

بسمه تعالی

دستورکار آموزشی آشنایی با PLC

آزمایشگاه کنترل دانشکده مهندسی مکانیک دانشگاه صنعتی شریف



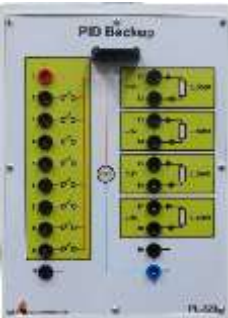
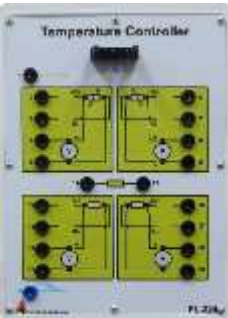
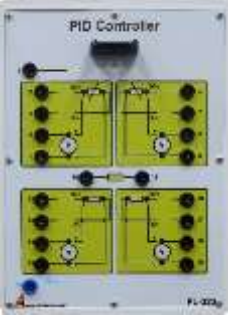
**کنترل کننده منطقی برنامه پذیر – S7 Siemens**

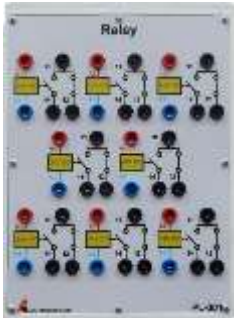
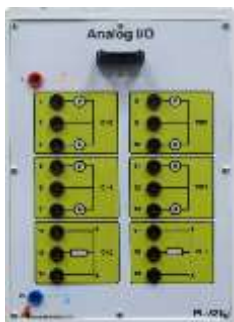
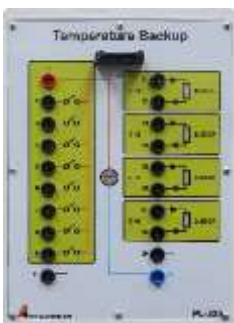
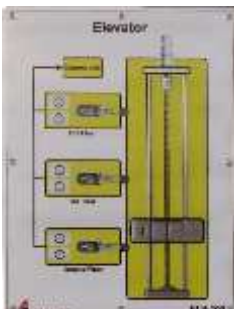
**دستور کار ویژه دانشجو**

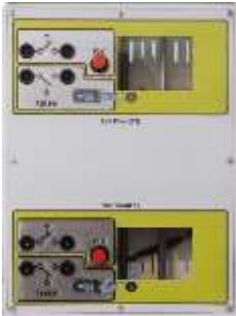
# ۱ معرفی ماژول های دستگاه

هدف : آشنایی با ماژول های دستگاه

| ردیف | نام ماژول              | توضیحات                                                                            | تصویر ماژول                                                                         |
|------|------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| ۱    | شبیه ساز چراغ راهنمایی | در یک تقاطع ۸ چراغ راهنمایی جهت خودروها و عابرین پیاده فراهم شده اند.              |    |
| ۲    | ورودی و خروجی دیجیتال  | از طریق این ماژول ورودی و خروجی های دیجیتال PLC فراهم شده اند.                     |   |
| ۳    | منبع تغذیه             | منبع تغذیه ۲۴ ولت پنج آمپر جهت تغذیه اجزای مختلف دستگاه مورد استفاده قرار می گیرد. |  |
| ۴    | HMI                    | رابط گرافیکی جهت مانیتورینگ و تنظیم به صورت گرافیکی با پورت های ارتباطی متفاوت     |  |

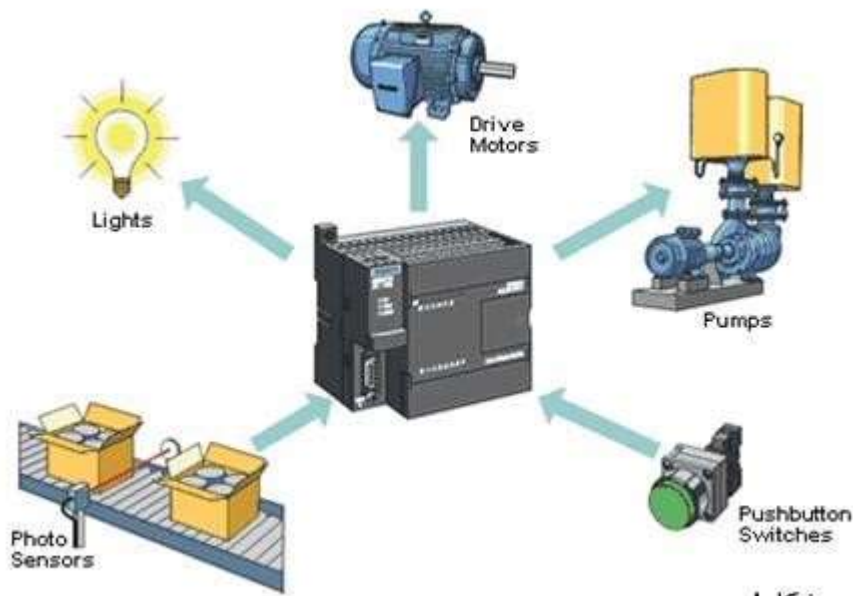
|                                                                                                                          |                                                                                                                                       |                                               |          |
|--------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|----------|
|  <p>PID Backup</p>                      | <p>پورت های مکمل موجود بر روی<br/>ماژول کنترل کننده PID</p>                                                                           | <p>جزء کمکی ماژول PID</p>                     | <p>۵</p> |
|  <p>Switch &amp; Indicator Panel</p>    | <p>۶ عدد کلید قطع و وصل همراه با سه<br/>شاستی مختلف جهت اعمال ورودی<br/>های دیجیتال مختلف به PLC</p>                                  | <p>پنل کلید و نمایشگر</p>                     | <p>۶</p> |
|  <p>Temperature Controller</p>         | <p>ورودی های مختلف ماژول کنترل<br/>کننده دما بر روی این ماژول فراهم<br/>شده است.</p>                                                  | <p>کنترل کننده دما</p>                        | <p>۷</p> |
|  <p>Voltage Isolator &amp; Sensor</p> | <p>شامل دو عدد ایزولاتور ولتاژ جهت<br/>اعمال ورودی آنالوگ به PLC به<br/>همراه دو عدد سنسور مجاورتی<br/>متفاوت و منبع تغذیه ۲۴ ولت</p> | <p>ایزولاتور ولتاژ و سنسورهای<br/>مجاورتی</p> | <p>۸</p> |
|  <p>PID Controller</p>                | <p>این ماژول جهت فراهم سازی ورودی<br/>های کنترل کننده PID استفاده می<br/>شود.</p>                                                     | <p>کنترل کننده PID</p>                        | <p>۹</p> |

|                                                                                     |                                                                                                         |                      |    |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|----------------------|----|
|    | <p>۸ عدد رله ۲۴ ولت به عنوان تجهیزات کمکی در کنار PLC جهت فراهم آوردن کاربردهای بیشتر مهیا شده اند.</p> | رله                  | ۱۰ |
|    | <p>جهت اعمال و دریافت ورودی و خروجی آنالوگ استفاده می شود.</p>                                          | ورودی و خروجی آنالوگ | ۱۱ |
|   | <p>پایه های مکمل ماژول کنترل کننده دما را فراهم می سازد.</p>                                            | پشتیبان ماژول دما    | ۱۲ |
|  | <p>ماژول مادر بوده و از طریق پورت های رابط به سایر ماژول ها متصل می گردد.</p>                           | بسته PLC سری S7-300  | ۱۳ |
|  | <p>جهت شبیه سازی آسانسور استفاده می گردد.</p>                                                           | آسانسور              | ۱۴ |

|                                                                                   |                                                                   |                               |           |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------|-------------------------------|-----------|
|  | <p>مشخص کننده کلید و میکروسوئیچ<br/>واقع در طبقه دوم می باشد.</p> | <p>ادامه شبیه ساز آسانسور</p> | <p>۱۵</p> |
|  | <p>مشخص کننده کلید و میکروسوئیچ<br/>واقع در طبقه دوم می باشد.</p> | <p>ادامه شبیه ساز آسانسور</p> | <p>۱۶</p> |

## ۲ مبانی عملکرد یک پی ال سی

پی ال سی ها در حقیقت نوعی از کنترلرهای قابل برنامه نویسی هستند که در طبقه بندی می توان آنها را جزء خانواده کامپیوتر های شخصی قرار داد. پی ال سی ها در صنعت برای کاربردهای صنعتی بکار برده شده و در این صورت نقش کنترل کننده های سیستم ها و فرآیند ها را به عهده دارند. پی ال سی ها ورودی ها را خوانده، تصمیم گیری کرده و خروجی ها را نسبت به برنامه نوشته شده موجود در حافظه، کنترل می کنند.



شکل ۱

### یک پی ال سی Basic PLC operation

سخت افزار یک پی ال سی بطور کلی از سه بخش اصلی تشکیل شده است. این بخش ها عبارتند از:

۱- بخش های ورودی (Input modules) که بوسیله این ماژول ها ما می توانیم انواع سیگنالهای دیجیتال و آنالوگ را به دستگاه پی ال سی اعمال کرده و این ماژول ها می توانند در صورت نیاز یک سیگنال آنالوگ را به دیجیتال تبدیل کرده و آن را برای استفاده CPU آماده کنند.

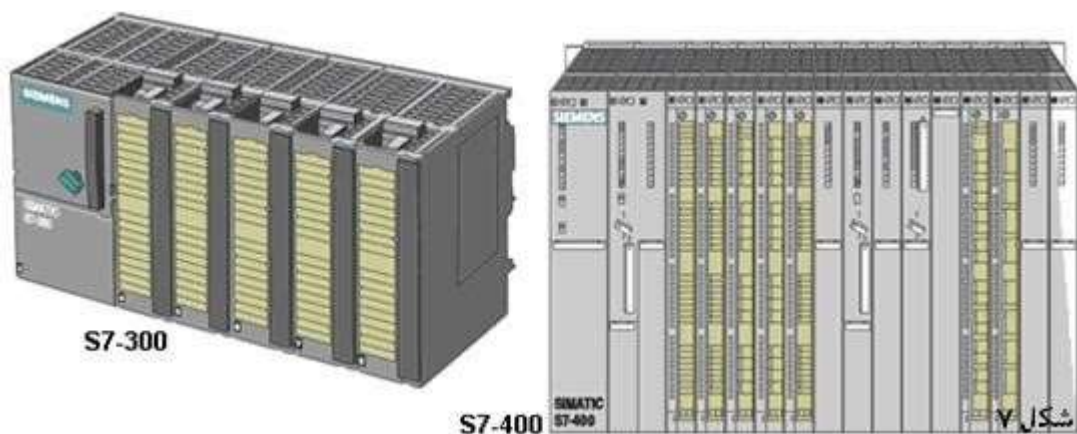
۲- بخش CPU در حقیقت بخش تصمیم گیری پی ال سی بوده و می تواند دستورات موجود در برنامه نوشته شده و بارگذاری شده در حافظه دستگاه را پردازش کند.

۳- بخش های خروجی (Output modules) در صورت نیاز پس از اینکه CPU تصمیم می گیرد به ادوات بیرون از پی ال سی مانند کنتاکتورها، لامپ ها، شیرهای برقی و غیره سیگنالهایی را اعمال کند. این سیگنال ها را به ترمینالهای خروجی اعمال می کند که در حقیقت این ترمینال ها بخشی از سخت افزار و بخش خروجی محسوب می شوند (Output).

modules). لازم به یادآوری است که در صورت نیاز بخش های خروجی می توانند عمل تبدیل یک ولتاژ دیجیتال به آنالوگ و یا برعکس را نیز انجام بدهند.

## ۲.۱ مدل های پی ال سی زیمنس Siemens SIMATIC S7-400/300

مدل های S7-300 و 400 زیمنس برای کاربردهای پیچیده تر از S7-200 بکار برده می شود. تعداد ورودی و خروجی ها این مدل ها خیلی بیشتر از مدل S7-200 می باشد. هر دو مدل S7-400/300 بصورت ماژولار ساخته شده اند و لذا در صورت تمایل مصرف کننده می تواند به نسبت نیاز خود تعداد ماژول های ورودی و خروجی را افزایش بدهد. همچنین این مدل ها فاقد منبع تغذیه بوده و لذا مصرف کننده باید از منبع تغذیه خارجی برای سیستم پی ال سی موجود استفاده کند. انتخاب اینکه مصرف کننده چه مدلی را باید انتخاب کند نیز منوط به سطح پیچیدگی سیستم کنترل کننده دارد. شکل شماره 7 پی ال سی های مدل های 300 و 400 را نشان می دهد.



## ۳ انواع ورودی و خروجی ها

### ۳.۱ Number systems- Decimal and Binary سیستم اعداد – اعشاری و باینری

از آنجایی که پی ال سی ها نوعی از کامپیوترهای صنعتی هستند. لذا این دستگاهها نیز اطلاعات را بصورت اعداد باینری صفر و ۱ در حافظه خود ذخیره می کنند. گاهی این اعداد صفر و یک برای نشان دادن وضعیت روشن و یا خاموش بودن یک خروجی بکار برده می شوند و همچنین بعنوان یک عدد تبدیل شده از اعداد اعشاری (مبنای ۱۰) به باینری بکار برده می شوند. سیستم های اعداد مختلف قابل تبدیل به یکدیگر هستند ولی تمام این سیستم ها سه مشخصه مشترک دارند. این سه مشخصه عبارتند از 1- تعداد اعداد 2- مبنای سیستم 3- وزن هر عدد در آن سیستم. مثلا سه مشخصه سیستم یک عدد در مبنای 10 یا اعشاری بقرار زیر است .

تعداد اعداد : ده عدد : 0, 1, 2, 3, 4, 5, 6, 7, 8 و 9

مبنای سیستم:

وزن هر رقم در این سیستم: 1, 10, 100, 1000

مشخصه سیستم باینری که در پی ال سی ها مصرف می شود نیز بقرار زیر است:

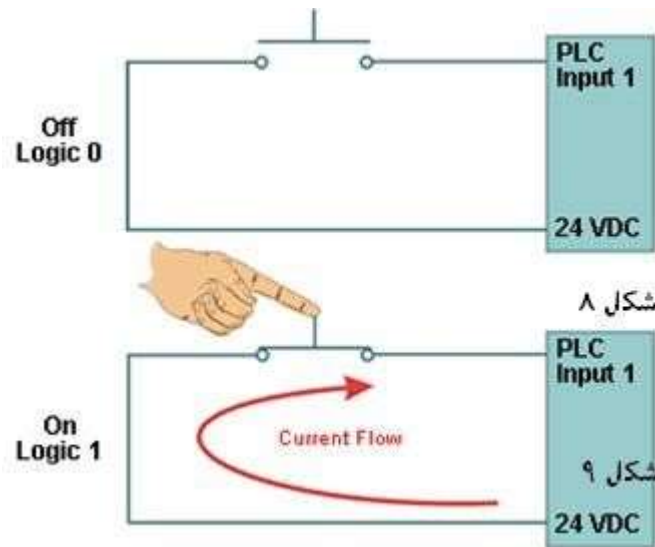
تعداد اعداد: دو عدد 0 و 1

مبنای سیستم: 2

وزن هر رقم در این سیستم: 1, 2, 4, 8, 16, 32, ..... 2<sup>n</sup>

### ۳.۲ اعداد منطقی 0 و 1 Logic 0, Logic 1

در صورتی که سیگنال های دیجیتال و آنالوگ هر دو قابل اعمال به پورت ها پی ال سی ها هستند ولی بخش CPU فقط می تواند سیگنال و اعداد دیجیتال را پردازش کند. و این سیگنال ها فقط می توانند از اعداد صفر و یک تشکیل شده باشند. صفر و یا OFF یا خاموش، یک و یا ON یا روشن. در این صورت سیستم اعداد باینری توسط پی ال سی ها برای نشان دادن وضعیت سیگنال های دیجیتال بکار برده می شود. عدد باینری 1 نشان دهنده وجود یک سیگنال و یا روشن بودن یک کلید و عدد باینری صفر نشان دهنده عدم وجود سیگنال و یا خاموش بودن همان کلید است. شکل های 8 و 9 را ملاحظه کنید.

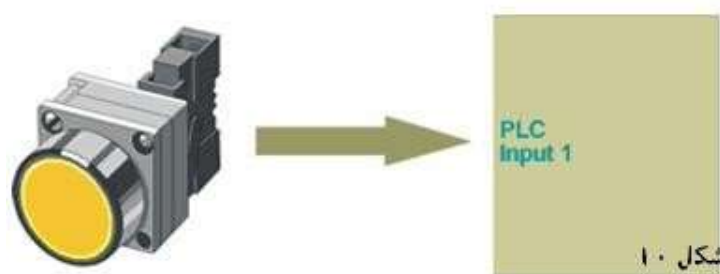


شکل شماره ۸ کلید خاموش است و شماره ۹ کلید روشن است.

همانطوریکه در شکل شماره ۸ ملاحظه می‌کنید وقتی که کلید خاموش است وضعیت مدار در حالت خاموش و یا OFF می‌باشد. وقتی که کلید فشرده می‌شود وضعیت مدار در حالت روشن و یا ON قرار داشته و در این صورت جریان در مدار برقرار می‌باشد.

### ۳.۳ سنسورها (Sensors)

در صحبت کردن بزبان پی ال سی ها ما از اصطلاحات خاصی استفاده می‌کنیم. دانستن معنی هرکدام از این اصطلاحات ما را در درک زبان بکارگرفته شده در ارتباط با کاربرد پی ال سی ها کمک می‌کند. مثلاً یک سنسور عبارتند از یک دستگاه که وظیفه‌اش تبدیل کردن یک مشخصه فیزیکی به یک سیگنال برقی می‌باشد که توسط پی ال سی مورد مصرف قرار بگیرد. سنسورها به ترمینال‌های ورودی یک پی ال سی وصل می‌شوند. مثلاً یک سنسور مجاورتی در صورتی که یک جسم جلوی آن بیاید، خروجی‌اش یک می‌شود و در صورتی که از جلوی آن دور شود، صفر می‌شود.



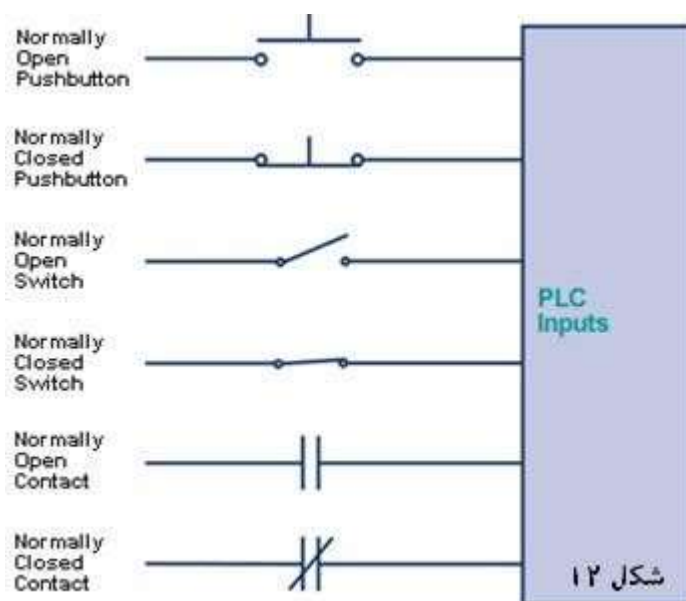
### ۳.۴ محرک و یا فعال کننده ها (Actuators)

محرک و یا فعال کننده ها نوعی تبدیل کننده هستند که سیگنال های تولید شده را به یک عمل فیزیکی تبدیل می کنند. یک کنتاکتور در حقیقت یک نوعی از محرک و یا فعال کننده می باشد که در صورت وصل شدن به برق، یک عمل مکانیکی تولید می کند که در نهایت باعث می شود کنتاکت های یک کلید، از حالت باز در حالت بسته قرار بگیرند. بسته به اینکه وضعیت خروجی یک ترمینال پی ال سی در حالت روشن و یا خاموش باشد، مطابق شکل ۱۱ موتور الکتریکی بترتیب در حالت روشن و یا خاموش قرار می گیرد. در صورت وصل شدن برق به بوبین کنتاکتور، هسته داخلی کنتاکتور جذب شده و لذا برق ورودی اولیه کنتاکتور به ترمینال های خروجی وصل شده و موتور الکتریکی شروع به کار می کند. یک کنتاکتور صنعتی معمولاً از سه تیغه اصلی تشکیل شده که در صورت جذب شدن، برق سه فاز را به موتور الکتریکی وصل می کند. شکل شماره ۱۱ را ملاحظه کنید.



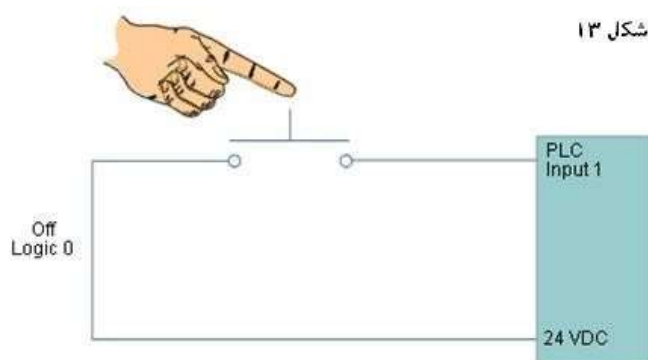
### ۳.۵ ورودی های دیجیتال (Discrete Inputs)

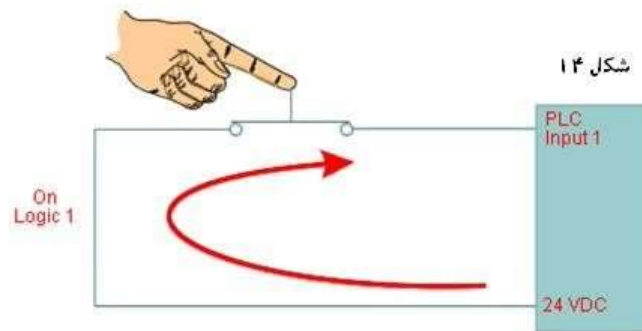
ورودی های دیجیتال منظور سیگنال های ورودی می باشد که ولتاژ آنها بین صفر (OFF or 0) و یا یک (OFF or 1) در تغییر است. مثلاً ورودی یک پوش باتن که قبلاً در مورد آن صحبت شد یک نوعی از سیگنال ورودی دیجیتال را تولید میکند. مثال های دیگر ورودی های دیجیتال میتوان از انواع کلید های کشویی (Toggle switches)، میکروسویچ (Limit switches)، ها سوییچ های حساس به فلز و آهن (Proximity switches)، چشم های الکترونیکی نام برد که در صورت فعال شدن معمولاً ولتاژ یک (ON or High) و در صورت خاموش بودن ولتاژ صفر (OFF or Low or 0) به ورودی پی ال سی اعمال میکنند. شکل ۱۲ را ملاحظه کنید.



کلید پوش باتن با تیغه باز) در حالت غیر فعال ( نمونه ای از ورودی دیجیتال محسوب میشود که می توان برای مثال از کلیدهای فشاری بزرگ " سبز" نام برد که معمولا برای استارت زدن یک الکتروموتور یک و یا سه فاز بکار گرفته می شود. در اصلاح کتب فنی بزبان انگلیسی این نوع از کلیدهای فشاری به اسم (Normally Open Pushbutton) اطلاق میشود.

برای درک عملکرد بهتر کلید فشاری Normally Open، شکل 13 را ملاحظه کنید. در این شکل یک سیم کلید به ورودی شماره 1 ترمینال دیجیتال پی ال سی وصل شده و سر دیگر به منبع تغذیه 24 ولت. در صورت فشار ندادن کلید (غیرفعال بودن کلید) هیچ ولتاژی به ورودی شماره 1 اعمال نمی شود. در این صورت وضعیت ولتاژ سیگنال ورودی در حالت خاموش و یا ( OFF = Low = 0 ) صفر می باشد. ولی در صورت فشار کلید، ولتاژ 24 ولت به ورودی ترمینال شماره 1 اعمال خواهد شد. در این صورت وضعیت ولتاژ ورودی ترمینال شماره 1 در حالت یک و یا روشن می باشد (ON=High=1) شکل 14 را ملاحظه کنید.





شکل ۱۴

### ۳.۶ سیگنال های ورودی نوع آنالوگ Analog Inputs

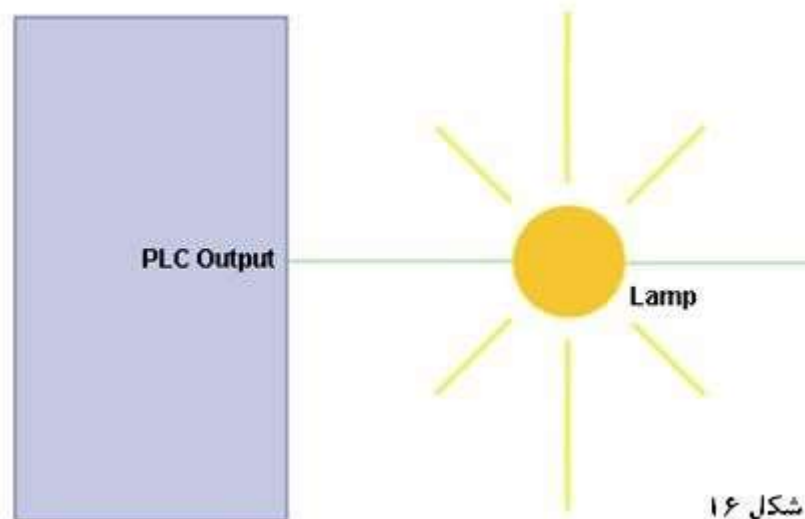
یک ترمینال ورودی آنالوگ پی ال سی طوری طراحی و ساخته شده که میتواند ولتاژهای آنالوگ را که سطح آن دایما در حال تغییر است را قبول کرده و آنرا برای پردازش و اعمال هرتصمیمی که از طرف نرم افزار به آن دیکته میشود بکار ببرد. معمولا سطح جریان ورودیهای آنالوگ میتواند از صفر تا 20 میلی آمپر و یا سطح ولتاژ از صفر تا 10 ولت VDC در تغییر باشد. سطح ولتاژ و جریان عنوان شده تقریبا استاندارد است و سازنده گان این ادوات، آنها را برای تولید ولتاژ و یا جریان های فوق طراحی میکنند. شکل شماره 15 را ملاحظه کنید.

در شکل فوق ما در نظر داریم که سطح مایع داخل مخزن را به پی ال سی گزارش کنیم تا دستگاه بتواند بر مبنای سطح داخل مخزن تصمیم هایی را اتخاذ کند. می دانیم که در صورت خروج آب از مخزن و یا ورد آب به داخل مخزن سطح آب بصورت خطی بالا خواهد آمد. پس میزان ولتاژی که به ورودی آنالوگ پی ال سی اعمال خواهد شد بین صفر و یا یک در تغییر نخواهد بود بلکه ولتاژی و یا جریانی خواهد بود که نسبت به سطح آب دایما در تغییر خواهد بود. مثلا چیزی در حدود 0 ولت وقتی که هیچ آبی وجود ندارد و 5 ولت وقتی که میزان آب تا حدودا یک دوم ظرف را پر کرده و 10 ولت وقتی که مخزن تا پر است. پس در این صورت یک سطح سنجی انتخاب میکنیم که بتواند ولتاژهای مختلفی را نسبت به حجم آب گزارش کند. در شکل 15 وظیفه سطح سنج الکترونیکی (level Transmitter) همین است.

### ۳.۷ (Discrete Outputs) خروجی های دیجیتال

سطح ولتاژ سیگنال خروجی دیجیتال فقط یکی از دو حالت گفته شده در مورد اعداد باینری را می تواند اختیار کند که یا روشن است یا خاموش به زبان دیگر عدد باینری یک یا صفر (0 or 1). بوبین های شیرهای برقی یا کنتاکتورها و انواع لامپهای سیگنال و یا محرک کننده ها مختلف را میتوان با وصل به ترمینال های خروجی یک پی ال سی حرکت کرد. در این صورت اگر لامپی یا کنتاکتوری به ترمینال خروجی وصل است میتوان آنرا روشن و یا خاموش کرد و در صورت وصل

سیم های یک الکتروموتور به یک کنتاکتور میتوان با فعال کردن یا نکردن بوبین کنتاکتور موتور الکتریکی را فعال و یا خاموش کرد. شکل 16 وصل شدن یک لامب را به خروجی دیجیتال پی ال سی را نشان میدهد.

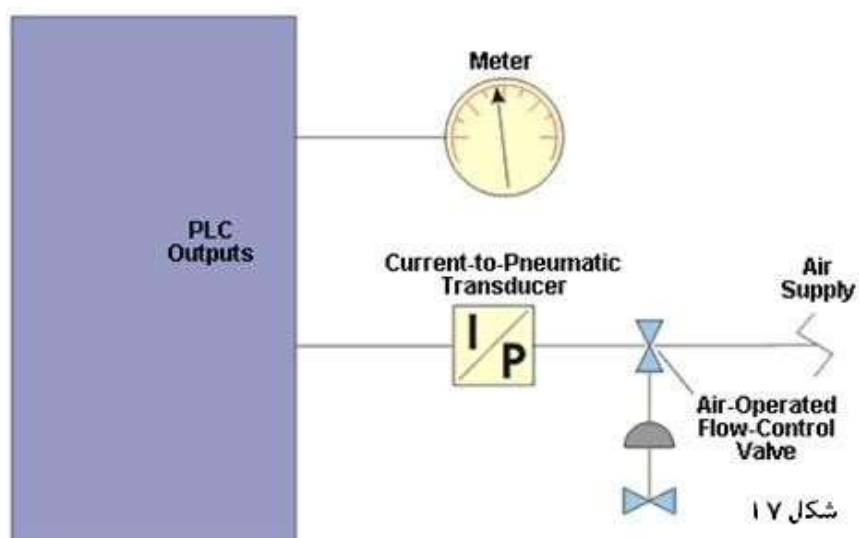


شکل ۱۶

### ۳.۸ خروجی های نوع آنالوگ (Analog outputs)

سطح ولتاژ یا جریان یک خروجی آنالوگ دائما در حال تغییر می باشد. ولتاژ این خروجی ممکن است بین صفر تا 10 ولت در نوسان باشد که در این صورت با وصل یک ولت متر به خروجی فوق میتوان تغییرات ولتاژ خروجی را بوضوح رویت کرد. نشان دهنده هایی مانند سرعت سنج اتوموبیل و یا خروجی های لودسل یک ترازوی الکترونیکی و یا ترانس دیوسرهای حرارت همگی آنالوگ هستند و لذا در صورت تمایل به وصل خروجی پی ال سی به این تیپ ادوات باید نوع سیگنال دریافتی نیز باید آنالوگ باشد. ادوات دیگری مانند شیرهای برقی نیز نیاز به سیگنال آنالوگ برای عملکرد خود دارند. طبق شکل 17 با استفاده از یک ترانسدیو سر تبدیل جریان پنوماتیک میتوانیم جریان را طوری تبدیل بکنیم که شیر برقی بتواند با آن کار کند.

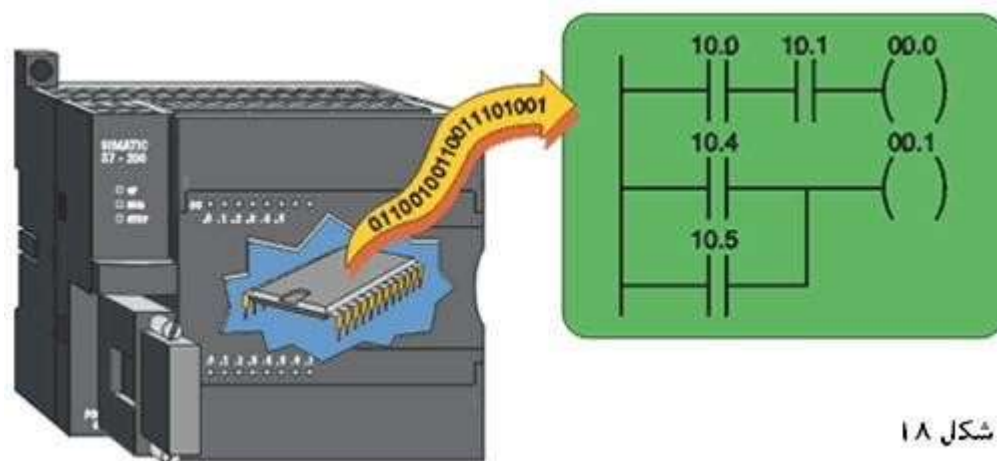
دیافراگم شیربرقی نیاز به سیگنال نوع آنالوگ دارد که بتواند خروجی شیر را به میزان لازمه باز و یا ببند و معمولا میزان سطح خروجی این شیر ها برای هدایت معایعات و یا گازها باید دائما در حال تغییر باشد.



## ۴ مقدمات برنامه نویسی

### ۴.۱ واحد پردازش مرکزی (CPU)

واحد پردازش مرکزی یک پی ال سی از یک میکروپراسسور تشکیل شده که وظیفه اش پردازش نرم افزار و تصمیم گیری بر مبنای دستوراتی را که بصورت کد در نرم افزار گنجانده شده است. این واحد دارای حافظه جانبی به میزان لازمه بوده و با اسکن کردن ورودی ها و تصمیم گیری بر مبنای نرم افزار کنترلی، می تواند کنترل یک فرایند را به عهده بگیرد. بطور کلی CPU عملیاتی مانند قطع و وصل کردن خروجی های دیجیتال، شمارش، اندازه گیری زمان، مقایسه داده ها و انجام عملیات پی در پی را به عهده دارد. شکل 18 را ملاحظه کنید.



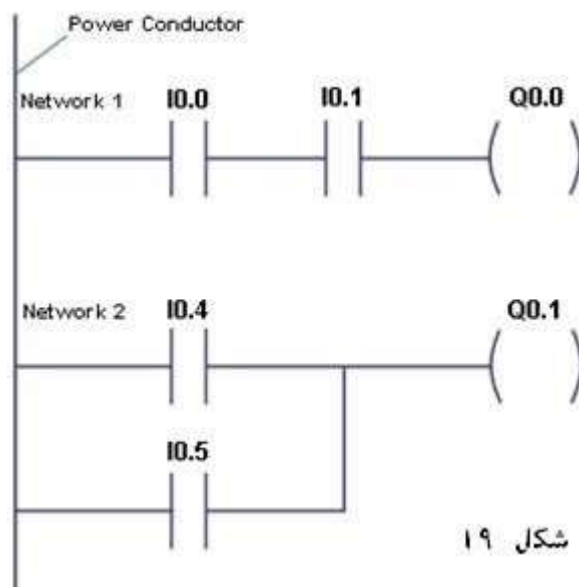
شکل ۱۸

### ۴.۲ برنامه نویسی پی ال سی با روش نرده بانی (Programming / Ladder Logic)

یک برنامه Ladder Logic تشکیل شده از یک یا چند دستور پی ال سی که بکمک اجرای این دستورات "کاری" صورت گیرد. در این صورت تهیه کردن یک برنامه پی ال سی یعنی تهیه کردن لیستی از دستورات. چندین روش تهیه یک برنامه پی ال سی وجود دارد که با استفاده از هر کدام از این تکنیک ها می توان یک برنامه نوشت. متداول ترین این تکنیک ها عبارتند از:

- 1- تکنیک نرده بانی (Ladder Logic)
- 2- تهیه لیستی از کدهای اسمبلی (Statement lists)
- 3- تهیه برنامه با استفاده از سمبل های گیت های منطقی (Function block).

در استفاده از تکنیک نرده بانی ادوات الکتریکی در یک خط پشت سرهم قرار می گیرند و شبه سیم کشی هایی هستند که در آن ادوات بطور سری به هم دیگر توسط سیم وصل شده اند. شکل 19 را ملاحظه کنید.



شکل ۱۹

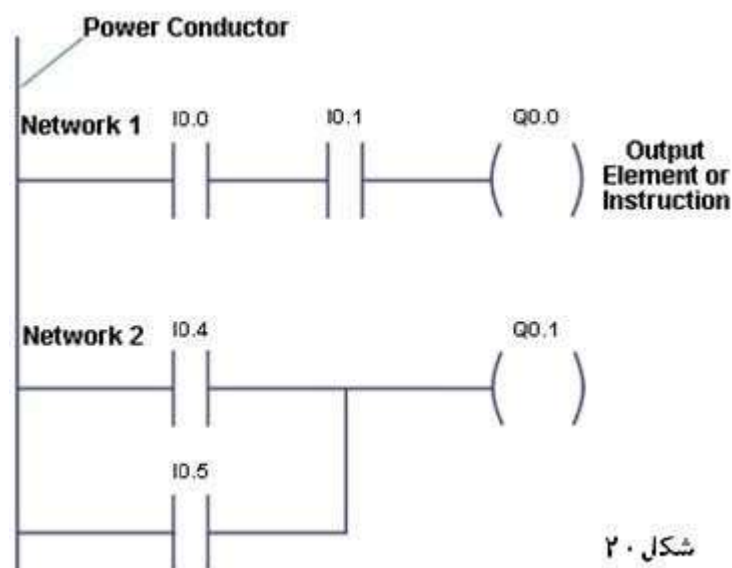
### ۴.۳ (Reading Ladder Logic Diagrams) خواندن یک دیاگرام نرده بانی منطق

خط عمودی سمت چپ نشان دهنده سیم فاز<sup>۱</sup> است. ادوات خروجی نیز از یک طرف به سیم زمین و صل می شوند که معمولاً در دیاگرام های سیم کشی نشان داده نمی شود.

یک برنامه نرده بانی منطقی (Ladder Logic diagram) از طرف چپ بر است و از بالا بطرف پایین خوانده می شود. اگر شکل 19 را یک نرده بان فرض کنیم، در این صورت هر کدام از ادوات موجود در هر ردیف حکم پله های نرده بان را پیدا می کنند که در اصطلاح به آن یک rung (رانگ) گفته می شود. اسم دیگر هر کدام از پله ها در اصطلاح یک شبکه (network) نامیده می شود. جمیع این دستورات پی ال سی که بصورت پله پله یکی پس از دیگری از بالا به پایین قرار دارند را "چندین شبکه" نامگذاری می کنند (networks).

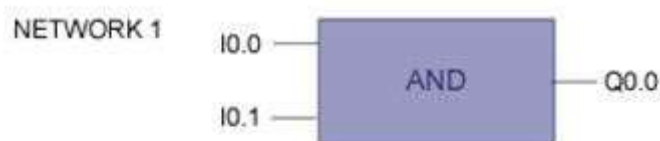
بنابراین طبق شکل 19 هر شبکه ممکن از چندین عنصر کنترل کننده تشکیل شده باشد ولی فقط می تواند یک عنصر خروجی داشته باشد.

<sup>۱</sup> power conductor

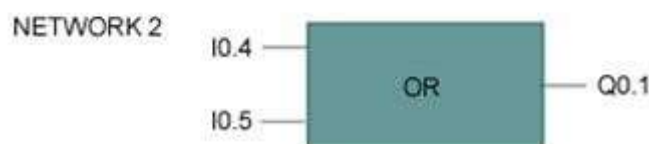


#### ۴.۴ برنامه نویسی به سبک استفاده از سمبل های گیت های منطقی

در اروپا بیشتر از این تکنیک برای نوشتن برنامه های پی ال سی استفاده می شود. در این تکنیک برای "اند" و یا "اور" و یا اصولاً هر فانکشن دیگر سمبل منطقی آن را در هر کدام از شبکه ها قرار داده می شود. مثلاً طبق شکل ۲۲ برای نوشتن برنامه شکل ۲۱ که در آن در شبکه اول دو ورودی با هم "اند" می شدند و در شبکه دوم "او" , در شکل ۲۲ نیز از دو سمبل "اند" و "اور" استفاده کرده. معمولاً می توان نرم افزارهای پی ال سی را که برای نوشتن نرم افزار منطقی بکار گرفته می شود را در اول شروع به کار تنظیم کرد و بعد از آن فقط با استفاده از آن تکنیک "خاص" شروع به تهیه و یا نوشتن "برنامه" کرد.



شکل ۲۲

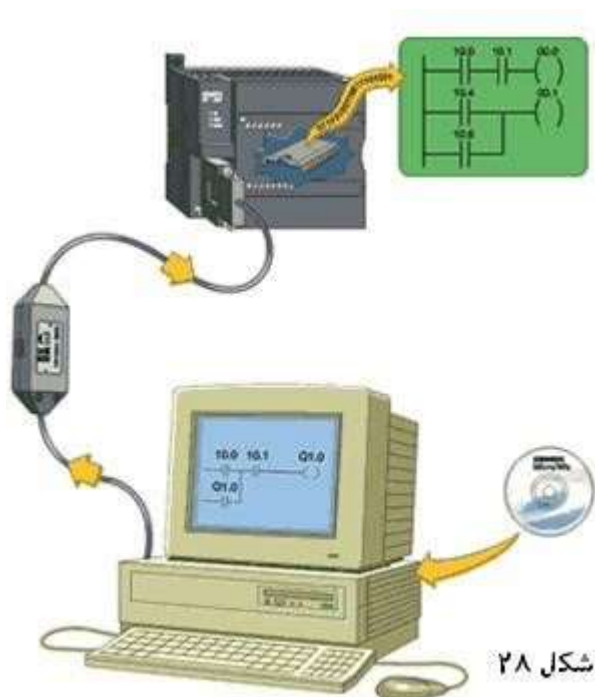


#### ۴.۵ اسکن کردن نرم افزار توسط پی ال سی (PLC scan)

یک پی ال سی اجرا کردن یک نرم افزار را با انجام خواندن های پی در پی (scan) که به آن "اسکن" کردن اطلاق می شود انجام می دهد. مطابق شکل ۲۳ پی ال سی اسکن کردن را با "خواندن" وضعیت (Input status) ورودی ها شروع می کند. به این ترتیب که تمام ورودی ها را اسکن می کند که ببیند هر کدام از ورودی ها مثلاً کلید های پوش باتن فشرده و یا فعال شده و یا خیر. اگر هر کدام از ورودی ها فعال شده باشد، وضعیت آنرا در حافظه مساوی وضعیت "روشن" و یا ON قرار می دهد (Read Inputs). سپس برنامه بارگذاری شده را که قبلاً در حافظه دستگاه قرار داده شده را اجرا می کند (Execute Program). سپس یک سری از روتین های مختلف مربوط به ارسال اطلاعات به دستگاه های جانبی مانند نشان دهنده های وصل شده به پی ال سی و یا برنامه مربوط به تست صحت عملکرد مدارات داخلی را انجام داده و اگر ایرادی در هر بخش مشاهده کند آن را بطریقی به اطلاع آپراتور می رساند (Diagnostics and communications). حال که پی ال سی در این مرحله ورودی ها را اسکن کرده و وضعیت هر کدام را می داند و غیر از آن خود برنامه کنترلی را نیز خوانده و می داند که در مورد تغییر هر کدام از وضعیت ها چه تصمیمی را باید اتخاذ کند، پس در مرحله چهارم خروجی ها را نسبت به خواسته برنامه تنظیم می کند. مثلاً اگر قرار بود طبق برنامه در صورت "اند" شدن دو ورودی خروجی در حالت "روشن" قرار بگیرد، حال که ورودی ها را در مرحله اول اسکن کرده و هر دو در حالت روشن بودند، حال در مرحله چهارم خروجی را در حالت روشن و یا (ON=1) قرار می دهد. پس از اتمام مرحله چهارم، مجدداً اسکن کردن ورودی ها با شروع انجام مرحله اول را پیگیری می کند. مدت زمانی که طول می کشد پی ال سی چهار مرحله فوق را انجام بدهد را در اصطلاح "زمان سایکل" و یا (cycle time) اطلاق می شود. بدیهی است که طول این زمان رابطه مستقیمی دارد با تعداد ورودی ها و یا خروجی ها و در کل طول برنامه در دست اجرا. هر چقدر طول برنامه بیشتر باشد در این صورت پی ال سی مجبور است وقت بیشتری صرف "اسکن" و اجرا برنامه کند و به همان نسبت زمان "اسکن سایکل" نیز طولانی تر خواهد شد.



شکل ۲۳



شکل ۲۸

#### ۴.۶ نرم افزار مورد نیاز برنامه نویسی

با استفاده از یک کامپیوتر و نرم افزار تهیه شده توسط شرکت زیمنس بنام step 7-micro/Win32 می توان برنامه مورد نیاز را با وصل کردن کابل مخصوص آنها درون حافظه پی ال سی بارگذاری کرده و پس از تست برنامه از آنها در فرآیندهای مختلف استفاده کرد.

#### ۵ STEP 7 - Micro/WIN32 نرم افزار

نرم افزار فوق به برنامه نویس این امکان را می دهد که با قراردادن دستورات مخصوص پی ال سی که باید طبق یک فرمت خاص نوشته شوند در محیط برنامه نویسی تهیه کرده و یک برنامه کنترلی را تهیه و اشکال زدایی کند. دستورات برنامه نویسی Instructions

دستورات استاندارد عبارتند از فرامینی که در اکثر برنامه های تهیه شده برای پی ال سی دیده می شوند. این دستورات عبارتند از دستورات مرتبط با تایمرها (timers), شمارندها (counters), دستورات ریاضی و منطقی (math and logical), افزایش یا کاهش دهنده (increment/ decrement / invert) و انتقال تبدیل و دستورات بلاک (Block instructions)

دستورات خاص عبارتند از دستوراتی که میتوان توسط کاربرد آنها یک گروه داده مانند بایت ویا (word) 16 بیت (دستکاری انجام داد (manipulate data). مانند دستورات conversion, find, table, shift

. for/ next and real-time instructions

دستورات سرعت بالا عبارتند از دستوراتی که باعث می شوند پی ال سی خارج از نوبت (cycle time) به آنها رسیدگی کند. در صورت فعال شدن این گونه دستورات، پی ال سی خارج از نوبت دیکته شده توسط سایکل تایم (cycle time) عمل کرده و اجرا کردن اینگونه دستورات را در الویت عملیات قابل انجام خود قرار می دهد. این دستورات عبارتند از شمارنده های سرعت بالا (high-speed counters) و وقفه ها (interrupts) خروجی ها و دستورات مخابره (output, and ) (transmit instruction).

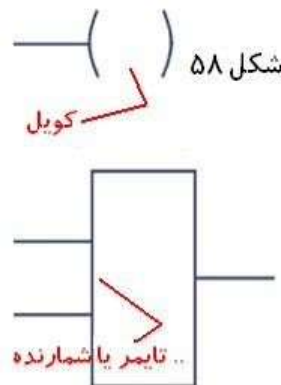
## ۵.۱ سمبل کنتاکت های حالت عادی باز و یا بسته Contact Symbols

زبان نرده بانی از دستوراتی تشکیل شده که با قرار گیری آنها کنار یکدیگر می توان یک برنامه کنترلی را طراحی کرد. دو دستور خیلی مهم این زبان برنامه نویسی عبارتند از کنتاکت در حالت عادی بسته (normally closed (NC) contact) و کنتاکت در حالت عادی باز (normally open (NO) contact). در صورت بسته بودن کنتاکت ها جریان می تواند در مدار بسته عبور کرده و مثلاً لامپی را روشن کند. در این صورت می گوئیم مدار روشن و یا ON است. شکل ۵۷ دو کنتاکت "حالت عادی باز" و "حالت عادی بسته" را نشان میدهد.



## ۵.۲ سمبل های بوبین (کویل) جعبه ای شکل (coils and boxes)

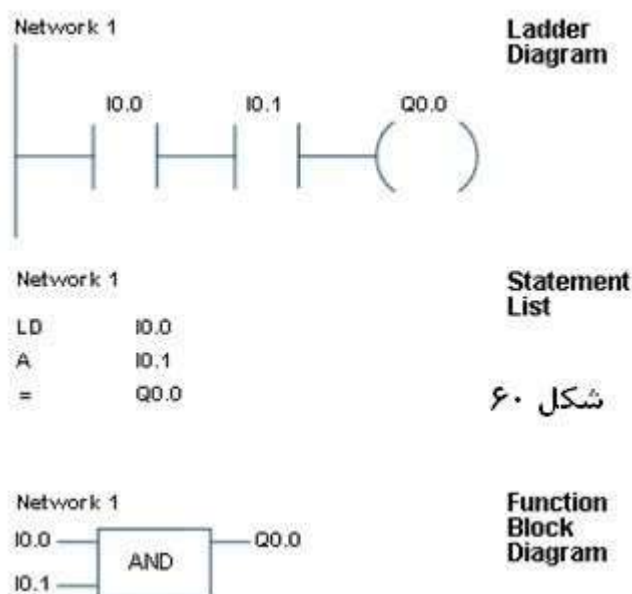
در برنامه های منطقی نرده بانی سمبل کویل (coil) نشان دهند وجود یک "بار" در مدار خروجی می باشد. و بطور کلی هر نوعی از "بار" خروجی در این برنامه ها با استفاده از سمبل کویل نشان داده می شود که در صورت بسته شدن مدار مرتبط با آن جریان در مدار فوق مسیرش را از باطری یا منبع تغذیه به بار خروجی طی کرده و در نهایت سبب روشن شدن و یا تحریک شدن خروجی فوق خواهد شد. در عمل ممکن است یک رله کنتاکتور جایگزین این "کویل" باشد و یا یک لامپ. پی ال سی اهمیت نمی دهد که بار خروجی مربوطه چه نوع "بار الکتریکی" است. از سمبل های "جعبه ای" شکل برای نشان دادن بعضی از ادوات دیگر مانند تایمرها (timers) شمارنده ها (counters) و یا دستورات مرتبط با عملیات ریاضی بکار گرفته می شوند.



### ۵.۳ قرار دادن عناصر برنامه منطقی نرده بانی (Entering Elements)

برای قراردادن هرکدام از عناصر ویا دستورات برنامه نویسی در نرم افزار کنترلی کافی است که مکان نمای کامپیوتر را در روی عنصر فوق در روی لیست این دستورات قرارداده و کلید راست ماوس را بفشاریم و در این صورت می توانیم سمبل آن دستور العمل را "درگ drag" کرده ودر روی "rung" خاصی که در نظر داریم قرار بدهیم. هرکدام از "پله" ها و یا "شبکه" ها می تواند چندین ورودی داشته باشد ولی تعداد خروجی فقط می تواند از یک "خروجی" تشکیل شده باشد. شکل 59 را ملاحظه کنید. رانگ اول از دو ورودی IO.0 و IO.1 تشکیل شده ولی فقط یک خروجی Q0.0 در مدار داریم. "رانگ" دوم از یک ورودی IO.2 و خروجی "بینام" تشکیل شده است.

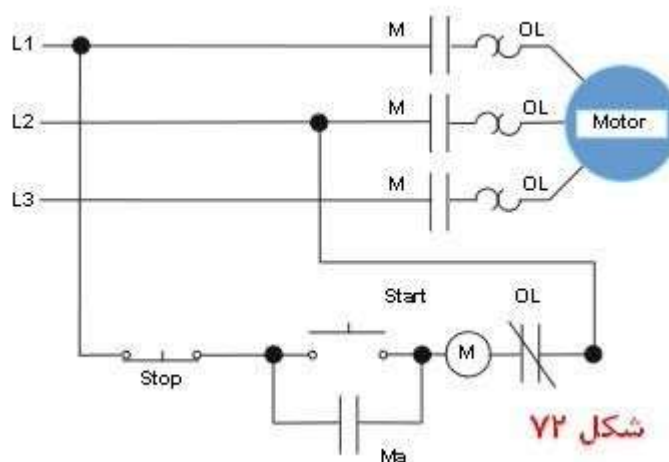
هر "رانگ" یا "شبکه" در یک برنامه نرده بانی نمایانگر یک عملکرد منطقی میباشد. برنامه شکل 60 نشان دهنده یک عملکرد AND منطقی میباشد. معنی و مفهوم برنامه شکل 60 بدین صورت عنوان میشود که اگر دو ورودی IO.0 و IO.1 فعال بشوند و یا درحالت ON قرار بگیرند ، خروجی نیز در حالت ON قرار خواهد گرفت . سه دستور بعدی در Network1 همان برنامه "اند" منطقی را نشان میدهد ولی برای تهیه آن از کدهای اسمبلی استفاده شده است که امروزه کمتر مرسوم است. و بخش بعدی نیز با استفاده از فرمت "Function block" همان برنامه "اند" منطقی باز نویسی شده . برنامه میکرو پی ال سی طوری تهیه شده که در عمل می تواند برنامه نوشته شده بزبان "Function Block" به فرمت نرده بانی تبدیل کند.



شکل ۶۰

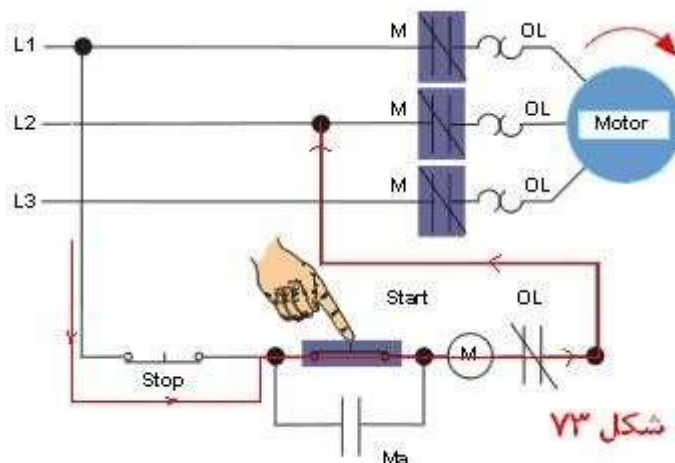
## ۶ دیاگرام سیم کش شده استارت کننده یک موتور سه فاز Hard-wired motor starter

شکل ۷۲ دیاگرام سیم کشی شده استارت و استوپ یک الکتروموتور سه فاز را نشان می‌دهد. در شکل فوق بوبین کنتاکتور بطور سری با پوش باتونهای استارت و استوپ سیم کشی شده اند. کنتاکت کمکی Ma بطور موازی با کلید " باز در حالت عادی " و کلید استارت قرار گرفته است. در شکل ۷۲ تا زمانی که کلید "استارت" فشرده نشده است، هیچ جریانی از سیم پیچ کنتاکتور عبور نکرده و لذا کنتاکتور فوق فعال و یا تحریک نخواهد شد، در نتیجه کنتاکت فوق جذب نشده و برق سه فاز به الکتروموتور اعمال نشده و بالطبع الکتروموتور نیز شروع بکار نخواهد کرد.

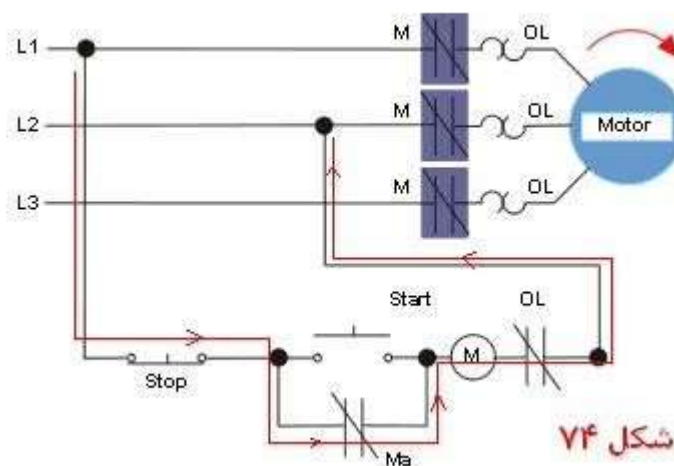


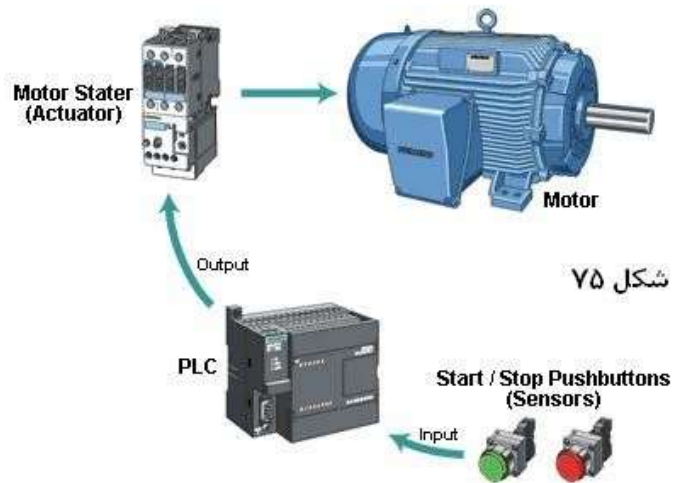
شکل ۷۲

در شکل 73 وقتی که کلید پوش باتن Start برای یک لحظه کوتاه فشرده می‌شود، جریان برق در مدار بسته نشان داده شده برنگ قرمز حرکت کرده و بوبین کنتاکتور را تحریک می‌کند. این تحریک باعث جذب شدن هسته درون کنتاکتور شده، برق سه فاز وارد سیم پیچی الکتروموتور می‌شود و موتور به کار می‌افتد.



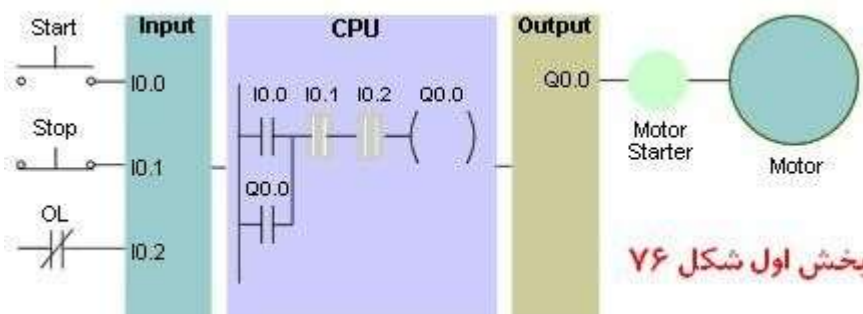
در شکل 74 ملاحظه می‌کنید که حتی پس از برداشتن انگشت از روی کلید Start، هنوز شفت موتور در حال حرکت است زیرا یکی از کنتاکت‌های باز کنتاکتور موازی با کلید Start قرار گرفته است و جریان پس از قطع شدن کلید استارت، کنتاکتور را بصورت تحریک شده نگاه می‌دارد. جریان برق در کنتاکتور پس از آزاد شدن کلید استارت در جریان می‌یابد و چون در مسیر برق کنتاکتور، هیچ تغییری حاصل نشده، موتور تا فشرده نشدن کلید "استاپ" فعال خواهد بود.



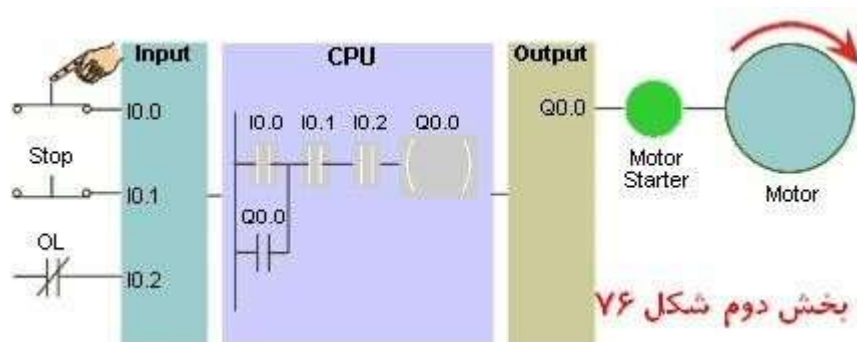


### ۶.۱ برنامه نویسی مدار شکل ۷۵

شکل ۷۶ سیم کشی مدار استارت / استوپ موتور سه فاز را نشان می‌دهد. همانطوریکه که در شکل ملاحظه می‌کنید، کلید استارت normally open start pushbutton به ورودی I0.0 پی ال سی، کلید استوپ normally closed stop pushbutton به ورودی I0.1 و یک کلید بسته در حالت عادی normally closed overload contacts به ورودی I0.2 وصل شده‌اند. هر سه این ورودی ها باید بصورت AND عمل کنند، لذا در شکل 76 ملاحظه می‌کنید که هر سه بطور سری در مدار قبل از خروجی کنتاکتور قرار داده شده‌اند. همچنین یکی از کنتاکتهای مرتبط با Q0.0 نیز بطور موازی با کلید استارت در نرم افزار مورد استفاده قرار گرفته است که با مدار بالای خودش تشکیل یک مدار OR را داده است. بخش بالایی شکل 76 مدار را در حالت غیر فعال نشان می‌دهد بدین معنا که هیچکدام از کلیدها فشرده نشده‌اند. در این صورت کلید پوش باتون " استارت " هنوز فشرده نشده و موتور در حالت استراحت است پس بدیهی‌ست که خروجی  $Q0.0 = 0 = \text{OFF}$  نیز خاموش است.

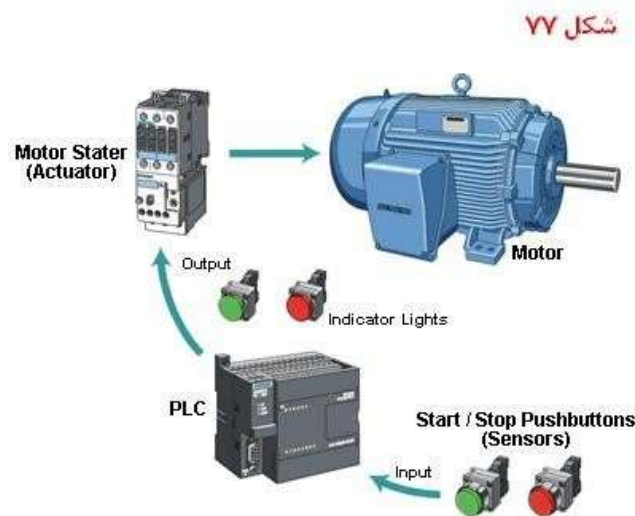


در بخش دوم شکل 76 با فشرده شدن کلید استارت مدار، "اند" بسته شده و عبور جریان سبب فعال شدن و یا روشن شدن خروجی  $Q0.0 = 1 = \text{ON}$  موتور شروع به حرکت می‌کند.

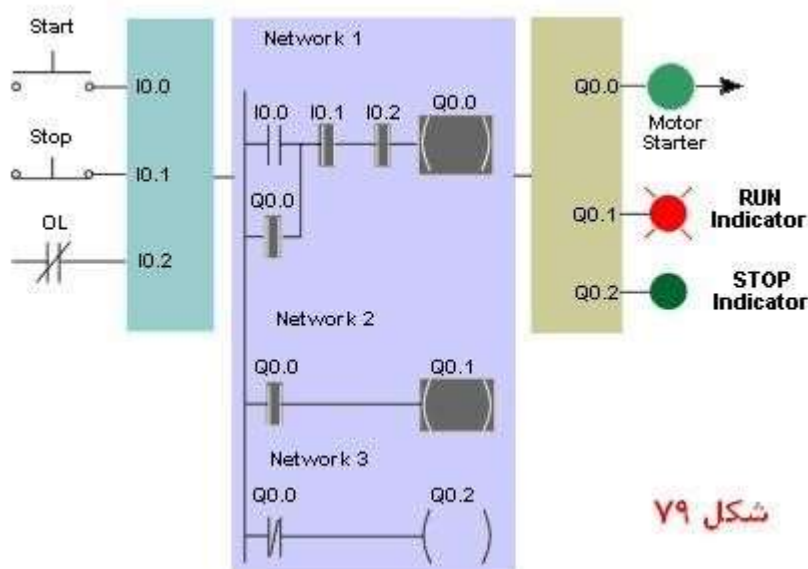
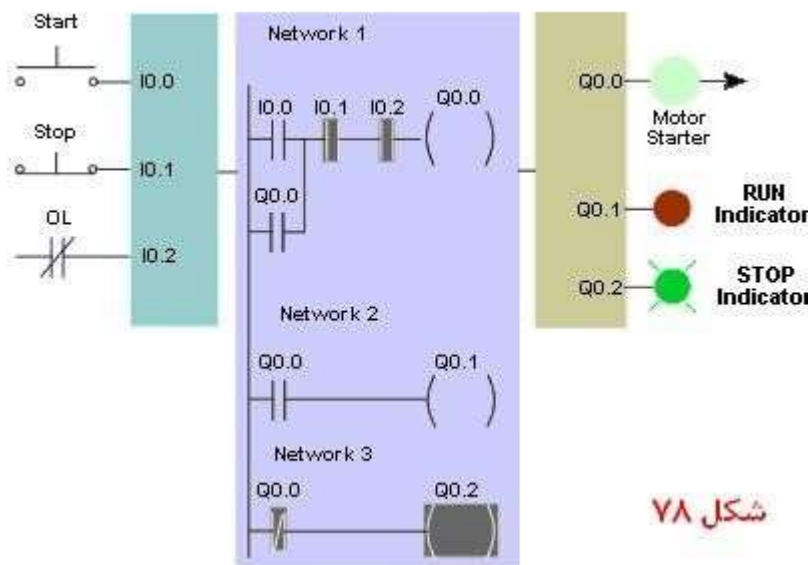


## ۶.۲ گسترش مدار شکل ۷۶

در صورت نیاز می‌توانیم مدار شکل 76 را با ۱ اضافه کردن دو عدد لامپ قرمز و سبز گسترش دهیم. می‌خواهیم مدار را طوری تغییر دهیم که وقتی کلید استارت فشرده شد، لامپ قرمز (Run indicator) روشن شود و در صورت خاموش کردن موتور، لامپ سبز (Stop indicator) روشن شود. شکل 78 را ملاحظه کنید.

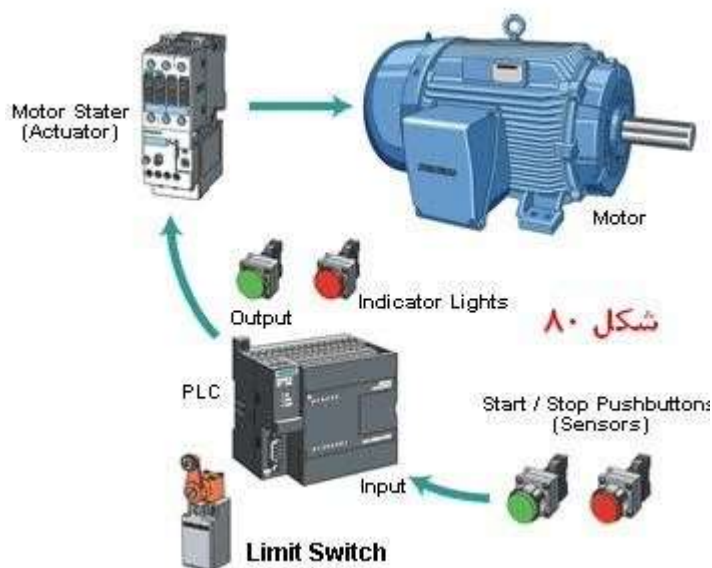


یک کنتاکت NO در شبکه دوم قبل از خروجی Q0.1 قرار داده شده است. خروجی فوق نشان‌دهنده RUN را کنترل می‌کند که در صورت روشن شدن، Q0.0 مربوط به کنتاکتور فعال خواهد شد. پس لامپ قرمز به موازات فعال شدن کنتاکتور، روشن می‌شود. در شبکه سوم نیز کنتاکت NC با مدار Q0.2 بطور سری قرار گرفته است و در صورت خاموش بودن Q0.0، لامپ سبز روشن می‌شود (خاموش بودن کنتاکتور).



### ۶.۳ اضافه کردن سوئیچ محدود کننده به مدار قبلی

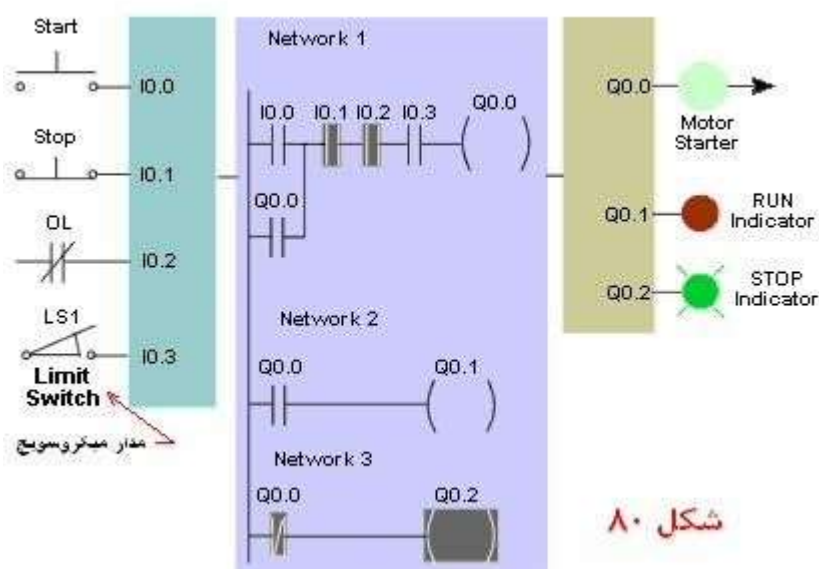
مطابق شکل ۸۰، می‌خواهیم یک میکروسوئیچ نیز به مدار قبلی اضافه کنیم. میکروسوئیچ نیز جز ورودی‌ها است لذا می‌توانیم ورودی شماره IO.3 را به آن اختصاص بدهیم. میکروسوئیچ مانند یک کلید باز یا NO عمل خواهد کرد (normally open contact).



شکل ۸۰

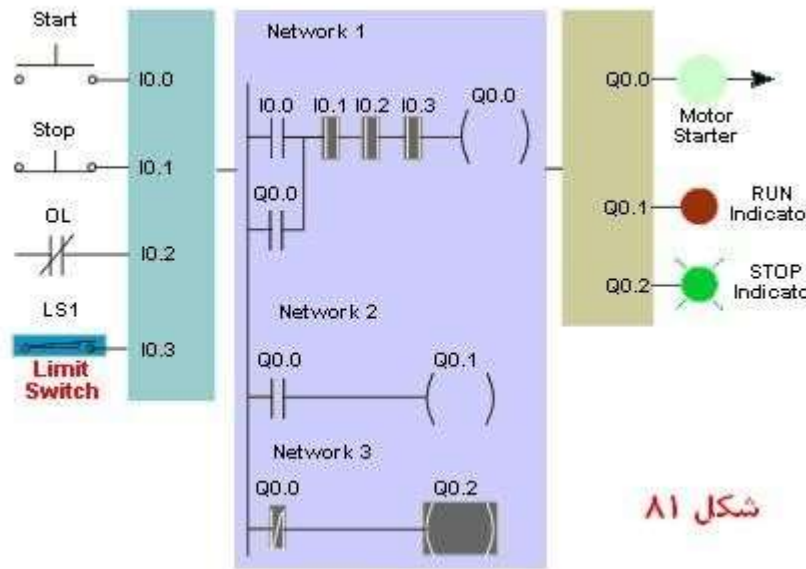
#### ۶.۴ ارایه نرم افزار نرده بانی برای مدار شکل ۷۸

در نرم افزار ارایه شده در شکل ۸۰ همانطوریکه ملاحظه می کنید در صورت فعال شدن یا فشردن کلید میکروسویچ، کنتاکت NO میکروسویچ به NC تغییر وضعیت خواهد داد و چون کنتاکت مربوطه با سه کنتاکت قبلی بصورت AND قرار گرفته پس در صورت فشردن میکروسویچ، موتور روشن خواهد شد. در شکل ۸۰ چون مدار میکروسویچ تحریک نشده، لذا ورودی I0.3 باز است و موتور در حالت توقف قرار داده شده و چراغ سبز رنگ متصل به خروجی Q0.2 روشن است.



شکل ۸۰

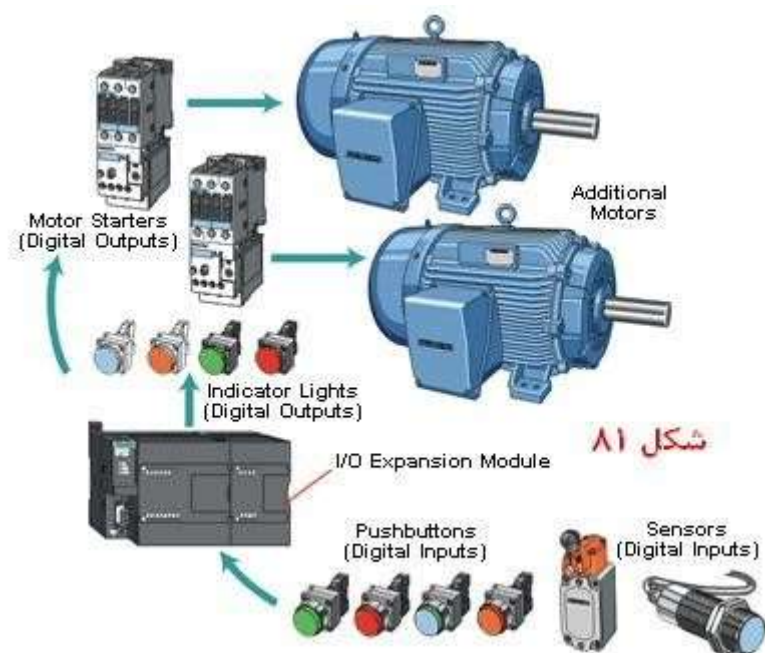
در شکل ۸۱ کلید میکرو سوئیچ تحریک یا فشرده شده و در نتیجه کنتاکت میکرو سوئیچ  $I0.3 = ON = 1$  در حالت روشن قرار گرفته است و اکنون با فشردن شدن ورودی استارت  $I0.0$  موتور روشن خواهد شد.



## ۶.۵ گسترش بیشتر مدار شکل ۸۱

در ادامه شکل ۸۱، قصد داریم سخت افزارهای بیشتری را به مدار قبلی اضافه کنیم. در عمل ما دو الکتروموتور داریم که توسط دو کنتاکتور فرمان داده می شوند. چهار لامپ نشان دهنده که فعال و یا متوقف بودن هر کدام از موتورها را نشان می دهند، چهار عدد پوش باتن که برای استارت و استوپ کردن موتورها بکار گرفته خواهند شد به همراه میکروسوییچ مدار قبلی بعلاوه یک سوئیچ حسی (proximity switch).

معمولاً عملکرد سوئیچ های حسی مانند میکروسوییچ ها است. این سوئیچ ها می توانند وجود یا عدم وجود یک جسم را تشخیص دهند. بدین صورت که در حالت عادی کنتاکت داخلی آنها بصورت NO است ولی در صورت قرار گیری جسم خاصی در مقابل این سوئیچ، به NC تغییر وضعیت می دهند. از بررسی شکل ۸۱ می توان نتیجه گرفت تا آنجاییکه تعداد ورودی / خروجی های یک پی ال سی اجازه می دهد، می توان ادوات مختلفی را به آن اضافه کنیم. پارامتر دیگری را که باید در گسترش دادن مدارات پی ال سی در نظر داشته باشیم مقدار حافظه داخلی پی ال سی است که معمولاً به اندازه کافی موجود است.



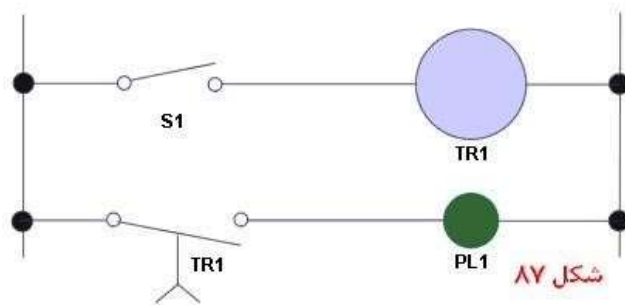
## ۶.۶ آشنایی با تایمر های پی ال سی ها ( Introduction to Timers )

تایمرها دستگاه‌هایی هستند که بوسیله آنها گذشت زمان را می‌توان اندازه‌گیری کرد. میزان گذشت زمان در چراغ‌های راهنمایی توسط تایمرها اندازه‌گیری می‌شود. شکل ۸۶ را ملاحظه کنید.

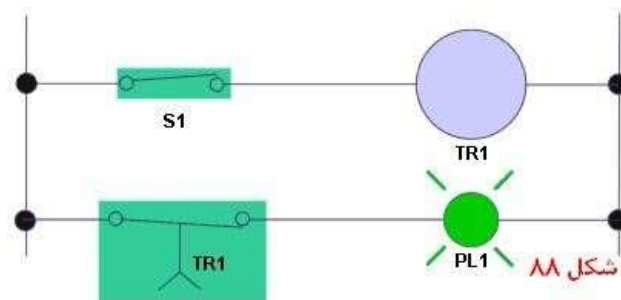


## ۶.۷ سیم کشی مدار تایمر معمولی Hard-wired Timing circuit

از نظر مداری عملکرد تایمر در پی ال سی ها بسیار شبیه تایمرهای معمولی است. در شکل ۸۷ در صورت بسته شدن زبانه باز کلید S1، تایمر TR1 که برای مدت 5 ثانیه تنظیم شده است شروع بکار می کند. این نوع تایمر از نوع ON delay timer است یعنی که پس از استارت شروع، باید مقدار زمان مشخصی بگذرد تا پس از اتمام زمان فوق تنظیم شده، تایمر عمل کرده و زبانه یا کنتاکت باز (TR1) بسته شود.



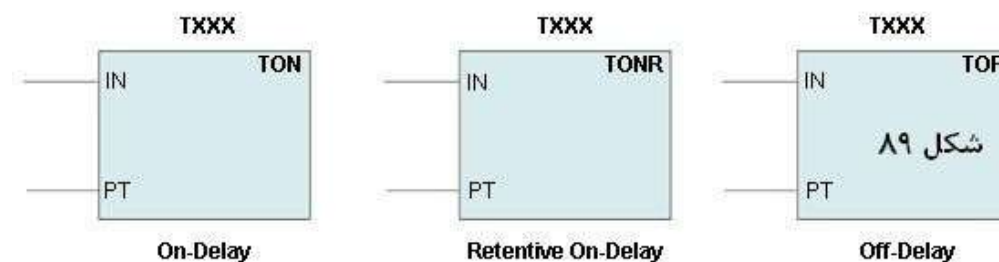
در شکل 88 ملاحظه می کنید که زمان اولیه تایمر را برای 5 ثانیه تنظیم کرده بودیم و با فشردن کلید S1، چیزی حدود 5 ثانیه طول کشید و سپس کنتاکت باز تغییر وضعیت داده و بسته شد. و چون کنتاکت با لامپ PL1 بطور سری قرار گرفته بود، لذا بسته شدن TR1 مانند یک کلید عمل کرده و سیم فاز را به یک سر لامپ وصل کرده و سبب روشن شدن لامپ گردید.



## ۷۲- تایمرهای پی ال سی S7-200

تایمرها با استفاده از سمبل های جعبه ای شکل در برنامه نویسی به روش نرده بانی مشخص می شوند. وقتی که یک تایمر سیگنال "فعال" (enable signal) دریافت می کند، شروع بکار کرده و عدد شمارش شده را دائما با "عدد PT"

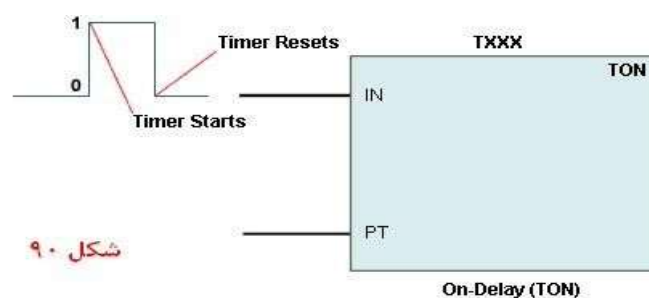
یا عدد تعیین شده (preset time) مقایسه می‌کند. خروجی تایمر مادامی که عدد شمارش شده کنونی مساوی با "عدد" تعیین شده نباشد، مساوی است با عدد باینری صفر و یا  $Low = OFF = 0$  ولی وقتی عدد شمارش شده مساوی یا بزرگتر از عدد "تعیین" شده می‌شود، خروجی به باینری وضعیت روشن  $High = ON = 1$  تبدیل می‌شود. در پی ال سی‌های خانواده زیمنس S7-200 سه نوع مختلف تایمر بنام‌های TON, TONR, TOF وجود دارند که جمع تعداد کل آنها 256 عدد می‌باشد که با دقت‌های یک میلی ثانیه (۴ تایمر)، ده میلی ثانیه (۱۶ تایمر) و ۱۰۰ میلی ثانیه (۲۳۶ تایمر) برای استفاده کننده در نظر گرفته شده است. اسامی این تایمرها عبارتند از 1- On-, Delay (TON) یا تایمر تاخیر در وصل 2- Retentive On-Delay (TONR), 3- تایمر تاخیر در قطع یا Off-Delay (TOF). تایمر TONR مانند تایمر TON عمل کرده با این تفاوت که در تایمر TON در صورتی که  $IN = 1 = ON$  تایمر مقدارش یک و مجدداً صفر شود تایمر TON از ابتدا زمان را محاسبه می‌کند ولی TONR از بقیه زمان باقی مانده، زمان را محاسبه می‌کند. در تایمر TOF بلافاصله بعد از یک شدن ورودی  $IN = 1 = ON$  خروجی تایمر  $ON = 1$  شده و پس از گذشت زمان مساوی یا بزرگتر از "عدد PT", خروجی به صفر تبدیل خواهد شد. شکل 89 که مربوط به سمبل‌های تایمرهای ذکر شده می‌باشد را ملاحظه کنید.



### ۷۳- بلاک دیاگرام عملکرد تایمر ON-Delay timer (TON)

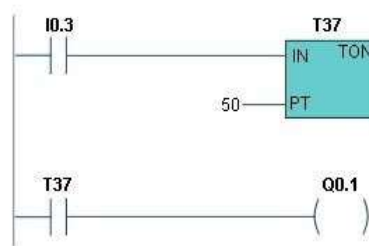
تعریف عملکرد تایمر, On-Delay (TON): زمانی که ورودی تایمر فوق فعال شود ( $IN = 1 = ON$ ) تایمر شروع به تایم‌گیری می‌کند و بمحض اینکه مقدار شمارش فوق بزرگتر یا مساوی "عدد PT" بشود، بیت تایمر فعال می‌شود. تایمر وقتی که ( $IN = 0$ ) به صفر تبدیل یا غیرفعال می‌شود، تایمر به عدد اولیه ریست می‌شود.

شکل 90 را ملاحظه کنید.

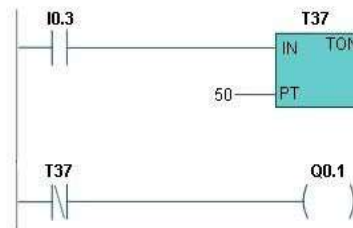


#### ۷۴- یک نمونه از برنامه کاربردی برای آشنایی با عملکرد تایمر ON-Delay timer (TON)

در مثال کاربردی شکل ۹۱، سوئیچ به ورودی I0.3 و یک لامپ به خروجی Q0.1 متصل شده است. وقتی که سوئیچ فعال می شود، I0.3 در حقیقت روشن شده و به باینری 1 تبدیل می شود که در این صورت تایمر T37 فعال می شود. "دقت یا Resolution" زمان تایمر فوق 100 میلی ثانیه بوده و زمان (PT) preset time مساوی 50 تعیین شده است. در این صورت تایمر پس از گذشت زمانی مساوی با 50 ضرب در 100 میلی ثانیه مساوی 5 ثانیه روشن می شود. در این صورت خروجی Q0.1 بعد از 5 ثانیه که از فشردن سوئیچ گذشت روشن خواهد شد.



شکل ۹۱

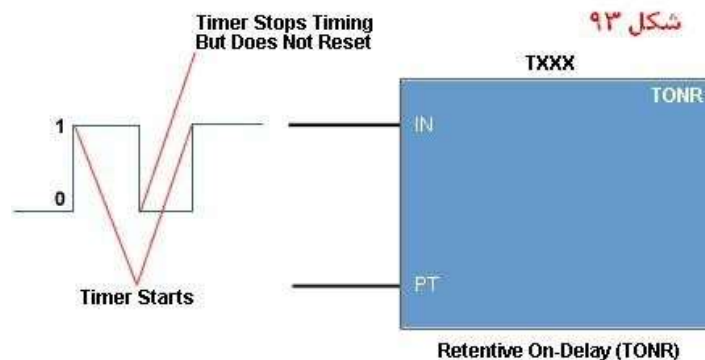


شکل ۹۲

در شکل ۹۲ با تعویض T37 که قبلاً NO بوده و جایگزینی آن با NC، خروجی Q0.1 بدون استارت خوردن ورودی روشن شده و با استارت خوردن پس از ۵ ثانیه خاموش می شود. ملاحظه می کنید که بدون تغییر هیچ سیم کشی یا تعویض کلید، می توانیم عملکرد مدار را عوض کنیم و این زیبایی استفاده از پی ال سی در مدارات کنترل است.

#### ۷۵- بلاک دیاگرام عملکرد تایمر Retentive On-Delay (TONR)

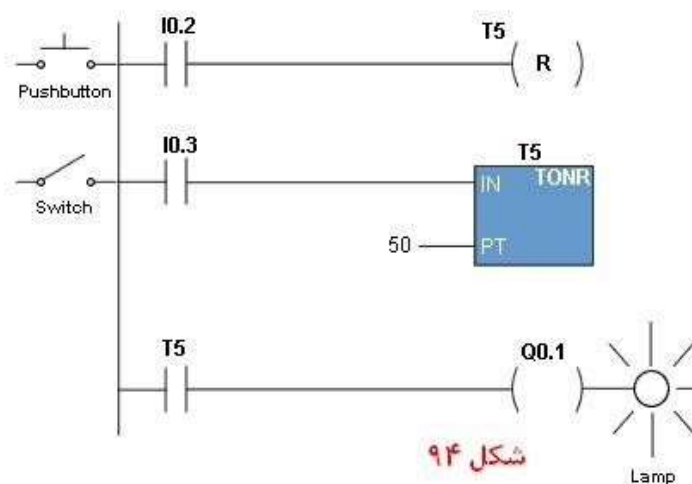
تایمر TONR مانند تایمر TON عمل کرده با این تفاوت که در تایمر TON در صورتی که  $IN = 1 = ON$  به عدد منطقی یک و صفر تبدیل شود و مجدداً یک شود تایمر TON از ابتدای زمان را محاسبه می کند ولی TONR از بقیه زمان باقی مانده قبلی، زمان فعلی را ادامه می دهد و در صورت اجرا دستور ریست (R) instruction، مقادیر شمرده شده تایمر ریست یا به عدد قبل از شروع، تنظیم می شود.



## ۷۶- یک نمونه از برنامه کاربردی برای آشنایی با عملکرد تایمر TONR

### Retentive On-Delay (TONR)

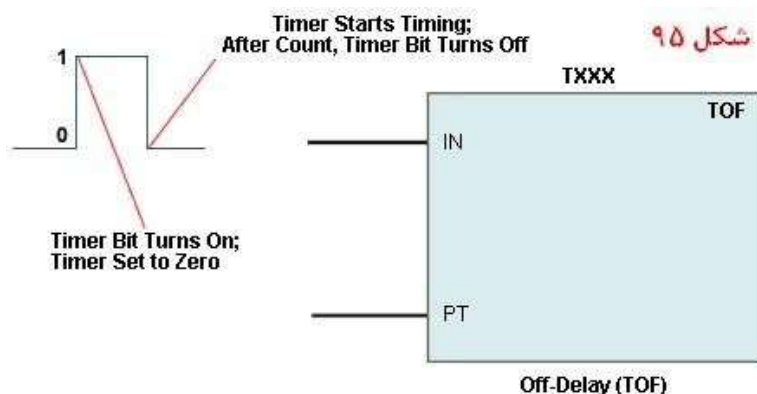
وقتی که در شکل ۹۴ ورودی یا سوییچ IO.3 بسته می شود، تایمر T5 شروع به اندازه گیری زمان می کند. اگر سوییچ فشرده شده و بعد از دو ثانیه رها یا باز یا غیر فعال گردد، عملکرد تایمر متوقف می شود. حال اگر سوییچ فوق دوباره فشرده شود، تایمر اندازه گیری زمان را از ۲ ثانیه بعد ادامه داده و زمان روشن شدن لامپ که قبلاً برای ۵ ثانیه بعد از فشرده شدن کلید ورودی تنظیم شده بود، حالا روشن می شود. بنا براین خاموش شدن تایمر پس از ۲ ثانیه اول سبب پاک شدن اندازه گیری زمان نشده و بعد از وصل مجدد کلید و اندازه گیری زمانی بمدت ۳ ثانیه در حالت دوم حال پس از ۵ ثانیه لامپ روشن می شود. با فشرده شدن کلید Pushbutton، وصل شده به ورودی IO.2 تایمر ریست reset خواهد شد. یعنی اینکه اگر پس از روشن خاموش شدن ورودی switch بعد از ۲ ثانیه تایمر را ریست کرده و دوباره ورودی switch تحریک کنیم، در این صورت ۲ ثانیه شمرده شده از حافظه تایمر "پاک" شده و هیچ اطلاعاتی در مورد ۲ ثانیه در حافظه تایمر باقی نخواهد ماند.



## ۷۷- بلاک دیاگرام عملکرد تایمر Off-Delay (TOF)

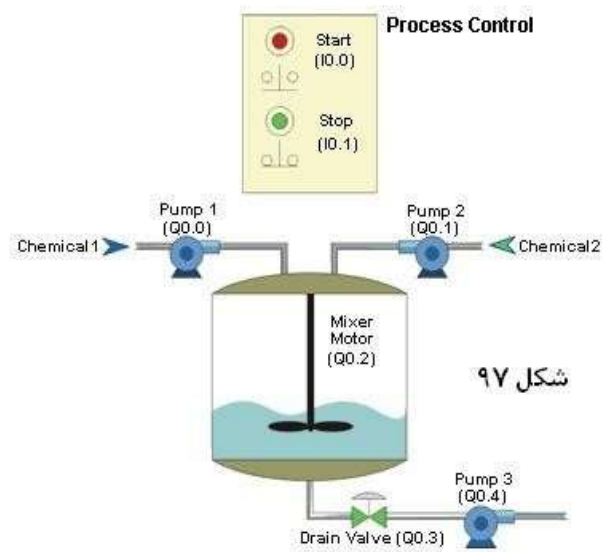
این تایمر مانند تایمرهایی که در راه پله های آپارتمانها مصرف می شود عمل می کند. بدین صورت که وقتی که ورودی تایمر برای یک لحظه روشن خاموش می شود، خروجی این تایمر به اندازه زمان تنظیم شده PT روشن شده و در خاتمه زمان فوق خاموش می شود. مانند چراغهای راه پله. که معمولا برای زمان به مدت 5 دقیقه تنظیم می شوند و این مدت معمولا کافی است که یک نفر از پله های طبقه همکف مثلا به طبقه پنجم برسد.

پس از اتمام زمان فوق، تمام لامپهای طبقات که بصورت موازی روشن بودند، خاموش می شوند.



## ۸۶- تمرین نمونه برای عملکرد تایمرها

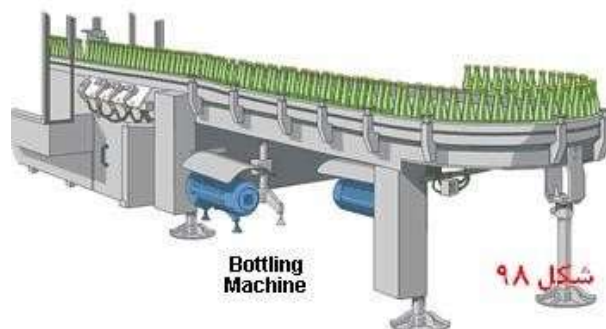
در شکل ۹۷ ملاحظه می کنید که یک تانک مواد شیمیایی داریم که از دو پمپ شماره های 1 و 2 مواد بداخل تانک جاری می شود. این دو مواد شیمیایی پس از مخلوط و هم زده شدن توسط پمپ شماره 3 تخلیه می شود. برنامه ای بنویسید که پس از فشار ورودی 0I.0 پمپ شماره 1 که خروجی Q0.0 را فرمان می دهد، شروع به کار کرده و به مدت 5 ثانیه کار کند. و پس از 5 ثانیه خاموش بشود. سپس پمپ شماره 2 شروع بکار کرده و بمدت 3 ثانیه کار کند تا مواد شیمیایی شماره 2 را بداخل تانک هدایت کند. سپس موتور مخلوط کننده (Mixer motor Q0.2) روشن شده و بمدت 60 ثانیه روشن باقی بماند تا 3 سابی دو مواد شیمیایی 1 و 2 را مخلوط کند. و سپس مخلوط کن نیز خاموش شود. سپس شیر خروجی (drain valve Q0.3) و بعد از آن بلافاصله پمپ شماره 3 با مشخصات (Pump 3 Q0.4) بمدت 8 ثانیه شروع بکار کند تا مواد را از تانک بیرون هدایت کند. در صورت نیاز می توانیم با فشار کلید یا سویچ استاپ دستی سیستم را خاموش کنیم که سویچ فوق به ورودی IO.1 وصل شده است.



شکل ۹۷

### ۸۷- شمارنده‌ها Counters

شمارنده‌های موجود در پی ال سی ها عملکردی مشابه شمارنده‌های مکانیکی دارند. این شمارنده‌ها عدد شمارش شده را با عدد "از قبل" ست شده مقایسه کرده و نسبت به اینکه عدد شمارش شده کمتر و یا مساوی و بزرگتر از "عدد" از قبل تعیین شده است، خروجی شان فعال می شود. معمولاً ما نیاز داریم یک عدد از قبل تعیین شده را (preset value) به شمارنده داده و می‌خواهیم شمارنده کمیتی را شمرده و وقتی که عدد شمارش به عدد تیظیم شده رسید، خروجی‌اش فعال گردد. بطور مثال شکل ۹۸ یک کانوایر انتقال بطری را نشان می‌دهد. در اینجا نیاز داریم که بطری‌های در گروه‌های ۶ تایی بداخل جعبه‌های مخصوص هدایت شده و سپس بطرف دستگاه بسته بندی هدایت بشوند.



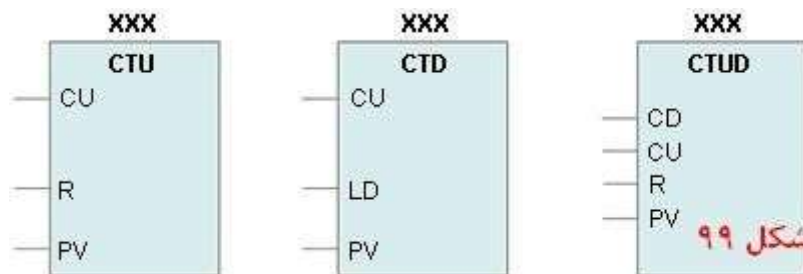
شکل ۹۸

### ۸۸- سمبل شمارنده‌ها در برنامه نویسی منطقی نرده بانی Ladder Logic

شمارنده ها نیز مانند تایمرها در برنامه‌های منطقی نرده بانی توسط سمبلهای چهارگوش نشان داده میشوند. در صورت تحریک شدن ورودی CU (Count up) کانتر یک واحد به "شمارش" قبلی اضافه می‌کند. در صورت تحریک شدن ورودی CD (Count Down) کانتر یک واحد از "شمارش" قبلی کم یا کسر می‌کند.

برای فعال کردن ورودی شمارش بالا یا پایین ، لازم است که سطح سیگنال باینری اعمال شده از 0 به ولتاژ 1 تغییر وضعیت بدهد (count input transitions from a logic 0 to a logic 1). در پی ال سی های سری Siemens 200- S7 بطور کلی میتوانیم از سه نوع شمارنده استفاده کنیم . که عبارتند از: 1- شمارنده های "بالا رونده" یا Up counters (CTU) 2- شمارنده های پایین رونده یا down counters (CTD) 3- شمارنده های بالا و پایین رونده یا up/down counters (CTUD). مانند تایمرها می توانیم تا 256 شمارنده در برنامه نوشته شده خود داشته باشیم که این شمارنده ها می توانند از شماره C0 تا C255 شماره گذاری بشوند.

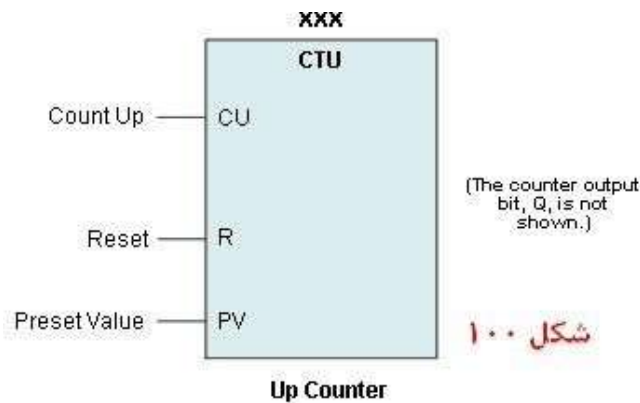
تذکار: دو شمارنده با یک شماره نمی توانیم در یک برنامه داشته باشیم . و بیشینه شماره قابل شمارش نیز عددی است بین 32767 +/- . شکل شماره 99 را ملاحظه کنید.



#### ۹۰- شرح عملکرد شمارنده بالارونده (CTU)

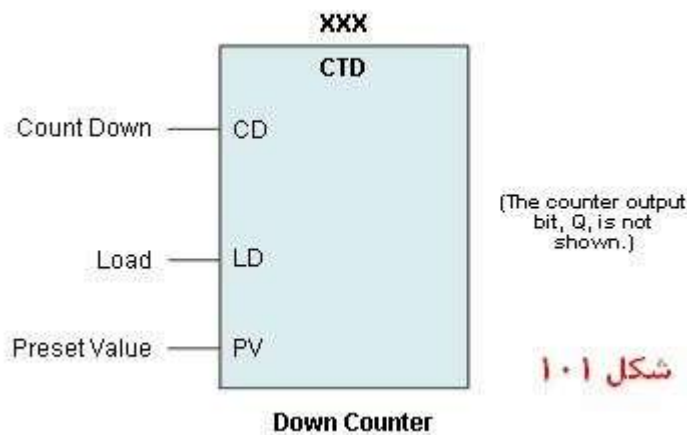
شمارنده بالا رونده در صورت تحریک شدن ورودی CU یک واحد به "شمارش کنونی اش" افزوده میشود تا مقدار عدد شمارش شده به عدد "از قبل" تعیین شده PV (preset value) برسد. هر زمان که ولتاژ سیگنال ورودی اعمال شده به پایه CU از عدد باینری صفر به یک افزایش پیدا می کند، به محتویات "شمارش کنونی" current value " یک واحد افزوده میشود. تحریک ورودی R سبب RESET ریست یا صفر شدن "عدد شمارش" شده میشود. با مساوی قرار دادن ورودی PV به هر عدد دلخواه دسی مال ، میتوان مقدار عدد

"از قبل تعیین" شده را تعیین کرد. در صورت مساوی یا بزرگتر شدن عدد "شمارش شده" با عدد از "قبل" تعیین شده ، خروجی (Q) output bit تغییر وضعیت از 0 به 1 و یا فعال خواهد شد. شکل 100 را ملاحظه کنید.



#### ۹۱- شرح عملکرد شمارنده پایین رونده Down Counter

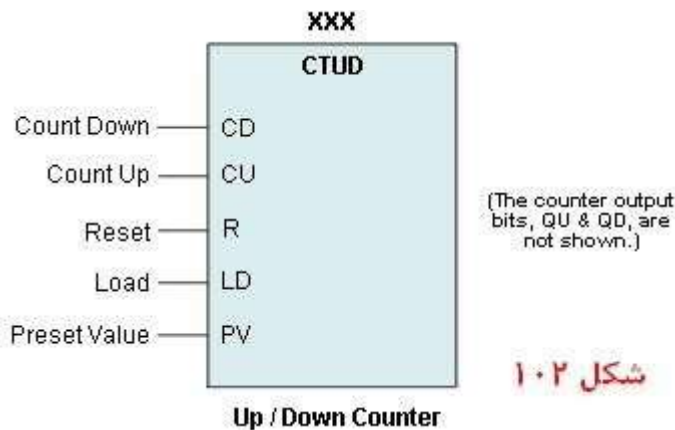
اگر مثلاً  $PV = 12$  قرار داده باشیم، هر زمان که ورودی CD از ولتاژ منطقی صفر به ولتاژ منطقی یک تغییر وضعیت می‌دهد (transitions from a logic 0 to a logic 1)، از عدد PV یک واحد "کسر" می‌شود و با داریم  $11 = 1 - 1$  (PV = 12). وقتی محتویات PV مساوی صفر می‌شود، 12 واحد شمرده شده، در این صورت خروجی (Q) output bit فعال می‌شود. در صورت تحریک شدن ورودی LD، شمارنده ریست شده و مقدار PV جایگزین مقدار "شمارش" کنونی می‌شود. شکل 101 را بررسی کنید.



#### ۹۲- شرح عملکرد شمارنده بالا و پایین رونده (S7-200 Up / Down counter)

این شمارنده به مقدار PV یک واحد هر زمان که ورودی CU توسط تغییر سیگنال از باینری صفر به باینری یک تغییر وضعیت می‌دهد "اضافه" می‌کند. وقتی که تغییر ولتاژ فوق به پایه ورودی CD اعمال می‌شود، یک واحد از عدد PV "کسر" می‌گردد. وقتی که مقدار شمارش شده کنونی مساوی مقدار PV می‌شود، خروجی تحریک می‌شود و یا در "وضعیت" روشن قرار می‌گیرد. وقتی که عدد شمارش شده "کنونی" مساوی صفر می‌شود، خروجی QD در وضعیت روشن قرار می‌گیرد output QD turns on. وقتی که مقدار شمارش "کنونی" مساوی مقدار از پیش تعیین PV می‌شود، خروجی QU تحریک

و یا روشن میشود output QU turns on. در صورت تحریک شدن ورودی (LD)، شمارنده، مقدار از پیش تعیین شده PV بجای "شمارش کنونی" بازگذاری می کند. به همان شکل در صورت تحریک شدن پایه R (reset input)، شمارنده ریست شده و مقدار "شمارش" کنونی مساوی صفر می شود. شمارش کردن شمارنده هر زمان که شمارش کنونی مساوی PV یا صفر می شود متوقف می گردد. توجه خروجی های QU و QD در شکل 102 در روی کانترا نشان داده نمی شوند.



### ۹۳- یک مثال کاربردی شمارنده ها Counter application example

طبق شکل 103 می توانیم از یک شمارنده بعنوان سیستمی که حساب تعداد ماشین های وارد و یا خارج شده از گاراژ را "نگاه میدارد" استفاده کنیم. در این صورت شمارنده باید هر زمان که یک ماشین وارد گاراژ می شود، به تعداد کل ماشین های موجود یک عدد اضافه کند و در صورت خارج شدن هر ماشین یکی کم کند. در این صورت سیستم باید هر زمان که بازوی آهنی به بالا حرکت می کند یعنی یک ماشین در حال وارد شدن به گاراژ است و لذا از ظرفیت کل گاراژ یکی کم می شود. و اگر کل ماشین هایی که می توانند در محوطه پارکینگ پارک کنند 150 عدد باشد، در این صورت هر دفعه گیت بالا می رود یک عدد از عدد 150 کم می شود. در نهایت اگر ظرفیت "تکمیل" بشود باید تابلوی علامت "ظرفیت تکمیل است" روشن شود. در صورت خارج شدن هر ماشین، یک عدد از جمع کل کم خواهد شد.



## ۷ برنامه نویسی نرده بانی برای کنترل ایاب و ذهاب ماشین ها در گاراژ

از شمارنده C48 که یک شمارنده بالا و پایین رونده است در این برنامه استفاده خواهیم کرد. یکی از سیم های سویچ که به بازوی ورودی گاراژ وصل شده بعنوان ورودی به پی ال سی وصل شده است Input IO.0. سویچ دوم به بازوی مکانیکی که در مقابل "درب" خروجی است در مدار قرار گرفته و سیم آن به ورودی دیجیتال به شماره input IO.1 مدار پی ال سی وصل شده است. ظرفیت کل پارکینگ مساوی 150 است پس در این صورت عدد  $PV = 150$  خواهد شد. تابلوی با علامت "ظرفیت تکمیل است" به خروجی (Q0.1) Output 2 وصل شده. در صورت روشن شدن چراغ، تابلوی فوق دیده خواهد شد و در این صورت ماشین ها آمدن به طرف در ورودی را باید متوقف کنند. پس هر زمان که ماشینی وارد گاراژ می شود، ورودی IO.0 تحریک خواهد شد و در صورت خارج شدن ورودی IO.1. در صورت مساوی شدن تعداد "شمارش کنونی" با عدد 150، خروجی Q0.1 فعال خواهد شد. در این صورت به محض اینکه یک ماشین از گاراژ خارج بشود، یک واحد از عدد 150 کسر شده و چراغ Q0.1 خاموش خواهد شد. ورودی IO.2 شمارنده را ریست خواهد کرد. برنامه نوشته شده فوق را در شکل 104 ملاحظه کنید.

