خروجی Q0.0 روشن شود.

* با رسیدن مقدار شمارنده به عدد 10، خروجی Q0.1 نیز روشن شود.

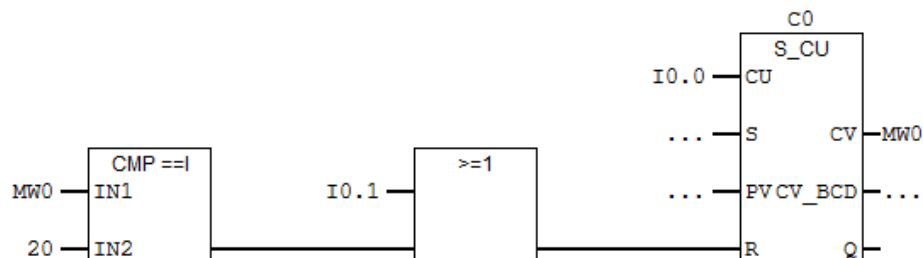
* با رسیدن مقدار شمارنده به عدد 15، خروجی Q0.2 نیز روشن شود.

* با رسیدن مقدار شمارنده به عدد 20، هر سه خروجی خاموش و شمارنده نیز ریست شود.



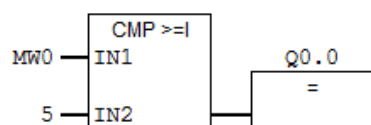
Network 1: Title:

Comment:



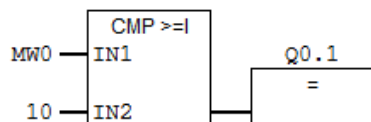
Network 2: Title:

Comment:



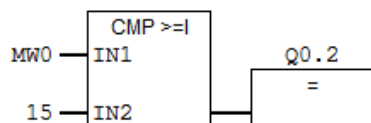
Network 3: Title:

Comment:



Network 4: Title:

Comment:



مثال 41

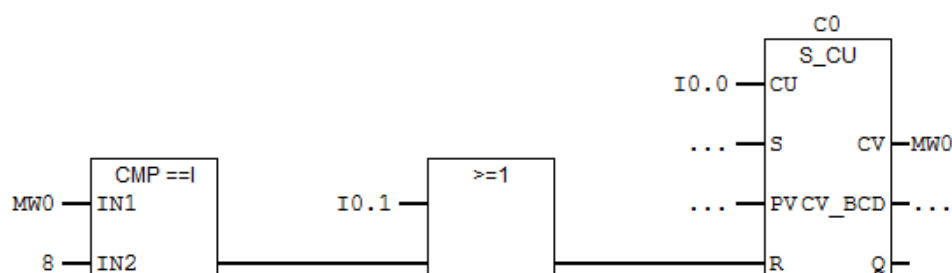
در این مثال قصد داریم برنامه ای طراحی کنیم که توسط یک شاستی بتوان 4 پمپ را به صورت زیر کنترل کرد. با هر بار فعال شدن شاستی یک پمپ وارد مدار شود. در این صورت پس از 4 بار فشردن شاستی I0.0 هر 4 پمپ روشن شوند. اگر شاستی برای بار پنجم فشار داده شد، به همان ترتیب که پمپ ها روشن شده بودند خاموش شوند. به عبارت دیگر با هر دفعه فشردن شاستی از مرحله 5، پمپ ها خاموش گردند. یعنی در مرحله 5 پمپ 4 خاموش و در مرحله 6 پمپ 3 خاموش و این روند تا پمپ 1 ادامه پیدا کند. در ضمن برنامه باید به گونه ای طراحی شود که کل این سیکل مجددا قابل تکرار باشد.



برنامه

Network 1: Title:

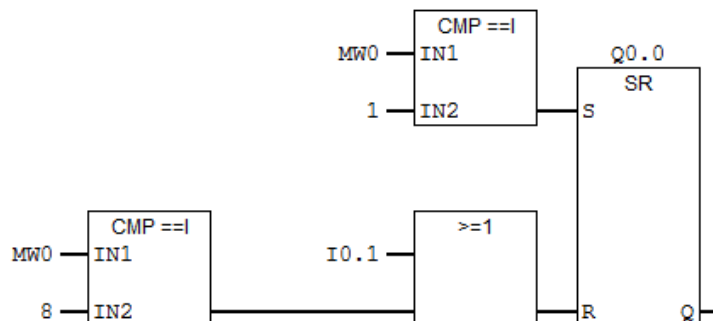
Comment:



ادامه

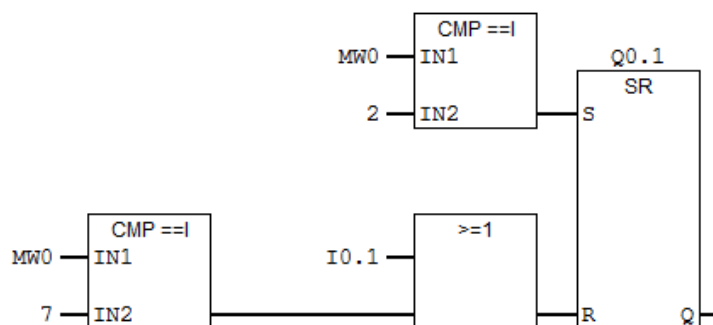
Network 2 : Title:

Comment:



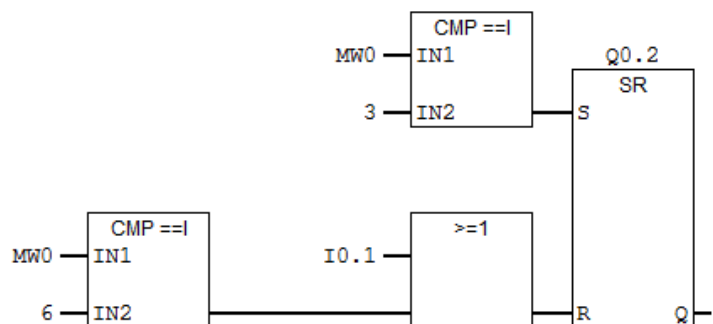
Network 3 : Title:

Comment:



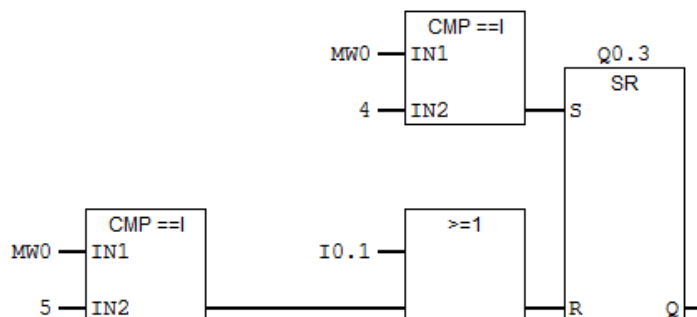
Network 4 : Title:

Comment:



Network 5 : Title:

Comment:

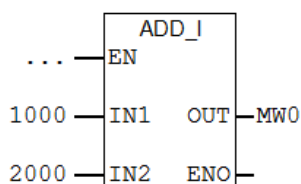


مثال 42

جمع دو مقدار 1000 و 2000 و ذخیره آن در MW0

Network 1: Title:

Comment:



مثال 43

جمع 3 مقدار 1000 ، 1500 و 2000 و ذخیره آن در MW2

Network 1: Title:

Comment:

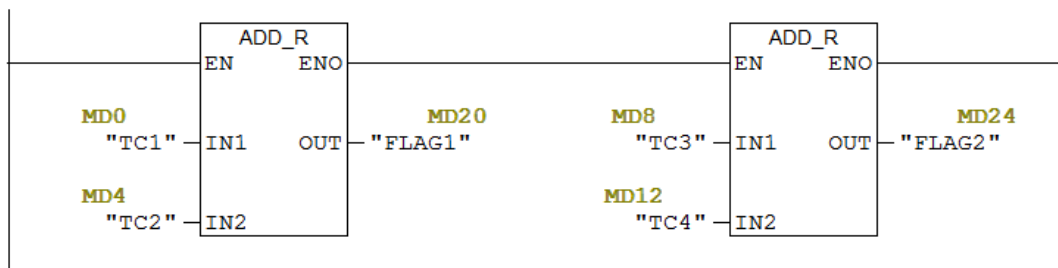


مثال 44

فرض کنیم در 4 حافظه 32 بیتی ، مقادیر مربوط به 4 سنسور دما قرار دارند. می خواهیم برنامه ای بنویسیم که میانگین این 4 دما محاسبه شود.

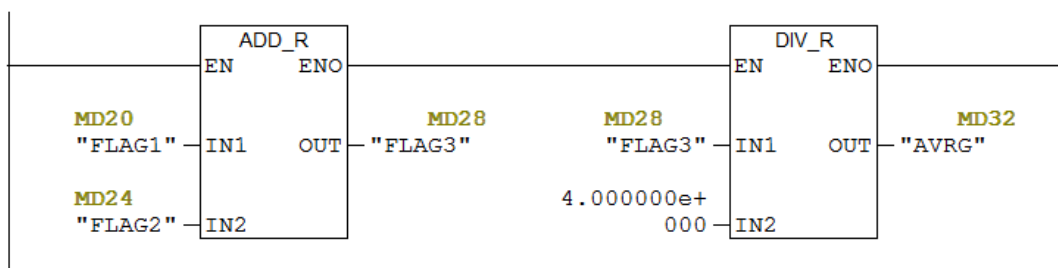
Network 1 : Title:

Comment:



Network 2 : Title:

Comment:



برای تست این برنامه به راحتی می توان در محیط شبیه ساز به فضاهای مربوط به TC1 تا TC4، مقادیر مورد نظر را به عنوان دما اعمال نمود و نتیجه نهایی را در MD32 مشاهده کرد.

بازنویسی مثال قبل توسط روش STL

Network 1: Title:

Comment:

```

L      "TC1"                MD0
L      "TC2"                MD4
+R
T      "FLAG1"              MD20
L      "TC3"                MD8
L      "TC4"                MD12
+R
T      "FLAG2"              MD24
L      "FLAG1"              MD20
L      "FLAG2"              MD24
+R
T      "FLAG3"              MD28
L      "FLAG3"              MD28
L      4.000000e+000
/R
T      "AVRG"                MD32
    
```

برنامه شکل زیر را تحلیل و بحث تعداد آکومولاتور و موضوع جانشینی در آکومولاتور بررسی شود.

Network 1: Title:

Comment:

```

L      1000
L      1500
L      2000
+I
T      MW      0
    
```

با اجرای برنامه فوق ، مقدار 3500 در MW0 قرار می گیرد.

دلیل؟

مثال 45- کنترل 3 ولو در بازه های مختلف دمایی

در یک پروسه صنعتی 4 ترموکوپل جهت اندازه گیری دما در 4 نقطه یک کوره نصب شده اند. نحوه کار بدین صورت می باشد که در برنامه کنترلی ابتدا می بایست میانگین این چهار دما محاسبه و با توجه به بازه های مختلف دمایی، شیرها باز یا بسته شوند.

در این مثال فرض می گیریم:

MD0: 1 دمای نقطه

MD4: 2 دمای نقطه

MD8: 3 دمای نقطه

MD12: 4 دمای نقطه

MD32:(AVG) میانگین دما

$0.0 < AVG < 85.0$ Valve1=Open

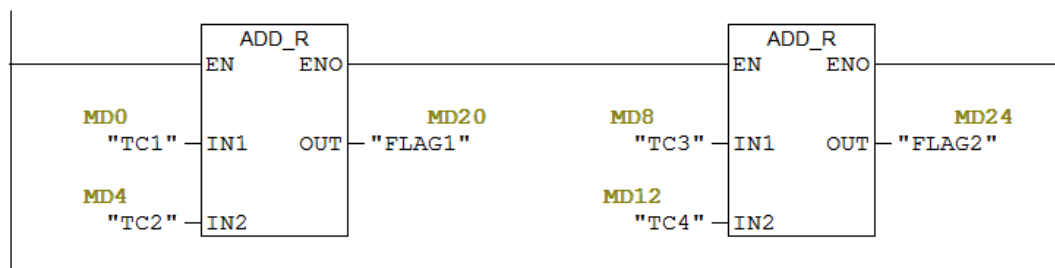
$100.0 < AVG < 150.0$ Valve2=Open

$165.0 < AVG < 250.0$ Valve3=Open

نکته: نحوه پردازش سیگنال های آنالوگ و برنامه نویسی آن در فصل های بعدی به طور کامل مورد بحث قرار خواهد گرفت. در این تمرین هدف آشنایی هر چه بیشتر با دستورات ریاضی می باشد.

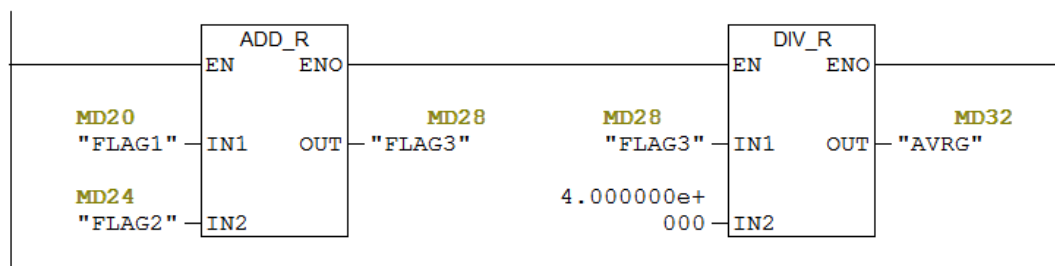
Network 1: Title:

Comment:



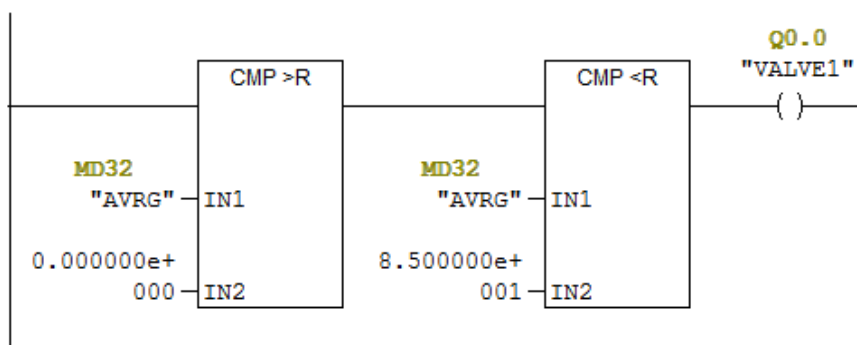
Network 2: Title:

Comment:



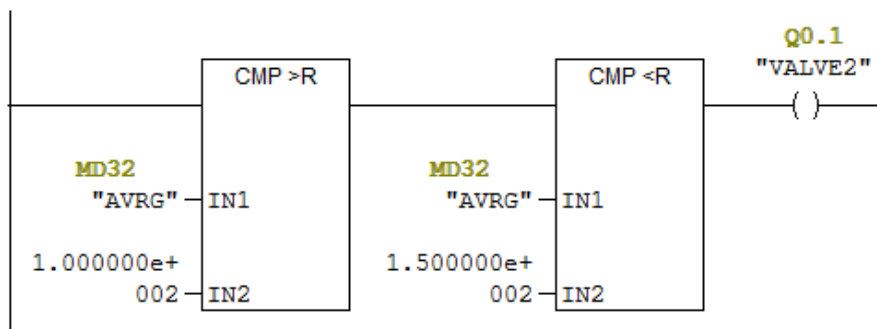
Network 3: Title:

Comment:



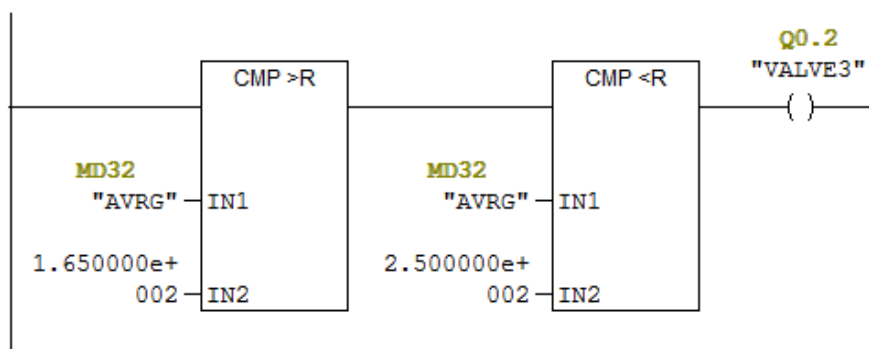
Network 4 : Title:

Comment:



Network 5 : Title:

Comment:

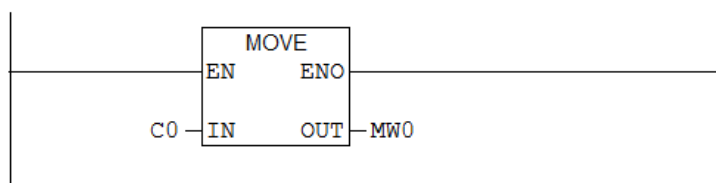


مثال 46

با اجرای برنامه این مثال همیشه و بدون در نظر گرفتن شرطی، مقدار شمارنده C1 به داخل حافظه MW0 منتقل می شود.

Network 1: Title:

Comment:

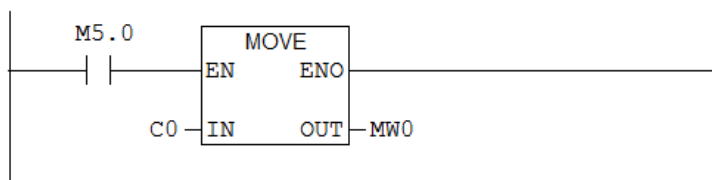


مثال 47

در برنامه این مثال جهت انتقال مقدار شمارنده C1 به MW0 حتما می بایست بیت M5.0 فعال باشد. در غیر این صورت عملیات انتقال انجام نمی شود.

Network 1: Title:

Comment:

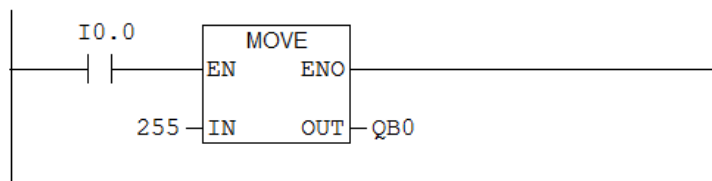


مثال 48

در این مثال با فعال شدن ورودی I0.0 ، بایت صفر خروجی یعنی QB0 روشن و با غیرفعال شدن آن نیز تمامی بیت های QB0 خاموش می شوند.

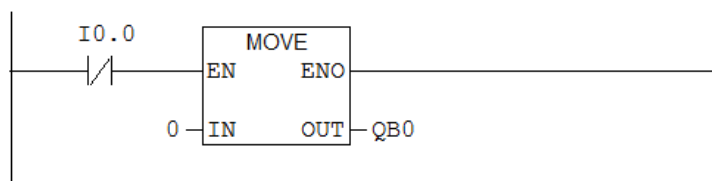
Network 1: Title:

Comment:



Network 2: Title:

Comment:



در این مثال با فعال شدن I0.0 عدد 255 به QB0 منتقل و با ارسال این عدد تمامی بیت های QB0 روشن و همچنین با غیرفعال شدن I0.0 نیز عدد صفر جهت خاموش کردن کل بایت QB0 منتقل می شود.

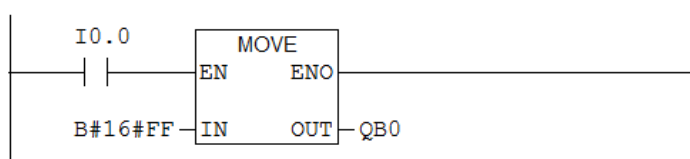
نکته

اعداد در مبنای 16 (Hex)

در S7 جهت نمایش و بارگذاری اعداد در مبنای 16 از فرمت زیر استفاده می شود.

Network 1: Title:

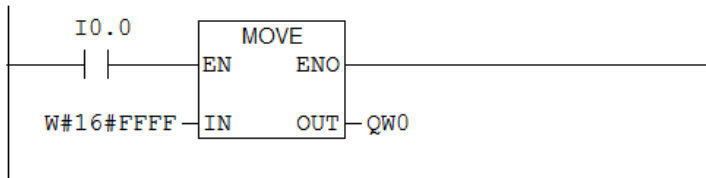
Comment:



اگر مقصد 16 بیتی باشد، از حرف W استفاده می شود.

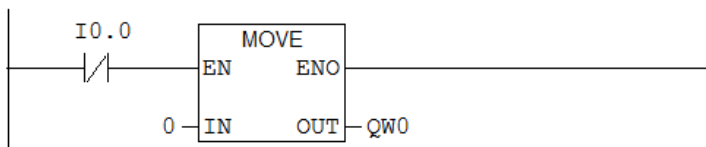
Network 1: Title:

Comment:



Network 2: Title:

Comment:



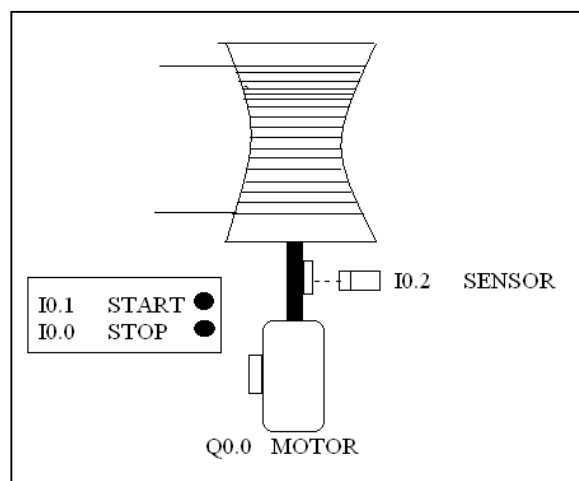
اعداد در مبنای 2 (Binary)

جهت بارگذاری اعداد در مبنای 2 از فرمت زیر استفاده می شود. X ها شامل 0 و 1 باینری می باشند.

2#XXXXXXXX

مثال 49

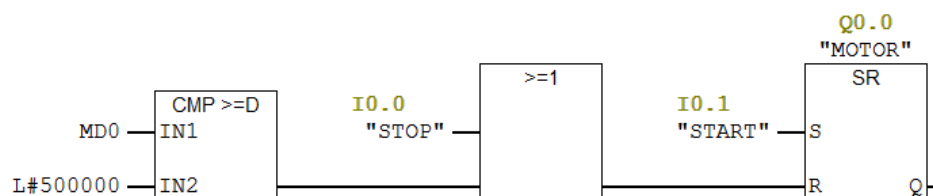
شکل زیر را در نظر بگیرید. طرز کار مدار بدین صورت می باشد که با فشردن کلید استارت موتور با سرعت کمی شروع به حرکت می نماید و عمل پیچیدن کابل شروع می شود. سپس توسط یک سنسور دور موتور شمرده می شود. می خواهیم زمانی که تعداد دور کابل به 500000 دور رسید، موتور قطع و شمارنده نیز ریست شود تا مدار برای قرقه بعدی آماده گردد.



برنامه این مثال را می خواهیم با استفاده از دستورات ریاضی طراحی کنیم. همانطور که در بحث های قبلی بیان شد، توسط دستور جمع در محدوده 32 بیتی صحیح می توان یک شمارنده 32 بیتی برای شمارش استفاده نمود.

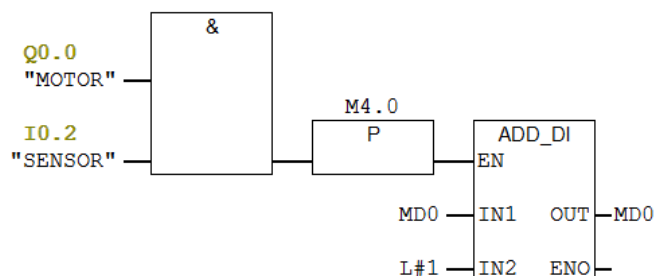
Network 1: Title:

Comment:



Network 2: Title:

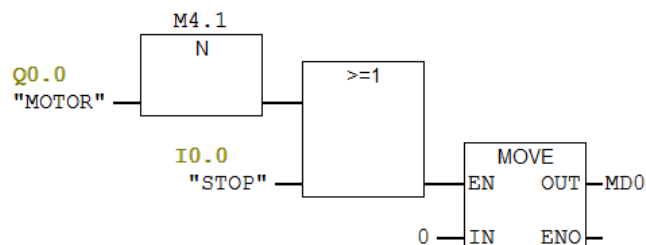
Comment:



ادامه

Network 3: Title:

Comment:



مثال 50

نوشتن لاجیک دستور MOVE شرطی به روش STL

Network 1: Title:

Comment:

```
A      I      0.0
L      255
T      QB      0
```

برنامه فوق اشتباه می باشد. در حالت عادی یک بیت نمی تواند به صورت فوق، شرط انتقال یک مقدار باشد. برای اصلاح برنامه فوق می بایست از دستورات پرش استفاده نمود.

Comment:

```
A      I      0.0
JC      M000
BEU
M000: L      255
T      QB      0
```

انواع پرش منطقی

- JC Jump if RLO = 1
- JCN Jump if RLO = 0
- JU Jump Unconditional
- JL Jump to Labels

پرش های ریاضی

- JZ Jump if Zero
- JN Jump if Not Zero
- JP Jump if Plus
- JM Jump if Minus
- JPZ Jump if Plus or Zero
- JMZ Jump if Minus or Zero

مثال 51

در این مثال با فعال شدن ورودی I0.0 ، بایت صفر خروجی یعنی QB0 روشن و با غیرفعال شدن آن نیز تمامی بیت های QB0 خاموش می شوند.

Network 1: Title:

Comment:

```

      A      I      0.0
      JC      M000
      L      0
      T      QB      0
      BEU
M000: L      255
      T      QB      0
    
```

مثال 52

در این مثال توسط کلید I0.0 می توان منطق بین دو شاستی I0.1 و I0.2 را تعیین کرد.

```

A      I      0.0
JC     M000
O      I      0.1
O      I      0.2
=      Q      0.0
BEU
A      I      0.1
A      I      0.2
=      Q      0.0
BE

```

مثال 53

در این مثال زمانی که مقدار عملیات تفریق منفی شود، کاراکتر E در MB10 قرار میگیرد.

```

L      MW      0
L      MW      2
-I
JM     M000
BEU
M000: L      'E'
T      MB      10
BE

```

مثال 54

در این مثال منفی شدن مقدار تفریق، باعث روشن شدن Q0.0 می شود. در غیر اینصورت خروجی Q0.0 خاموش می باشد.

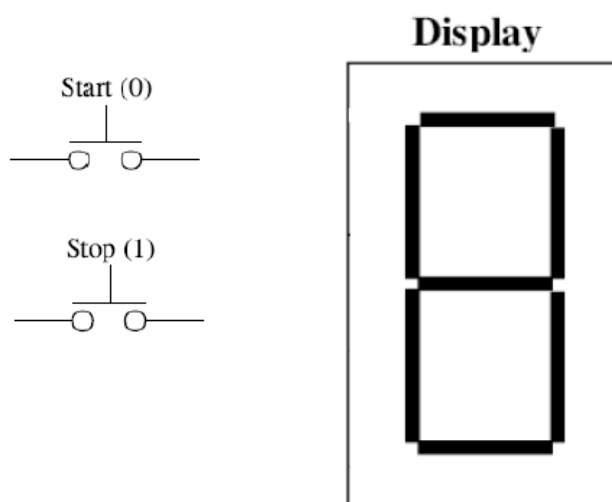
```

L      MW      0
L      MW      2
-I
JM     M000
SET
R      Q      0.0
BEU
M000: SET
S      Q      0.0

```

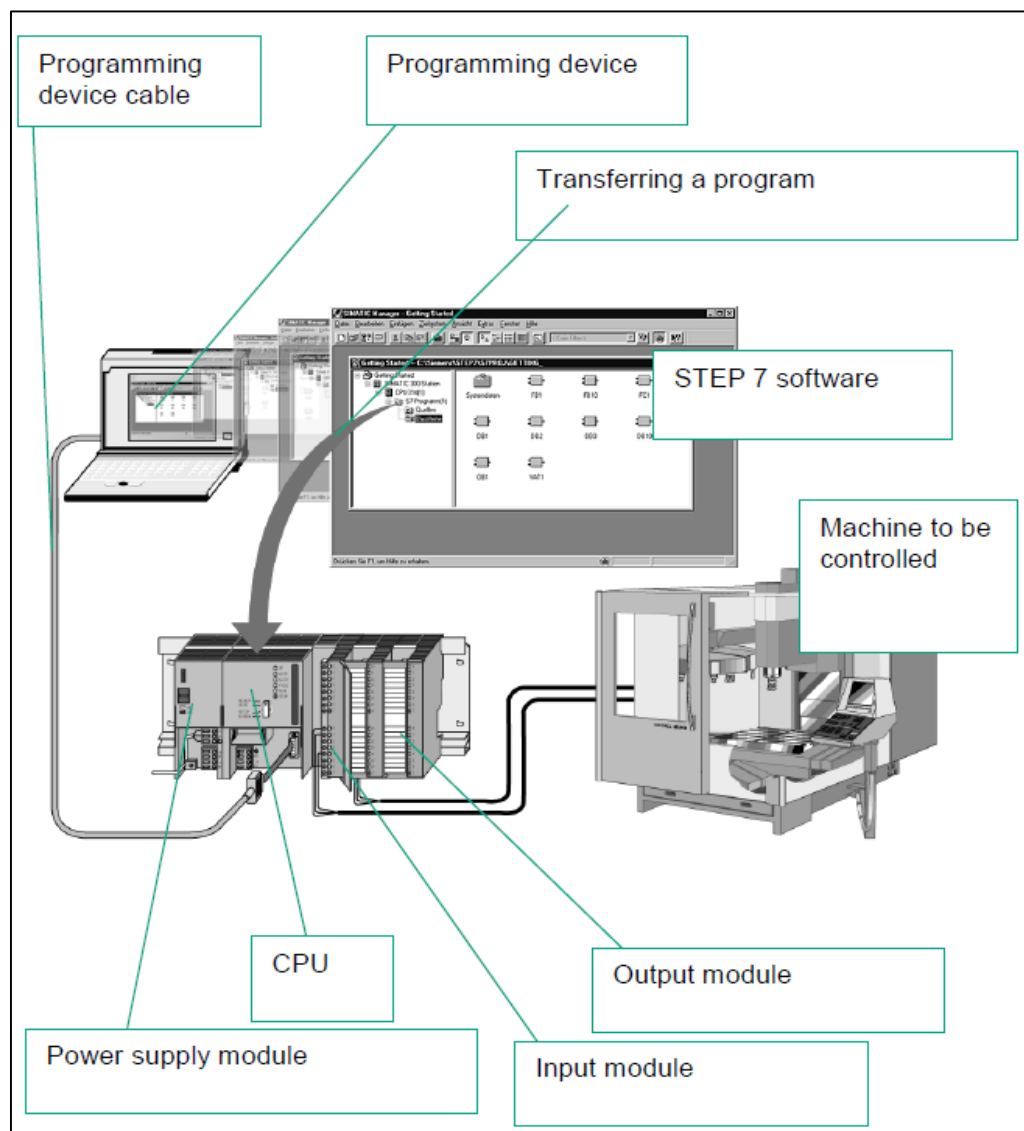
تمرین

مطلوب است طراحی برنامه مربوط به یک سون سگمنت که با فشردن شاستی استارت از عدد 0 شروع به شمارش کرده و هر یک ثانیه یک واحد به آن افزوده شود. با رسیدن به عدد 9 مجدداً صفر شده و این روند تا فشردن شاستی استپ تکرار شود. در ضمن جهت کنترل سون سگمنت از یک بایت خروجی PLC استفاده شده است. (طراحی برنامه به روش STL)



آشنایی با امکانات نرم افزار

SIMATIC MANAGER



1-وضعیت Monitor

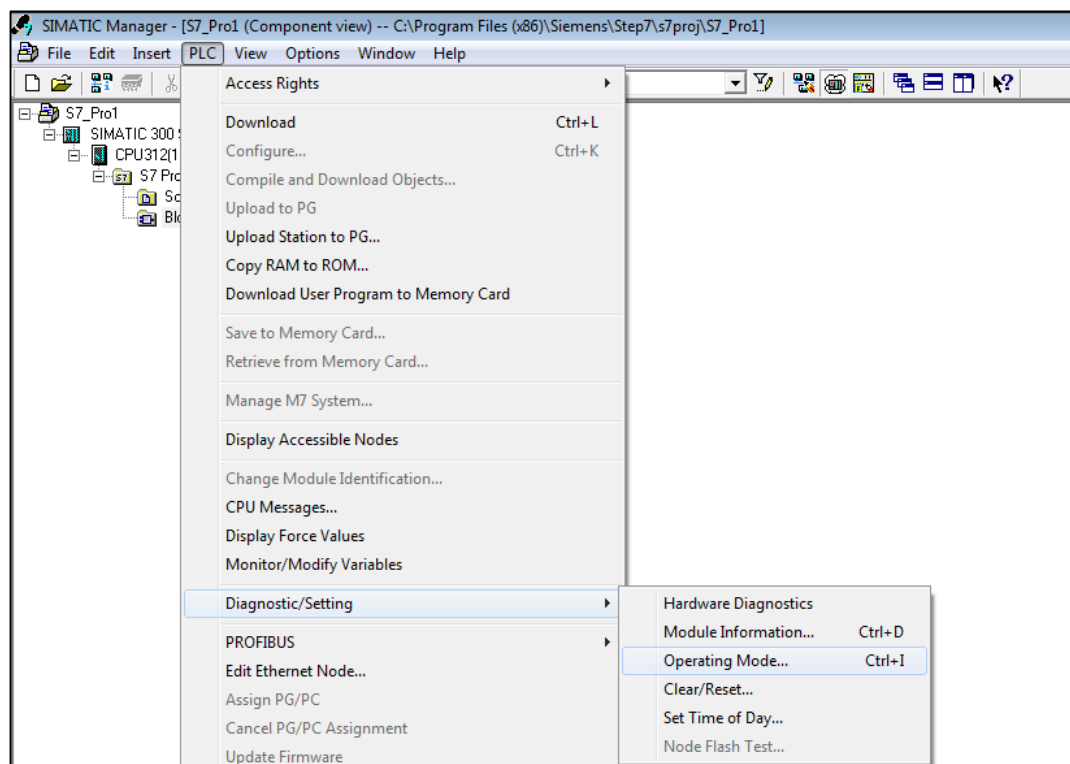
اکثر نرم افزارهای PLC دارای ابزار Monitor یا Status می باشند. ابزار مانیتور یکی از قابلیت های مهم و کاربردی در یک نرم افزار می باشد. با استفاده از این ابزار می توان وضعیت I/O ها و همچنین سایر دستورات و توابع کنترلی را در محیط نرم افزار مشاهده نمود.



آیکون مربوط به فعال و یا غیر فعال کردن وضعیت مانیتور

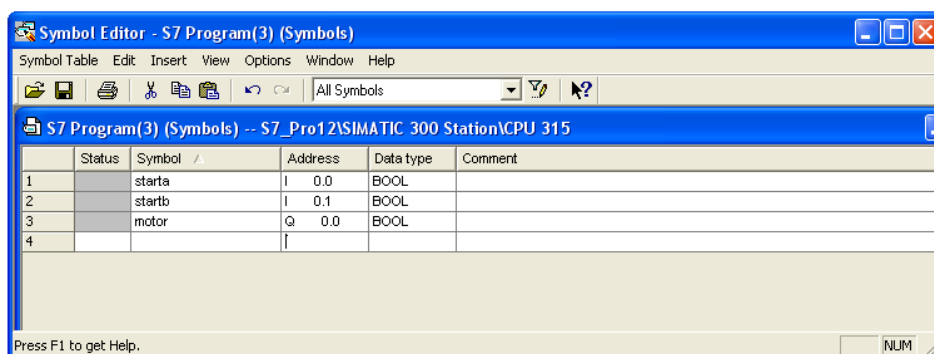
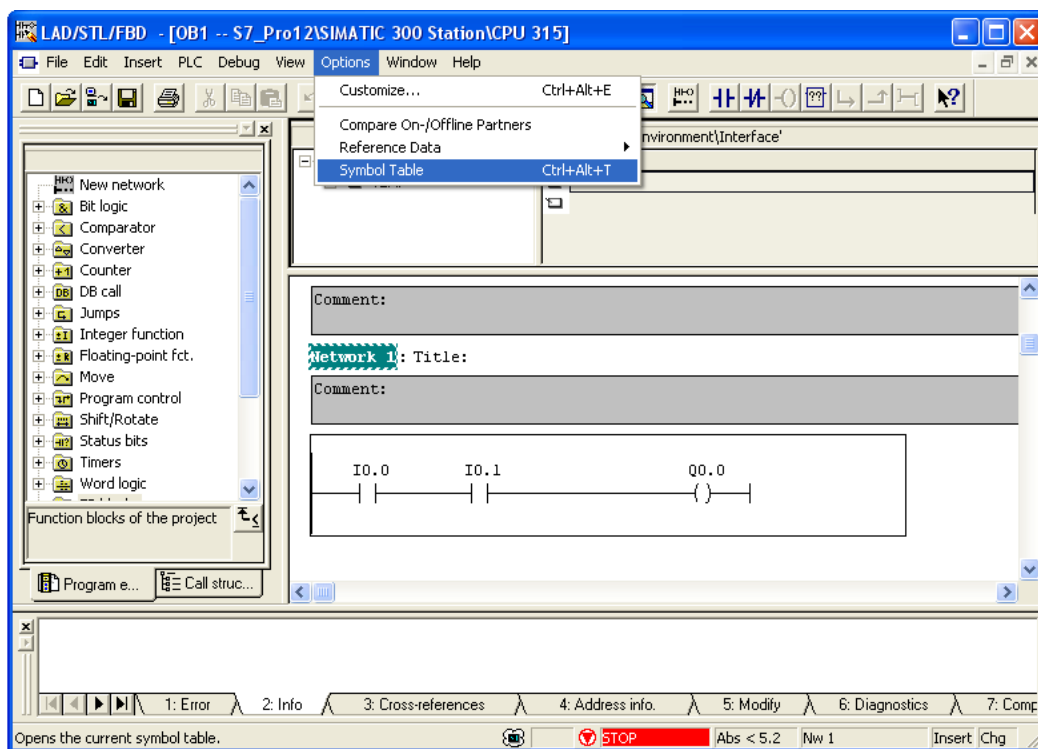
2-نحوه Run و Stop کردن CPU

در نرم افزار مورد بحث نیز امکان تغییر مد کاری CPU وجود دارد. شرط اینکه بتوان مد کاری CPU را از طریق PC تغییر داد این است که PLC در حالت سخت افزاری در وضعیت Run باشد، یعنی کلید را در مد Run قرار داده باشیم.



3-جدول سمبل ها Symbolic Table

توسط این جدول می توان بر روی آدرس های برنامه برچسب های دلخواه قرار داد.



بعد از تکمیل جدول فوق، گزینه SAVE را انتخاب تا برچسب های در نظر گرفته شده به برنامه اعمال شوند.

4- کار با جدول VAT

جدول VAT یکی از قابلیت‌های مهم در نرم افزار مورد بحث می باشد. در قسمت زیر به برخی از قابلیت‌های این جدول اشاره شده است:

1- مانیتور کردن متغیرهای I/O/M/T/C/DB

2- انجام عملیات Force

3- اعمال مقادیر جدید به متغیرها (Modify)

در این جدول تمامی متغیرها می توانند تعریف شوند.

منظور از متغیرها موارد زیر می باشند:

* ورودی

* خروجی

* فضاهای حافظه

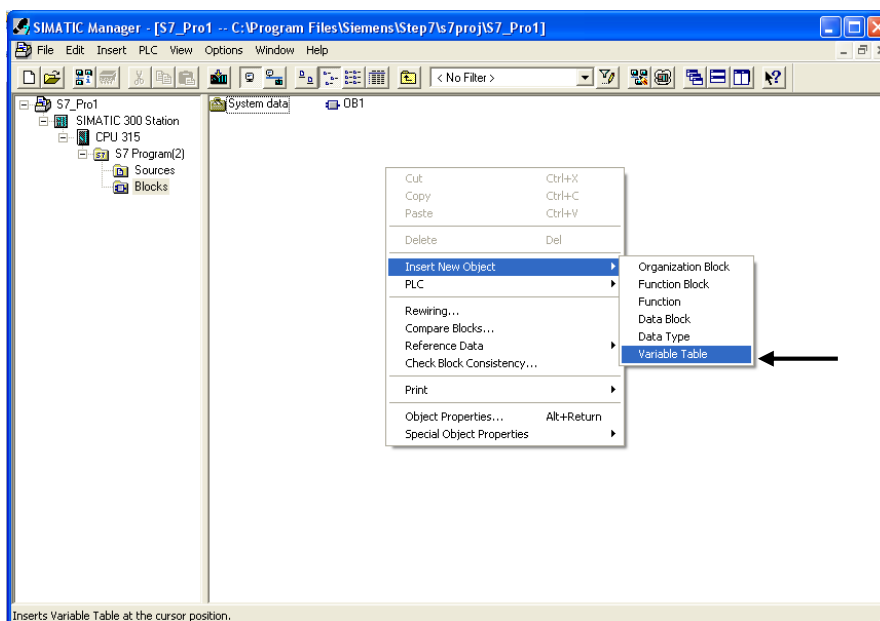
* شمارنده ها

* تایمرها

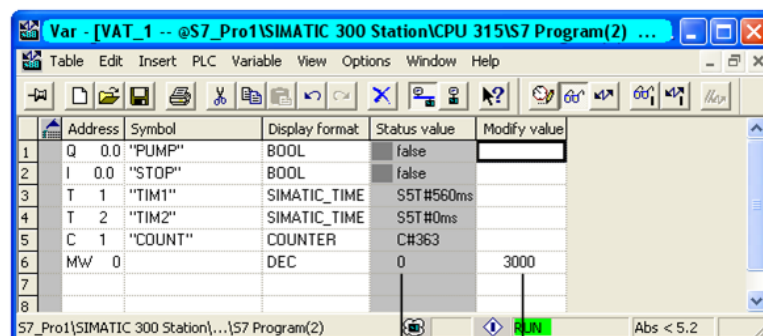
* ورودی و خروجی های آنالوگ

* فضاهای بلوک های اطلاعاتی DB

* و...



این جدول دارای یک ستون دیگر با نام Modify Value می باشد که توسط این قسمت می توان مقادیر جدیدی که به عنوان Set Point می باشند را به PLC ارسال نمود. فرض کنید عدد موجود در حافظه MW0 به عنوان S.P (نقطه مطلوب) یک پروسه می باشد. این عدد متغیر بوده و این قابلیت را دارد که محتویاتش توسط نرم افزار و بدون STOP کردن CPU توسط کاربر تغییر کند. در این صورت می توان از این جدول استفاده نمود. جهت انجام این کار ابتدا آدرس مورد نظر را در ستون های جدول وارد و عدد مورد نظر را نیز در قسمت Modify Value وارد کنید.



مقدار جدید مقدار جاری

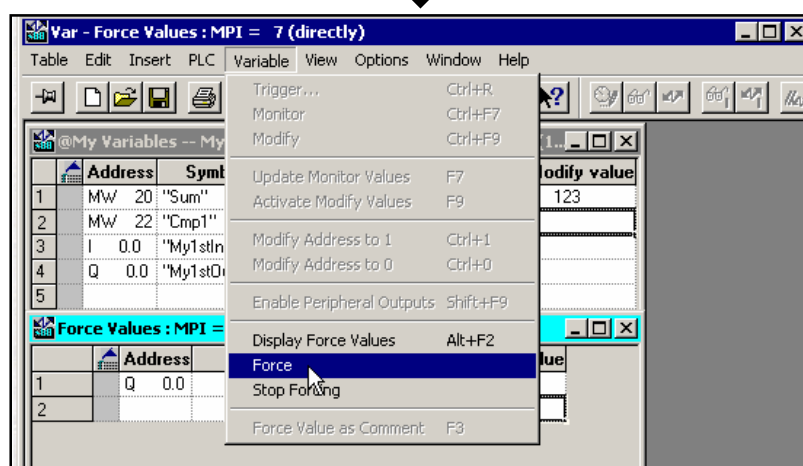
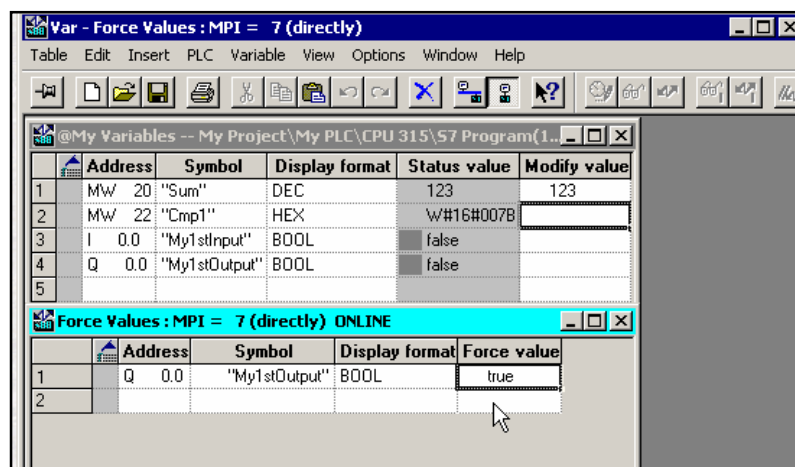
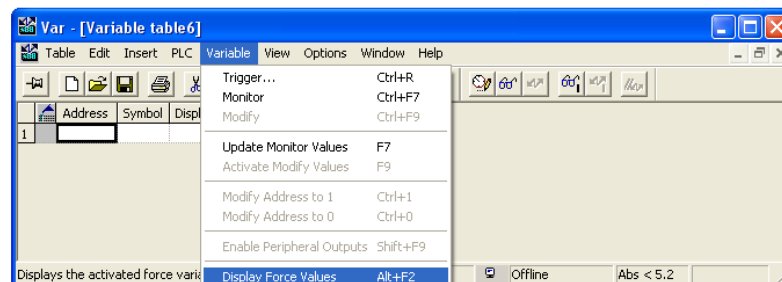
بعد از اینکه عدد مورد نظر را وارد نمودید، نوبت به ارسال این عدد به فضای مورد نظر در CPU می باشد. جهت انجام این کار کافیست در حالت مانیتور بر روی آیکون شکل زیر کلیک کنید.



آیکون Modify Value

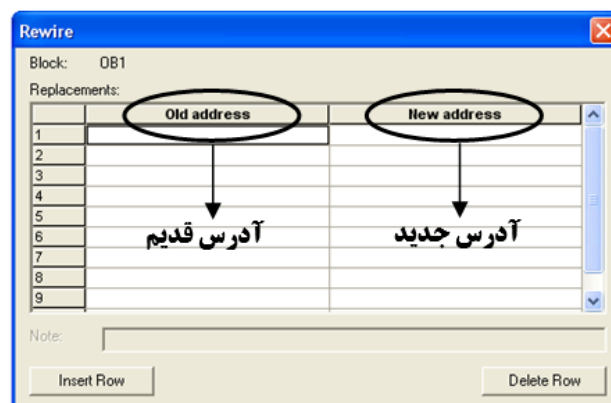
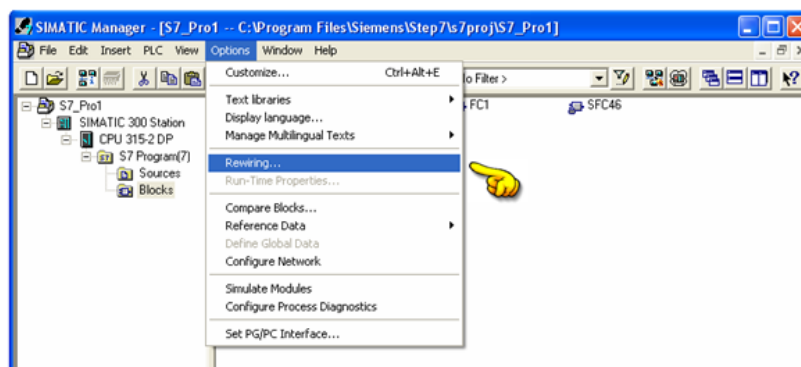
5- نحوه Force کردن متغیرها

ابتدا لازم است در مورد Force توضیح مختصری داده شود. حالت Force یکی از امکانات کاربردی در اکثر نرم افزارهای اتوماسیون صنعتی می باشد. توسط این قابلیت کاربران متخصص می توانند بر روی I/O های PLC، مخصوصا خروجی های آن از طریق نرم افزار تاثیر مناسب را بگذارند. منظور از تاثیر همان فعال و یا غیر فعال کردن آنها می باشد.



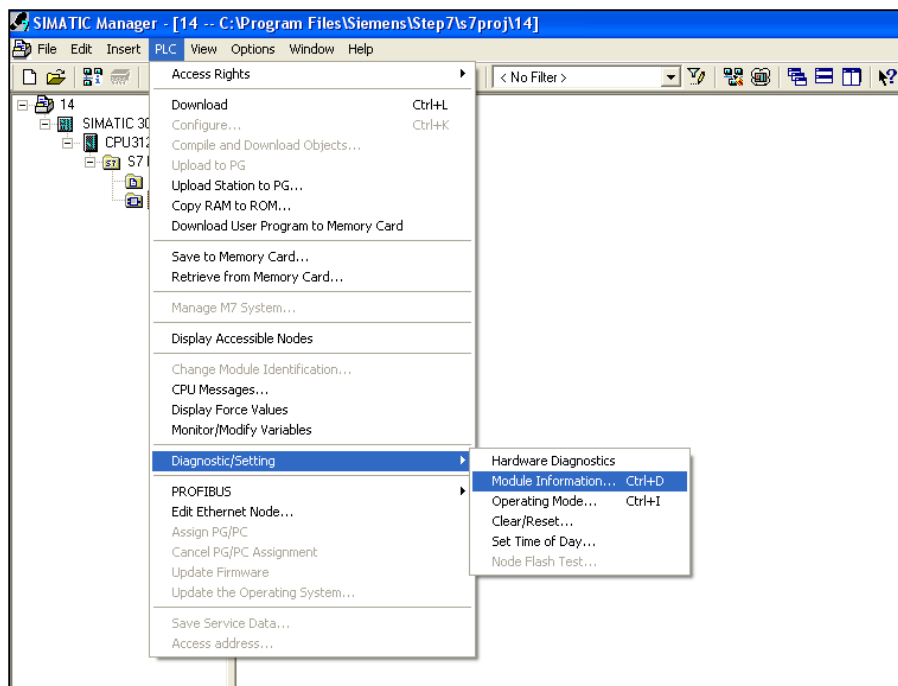
6-آشنایی با قابلیت Rewiring

توسط این قابلیت این امکان برای کاربر وجود دارد که به راحتی و در کوتاه ترین زمان بتواند در کل پروژه خود یک یا چند آدرس را به آدرس جدید تغییر دهد.



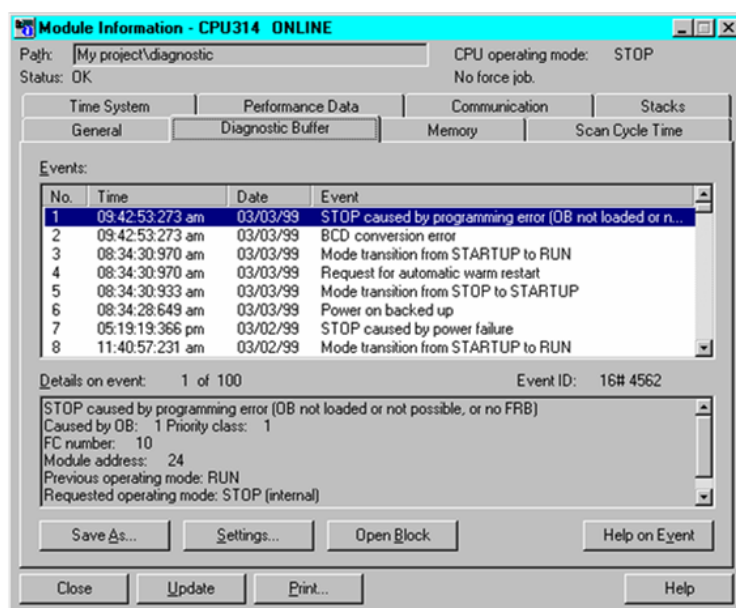
7-آشنایی با پنجره Module Information

توسط این گزینه، اطلاعات جامعی از CPU در حالت آنلاین در دسترس قرار می گیرد. این اطلاعات شامل پیغام های CPU، زمان سیکل اسکن، مشاهده فضاهای حافظه و می باشد



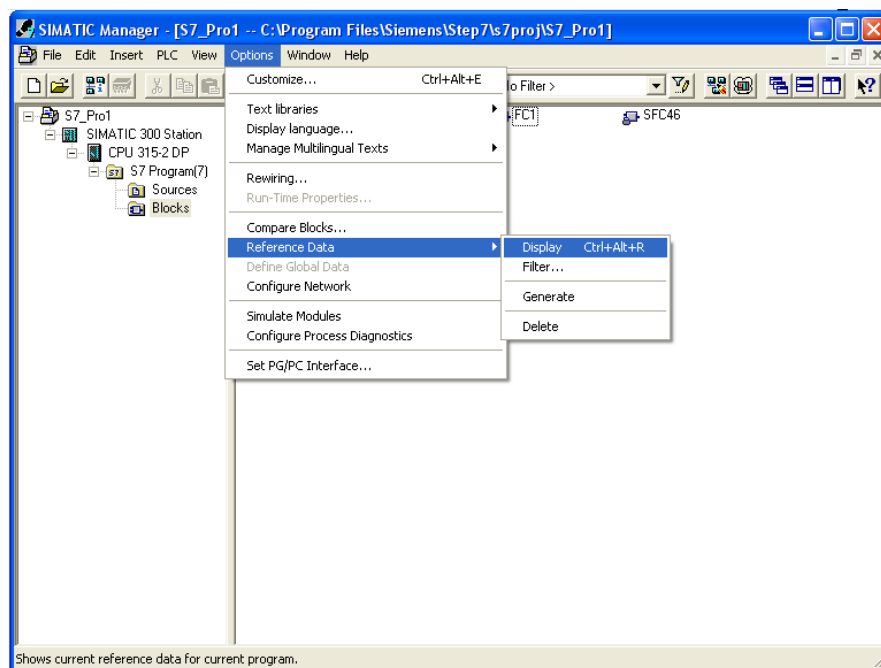
با مراجعه به Diagnostic Buffer می توان به پنجره مربوط به پیغام های CPU دسترسی پیدا کرد. تمامی فالت ها و سایر موارد به همراه ساعت و تاریخ در این بافر ثبت می شود. این بافر در بحث عیب یابی بسیار مفید

می باشد.

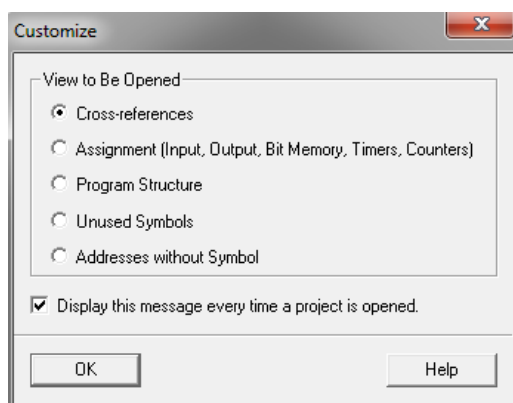


8-قابلیت Reference Data

این گزینه اطلاعات مفیدی در رابطه با برنامه در اختیار کاربر قرار میدهد. این اطلاعات شامل آدرس های استفاده شده در برنامه، ساختار بلوک های برنامه و می باشد.



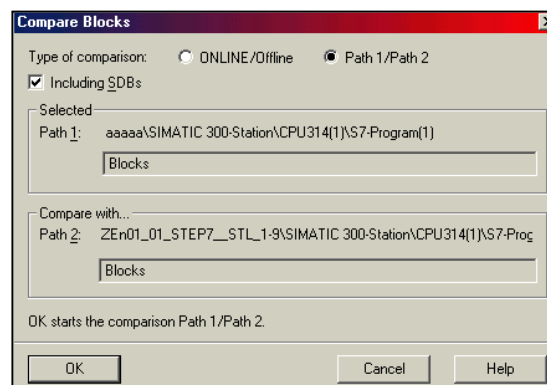
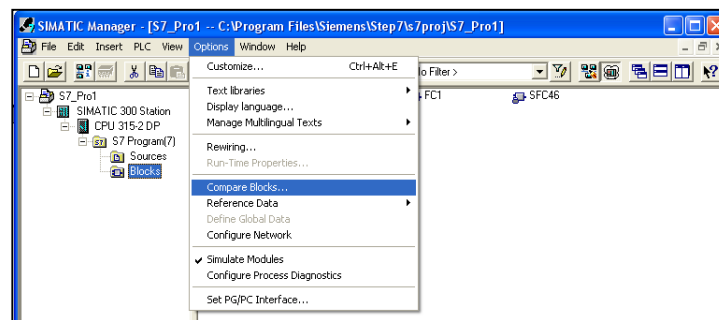
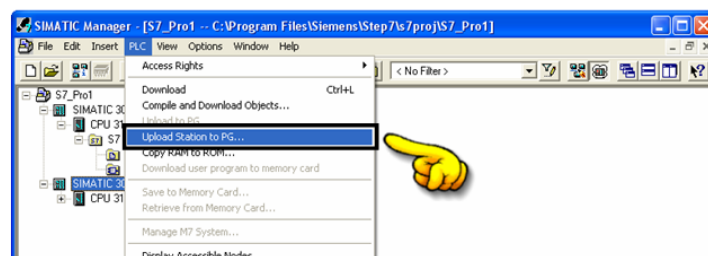
بعد از اینکه بر روی گزینه Display کلیک نمودید، پنجره ای مطابق شکل زیر نمایان می شود.



با انتخاب هر یک از گزینه های فوق، به صفحه ای جهت مشاهده اطلاعات مختلف در رابطه با بخش های مختلف پروژه مراجعه می شود.

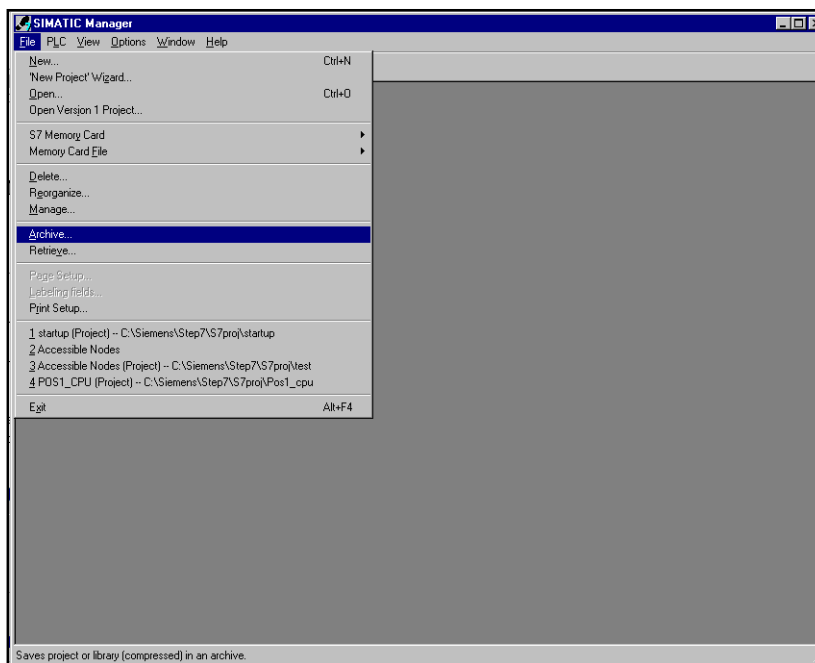
9- مقایسه بلوک ها Compare Blocks

توسط این ابزار می توان یک پروژه یا یک یا چند بلوک از یک پروژه در PC را با بلوکهای داخل PLC مقایسه نمود. عملیات مقایسه می تواند در حالت آنلاین با PLC و یا در حالت آفلاین بین دو پروژه داخل PC باشد.

**10- گرفتن Backup از کل پروژه داخل PLC**

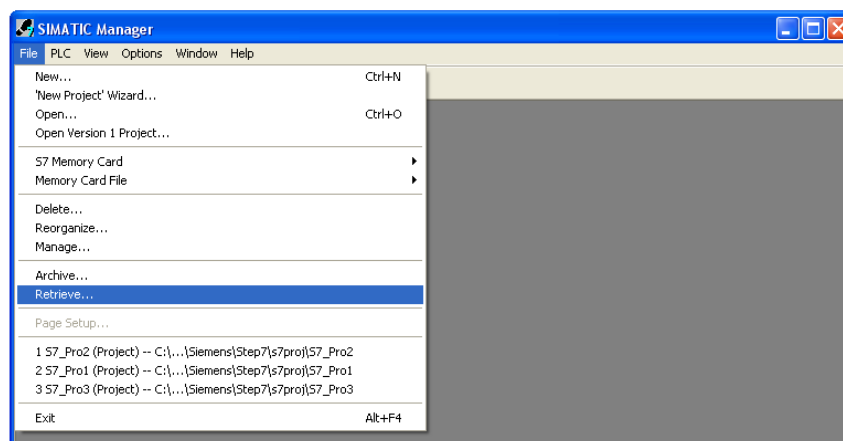
11-بایگانی پروژه Archive

توسط این قابلیت در نرم افزار کاربران می توانند پروژه خود را با حجمی بسیار کم بر روی هارد و یا یک دیسکت بایگانی کنند. این موضوع یکی از امکانات مهم در اکثر نرم افزارهای مربوط به PLC می باشد.



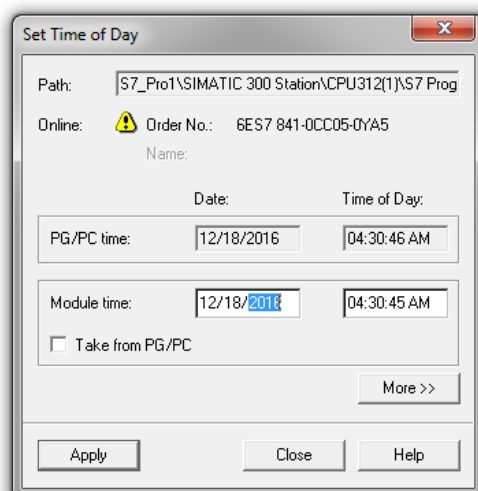
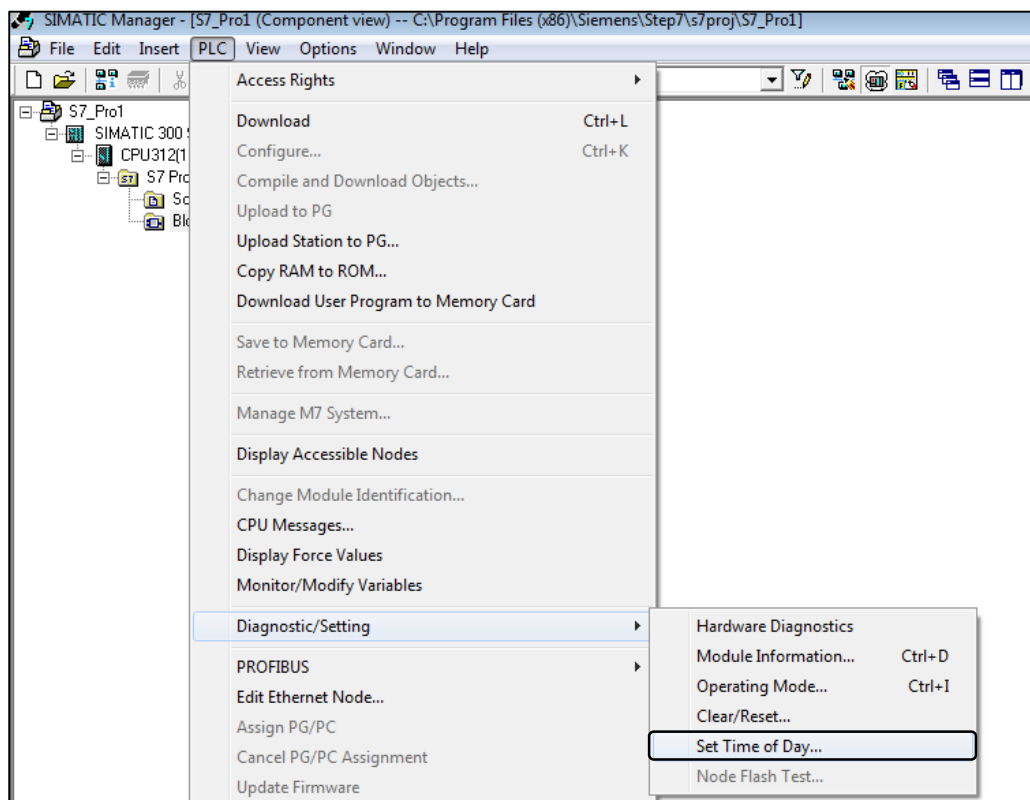
12-بازیابی پروژه بایگانی شده (Retrieve)

توسط گزینه Retrieve که در منوی File قرار گرفته شده است می توانید فایل آرشیو شده را مجدداً بازیابی کنید. پروژه ای که با قابلیت Archive ذخیره شده باشد حتماً می بایست با Retrieve باز گردد.



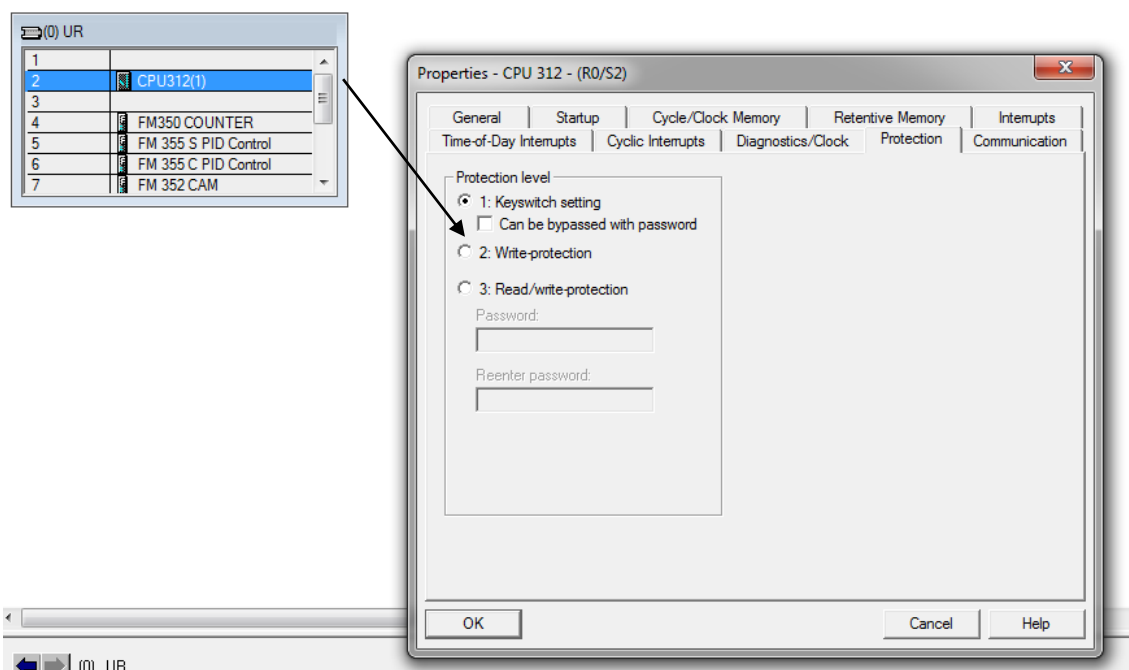
13-تنظیم ساعت و تاریخ CPU

توسط این گزینه می توان ساعت و تاریخ CPU را تنظیم و با کلیک بر روی گزینه Apply به CPU اعمال نمود.



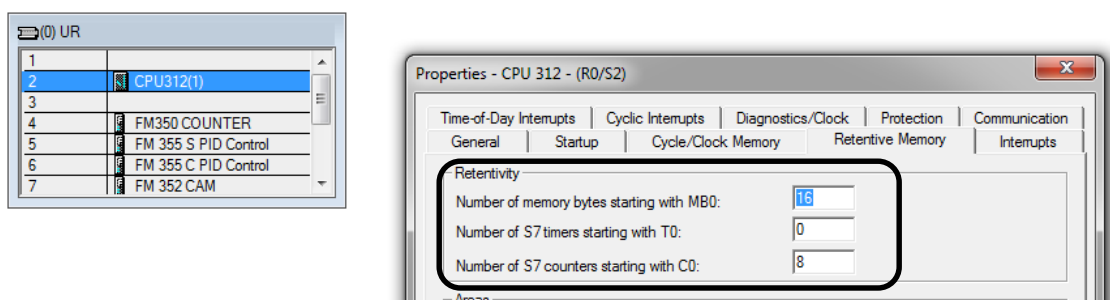
14-محافظت از برنامه CPU

توسط این گزینه می توان CPU را در مقابل نوشتن و یا خواندن/نوشتن محافظت کرد.



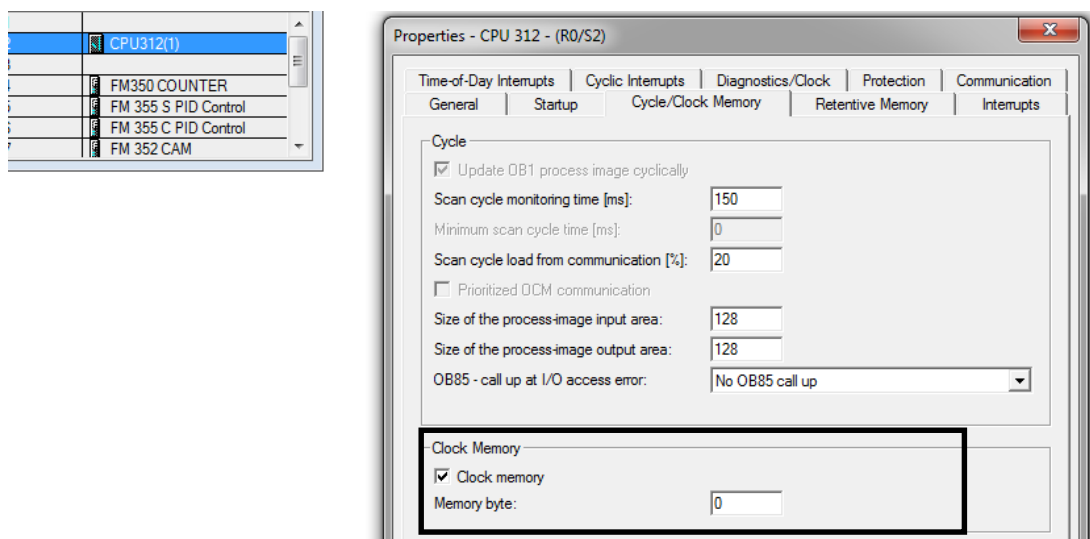
15-سازماندهی فضای حافظه

توسط این گزینه می توان فضاهای حافظه را در دو گروه پایدار و ناپایدار مدیریت کرد.



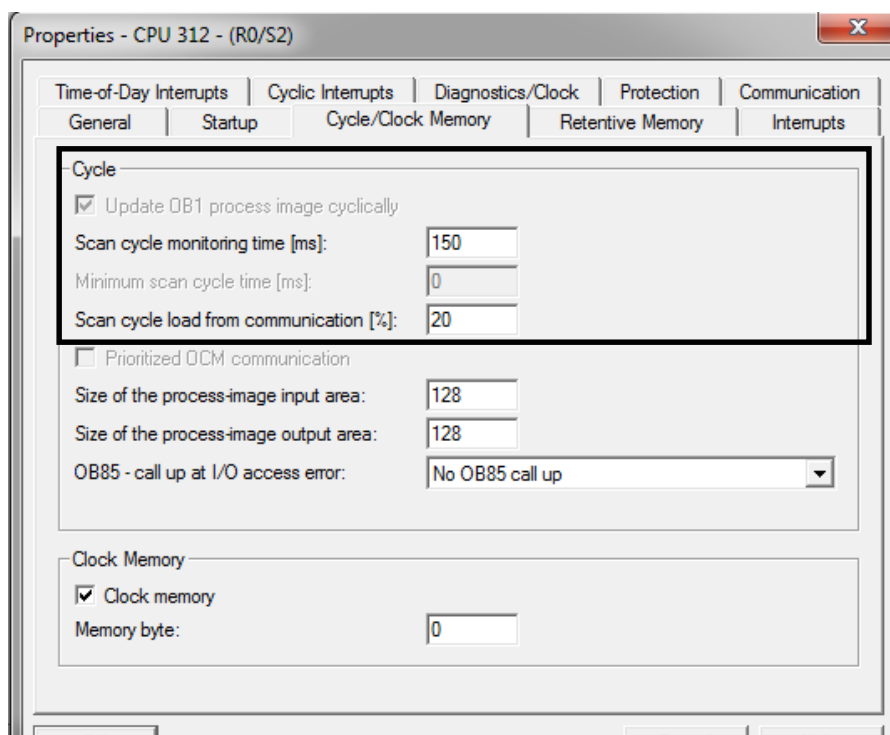
16- استفاده از پالس CPU

با استفاده از این گزینه می توان یک بایت را جهت دریافت پالس های آماده CPU اختصاص داد.

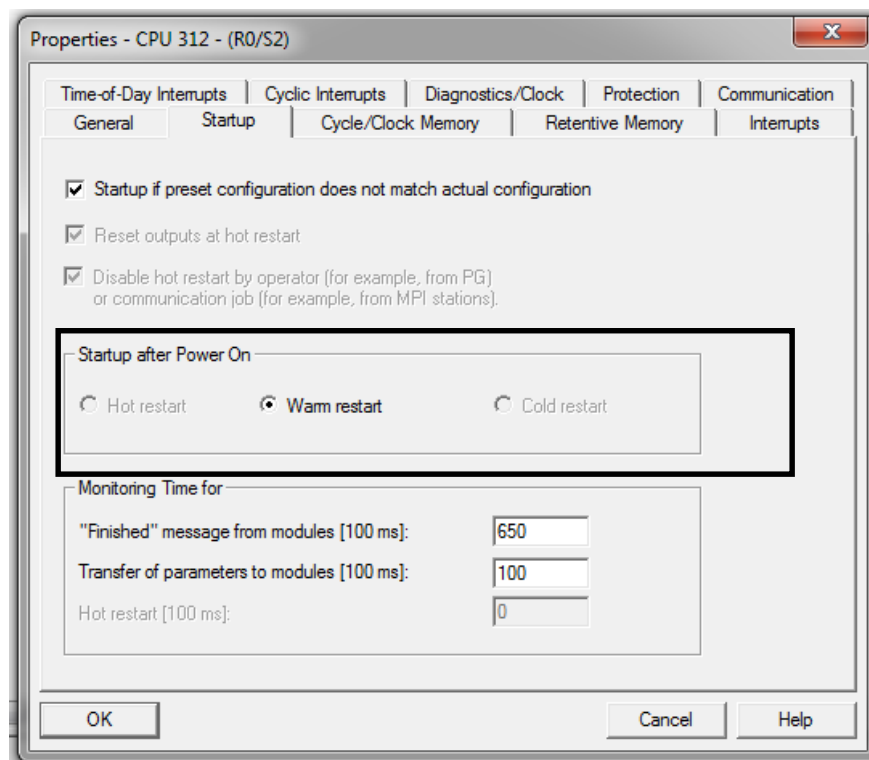


17- تنظیمات مربوط به حداقل و حداکثر زمان سیکل اسکن

توسط این سربرگ می توان حداکثر و حداقل زمان مجاز برای سیکل اسکن را مشخص کرد. تعیین حداقل زمان در CPU های 400 امکان پذیر می باشد.

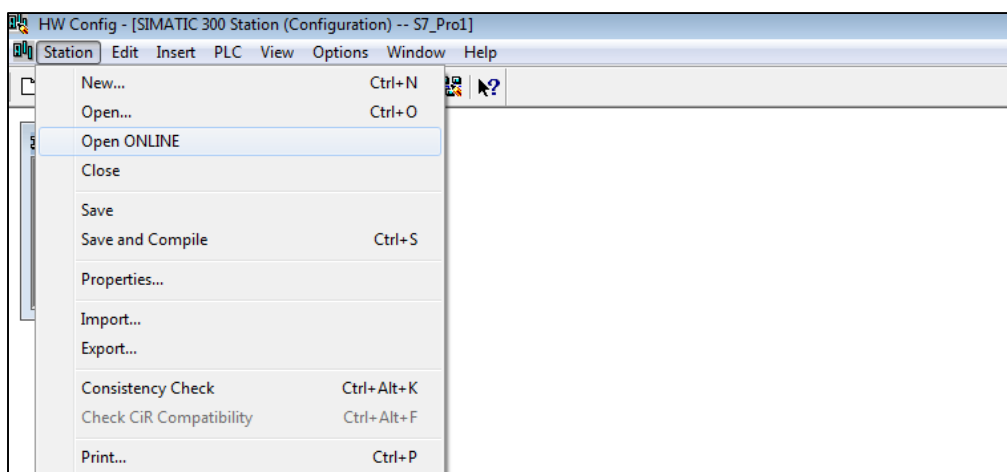


18- تعیین نوع ری استارت CPU



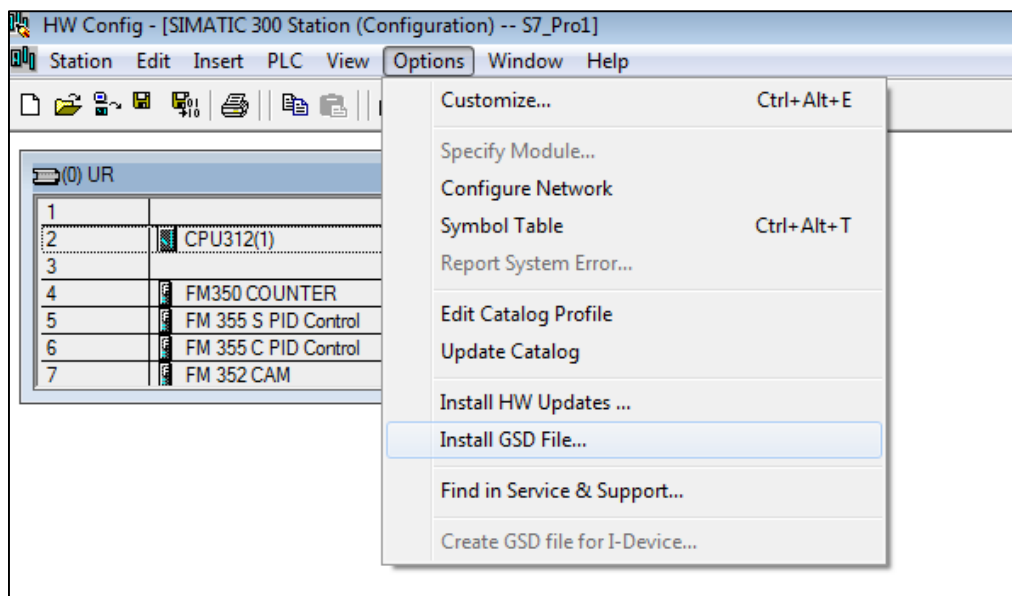
19- مشاهده آنلاین سخت افزار

توسط این گزینه می توان سخت افزار پروژه را در حالت آنلاین مشاهده نمود.



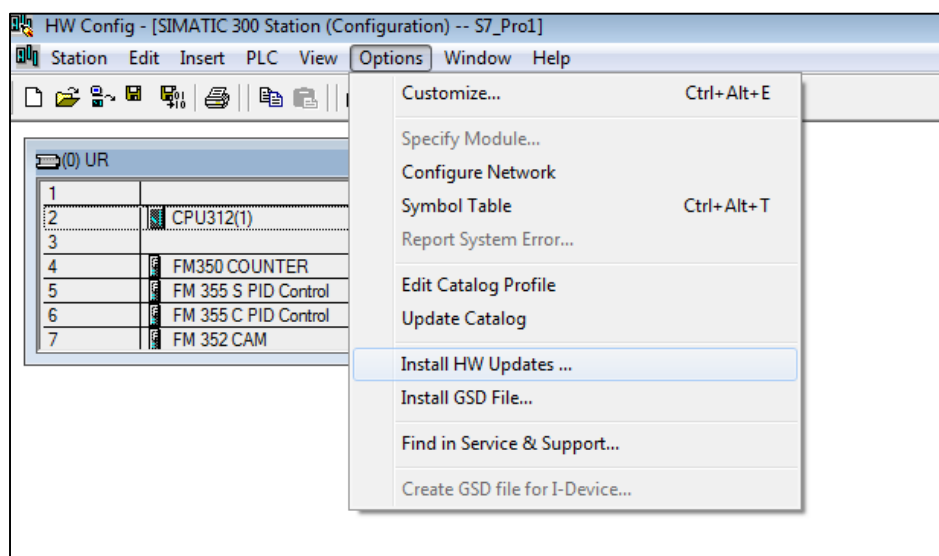
20- نصب فایل GSD

توسط این گزینه می توان تجهیزات برندهای دیگر و یا سایر محصولات زیمنس را به لیست کاتالوگ اضافه نمود. این ایستگاه ها ، توسط فایل هایی با نام GSD به لیست اضافه می شوند.



21- آپدیت کردن لیست کاتالوگ

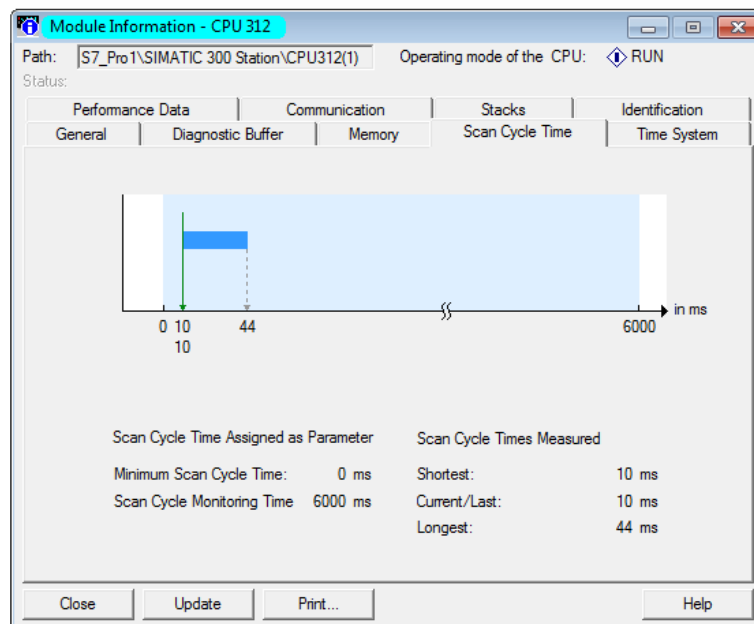
با انتخاب این گزینه، نرم افزار با اتصال به سایت زیمنس و دانلود مازول های جدید، لیست کاتالوگ را آپدیت می کند.



22-مشاهده زمان سیکل اسکن CPU

توسط این زبانه می توان مدت زمان جاری سیکل اسکن، بیشترین زمان و کمترین زمان سیکل را در حالت آنلاین

مشاهده کرد.



23-مشاهده مشخصات CPU به صورت آنلاین

توسط این زبانه می توان به مواردی از قبیل بلوک های سیستمی، تعداد تایمر، کانتر و بلوکهای اینترپت که یک

CPU می تواند ساپورت می کند.

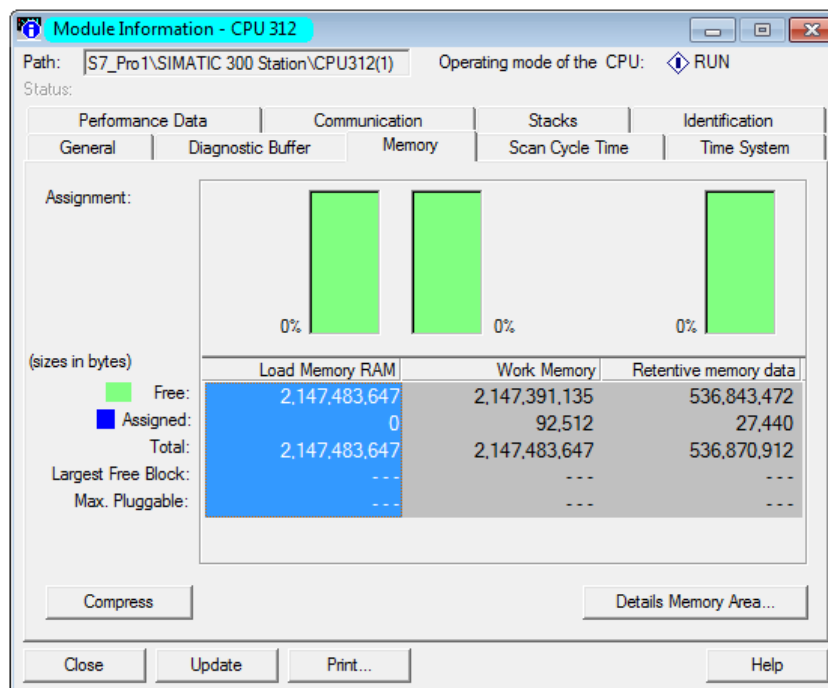
No.	Function
OB1	Free scan cycle - start event: sta...
OB10	Time-of-day interrupt - start eve...
OB11	Time-of-day interrupt - start eve...
OB12	Time-of-day interrupt - start eve...
OB13	Time-of-day interrupt - start eve...
OB14	Time-of-day interrupt - start eve...
OB15	Time-of-day interrupt - start eve...
OB16	Time-of-day interrupt - start eve...

No.	Name	Symbol Comment
SFB0	CTU	Count Up
SFB1	CTD	Count Down
SFB2	CTUD	Count Up/Down
SFB3	TP	Generate a Pulse
SFB4	TON	Generate an On ...
SFB5	TOF	Generate an Off ...
SFB8	USEND	Uncoordinated S...
SFB9	URCV	Uncoordinated ...

Address type	Quantity	Area from	to / max. length
Process Image Inputs	131072 (Bits)	I0.0	I16383.7
Process Image Outputs	131072 (Bits)	Q0.0	Q16383.7
Bit Memory	131072 (Bits)	M0.0	M16383.7
Timers	2048	T0	T2047
Counters	2048	C0	C2047
Local Data	32768 (Bytes)		

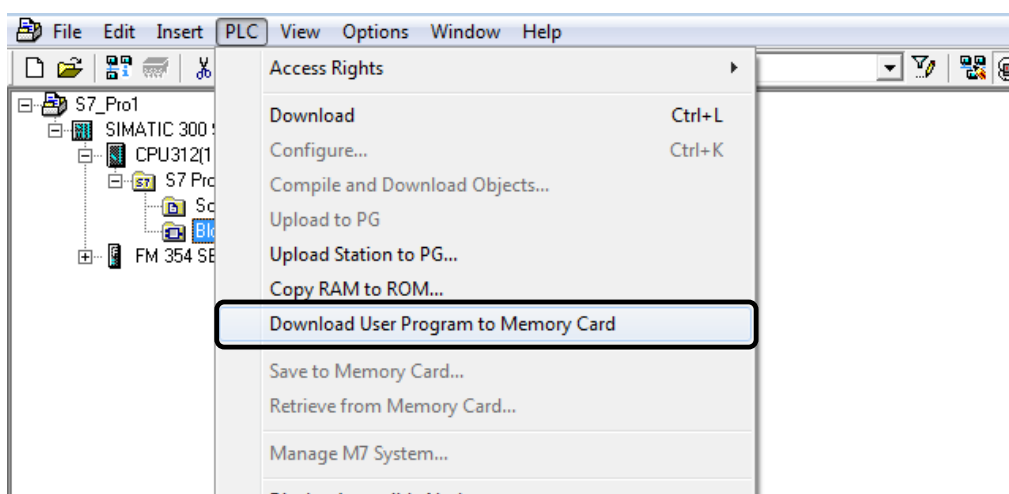
24-مشاهده حجم فضای حافظه استفاده شده

توسط این ابزار می توان مقدار فضای حافظه Load Memory و System Memory مربوط به CPU را مشاهده و در رابطه با فضای اشغال شده اطلاعات لازم را کسب نمود.

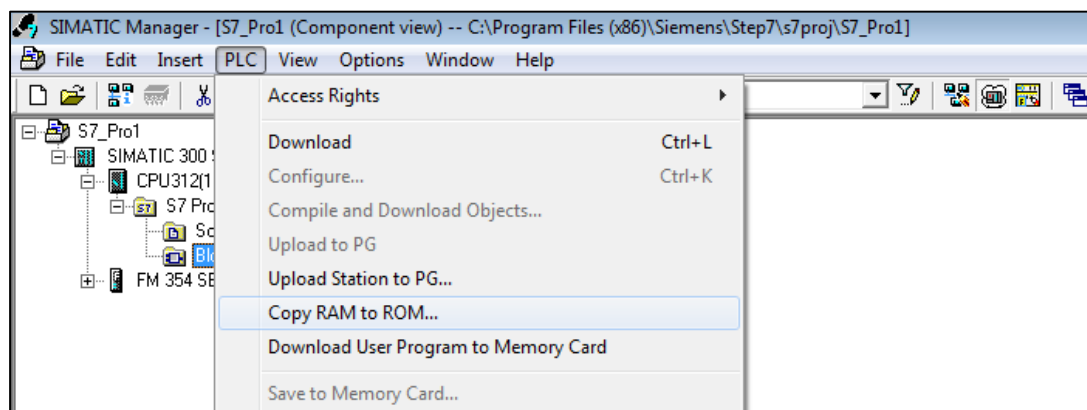


25-انتقال برنامه PC به روی کارت MC از نوع فلش

توسط این گزینه در حالت STOP می توان پروژه را به صورت کلی بر روی کارت حافظه MC دانلود نمود.

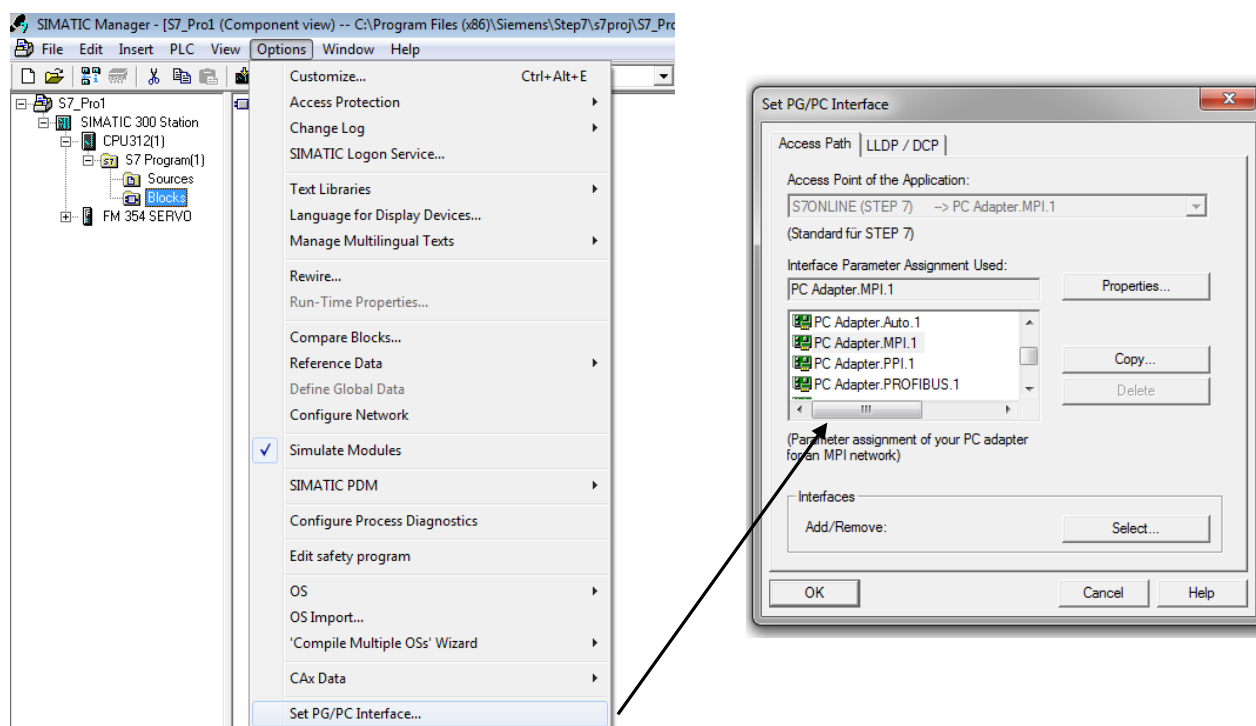


26- انتقال برنامه از روی RAM داخلی به روی ROM داخلی (CPU 314 IFM)



27- تنظیمات درایورها و کابل های ارتباطی

توسط این گزینه می توان تنظیمات مربوط به کابل های ارتباطی نصب شده را انجام داد.



شرکت فنی و مهندسی

نوآوران صنعت پارسه

(مشاور، طراح و مجری پروژه های برق و اتوماسیون صنعتی)

Consulting, designing and executing of Industrial Automation Projects

SIEMENS



OMRON



DELTA



LS



با مدیریت مهندس اکبر اویسی فر

این شرکت با بهره گیری از مهندسين با تجربه در عرصه علم و صنعت به عنوان مجری پروژه های اتوماسیون صنعتی در سطح کشور می باشد. تیم اجرایی این شرکت در سال های اخیر توانسته اند با تضمین راه اندازی ماشین آلات و خطوط تولید، توجه بسیاری از صنایع داخلی و همچنین کارشناسان خارجی را جلب نمایند. تیم اجرایی این واحد قادر است مراحل اجرایی پروژه های اتوماسیون صنعتی را در صنایع مختلف از نصب و راه اندازی تا آموزش، تعمیر و نگهداری و پشتیبانی فنی، بر عهده گیرد.



زمینه های فعالیت شرکت

- طراحی و مونتاژ تابلوهای برق صنعتی و PLC
- تامین کننده تجهیزات برق و اتوماسیون صنعتی با قیمت رقابتی
- طراحی، راه اندازی و عیب یابی شبکه های صنعتی
- طراحی و راه اندازی پانل های صنعتی جهت مانیتورینگ (HMI) پروسه به صورت کاملاً حرفه ای
- پیاده سازی سیستم مانیتورینگ مبتنی بر PC در سایت های کنترل
- نصب و راه اندازی درایوهای صنعتی
- آموزش دوره های تخصصی اتوماسیون صنعتی زیرمنس
- تعمیر تخصصی ماژول های PLC، HMI، DRIVE

از دیگر توانایی های مهندسین اتوماسیون این شرکت ارائه خدمات مربوط به عیب یابی سیستم های کنترلی در محل متقاضی می باشد. امروزه یکی از مشکلات مراکز صنعتی توقف خطوط تولید به دلیل رخ دادن خطا در سیستم کنترلی می باشد. این توقف برای مدت طولانی می تواند منجر به وارد آمدن خسارت های جبران نشدنی به مجموعه شود. تیم اجرایی این شرکت با سوابق درخشان در بحث رفع عیب و راه اندازی تجهیزات کنترلی، در خدمت اتوماسیون صنایع می باشند. با توجه به استقرار این تیم در استان قزوین، تمامی شرکت های موجود در استان قزوین و حومه می توانند از خدمات بخش عیب یابی این شرکت در کوتاه ترین زمان بهره مند شوند.



- ✓ رفع فالت های PLC در کوتاه ترین زمان
- ✓ تعویض و جایگزینی ماژول های جدید در سیستم های کنترلی
- ✓ رفع فالت های شبکه های صنعتی (پروفیباس، اترنت و...)
- ✓ رفع فالت های درایوهای AC و SERVO
- ✓ تغییرات در برنامه PLC و HMI مطابق با خواسته کارفرما
- ✓ از بین بردن پسوردهای نرم افزاری و سخت افزاری PLC
- ✓ گرفتن Backup از برنامه PLC و HMI
- ✓ رفع مشکلات نرم افزاری در سیستم های مانیتورینگ
- ✓ کالیبره کردن تجهیزات ابزار دقیق و شیرهای تدریجی
- ✓ تعمیر تخصصی ماژول های PLC، HMI و DRIVE

مشکلات و توقفات سیستم های کنترلی واحدهای صنعتی خود را با ما در میان بگذارید

واحد فروش

واحد فروش این شرکت نیز فعالیت خود را به طور رسمی با هدف تامین تجهیزات اتوماسیون صنعتی زیرمنس شروع کرده است. واحد فروش این شرکت در یک اقدامی بی سابقه اقدام به تامین تجهیزات اتوماسیون صنعتی زیرمنس با قیمتی رقابتی نموده است. این شرکت آمادگی خود را به منظور تامین تجهیزات اتوماسیون صنعتی زیرمنس، مورد نیاز واحدهای صنعتی استان قزوین و سایر استان ها و همچنین واحدهای آموزشی اعلام می دارد.

• SIMATIC S7-200,S7-300,S7-400,1200

• SIMATIC HMI

• SIMATIC NET

• SIMATIC PCS7

• SIMATIC M7,C7

• MICRO MASTER DRIVE

• SINAMICS DRIVE

• MASTER DRIVE

• SIMOREG DRIVE

• SINUMERIC 840



از دیگر فعالیت های دیگر تیم اجرایی این شرکت تعمیر و نگهداری ماشین آلات صنعتی و بازرسی های دوره ای تجهیزات کنترلی و ابزار دقیق مراکز صنعتی می باشد. از آنجایی که امروزه در اکثر مراکز صنعتی از سیستم های کنترلی هوشمند استفاده می شود و این تجهیزات نیز به دلیل حساسیت بالا در عملکردشان، نیاز به تعمیر و نگهداری و بازرسی منظم توسط گروهی متخصص و با تجربه را دارند. واحد اجرایی این شرکت می تواند به عنوان مجری عملیات تعمیر و نگهداری در واحد های صنعتی مشغول فعالیت شوند.

هدف این تیم از انجام عملیات تعمیر و نگهداری

* ارائه آموزش های مورد نیاز به کادر فنی در زمان روی دادن خطا در سیستم های کنترلی

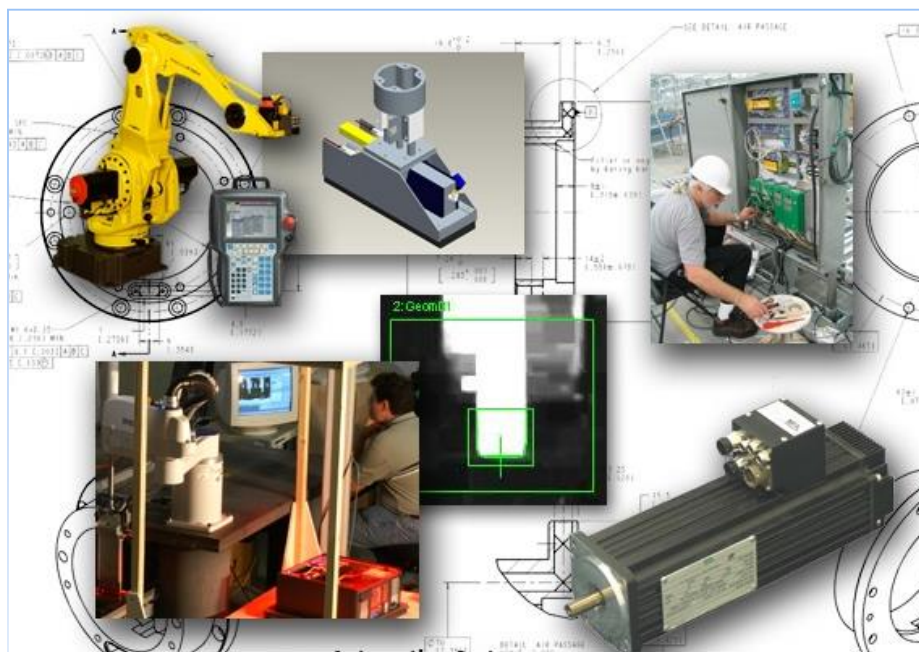


* ساماندهی مدیریت انبار و قطعات مصرفی در خرابی ها

* کاهش خرابی ها و توقفات خطوط تولید

* انجام بازرسی های منظم سیستم های برقی و کنترلی

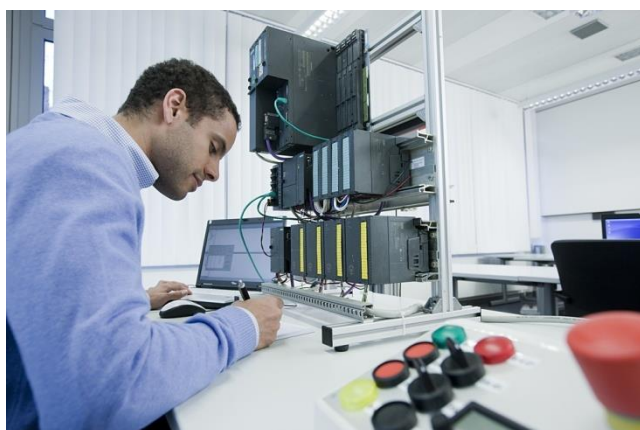
* نظارت پیوسته بر فرآیند تولید و بهبود آن



واحد آموزش

واحد آموزش این شرکت با در اختیار داشتن اساتید با تجربه در عرصه علم و صنعت، اقدام به برگزاری دوره های تخصصی اتوماسیون نموده است. تمامی دوره های این شرکت از لحاظ محتوا و کار عملی مطابق با استانداردهای زیمنس بوده و در هر یک از دوره ها شرکت کننده ها به طور کاملاً تخصصی با سیستم های کنترلی آشنا می شوند. این شرکت افتخار برگزاری دوره های آموزشی برای بسیاری از مهندسين و تکنسینهای شرکت های واقع در استان های مختلف به ویژه استان قزوین را دارا می باشد.

لیست دوره های آموزشی



- LOGO
- PLC- S5
- PLC S7-200
- PLC S7-300,400 Level(I)
- PLC S7-300,400 Level(II)
- PLC S7-1200
- مانیتورینگ با WinCC
- مانیتورینگ با Protocol
- مانیتورینگ با WinCC Flexible
- شبکه های صنعتی Ethernet و Profibus
- کنترل دور موتور با Micromaster
- S7 PID
- PCS7
- SINAMICS Servo Drive
- TIA Software(S7-300,400)
- TIA Software(S7-1500)
- Simotion Scout
- Eplan

این شرکت جهت پیشبرد اهداف آموزشی خود اقدام به چاپ کتاب های تخصصی اتوماسیون و همچنین توزیع سی دی های آموزشی جهت استفاده مهندسین و دانشجویان نموده است. برخی از کتاب های تالیف شده توسط دپارتمان آموزش این شرکت به شرح زیر می باشد.

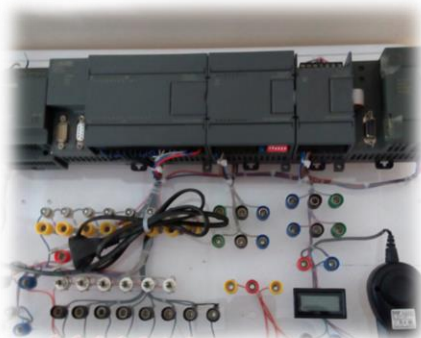


برخی از ست های آموزشی واحد آموزش این شرکت



✓ ست آموزشی PLC S7-300

- ❖ مجهز به پورت PN - DP
- ❖ مجهز به ماژول IM
- ❖ مجهز به ماژول FM
- ❖ مجهز به ماژول AI-AO



✓ ست آموزشی PLC S7-1200

- ❖ مجهز به ماژول شبکه پروفیباس
- ❖ مجهز به ماژول AI-AO
- ❖ مجهز به پانل KTP400

✓ ست آموزشی PLC S7-200

- ❖ مجهز به ماژول شبکه پروفیباس EM277
- ❖ مجهز به ماژول AI-AO
- ❖ مجهز به پانل TD200



✓ ست آموزشی درایو MICROMASTER

- ❖ مجهز به ماژول شبکه پروفیباس
- ❖ مجهز به ماژول انکودر
- ❖ مجهز به ماژول PC جهت اتصال به نرم افزار



✓ ست آموزشی شبکه های صنعتی

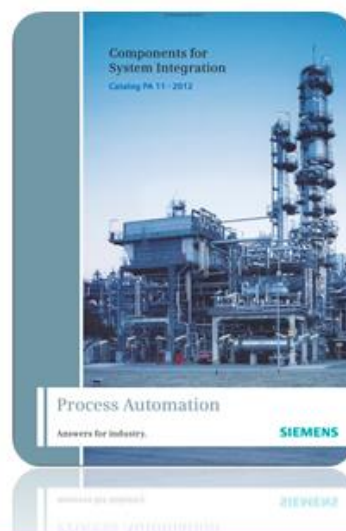
- ❖ مجهز به ماژول Ethernet و Profibus
- ❖ مجهز به ایستگاه ET200S و ET200M
- ❖ مجهز به درایو زیمنس با ماژول Profibus
- ❖ مجهز به انکودر دارای اینترفیس شبکه پروفیباس

✓ ست آموزشی مانیتورینگ

- ❖ پانل MP277
- ❖ کامپیوتر مجهز به CP5611
- ❖ پانل DELTA

فروشگاه اینترنتی نرم افزار، کتاب و جزوات تخصصی اتوماسیون صنعتی

SIEMENS



<http://www.indautomation.sellfile.ir>

برخی از مراکز صنعتی و آموزشی که در سال های اخیر افتخار همکاری با آنها را در زمینه اجرای پروژه، ارائه خدمات فنی و مشاوره و آموزش دوره های تخصصی را داشته ایم به شرح زیر می باشند.



ارتباط با ما

028-32238157



0935-457-7256



Akb_Oveisifar@yahoo.com



اعضای تیم اجرایی

مهندس اکبر اویسی فر (کارشناس ارشد برق-الکترونیک)
مهندس عارف رحمانی (کارشناس ارشد برق-الکترونیک)
مهندس نازنین زنجیریان (کارشناس ارشد برق-کنترل)