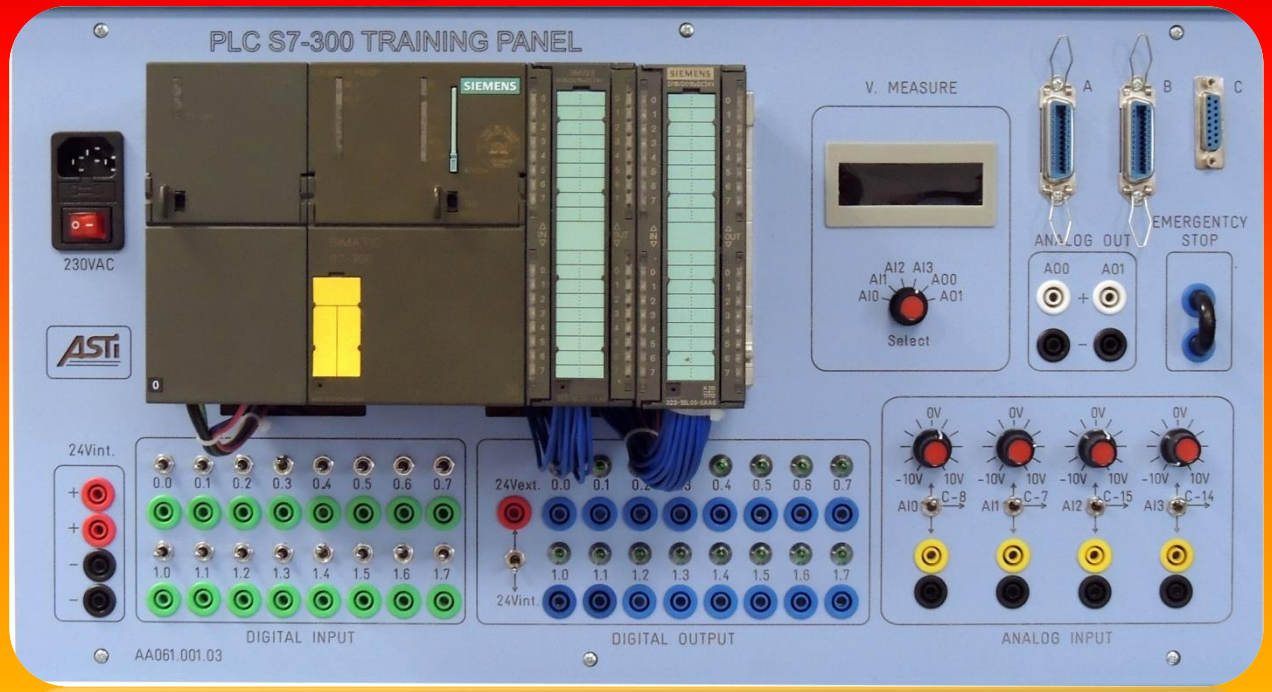
PLC S7 SEIMENS

سطح مقدماتی



تهیه کننده : مهندس عباس محمدی

MohammadiSite.ir

رئوس مطالب

- **فصل اول: آشنایی با اتوماسیون صنعتی و تاریخچه PLC**

فصل دوم: ارکان و اجزای PLC ها

فصل سوم: مدار منطقی و منطق دیجیتال

فصل چهارم: برنامه نویسی PLC S7 با نرم افزار SIMATIC STEP 7

فصل اول

آشنایی با اتوماسیون صنعتی و تاریخچه PLC

اتوماسیون صنعتی چیست؟

اتوماسیون صنعتی به بهره گیری از سایر تجهیزات نرم افزاری وسخت افزاری به جای متصدیان انسانی برای کنترل دستگاه ها و فرایندهای صنعتی گفته می شود.

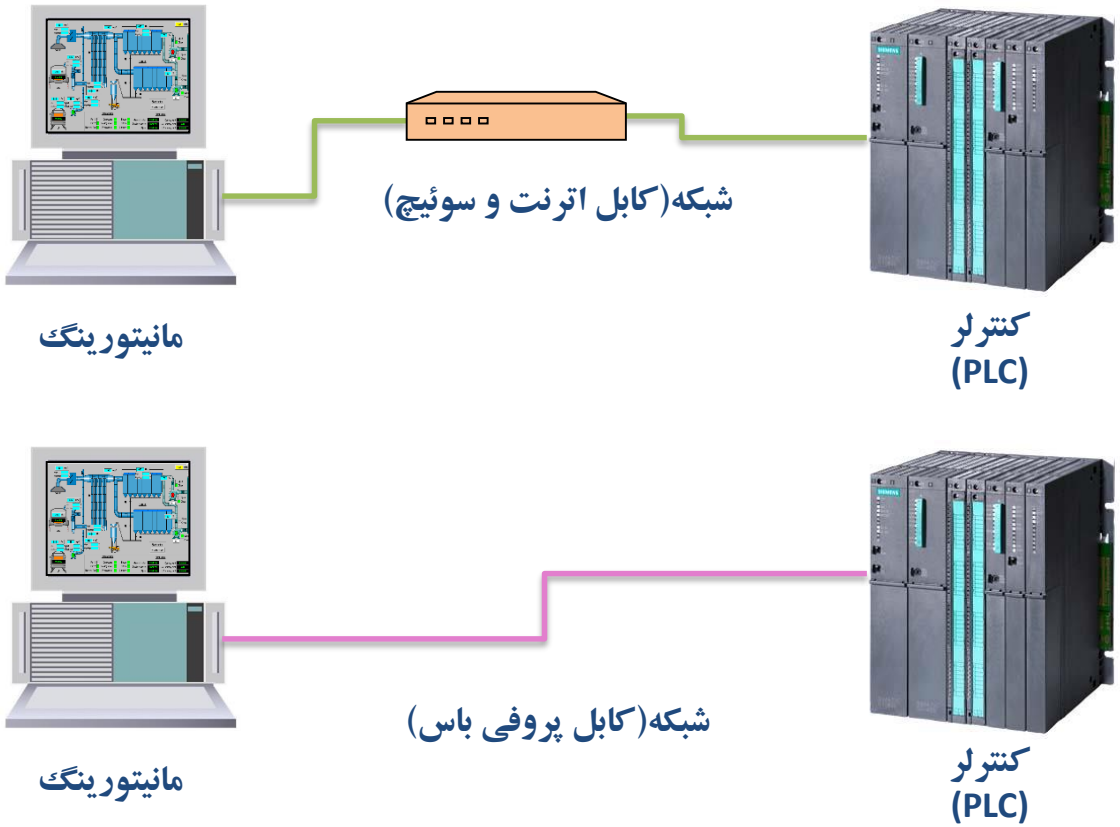
اتوماسیون صنعتی = قلب و مغز یک کارخانه

مزایای بکارگیری اتوماسیون صنعتی

- تکرارپذیری فعالیتها و فرایندها
- افزایش کیفیت محصولات تولیدی
- افزایش سرعت تولید (کمیت تولید)
- کنترل کیفیت دقیقتر و سریعتر
- کاهش پسماندهای تولید (ضایعات)
- برهم کنش بهتر با سیستمهای بازرگانی
- افزایش بهره وری واحدهای صنعتی
- بالا بردن ضریب ایمنی برای نیروی انسانی و کاستن از فشارهای روحی و جسمی

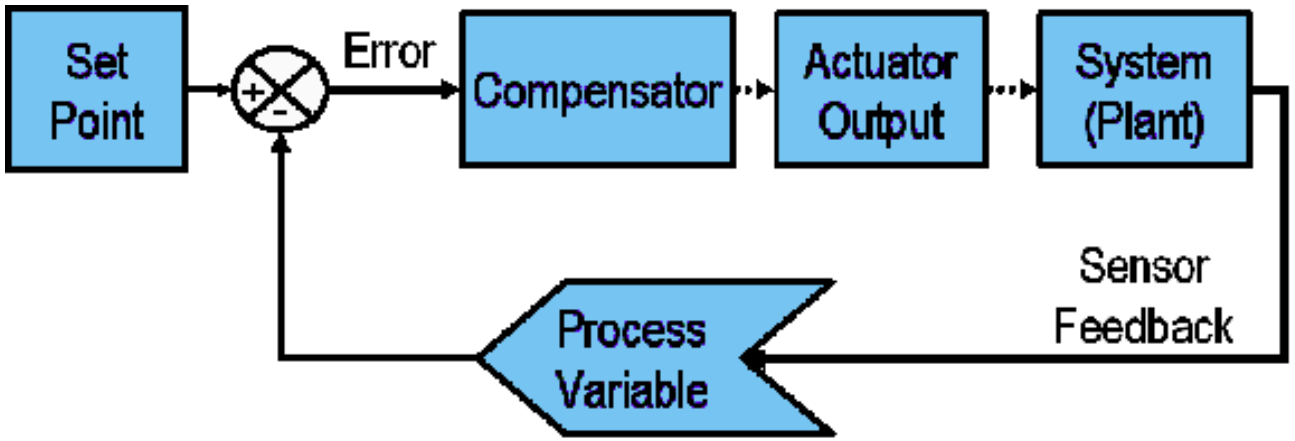
ارکان مختلف اتوماسیون صنعتی

- کنترلر
- مانیتورینگ (HMI)
- شبکه های صنعتی



کنترلر

کنترلر مغز یک فرایند صنعتی است و تمامی فرامینی را که یک متخصص اعمال می کند، تا پروسه ، جریان استاندارد خود را پیش بگیرد و نهایتاً پاسخ مطلوب حاصل شود از طریق کنترلر به سیستم فهمانده می شود.



پاسخ مطلوب $|SP-PV| \approx 0$

هدف از کنترل

بخشی از وظایف انسان در صنعت به تجهیزات خودکار واگذار میگردد.

سیستمهای کنترل

رله کنتاکتوری
Relay

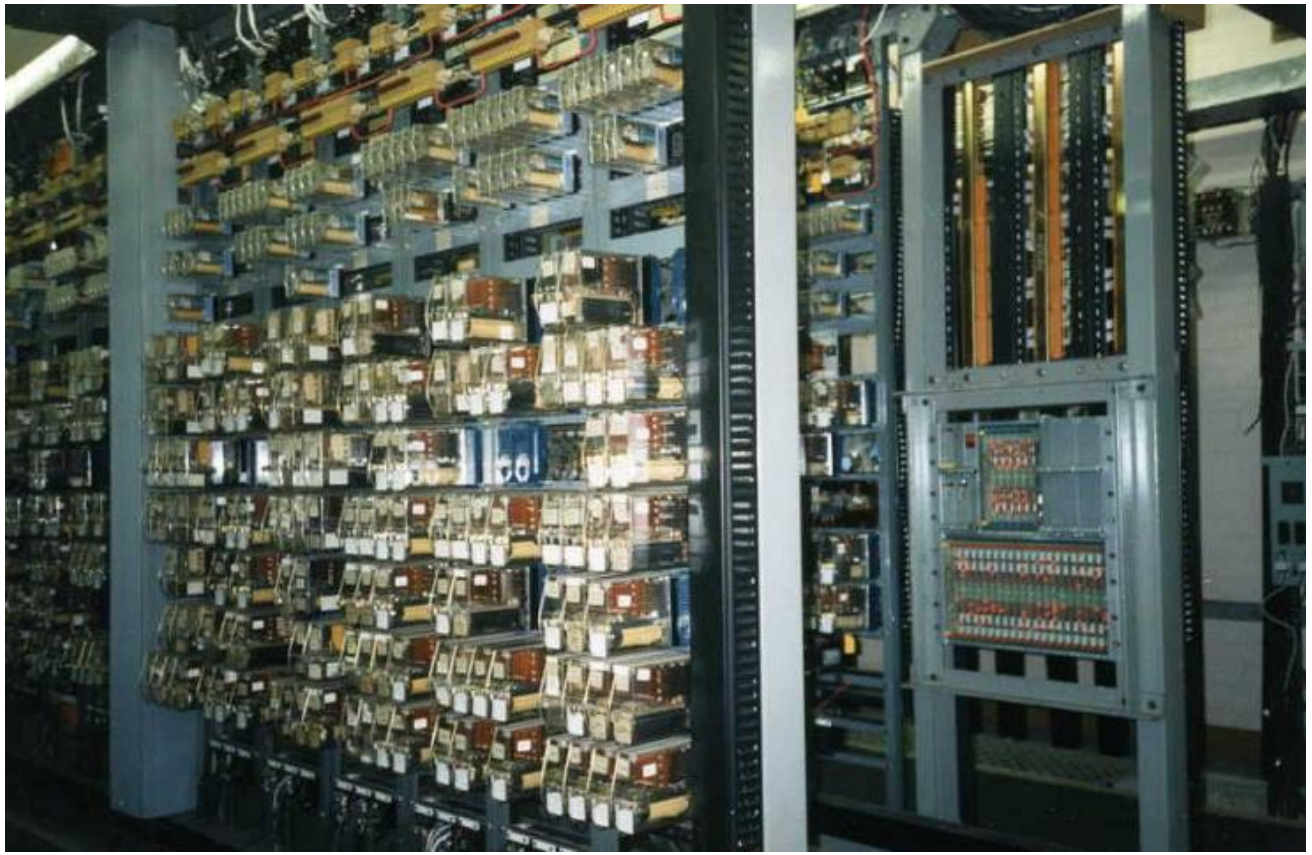
کنترل مبتنی بر
فشار هوا
(پنوماتیک)

کنترل مبتنی بر
کامپیوتر
(PC Base)

کنترل مبتنی بر کنترلرهای
منطقی برنامه پذیر
PLC

کنترلر

سیستم های کنترلر – رله کنتاکتوری

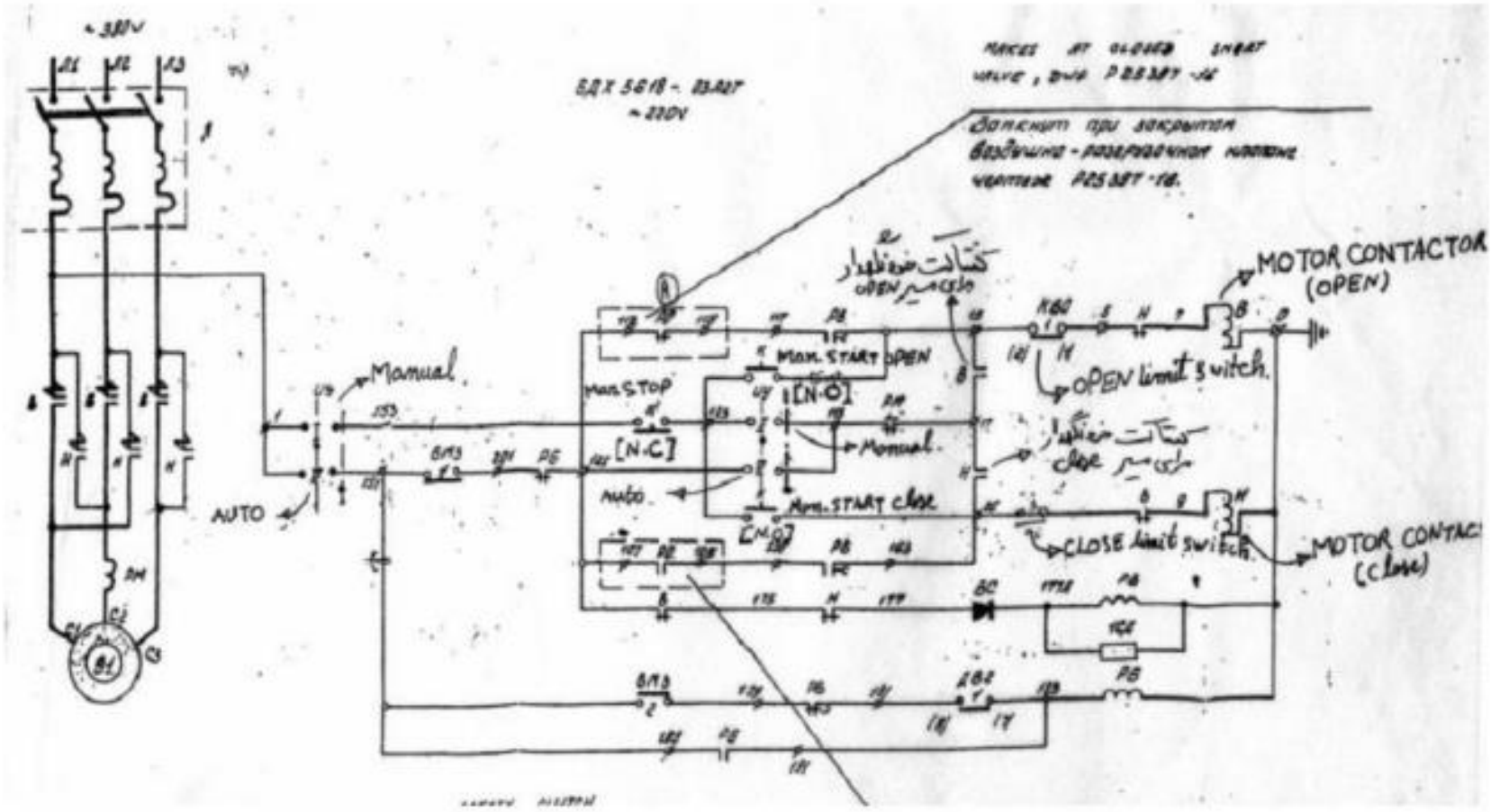


کنترل– رله کنتاکتوری

تا قبل از به کار گیری تکنولوژی **PLC** ها سیستم های کنترل با استفاده از ترکیبات رله و کنتاکتوری پیاده سازی می شدند (مدل **Hard-Wired**) .

در این مدل ابتدا می بایست نقشه های کلیه مدارات کنترلی طراحی شود.

نمونه نقشه از سیستم رله کنتاکتوری



تهیه کننده : مهندس عباس محمدی

کنترلر

سیستم های کنترلر پنوماتیکی

مشکلات:

- حجم زیاد عناصر فیزیکی سیستم
- عیب یابی
- تغییر و توسعه

پنوماتیکی(نیوماتیکی)+کنترل آنالوگ

کنترلر

سیستم های کنترلر مبتنی بر کامپیوتر

- دستگاه های CNC
- سیستم های کنترل قوس در کوره های قوس الکتریک

تهیه کننده : مهندس عباس محمدی

کنترلر

سیستم های کنترلر جدید

سیستم های کنترلی جدید مبتنی بر PLC و DCS هستند

PLC = Programmable Logic Controller
کنترل کننده منطقی برنامه پذیر

DCS = Distributed Control System
سیستم کنترل توزیع شده(غیر متمرکز)

پرکاربرد ترین PLC در صنعت داخلی Siemens می باشد.
پرکاربرد ترین DCS در صنعت داخلی Yokogawa می باشد.

کنترلر

سیستم های کنترلر جدید

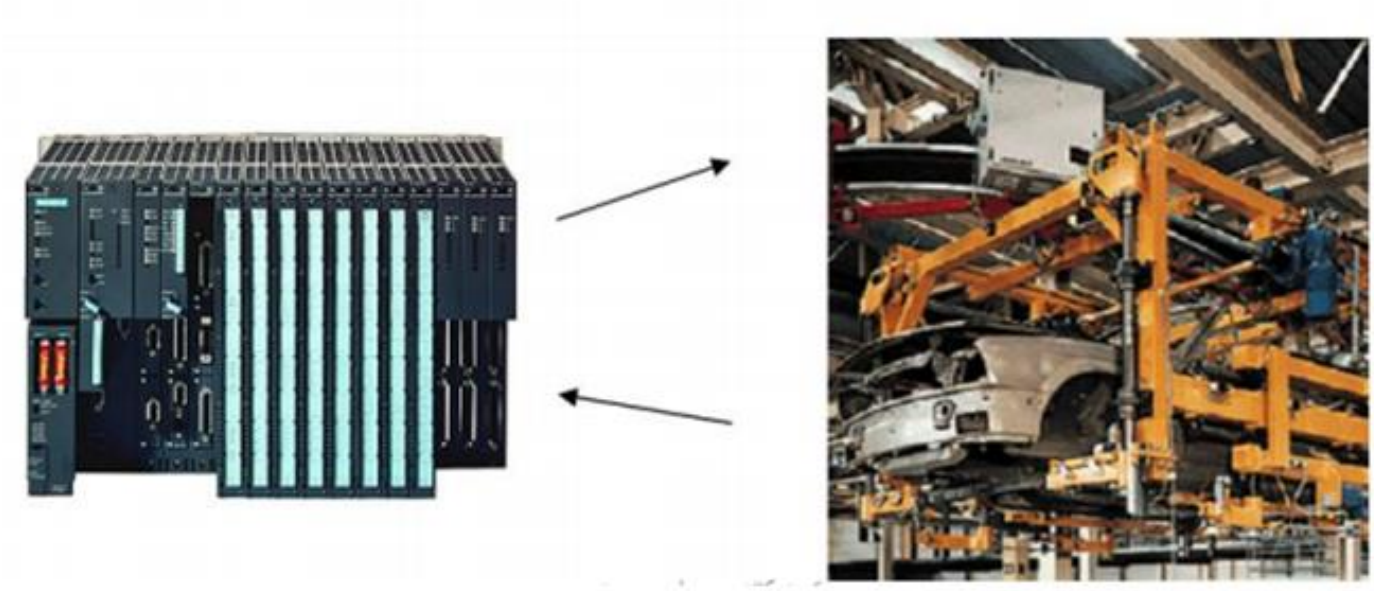
PLC- Siemens

DCS- Yokogawa

PLC

تعریف PLC

PLC ها یک نوع کنترل کننده منطقی (Logical) از خانواده کامپیوترها هستند که برای کاربردهای صنعتی طراحی و ساخته شده اند. از PLC ها برای انجام خودکار عملیات در کارخا نجات تولیدی استفاده می شود .



مزایای PLC نسبت به سیستم های کنترلر قدیمی

- هزینه نصب و راه اندازی آنها پایین می باشد.
- برای نصب و راه اندازی آنها زمان کمتری لازم است .
- تعمیر و نگه داری آنها بسیار ساده می باشد.
- به سادگی قابلیت گسترش دارند .
- قابلیت انجام عملیات پیچیده را دارند.
- ضریب اطمینان بالایی در اجرای فرایندهای کنترلی دارند .
- ساختار مدولار دارند که تعویض بخشهای مختلف آن را ساده میکند.
- اتصالات ورودی - خروجی و سطوح سیگنال استاندارد دارند.
- زبان برنامه نویسی آنها ساده و سطح بالاست.
- در مقابل نویز و اختلالات محیطی حفاظت شده اند.
- تغییر برنامه در هنگام کار آسان است.
- امکان ایجاد شبکه بین چندین **PLC** به سادگی میسر است .
- امکان کنترل از راه دور (به عنوان مثال از طریق خط تلفن یا سایر شبکه های ارتباطی) قابل حصول است .
- امکان اتصال بسیاری از تجهیزات جانبی استاندارد از قبیل چاپگر ، بارکد خوان و ... به **PLC** ها وجود دارد

در ابتدا ترغیب کردن صنتگران به استفاده از **PLC** کار چندان ساده ای نبود چون به راحتی قانع نمی شدند که یک مجموعه کوچک از قطعات الکترونیکی به همراه چند خط برنامه بتواند وظایف ۴۰ – ۵۰ تابلوی متشکل از مدارات رله – کنتاکتوری را انجام دهد . اما استفاده از **PLC** با توجه به مزایایی که داشت به تدریج رایج شد و سازندگان متعددی نیز در این رشته پدیدار شدند . با پیشرفت علم الکترونیک **PLC** ها نیز از قابلیت های بهتر و بیشتری برخوردار شدند و در صنایع مختلف به کار گرفته شدند. هم اکنون بیش از میلیونها **PLC** در سراسر دنیا در حال کار هستند و روز به روز نیز به تعداد آنها افزوده می شود.

تاریخچه PLC

PLC ها تاریخچه کوتاهی دارند و از تولد اولین آنها عمر چندانی نمی گذرد . اولین **PLC** ها در دهه ۷۰ برای استفاده در صنایع اتوموبیل سازی طراحی شدند. نخستین بار کنترلرهای برنامه پذیر توسط شرکت **Modicon** در سال ۱۹۶۸ به در صنعت معرفی شدند که با هدف جایگزینی رله های مکانیکی از آنها استفاده می شد.

اولین PLC، مدل MODICON 084 در سال ۱۹۶۹ اختراع شد.

اولین PLC موفق به طور عملی، مدل Modicon 184 در سال ۱۹۷۳ به صنعت عرضه شد.

سیر تحول PLC ها

سال میلادی	تحول
۱۹۶۸	مفاهیم اولیه توسعه یافت
۱۹۶۹	ارائه اولین سخت افزار با ۱ کیلوبایت حافظه و ۱۲۸ ورودی و خروجی و دستورات پردازش بیت
۱۹۷۱	اولین کاربرد صنعتی PLC
۱۹۷۲	اضافه شدن دستورات کانتر و تایمر
۱۹۷۳	اضافه شدن دستورات محاسباتی و جابه جایی دیتا و برقراری ارتباط بین PLC و PC
۱۹۷۴	توسعه حافظه به ۱۲ کیلو بایت و تعداد ورودی و خروجی به ۱۰۲۴
۱۹۷۵	قابلیت کنترل PID
۱۹۷۸	برقراری ارتباط بین دو یا چند سیستم
۱۹۸۳	توسعه حافظه به 4 MB و تعداد ورودی و خروجی به 8192
۱۹۹۳	استاندارد سازی زبانهای برنامه نویسی

استاندارد PLC ها

۱- در سال ۱۹۷۹ یک گروه متخصص از سازمان IEC جهت استاندارد سازی PLC ها شروع به فعالیت نمودند.

۲- پس از حدود ۱۲ سال استاندارد IEC1131 برای PLC ها ارائه گردید.

۳- در سال ۱۹۹۰ بخش ۱و۲ و در سال ۱۹۹۳ بخش ۳ و در سال ۱۹۹۵ بخش ۴ این استاندارد تدوین شد.

۴- مهمترین بخش این استاندارد بخش سوم یعنی IEC1131-3 است که ۵ روش برنامه نویسی استاندارد شده و سازندگان PLC ها موظفند که همه یا بخشی از آنها را پشتیبانی کنند.

سازندگان مطرح PLC

در حال حاضر شرکت های زیادی در اکثر کشورهای توسعه یافته تولید کننده **PLC** و قطعات مربوطه هستند . در اینجا به نام چند سازنده که از اعتبار و معروفیت جهانی و معروفیت جهانی برخوردارند اشاره می شود:

- Siemens
- Omron
- Modicon
- GE Fanuc
- Allen-Bradley

شرکت های مطرح سازنده PLC

سازندگان مطرح PLC

شرکت آمریکایی Allen-Bradely سازنده PLC های سری Control Logix

Rockwell Automation

Allen-Bradley

سازندگان مطرح PLC

شرکت ژاپنی OMRON سازنده PLC های سری SYSMAC

OMRON

سازندگان مطرح PLC

شرکت آمریکایی Modicon سازنده PLC های سری Quantum

MODICON
Schneider Electric

سازندگان مطرح PLC

شرکت آمریکایی GE Fanuc سازنده PLC های سری GE Series 90

سازندگان مطرح PLC

شرکت آلمانی Siemens سازنده PLC های سری S5 و S7

SIEMENS

در این دوره با PLC SIEMENS آشنا خواهید شد

انواع PLC های زیمنس از نظر سخت افزاری

Compact PLC (یکپارچه)

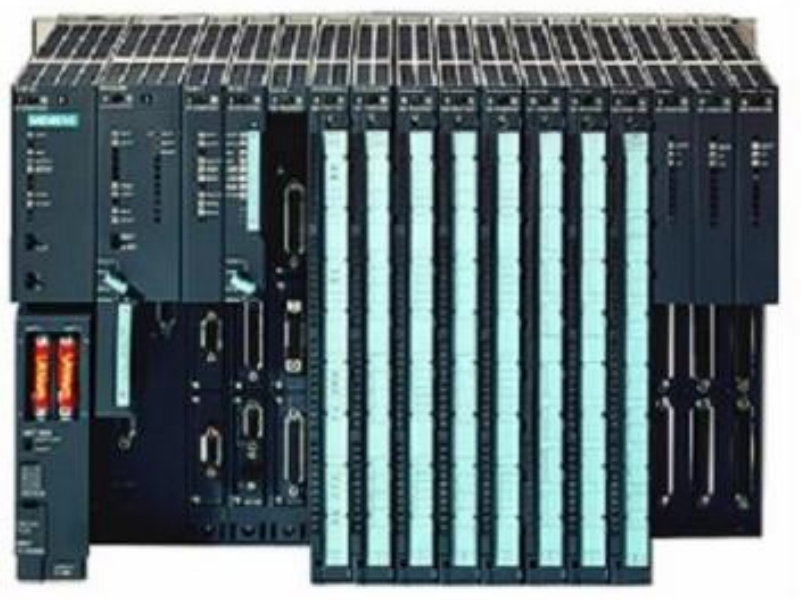
Modular PLC (جدا ازهم)

PLC یکپارچه

PLC یکپارچه خود کفاست . یعنی سخت افزار آن ، منبع تغذیه ، **CPU** و تعداد محدودی ورودی وخروجی را بصورت یک بسته یکپارچه شامل می شود. این نوع **PLC** مختصرتر ، ساده تر و ارزانتر و دارای عملکردی محدودتر از گونه دیگر می باشد و از آن برای کنترل کردن نقاله های کوچک ، ماشین های تراش، پرس های ضربه ایی، سیستم های کنترل هیدرولیکی و بادی ، .... می توان استفاده نمود.

PLC ماژولار

PLC ماژولار از کنـار هم قرار گرفتن ماژولهای مختـلف ، مانند : ماژول منبع تغذیه، ماژول CPU، ماژولهای ورودی، ماژولهای خروجی، کارت های شبکه ساخته می شوند.

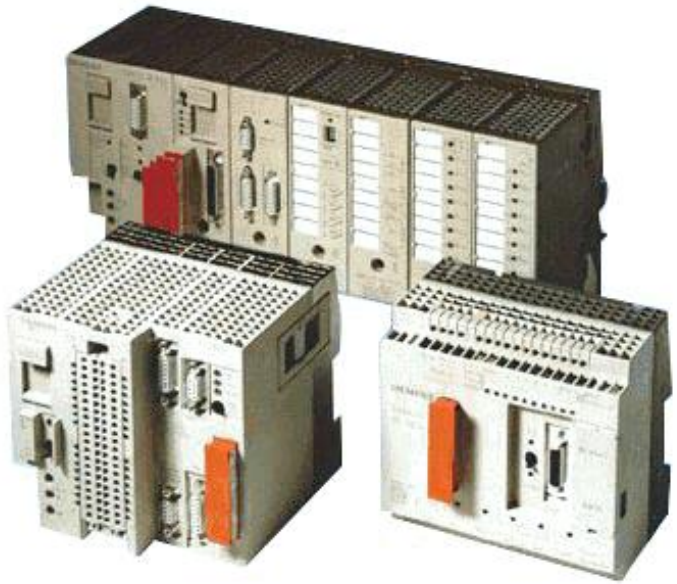


به علت اینکه در طراحی سیستم کنترل بر اساس این PLC می توان انواع مختلف و تعداد متفاوتی از ماژولها را کنار هم قرار داد ، این سیستم قابلیت انعطاف بیشتری دارد و می توان آن را برای کاربرد های خاص ، مانند سیستم ها کنترل خود کار ماشین ها و کنترل فرایند ، طراحی نمود .

همچنین **PLC** های ماژولار تعداد ورودی ها و خروجی های بیشتری دارند و قابل توسعه دادن **(Expansion)** هستند. در نتیجه می توان آنها را به راحتی برای کار در سیستم های بزرگتر ، با اعمال تغییراتی، استفاده نمود.

انواع مختلف PLC های تولید شده در Siemens

1.Simatic S5



1.S5-90U
2.S5-95U
} بصورت **Compact** بوده و حوزه عملکرد محدود دارند.

3.S5-100U
4.S5-115U
} بصورت مدولار بوده و حوزه عملکرد متوسط دارند.

5.S5-135U
6.S5-155U
} بصورت مدولار بوده و حوزه عملکرد وسیع دارند.

Required Software: STEP 5

2.LOGO! Logic Modules

بصورت **Compact** بوده و حوزه عملکرد محدود دارند و برنامه ریزی آن توسط کلیدهای روی آن انجام می شود.

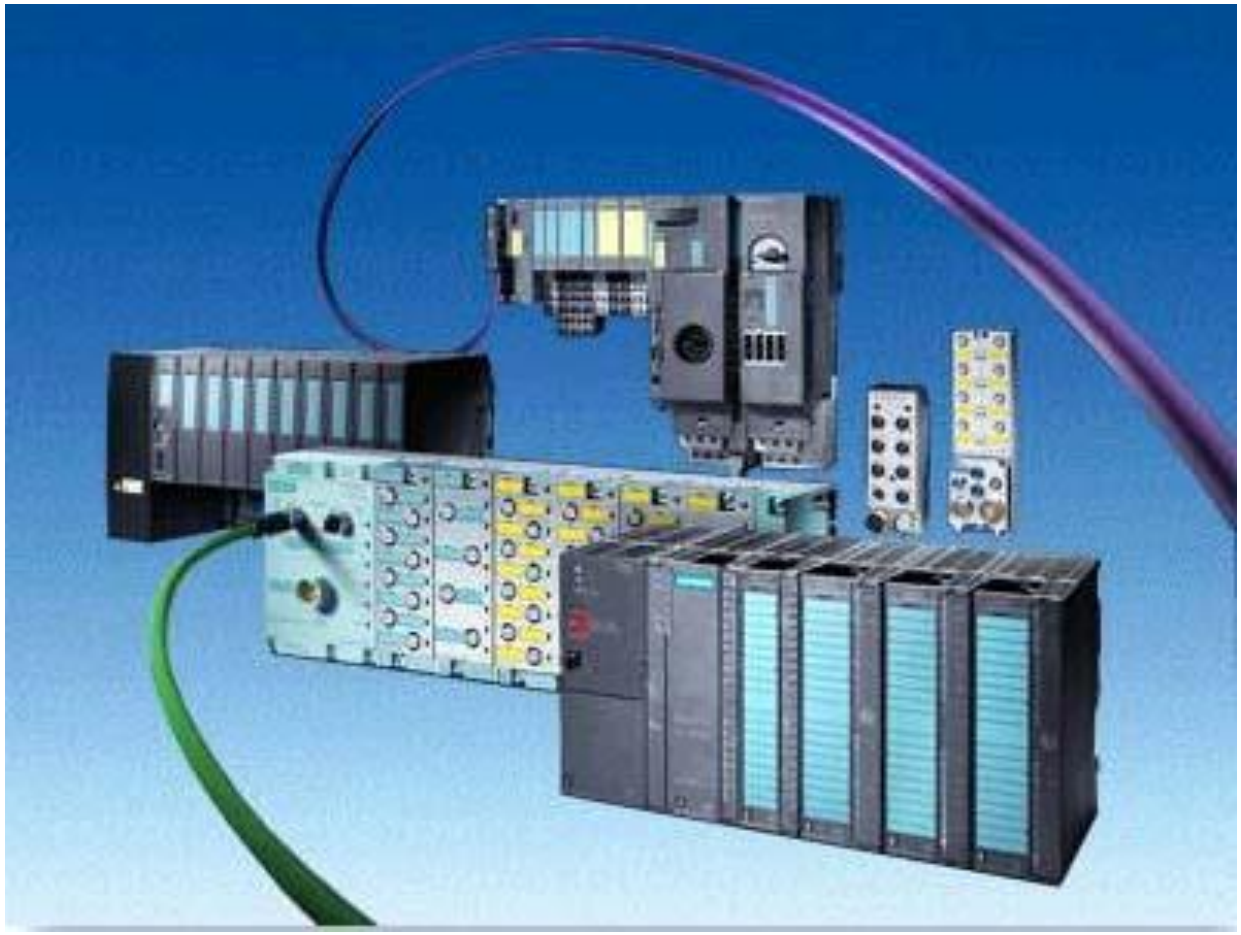
Required Software: LOGO! Soft Comfort

3.Simatic 505

بصورت **Compact** بوده و برای کاربرد در حوزه های کوچک و متوسط طراحی شده اند.

Required Software: TISOFT

4.Simatic Step7



S7-200

- یک **Micro PLC** ارزان قیمت است.
- برای مقاصد ساده تا نسبتا پیچیده کنترلی بکار می رود.
- برنامه نویسی و کار با آن آسان است.
- انواع مختلف دارد و در برخی از انواع آن می توان مدول اضافی نیز در کنار **CPU** قرار داد.
- برنامه نویسی آن با نرم افزار **STEP7-Micro/Win** انجام می شود.

S7-300

- یک **Mini PLC** است.
- حوزه عملکرد آن متوسط است.
- مدولار است.
- مدولهای آن تنوع زیادی دارد.
- بسهولت قابل توسعه است.
- برنامه نویسی آن با نرم افزار **STEP7** انجام می شود.

S7-300F

- پایه آن **S7-300** است.
- در انتهای کد **CPU** حرف **F** معرف این نوع است مانند: **CPU 315F**.

- این **PLC** ها برای سیستم های **Fail Safe** بکار می روند، یعنی در مواردی که فرآیند خطرناک و در صورت بروز خطا امکان انفجار یا آتش سوزی وجود داشته باشد استفاده می شود.
- **Fail Safe** سیستمی است که در شرایط بروز خطا، فرآیند را به سمت شرایط ایمن هدایت می کند.
- حداکثر ایمنی برای کنترل پروسه های خطرناک را توسط پردازش همزمان دو برنامه **Standard**و **Safety**فراهم می کنند. برای نوشتن برنامه **Safety**نیاز به نرم افزار **F-System**می باشد
- *به عنوان مثال : در سیستم کنترل **Fail Safe**می توان تعیین کرد که اگر کنترلر دچار مشکل شده ، فرمان هایی که به عملگرها ارسال شده در چه حالتی قرار گیرند.*
- برای پیکربندی این سیستم علاوه بر **Step7** نرم افزار **F-System** نیز موردنیاز است.

S7-300C

- شبیه **S7-300** است با این تفاوت که **CPU** همراه با مدول دیگری مانند ورودی و خروج بصورت **Compact** عرضه شده است.

- در انتهای کد **CPU** حرف **C** معرف این نوع است مانند: **CPU 314C**.

S7-400

- حوزه عملکرد آن وسیع است.
- مدولار است.
- حجم زیادی از سیگنالها را می تواند پوشش دهد.
- بسهولت قابل توسعه است.
- در مقایسه با **S7-300** سرعت پردازش بالاتر، حافظه بیشتر و امکانات وسیعتری را داراست.
- برنامه نویسی آن با نرم افزار **STEP7** انجام می شود.

S7-400H

- پایه آن همان S7-400 است.
- در سیستم هایی که دسترسی پذیر بالایی موردنیاز باشد بکار می رود، یعنی پروسه ای که اگر متوقف شود منجر به خسارت زیاد می شود
- جایی که هزینه راه اندازی مجدد سیستم پس از رفع عیب بالاست.
- به این سیستم ها Redundant نیز گفته می شود و دارای دو CPU مشابه است که یکی به عنوان Master و دیگری به عنوان StandBy کار میکنند.
- در صورت بروز خطا در سیستم Master سیستم Standby به صورت خودکار درمدار می آید.
- برای پیکربندی این سیستم علاوه بر Step7 نرم افزار H-System نیز موردنیاز است.

S7-400FH

- پایه آن همان **S7-400** است.
- توانایی های **S7-400H** را داراست.
- توانایی های **F-system** را نیز دارد یعنی برای کاربردهایی که درجه ایمنی بالا نیاز دارند نیز مناسب است.

5.Simatic C7

C7 ترکیبی از S7-300 و Operator Panel است که علاوه بر اینکه کار کنترلی انجام میدهد برروی نمایشگر آن می توان پیغامها، رخدادها و مقادیر مربوط به فرایند را دید و فانکشن هایی را نیز توسط صفحه کلید روی آن اعمال نمود. C7 بصورت Compact بوده و انواع مختلف دارد که توانایی آنها باهم متفاوت است.

Required Software: Step7+Protool

انواع PLC های زیمنس

S5

TI 505

Quadlog

LOGO

S7-200/300/400/400H/400F/400FH

C7

S7-1200

S7-1500

انواع PLC های خانواده S7-300

Compact

Normal

Profinet Interface

Fail Safe

انواع PLC های خانواده S7-400

S7-400

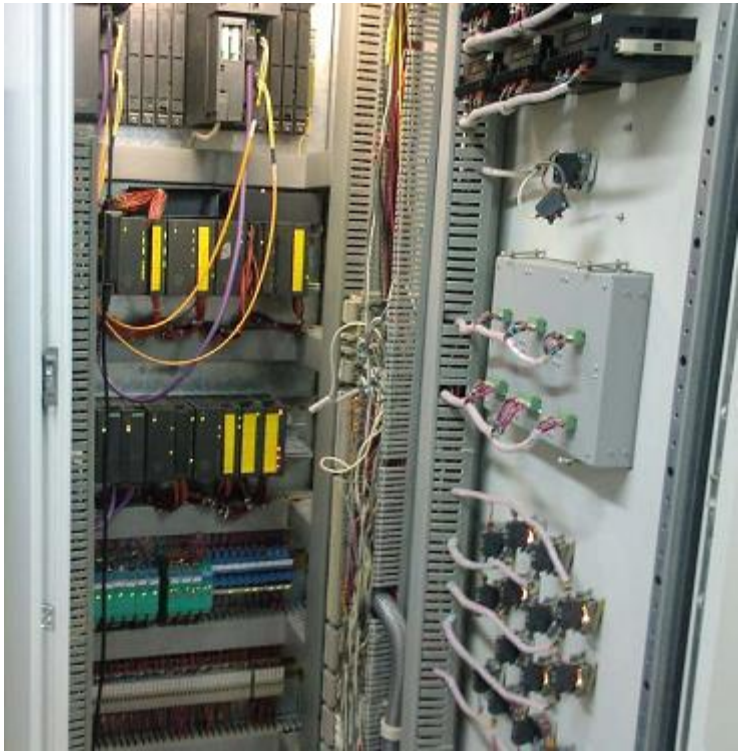
S7-400H

S7-400F

S7-400FH

پرکاربردترین PLC های زیمنس در صنایع داخلی کشور

خانواده S7-300
فرایندهای نسبتاً کوچک و متوسط



خانواده S7-400
فرایندهای بزرگ

نرم افزار STEP 7 جهت برنامه ریزی PLC های سری 7 زیمنس می باشد که دارای انواع کلی زیر است:

PLC S7-200

STEP 7-Microwin

PLC S7-300/400/C7

STEP 7

PLC S7-1200/1500

TIA PORTAL

فصل دوم

ارکان و اجزای PLC

اجزای اصلی PLC

POWER SUPPLY

INPUT MODULE

PROCESSOR

OUTPUT MODULE

INPUT

OUTPUT

PUSH BUTTON

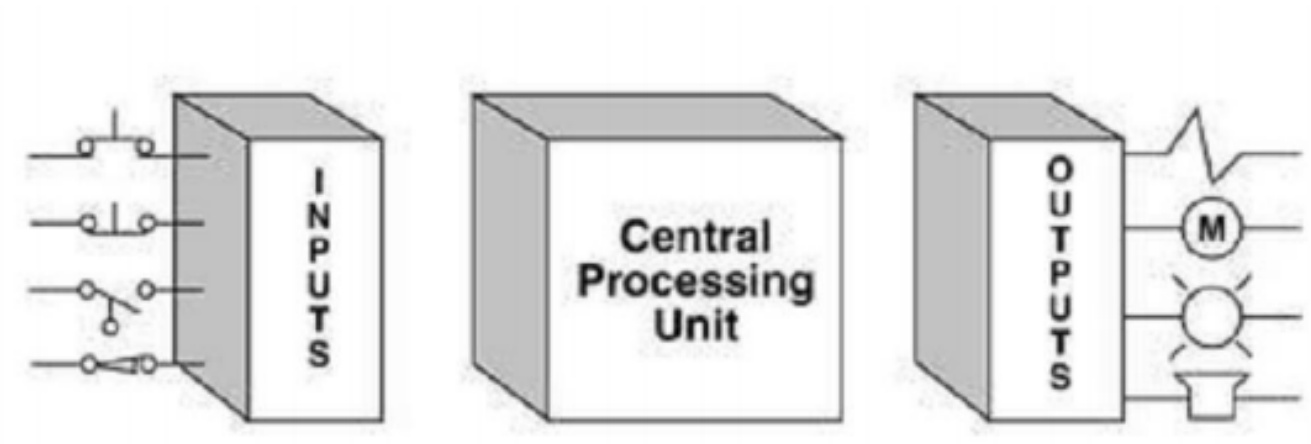
PG= PROGRAM DEVICE

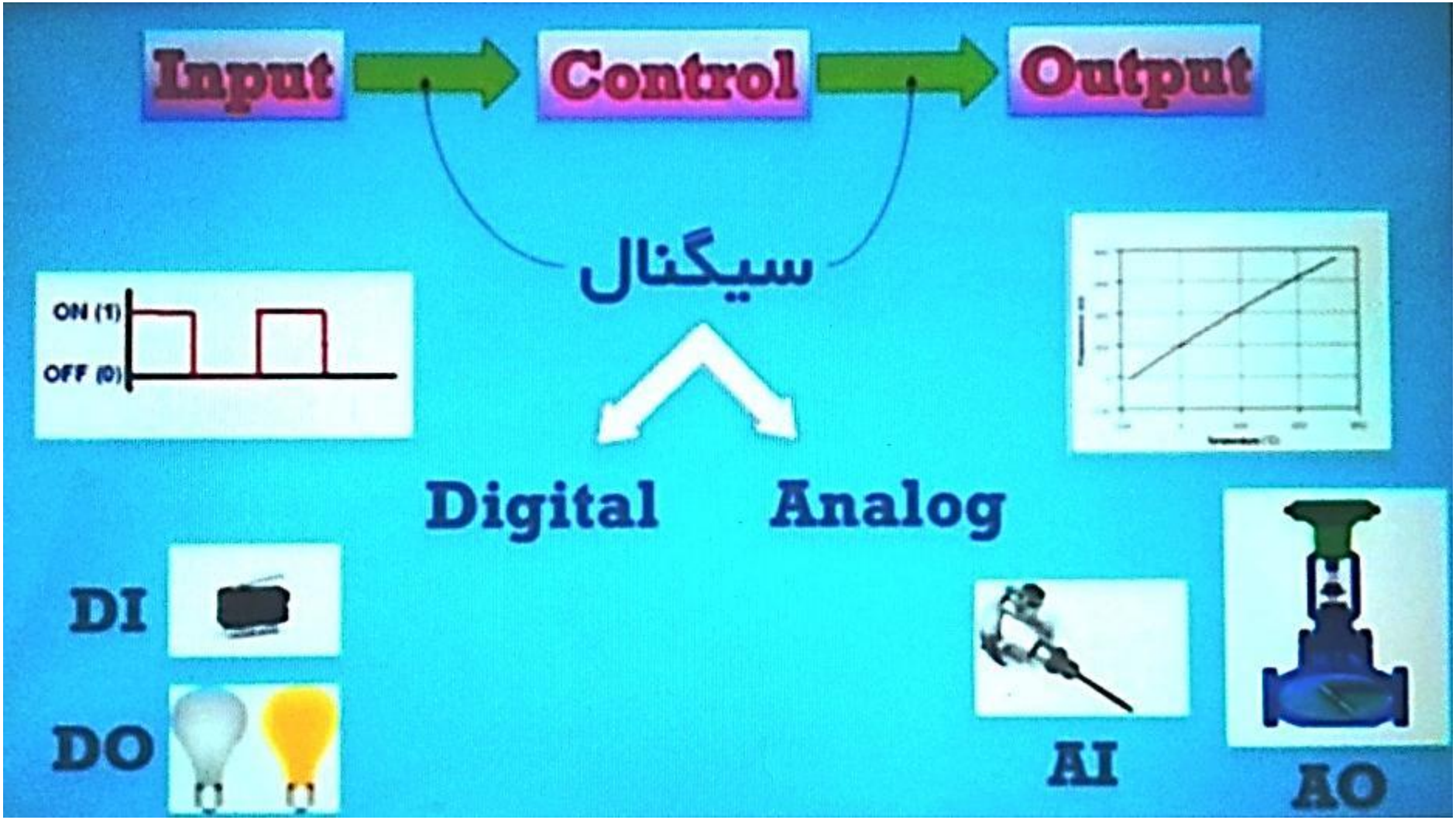
LAMP

تهیه کننده : مهندس عباس محمدی

وظیفه اصلی PLC

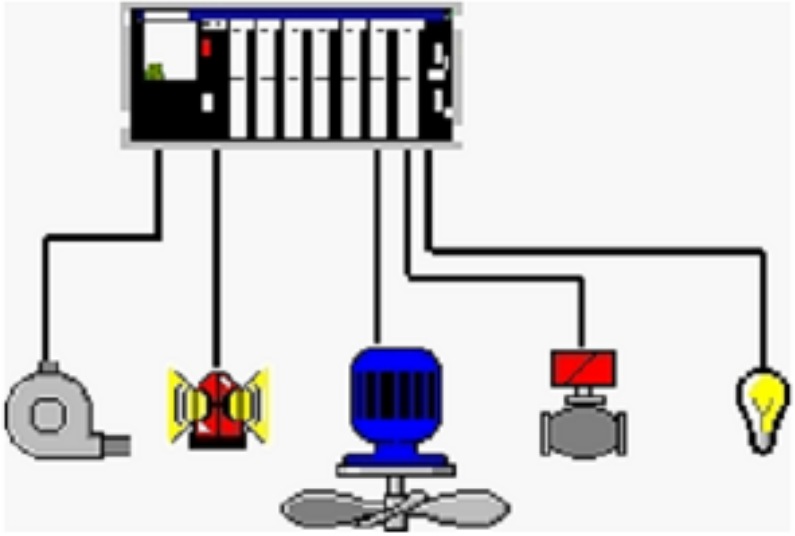
کار عمده و اصلی یک PLC ، گرفتن اطلاعات از واحد تحت کنترل به عنوان ورودی سیستم ، تصمیم گیری با توجه به مقادیر ورودی ها و برنامه ایی که در آن تعبیه شده و درنهایت ایجاد خروجی ها و ارسال آنها به سخت افزارهای میانی جهت هدایت ماشینهای تحت کنترل می باشد.




Input
Control
Output
سیگنال
ON (1)
OFF (0)
Digital
Analog
DI
DO
AI
AO

برنامه های PLC

برنامه درون PLC ، مجموعه ایی از دستور العمل هایی است که کاربر آنها را متناسب با نحوه عملکرد مکانیسم و فرایند موجود ایجاد کرده و در درون حافظه PLC قرار می دهد. وقتی برنامه اجرا می شود، PLC سیستم را بر اساس مشخصات فرایند مورد نظر، راه می برد.



سخت افزار PLC

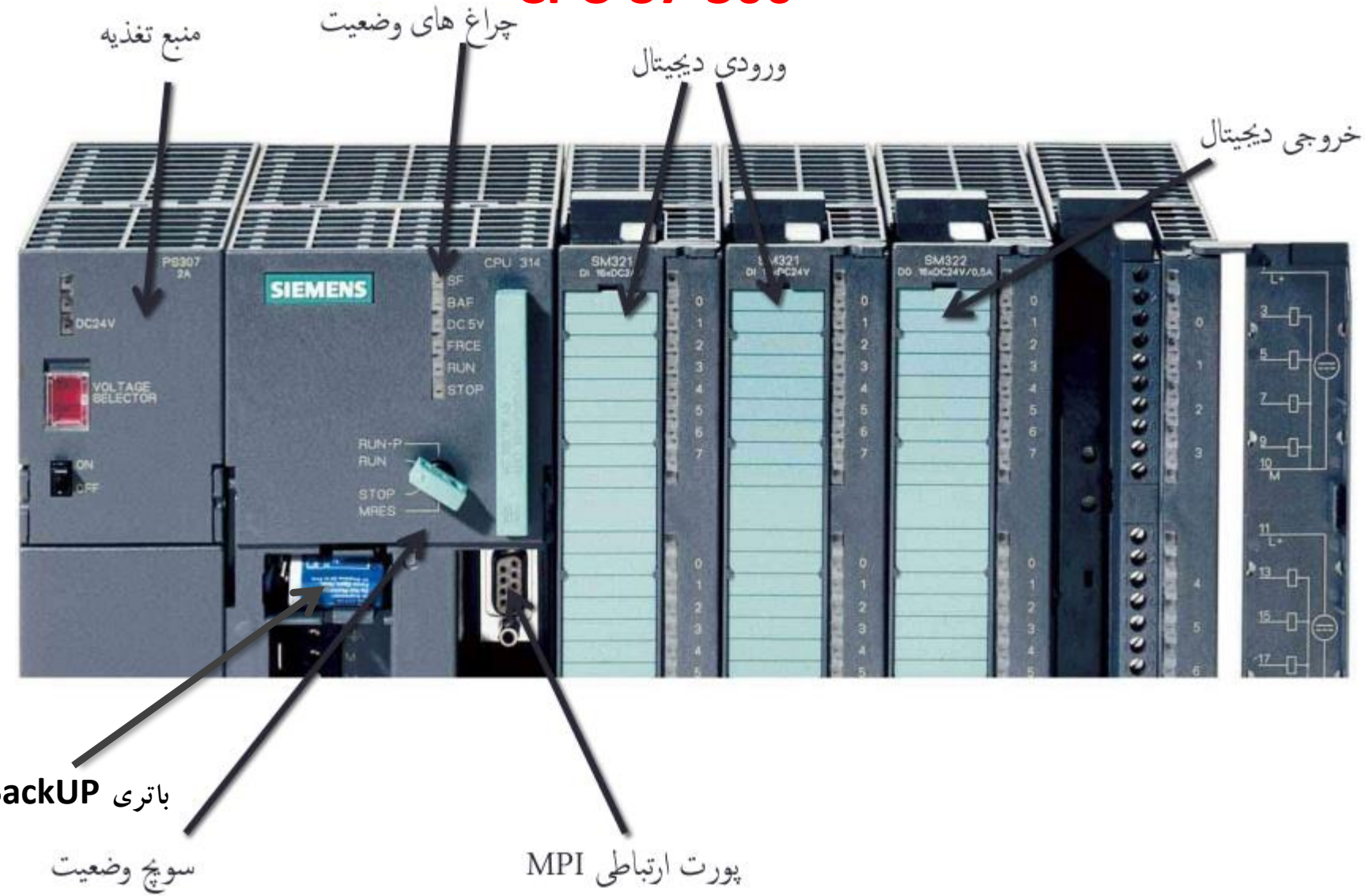
از لحاظ سخت افزاری می توان قسمت های تشکیل دهنده ی یک سیستم Simatic s7-PLC 300/400 را به صورت زیر تقسیم نمود :

۱. واحد منبع تغذیه PS (Supply Power)
۲. واحد پردازش مرکزی CPU (Unit Processing Central)
۳. قفسه ها Racks
٤. ماژول های واسط IM (Interface Module)
٥. مازول های سیگنال SM (Signal Module)
٦. ماژول های تابع FM (Function Module)
۷. پردازنده های ارتباطی CP (Communication processor)
۸. زیر شبکه ها Subnet

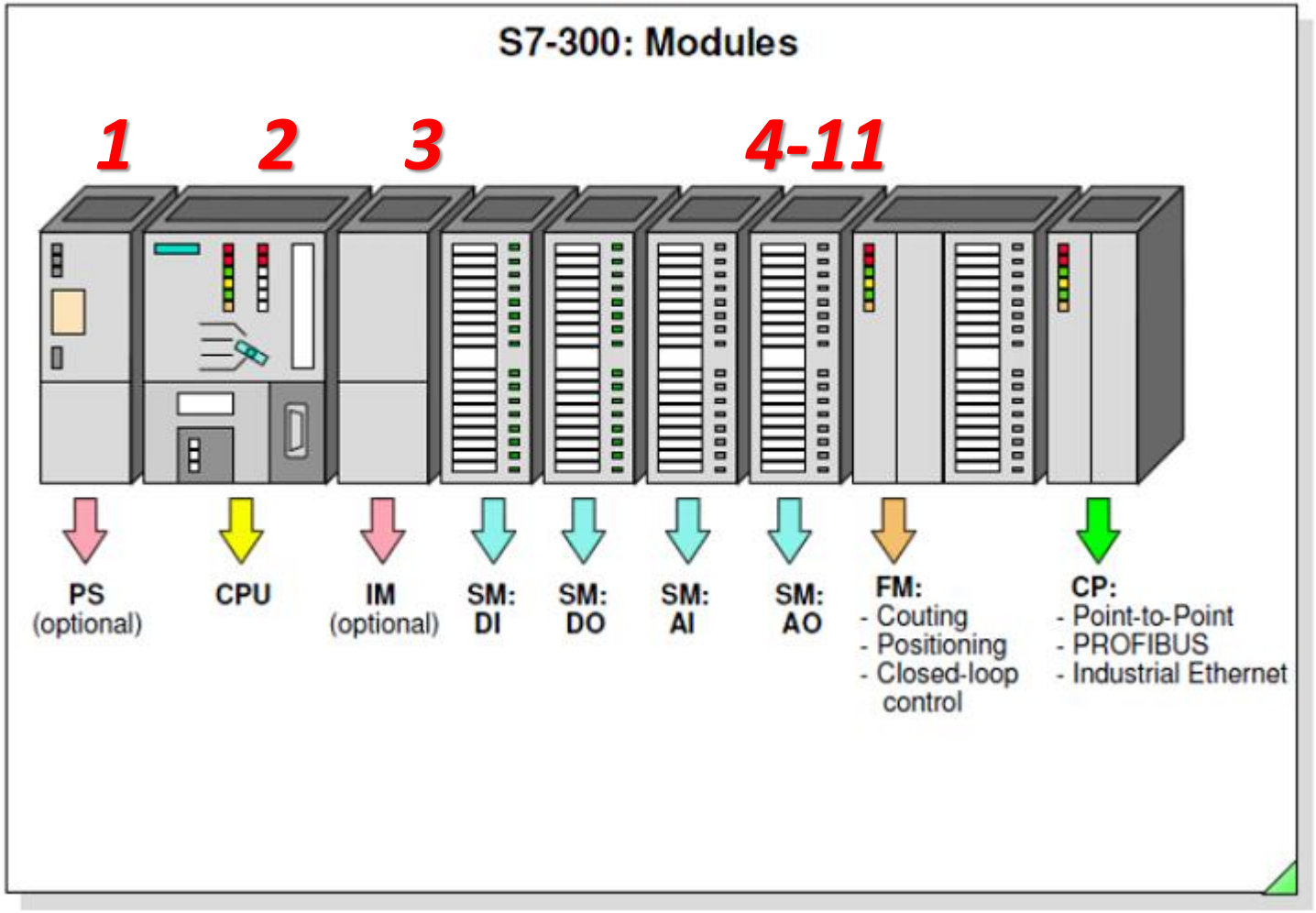
بررسی

PLC S7-300

CPU S7-300


منبع تغذیه
چراغ های وضعیت
ورودی دیجیتال
خروجی دیجیتال
PS307 2A
DC24V
VOLTAGE SELECTOR
ON
OFF
SIEMENS
CPU 314
SF
BAF
DC 5V
FRCE
RUN
STOP
RUN-P
RUN
STOP
MRES
SM321
SM322
باتری BackUP
سوئچ وضعیت
پورت ارتباطی MPI

ماژول های PLC زیمنس



سفارش دهی تجهیزات زیمنس

تمام تجهیزات اتوماسیونی ساخت شرکت زیمنس یک کد فنی منحصر بفرد تحت عنوان ***Order Number*** *دارند که از طریق این کد قابل سفارش هستند:*

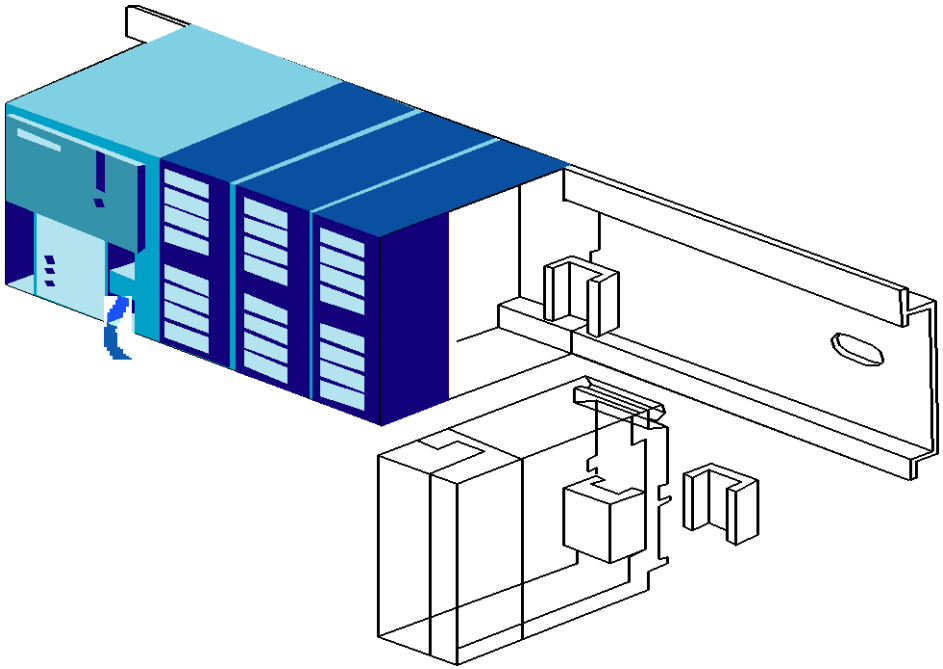
Order Number	نام قطعه
6ES7 307-1EA00-0AA0	PS 307 5A
6ES7 315-2AF00-0AB0	CPU 315-2DP
6ES7 321-1BH02-0AA0	DI 16xDC24V
6ES7 322-1BH01-0AA0	DO 16xDC24V/0.5A
6ES7 331-7KB82-0AB0	AI 2x12Bit
6ES7 332-7ND02-0AB0	AO 4x16Bit
6ES7 365-0BA00-0AA0	IM 365 S-R
6ES7 350-1AH00-0AE0	FM 350-1 COUNTER MODULE
6GK7 343-1GX00-0XE0	CP 343-1
6GK7 342-5DA00-0XE0	CP 342-5

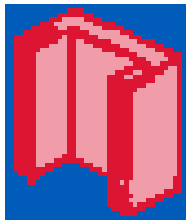
ریل – رک

RACKS یک کنترلر قابل برنامه ریزی ممکن است شامل چندین قفسه باشد، که از طریق کابل های باس به یکدیگ ر متصل می شوند . منبع تغذیه ،CPU ، و ماژول های I/O (CP , FM , SM) به قفسه مر کزی متصل می شوند . اگر در قفسه مرکزی برای ماژول های I/O ناحیه کافی وجود نداشته باشد یا اگر بخواهید همه ماژول های I/O یا بعضی از آنها جدا از قفسه مرکزی قرار داده شوند ،می توان آنها را در قفسه های توسعه (Expation rack) موجود قرار داد که از طریق ماژول های واسط به قفسه مرکزی متصل می شوند . رکهای سری ۳۰۰ دارای ۱۱ اسلات هستنداین سری فقط یک نوع رک دارد که هم به عنوان رک اصلی و هم به عنوان رک اضافی استفاده می گردد و فقط نقش نگهدارندگی برای ماژولها دارد.ماژول ها باید روی آن کنار یکدیگر و بدون فاصله قرار گیرند.

رک برای PLC S7-300

۱. یازده اسلات دارد.
۲. بصورت ریل است.
۳. فقط نقش نگهدارنده برای ماژولها دارد.
۴. ماژولها باید روی آن کنار هم و بدون فاصله قرار گیرند.
۵. فقط یک نوع دارد که هم بعنوان رک اصلی و هم بعنوان رک اضافی استفاده میگردد.



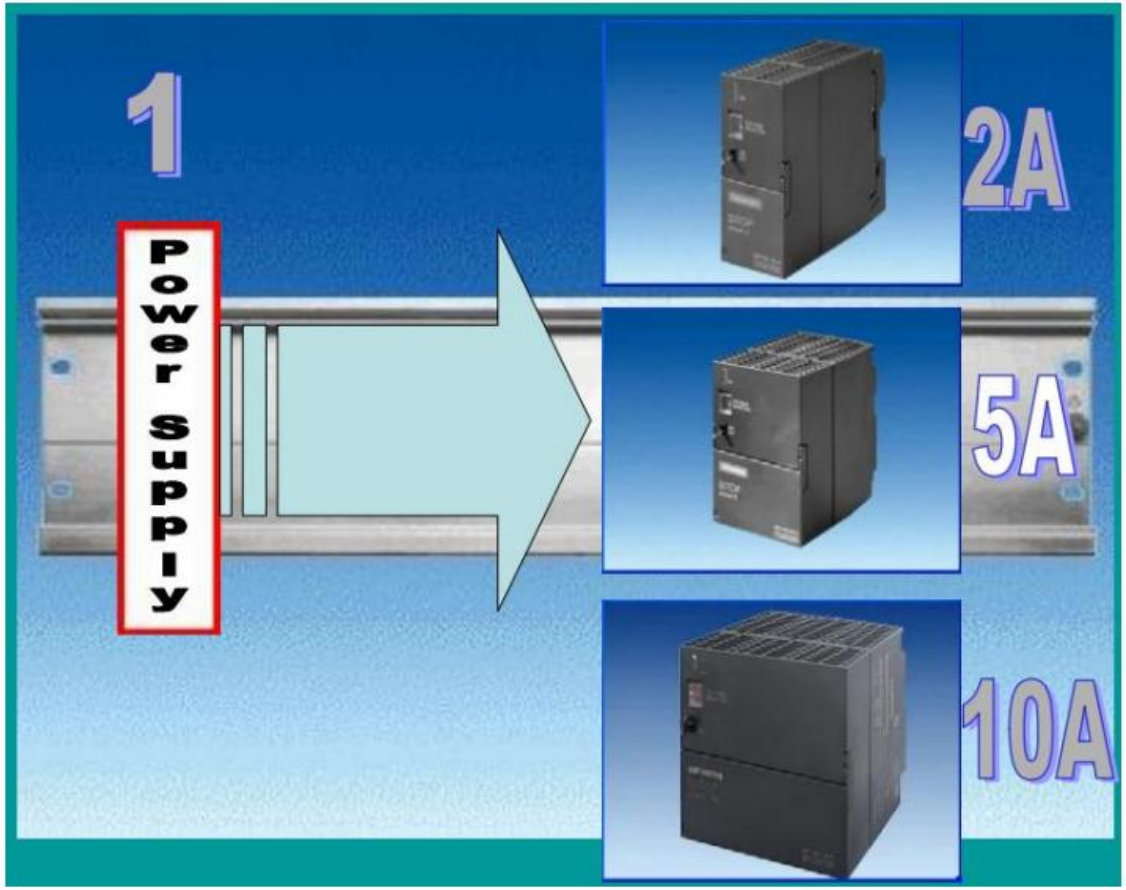


Bus Connector

منبع تغذیه PLC

منبع تغذیه ، برای برق رسانی به تمام قطعات سیستم به کار می رود و مقادیر معمول آن ۲۴ ولت مستقیم ، ۱۱۰ ولت متناوب و ۲۳۰ ولت متناوب است. بنابر این ، کاربر باید پیش از خرید PLC ، امکان تغذیه PLC را بررسی کند.

PS منبع تغذیه ولتاژهای مورد نیاز PLC را تامین می کند . در سری S7-300 سه نوع PS وجود دارد که هر سه نوع PS ولتاژ 120/230 V AC را دریافت می کنند و ولتاژ 24 V DC را به خروجی خود می دهند که این PS ها به صورت SWITCH عمل می نمایند.تفاوتهای این PS ها در جریان خروجی آنها (OUTPUT CURRENT) است که جریان خروجی آنها در رنجهای 5 A , 2 A و 10 A است.

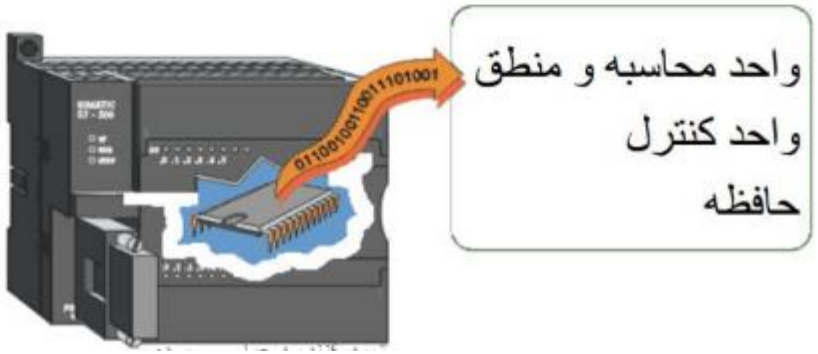


Type of Power Supply	PS 305	PS 307
Input Voltage	24/48/72/96/110 VDC	120/230 VAC
Output Voltage	24 VDC	24 VDC
Output Current	2A	2/5/10A

واحد پردازشگر مرکزی CPU

CPU ، مغز PLC است ، یعنی دستگاهی که تمام عملکرد ها را با ترتیبی درست کنترل می کند . اما باید توجه داشت که CPU هوشمند نیست وخودش فکر نمی کند بلکه فقط از مجموعه دستور العمل هایی که در حافظه آن قـرار می گیرد ، تبـعیـت می کند . CPU ها بر اساس مقدار حافظه و سرعتی که دارند و همچنین تعداد ورودی و خروجی هایی که می پذیرند ، دسته بندی می شوند.

ساختار داخلي CPU :



CPU در S7-300

Option :

1.31X C (Compact, CPU+I/O)
2.31X F (Failsafe)
3.SIPLUS (برای محیط های اسیدی و شیمیای)
4.31X IFM (Integrated Function Module)
5.31X T (Technologic , Include Motion Control Function Module)

Connection :

1.31X (MPI)
2.31X -2DP (MPI+DPI)
3.31X-2PtP (MPI+PtP)
4.31X -2PN (MPI+PN)
5.31X-2PN/DP (MPI/DP+PN)
6.31X-3PN/DP (MPI/DP+PN+PN)

MPI: Multi Point Interface

PtP: Point to Point

PN: Profinet (Industrial Ethernet+Profibus DP)

CPU 314

CPU 313C

CPU 312C

CPU 312

CPU 318-2DP

CPU 317-2DP

CPU 315-2DP

S7-300

CPU	314	315-2DP	316	318-2DP
Memory	24k Data and program	32k Data and program	128k Data and program	512k Data and program
Bit memory	2k	2k	2k	8k
Timer	128	128	128	512
Counter	64	64	64	512
Digital I/O	Max 1024	Max 8192	Max 16384	Max 65536
Analog I/O	Max 256	Max 1024	Max 1024	Max 1024
Interface	MPI	MPI/DP	MPI	MPI/DP

CPU 314

CPU 316

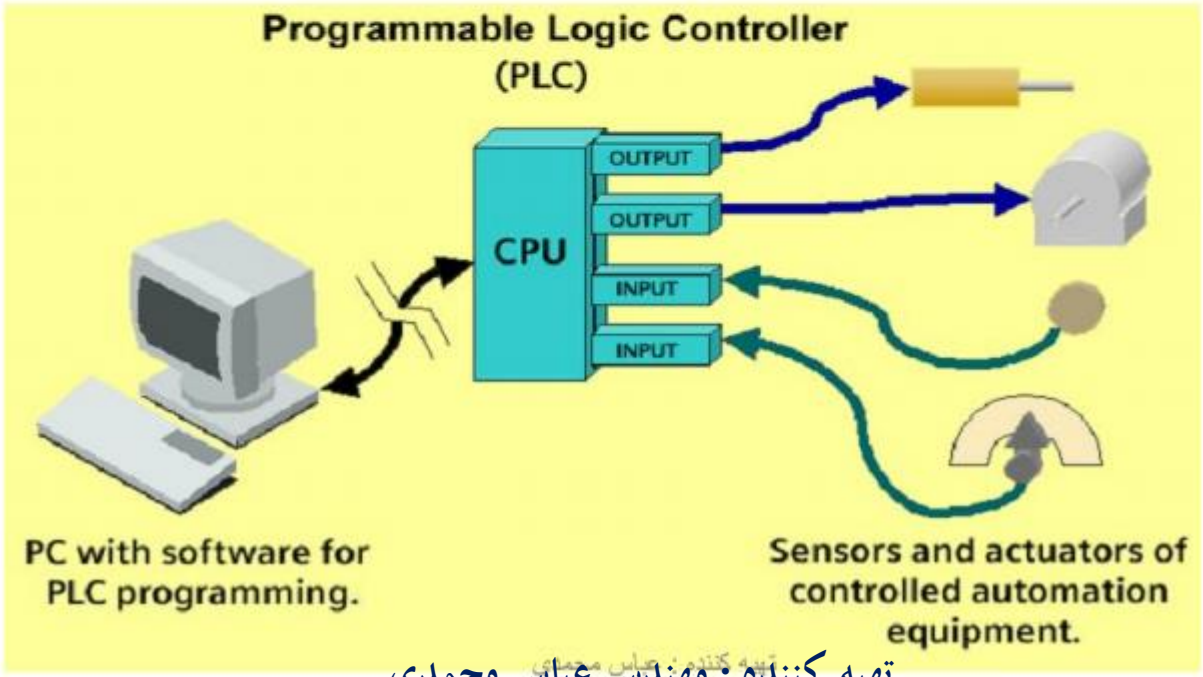
CPU 315-2DP

CPU 318-2DP

Signal Module

واحدهای ورودی – خروجی (SM)

سیگنالها و پیغامهایی که در واحد تحت کنترل هستند توسط کارت های ورودی و خروجی با **CPU** در ارتباط هستند. کارت های ورودی و خروجی بر اساس مدارات درون خود با سیگنالها و بوسیله باس داخلی با **CPU** در ارتباط می باشند.

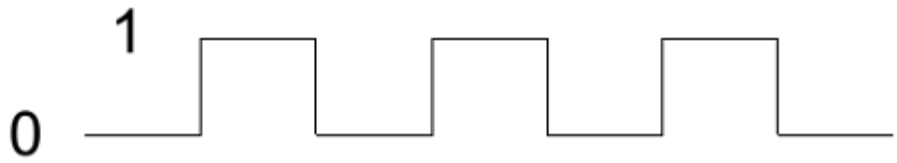


سیگنال دیجیتال

تعریف سیگنال دیجیتال:

سیگنالی است که تنـها دو وضعیت داشته باشد. مثلا اگر ولتـاژ باشد تنـها دو مقـدار ۰ و ۲۴ ولت را شامل شود. که در این صورت یکی از آنهـا نماینـده منطـق صـفر و دیگـری نماینـده منطق یک خواهد بود.

تمام وضعیت های دو حالته با سیگنال دیجیتال مشخص می شوند. مثـل خـاموش / روشـن بودن یک موتور یا باز / بسته بودن یک شیر برقی.



سیگنال آنالوگ

سیگنالی است که چندین وضعیت را در یک رنج تغییر سیگنال شامل می شود. مثلا سیگنال آنالوگ در رنج ۴ تا ۲۰ میلی آمپر .

سیگنال های آنالوگ معمول به شرح زیر هستند:

- **4-20 mA ,0-20 mA**
- **1-5 Volt ,0-10 Volt**
- **0-500 mV Thermocouple**
- **RTD**

سیگنال دیجیتال عمدتا در وضعیتهای دو گانه مانند **…., Stop – Start , OFF-ON**

کاربرد دارد، در حالیکه سیگنالهای آنالوگ حاوی مقداری هستند مثلا دمای از ۰ تا ۵۰ درجه ، فشار ۰ تا ۱۰۰ بار ، ....

تعریف ورودی ها و خروجی هایPLC :

برای کنترل کردن سیستمی توسط PLC لازم است که اتفاقاتی که در سیستم روی میدهد به اطلاع PLC برسد تا PLC با بررسی آنها و اجرای منطقی که برایش تعریف شده ، فرامین لازم جهت کنترل سیستم را صادر کند .

ماشین ها وابزار هایی که در سیستم وجود دارند ، وضعیت های خود و یا اتفاقاتی را که رخ میدهد توسط سیگنالهای الکتریکی بیان می کنند ، که این سیگنالها بنا به نوع آن ماشین و یا اتفاق ، ماهیت های متفاوتی دارند . مثلا موتوری که در حالت کار کردن است یک ولتاژ ۲۴ ولتی را طریق یکی از کنتاکت رله هایش منتقل می کند و چنانچه متوقف باشد، دیگری ولتاژی را منتقل نمی کند . بنابراین دریافت ولتاژ ۲۴ ولت از آن کنتاکت را می توان نشانه کارکردن موتور و عدم دریافت آن را نشانه متوقف بودن آن موتور دانست .

ورودی دیجتال

ورودی های دیجیتال ، سیگنال های دیجیتالی هستند که از محیط بیرون توسط سخت افزاری یه نام کارت ورودی دیجیتال در PLC دریافت می شوند.

سیگنال های دریافتی از المان های زیر ورودی دیجیتال محسوب می شوند:

- **کنتاکت های رله ها**
- **Limit Switch** ها
- **Push Button** ها
- **Proximity Switch** ها
- **Process Switch** ها

تهیه کننده : مهندس عباس محمدی

Signal Module

(DI, DO, AI, AO)

Digital Input

SM 321; DI 32 24 VDC
SM 321; DI 16 24 VDC
SM 321; DI 16 120 VAC
SM 321; DI 8 120/230VAC
SM 321; DI 32 120 VAC

Digital Output

SM 322; DO 32 24 VDC/0.5 A
SM 322; DO 16 24 VDC/0.5 A
SM 322; DO 8 24 VDC/2 A
SM 322; DO 16 120 VAC/1 A
SM 322; DO 8 120/230VAC/2A

Digital I/O

SM 323; DI 16/DO 16
24 VDC/0.5 A

SM 323; DI 8/DO 8
24 VDC/0.5 A

Analog Input / Output

SM 331; AI 8 12 Bit
SM 331; AI 2 12 Bit
SM 332; AO 4 12 Bit
SM 332; AO 4 16 Bit
SM 334; AI 4/AO 212 Bit

Digital Input (DI) :

تقسیم بندی کارت های Digital Input		
از نظر تعداد ورودی	**از نظر ولتاژ**	**از نظر قابلیت های خاص**
۴ ورودی	24 VDC	بدون ویژگی خاص
۸ ورودی	48 VDC	تشخیص قطع شدن تغذیه Diagnostic Interrupt
۱۶ ورودی	120 VDC	ایجاد وقفه بر اساس لبه ورودی Hardware Interrupt
۳۲ ورودی	230 VDC	تاخیر در گرفتن ورودی Input Delay

Digital Output (DO) :

تقسیم بندی کارت های Digital Output			
از نظر تعداد خروجی	**از نظر جریان خروجی**	**از نظر ولتاژ**	**از نظر قابلیت های خاص**
۴ ورودی	بدون رله : 0.5, 1, 1.5, 2A	24 VDC	بدون ویژگی خاص
۸ ورودی		48 VDC	تشخیص قطعی Wire Break
۱۶ ورودی	با رله : 5, 8A	120 VDC	تشخیص اتصال کوتاه Short Circuit
۳۲ ورودی		230 VDC	واکنش در موقع توقف CPU Reaction to CPU Stop

DI/DO :

تقسیم بندی کارت های DI/DO			
از نظر تعداد ورودی/ خروجی	از نظر جریان خروجی	از نظر ولتاژ	از نظر قابلیت های خاص
۱۶ورودی + ۱۶ خروجی ۸ ورودی + ۸ خروجی	0.5A	24 VDC	بدون ویژگی خاص

Analog Input (AI) :

تقسیم بندی کارت های Analog Input		
از نظر تعداد ورودی	**از نظر نوع سیگنال**	**از نظر قابلیت های خاص**
۲ ورودی	ولتاژ	بدون ویژگی خاص
۴ ورودی	جریان	ایجاد وقفه Hardware Interrupt
۸ ورودی	مقاومت	تشخیص قطعی Diagnostic Interrupt
	TC	
	RTD	
	ترکیبی از موارد فوق	

Analog Output (AO):

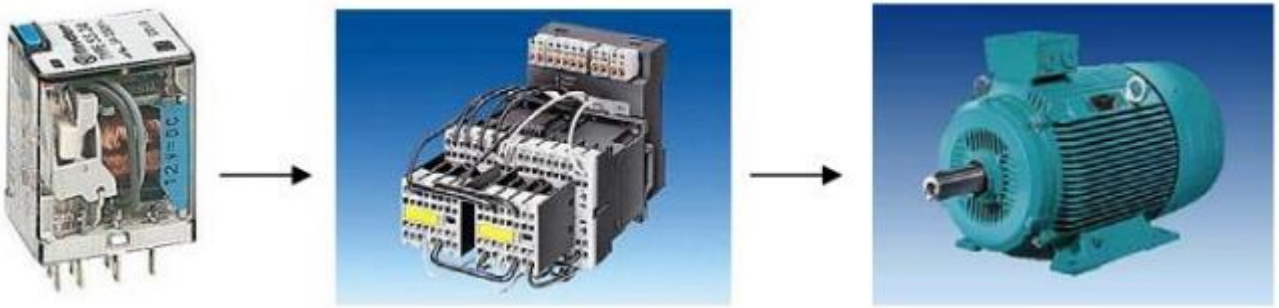
تقسیم بندی کارت های Analog Output		
از نظر تعداد خروجی	**از نظر نوع سیگنال**	**از نظر قابلیت های خاص**
۲ ورودی	ولتاژ	بدون ویژگی خاص
۴ ورودی	جریان	تشخیص قطعی Wire Break
۸ ورودی	ترکیبی از موارد فوق	تشخیص اتصال کوتاه Short Circuit
		واکنش در موقع توقف CPU Reaction to CPU Stop

خروجی دیجیتال

خروجی های دیجیتال سیگنال های دیجیتالی هستند که از PLC توسط سخت افزاری به نام کارت خروجی دیجیتال به محیط بیرون منتقل می شوند.

موارد زیر نمونه هایی از خروجی دیجیتال هستند:

- تحریک کردن بوبین یک رله
- روشن / خاموش کردن چراغ سیگنال ها



ورودی آنالوگ

ورودی های آنالوگ ، سیگنال های آنالوگی هستند که از محیط بیرون توسط سخت افزاری یه نام کارت ورودی آنالوگ در **PLC** دریافت می شوند. در کارت ورودی آنالوگ ، عمل تبدیل آنالوگ به دیجیتال صورت می گیرد

سیگنال های دریافتی از المان های زیر ورودی آنالوگ محسوب می شوند:

- **Temperature Instrument**
- **Pressure Instrument**
- **Level Instrument**
- **Flow Instrument**
- **Load Cell**

Temperature Transmitter

Flow Transmitter

Load Cell

Level Transmitter

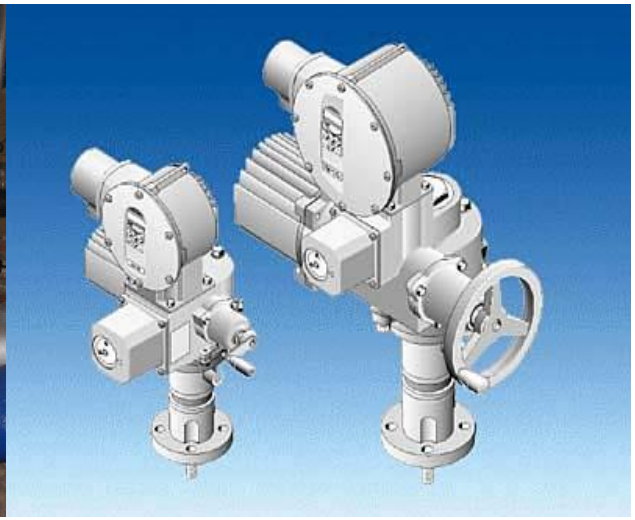
Pressure Transmitter

خروجی آنالوگ

خروجی های آنالوگ سیگنال های آنالوگ ی هستند که از **PLC** توسط سخت افزاری به نام کارت خروجی آنالوگ به محیط بیرون منتقل می شوند. در کارت خروجی آنالوگ ، عمل تبدیل دیجیتال به آنالوگ صورت می گیرد.

موارد زیر نمونه هایی از خروجی آنالوگ هستند:

- **سیگنال ارسالی به کنترل والوها**
- نمایشگرها



Control Valve

Display

Function Module

ماژول FM

FM فرآیند های بحرانی یا پیچیده را مستقل از CPU اجرا می کند.و به اصطلاح باری از دوش CPU بر می دارد . ورودی ها مستقیما به این ماژول ها داده می شود و خروجی ها نیز مستقیما از آنها ارسال می شوند . در عین حال FM ها می توانند با CPU تبادل دیتا داشته باشند . برای تنظیم های مربوط به فانکشن داخلی FM علاوه بر STEP7 پکیج نرم افزاری دیگری نیاز است که این پکیج علاوه بر ابزار پیکر بندی فانکشن بلاکهای خاص FM که توسط آنها پارامتربه FM اختصاص داده می شوند عرضه می شوند.

Function Module S7-300

FM350-1 / FM350-2	کانتر برای شمارش های سریع هستند.پالس های فرکانس بالا که توسط کارتهای DI قابل آشکارسازی نیستند.
FM351/FM353/FM354	برای کنترل موقعیت موتورهای پله ای و سروموتورها و درآیوهای لرزشی بکار می روند.
FM355	این ماژول یک لوپ کنترلر ۴ کاناله است که می تواند برای کنترل فشار و دما بکار رود

FM353

Communication Processor
CP

به منظور ارتباط PLC با شبکه های مختلف صنعتی نیاز به پورت شبکه روی CPU یا نصب کارت شبکه در کنار CPU می باشد.

انواع کارت های شبکه:

۱- کارت شبکه پروفی باس:

- کارت شبکه پروفی باس (Profibus-DP و Profibus-FMS)
- کارت شبکه برای Profibus-PA روی رک عرضه نشده است و از مبدل DP/PA برای Profibus-PA استفاده می گردد.

۲- کارت شبکه اترنت:

- کارت های با امکانات کم ولی مقرون به صرفه مانند کارت CP343-1 Lean
- کارت های با امکانات متوسط و پر کاربرد مانند کارت CP343-1
- کارت های با امکانات پیشرفته مانند کارت CP343-1 Advanced

Communication Processor
CP

۳- کارت شبکه ASI:

- کارت ASI برای اتصال سیگنالهای دیجیتال
- کارت ASI برای اتصال سیگنالهای دیجیتال و آنالوگ

۴- کارت شبکه PtP:

- کارت CP341 که پروتکل ASCI و RTU مدباس را ساپورت می کند.
- کارت CP340 که فقط پروتکل ASCI مدباس را ساپورت می کند.

Communication Processor - S7 300

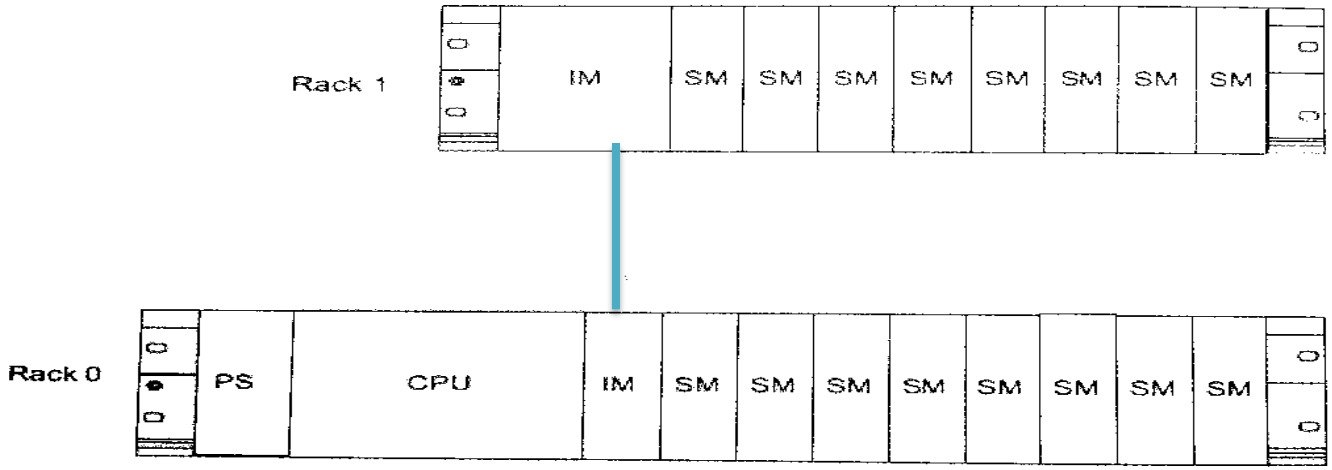
CP 342-5
برای ارتباط با شبکه **PROFIBUS-DP** بکار می رود

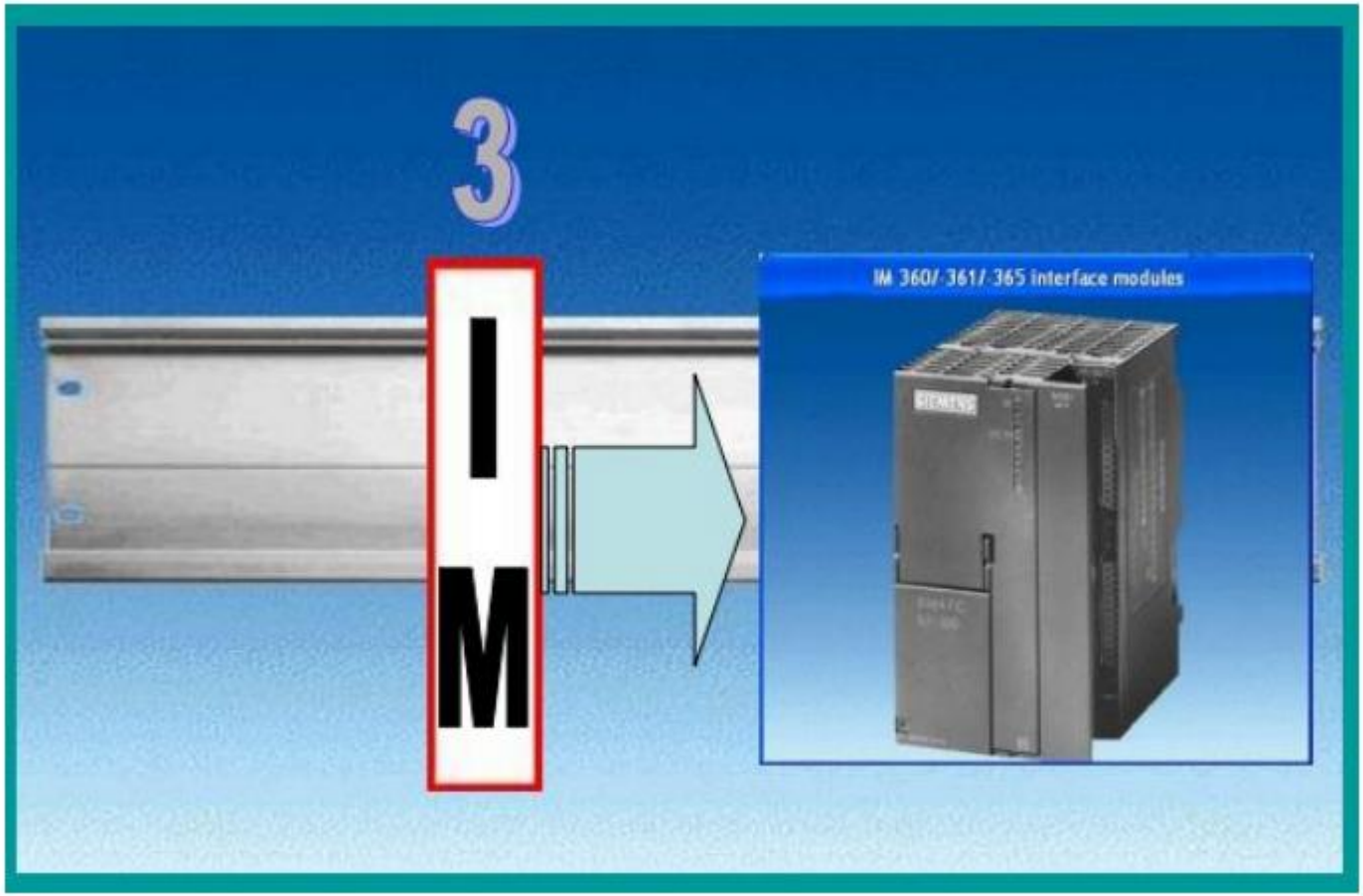
CP 343-1
برای ارتباط با شبکه **Industrial Ethernet** بکار می رود

تهیه کننده : مهندس عباس محمدی

Interface Module- S7 300

- رک ۳۰۰ دارای ۱۱ اسلات می باشد
- اسلات اول برای PS واسلات دوم برای CPU استفاده می شود.
- اسلات سوم جهت ماژول IM استفاده می شود(در صورت نیاز)
- اسلات ۴ تا ۱۱ برای ماژول های I/O و ماژول های FM و CP و ... استفاده می شود.
- در صورتی همه اسلاتهای رک اصلی پر شود و باز هم ماژول اضافی موردنیاز باشد از Expansion Rack (رک توسعه) استفاده می کنیم.
- IM های بصورتی جفتی استفاده می شوند. یک IM در رک اصلی و یک IM در رک توسعه و ارتباط آنها از طریق یک کابل مخصوص برقرار می شود.
- در رک توسعه می توان تا ۸ ماژول اضافه کرد و اگر باز هم ماژول اضافی موردنیاز باشد می توان رک توسعه دیگری نیز بکار برد، در S7-300 حداکثر ۳ رک توسعه می توان استفاده کرد.




3
I
M
IM 360/-361/-365 interface modules
SIEMENS

Interface Module

انواع IM در S7-300

در S7-300 سه نوع IM وجود دارد که عباتند از:

- IM 360 S: جهت استفاده در رک اصلی
- IM 361 R: جهت استفاده در رک توسعه
- IM 365 S-R: جهت استفاده در رک اصلی و توسعه

دو روش برای برقراری اتصال از طریق IM وجود دارد:

۱- IM360S / IM361R

۲- استفاده از دو IM365

Interface Module

روش اول:

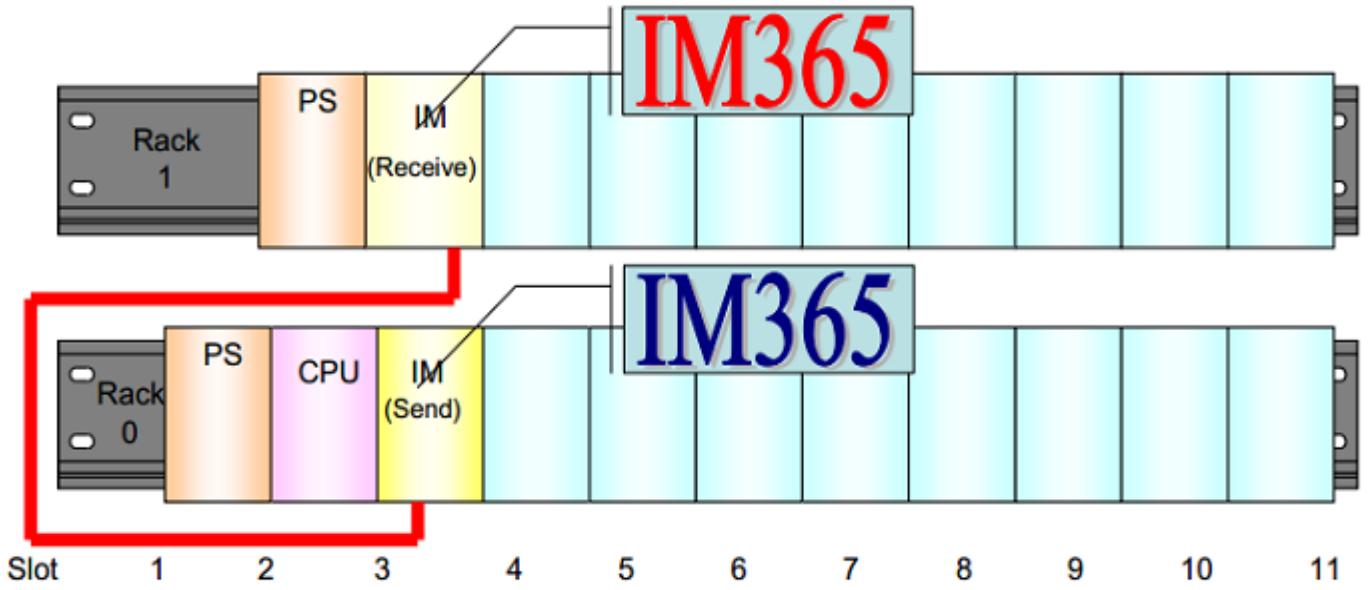
۱- طول کابل IM می تواند ۱، ۲/۵، ۵ یا ۱۰ متر باشد.

۲- در رک توسعه علاوه بر SM می توان CP و FM نیز قرار داد.

۳- تا ۳ رک توسعه می توان اضافه نمود.

۴- رک توسعه نیاز به منبع تغذیه دارد.

Interface Module



روش دوم:

۱-می توان فقط یک رک توسعه ایجاد کرد.

۲- رک توسعه نیاز به منبع تغذیه ندارد.

۳- طول کابل IM در این حالت ۱ متر است.

۴-در رک توسعه CP های اترنت و پروفی باس و برخی از FM ها را نمی توان قرار داد.

بررسی عملکرد
CPU
در PLC S7

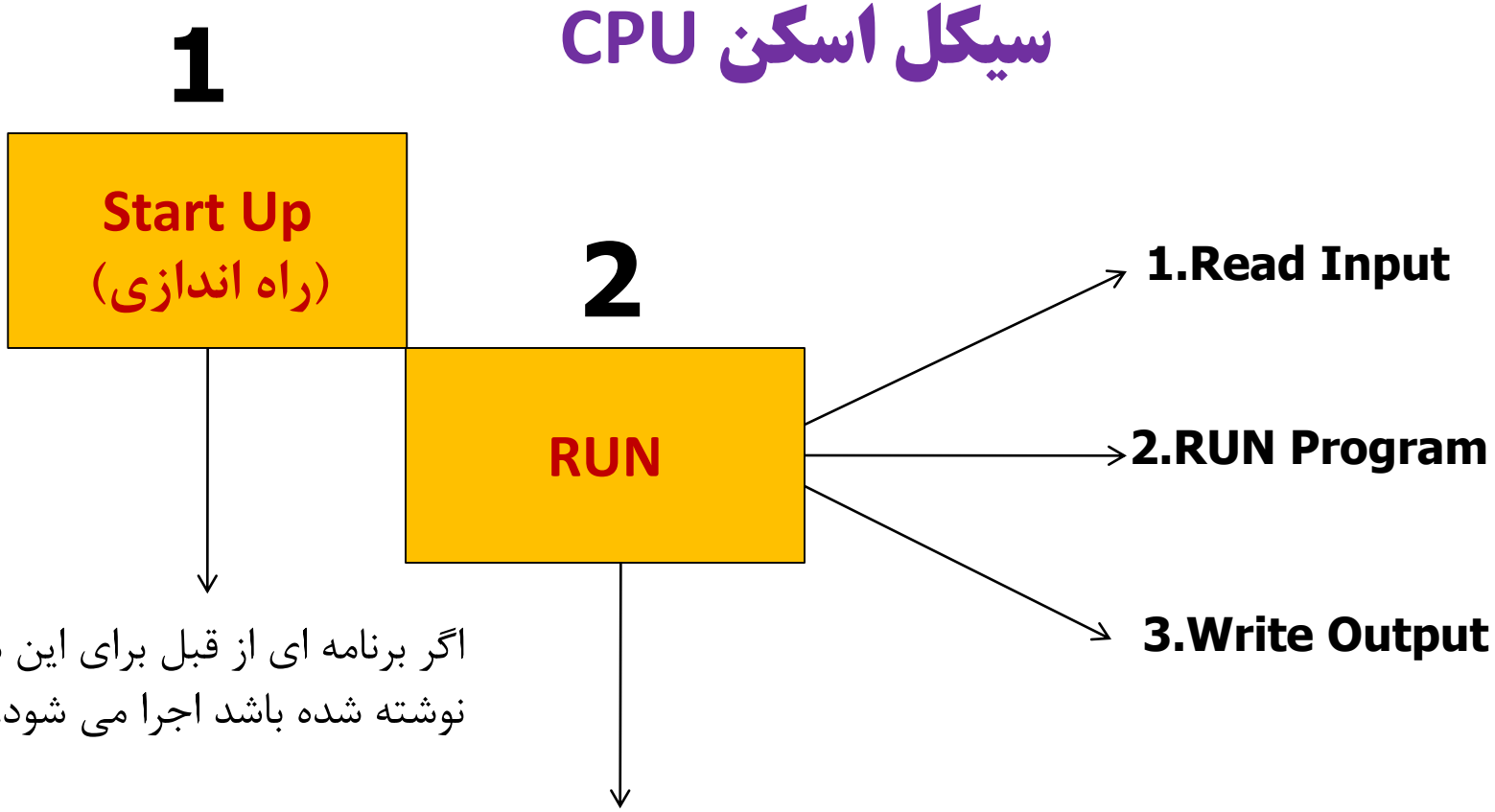
تهیه کننده : مهندس عباس محمدی

سیکل اسکن CPU

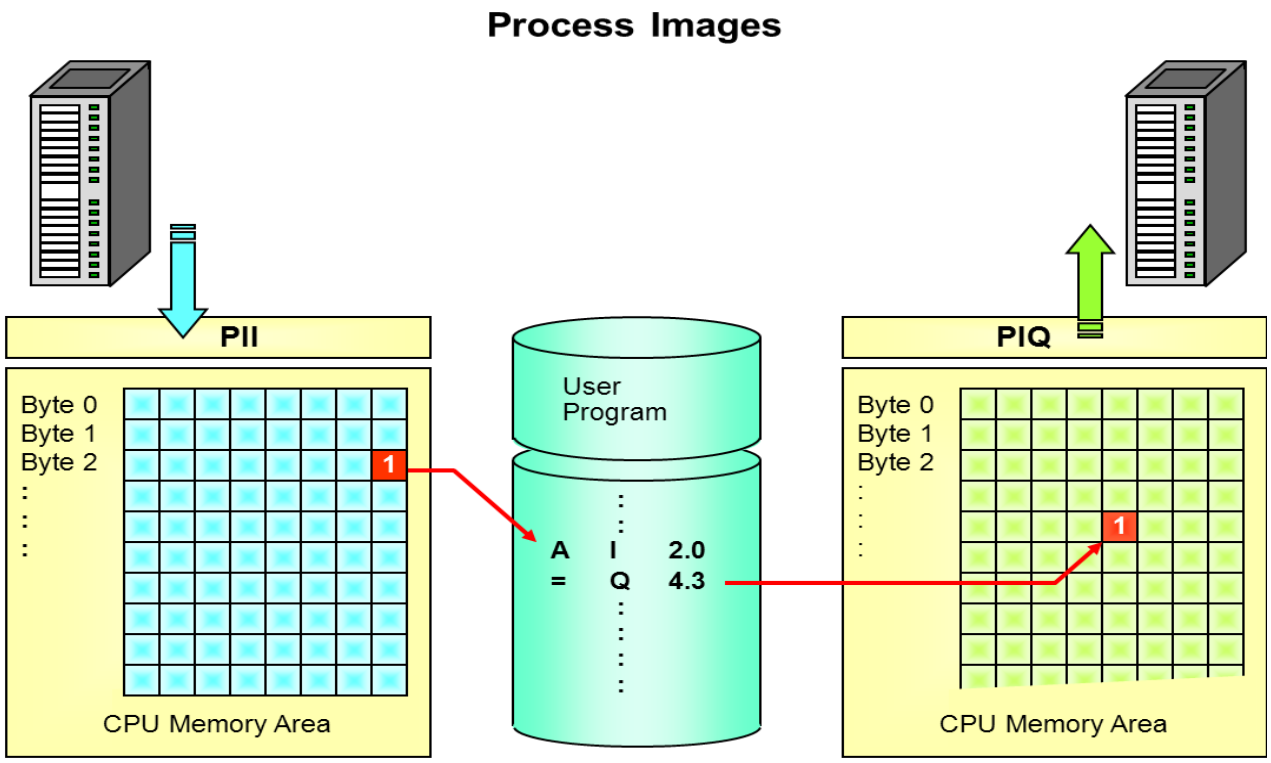
در هر سیکل اسکن قبل از اجرای برنامه ، ***CPU*** وضعیت تمام ورودی ها را به طور یکجا می خواند و در مکانی از حافظه موسوم به ***PII*** (جدول تصویر ورودی ها) ذخیره می کند.سپس شروع به اجرای برنامه می کند. ***CPU*** در حین اجرای برنامه به ورودی ها مراجعه نمی کند .اگر در طول سیکل اسکن تغییراتی در ورودی ها حاصل شود ، این تغییرات تا سیکل اسکن بعدی به ***PII*** منتقل نمی شود . ***PLC*** در حین اجرای برنامه ، خروجی ها را در مکانی از حافظه موسوم به ***PIQ*** منتقل می کند .و در پایان ، بعد از اجرای کامل برنامه ،نتایج را به طور یکجا به خروجی ها ارسال می کند.پس از آن سیکل اسکن بعدی به همین ترتیب آغاز می شود.

PII(Process Image Input) : محلی از حافظه درون CPU که مقادیر ورودی ها در آن کپی می شود.
PIQ(Process Image Output) : محلی از حافظه درون CPU که مقادیر خروجی ها در آن کپی می شود.
OB1: بلوکی که برنامه اصلی جهت اجرا در آن قرار گرفته است و مرتباً در حال اجراست.

سیکل اسکن CPU

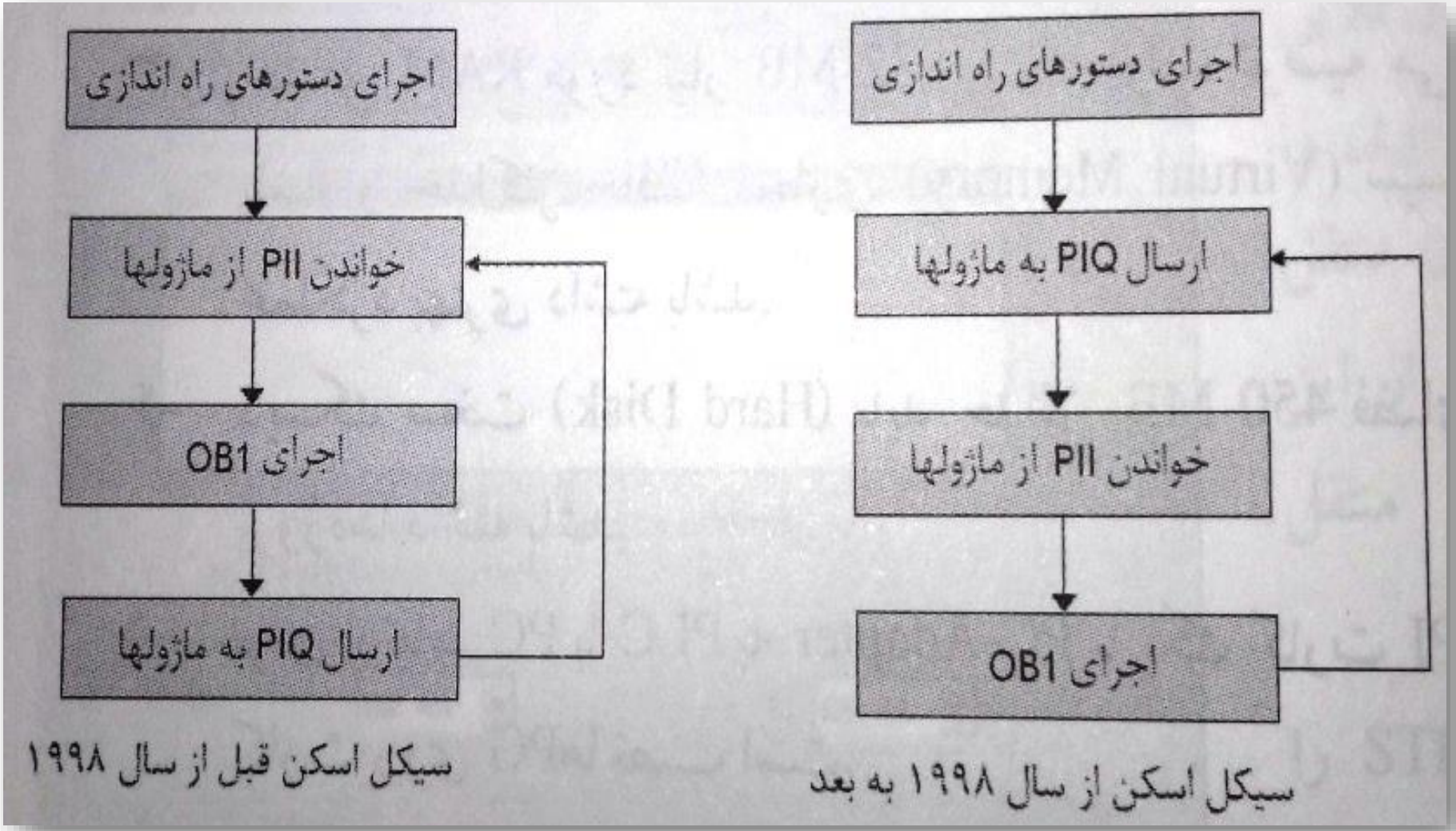


سیکل اسکن CPU

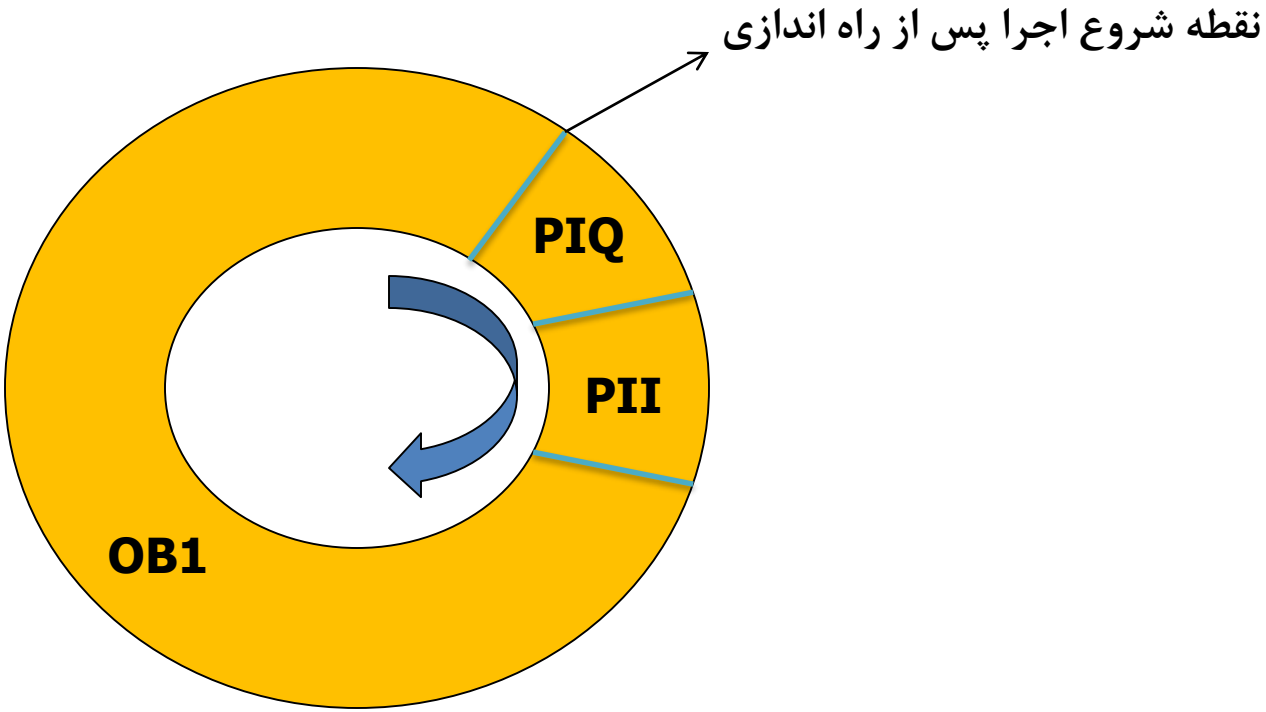


باید زمان سیکل اسکن **CPU** زمان کمی باشد تا امکان بروز رسانی **PII** و **PIQ** وجود داشته باشد.

سیکل اسکن CPU



سیکل اسکن CPU

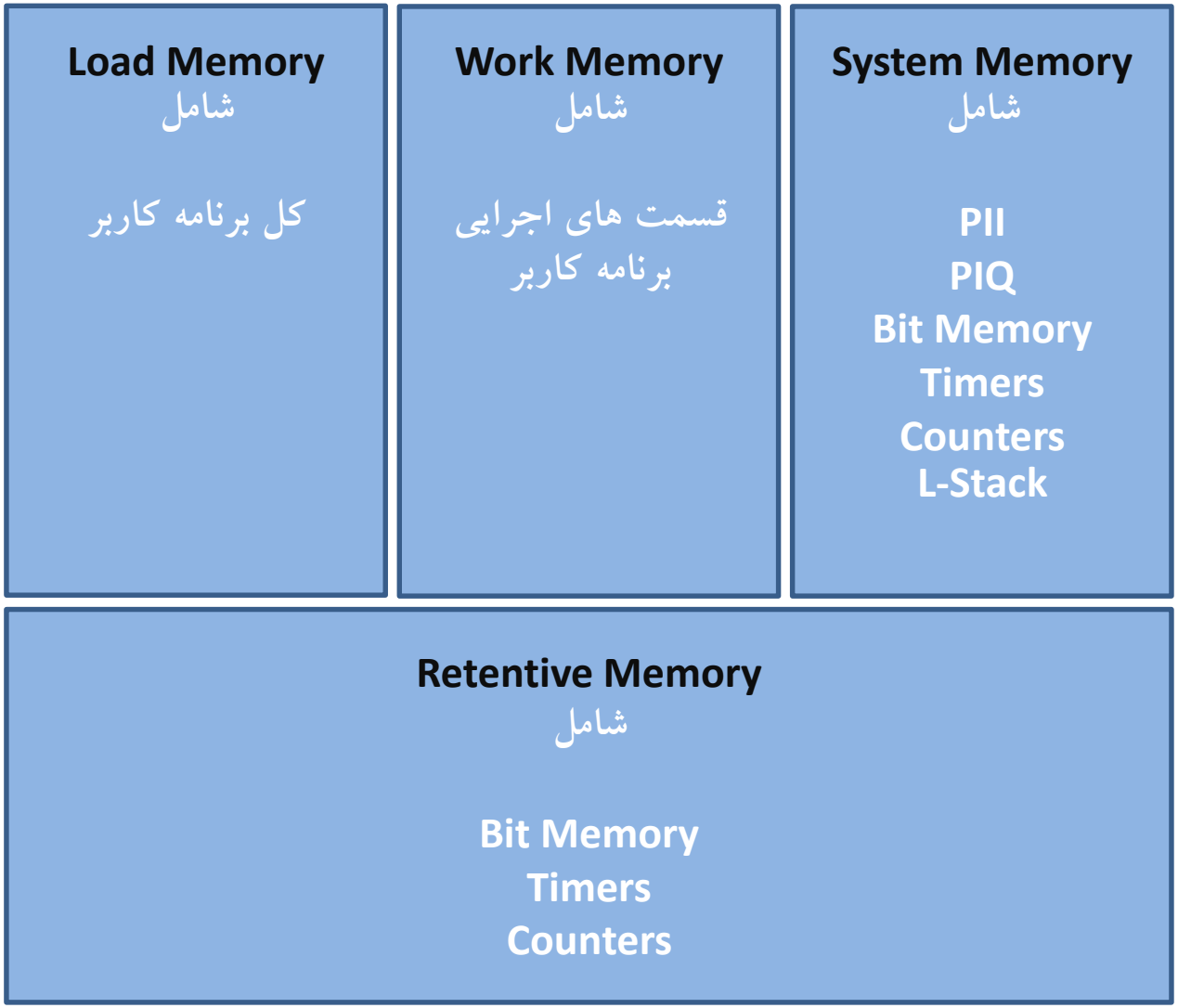


زمانی که طول می کشد یک سیکل اسکن اجرا شود از جمع زمانهای زیر بدست می آید:

- زمان مربوط به خواندی ورودی و Update کردن PII
- زمان مربوط به ارسال خروجی از PIQ
- زمان پردازش برنامه اصلی
- زمان مربوط به سیستم عامل
- زمان مربوط به تبادل دیتا با شبکه

بخش های حافظه

نواحی مختلف حافظه در CPU های S7



بخش های حافظه

نواحی مختلف حافظه در CPU های S7

Load Memory

زمانی که برنامه کاربر شامل پیکربندی سخت افزار، تنظیمات شبکه و دستورات برنامه نویسی و سایر جزئیات به PLC دانلود می گردد، در قسمت Load Memory قرار می گیرد.

نکات:

۱- در CPU های 400 و در CPU های 300 قدیمی، حافظه Load Memory بصورت داخلی وجود دارد که از جنس RAM است(برای عدم پاک شدن اطلاعات در صورت قطع تغذیه نیاز به باتری پشتیبان دارد). این حافظه را می توان با کارت حافظه MC افزایش داد، و می توان بدون کارت حافظه برنامه را به CPU دانلود کرد و با آن ارتباط برقرار نمود. کارت حافظه برای افزایش حافظه است و در نوع فلش نقش نگهداری اطلاعات در صورت قطع تغذیه را نیز دارد.

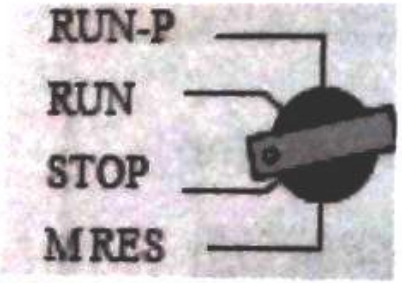
در این CPU ها برنامه به طور عادی در RAM قرار می گیرد و برای انتقال برنامه در کارت فلش بایستی از برنامه Step7 اقدام نمود.

۲- در CPU های 300 جدید حافظه Load Memory داخلی وجود ندارد و بصورت الزامی بایستی از کارت حافظه از نوع فلش استفاده کرد. این کارتها به MMC معروف هستند، بدون کارت حافظه امکان دانلود و برقراری ارتباط با CPU وجود ندارد.

در این CPU ها برنامه به طور عادی در کارت فلش انتقال می یابد.

۳- حافظه Load Memory هر CPU را تا حد مشخصی توسط کارت حافظه می توان افزایش داد.

کلید ۳ وضعیتی روی CPUهای جدید

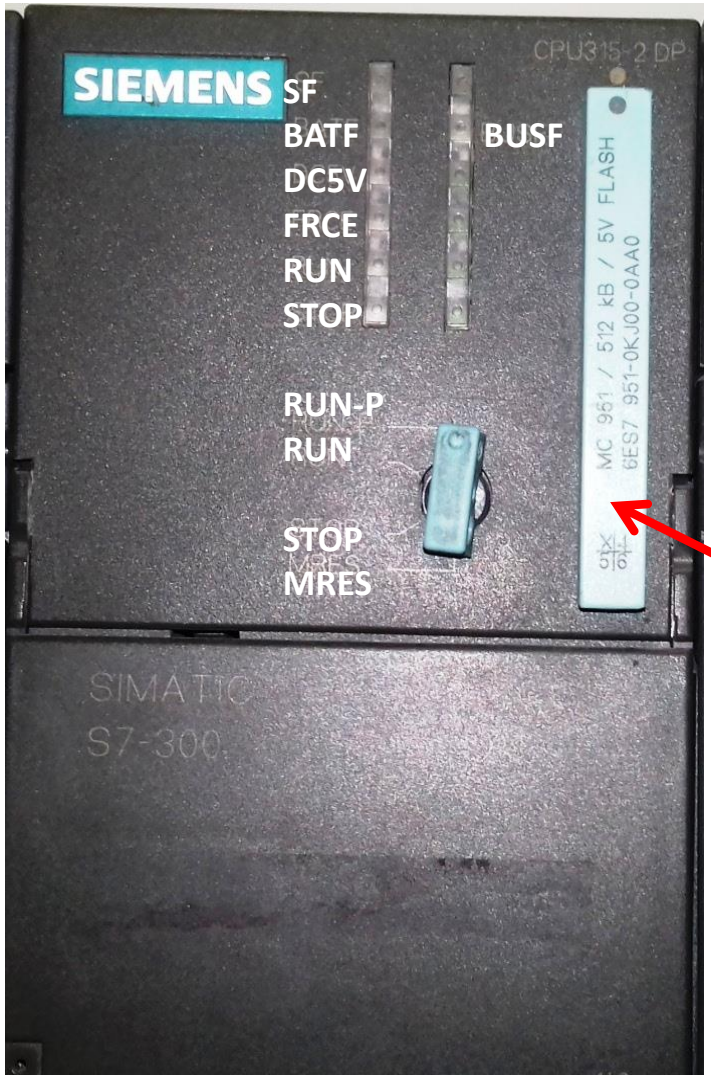


کلید ۴ وضعیتی روی CPUهای قدیمی

مدهای کاری PLC

STOP	در این مد پردازش برنامه متوقف می شود، دسترسی به **I/O** ها وجود ندارد، در این حالت می توان به **PLC** دانلود و یا از آن آپلود نمود
START-UP	وقتی **CPU** توسط کلید روی آن یا از طریق نرم افزار راه اندازی می شود، اصطلاحاً **CPU** در حالت **STARTUP** قرار گرفته است.
RUN	وقتی **CPU** از مد راه اندازی عبور کرد وارد مد **RUN** می شود. در این مد، **CPU** به اجرای سیکل اسکن که یک کار تکراری است می پردازد و تا زمانی که به مد **STOP** نرفته اجرای سیکل ادامه می یابد.در مد **RUN** فقط امکان آپلود از **CPU** وجود دارد ولی امکان دانلود وجود ندارد.
RUN-P	این مد همانند مد **RUN** عمل می کند با این تفاوت که امکان دانلود نیز در این مد وجود دارد. این حالت در **CPU** های قدیمی وجود داشت. در **CPU** های جدید اگرچه دکمه **RUN-P** حذف شده ولی عملاً **RUN** مفهوم **RUN-P** را دارد.
HOLD	در این وضعیت پردازش برنامه متوقف شده و می توان برنامه را مرحله به مرحله تست نمود.از این وضعیت بمنظور عیب یابی استفاده می گردد و استفاده از آن در شرایط نرمال خطرناک است

انواع حافظه ها از نوع Load Memory



MMC (Micro Memory Card)

MC (Memory Card)

بخش های حافظه

نواحی مختلف حافظه در CPU های S7

Work Memory

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

نکته:

*به جز در **CPU 417** این حافظه در سایر **CPU** های **S7-300** و **S7-400** قابل توسعه نمی باشد.*

System Memory

در این بخش حافظه، برنامه ذخیره نمی شود بلکه مخصوص ذخیره سازی دیتاهای مختلف است. حافظه ورودی ها(**PII**)، حافظه خروجی ها(**PIQ**)، حافظه تایمرها و کانترها از مهمترین این قسمت ها هستند.
این قسمت از حافظه **CPU** غیر قابل توسعه است.

Retentive Memory

بخش از حافظه **CPU** است که در**CPU** های **S7-300** اگر دیتا در آن قرار گیرد در صورت قطع تغذیه حتی بدون وجود باتری پشتیبان نیز دیتا پاک نمی شود. در**CPU** های **S7-400** پاک شدن یا ماندگار بودن این حافظه بستگی به نوع راه اندازی دارد.

چراغ های روی CPU

CPU S7-300

چراغ ها و نشانگرهای روی CPU

عنوان LED	نوع CPU	وضعیت عادی	وضعیت در زمان فعال شدن	توضیح
STOP	300/400	خاموش	روشن-زرد	CPU در حالت STOP قرار دارد
RUN	300/400	روشن	روشن-سبز	CPU در حالت RUN قرار دارد
FRCE	300/400	خاموش	روشن-زرد	حالت FORCE فعال شده است
DC 5V	300	روشن	روشن-سبز	تغذیه ۵ ولت مربوط به CPU و BUS برقرار است
BATF	300قدیمی	خاموش	روشن-قرمز	خطا در باتری پشتیبان
SF	300	خاموش	روشن-قرمز	وجود خطای سیستمی
INTF	400	خاموش	روشن-قرمز	وجود خطای داخلی
EXTF	400	خاموش	روشن-قرمز	وجود خطای خارجی
BF-BUSF	300/400	خاموش	روشن یا چشمک زن-قرمز	اشکال در BUS شبکه

آشنایی با LED SF در CPUهای زیمنس

- یکی از نمایشگرهایی که بر روی اکثر PLCها تعبیه می شود نمایشگر مربوط به اعلام حالت خطا در سیستم می باشد.
- این نمایشگر تحت عناوین مختلف در PLCهای مختلف قابل روئیت می باشد.
- در PLCهای زیمنس این نمایشگر تحت عنوان SFملاحظه می گردد. رنگ این نمایشگر قرمز بوده و جهت اعلام خطا در سیستم استفاده می شود.
- خطاهایی که منجر به روشن شدن این LEDمی شوند می توانند به صورت نرم افزاری و یا سخت افزاری باشند.
- معمولا در حین کار پروسه، خطاها بیشتر در قالب خطای سخت افزاری رخ می دهند.
- چند مورد از خطاهای سخت افزاری که می تواند منجر به روشن شدن این نمایشگر شود اشاره شده است.

۱-آسیب دیدن ماژول ها

۲-آسیب دیدن کارت حافظه

۳-اعمال وقفه از سایر کارت ها به CPU

۴- خطا در تغذیه

۵-خطا در پیکربندی سخت افزاری کارت ها

آشنایی با LED BF در CPUهای زیمنس

برروی CPUهایی که قابلیت اتصال به شبکه را دارا می باشند نمایشگری نیز جهت اعلام خطا در شبکه تعبیه می شود. این نمایشگر در CPUهای زیمنس با نام BFشناسایی می گردد. رنگ این LEDنیز قرمز بوده و زمانی که خطایی در شبکه رخ دهد روشن می شود. به عنوان مثال فرض کنید که یک CPU 315-2DPموجود می باشد. این CPUبا ۲۰ اسلیو در شبکه پروفیباس در ارتباط می باشد. اگر در این ارتباط مشکلی ایجاد شود این LEDروشن می گردد. در ادامه به چند مورد از خطاهایی که می تواند منجر به روشن شدن این LEDشود اشاره شده است:

۱-خطا در پیکربندی ایستگاه ها در نرم افزار

۲-خطا در آدرس دهی ایستگاه ها

۳-قطعی و یا زدگی در کابل شبکه

۴-آسیب دیدن ماژول های مربوط به ایستگاه های شبکه

۵-صحیح نبودن وضعیت ترمینیتور در باس ارتباطی

۶-خطا در سرکابل شبکه

این نمایشگر به ازای خطاهای مختلف در دو حالت چشمک زن و ثابت قرار می گیرد. روشن شدن این نمایشگر بر روی نمایشگر SFنیز تاثیر گذاشته و در صورت عدم وجود بلوک وقفه مربوطه، CPUبه حالت STOPسوئیچ می گردد.

راه اندازی مجدد CPU

- وقتی **CPU** در مد **RUN** و در حال اجرای سیکل اسکن است، اگر به هر دلیلی **Stop** شود و دوباره **Run** گردد به این حالت راه اندازی مجدد یا **Restart** گفته می شود. مانند قطع و وصل شدن تغذیه برق **CPU**

در این حالت سوالاتی که ممکن است پیش بیاید:

1. آیا برنامه از جایی که قطع شده ادامه یابد یا از ابتدا شروع شود؟

2. آیا مقادیری که تا آن نقطه از برنامه به عنوان نتیجه پردازش بدست آمده بود و در **Work Memory** ذخیره شده بود، پس از راه اندازی مجدد باقی می مانند یا پاک شوند؟ به عنوان مثال اگر یک شمارنده تا عددی شمارش کرده باشد و تغذیه قطع و وصل شود چه مقداری خواهد داشت؟

- برای پاسخ به سوالات فوق لازم است بدانیم که برای **CPU** های **S7** سه نوع راه اندازی مجدد وجود دارد که در هر نوع شرایط متفاوتی ایجاد می شود:

1. *راه اندازی سرد (**Cold Restart**)*
2. *راه اندازی گرم (**Warm Restart**)*
3. *راه اندازی داغ (**Hot Restart**)*

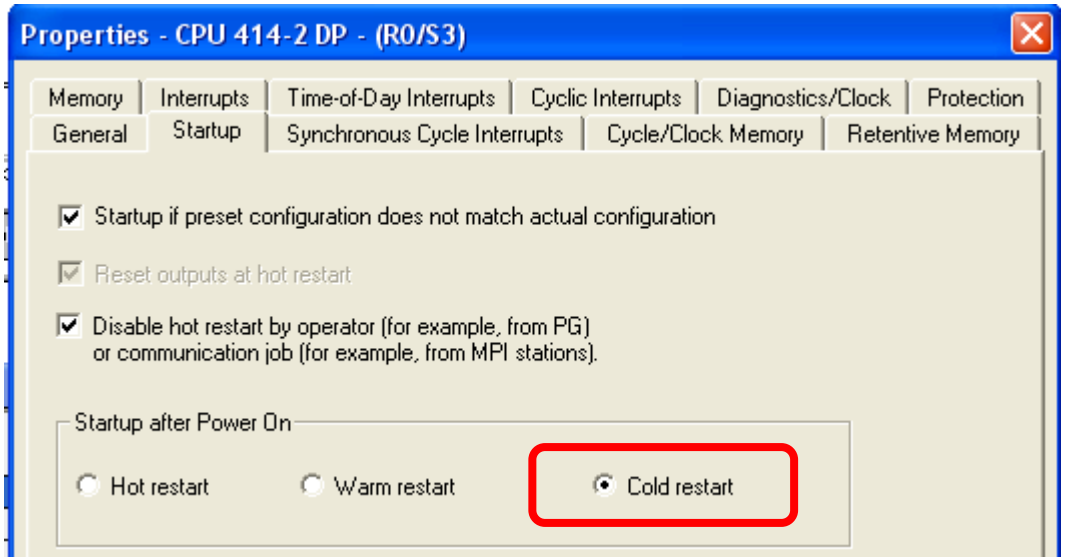
راه اندازی مجدد CPU
Cold Restart

در این نوع راه اندازی:

- اجرای برنامه کاربر از ابتدا آغاز می شود.
- کلیه محتویات **System Memory** چه بصورت ماندگار و چه بصورت غیرماندگیر تعریف شده باشند، اطلاعات خود را از دست می دهند.
- این نوع راه اندازی در همه **CPU** های سری ۴۰۰ پشتیبانی می شود و در **CPU** های سری ۳۰۰ فقط برای **CPU318-2** وجود دارد.
- در این نوع راه اندازی **OB101** توسط **CPU** فراخوانی می شود.

نحوه تنظیم:

با دوبار کلیک بر روی **CPU** در قسمت **HW Config** گزینه های راه اندازی در بخش **Startup** نمایش داده می شود.



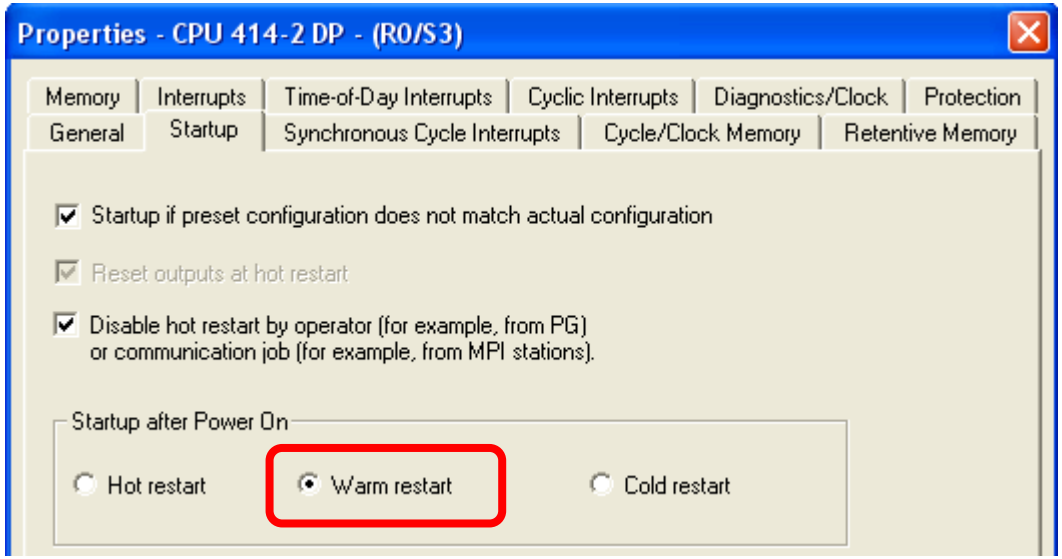
راه اندازی مجدد CPU
Warm Restart

در این نوع راه اندازی:

- **این نوع راه اندازی بصورت پیش فرض برای CPUها تنظیم شده است.**
- **اجرای برنامه کاربر از ابتدا آغاز می شود.**
- **کلیه محتویات System Memory که بصورت غیرماندگیر تعریف شده باشند، اطلاعات خود را از دست می دهند.**
- **در این نوع راه اندازی OB100 توسط CPU فراخوانی می شود.**

نحوه تنظیم:

با دوبار کلیک بر روی CPU در قسمت HW Config گزینه های راه اندازی در بخش Startup نمایش داده می شود.



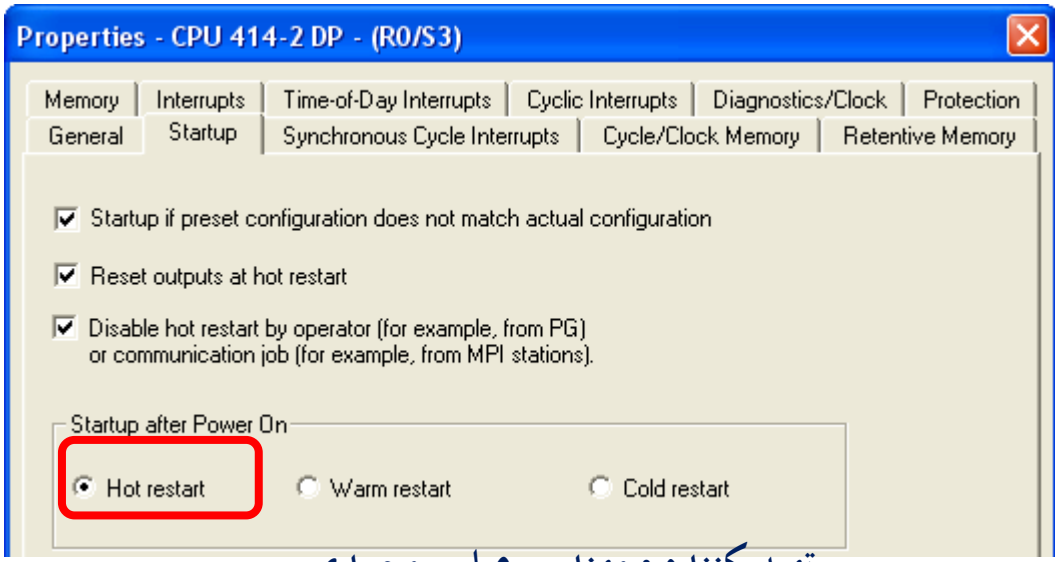
راه اندازی مجدد CPU
Hot Restart

در این نوع راه اندازی:

- اجرای برنامه کاربر از جایی که قطع شده بود ادامه پیدا می کند.
- کلیه محتویات **System Memory** چه بصورت ماندگار و چه غیرماندگیر تعریف شده باشند، اطلاعات خود را حفظ می کنند.
- به منظور انجام این نوع راه اندازی، **CPU** باید دارای باطری پشتیبان باشد.
- این نوع راه اندازی فقط مخصوص **CPU**های سری ۴۰۰ است.
- در این نوع راه اندازی **OB102** توسط **CPU** فراخوانی می شود.

نحوه تنظیم:

با دوبار کلیک بر روی **CPU** در قسمت **HW Config** گزینه های راه اندازی در بخش **Startup** نمایش داده می شود.



ریست کردن CPU

- **Reset کردن CPU یعنی پاک کردن حافظه CPU، یعنی هم مقادیر متغیرهای حافظه و هم برنامه ای که توسط کاربر به حافظه ارسال شده پاک شود.**

- **وقتی CPU ریست می شود، چه اتفاقی می افتد:**

1. **برنامه ای که در حافظه Load Memory و Work Memory وجود دارد، پاک کی شود.**
2. **مقادیر متغیرهای System Memory نیز پاک می شوند.**
3. **اگر کارت حافظه از نوع RAM وجود داشته باشد، محتویات آن نیز پاک می شود ولی اگر حافظه ROM یا Flash EPROM باشد با ریست پاک نمی شود.**

نکته:

Reset و Cold Restart باهم متفاوت هستند:

در Cold Restart حافظه های Load و Work پاک نمی شوند و فقط System Memory پاک می گردد.

- **به دو روش می توان CPU را Reset نمود:**

1. **Reset با سوئیچ MRES**
2. **Reset از طریق نرم افزار**

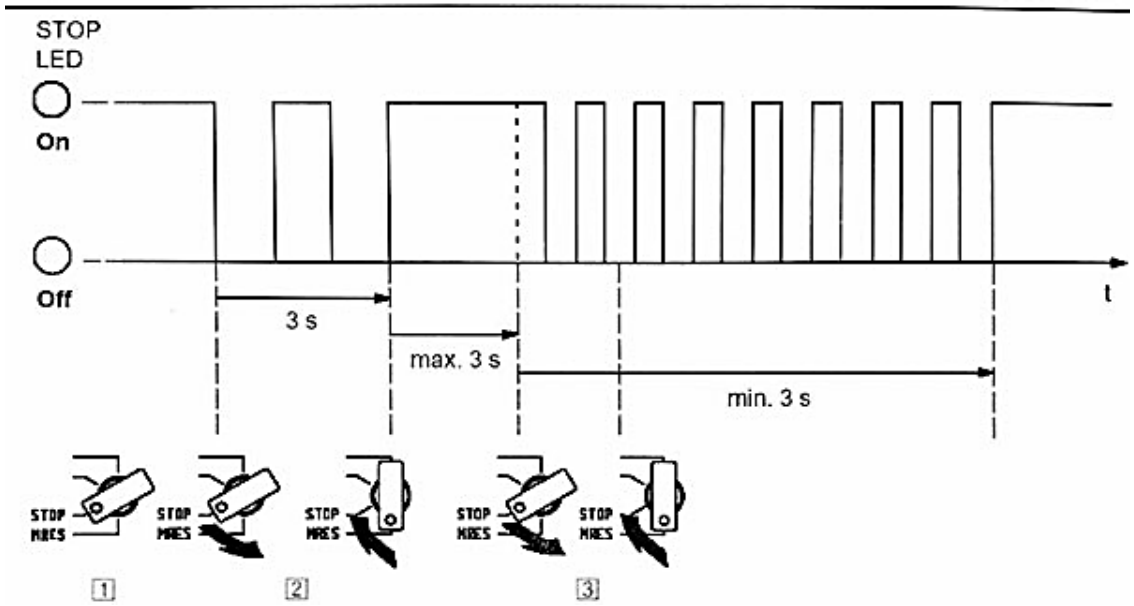
ریست کردن CPU
Reset با سوییچ MRES

برای ریست کردن CPU باید سوییچ طبق سیکل زیر بین وضعیت MRES و Stop جابه جا شود:

1. سوییچ در وضعیت **Stop** است و **LED** مربوط به **STOP** روشن است.
2. سوییچ را از **Stop** به **MRES** می بریم و ۳ ثانیه نگه می داریم، در طول ۳ ثانیه **LED** مربوط به **Stop** چشمک زن است، با گذشت ۳ ثانیه **LED** مربوط به **Stop** ثابت می شود.
3. وقتی **LED** مربوط به **Stop** ثابت شد، سوییچ را از **MRES** به **Stop** برمی گردانیم و با مکث کوتاهی (حداکثر ۳ ثانیه) سوییچ را از **Stop** به **MRES** می بریم که در این حالت **LED** فوق به حالت چشمک زن سریع در می آید.

نکته:

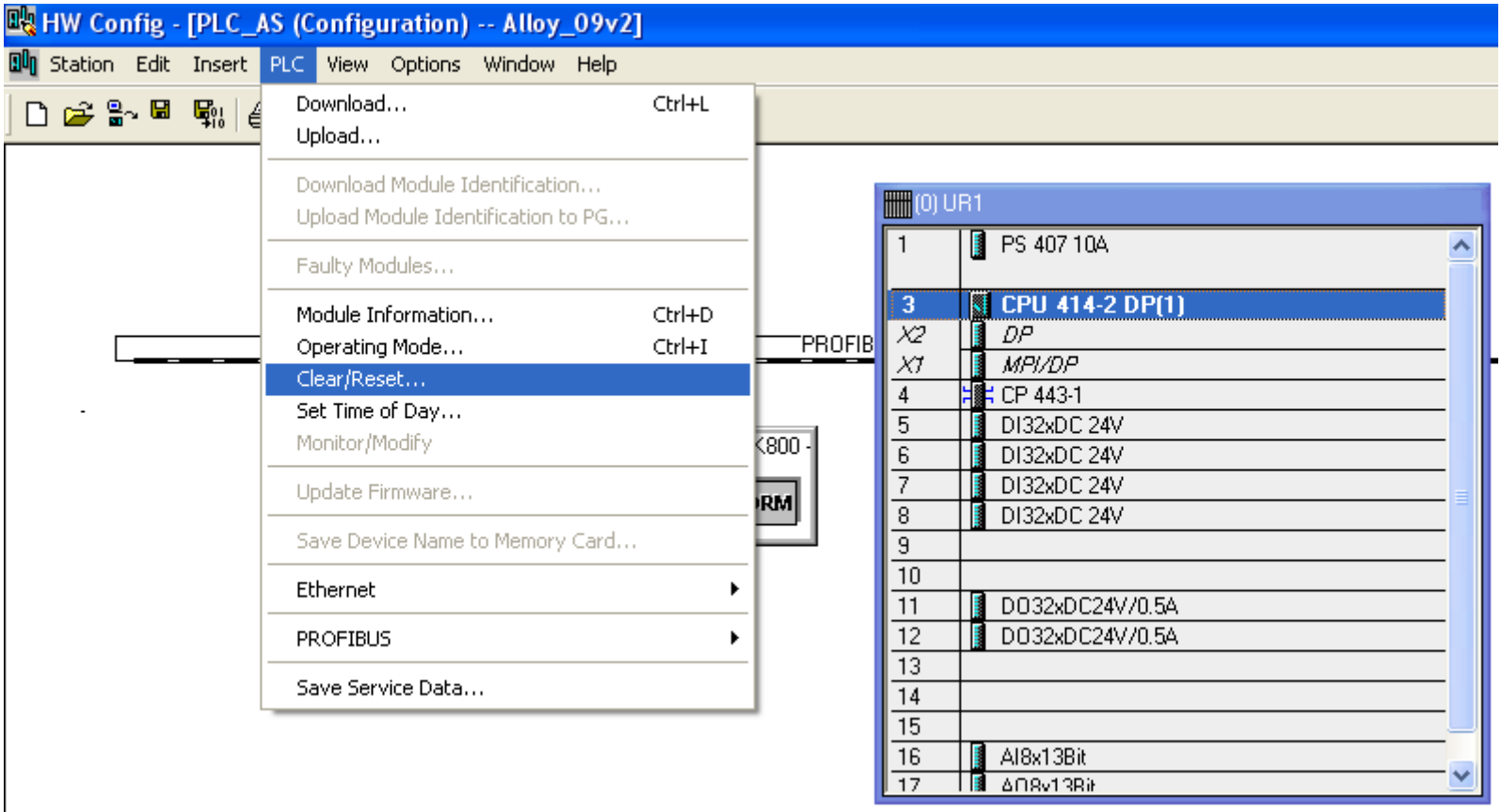
- از این روش نمی توان برای **CPU**های جدید **S7-300** و **S7-400** استفاده نمود. چرا؟



ریست کردن CPU
Reset از طریق نرم افزار

می توان از طریق نرم افزار **STEP7** و یا زیربرنامه های آن عملیات ریست را انجام داد:

1. در **HW Config** ابتدا **CPU** را انتخاب می کنیم.
2. از منوی **PLC-> Clear / Reset**، کلیک می کنیم.



فصل سوم

مدار منطقی و منطق دیجیتال

تبدیل دسیمال به باینری و برعکس

تبدیل دسیمال به مبنای ۸ (اکتال) و برعکس

تبدیل دسیمال به مبنای ۱۶(هگزادیسمال) و برعکس

تبدیل دسیمال به کد **BCD**

منطق دیجیتال

منطق دیجیتال :

در مدار منطقی دیجیتال از المان های الکترونیکی نظیر دیود و ترانزیستور استفاده می شود . از ترکیب چند المان توابع منطقی ایجاد شده که هرکدام منطق خاصی را پیروی می کنند . در این مدارها از دو اصطلاح صفر و یک بسیار استفاده می کنند ، مفهوم این دو اصطلاح بدین شرح است : در یک سیستم تنها چیزی که برای المان های الکتریکی قابل فهم است ، بود یا نبود ولتاژ است چون منطق دیجیتال از این خاصیت تبعیت می کند پس باید دو سطح از ولتاژ را برای درک سیستم تعریف نمود مثل 0 ولت و 24 ولت . در این سیستم سطح ولتاژ 24 ولت ، یک و سطح ولتاژ 0 ولت ، صفر تلقی می شود .

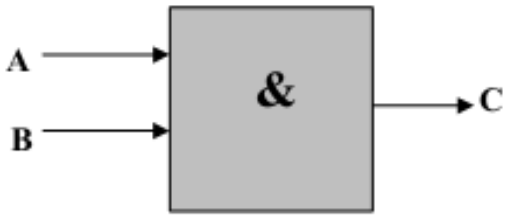
توابع منطقی دیجیتال دارای یک یا چند ورودی و یک خروجی می باشند که وضعیت خروجی متناسب با وضعیت ورودی می باشد . در مدارهای منطقی یا دیجیتال عناصری وجود دارد که توانایی انجام عملیات بر روی صفر و یکها را دارند که به آنها گیت (Gate) می گویند . هفت گیت منطقی دیجیتال موجود می باشد : AND ، NOT ، NAND ، OR ، NOR ، XOR ، XNOR

گیت منطقی AND :

در این تابع خروجی فقط زمانی که تمام ورودیها در وضعیت یک قرار دارند یک می شود . عملکرد این تابع مثل تیغه های باز سری می باشد .

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

جدول درستی



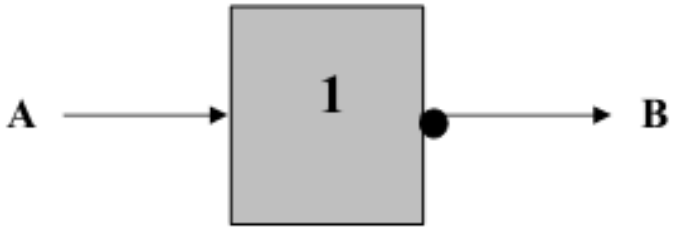
منطق دیجیتال

گیت منطقی NOT :

این تابع فقط یک ورودی دارد و همیشه وضعیت خروجی عکس وضعیت ورودی است . این تابع معادل کنتاکت بسته در مدار فرمان می باشد .

A	B
0	1
1	0

جدول درستی

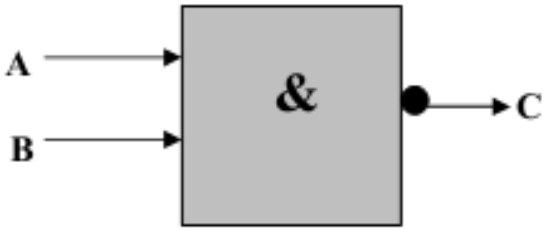


گیت منطقی NAND :

خروجی این تابع فقط زمانی که همه ورودیها یک باشند در وضعیت صفر قرار می گیرد . در حقیقت این تابع ، عکس تابع منطقی AND عمل می کند .

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

جدول درستی



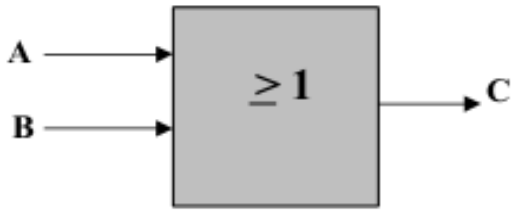
منطق دیجیتال

گیت منطقی OR :

خروجی این تابع فقط زمانی که همه ورودی ها صفر باشند در وضعیت صفر قرار می گیرد . این تابع معادل تیغه های باز موازی می باشد .

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

جدول درستی

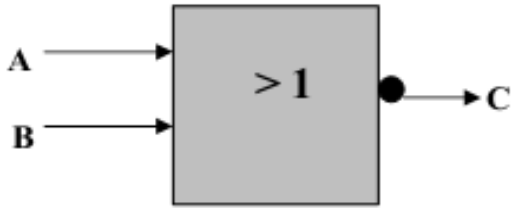


گیت منطقی NOR :

خروجی این تابع فقط زمانی که همه ورودی ها صفر باشند در وضعیت یک قرار می گیرد . در حقیقت این تابع ، عکس تابع منطقی OR عمل می کند .

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

جدول درستی

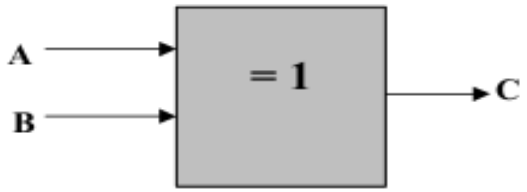


منطق دیجیتال

گیت منطقی XOR :

خروجی این تابع فقط زمانی که تنها یکی از ورودی ها یک باشد در وضعیت یک قرار می گیرد . این تابع همانند مدار کلید تبدیل عمل می کند . اصطلاحا به این تابع ، گیت فرد می گویند .

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

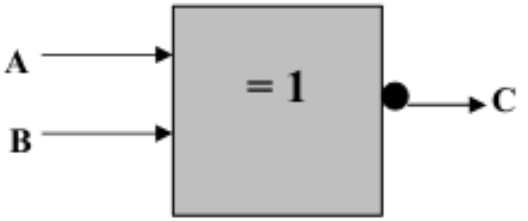


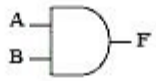
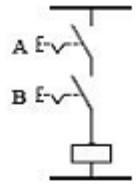
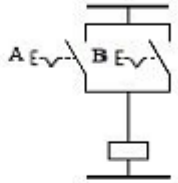
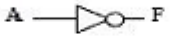
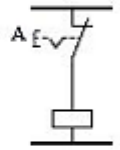
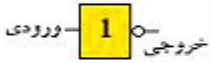
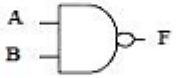
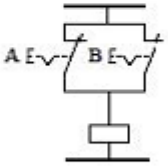
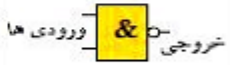
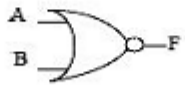
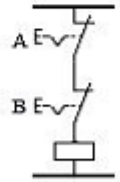
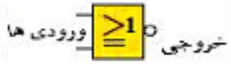
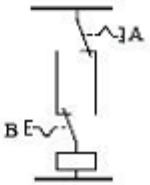
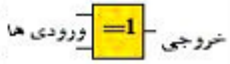
گیت منطقی XNOR :

خروجی این تابع فقط زمانی که دو ورودی در یک وضعیت باشند در وضعیت یک قرار می گیرد . در حقیقت این تابع ، عکس تابع منطقی XOR عمل می کند . اصطلاحا به این تابع ، گیت زوج می گویند .

A	B	C
0	0	1
0	1	0
1	0	0
1	1	1

جدول درستی



عملگر منطقی	جدول صحت	علامت اختصاری	مدار کلیدی	شکل بلوکی
AND	خروجی / ورودی ها A B F 0 0 0 0 1 0 1 0 0 1 1 1			
OR	خروجی / ورودی ها A B F 0 0 0 0 1 1 1 0 1 1 1 1			
NOT	A F 0 1 1 0			
NAND	خروجی / ورودی ها A B F 0 0 1 0 1 1 1 0 1 1 1 0			
NOR	خروجی / ورودی ها A B F 0 0 1 0 1 0 1 0 0 1 1 0			
XOR	خروجی / ورودی ها A B F 0 0 0 0 1 1 1 0 1 1 1 0			

تهیه کننده : مهندس عباس محمدی

منطق دیجیتال

مفهوم بیت :

با ترکیب چند تابع منطقی سلول حافظه تشکیل می شود ، این بدان معنی است که وضعیت صفر و یا یک بودن ورودی یا خروجی را در خود حفظ می کند ، به این سلول حافظه یک بیت گفته می شود .

اعداد را می توان در مبناهای عددی مختلف نمایش داد . آشناترین مبنای اعداد ، مبنای ده می باشد . در مبنای ده کلیه اعداد با ترکیبی از اعداد 0 تا 9 حاصل می گردند . از دیگر مبناهای عددی رایج می توان به مبنای دو اشاره نمود ، همانند اعداد مبنای ده هر رقم یک عدد در مبنای دو دارای ارزش خاصی می باشد . در این مبنا تنها اعداد صفر و یک موجود می باشند ، مثلا عدد 01101 یک عدد پنج رقمی در مبنای دو می باشد . هر رقم در مبنای دو را یک بیت و هر هشت بیت را یک بایت و هر دو بایت را یک کلمه می نامند . جهت بدست آوردن معادل مبنای دو یک عدد دهدهی این عدد را بطور متناوب بر دو تقسیم می کنیم تا جاییکه خارج قسمت نهایی بر دو قابل تقسیم نباشد ، باقیمانده های بدست آمده را از انتها به ابتدا به ترتیب از چپ به راست بعد از آخرین خارج قسمت می نویسیم و اینگونه معادل دودویی اعداد بدست می آید :

$$(41)_{10} = (\ ?\ )_{2}$$

41 ÷ 2 ⟶ خارج قسمت 20 و باقیمانده 1

20 ÷ 2 ⟶ خارج قسمت 10 و باقیمانده 0

10 ÷ 2 ⟶ خارج قسمت 5 و باقیمانده 0

5 ÷ 2 ⟶ خارج قسمت 2 و باقیمانده 1

2 ÷ 2 ⟶ خارج قسمت 1 و باقیمانده 0

چون خارج قسمت بر دو بخش پذیر نیست لذا طبق روش گفته شده معادل باینری عدد را می نویسیم :

$$(41)_{10} = (101001)_{2}$$

منطق دیجیتال

جهت تبدیل یک عدد از مبنای دو به مبنای ده می توان هر رقم را در ارزش مکانی خود ضرب نمود و سپس حاصلضربهای بدست آمده را با هم جمع نمود :

$$(\ 101001\ )_{2} = (\ ?\ )_{10}$$

$$101001 \longrightarrow 1\times 32 + 0\times 16 + 1\times 8 + 0\times 4 + 0\times 2 + 1\times 1 = 41$$

منطق دیجیتال

مبنای شانزده	مبنای دو	مبنای ده
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

جهت تبدیل یک عدد دودویی به عدد مبنای شانزده کافی است از سمت راست اعداد را چهار رقم چهار رقم جدا نموده و سپس معادل مبنای شانزده آنها را جایگزین نماییم .

$$(10001101)_2 = (\ ?\ )_{16}$$

$$1000\ ,\ 1101 \Longrightarrow 8D$$

$$(10001101)_2 = (8D)_{16}$$

جهت تبدیل یک عدد از مبنای شانزده به مبنای دو به جای هر عدد معادل دودویی چهار رقمی آن را جایگزین می کنیم .

$$(A3B)_{16} = (\ ?\ )_2$$

$$1010\ ,\ 0011\ ,\ 1011 \Longrightarrow A\ ,\ 3\ ,\ B$$

$$(A3B)_{16} = (101000111011)_2$$

تهیه کننده : مهندس عباس محمدی

منطق دیجیتال

جهت تبدیل یک عدد دودویی به عدد مبنای هشت کافی است از سمت راست اعداد را سه رقم سه رقم جدا نموده و سپس معادل مبنای هشت آنها را جایگزین نماییم .

مبنای هشت	مبنای دو
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

$$(001101)_2 = (\ ?\ )_8$$

$$001\ ,\ 101 \Longrightarrow 1\ ,\ 5$$

$$(001101)_2 = (\ 15\ )_8$$

جهت تبدیل یک عدد از مبنای هشت به مبنای دو به جای هر عدد معادل دودویی سه رقمی آن را جایگزین می کنیم .

$$(23)_8 = (\ ?\ )_2$$

$$010\ ,\ 011 \Longrightarrow 2\ ,\ 3$$

$$(23)_8 = (\ 010011\ )_2$$

منطق دیجیتال

کد BCD : در BCD هر رقم در مبنای دهدهی بطور جداگانه به شکل دودویی کد می شود . هر رقم در چهار بیت کد می شود ، چون بزرگترین رقم دسیمال یعنی 9 در باینری چهار رقمی است .

اعداد صحیح Integer : INT عدد صحیح شانزده بیتی می باشد ، بیت پانزدهم نشان دهنده علامت عدد است . اگر صفر باشد عدد مثبت و اگر یک باشد عدد منفی می باشد . بازه این اعداد بین 32768 – تا 32767 + می باشد .

مکمل یک :

بدین صورت بدست می آید که کافیست تمام بیت های عدد موردنظر را NOT کنیم . به مثال زیر توجه کنید :

مکمل یک عدد 1001101 برابر است با 0110010

مکمل دو :

بدین صورت بدست می آید که کافیست تمام بیت های بعد از اولین بیت یک از سمت راست عدد را NOT کنیم . به مثال زیر توجه کنید :

مکمل دو عدد 1010010 برابر است با 0101110

مدارهای ترتیبی و ترکیبی

- مدارات AND و OR مدارات ترکیبی هستند چون حافظه ندارند و خروجی مدار انها در هر مرحله وابسته به ورودی در همان لحظه است.

- مدارات دیگری هستند که بدلیل داشتن عنصر حافظه می توانند حالت قبلی را ذخیره کنند مانند مدارت فلیپ فلاپ ها :

SR , RS

مدارهای ترتیبی
فلیپ فلاپ

دستور فلیپ فلاپ های RS و SR :

فلیپ فلاپ RS عمل ست و ریست را وقتی که $RLO = 1$ باشد انجام می دهد بنا براین اگر $RLO = 0$ باشد این دستور اجرا نمی شود .عمل ریست کردن بیت آدرس داده شده وقتی اتفاق می افتد که ورودی $R = 1$ باشد و ورودی $S = 0$ باشد . عمل ست کردن بیت آدرس داده شده وقتی اتفاق می افتد که $R = 0$ و ورودی $S = 1$ باشد.**اگر هر دو ورودی یک شوند عمل ست انجام می شود .**

فلیپ فلاپ SR عمل ست و ریست را وقتی که $RLO = 1$ باشد انجام می دهد بنا براین اگر $RLO = 0$ باشد این دستور اجرا نمی شود .عمل ست کردن بیت آدرس داده شده وقتی اتفاق می افتد که ورودی $R = 0$ باشد و ورودی $S = 1$ باشد . عمل ریست کردن بیت آدرس داده شده وقتی اتفاق می افتد که $R = 1$ و ورودی $S = 0$ باشد.**اگر هر دو ورودی یک شوند عمل ریست انجام می شود .**

فصل چهارم

برنامه نویسی PLC S7 با نرم افزار SIMATIC STEP 7

انواع داده و متغیرهای IEC

نوع داده	فرمت	مثال
ورودی بیتی	Ix.x	I0.3
ورودی بایتی	IBx	IB4
ورودی ورد	IWx	IW20
ورودی دبل ورد	IDx	ID22
خروجی بیتی	Qx.x	Q0.3
خروجی بایتی	QBx	QB4
خروجی ورد	QWx	QW20
خروجی دبل ورد	QDx	QD22
حافظه بیتی	Mx.x	M0.3
حافظه بایتی	MBx	MB4
حافظه ورد	MWx	MW20
حافظه دبل ورد	MDx	MD22

انواع داده و متغیرهای IEC

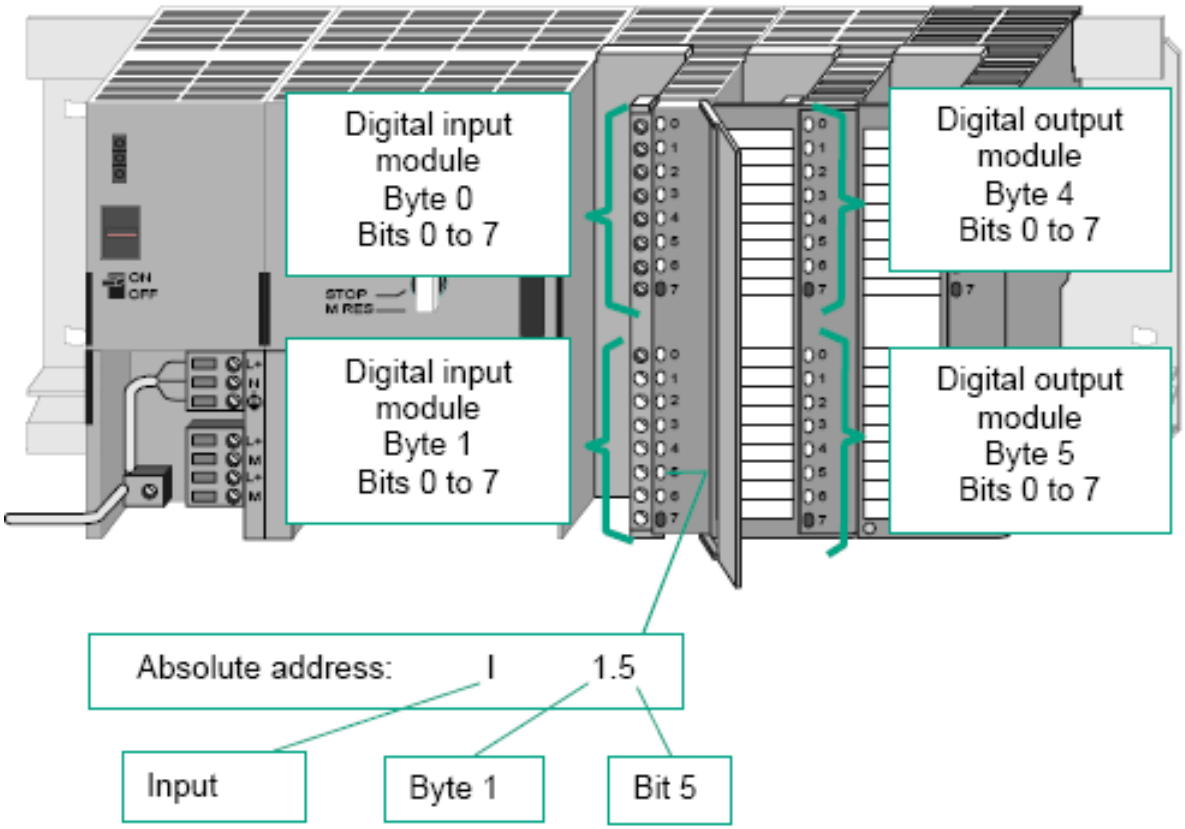
نوع داده	فرمت	مثال
دیتا بلاک بیتی	DBxx.x	DBx0.3
دیتا بلاک بایتی	DBBx	DBB4
دیتا بلاک ورد	DBWx	DBW20
دیتا بلاک ورد	DBDx	DBD22
لود بیتی	Lx.x	L0.3
لود بایتی	LBx	LB4
لود ورد	LWx	LW20
لود دبل ورد	LDx	LD22
ورودی جانبی بایتی	PIBx	PIB40
ورودی جانبی ورد	PIWx	PIW50
ورودی جانبی دبل ورد	PIDx	PID60

انواع داده و متغیرهای IEC

نوع داده	فرمت	مثال
خروجی جانبی بایتی	PQBx	PQB40
خروجی جانبی ورد	PQWx	PQW50
خروجی جانبی دبل ورد	PQDx	PQD60
تایمر	Tx	T4
کانتر	Cx	C2

آدرس دهی

آدرس دهی مطلق (Absolute Address)
آدرس دهی سمبولیک (Symbolic Address)



مقدمات برنامه نویسی

زبانهای برنامه نویسی در استاندارد IEC 1131-3 :

۱- LD (ladder diagram) ، مناسب برای کسانی که با دیاگرام رله ای آشنایی دارند .

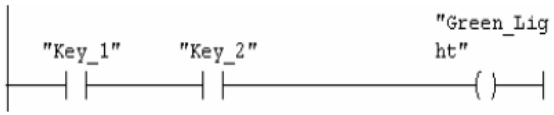
۲- FBD (function black diagram) ، // با دیاگرام مدار منطقی آشنایی دارند .

۳- SFC (sequential fanctial chart) ، // روش گرافیکی برای کنترل ترتیبی

۴- IL (instruction list) ، // با زبان اسمبلی آشنایی دارند .

۵- ST (structured text) ، // برنامه نویسی سطح بالا مانند C و پاسکال

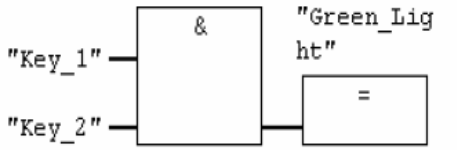
Ladder Logic (LAD)
Suitable for users from the electrical engineering industry, for example.



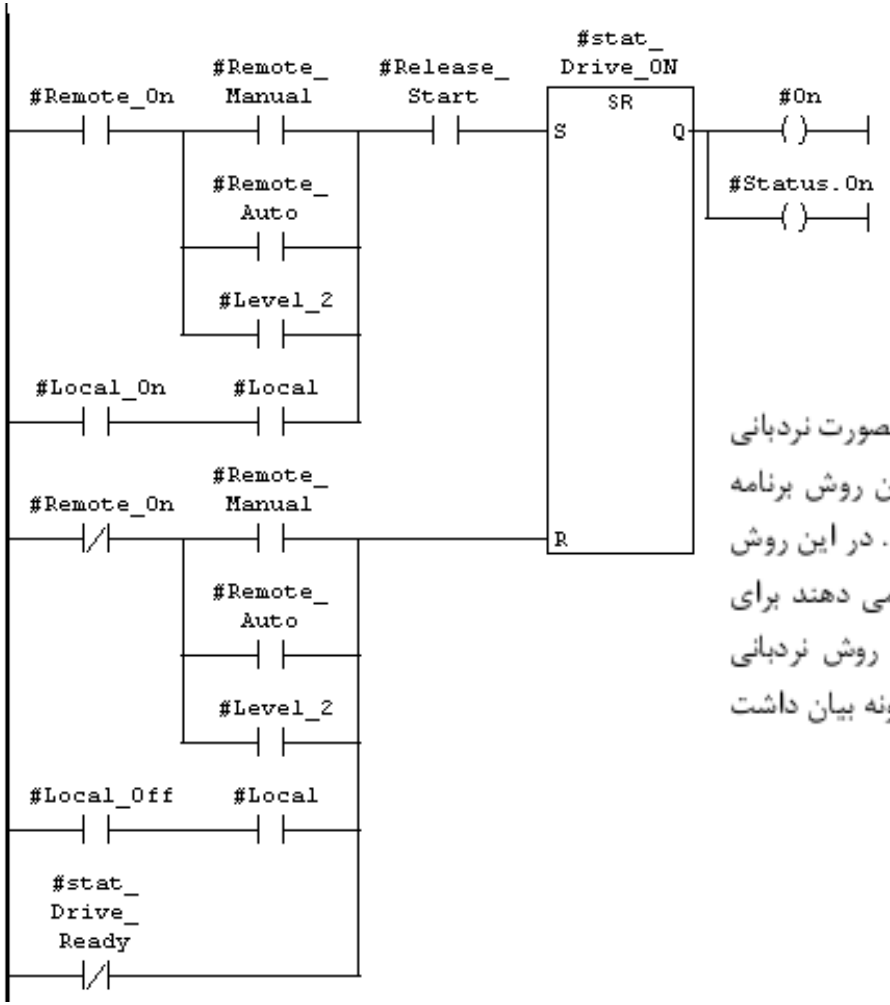
Statement List (STL)
Suitable for users from the world of computer technology, for example.

```
A     "Key_1"
A     "Key_2"
=     "Green_Light"
```

Function Block Diagram (FBD)
Suitable for users from the world of circuit engineering, for example.



زبان LAD



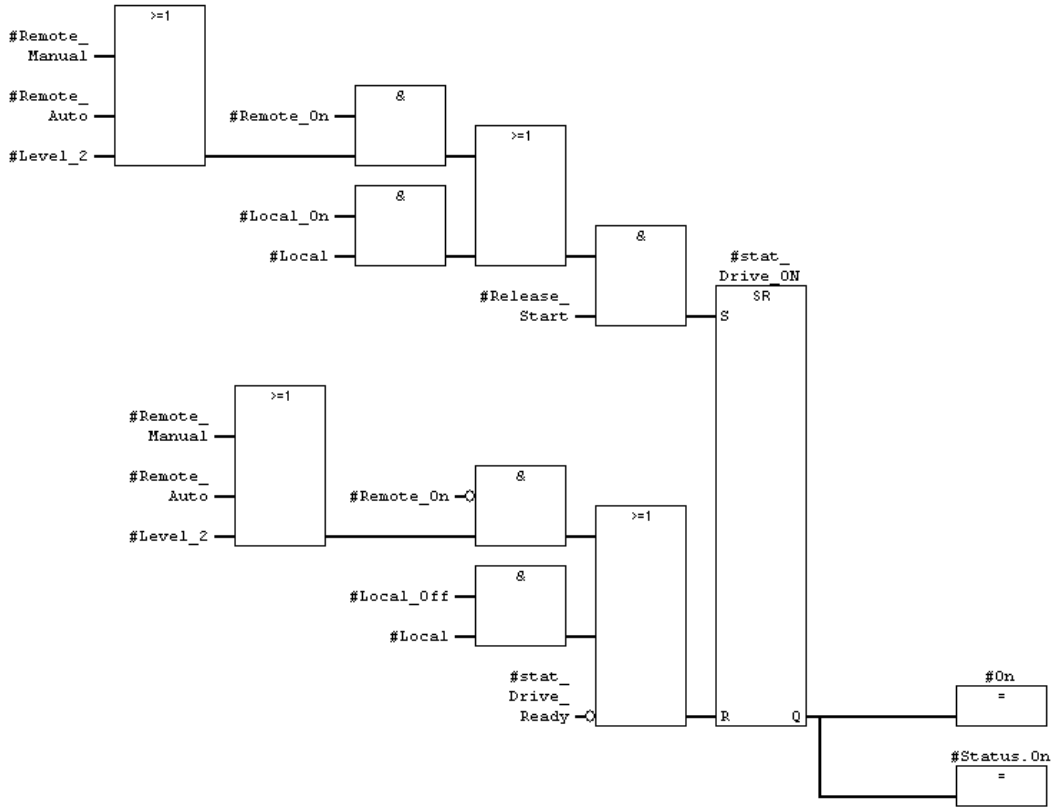
برنامه نویسی به روش نردبانی LAD :

از آنجا که تمام نقشه های کنترل و فرمان منطقی قبل از ظهور PLC ها بصورت نردبانی و یا چیزی شبیه به آن تهیه و طراحی می شد ، لذا سازندگان PLC این روش برنامه نویسی را بعنوان یکی از روش های ممکن برنامه نویسی انتخاب نمودند . در این روش آن دسته از عناصر نردبان که تابع یا عمل خاص و پیچیده ای را انجام می دهند برای سهولت با یک جعبه نمایش داده می شوند . دستورات نوشته شده به روش نردبانی بترتیب از چپ به راست و از بالا به پایین انجام می گردند . میتوان اینگونه بیان داشت که این زبان براساس نقشه های مدار فرمان ، طراحی شده است .

زبان FBD

برنامه نویسی به روش FBD :

در این روش برنامه بصورت بلوکی نوشته شده که در آن هر بلوک بیانگر یک عملگر می باشد ، بدین ترتیب برنامه های نوشته شده به روش FBD عبارتند از یک سری جعبه که به یکدیگر متصل گردیده اند . این روش معمولا بطور مستقل کاربرد چندانی ندارد و اغلب برای عیب یابی و یا شناخت منطق کنترل سیستم بسیار مفید است . این زبان براساس مدارهای الکترونیک و دیجیتال طراحی شده است .



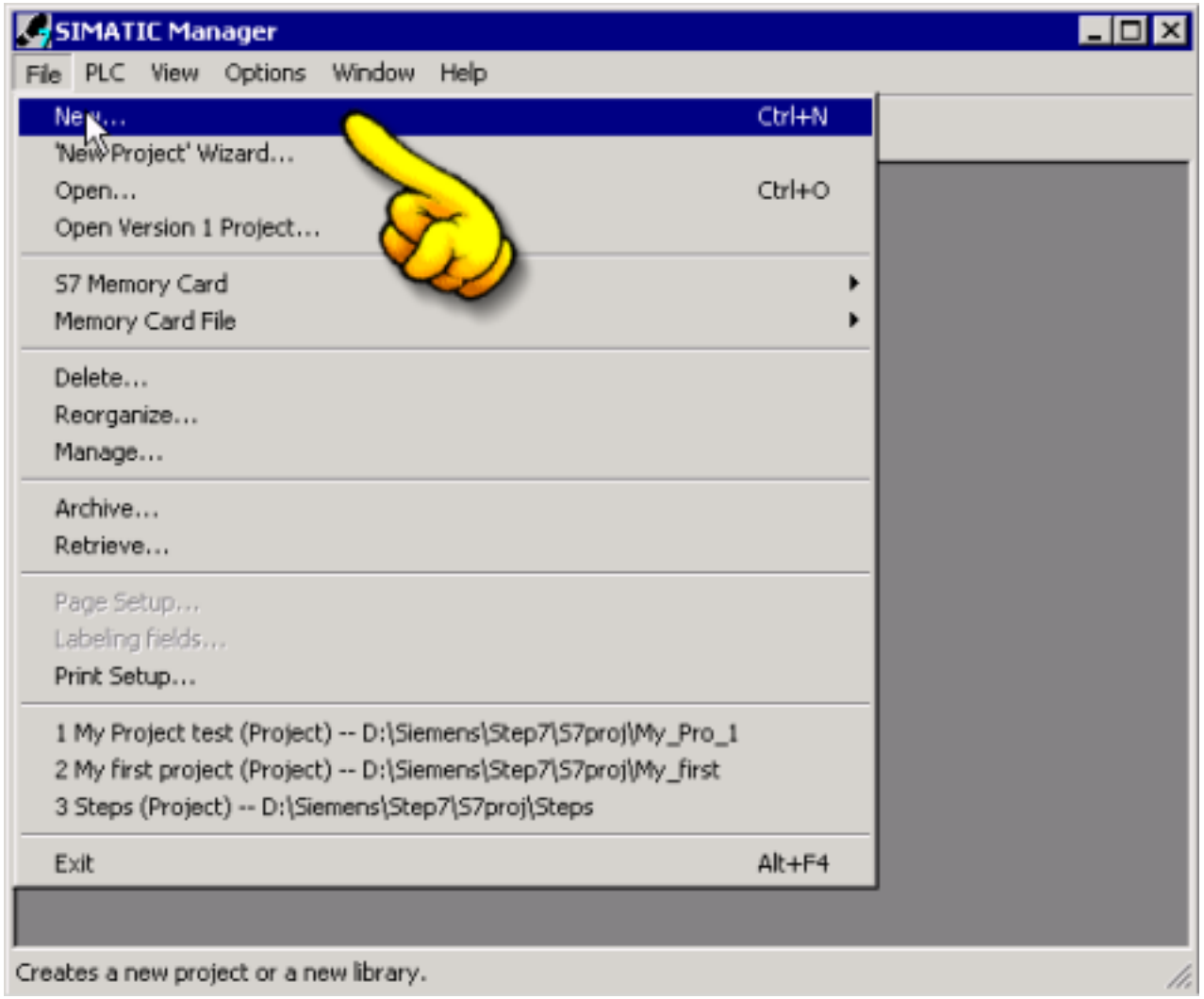
زبان STL

برنامه نویسی به روش لیست جملات STL :

در این روش هر عمل منطقی توسط یک جمله یا عبارت مناسب نوشته می شود . نکته قابل توجه در این روش برنامه نویسی آن است که هر PLC دارای کد دستورات منحصر بفردی می باشد که این دستورات به نوع CPU بکار رفته بستگی دارد . این زبان براساس زبان برنامه نویسی کامپیوتر ایجاد شده است . زبان برنامه نویسی در حالت STL مثل زبان بیسیک یا اسمبلی بوده و نوشتاری است . روش STL نیازهای گرافیکی بسیار کمتری نسبت به دو روش قبل دارد ، لذا نوع و تعداد دستورات قابل درک و اجرا در این روش بیشتر از روش های LAD و FBD می باشد . به همین دلیل برنامه هایی که به روش LAD یا FBD نوشته می شود معمولا قابل تبدیل به STL می باشد ، درحالیکه عکس این قضیه همواره ممکن نیست . در برنامه نویسی به روش STL هر چند خط برنامه که عمل خاصی را انجام می دهد یک Segment می گویند .

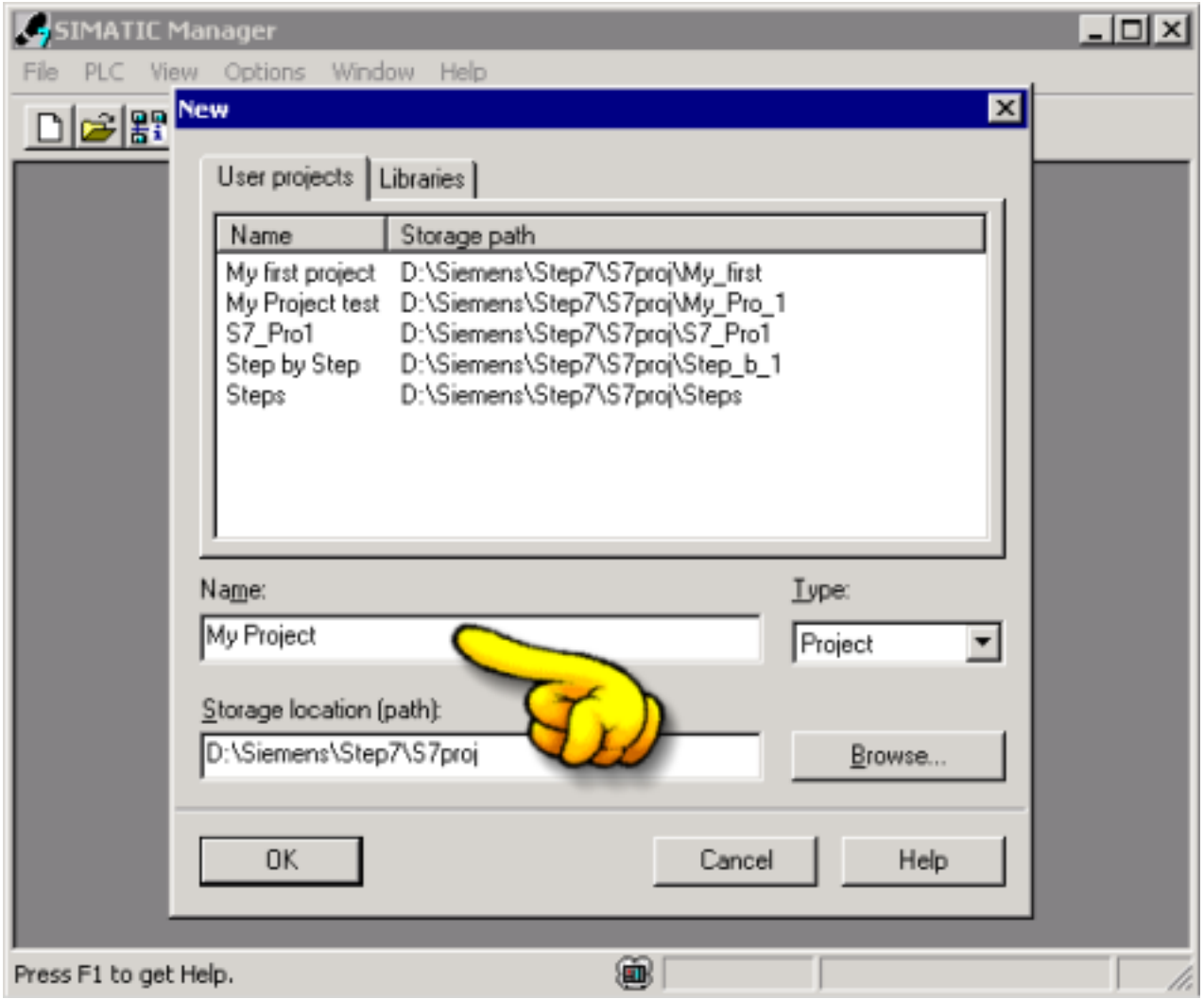
```
A(
A     #Remote_On
A(
O     #Remote_Manual
O     #Remote_Auto
O     #Level_2
)
O
A     #Local_On
A     #Local
)
A     #Release_Start
S     #stat_Drive_ON
A(
AN    #Remote_On
A(
O     #Remote_Manual
O     #Remote_Auto
O     #Level_2
)
O
A     #Local_Off
A     #Local
ON    #stat_Drive_Ready
)
R     #stat_Drive_ON
A     #stat_Drive_ON
=     #On
=     #Status.On
```

ایجاد و پیکربندی یک پروژه در محیط Simatic manager



۱- از منوی **File** گزینه **New...** را برای ایجاد یک پروژه جدید کلیک می کنیم.

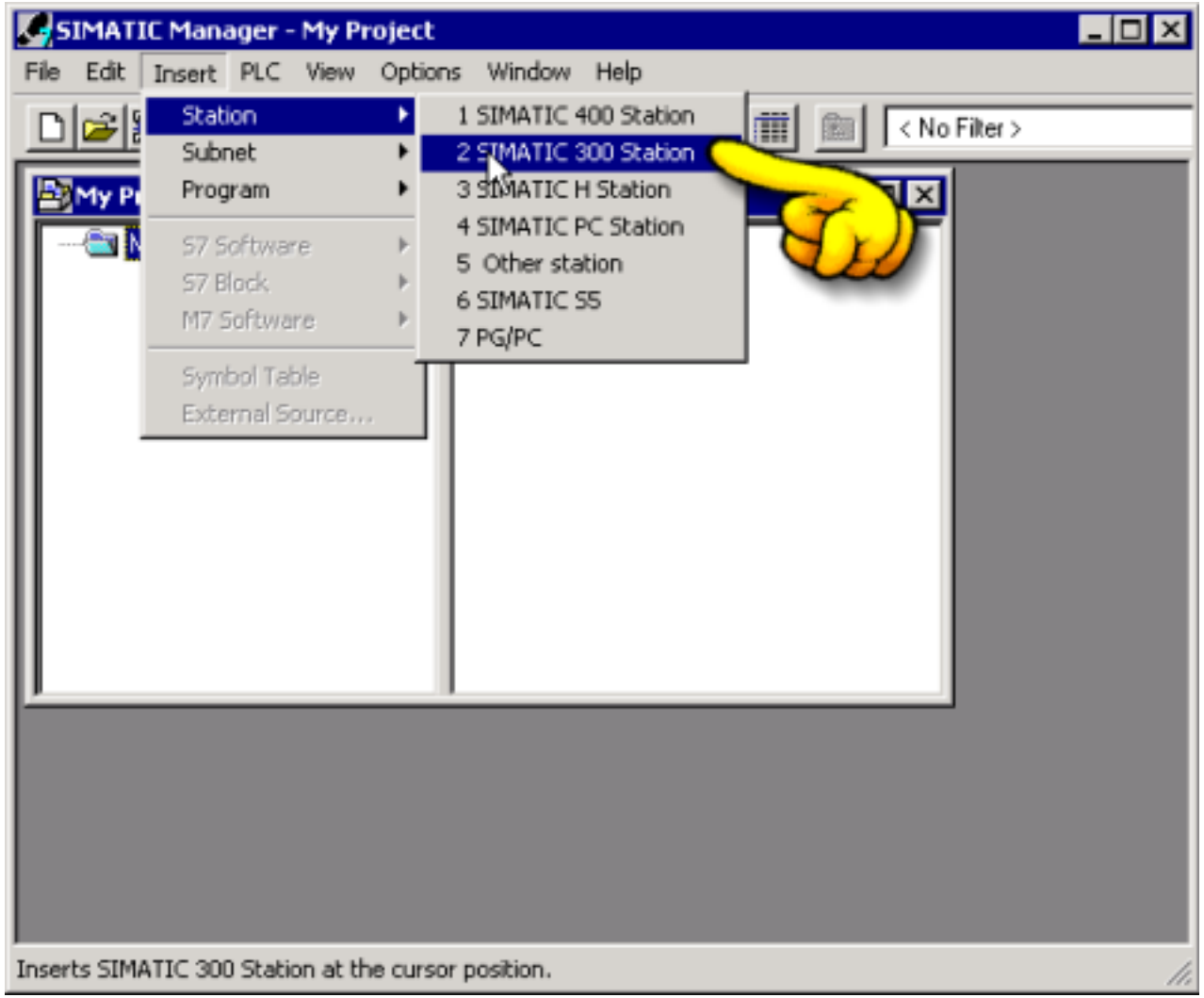
ایجاد و پیکربندی یک پروژه در محیط Simatic manager



۲- در کادر بازشده نام پروژه جدید و مسیر ذخیره آن را می نویسیم.

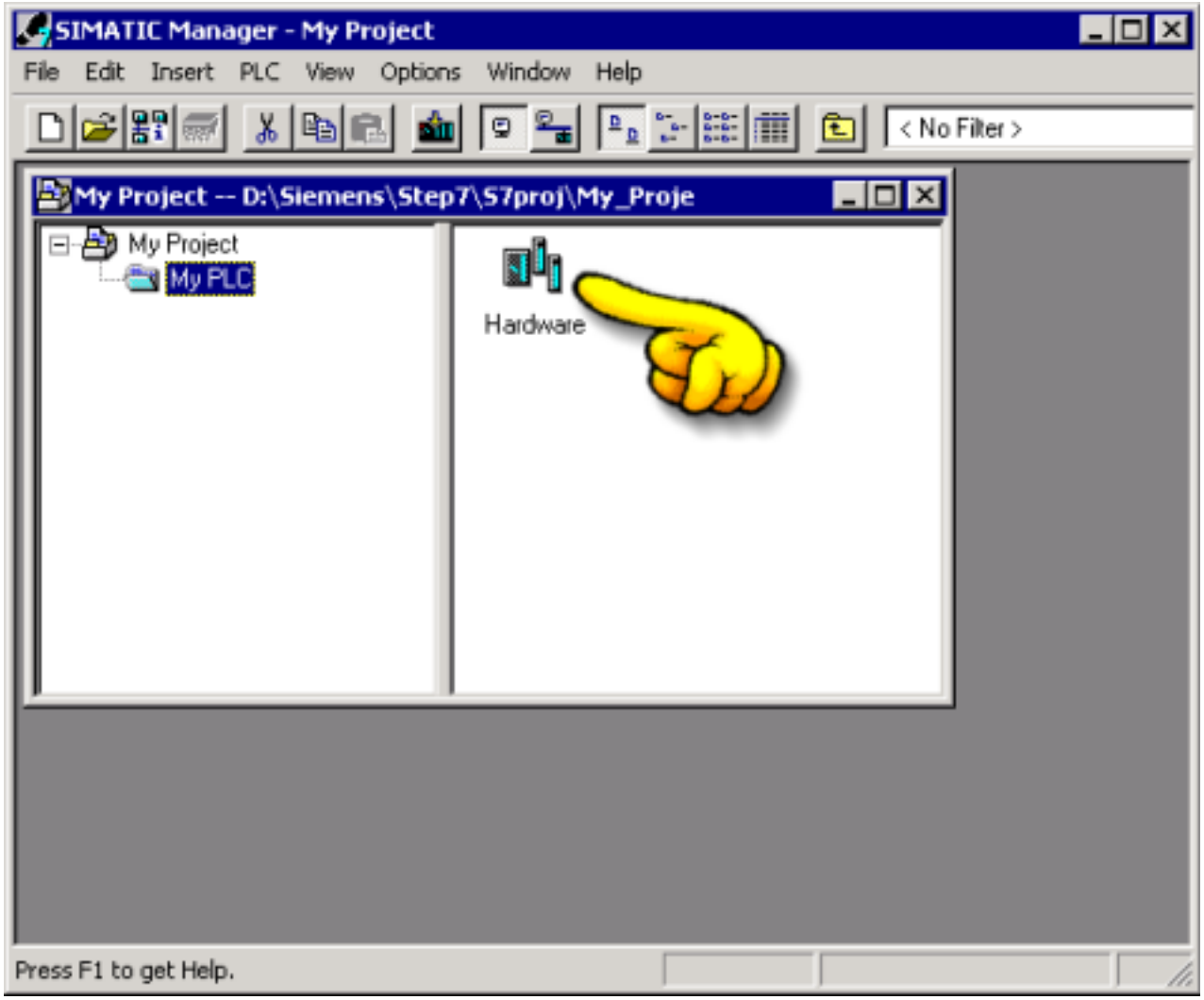
پیکربندی سخت افزار

ایجاد و پیکربندی یک پروژه در محیط Simatic manager



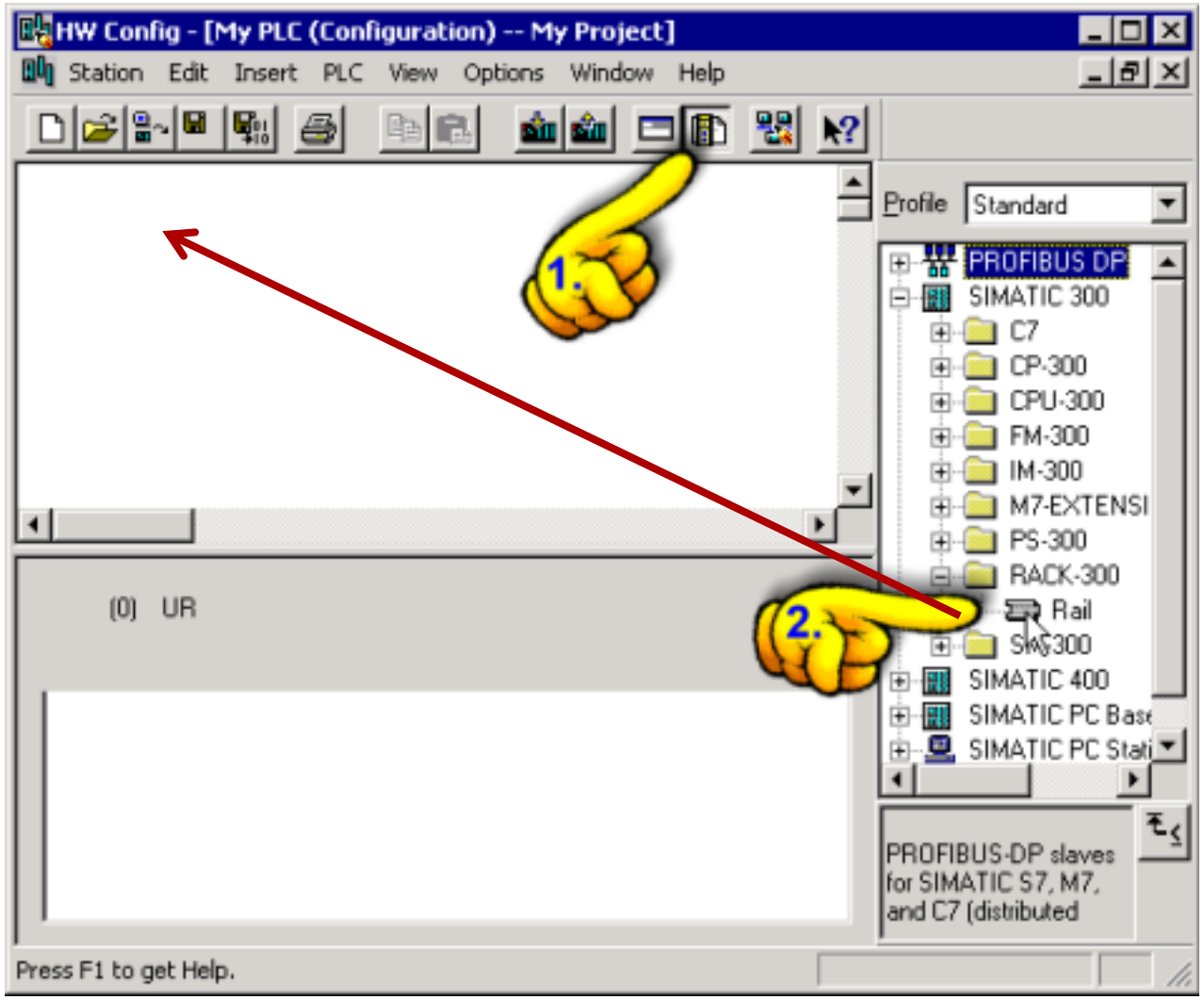
۳-از مسیر فوق گزینه SIMATIC 300 Stationرا برای ایجاد یک ایستگاه از نوع ۳۰۰ کلیک می کنیم.

ایجاد و پیکربندی یک پروژه در محیط Simatic manager



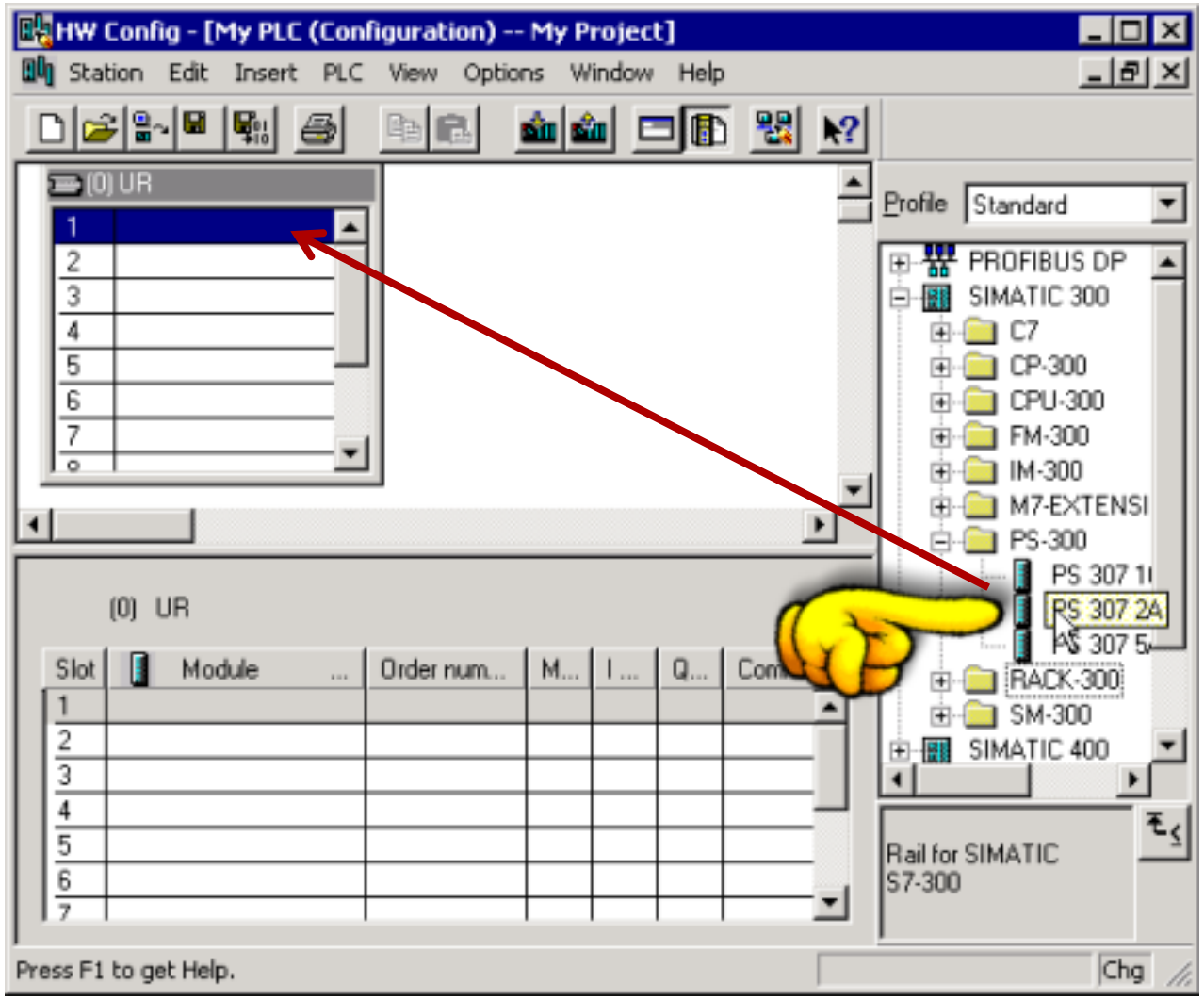
۴- در پنجره فوق روی گزینه **Hardware** کلیک کرده تا به پیکربندی اجزای سخت افزاری بپردازیم.

ایجاد و پیکربندی یک پروژه در محیط Simatic manager



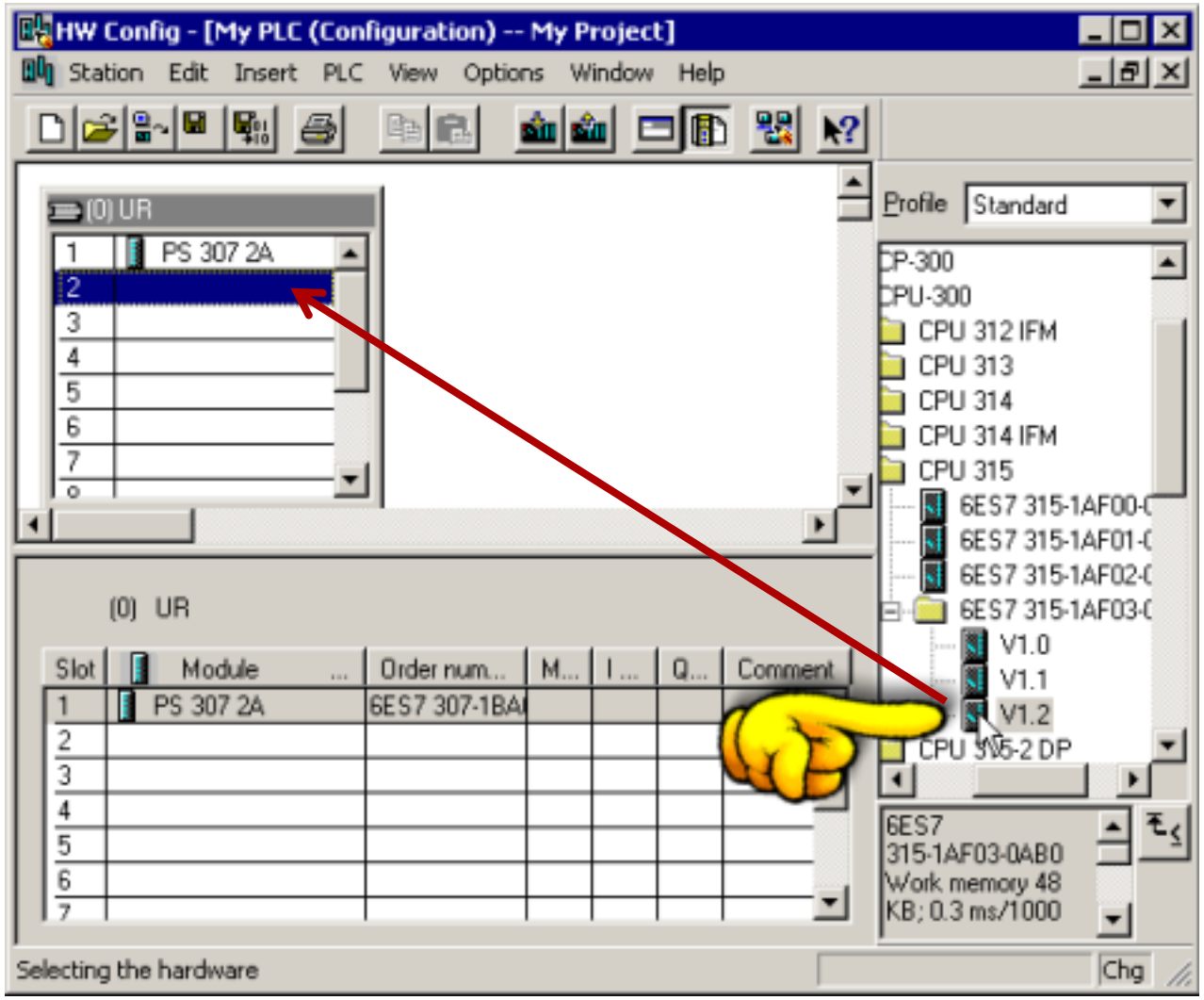
۵- در قسمت پیکربندی سخت افزار ابتدا می بایست رک را انتخاب نماییم.

ایجاد و پیکربندی یک پروژه در محیط Simatic manager



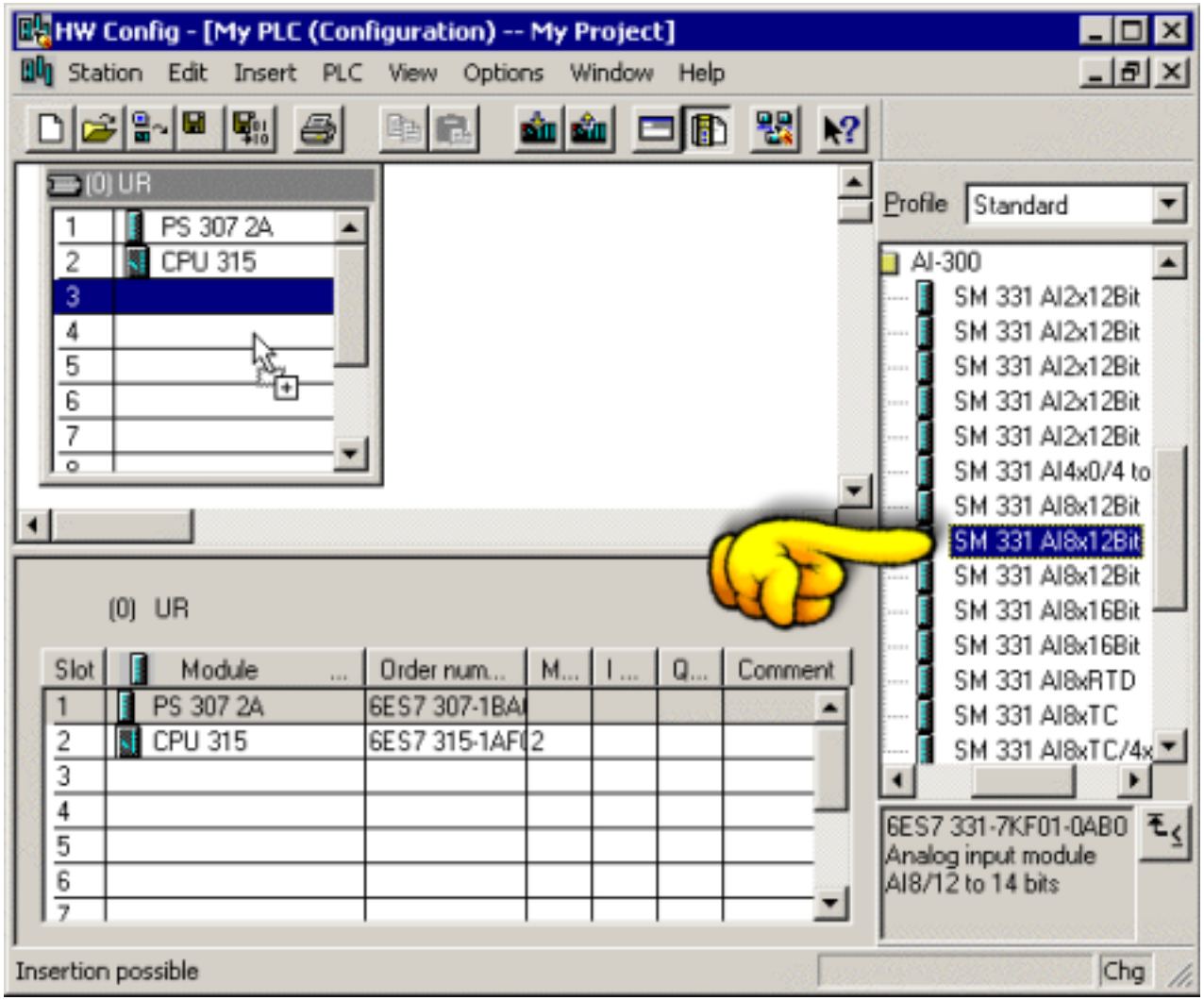
۶- اولین سطر از ریل را با منبع تغذیه مورد نظر پر می کنیم.

ایجاد و پیکربندی یک پروژه در محیط Simatic manager



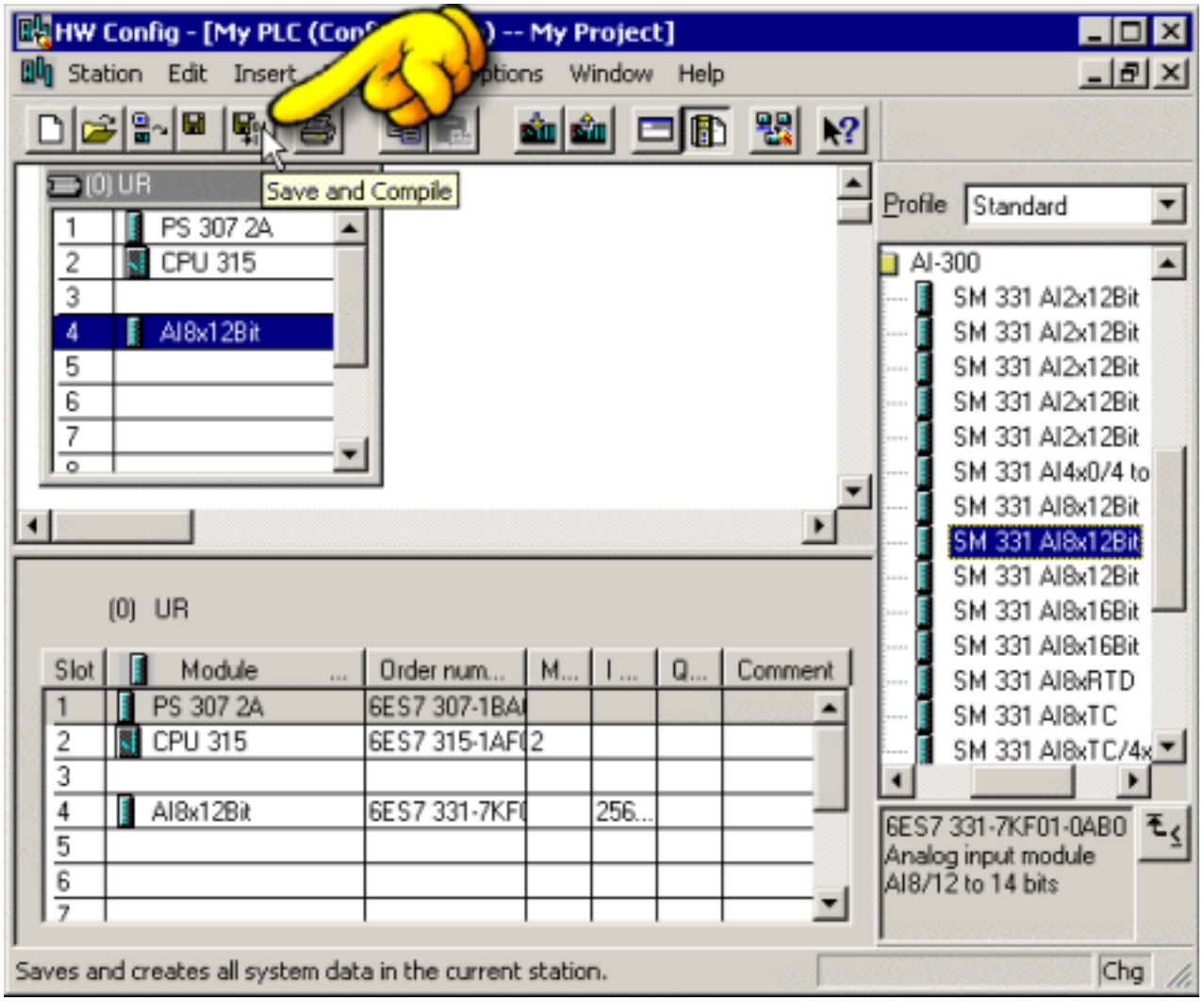
۷- دومین سطر از ریل را با CPU مورد نظر پر می کنیم.

ایجاد و پیکربندی یک پروژه در محیط Simatic manager



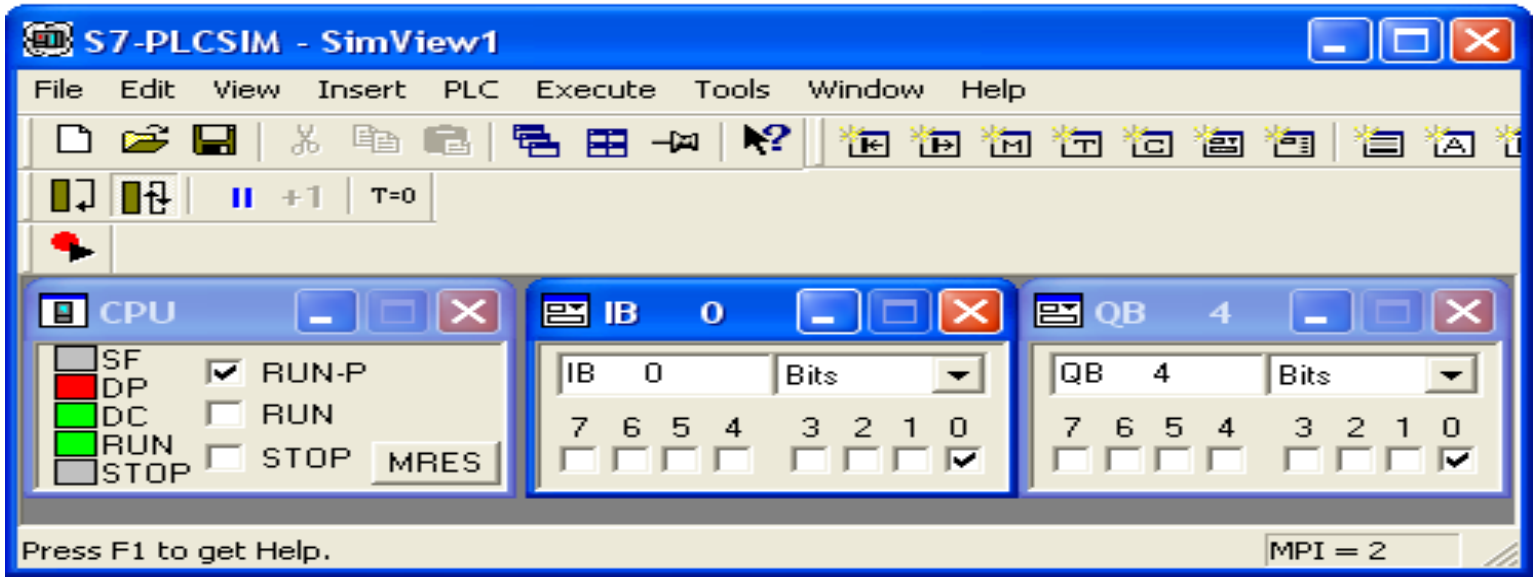
۸- سومین سطر از ریل را می توان با کارت های سیگنال (DI,DO,AI,AO) مورد نظر پر کرد.

ایجاد و پیکربندی یک پروژه در محیط Simatic manager



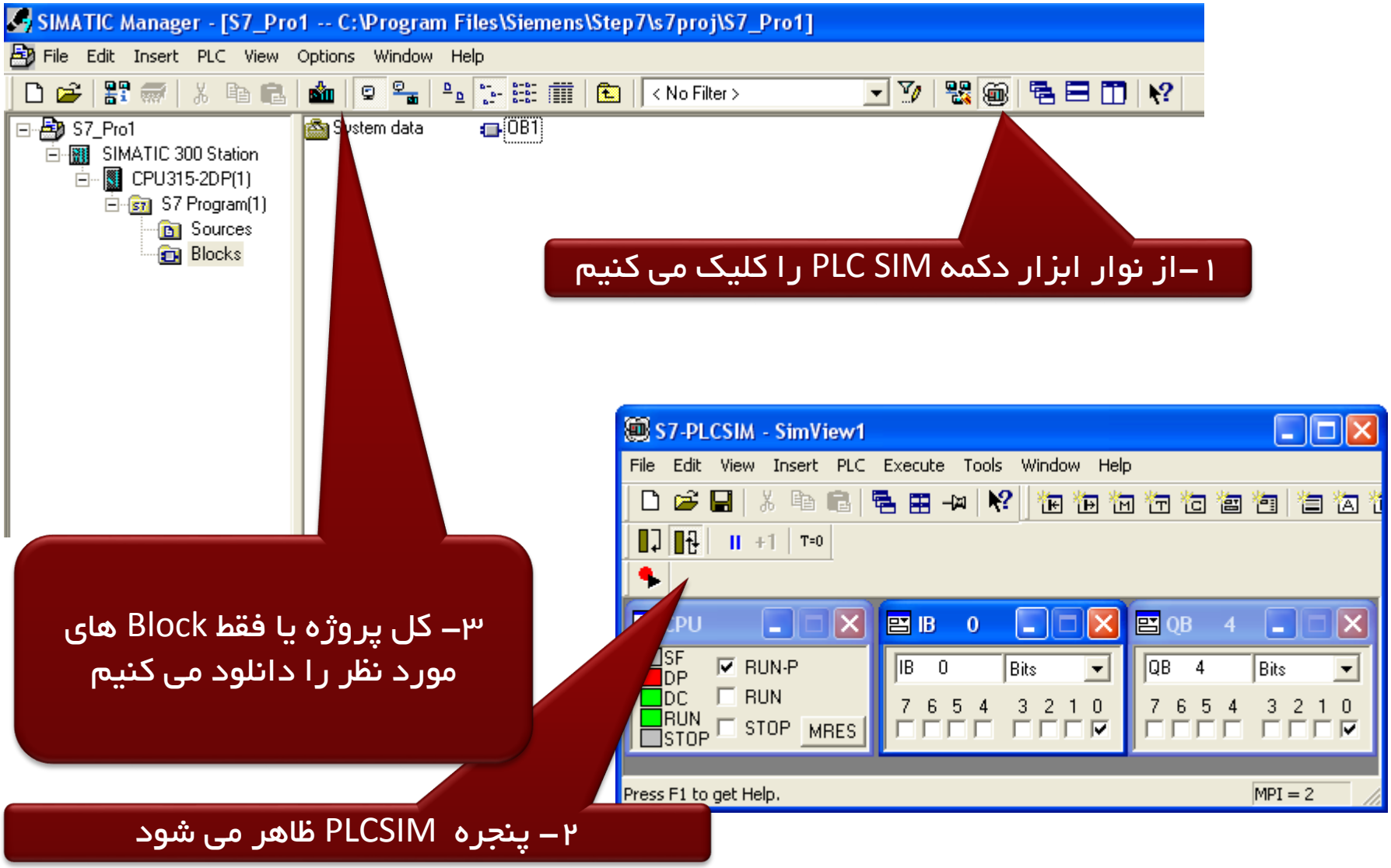
۹- در پایان از دکمه فوق الذکر ، پروژه را ذخیره و کامپایل می کنیم.

شبیه سازی سیگنال ها و حافظه ها با استفاده از نرم افزار PLCSIM

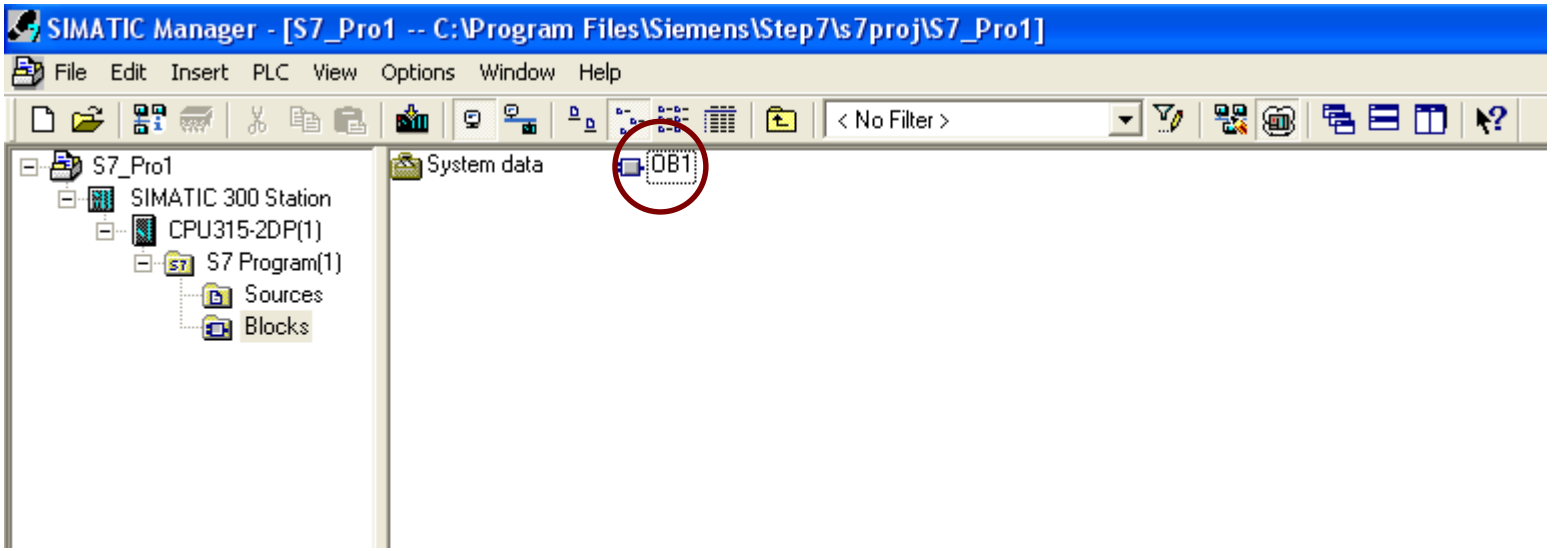


با استفاده از این نرم افزار می توان بدون دسترسی داشتن به یک PLC واقعی تمامی ورودی ها، خروجی ها، حافظه ها، تایمر و کانترهای را شبیه سازی نمود.

شبیه سازی سیگنال ها و حافظه ها با استفاده از نرم افزار PLCSIM

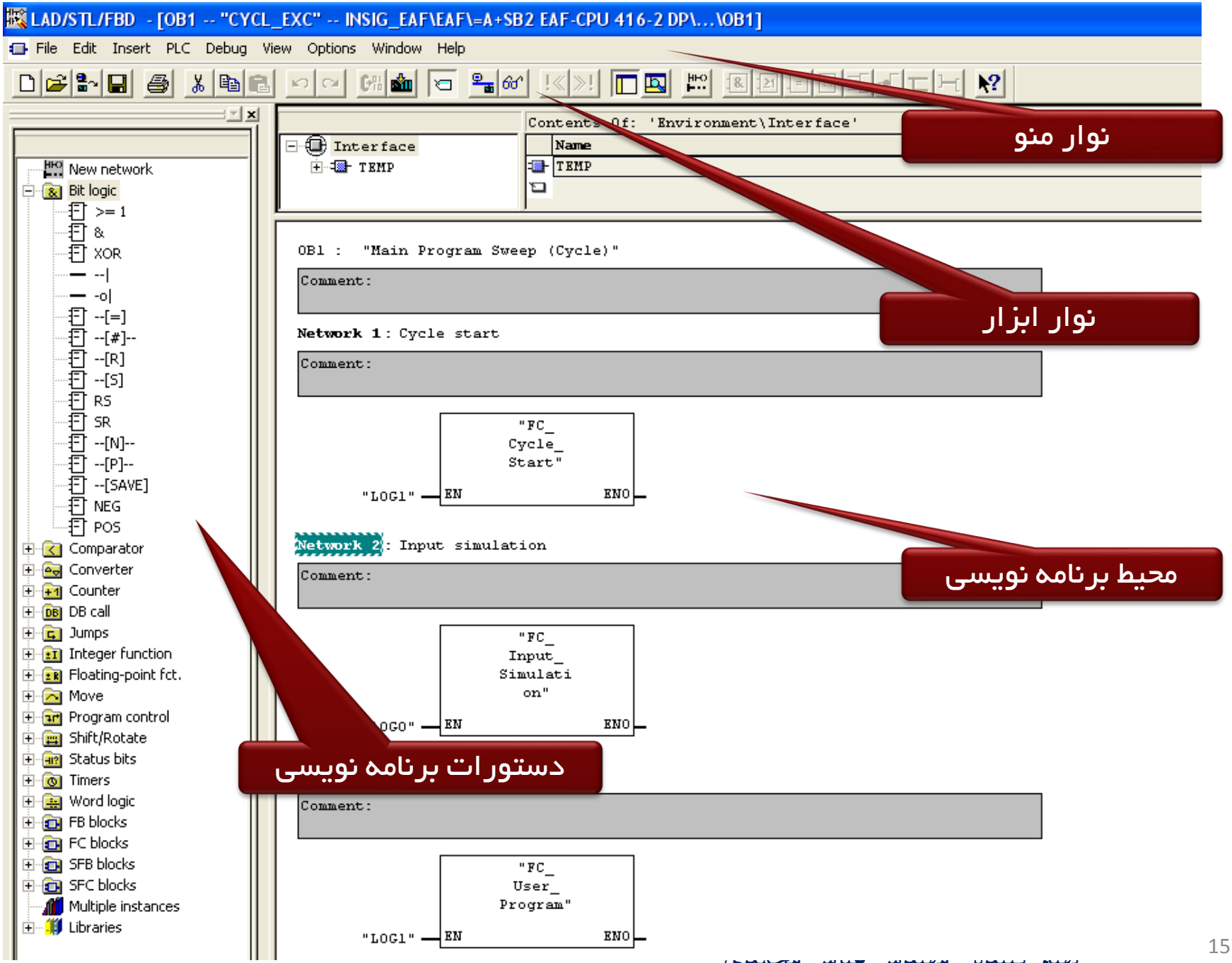


بلوک های سازماندهی
OB ها

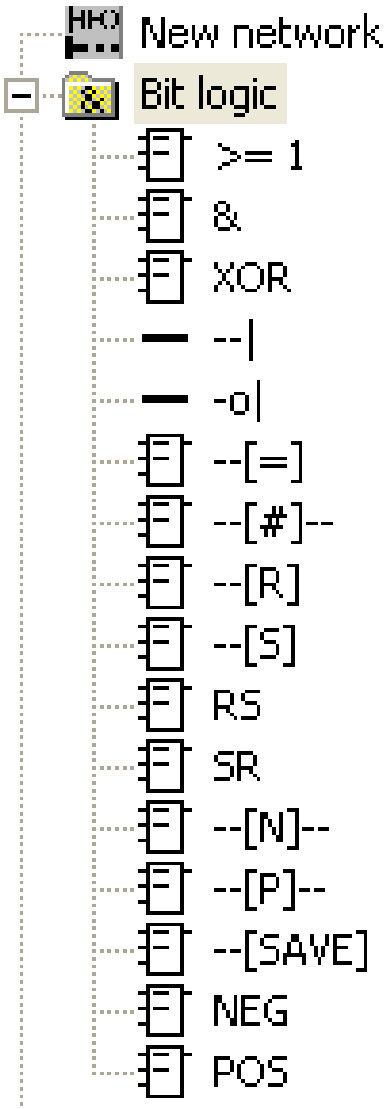


رابط بین سیستم عامل و کاربر می باشند که توسط سیستم عامل فراخوانی می شود.تعداد ***OB*** ها بستگی به مدل ***CPU*** دارد .هر ***OB*** شامل یک شماره خاص است .

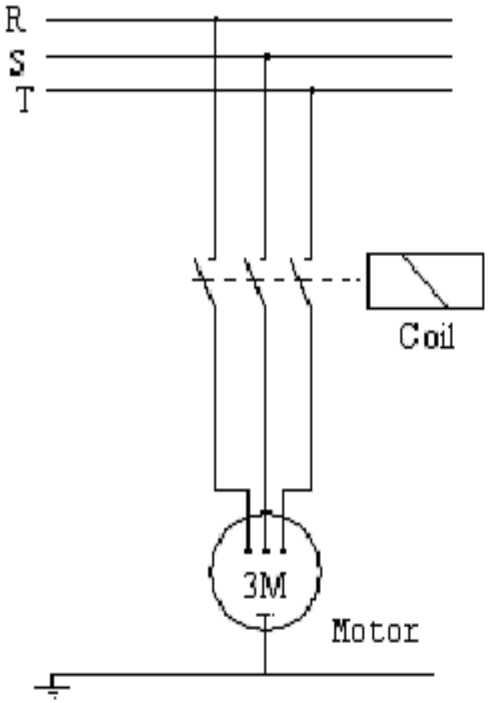
OB1 : در شروع هر سیکل برنامه سیستم عامل ***OB 1*** را فراخوانی می کند .در واقع اولین دستور در ***OB1*** شروع برنامه کاربر و آ خرین دستور آن پایان برنامه است . ***OB 1*** مهمترین بلوک سازماندهی است که ساختار برنامه کار بر را مشخص می کند .هر ***OB*** دارای یک درجه اولویت است و در این میان OB1 دارای کمترین اولویت است و امکان دارد در هر زمانی توسط OB های وقفه ، قطع شود .


LAD/STL/FBD - [OB1 -- "CYCL_EXC" -- INSIG_EAF\EAF\=A+SB2 EAF-CPU 416-2 DP\...\OB1]
File Edit Insert PLC Debug View Options Window Help
Contents Of: 'Environment\Interface'
Interface
TEMP
Name
TEMP
OB1 : "Main Program Sweep (Cycle)"
Comment:
Network 1: Cycle start
Comment:
"FC_ Cycle_ Start"
"LOG1"
EN
ENO
Network 2: Input simulation
Comment:
"FC_ Input_ Simulati on"
EN
ENO
Comment:
"FC_ User_ Program"
"LOG1"
EN
ENO
New network
Bit logic
>= 1
&
XOR
--|
-o|
--[=]
--[#]--
--[R]
--[S]
RS
SR
--[N]--
--[P]--
--[SAVE]
NEG
POS
Comparator
Converter
Counter
DB call
Jumps
Integer function
Floating-point fct.
Move
Program control
Shift/Rotate
Status bits
Timers
Word logic
FB blocks
FC blocks
SFB blocks
SFC blocks
Multiple instances
Libraries
نوار منو
نوار ابزار
محیط برنامه نویسی
دستورات برنامه نویسی

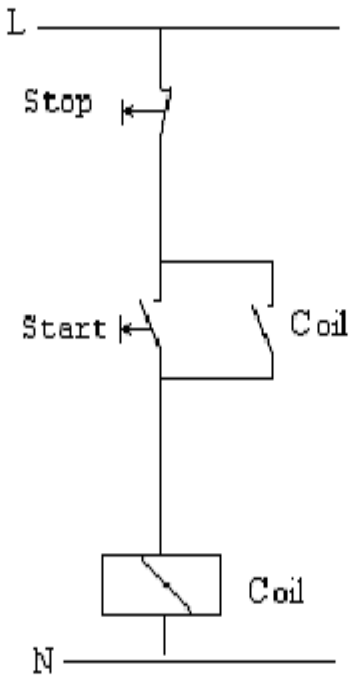
Bit Logic عملیات



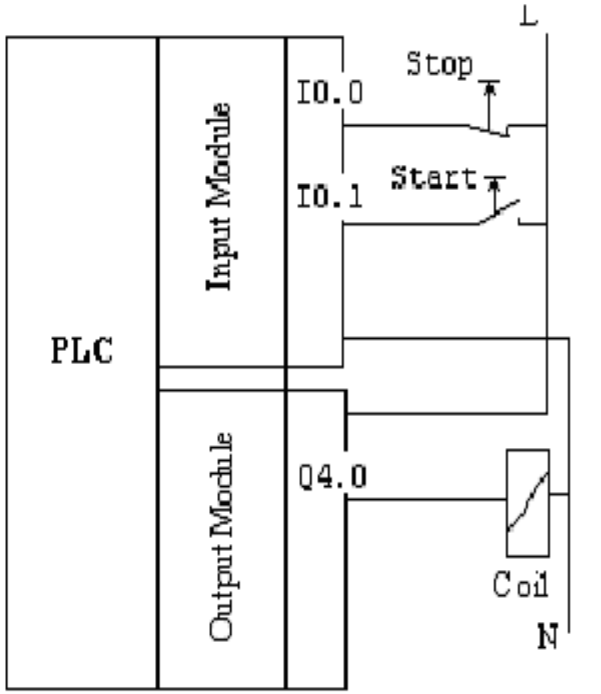
کاربرد	دستور LAD	دستور FBD
خواندن وضعیت یک بیت	---| |---	-
خواندن وضعیت معکوس یک بیت	---| / |---	-
عکس کردن نتیجه لاجیک	---| NOT |---	-O|
اختصاص نتیجه لاجیک به یک بیت	--( )	=
ذخیره‌سازی نتایج میان برنامه	--(#)--	#
ری‌ست کردن یک بیت	--(R)	R
ست‌کردن یک بیت	--(S)	S
ست و ری‌ست کردن یک بیت با اولویت ست	RS S Q R	RS R S Q
ست و ری‌ست کردن یک بیت با اولویت ری‌ست	SR S Q R	SR S R Q
تشخیص لبه منفی نتیجه عملیات لاجیک	--(N)--	N
تشخیص لبه مثبت نتیجه عملیات لاجیک	--(P)--	P
ذخیره کردن RLO در یک بیت حافظه	-(SAVE)	SAVE
تشخیص لبه منفی یک بیت با آدرس مشخص	NEG Q M_BIT	NEG M_BIT Q
تشخیص لبه مثبت یک بیت با آدرس مشخص	POS Q M_BIT	POS M_BIT Q
OR دو یا چند بیت	-	>=1
AND دو یا چند بیت	-	&
Exclusive OR دو یا چند بیت	-	XOR



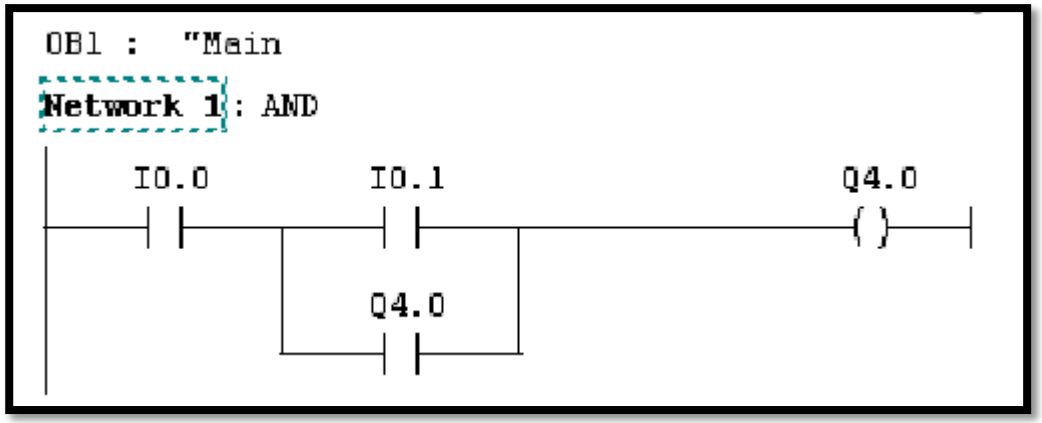
مدار الکتریکی قدرت



مدار الکتریکی فرمان



نحوه ارتباط کوئیل‌ها کنتاکتورها به خروجی‌ها
و ارتباط کلیدها به ورودی‌های PLC

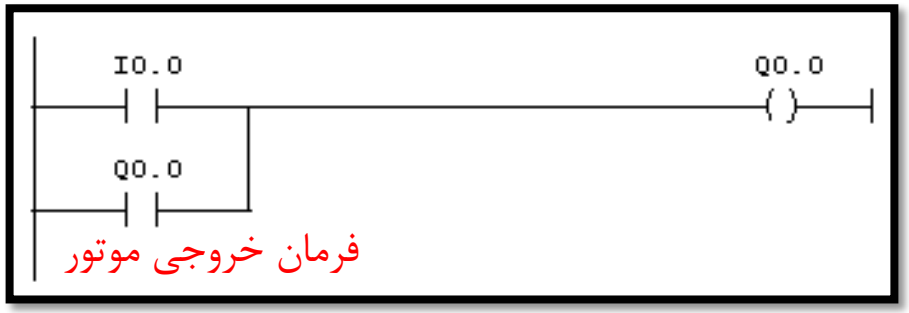


مثال: برنامه زیر را با PLC SIM تست نمایید:

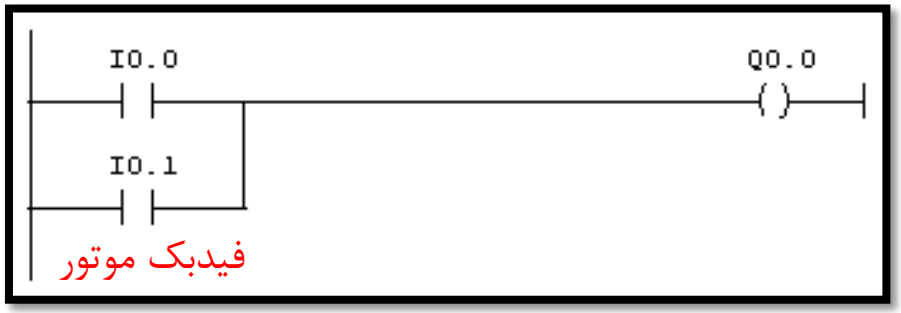
مثال : در برنامه فوق یک شستی (Push Button) که بصورت NO است برای استارت موتور استفاده شده است ، آیا منطق برنامه موجب استارت دائم موتور خواهد شد؟

خیر- برای حل مشکل فوق و برای اینکه منطق برنامه بصورتی باشد که خروجی پس از فعال شدن حالت خود را حفظ نماید ۳ راه حل وجود دارد:

روش اول



روش دوم

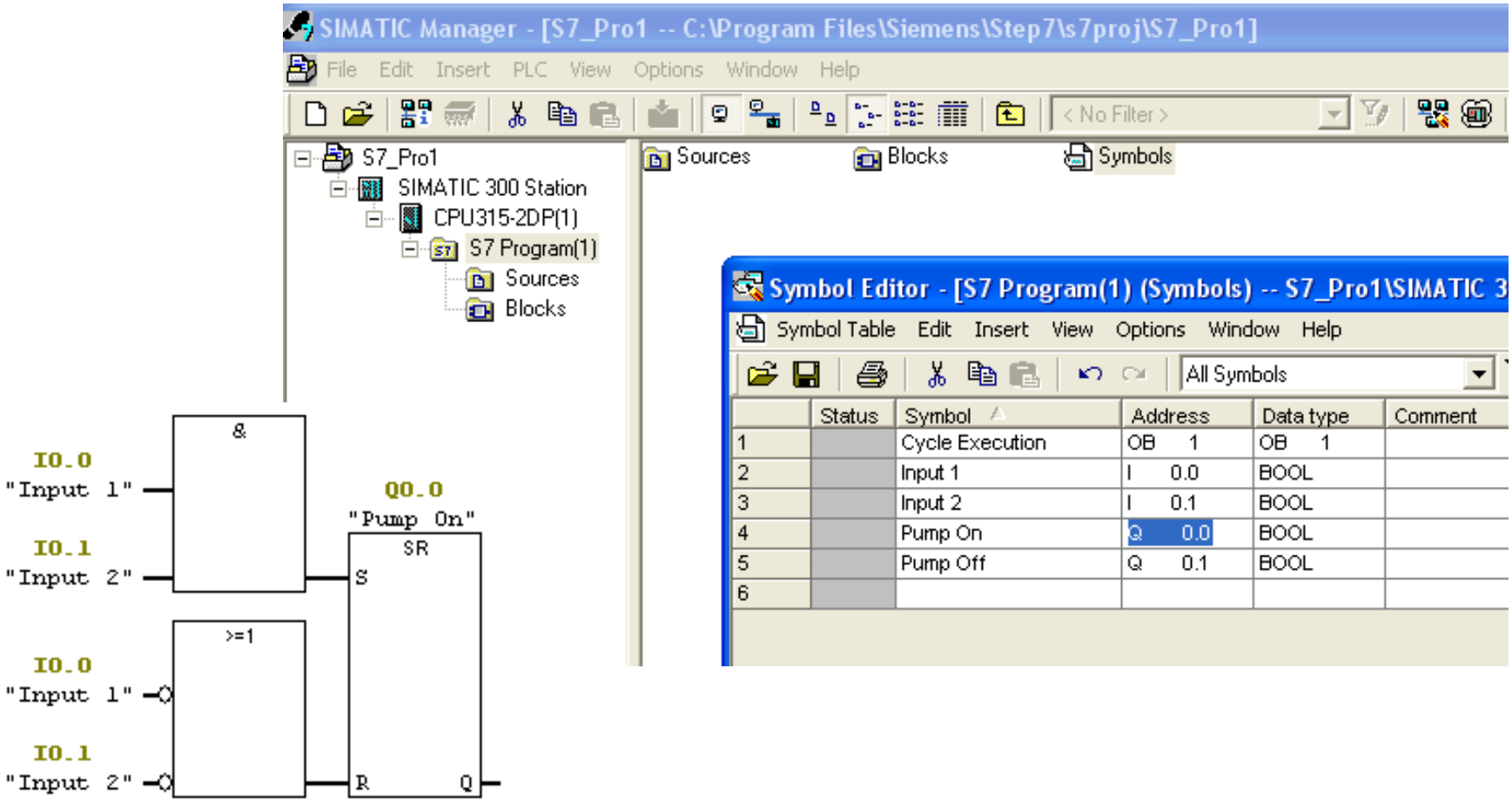


راه حل سوم که از تمامی روش ها بهتر است را در ادامه خواهید دید

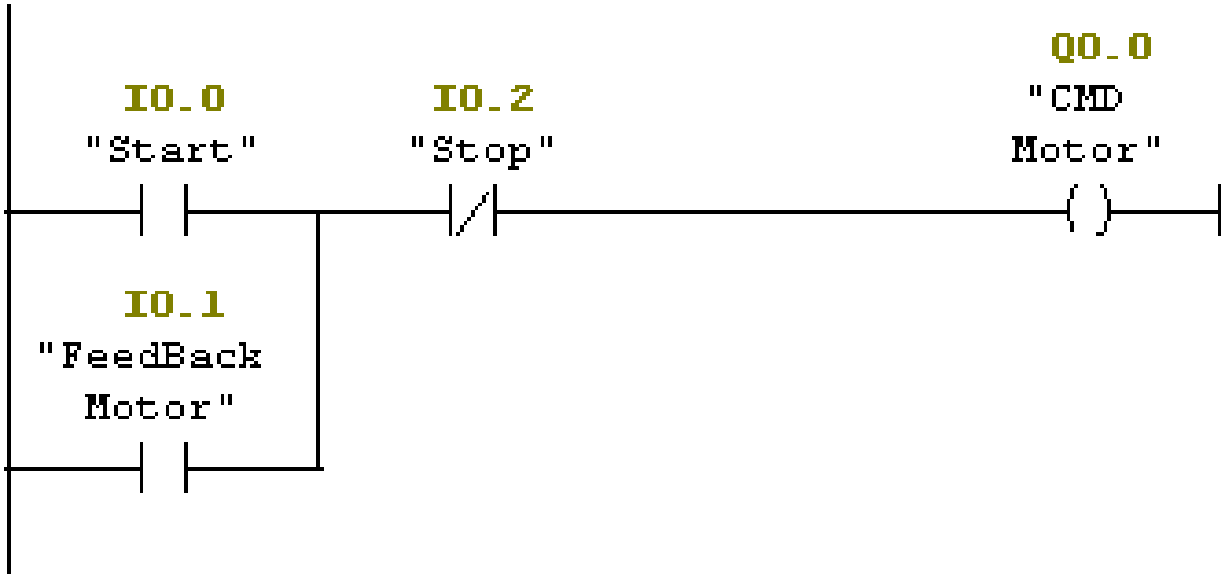
جدول Symbol

در برنامه S7 تک تک متغیرها به همراه آدرسشان در جدولی بنام جدول سمبل ها نگهداری می شوند.
مزیت این جدول این است که می توان برای هر متغیر ، از نام های سمبلیک استفاده نمود.

مثلاً Q0.0 بنام Pump On



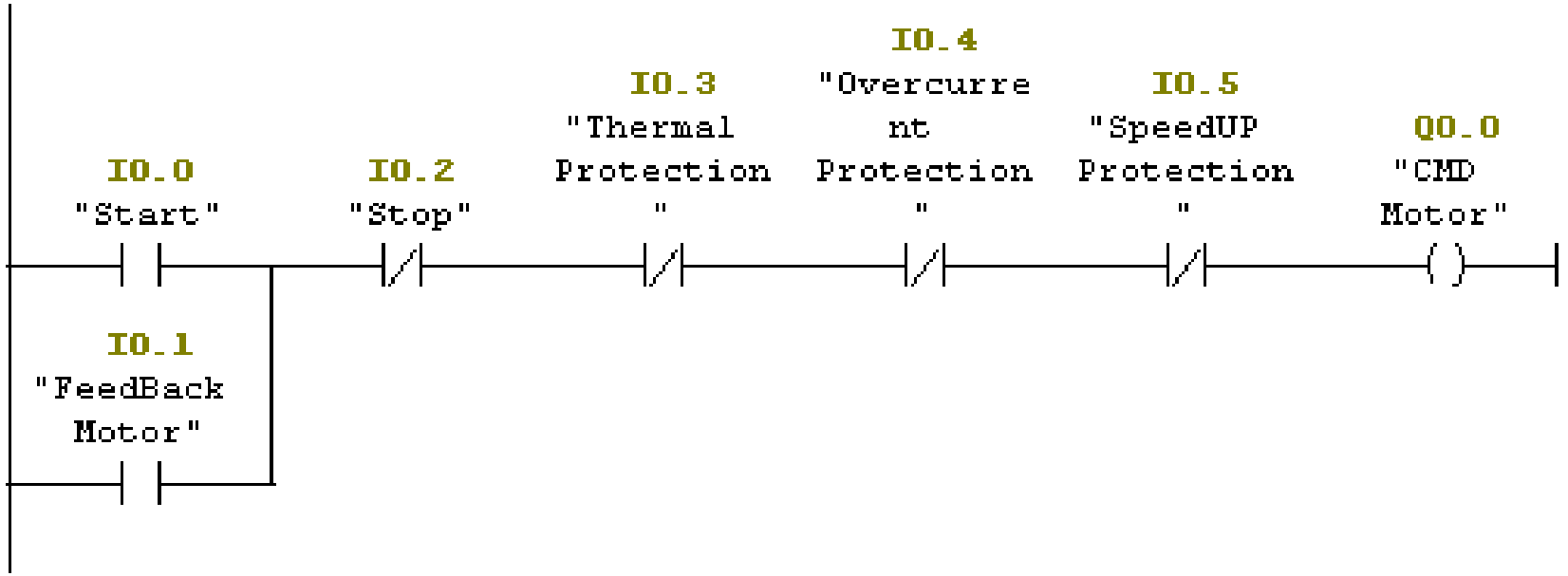
مثال: در برنامه مثال قبل یک شستی Stop که به I0.2 متصل است قرار دهید، این شستی بصورت NC می باشد.(آدرس ها را Symbol گذاری کنید)



مثال: در برنامه مثال قبل کنترل موتور کافی نیست و لازم است اینترلاک های ایمنی نیز در نظر گرفته شوند.(اینترلاک های اضافه جریان و دمای بالای موتور یا دور بالای موتور) کنتاکتهای حفاظتی بصورت NC هستند و در صورت عملکرد باز می شوند و هر کدام که عمل کنند موتور خاموش خواهد شد، برنامه آن را به در نظر گرفتن مسائل حفاظتی موتور تغییر دهید:

آدرس	سمبل
I 0.0	Start
I 0.1	Feed Back Motor
I 0.2	Stop
I 0.3	Thermal Protection
I 0.4	Over Current Protection
I 0.5	Speed UP Protection
Q 0.0	CMD Motor

پاسخ:


I0.0
"Start"
I0.1
"FeedBack
Motor"
I0.2
"Stop"
I0.3
"Thermal
Protection
"
I0.4
"Overcurre
nt
Protection
"
I0.5
"SpeedUP
Protection
"
Q0.0
"CMD
Motor"

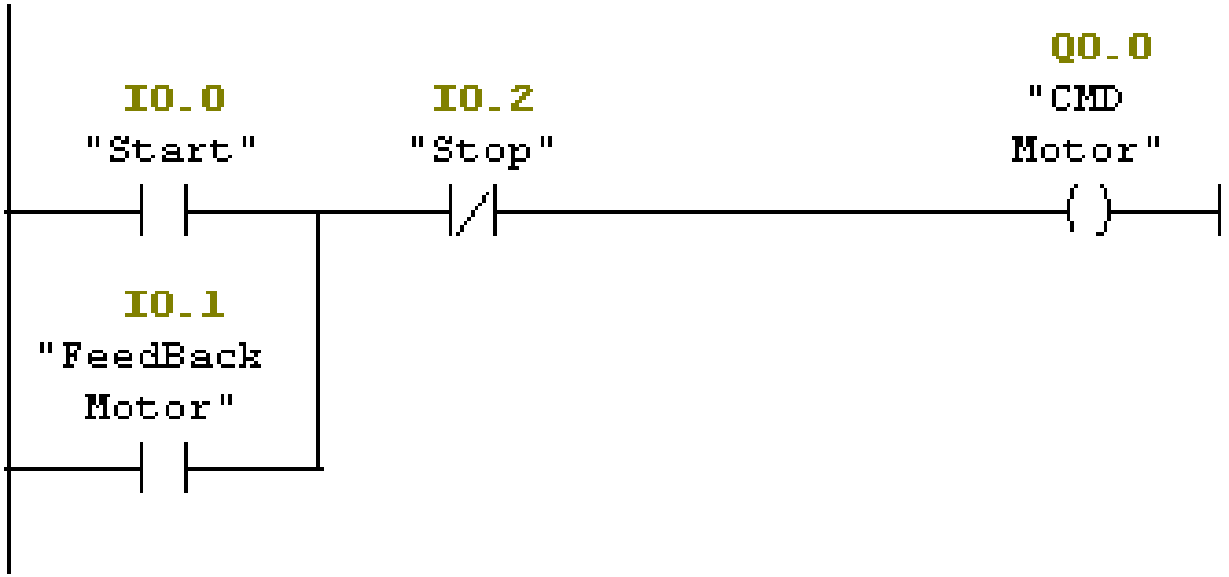
مثال: در فرایندی سه پمپ که موتور آنها به ترتیب به آدرس های Q2.0 ,Q2.1 و Q2.2 متصل هستند وجود دارند نمایش وضعیت آنها با فرکانس چشمک زن چراغ Q2.7 برای اپراتور نشان داده شده اند. برنامه را بصورتی بنویسید که:

الف- اگر هر ۳ پمپ روشن باشند چراغ Q2.7 با فرکانس کم چشمک بزند.
ب- اگر دو پمپ از ۳ پمپ روشن باشد چراغ Q2.7 با فرکانس متوسط چشمک بزند.
ج- اگر یک پمپ از ۳ پمپ روشن باشد چراغ Q2.7 با فرکانس زیاد چشمک بزند.

با استفاده از Clock Memory مربوط به CPU انجام شود:

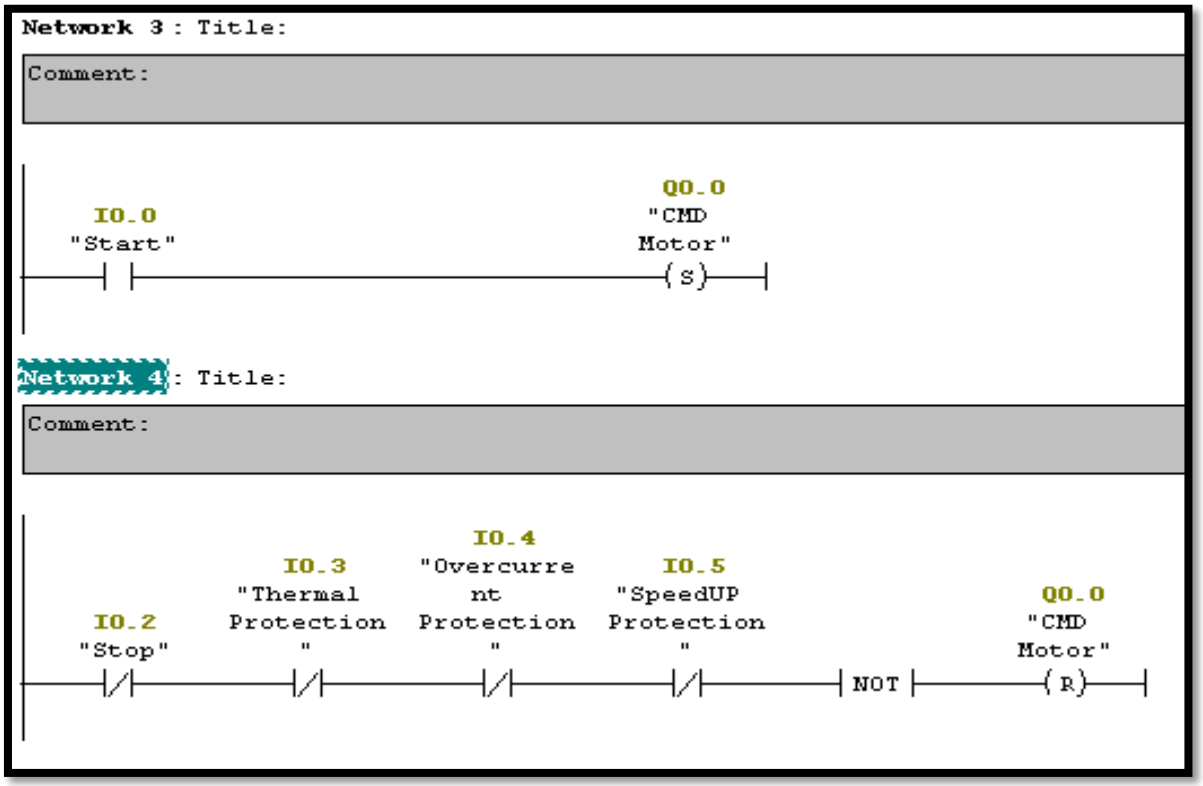
Bit	7	6	5	4	3	2	1	0
Duration(s)	2	1.6	1	0.8	0.5	0.4	0.2	0.1
Frequency(Hz)	0.5	0.625	1	1.25	2	2.5	5	10

مثال: برنامه زیر را با دستورات Set و Reset انجام دهید:

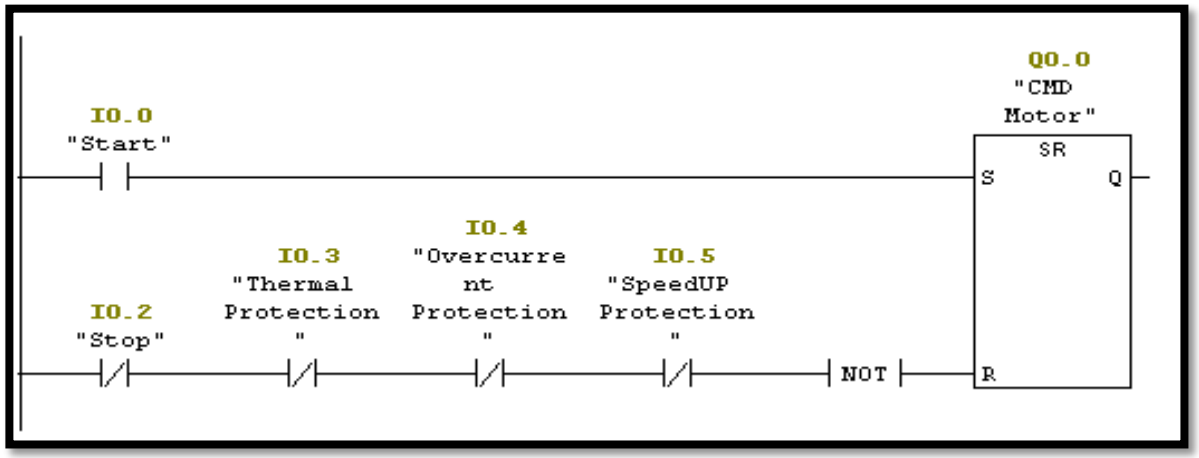


پاسخ:

روش اول

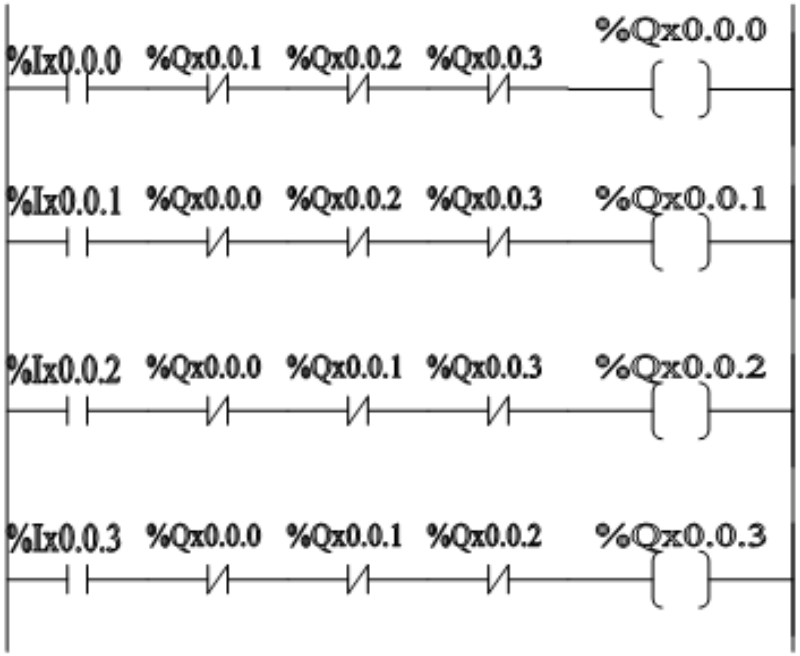


روش دوم



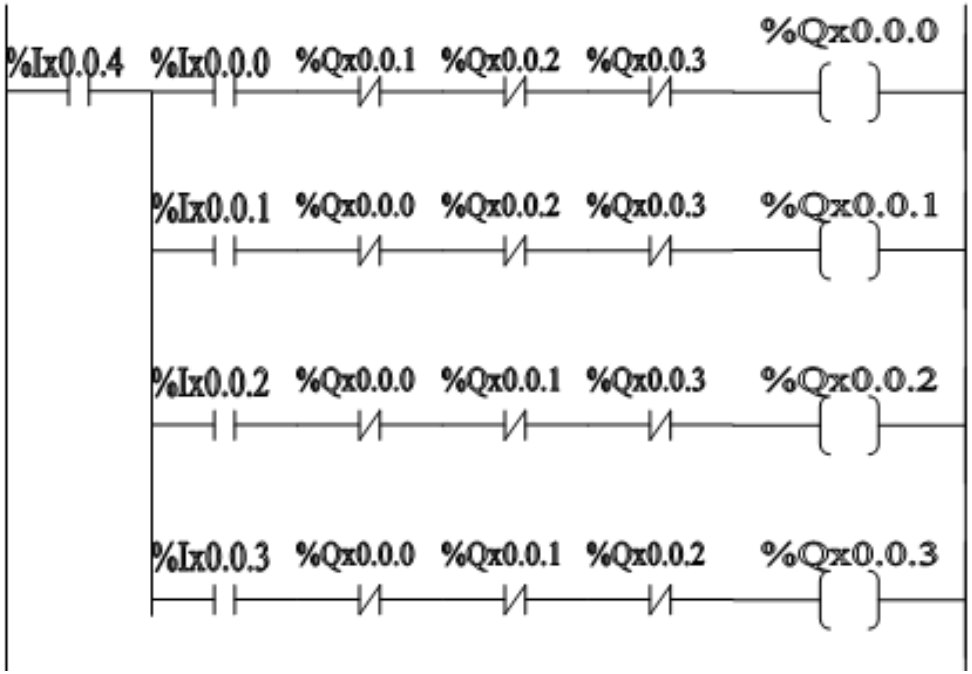
مثال :

برنامه ای بنویسید که در روی یک میز مسابقه که ۴ نفر شرکت کننده دارد هر کسی که کلید را زودتر فشار داد چراغ او فقط روشن شود و چراغ دیگران خاموش بماند .



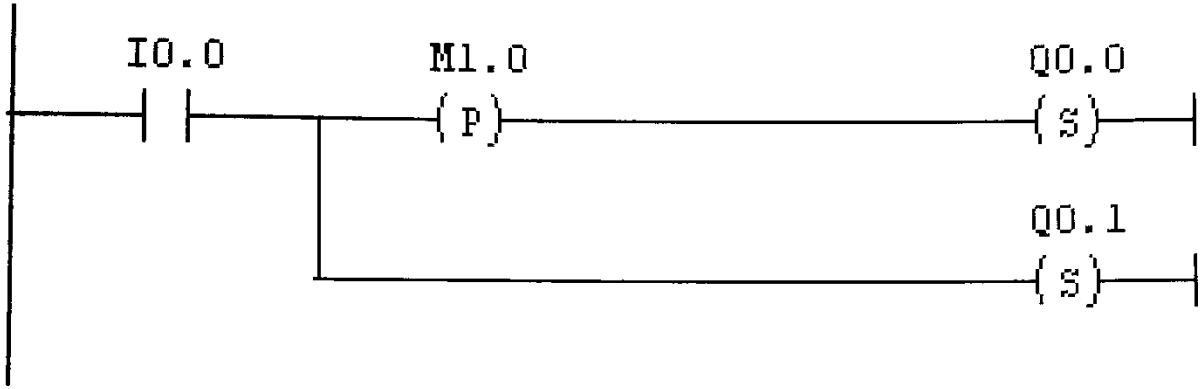
مثال :

همان برنامه بالا فقط با در نظر گرفتن این نکته که هر موقع که مجری برنامه اجازه داد چراغ هـا روشـن شوند .

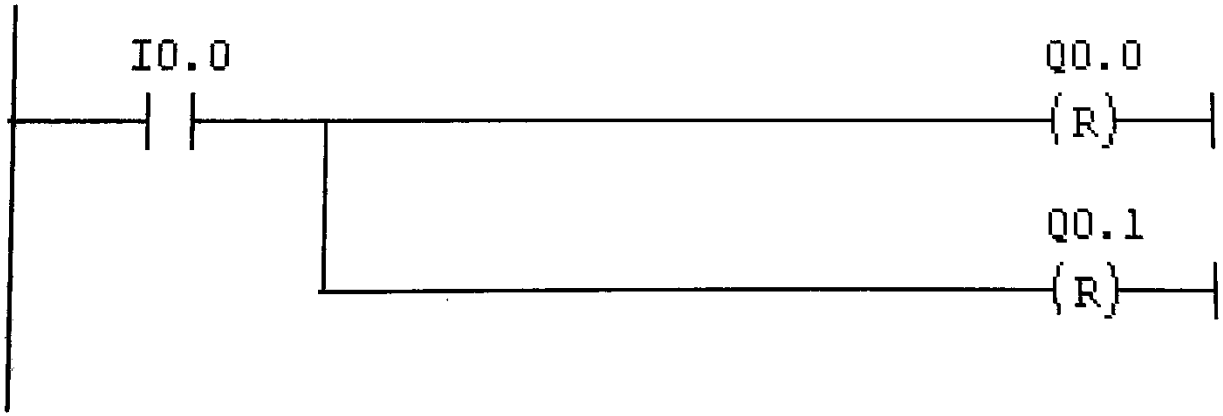


عملکرد برنامه زیر را بررسی نمایید:

Network 1 :

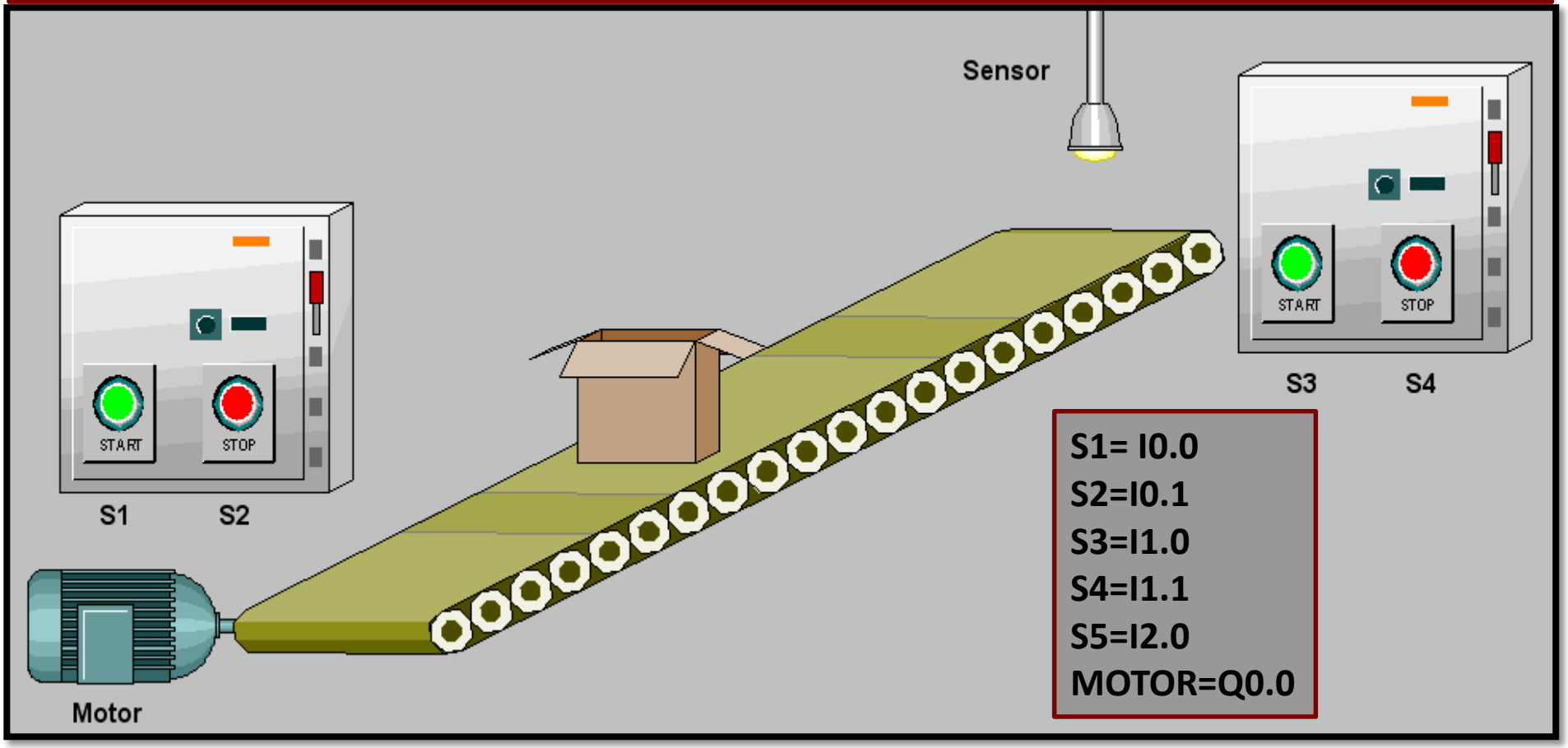


Network 2 :

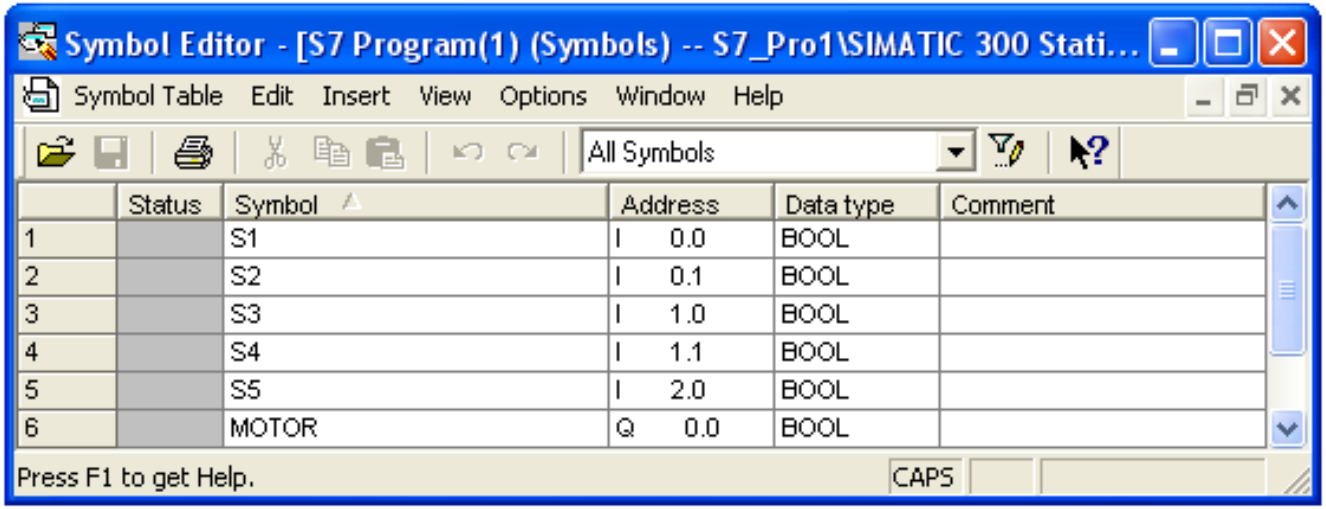


تمرین:

در شکل زیر دو کلید فشاری S1 وS2 برای استارت و استپ کانوایر در سمت آغازین و کلیدهای S3 و S4 برای استارت و استپ در سمت پایانی کانوایرتعبیه شده اند. همچنین سنسور S5 برای توقف کانوایر نصب شده است. این سنسور در وضعیت نرمال بسته می باشد.

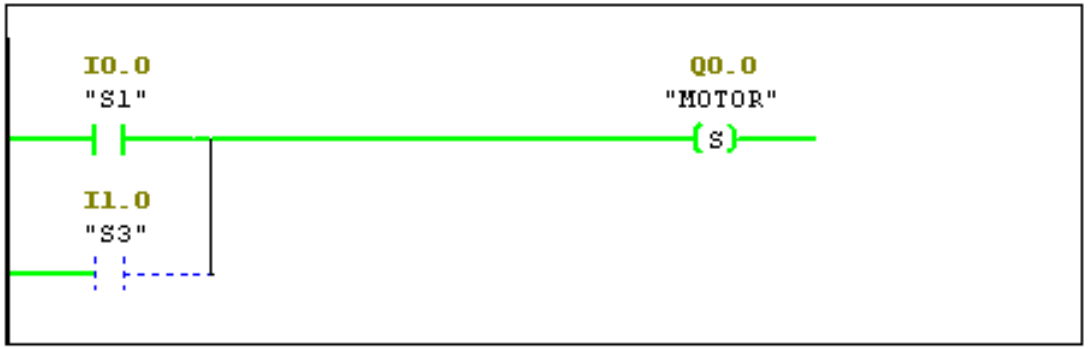


ابتدا جدول سمبل ها را تشکیل می دهیم:



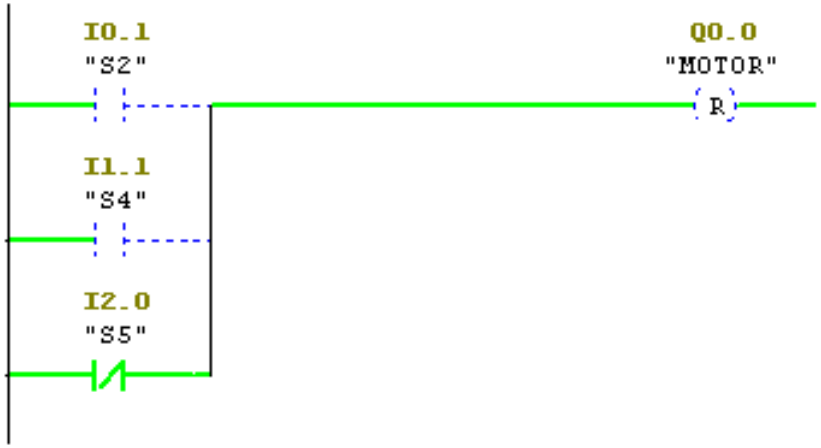
Network 6 : Title:

Comment:

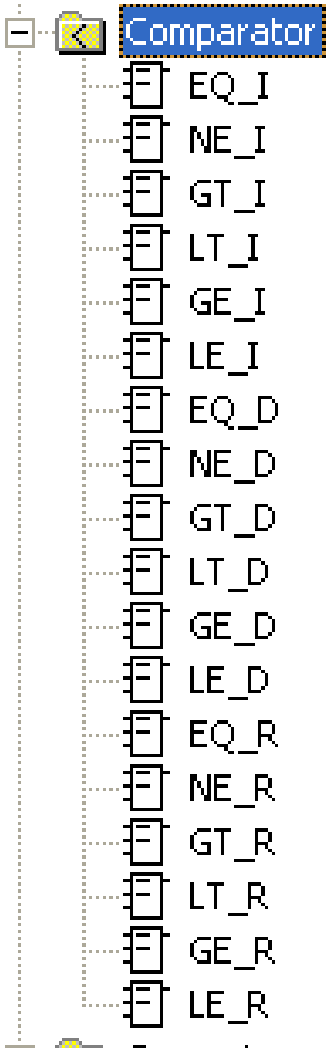


Network 7 : Title:

Comment:



دستورات مقایسه کننده



EQ مساوی
NE مخالف
GT بزرگتر
LT کوچکتر
GE بزرگتر مساوی
LE کوچکتر مساوی

I عدد صحیح
D عدد صحیح دوبل
R عدد اعشاری

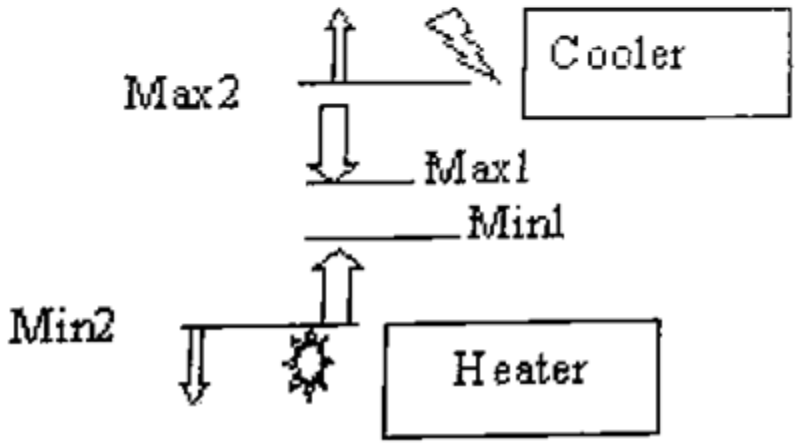
مثال : برنامه کنترل زیر را طوری بنویسید که :

الف : اگر دما از ۱۰ درجه کمتر باشد Hater فعال شود.

ب : اگر دما از ۳۰ درجه بیشتر شود Cooler فعال شود.

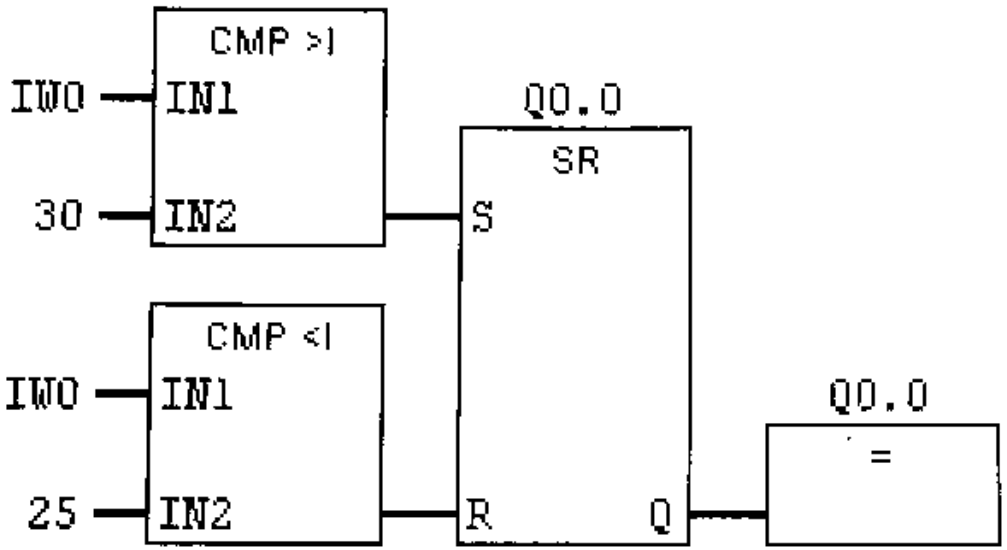
ج : اگر دما از ۳۰ درجه کمتر شد کولر بلافاصله خاموش نشود بلکه تا زمانی که دمای سیستم ۲۵ درجه نشده روشن باشدتا سیستم کاملا خنک شود.همچنین اگر دما از ۱۰ درجه بیشتر شد گرمکن بلافاصله خاموش نشود بلکه تا ۱۵ درجه روشن بماند.

توجه : اگر دما ی سیستم در محدوده $15 < \theta < 25$ باشد Cooler و Hater هیچ کدام فعال نباشند.

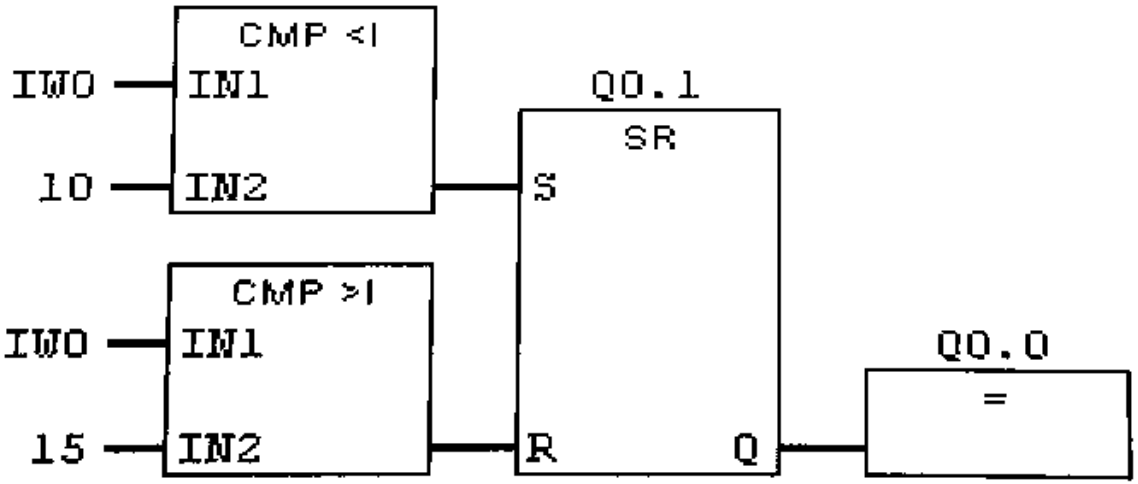


OB1

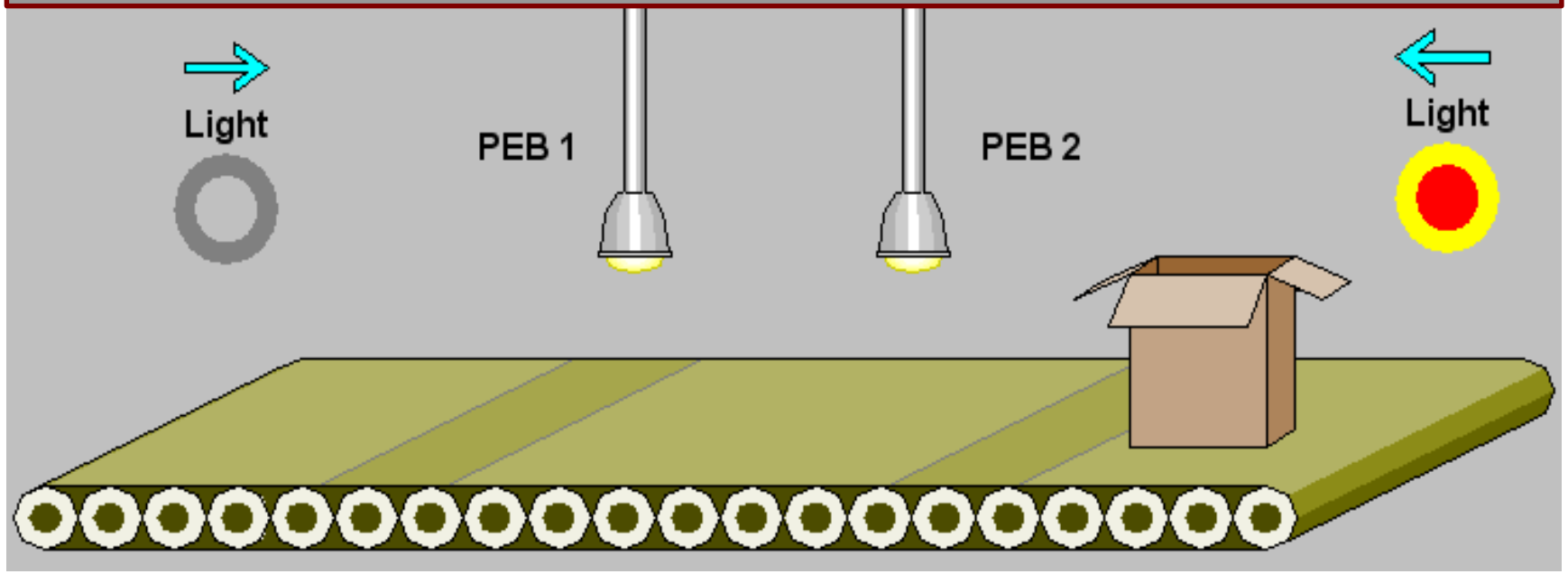
Network 1



Network 2



تمرین:
سنسورهای فتوالکتریک **PEB1** و **PEB2** که هر دو نرمال باز هستند برای تشخیص جهت حرکت نوار نقاله شکل زیر طراحی شده اند. این سنسورها حضور جسم را تشخیص می دهند، می خواهیم وقتی نوار نقاله به سمت راست حرکت کند لامپ **Q4.0** و وقتی به سمت چپ حرکت می کند لامپ **Q4.1** روشن شود.

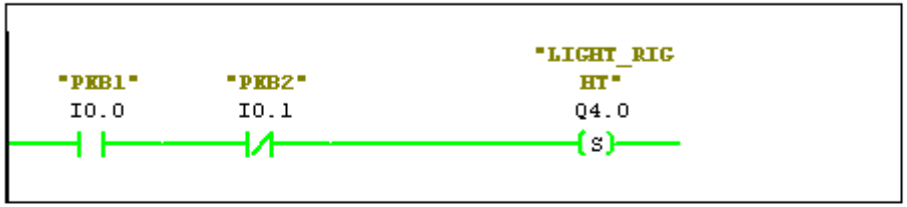


OB1 : "Main Program Sweep (Cycle)"

Comment:

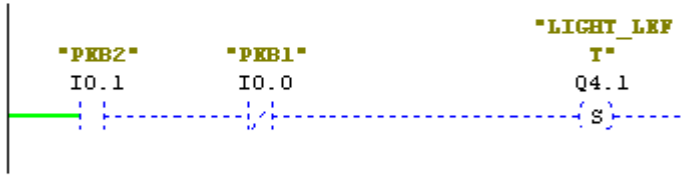
Network 1: Title:

Comment:



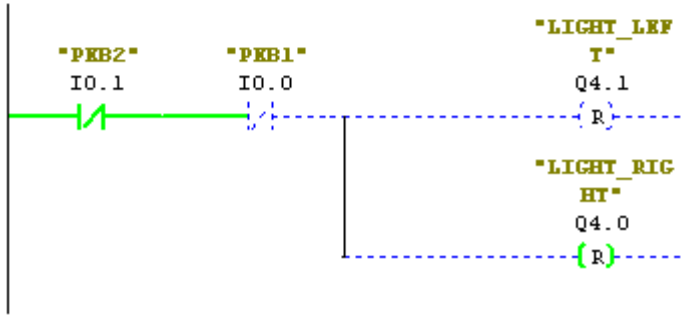
Network 2 : Title:

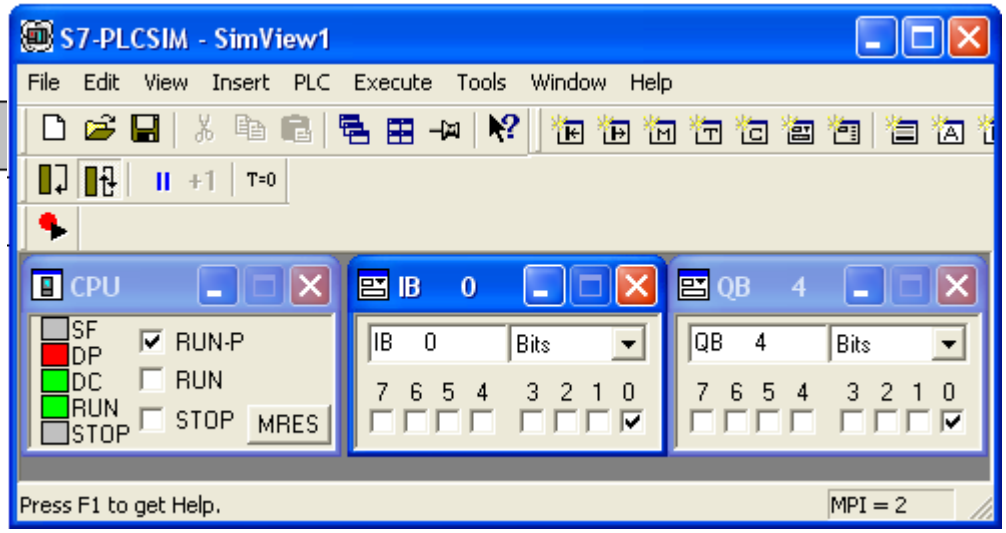
Comment:



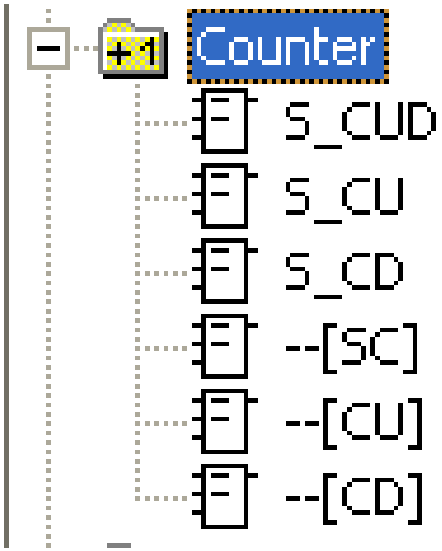
Network 3 : Title:

Comment:



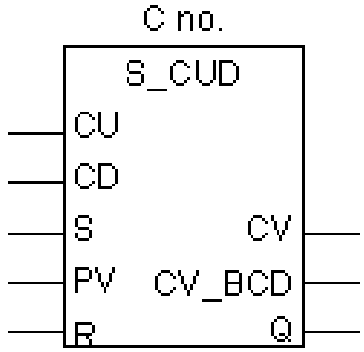


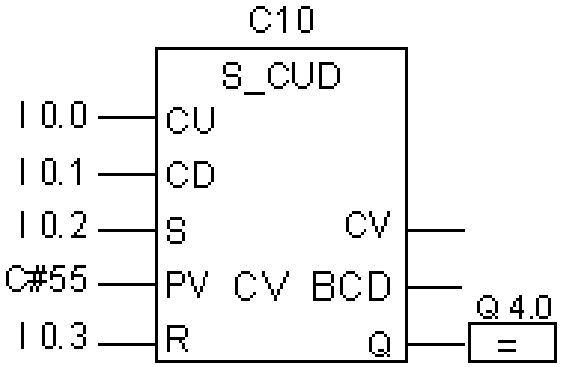
دستورات شمارنده



S_CUD	کانتر صعودی یا نزولی
S_CU	کانتر صعودی
S_CD	کانتر نزولی
SC	کانتر بیتی ، انتقال یک مقدار دلخواه به یک کانتر
CU	کانتر بیتی ، افزایش یک واحد
CD	کانتر بیتی ، کاهش یک واحد

شمارنده صعودی/نزولی



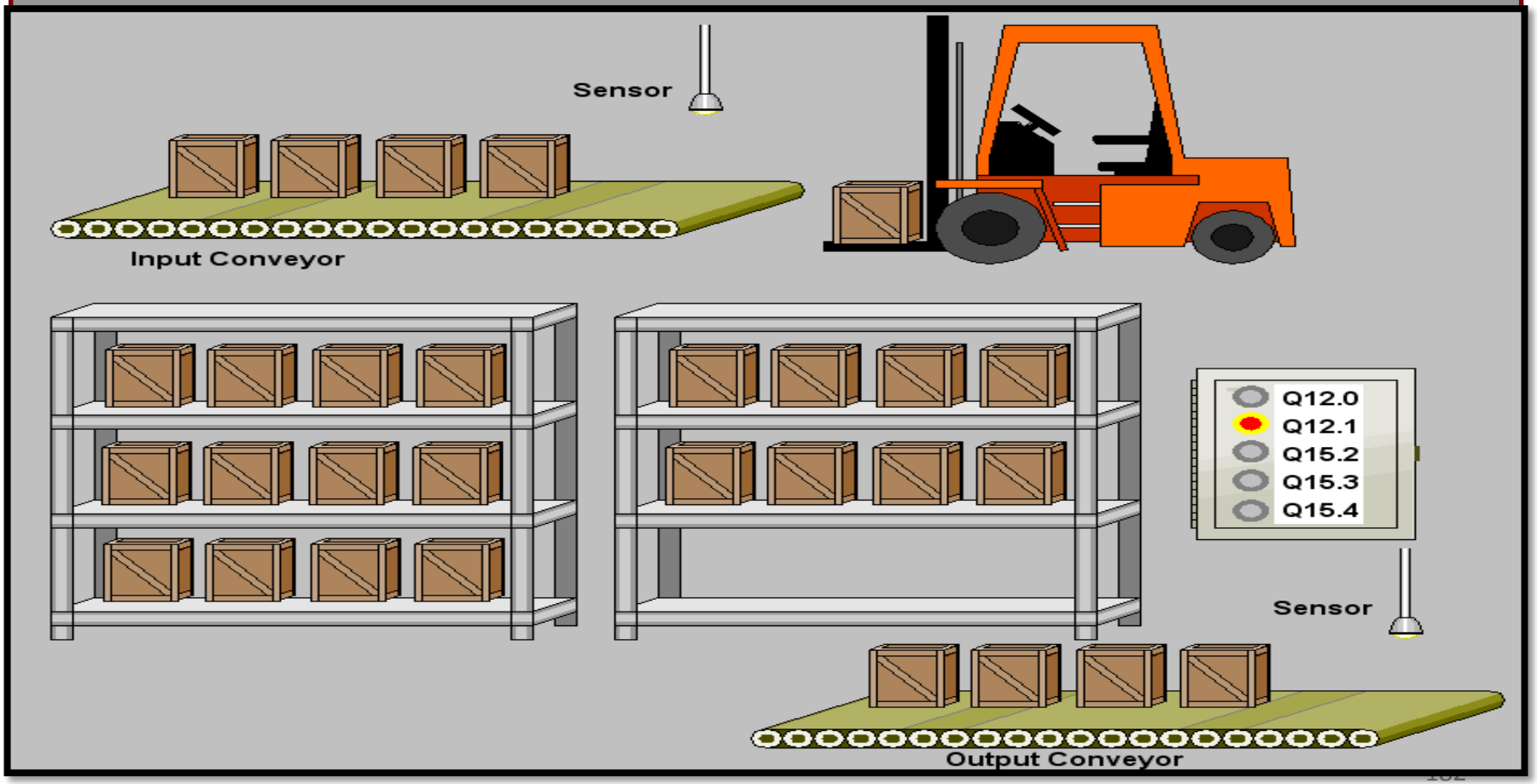


no	شماره کانتر
CU	ورودی فعال کننده افزایش کانتر
CD	ورودی فعال کننده کاهش کانتر
S	ورودی برای ست کردن مقدار اولیه به کانتر
PV	مقدار اولیه بصورت کانتر
R	ورودی ریست کننده کانتر
Q	خروجی نمایش وضعیت کانتر
CV	مقدار لحظه ای کانتر بصورت **HEX**
CV_BCD	مقدار لحظه ای کانتر بصورت **BCD**

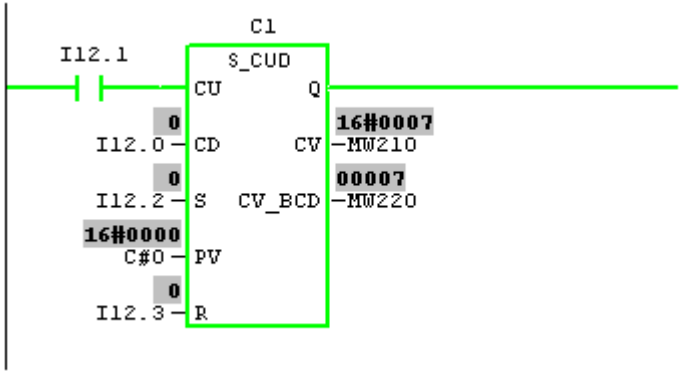
تمرین:

برنامه ای بنویسید که:

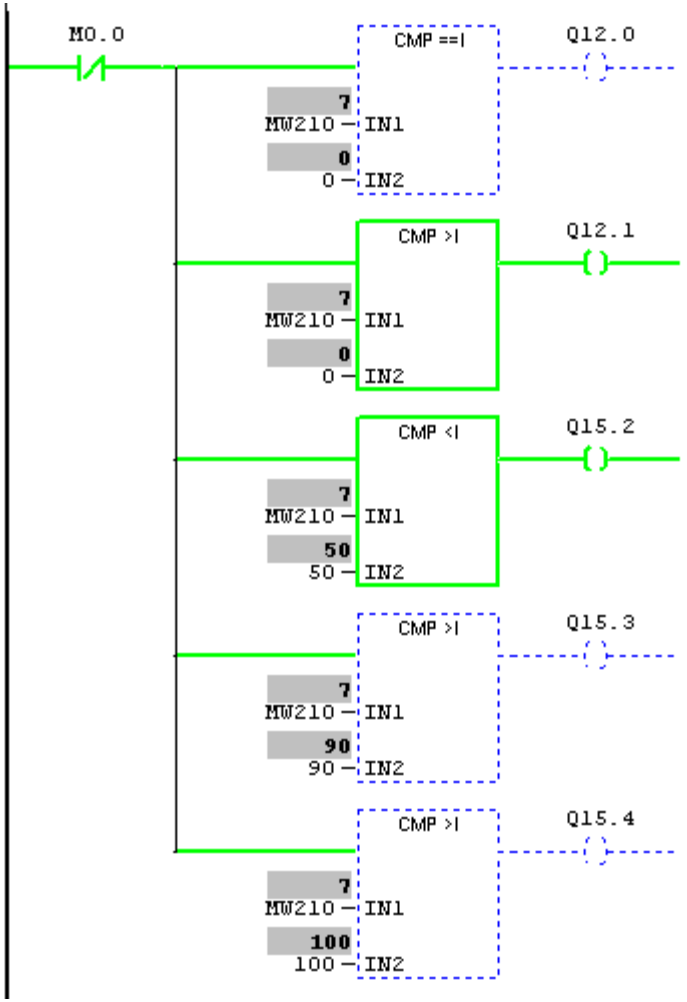
- اگر انبار خالی بود چراغ Q12.0 روشن شود.
- اگر بسته ای در انبار وجود داشت چراغ Q12.1 روشن باشد.
- اگر بسته در انبار از ۵۰ عدد کمتر باشد چراغ Q15.2 روشن شود.
- اگر بسته در انبار از ۹۰ عدد بیشتر باشد چراغ Q15.3 روشن شود.
- اگر تعداد بسته ها از ۱۰۰ عدد بیشتر شد چراغ Q15.4 روشن شود



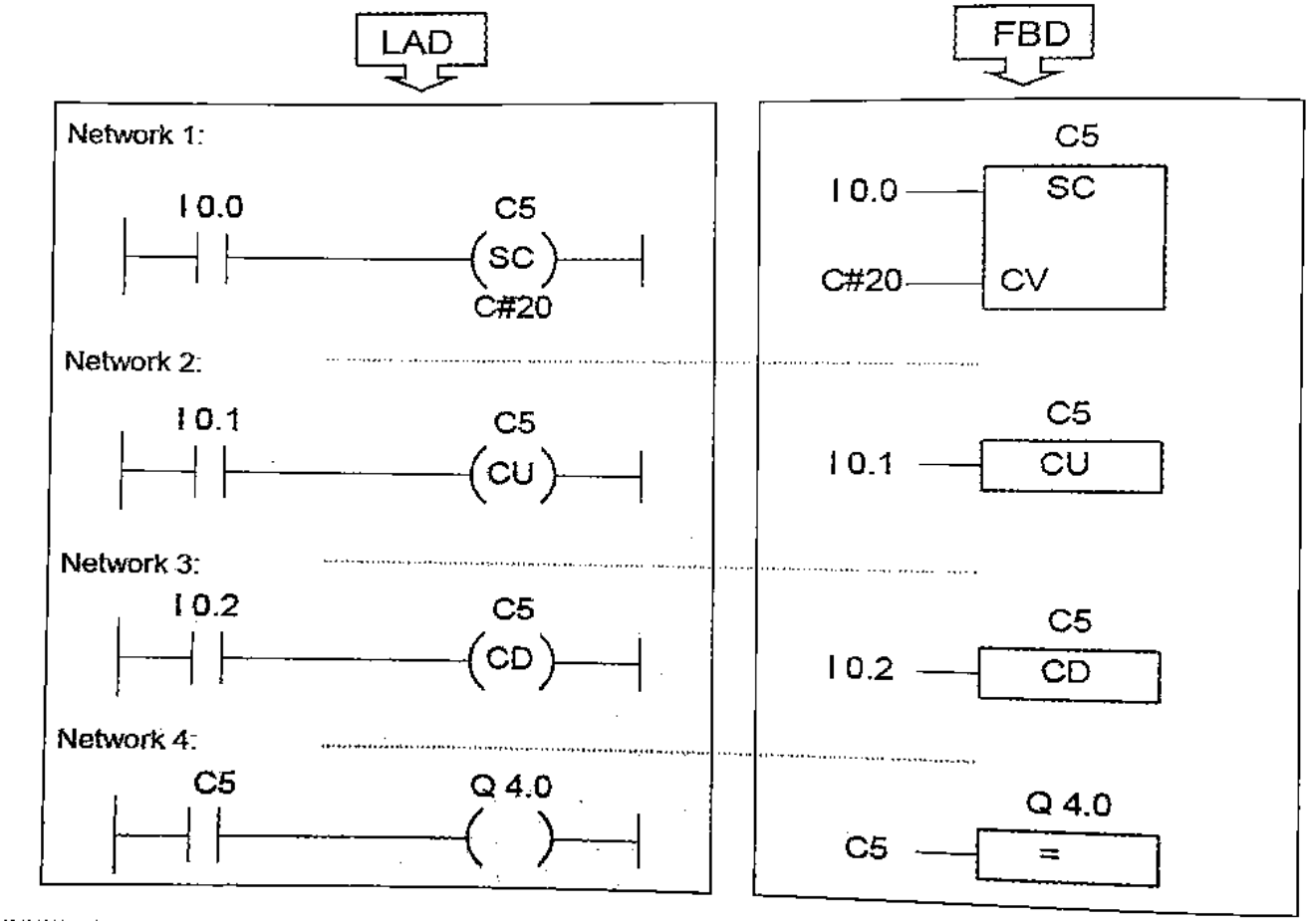
Network 1



Network 2

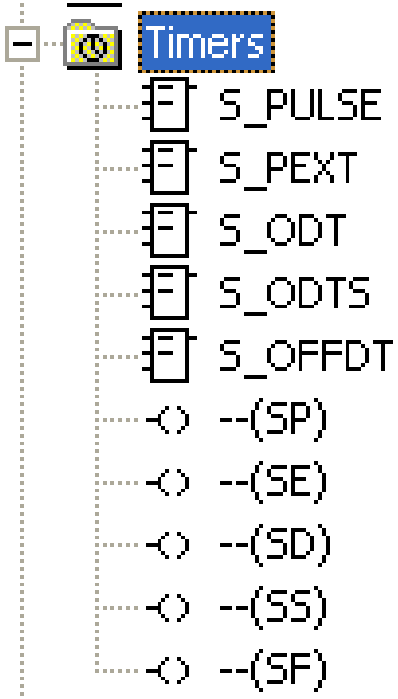


شمارنده های بیتی



با فعال شدن I0.0 مقدار ۲۰ به خروجی کانتر C5 انتقال پیدا می کند
با فعال شدن I0.1 یک عدد به مقدار شمارش شده توسط کانتر C5 اضافه می شود
با فعال شدن I0.2 یک عدد از مقدار شمارش شده توسط کانتر C5 کم می شود
در صورتی که کانتر C5 فعال باشد (مقداری بیشتر از صفر داشته باشد) ، Q4.0 فعال می شود

تایمرها



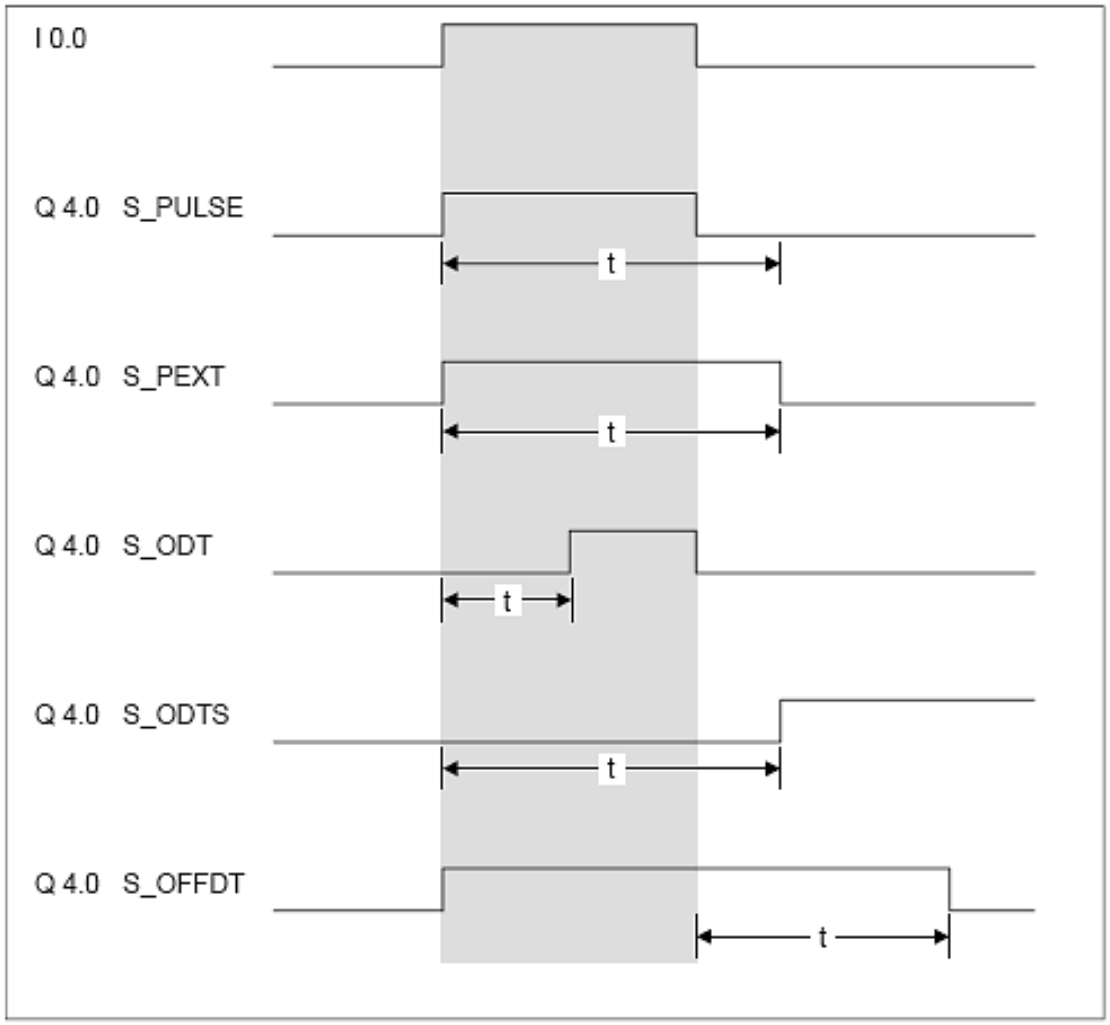
تایمر	توضیحات
S-Pulse Pulse Timer	تایمر پالس
S-PEXT Extended Pulse Timer	تایمر پالس ادامه دهنده
S-ODT On Delay Timer	تایمر تاخیر در وصل
S-ODTS Retentive On-Delay Timer	تایمر تاخیر در وصل ماندگار
S-OFFDT Of Delay Timer	تایمر تاخیر در قطع

تایمرهای **S7** شامل دو دسته هستند:
۱- تایمرهای پایه که در **S5** موجود بودند و از آنها در **S7** نیز استفاده شده است.
۲- تایمرهایی که از استاندارد **IEC** به **S7** آمده اند.(تایمرهای سیستمی)
تایمرهای نوع اول از **10 MS**تا **2H46M30S** یا **9990** ثانیه می شمارند
تایمرهای نوع دوم از **1 MS**تا **24D20H31M23S648MS** می شمارند
فرمت تایم **TV:S5T#aHbMcSdMS** می باشد مثلاً **S5T#2H15M**

تایمرها

تایمر	توضیحات
S-Pulse Pulse Timer	**با یک شدن ورودی بکار می افتد و خروجی آن تا وقتی شرایط زیر برقرار است ۱ باقی می ماند:** **۱- زمان t تمام نشده باشد. ۲-ورودی ۱ باشد**
S-PEXT Extended Pulse Timer	**با یک شدن ورودی بکار می افتد و خروجی آن تا وقتی زمان t تمام نشده باشد ۱ باقی می ماند حتی اگر ورودی صفر شود.**
S-ODT On Delay Timer	**با یک شدن ورودی بکار می افتد و خروجی آن ابتدا صفر و پس از گذشت زمان t بشرط اینکه ورودی هنوز ۱ باشد، یک می شود و با صفر شدن خروجی صفر می گردد.**
S-ODTS Retentive On-Delay Timer	**با یک شدن ورودی بکار می افتد و خروجی آن ابتدا صفر و پس از گذشت زمان t حتی اگر ورودی صفر شده باشد، یک می شود و یک می ماند، تنها تایمری است که به Reset نیاز دارد.**
S-OFFDT Of Delay Timer	**با یک شدن ورودی بکار می افتد ولی زمان t از وقتی شروع می شود که ورودی صفر شود، پس از آن به اندازه t خروجی یک باقی می ماند.**

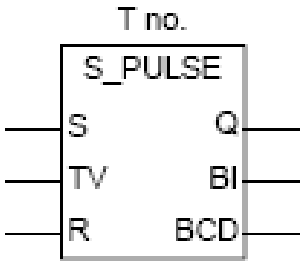
تایمرها



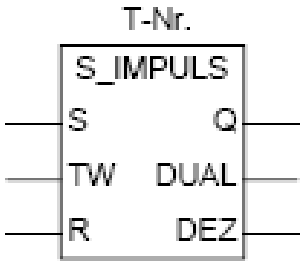
Pulse تایمر

همه انواع تایمرها پایه های یکسانی دارند:

English

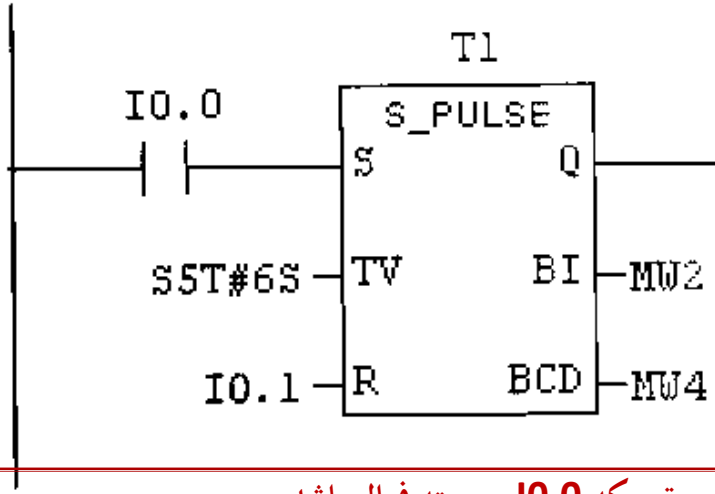


German

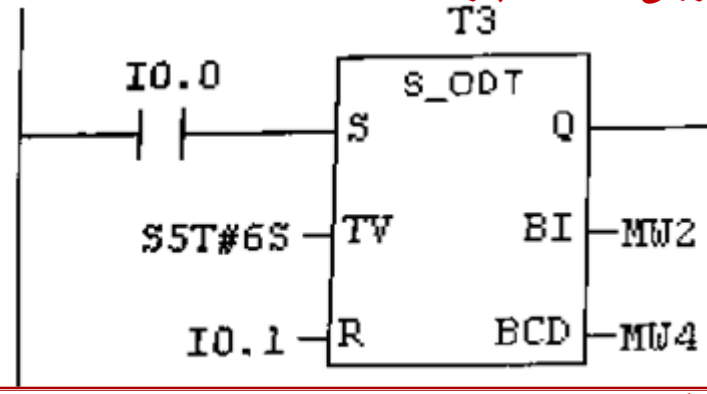


Parameter English	Parameter German	Data Type	Memory Area	Description
T no.	T-Nr.	TIMER	T	شماره تایمر
S	S	BOOL	I, Q, M, L, D	فعال کننده تایمر
TV	TW	S5TIME	I, Q, M, L, D	زمان تایمر
R	R	BOOL	I, Q, M, L, D	ریست کننده تایمر
BI	DUAL	WORD	I, Q, M, L, D	زمان باقی مانده کار تایمر به فرم **Integer**
BCD	DEZ	WORD	I, Q, M, L, D	زمان باقی مانده کار تایمر به فرم **BCD**
Q	Q	BOOL	I, Q, M, L, D	خروجی تایمر بصورت بیتی

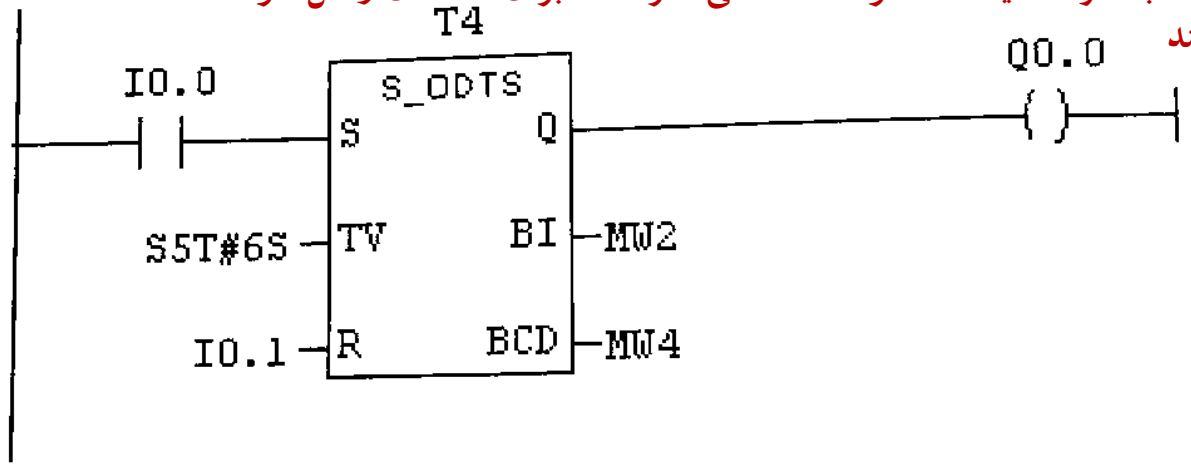
با برقراری ورودی I0.0 تایمر T1 به مدت ۶ ثانیه فعال خواهد شد
و خروجی Q4.0 نیز فعال می شود



با برقراری ورودی I0.0 تایمر T3 بعد از ۶ ثانیه فعال خواهد شد در صورتی که I0.0 پیوسته فعال باشد
و خروجی Q4.0 را فعال می کند



با برقراری ورودی I0.0 تایمر T4 بعد از ۶ ثانیه فعال خواهد شد حتی اگر I0.0 برای لحظه ای وصل شود
و خروجی Q4.0 را فعال می کند



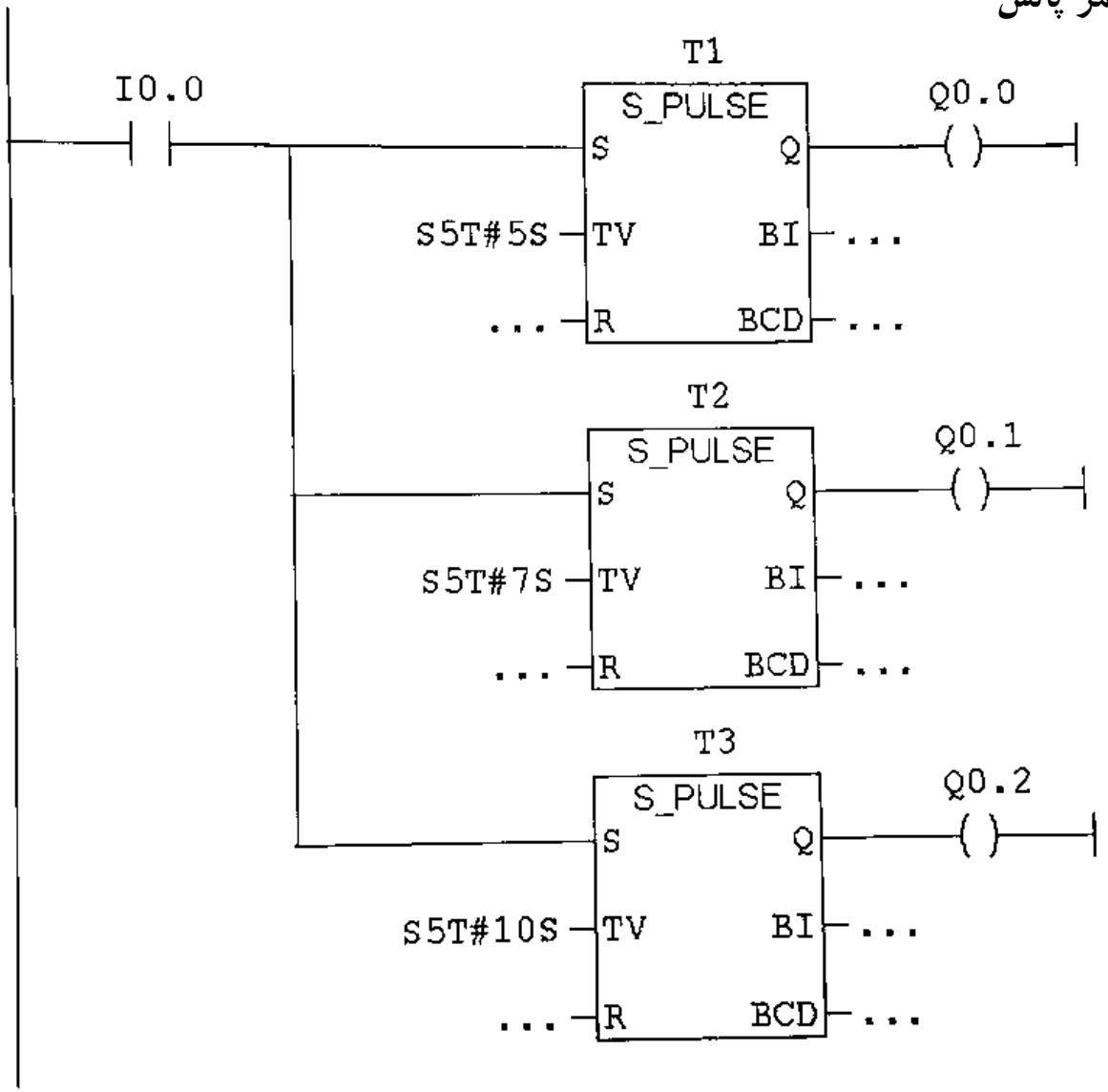
تهیه کنند

مثال برنامه‌ای بنویسید که با فعال شدن ورودی I0.0، سه لامپ (خروجی) فعال شده و به‌ترتیب زیر خاموش شوند:

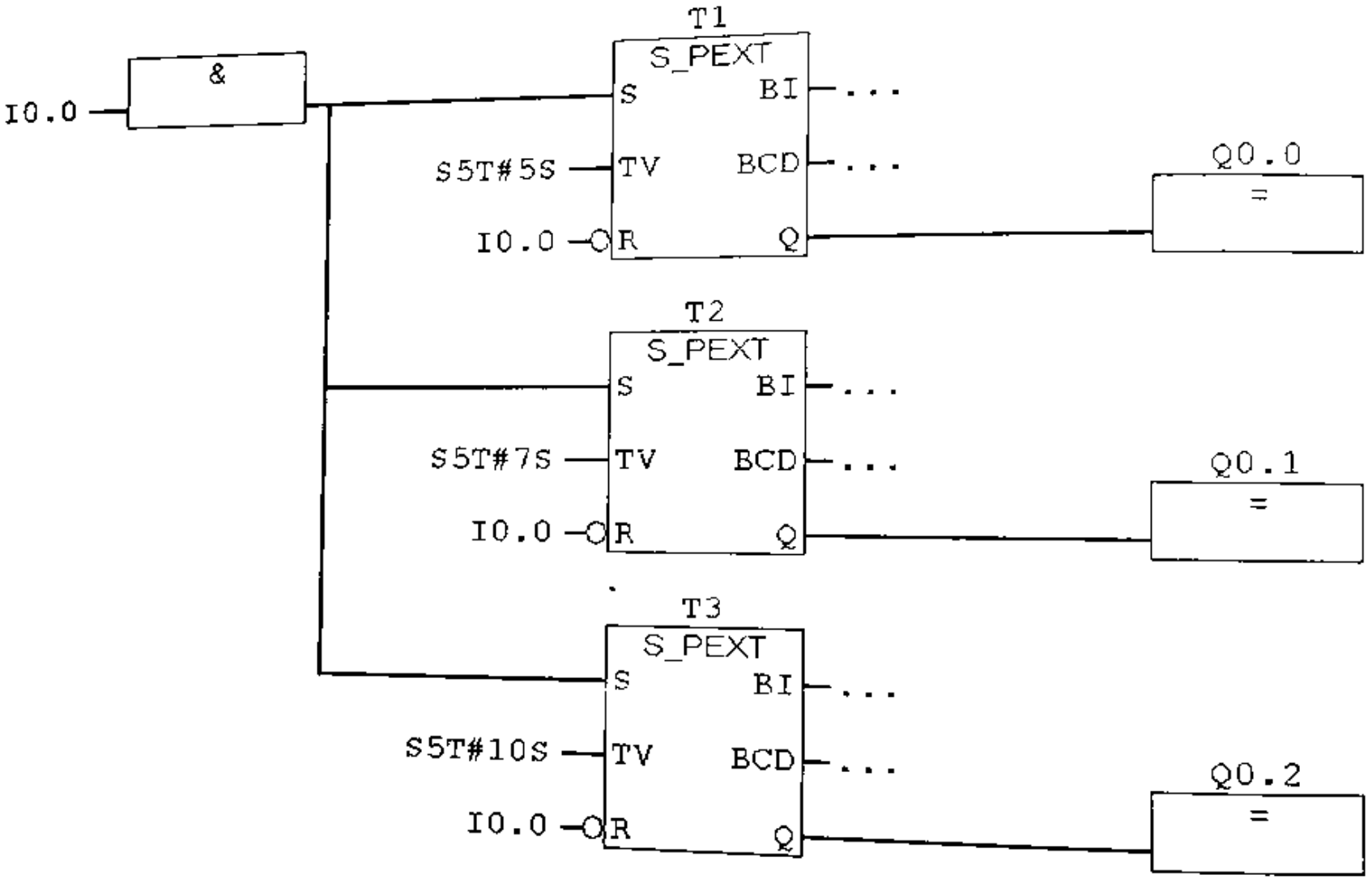
- **لامپ اول (Q0.0):** ۵ ثانیه
- **لامپ دوم (Q0.1):** ۷ ثانیه
- **لامپ سوم (Q0.2):** ۱۰ ثانیه

اگر در بین کار ورودی قطع شد هر سه لامپ خاموش شوند.

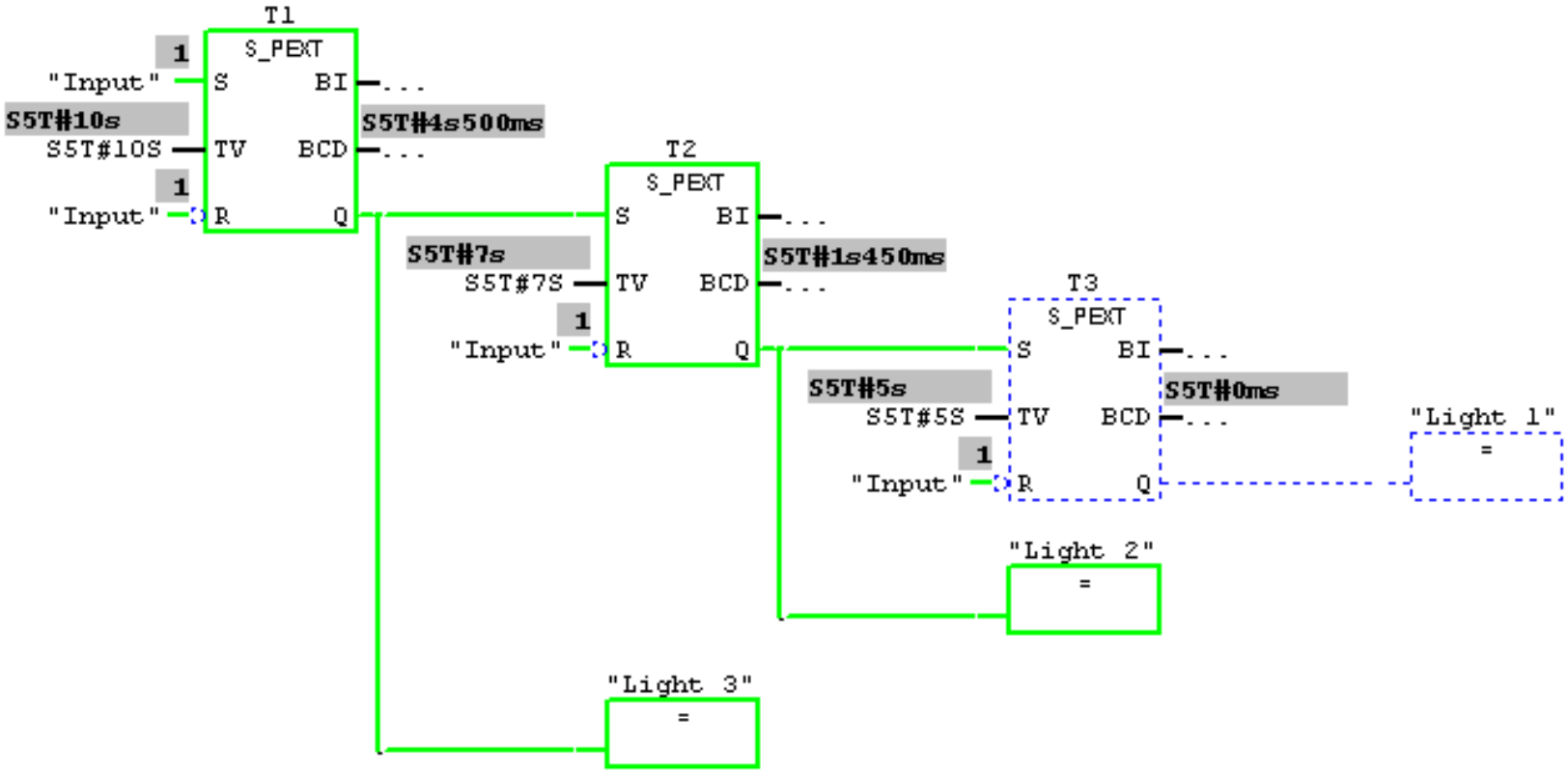
جواب: با استفاده از تایمر پالس



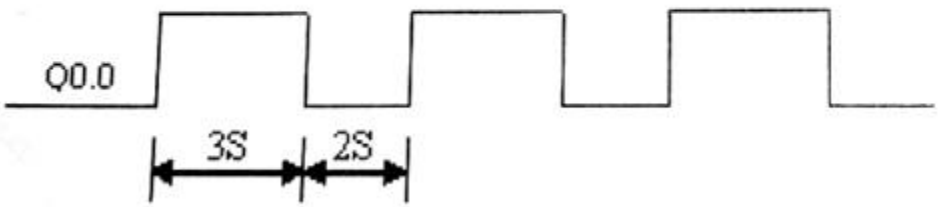
جواب: با استفاده از تایمر پالس ماندگار



جواب: با استفاده از تایمر پالس ماندگار

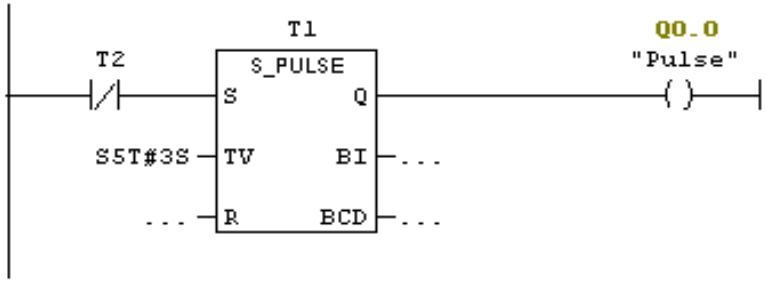


برنامه ای بنویسید که پالس زیر را تولید نماید.



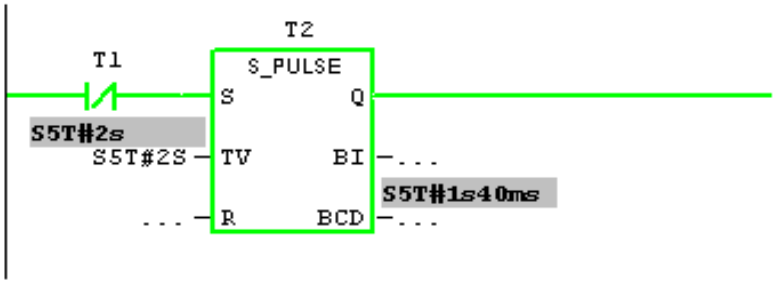
Network 5 : Title:

Comment:



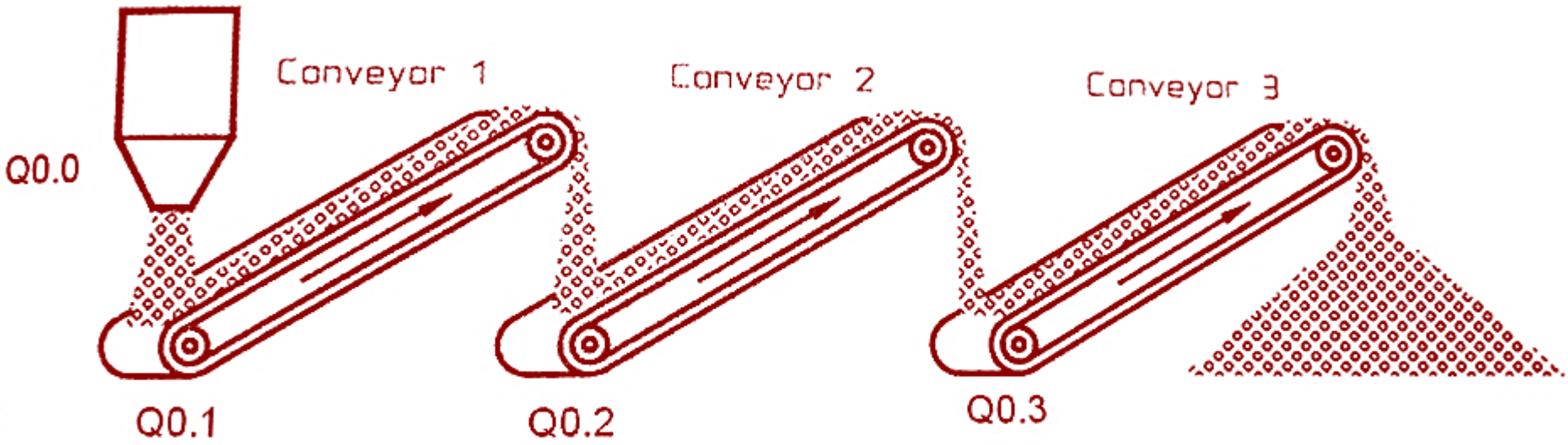
Network 6 : Title:

Comment:



در سیستم حمل مواد سه نوار نقاله مانند شکل وجود دارد. وقتی اپراتور شستی استارت I0.0 را فعال کرد لازم است سیستم به‌صورت زیر به‌ترتیب استارت شود :

- ابتدا نوار ۳ روشن شود.
- پس از روشن شدن نوار ۳، با ۱۰ ثانیه تأخیر نوار ۲ روشن شود.
- پس از روشن شدن نوار ۲، با ۱۰ ثانیه تأخیر نوار ۱ روشن شود.
- پس از روشن شدن نوار، با ۵ ثانیه تأخیر دریچه بونکر باز شود.

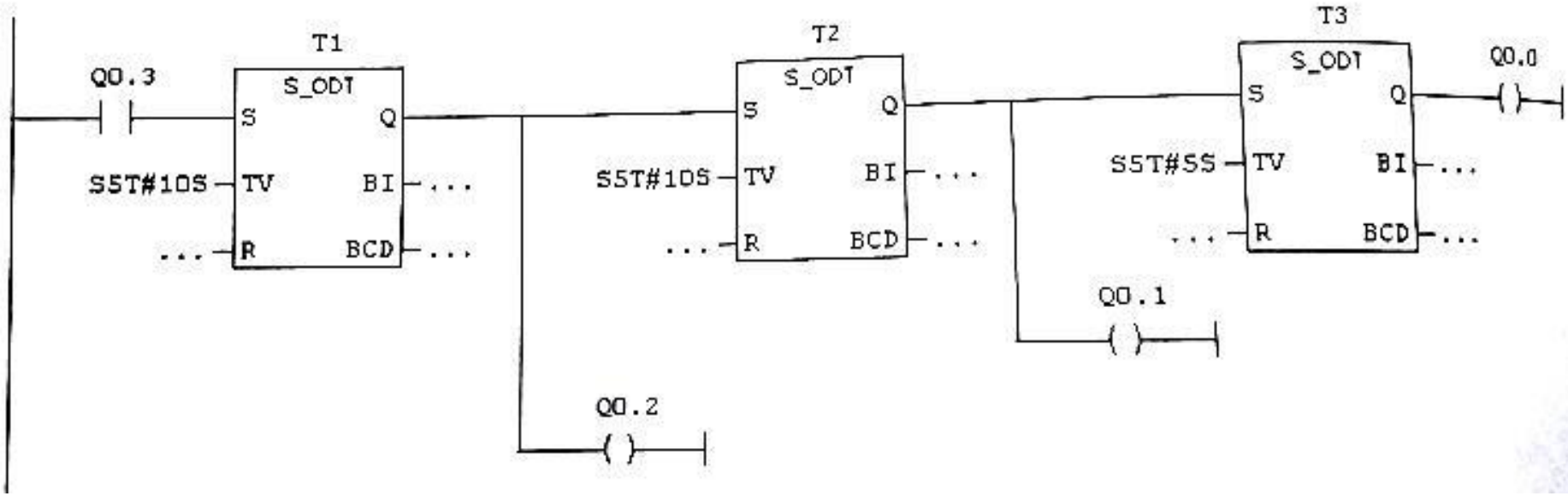


کنترل ترتیبی نوار نقاله‌های حمل مواد

OB1 : "Main Program Sweep (Cycle)"

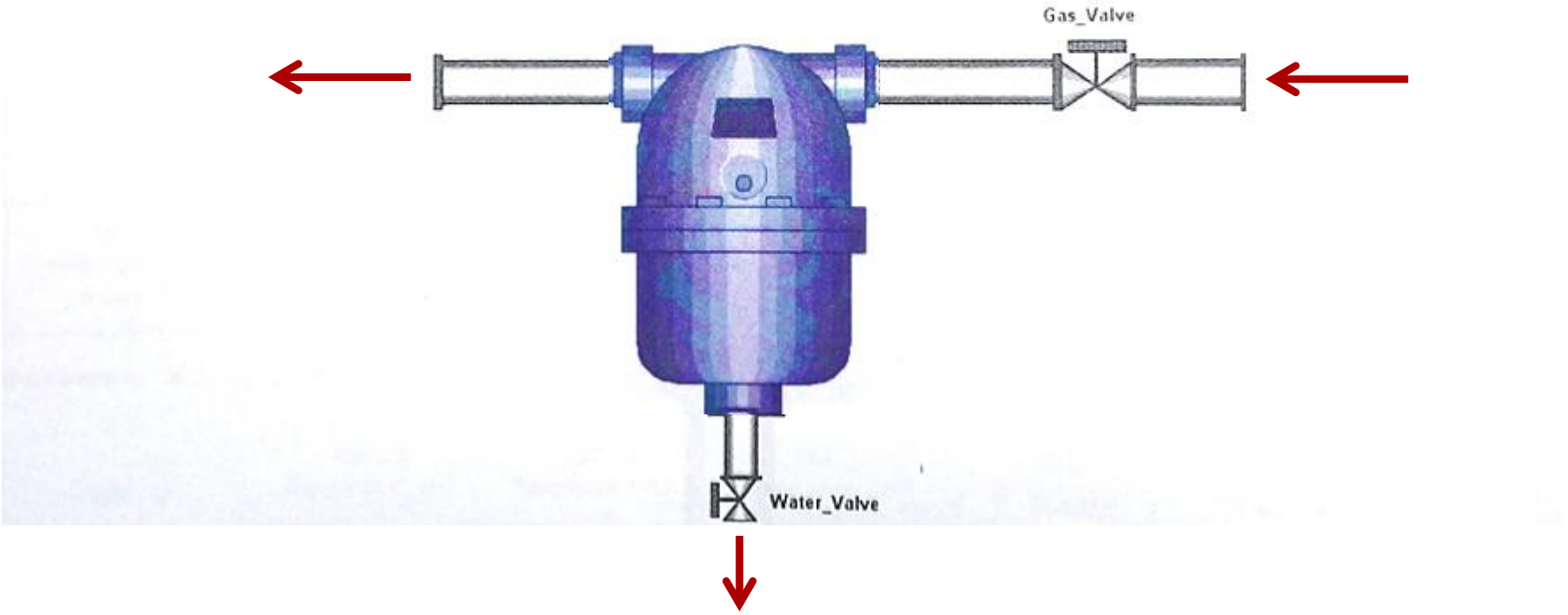
Network 1 : Title:

Network 2 : Title:



مثال کنترل سیکل کاری Dryer

در فرآیندی گاز مرطوب وارد خشک‌کن شده و گاز خشک خارج می‌شود. پس از ۲ ساعت لازم است مسیر گاز بسته شده و مسیر تخلیه آب به مدت ۲۰ دقیقه باز و سپس بسته شود. با بسته‌شدن مسیر آب مجدداً مسیر گاز باز شده و این سیکل مرتباً تکرار شود.

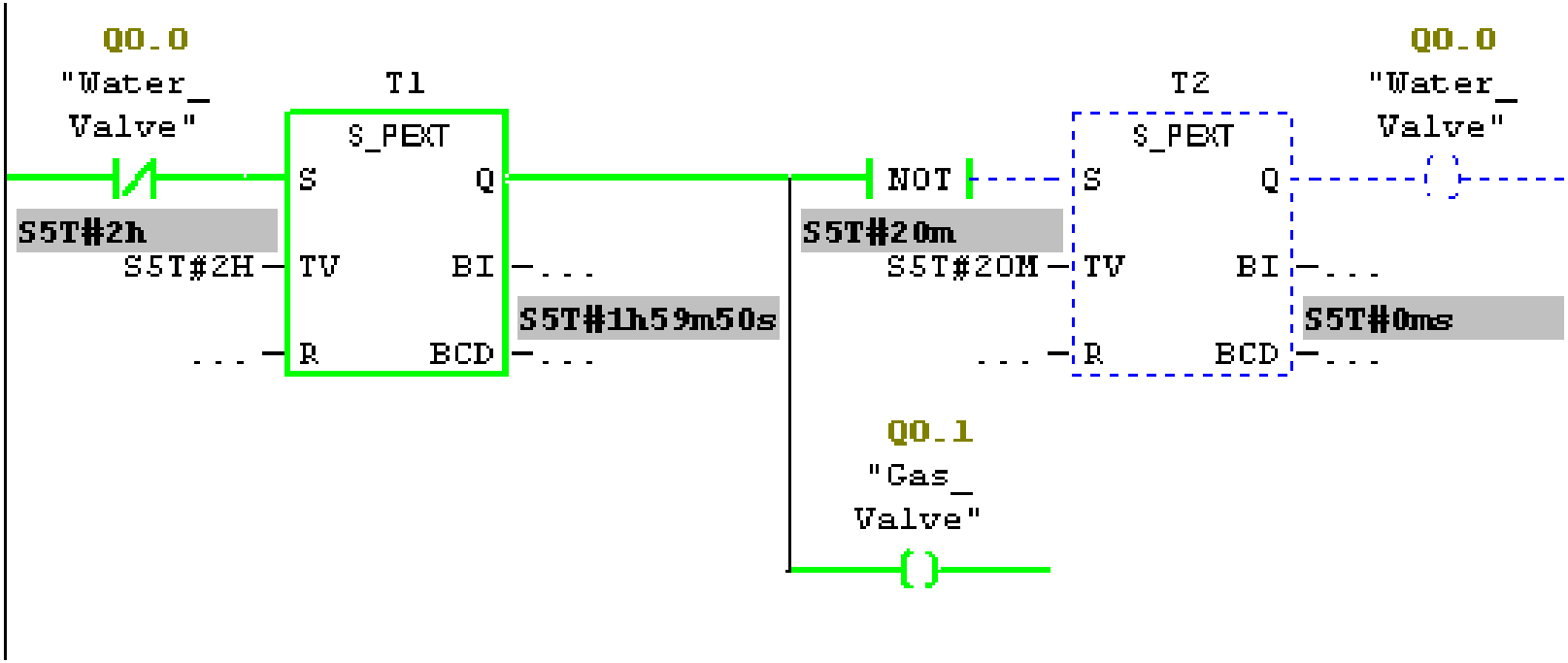


OB1 : "Main Program Sweep (Cycle)"

Comment:

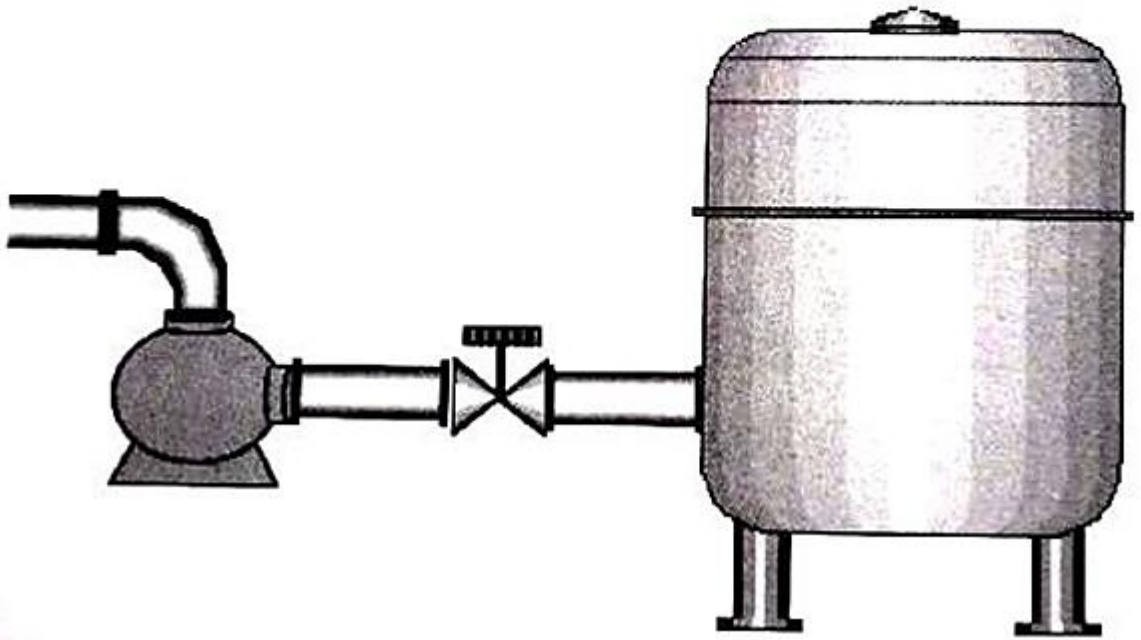
Network 1: Title:

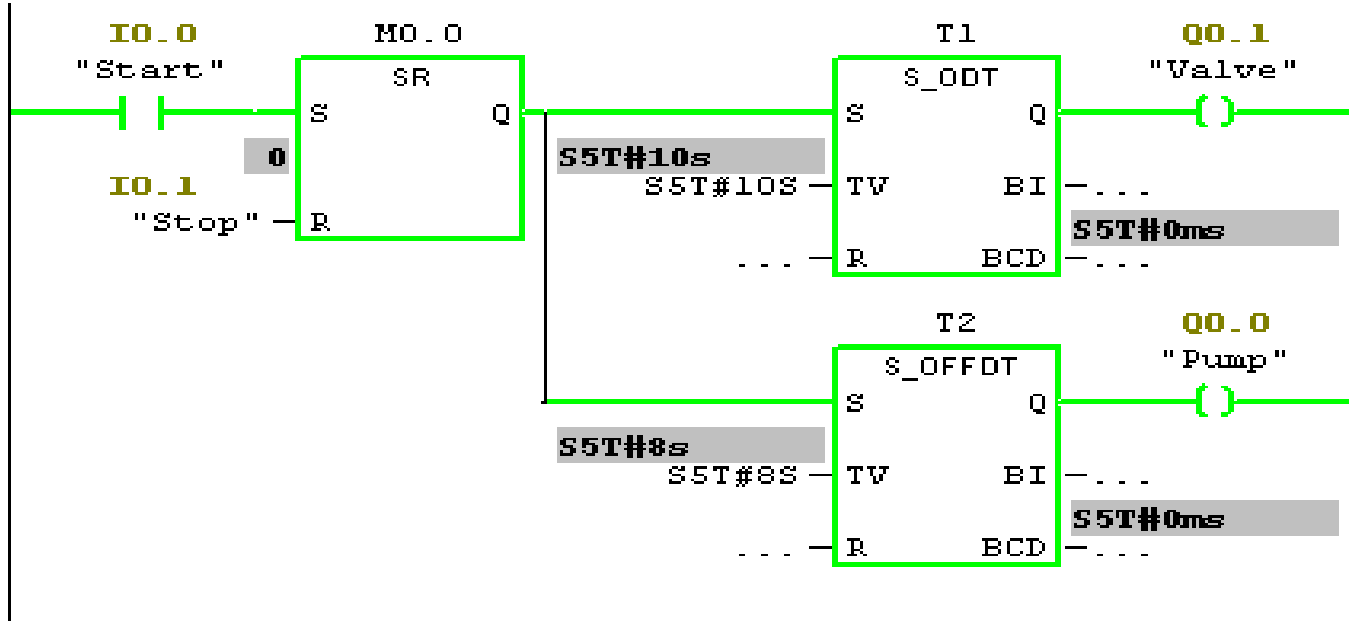
Comment:

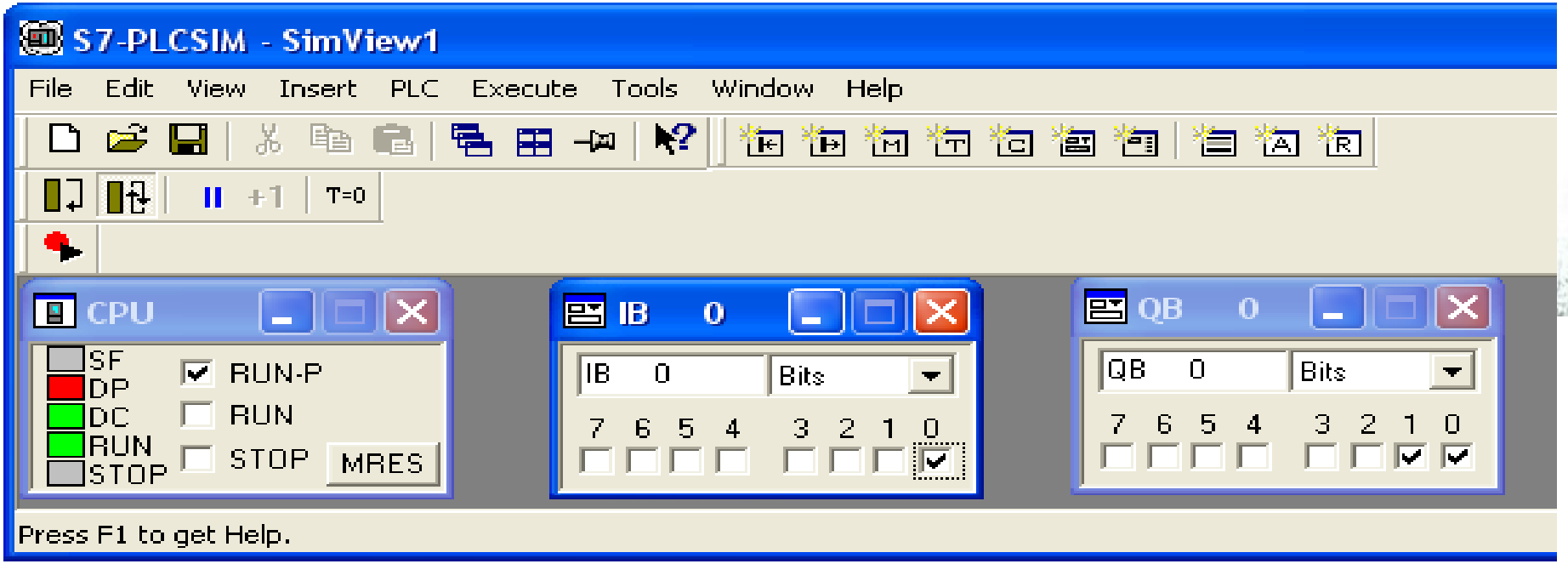


در فرآیندی از پمپ برای انتقال سیال به مخزن استفاده شده است و در مسیر انتقال از پمپ تا مخزن یک ولو On/Off به‌کار رفته است. برای استارت و استپ شرایط زیر بایستی برآورده شود:

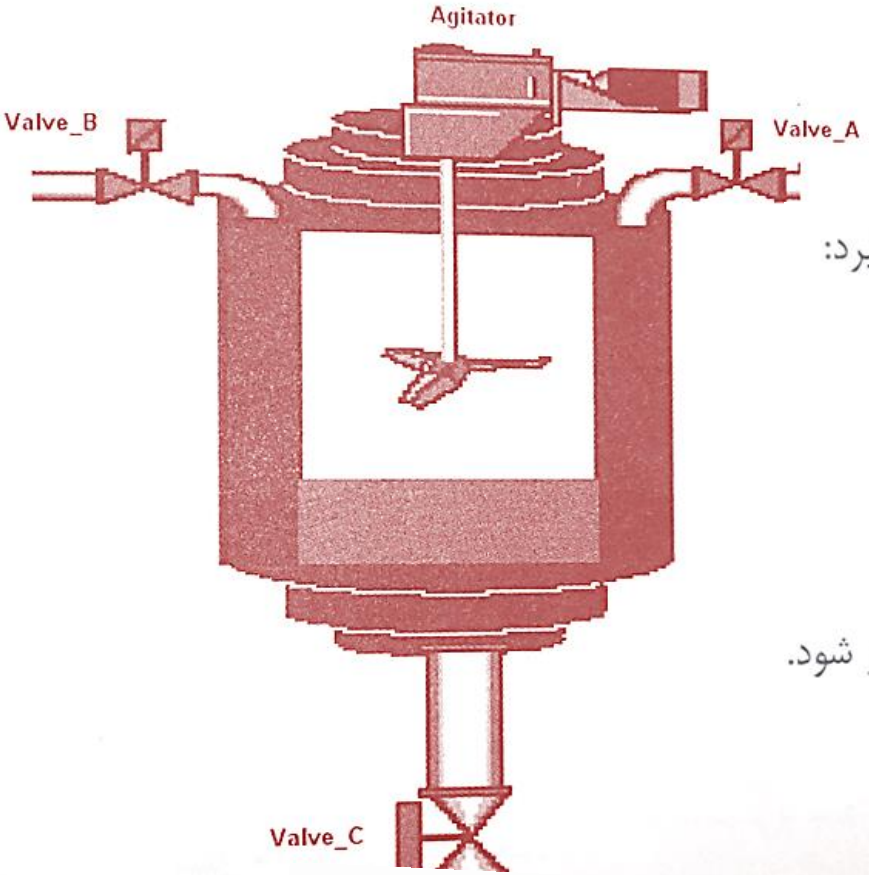
- با استارت اپراتور ابتدا پمپ بکار بیفتد و با ۱۰ ثانیه تأخیر ولو باز شود (تا موتور زیر بار راه‌اندازی نشود)
- با استپ اپراتور ابتدا ولو بسته شود، سپس با ۸ ثانیه تأخیر پمپ از کار بیفتد.




I0.0
"Start"
M0.0
SR
S
Q
0
I0.1
"Stop"
R
T1
S_ODT
S
Q
S5T#10s
S5T#10S
TV
BI
...
R
BCD
...
Q0.1
"Valve"
S5T#0ms
T2
S_OFFDT
S
Q
S5T#8s
S5T#8S
TV
BI
...
R
BCD
...
Q0.0
"Pump"
S5T#0ms


S7-PLCSIM - SimView1
File
Edit
View
Insert
PLC
Execute
Tools
Window
Help
T=0
CPU
SF
DP
DC
RUN
STOP
RUN-P
RUN
STOP
MRES
IB 0
IB 0
Bits
7 6 5 4 3 2 1 0
QB 0
QB 0
Bits
7 6 5 4 3 2 1 0
Press F1 to get Help.

مثال



وقتی اپراتور شستی استارت را فشار دهد مراحل زیر به‌ترتیب انجام گیرد:

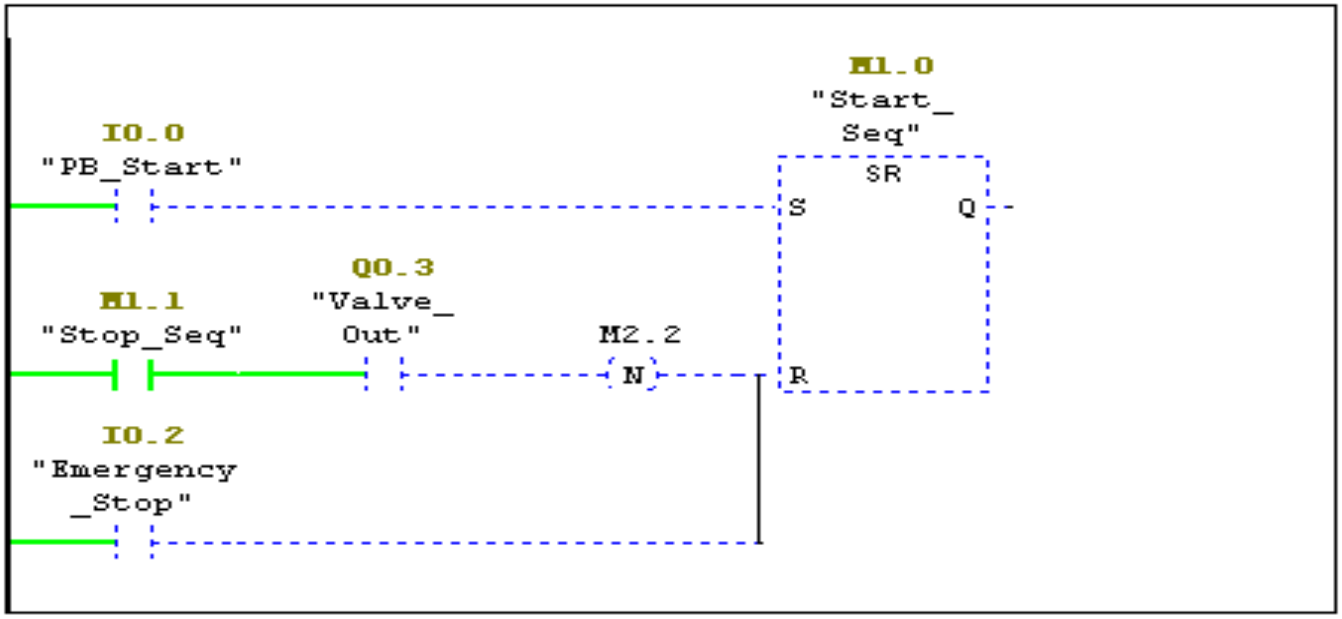
۱. ولو مسیر A به‌مدت ۲۰ ثانیه باز سپس بسته شود.
۲. ولو مسیر B به‌مدت ۱۵ ثانیه باز سپس بسته شود.
۳. موتور همزن به‌مدت ۱۲ ثانیه کار کند سپس از کار بیفتد.
۴. ولو خروجی C به‌مدت ۲۵ ثانیه باز سپس بسته شود.
۵. تا زمانی که اپراتور استپ را فعال نکرده سیکل از مرحله ۱ تکرار شود.

تذکر

- وقتی اپراتور شستی استپ را فشار دهد سیکل فعلی کامل شده ولی سیکل جدید شروع نشود.
- اگر اپراتور کلید قطع اضطراری را فعال کند سیکل در همان مرحله متوقف گردد.

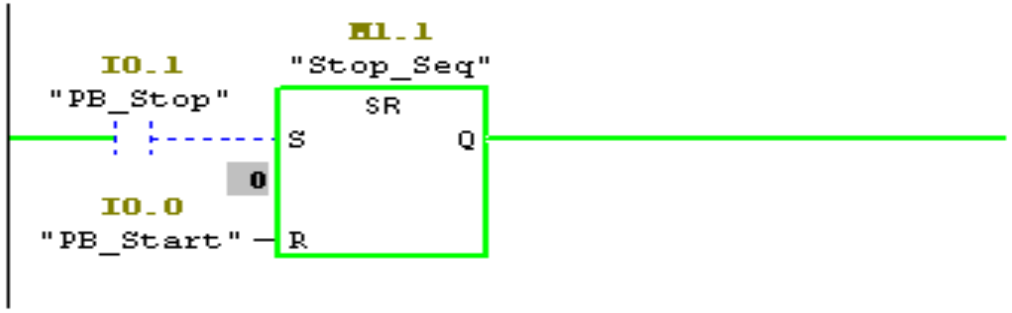
Network 1: Title:

Comment:



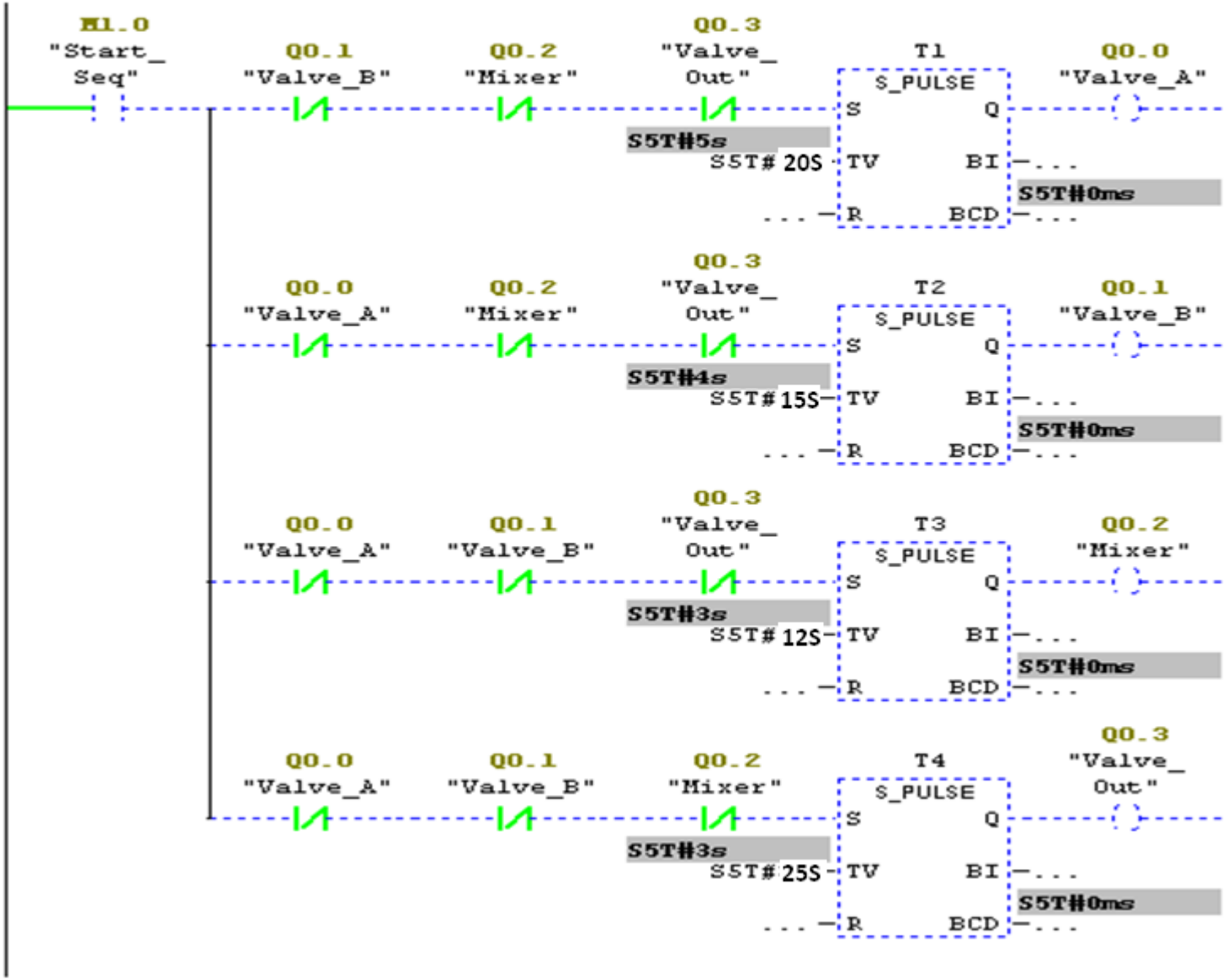
Network 2 : Title:

Comment:

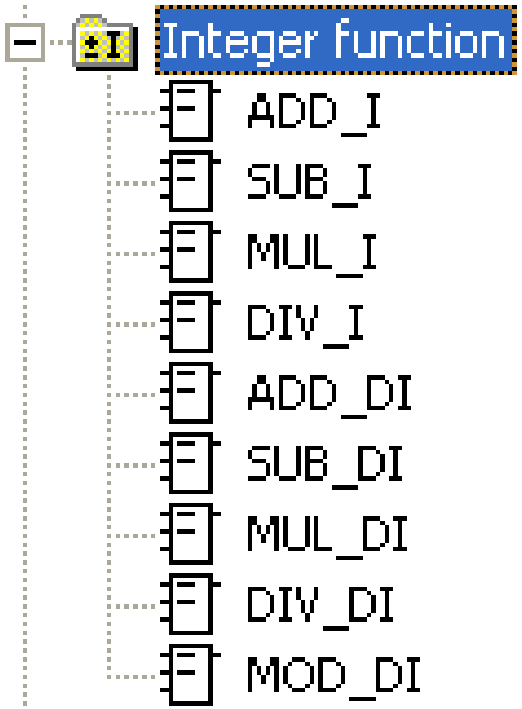


Network 3 : Title:

Comment:

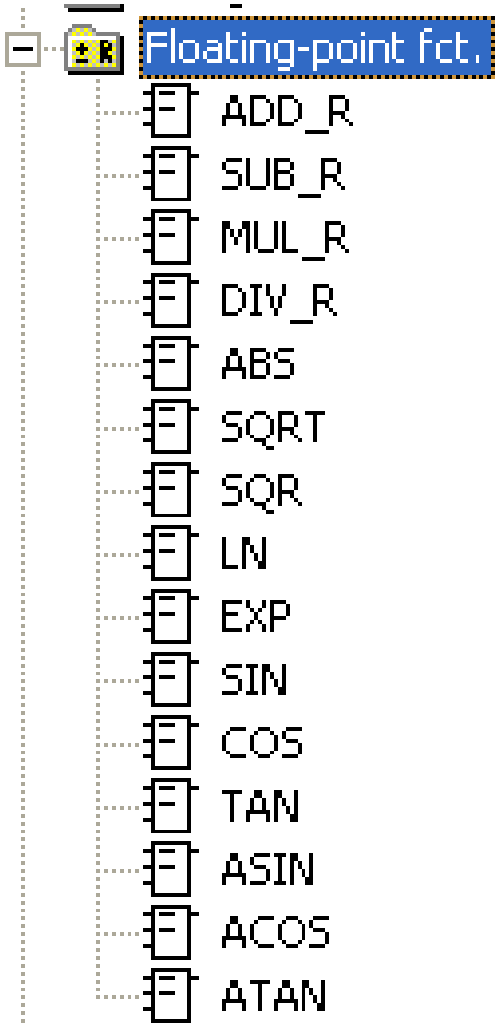


دستورات توابع صحیح



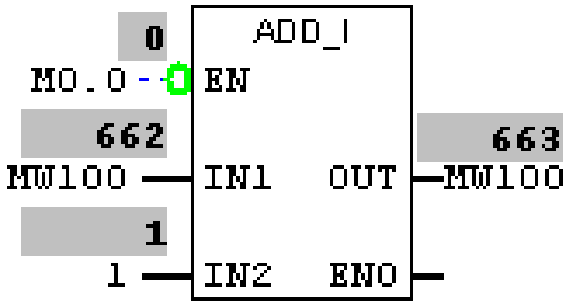
ADD جمع
SUB تفریق
MUL ضرب
DIV تقسیم
MOD باقی مانده صحیح تقسیم

دستورات توابع اعشاری

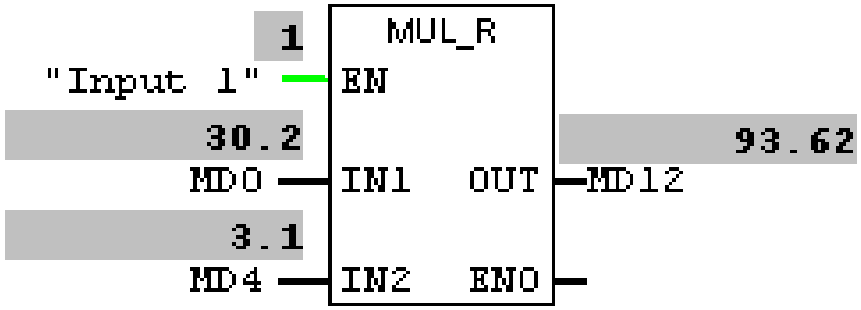


عنوان	مفهوم	شرح
ADD_R	Add Real	جمع‌کننده دو عدد Real
SUB_R	Subtract Real	تفریق‌کننده دو عدد Real
MUL_R	Multiply Real	ضرب‌کننده دو عدد Real
DIV_R	Divide Real	تقسیم‌کننده دو عدد Real
ABS	Absolute	قدر مطلق یک عدد Real
SQRT	Square Root	ریشه دوم (جذر) یک Real
SQR	Square	توان دوم یک عدد Real
LN	Natural Logarithm	لگاریتم طبیعی یک عدد Real
EXP	Exponential Value	عدد e (۲.۷۱۸۲۸) به‌توان دلخواه
SIN	Sine Value	سینوس یک عدد Real بر حسب رادیان
COS	Cosine Value	کسینوس یک عدد Real بر حسب رادیان
TAN	Tangent Value	تانژانت یک عدد Real بر حسب رادیان
ASIN	Arc Sine Value	آرک سینوس یک عدد Real
ACOS	Arc Cosine Value	آرک کسینوس یک عدد Real
ATAN	Arc Tangent Value	آرک تانژانت یک عدد Real

مثال

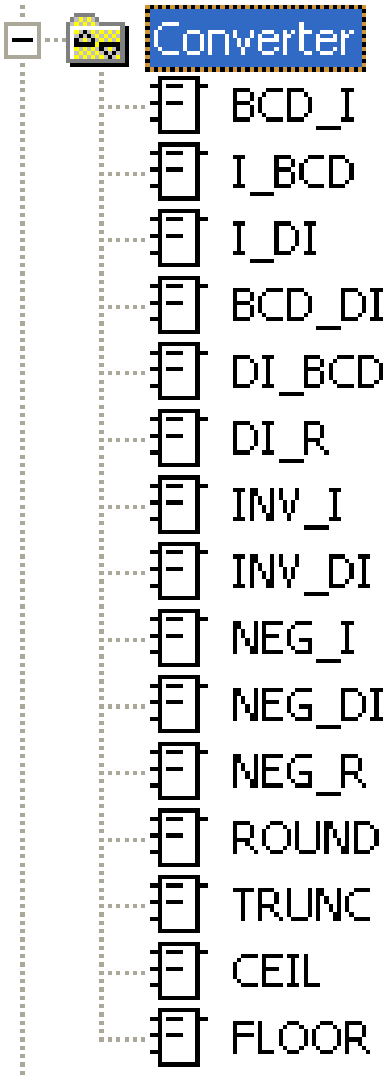


IN1 عدد صحیح اول(MW100)
IN2 عدد صحیح دوم(عددثابت ۱)
OUT نتیجه حاصل جمع(MW100)



IN1 عدد اعشاری اول(MD0)
IN2 عدد اعشاری دوم(MD4)
OUT نتیجه حاصل ضرب(MD12)

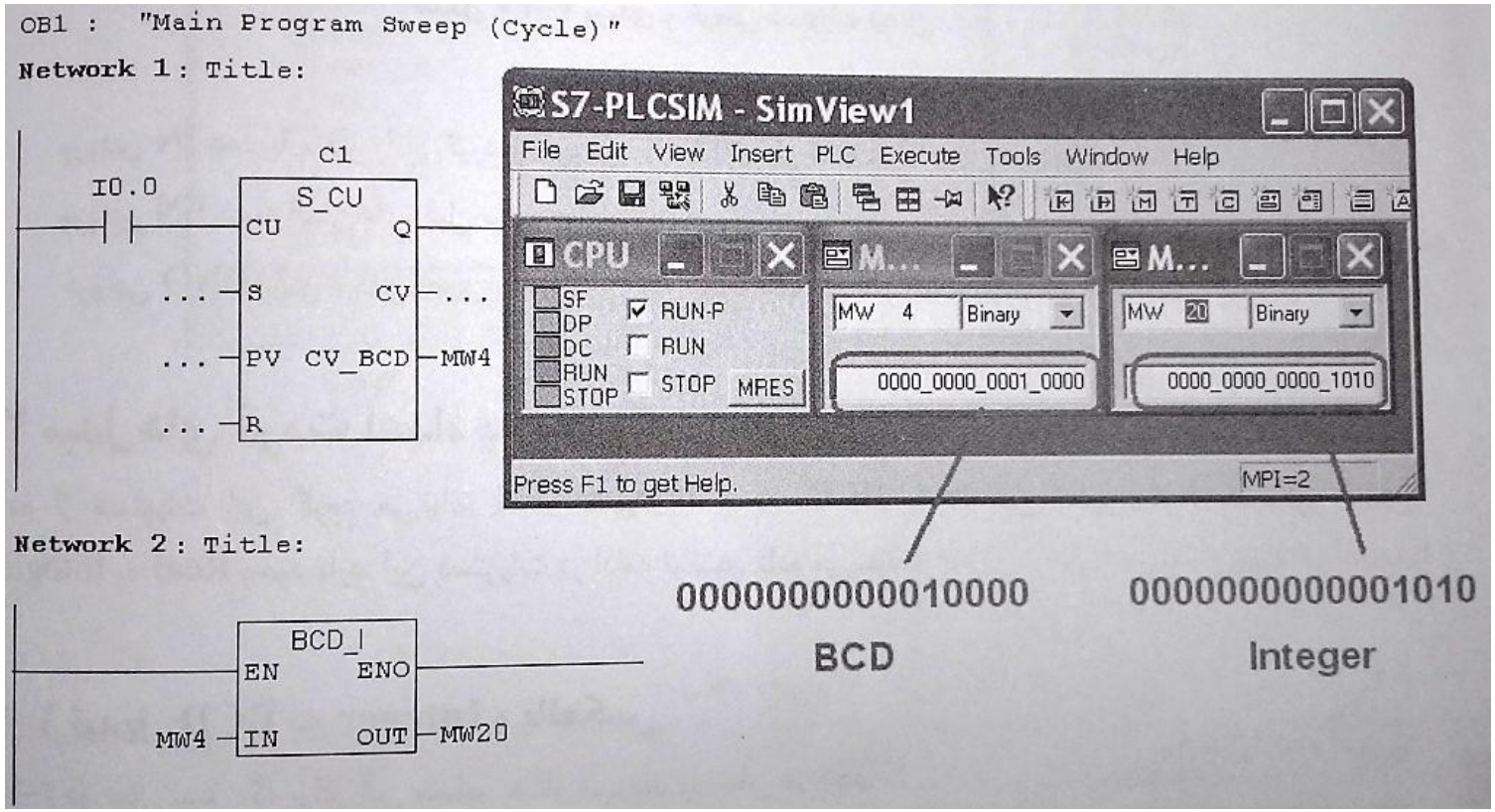
دستورات تبدیل نوع داده



دستور تبدیل	توضیحات
BCD_I	تبدیل دیتا تایپ BCD به Integer
I_BCD	تبدیل دیتا تایپ Integer به BCD
I_DI	تبدیل دیتا تایپ Integer به Double Integer
BCD_DI	تبدیل دیتا تایپ BCD به Double Integer
DI_BCD	تبدیل دیتا تایپ Double Integer به BCD
DI_R	تبدیل دیتا تایپ Double Integer به Real
INV_I	متمم یک عدد صحیح ۱۶ بیتی (معکوس‌کردن بیت‌ها)
INV_DI	متمم یک عدد صحیح ۳۲ بیتی (معکوس‌کردن بیت‌ها)
NEG_I	متمم دوی عدد صحیح ۱۶ بیتی (قرینه عدد)
NEG_DI	متمم دوی عدد صحیح ۳۲ بیتی (قرینه عدد)
NEG_R	قرینه عدد اعشاری
ROUND	رند کردن عدد اعشاری به نزدیک‌ترین عدد صحیح
TRUNC	رند کردن عدد اعشاری به عدد صحیح
CEIL	رند کردن عدد اعشاری به عدد صحیح بالاتر
FLOOR	رند کردن عدد اعشاری به عدد صحیح پایین‌تر

دستورات تبدیل نوع داده

تبدیل BCD به Integer (BCD-I)



تبدیل عدد ۱۰ که BCD می باشد به معادل عدد صحیح آن

دستورات تبدیل نوع داده

تبدیل Integer به BCD (I-BCD)

بزرگترین عدد صحیح ۱۶ بیتی که توسط این تابع می تواند تبدیل به عدد BCD شود ۲۴۵۷ دسیمال و معادل BCD ۹۹۹ و 1001,1001,1001 باینری می باشد. در صورتی که ورودی این تبدیل کننده از ۲۴۵۷ بیشتر شود خروجی از دیدگاه BCD نامعتبر خواهد بود.

***نکته:** یک عدد BCD را همیشه می شود به یک عدد صحیح تبدیل کرد ول عدد صحیح را همواره نمی توان به BCD تبدیل نمود.وقتی عدد صحیح بصورت باینری شود و به قسمت های ۴ بیتی تفکیک شد در صورتی که معادل یک ۴ بیت از عدد ۹ بیشتر شود معادل BCD نخواهد داشت. مثلاً: ۱۰۰۰۱۰۰۰۱۱۱۱ معادل BCD ندارد.*

تبدیل BCD به Double Integer (BCD-DI)

اعداد BCD می توانند ۳۲ بیتی باشند، در این حالت ماکزیمم مقدار آنها 9999999 خواهد بود. برای تبدیل این عدد BCD به عدد صحیح بایستی از این مبدل استفاده شود.

دستورات تبدیل نوع داده

تبدیل Double Integer به BCD (DI-BCD)

این تبدیل یک عدد صحیح ۳۲ بیتی را در ورودی گرفته و آنرا به BCD بر می گرداند.

تبدیل Integer به Real

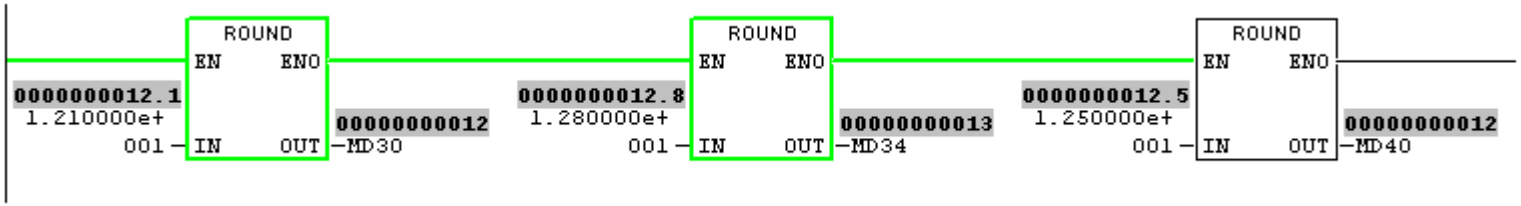
برای انجام این تبدیل دستور مستقیمی وجود ندارد برای این کار ابتدا می بایست عدد مورد نظر ابتدا از INT به DINT تبدیل و سپس به REAL تبدیل شود.

تبدیل Real به Integer

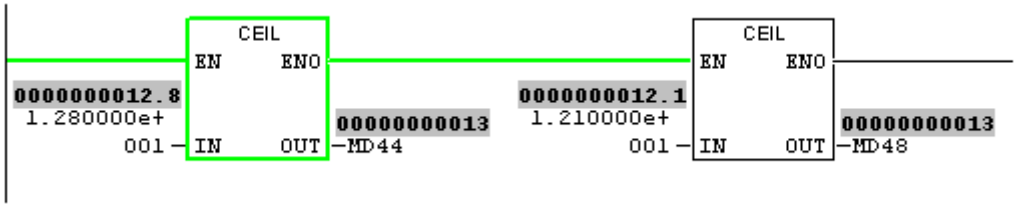
تبدیل مستقیمی برای تبدیل یک متغیر Real به عدد صحیح ۱۶ بیتی وجود ندارد ولی می توان آنرا از خروجی TRUNC گرفت و کافیست ۱۶ بیت با ارزش کمتر را جدا کنیم.

مبدل های Round کننده اعداد اعشاری

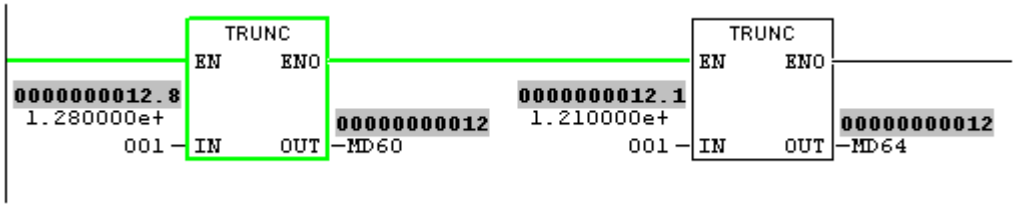
ROUND



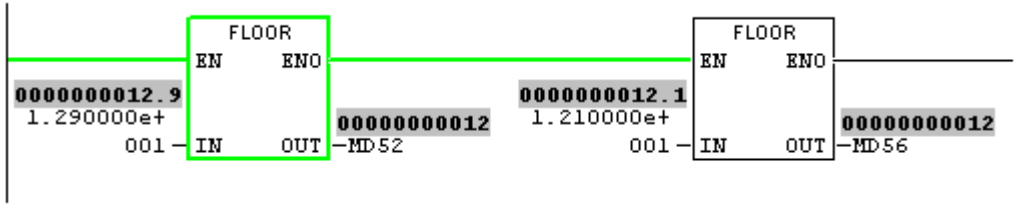
CEIL



FLOOR



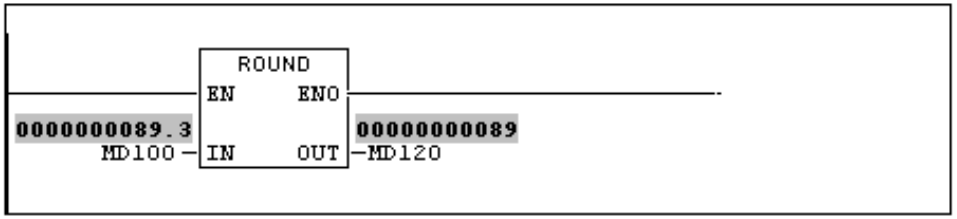
TRUNC

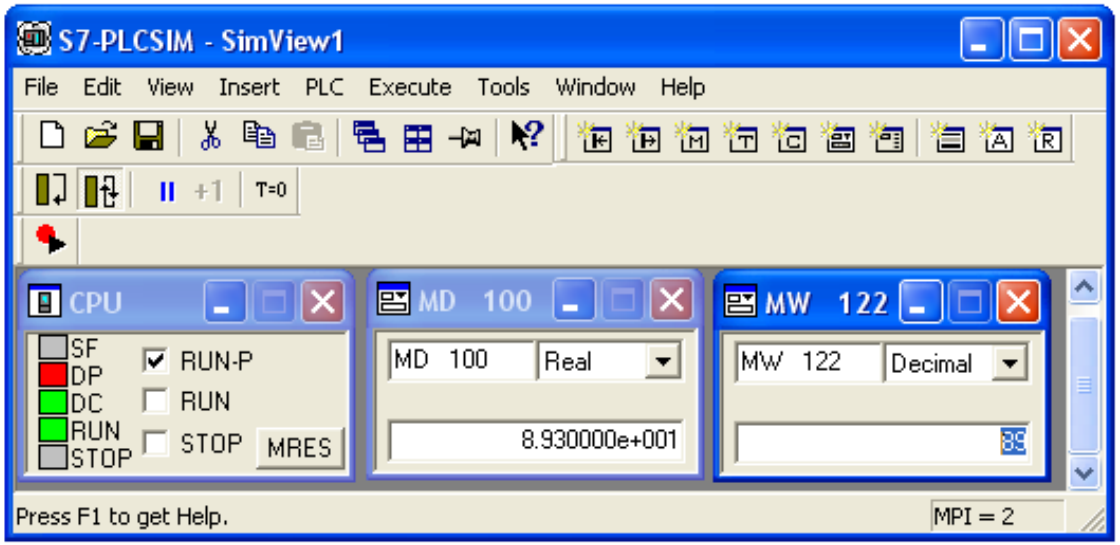


دستورات تبدیل نوع داده

تبدیل Real به Integer

تبدیل مستقیمی برای تبدیل یک متغیر **Real** به عدد صحیح ۱۶ بیتی وجود ندارد ولی می توان آنرا از **TRUNC** یا **ROUND** خروجی گرفت و کافیست ۱۶ بیت با ارزش کمتر را جدا کنیم.



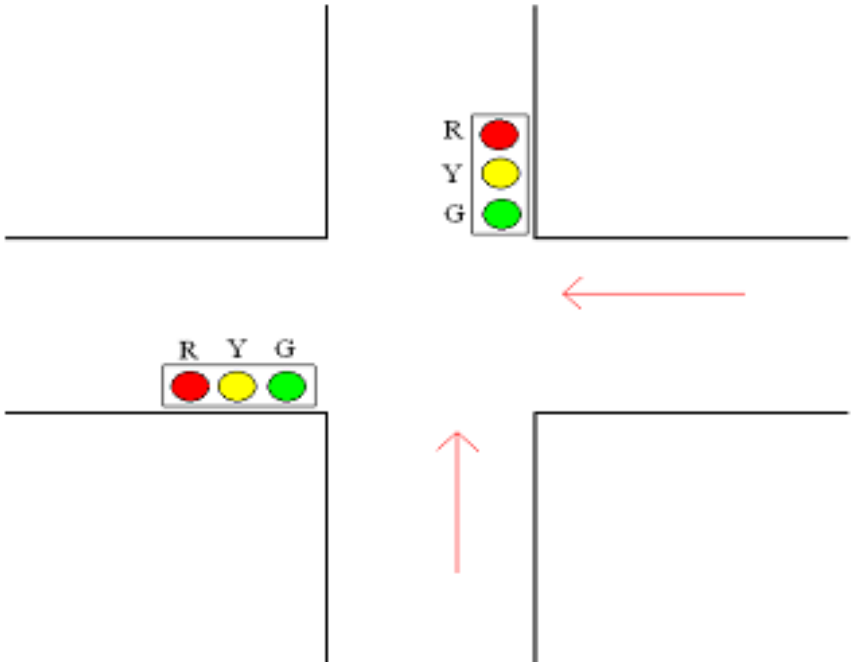


تمرینات

تمرین 1 :

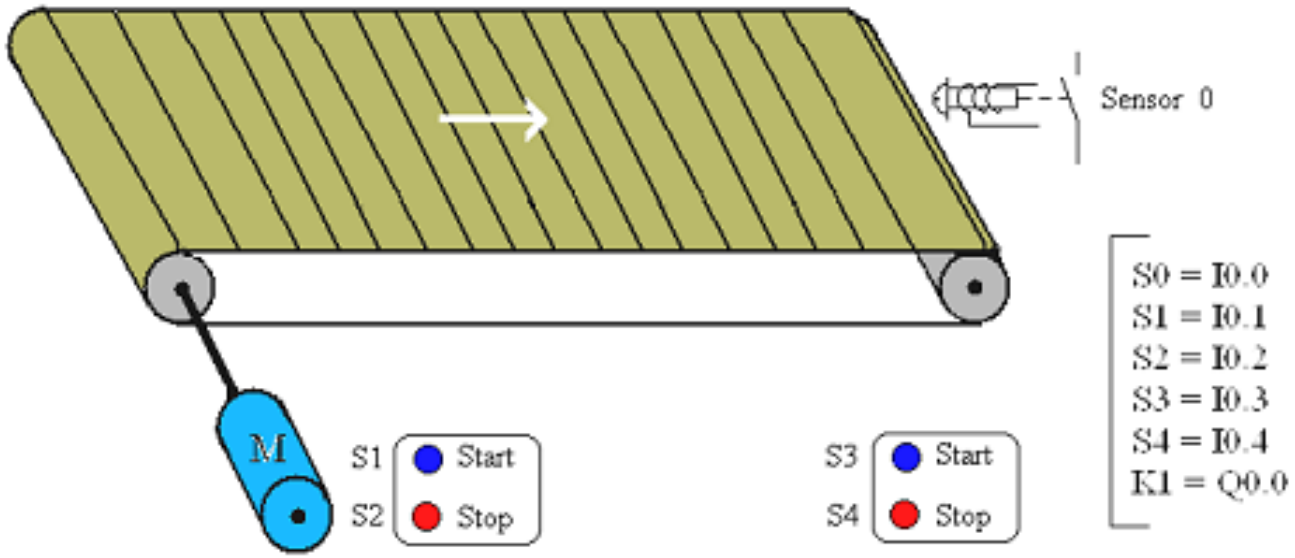
در یک چهارراه سیستم چراغ راهنمایی به صورت زیر است :

مدت زمان چراغ قرمز 30 ثانیه و مدت زمان چراغ زرد 5 ثانیه و مدت زمان چراغ سبز 30 ثانیه می باشد . سیستم کنترل آن را طراحی کنید .



تمرین 2 :

در شکل زیر دو کلید فشاری S1 و S2 به ترتیب برای استارت و استپ در سمت آغازین کانوایر وجود دارد ، همچنین در بخش انتهایی کانوایر دوکلید فشاری S3 و S4 برای استارت و استپ کانوایر تعبیه شده است . از طریق هر دو بخش آغازین و انتهایی کانوایر می توان آن را استپ یا استارت نمود . لازم به ذکر است که سنسور S0 برای توقف کانوایر هنگام رسیدن جسم به انتهای کانوایر نصب شده است . برنامه کنترلی این کانوایر را بنویسید .



تمرین 3 :

سیستم کنترل میز مسابقه سه نفره ای را به گونه ای طراحی کنید که اگر هر کدام از شاسی های S1، S2، S3 را که زودتر فشار داده شود ، چراغ مربوط به آن روشن شده و چراغهای دیگر عمل نکند .

تمرین 4 :

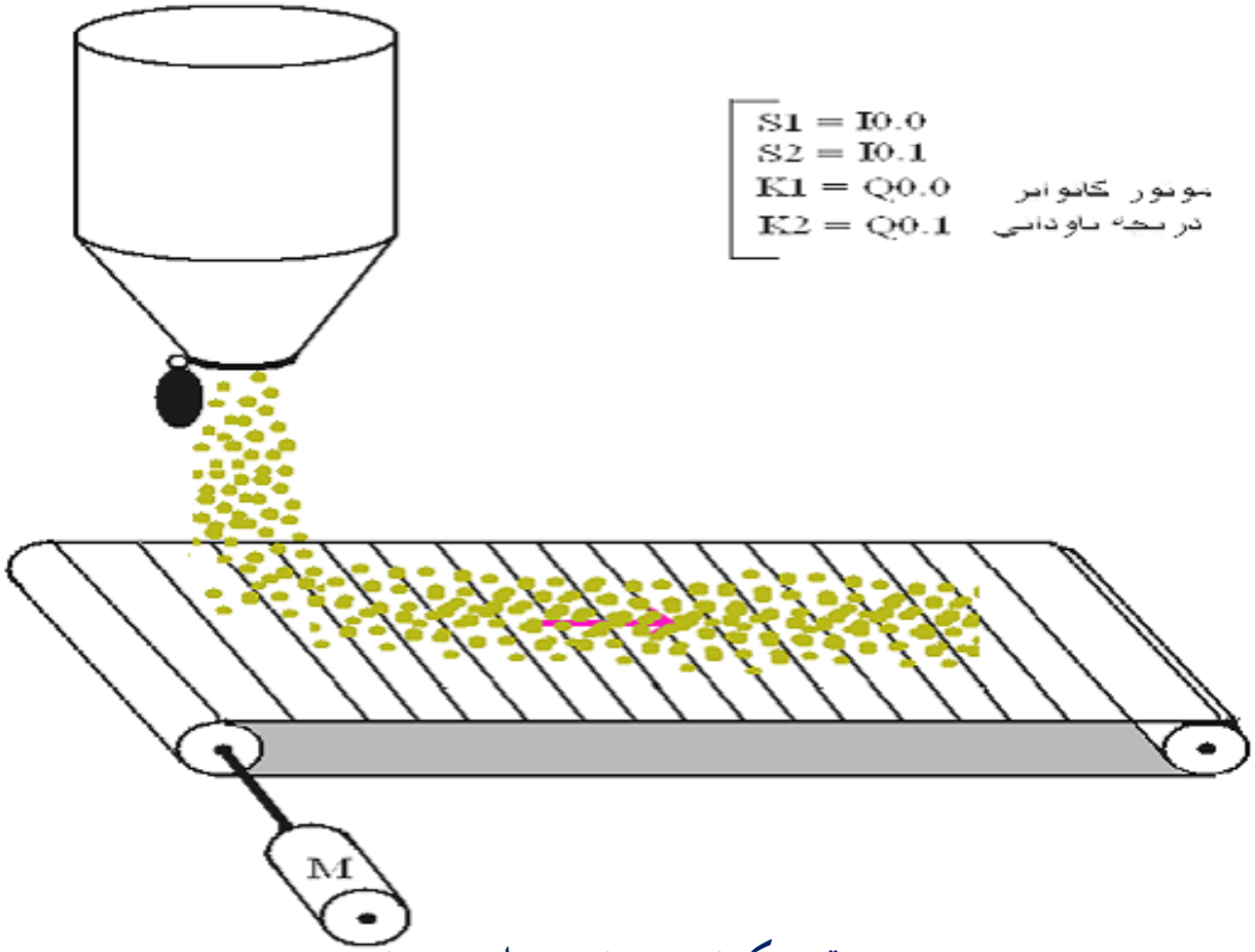
سیستمی را طراحی نمایید که با فشار دادن شاسی S1 کنتاکتور K1 مگنت کرده و در حالت مگنت باقی بماند ، زمانی که برای بار دوم شاسی S1 را فشار دادیم کنتاکتور K1 قطع شود .

تمرین 5 :

در یک ماشین ابزار سیستم کنترل به صورت زیر است :
با فشار دادن شاسی S1 موتور بطور دائم کار می کند و با فشار دادن شاسی S0 موتور خاموش می گردد . با فشار دادن شاسی S2 موتور بطور لحظه ای کار می کند و هنگامی که شاسی S2 رها شود موتور خاموش می گردد . (سیستم کنترل لحظه ای و دائم)

تمرین 6 :

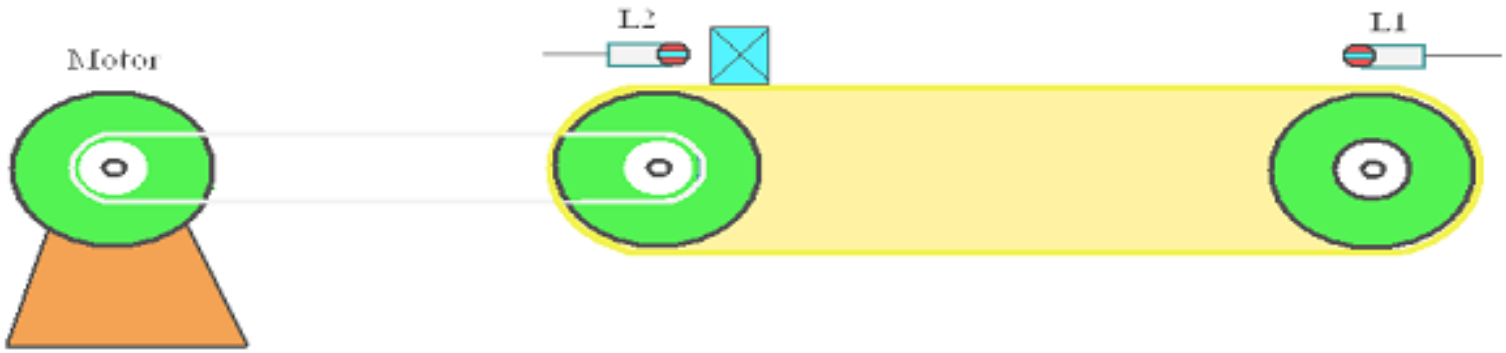
شکل زیر یک انتقال دهنده خرده سنگ از مخزن بر روی کانوایر می باشد . یک دریچه در قسمت ناودانی مخزن خرده سنگ وجود دارد و از آنجا خرده سنگ بر روی کانوایر ریخته می شود ، یک موتور حرکت کانوایر را کنترل می کند . اگر مکانیزم موتور کانوایر متوقف شود و یا مکانیزم عمل معیوب گردد ، دریچه ناودانی مخزن باید بسته شود . وقتی که دریچه مخزن انرژی می گیرد باز می گردد و با قطع انرژی آن بسته می شود شاسی فشاری S1 باعث خاموش شدن سیستم و شاسی S2 باعث روشن شدن سیستم می گردد .



تمرین 7 :

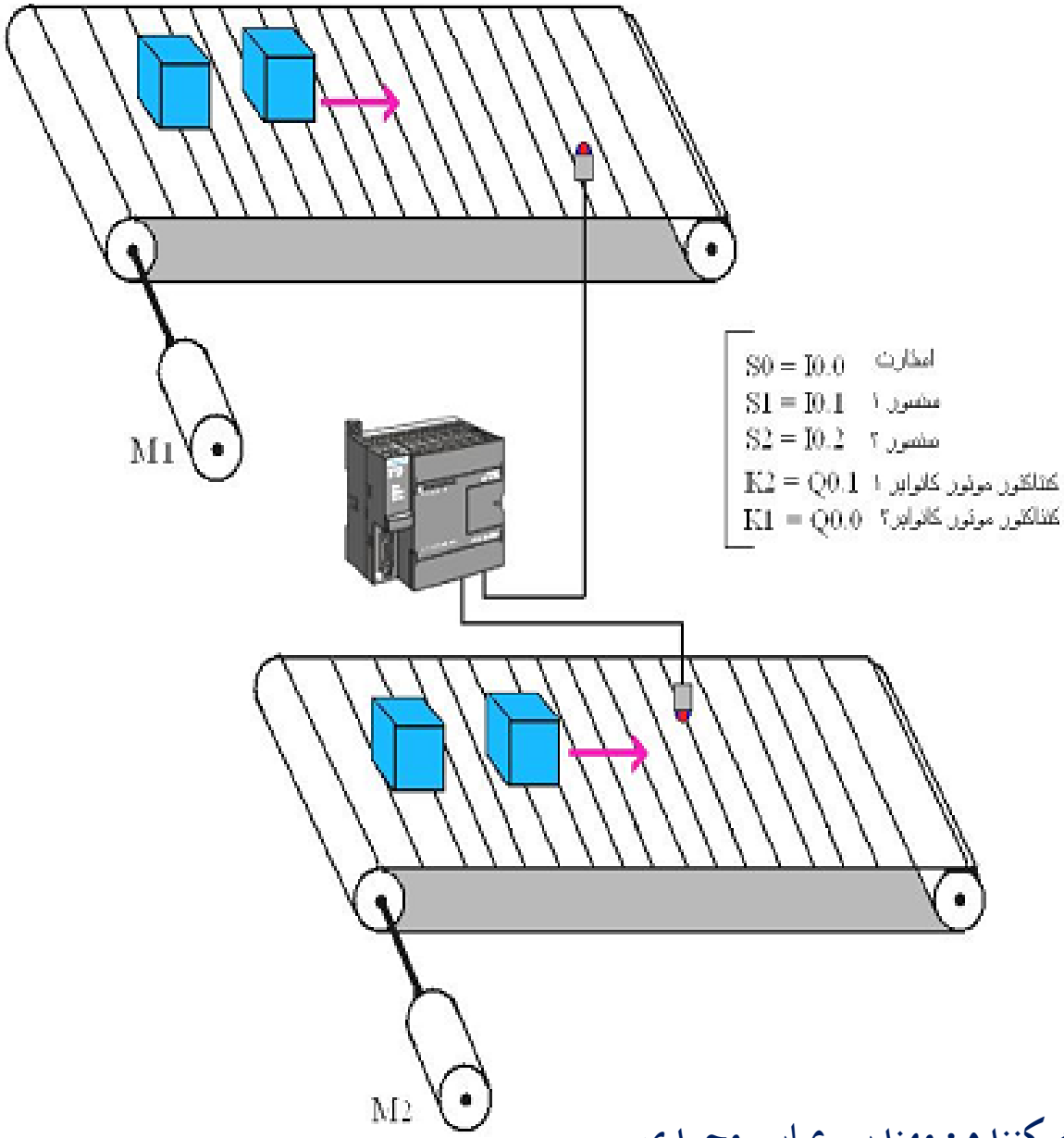
قطعه کاری بر روی یک کانوایر قرار دارد که می تواند در محدوده دو سنسور نوری L1 و L2 حرکت کند . وقتی که شاسی استارت فشار داده می شود ، در صورتی که سنسور L2 توسط قطعه کار تحریک شده باشد کانوایر توسط موتور به سمت جلو حرکت می کند . با حرکت کانوایر و برخورد آن به سنسور L1 کانوایر توسط موتور تغییر جهت می دهد .

S1 = I0.0
L1 = I0.1
L2 = I0.2
K1 = Q0.0 راست گرد
K2 = Q0.1 چپ گرد



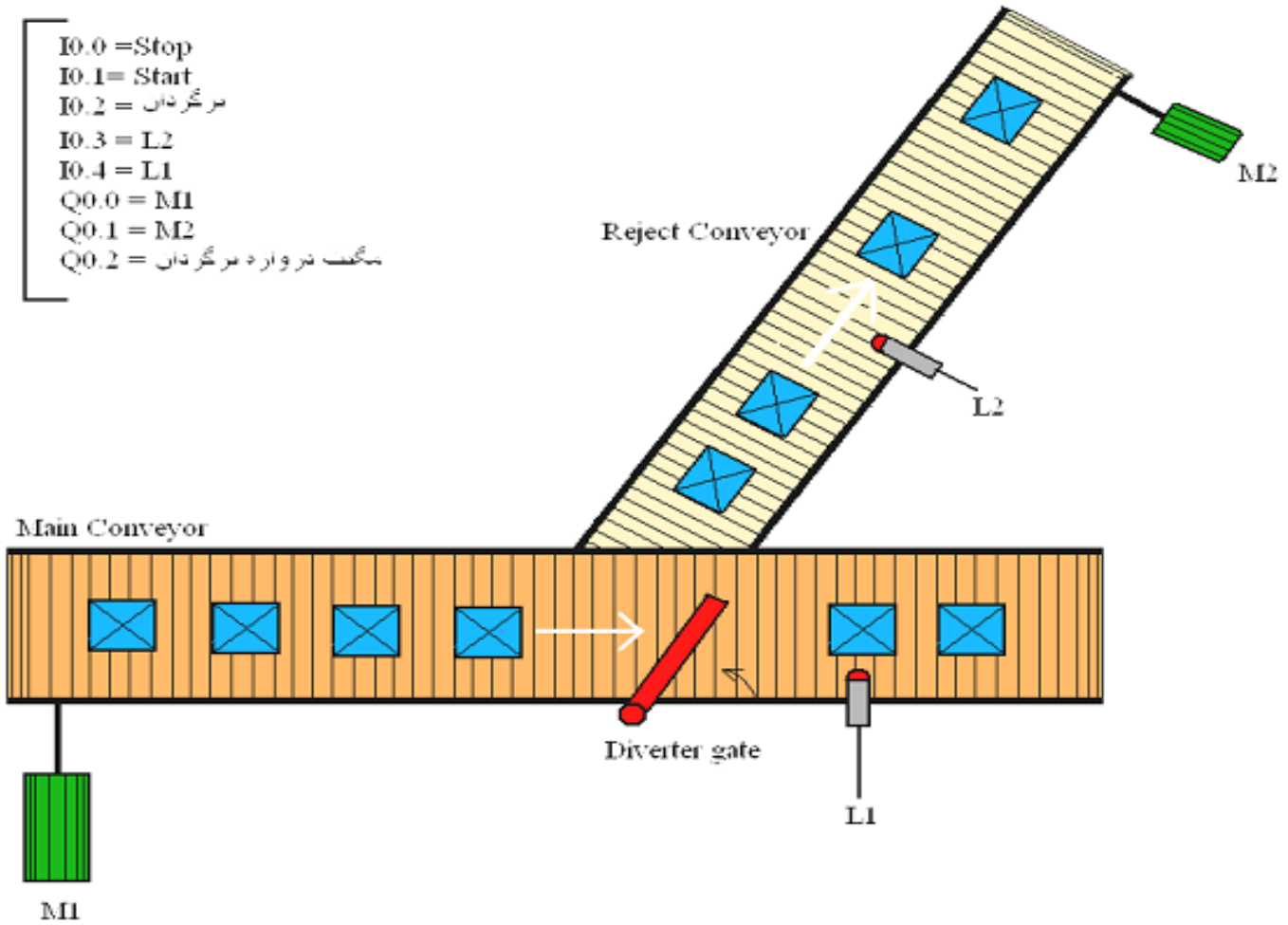
تمرین 8 :

در شکل زیر دو کانوایر که در ساخت یک فرآیند نقش دارند نشان داده شده است . هر کانوایر دارای یک سنسور می باشد که قطعه کارهای عبوری از کنار آن را شمارش می کند . برنامه هایی بنویسید که قطعه کارهای عبوری از هر یک از دو کانوایر را بطور اختصاصی شمارش نماید . زمانی که قطعات کانوایر 1 از 100 عدد گذشت کانوایر 1 خاموش شود و زمانی که سنسور کانوایر 2 از 200 عدد گذشت کانوایر 2 نیز خاموش شود . با زدن مجدد شاسی استارت مراحل از اول آغاز گردد .



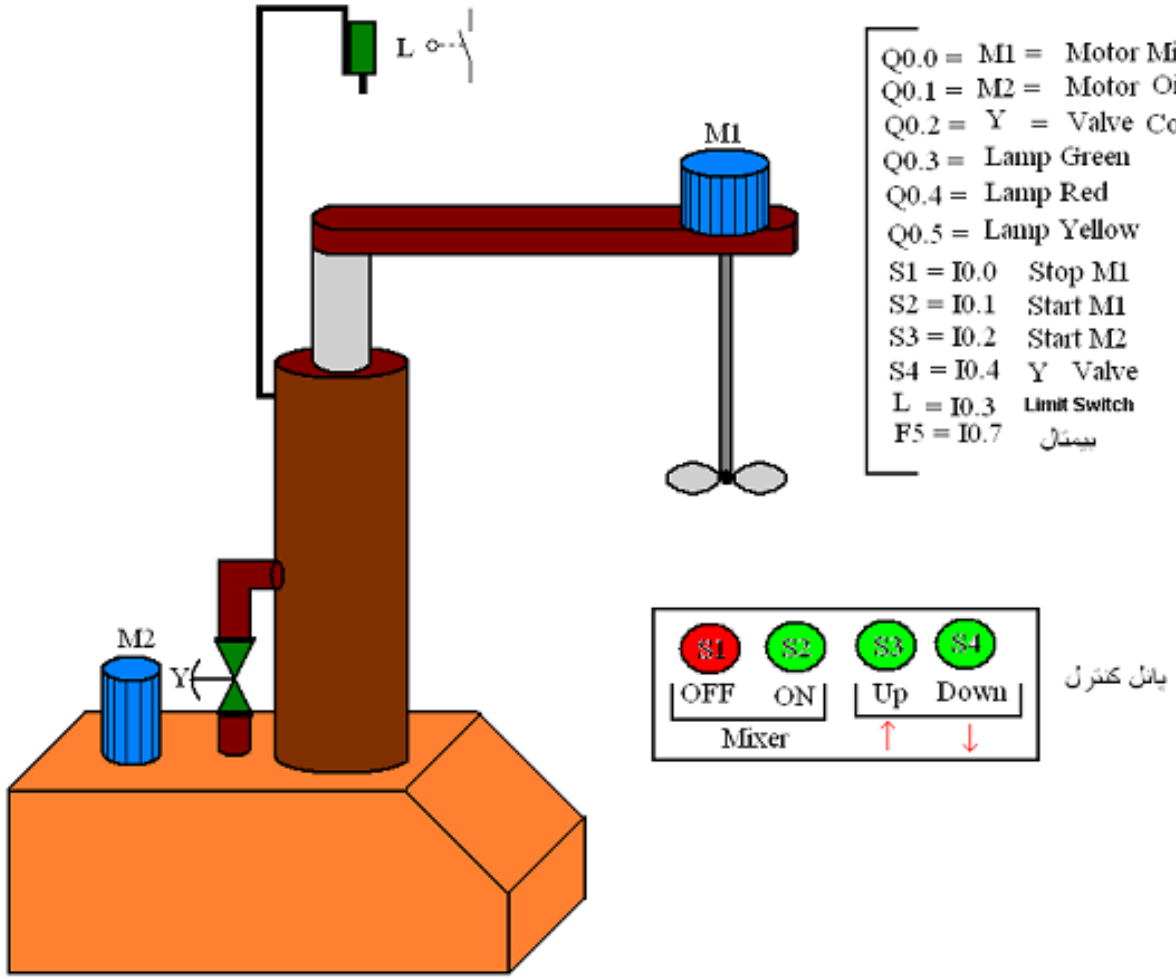
تمرین 9 :

در شکل زیر یک کانوایر اصلی (Main Conveyor) با یک دروازه برگردان (Diverter gate) برای قسمتهای معیوب به داخل کانوایر برگشتی (Reject Conveyor) را نشان می دهد . اگر در بازرسی یک قطعه کار معیوب باشد ، دروازه برگردان توسط اپراتور فعال شده و قطعات به داخل کانوایر برگشتی انتقال پیدا می کند . یک سنسور در کنار کانوایر اصلی و یک سنسور در کنار کانوایر برگشتی وجود دارد و قطعات عبوری را شمارش می کند . برنامه بنویسید که اگر قطعات عبوری از کانوایر برگشتی از 10 عدد و قطعات عبوری از کانوایر اصلی از 50 عدد گذشت سیستم کانوایر متوقف شود (سیستم عملکرد برگردان به این صورت است که با برق دار شدن برگردان عمل کرده و با قطع برق به حالت عادی باز می گردد)



تمرین 10:

شکل زیر میکسر یک کارخانه صنایع رنگ سازی می باشد سیستم عملکرد میکسر به این صورت است که با زدن شاسی S2 موتور اصلی میکسر (M1) شروع به کار می کند . جهت بالا و پایین کردن پروانه میکسر از یک پمپ هیدرولیک استفاده شده است با روشن شدن موتور M2 پمپ هیدرولیک عمل کرده و پروانه میکسر تا محدوده میکروسویچ L بالا رفته و با تحریک شدن میکروسویچ L متوقف می شود . جهت پایین آوردن میکسر شیر برقی Y تعبیه شده که روغن را وارد مخزن اصلی می کند . سیستم کنترل آن را برنامه نویسی نمایید (در صورتی که موتور میکسر دچاراضافه بار گردید سیستم خاموش شده و چراغ نارنجی رنگ چشمک بزند)



تمرین 11:

یک شمارنده ماشین برای شمارش تعداد ماشین وارد شده و خارج شده از یک پارکینگ با ظرفیت 25 اتوموبیل مورد نیاز است :

الف : یک ورودی تعداد اتوموبیل های وارد شده و ورودی دیگر تعداد اتوموبیل های خارج شده را می شمارد .

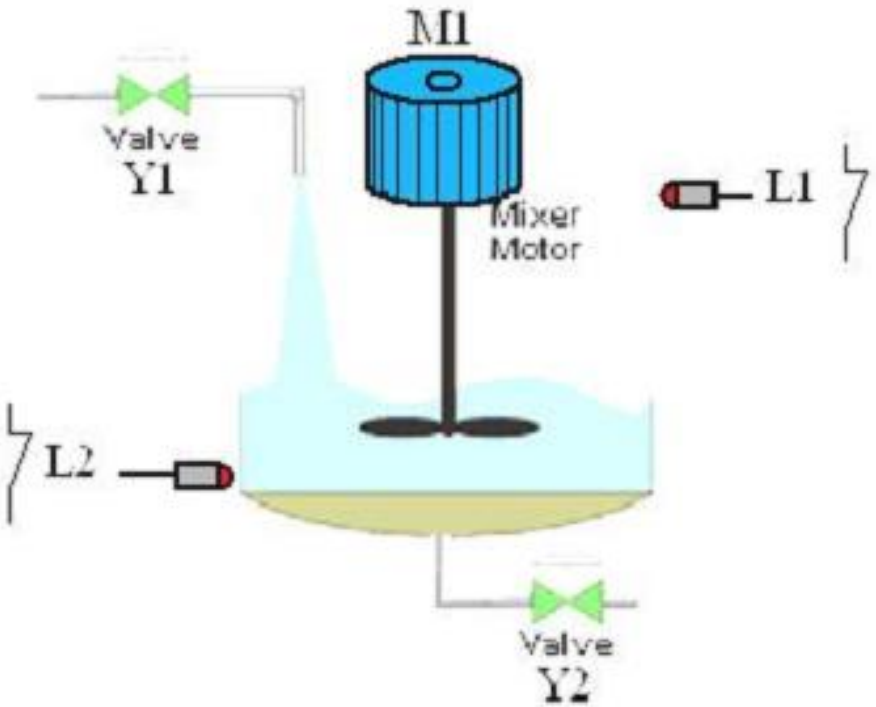
ب : وقتی تعداد اتوموبیل های داخل پارکینگ به 25 عدد رسید ، خروجی کانتر با نمایش عبارت ``Full`` پربودن پارکینگ را مشخص نماید .

ج : وقتی تعداد اتوموبیل های داخل پارکینگ کمتر از 25 عدد بود ، عبارت ``Vacancy`` بمعنی ظرفیت داشتن پارکینگ روشن شود .

د : یک کلید ورودی می تواند توسط متصدی پارکینگ ، پارکینگ را در موقعیت بسته نگهدارد .

تمرین 12:

شکل زیر یک میکسر را نشان می دهد وقتی که شاسی استارت فشار داده شود سلنوئید Y1 فعال شده و مایع می تواند وارد مخزن شود . سنسورهای L1 و L2 سطح بالا و پایین مایع مخزن را مشخص می کنند و هر دو دارای کنتاکت NC می باشند (وقتی که مخزن خالی است L1 و L2 بسته هستند) زمانی که مخزن پر شد سنسور L1 سلنوئید Y1 را قطع و فرمان شروع به کار موتور میکسر را صادر می کند . موتور میکسر برای 30 ثانیه فعال بوده و سپس خاموش می گردد . وقتی که موتور خاموش شد سلنوئید Y2 فعال شده و مایع مخزن را تخلیه می کند پس از خالی شدن مخزن سنسور L2 به حالت عادی برگشته (بسته شده) و سلنوئید Y2 قطع می گردد . برنامه کنترل آن را بنویسید .



Y1 = Q0.0
Y2 = Q0.1
L1 = I0.1
L2 = I0.2
S1 = I0.0
M1 = Q0.2

تمرین 13 :

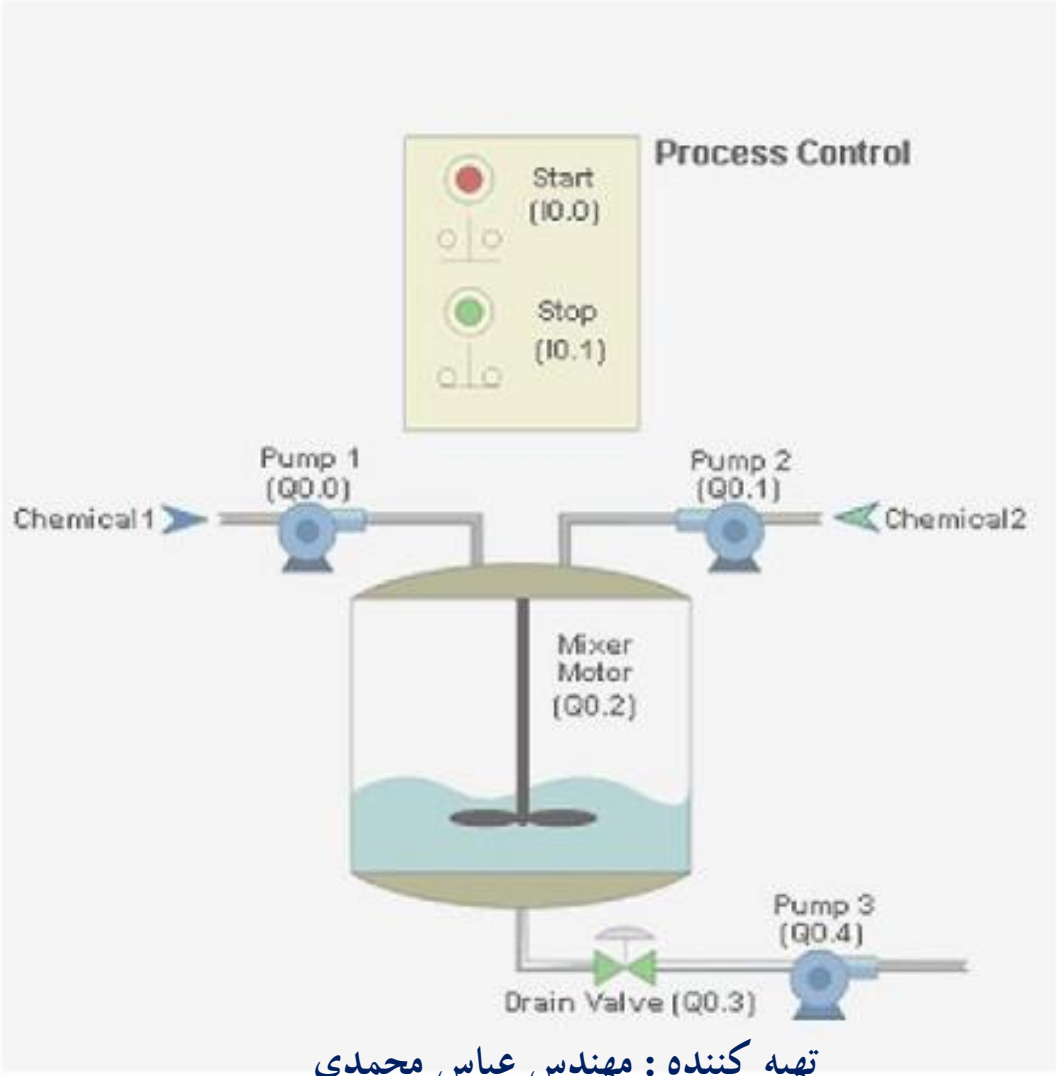
برنامه ای بنویسید مربوط به روشن و خاموش کردن چراغ راه پله های یک ساختمان بطوری که با وارد شدن به طبقه اول ساختمان با فشار دادن کلید چراغ طبقه اول روشن شود و در ابتدای ورودی طبقه دوم با فشار دادن کلید چراغ طبقه دوم روشن و چراغ طبقه اول خاموش شود و این روند تا طبقه آخر ادامه داشته و عکس آن از بالا به پایین نیز صادق باشد . همچنین در هر یک از طبقات در صورت انصراف بتوان برگشت . (سه طبقه)

تمرین 14 :

تایمری طراحی کنید که هم تاخیر در وصل باشد و هم تاخیر در قطع .

تمرین 15 :

شکل زیر یک مخزن به همراه میکسر مواد شیمیایی می باشد . سیستم عملکرد آن به این صورت است که با فشار دادن شاسی استارت ، پمپ یک شروع به کار کرده و یک مایع شیمیایی را وارد مخزن می کند پس از 20 ثانیه از شروع کار پمپ یک ، پمپ 2 نیز روشن شده و یک مایع شیمیایی دیگر را وارد مخزن می کند . پس از 10 ثانیه هر دو پمپ خاموش شده و موتور میکسر به مدت 15 ثانیه مواد شیمیایی را میکس می نماید سپس شیر خروجی باز شده و پمپ 3 به مدت 30 ثانیه روشن شده و مواد مخزن را تخلیه می کند برنامه آن را بنویسید .



تهیه کننده : مهندس عباس محمدی

ادامه دارد...

راه های ارتباط با بنده:

MohammadiSite.ir
oxinauto@gmail.com
۰۹۱۶۳۰۷۷۸۶۵